When Random Forests Meet Neural Networks : A Finite-Sample Analysis

Ludovic Arnould

Sorbonne Université
LPSM

Doctoral School **École Doctorale Sciences Mathématiques de Paris Centre**

University Department **Laboratoire de Probabilités, Statistique et Modélisation**

Thesis defended by **Ludovic Arnould**

Defended on **October 20, 2023**

In order to become Doctor from Sorbonne Université

Academic Field **Applied Mathematics**

Speciality **Statistics**

When Random Forests Meet Neural Networks

A Finite-Sample Analysis

Thesis supervised by	Gérard Biau	Supervisor
	Claire Boyer	Co-Supervisor
	Erwan Scornet	Co-Supervisor

Committee members

Referees	Sylvain Arlot	Université Paris Saclay	
	Jason Klusowski	Princeton University	
Examiners	Sylvain Le Corff	Sorbonne Université	Committee President
	Florence D'Alché-Buc	Télécom Paris	
	Gilles Louppe	University of Liège	
	Robin Genuer	Université de Bordeaux	
Supervisors	Gérard Biau	Sorbonne Université	
	Claire Boyer	Sorbonne Université	
	Erwan Scornet	Sorbonne Université	

Colophon

Doctoral dissertation entitled "When Random Forests Meet Neural Networks", written by Ludovic Arnould, completed on January 16, 2024, typeset with the document preparation system LaTeX and the yathesis class dedicated to theses prepared in France.

This thesis has been prepared at

Laboratoire de Probabilités, Statistique et Modélisation
Sorbonne Université
Campus Pierre et Marie Curie
4 place Jussieu
75005 Paris
France

☎ +33 1 57 27 93 16
Web Site `https://www.lpsm.paris/`

Veni, vedi, scripsi.

Les hommes passent, les institutions demeurent.

Mon Fromager

Remerciements

Il y a fort peu longtemps, au royaume Chtatistik, un homme combattait, en quête de gloire, de richesses, mais surtout, de savoir. Seul ? Non. Il était accompagné, épaulé, tiré, tracté, porté même parfois, par une foule d'individus, au premier rang desquels ses encadrants, qu'il se doit de remercier ici. Ces remerciements sont pour moi l'occasion de rendre hommage à toutes celles et ceux qui ont rendu cette thèse possible, joyeuse et riche en enseignements, à celles et ceux qui m'ont permis de m'épanouir avant, pendant, et, je l'espère, c'est un vœu que je forme, après la thèse.

On entend souvent des doctorants se plaindre de leur isolement, de l'absence de leurs encadrants : cette pensée ne m'a jamais traversé l'esprit une seule seconde. Claire, Erwan, merci d'avoir consacré tout ce temps et cette énergie à me guider, à orienter mes lectures, à remettre sur le droit chemin mes errements mathématiques, à corriger inlassablement mes innombrables fautes rédactionnelles. Merci de m'avoir montré l'importance de la rigueur mathématique et rédactionnelle que vous avez poussée à des sommets que je n'apercevais même pas avant mon entrée en thèse. Merci pour votre dynamisme et votre passion de la recherche dont je garderai pour toujours des traces en moi, même si je n'en fais pas le cœur de mon activité professionnelle. Claire, merci de m'avoir montré que l'on peut toujours chercher à tisser des liens avec des objets mathématiques éloignés, que l'on peut clarifier encore et encore le discours et la rédaction pour rendre la transmission de l'information la plus efficace et agréable possible, que l'on peut, enfin, faire rentrer 12 journées de travail en 5 lorsque l'on est rigoureux, organisé et que l'on aime ce que l'on fait. Erwan, merci de m'avoir transmis l'amour des forêts, de m'avoir montré l'art de prendre du recul sur un calcul pour en dégager du sens et des voies de sortie, merci pour ta fine intuition mathématique et pour ton approche tranquille et efficace du travail qui m'ont beaucoup inspiré.

Merci à Patrick Lutz d'avoir été un collaborateur très doué et très aimable. Merci à Benjamin Guedj de m'avoir accueilli chaleureusement à UCL le temps d'un trop court séjour. Merci Gérard d'avoir toujours été aussi sympathique et merci pour ta réactivité infaillible qui mériterait la création d'une nouvelle unité de temps : la biauseconde.

Merci à mon jury d'avoir consacré du temps et de l'attention à mes travaux. Thanks to Jason Klusowski for reviewing my thesis. Merci à Sylvain Arlot pour son travail colossal de lecture et d'évaluation de ma thèse. Sylvain, personne avant toi n'avait lu mes travaux avec autant d'attention, merci du fond du cœur d'avoir fait cet effort, d'avoir aussi rigoureusement évalué ma thèse et de m'avoir permis d'améliorer ce manuscrit.

J'ai trouvé au labo des conditions matérielles idéales pour réaliser ma thèse, partir en conférence, télé-travailler, et sans cela la qualité de mon travail en aurait souffert. Je dois donc remercier le secrétariat, et avant tout ma gestionnaire Nathalie Bergame, ainsi que l'informaticien

Hugues Moretto, dont la réactivité et l'efficacité ont été sans faille pendant les trois années que j'ai passées au LPSM.

Merci aussi aux personnes qui font le ménage dans nos locaux et qui, malgré des conditions de travail difficiles dénoncées maintes fois par les syndicats, arrivent toujours à nous sourire en nous disant bonjour. Merci aux gardiens de Jussieu qui font des clins d'œil sans demander l'ouverture de mon sac quand j'arrive à vélo. Merci aux syndicats de doctorants, d'étudiants, et de permanents qui donnent de leur temps et de leur énergie pour défendre l'amélioration (en ce moment, la non-détérioration) de nos conditions de travail, d'étude et de recherche.

Merci au codeur anonyme de StackOverflow, au détricotteur de bugs, au sauveur de programme, à l'optimisateur de boucles `for`, au chasseur de badges dorés et de réponses validées, merci à toi qui m'a expliqué comment fonctionnait le parallélisme en Python, merci à tous les bienfaiteurs invisibles rédacteurs de librairies OpenSource (Python, Numpy, Sklearn, Pytorch, etc), merci au mathématicien médaillé Fields qui raconte ses découvertes mathématiques sur son blog ou sur MathOverflow, au professeur à la retraite qui m'a expliqué comment cette somme aux termes extra-terrestres pouvait s'évaluer par analogie avec la distribution gaussienne, au doctorant qui prend le temps d'expliquer à un jeune étudiant comment résoudre cette équation qu'il connaît par cœur et qui lui permettra de finir son devoir maison à temps.

J'aimerais aussi remercier toutes celles et ceux qui m'ont précédé sur le chemin sinueux des mathématiques et de la recherche, qui l'ont défriché et balisé pour moi, qui m'ont initié à la beauté de ses paysages.

Je dois avant tout remonter à mon professeur de mathématiques de collège, M. Bisson, dont les expressions loufoques égayaient nos cours et nos raisonnements laborieux. Mon attrait pour les mathématiques doit aussi beaucoup à mon professeur de première qui aurait souhaité nous voir démontrer géométriquement, équipés d'un compas, d'un caillou et d'un dé à coudre, l'irrationalité de pi en question bonus de DM. Plus tard, j'ai eu la chance de bénéficier, à Sorbonne Université, de l'enseignement de professeurs aussi bons mathématiciens que bons pédagogues, parmi lesquels Frédéric le Roux en Topologie, Thierry Lévy en Probabilités, Gérard Biau et Maxime Sangnier en Apprentissage statistique, Etienne Roquain en Statistiques en grande dimension, Ismaël Castillo en Statistiques Bayésiennes, Jean-Yves Chemin en Analyse fonctionnelle, Eddie Aamari en Inférence géométrique, Anna Ben-Hamou en Inégalités de concentration.

Enfin, mon parcours en "sciences asociales" aurait sûrement été beaucoup plus court sans le soutien de Fabien Moutarde, sans lequel ma carrière aurait bifurqué vers d'obscures contrées juridiques, et sans lequel je n'aurais pas été initié aux promesses du Machine Learning qui continue de m'émerveiller, encore et encore.

Ces trois années de thèses n'auraient pas eu la même saveur sans la présence de nombreux amis, camarades et collègues au sein du laboratoire et dans les nombreux lieux qui m'ont servi, au hasard d'une heure, d'une demi-journée ou, en cas d'épidémie mondiale, de mois entiers, de bureaux éphémères. Je tiens donc d'abord à remercier chaleureusement, dans son ensemble, tout le laboratoire du LPSM où j'ai pu trouver des conditions idéales pour mener à bien ma thèse, à la fois sur les plans scientifiques et humains. Cet endroit restera toujours pour moi un sanctuaire d'ouverture d'esprit, de curiosité intellectuelle et de bienveillance.

Ce caractère sacré tient beaucoup à la présence infaillible des *permanents* qui apportent chaque jour, au labo, un lot de bonne humeur (oui même toi Antoine), de vie scientifique et de mots croisés diaboliques. Merci aux éternels de la salle café, aux organisateurs du séminaire de stats, aux finisseurs de mots croisés, aux jeunes, aux moins jeunes, aux joueurs de pétanques communistes, aux amoureux du théâtre et de la littérature, aux cinéphiles, à Anna, Antoine,

Arnaud, Badr, Claire, Eddie, Erwan, Etienne, Ismaël, Maxime, Olivier, Stéphane, Sylvain et à tous les autres. Parmi les innombrables vertus des permanents, j'aimerais souligner leur capacité manifeste à recruter d'excellents doctorants et stagiaires qui essayent toujours plus, chaque jour, par leurs efforts acharnés, de se hisser à leur hauteur.

En comparaison aux récits de vie de labo d'autres doctorants de France et de Navarre rencontrés en conférence, je dirais que la particularité du LPSM doit beaucoup à son immense cortège de doctorants dont un de ses millésimes les plus nobles fut, sans surprise, celui de l'année 2020. Quels arômes goute-t-on dans ce grand cru aux nombreux cépages ? Au nez d'abord, on ressent de la jeunesse, de la diversité, de la tolérance, de l'intelligence, de la joie. Nous buvons une gorgée, et il nous faut alors nous prêter au difficile exercice de séparer et d'isoler cette infinie diversité d'arômes, de réussir à dire quelque chose de plus que "hhhm très bon. Minéral ? Agrumes ? ", de rendre leur juste part, enfin, à celles et ceux qui ont donné son bon goût à ce vin. Merci Miguel pour ton amitié et ton énergie débordante, pour avoir fait briller dans la nuit nos soirées aux quais, pour m'avoir montré par la fenêtre un univers qui m'était méconnu : je parle bien sûr des pommes au piment, des concombres au piment, des chips au piment, des shoots de piment, bref du monde caché derrière cet ingrédient magique qui réchauffe les lèvres et qui fait frétiller les papilles gustatives. Merci Ariane et Iqraa d'avoir magnifiquement contribué à égayer la salle café et toutes les pauses qui s'y déroulèrent par vos rires incessants. Merci Ariane de m'avoir littéralement montré la voie (la bleue, à l'escalade) et de m'avoir initié à l'infinie diversité de la préparation des pâtes dont je sens que je n'ai effleuré que la partie visible de l'iceberg. Merci Iqraa d'avoir pris soin de ma plante (devrais-je dire de l'avoir ressuscitée ?) pendant ces 3 années. P.S. : Il te reste 45 minutes pour l'emballer dans un paquet-cadeau et pour changer son prénom auprès de l'état civil avant de me la rendre ! Merci Alexis d'avoir allégé, par ton détachement, tes petites blagues, et tes publications d'article auxquelles nos encadrants ont consacré une partie de leur temps, l'écrasant fardeau de la piété filiale que je portais seul avant ton arrivée. Merci Francesco pour ta bonne humeur que je n'ai jamais vue céder, pour ton approche très apaisée et très saine des problèmes du quotidien et de la société en général. Merci Antonio d'avoir toujours dénoncé haut et fort tes injustices et tes indignations, pour ton auto-dérision qui m'a toujours fait rire et pour avoir adouci l'aridité des bureaux du LPSM en présentant quotidiennement à nos yeux un assortiment de couleurs très satisfaisant à regarder. Merci Pierre pour les quelques discussions scientifiques que nous avons eues et que j'aurais souhaité plus nombreuses, pour ta culture et ton envie de chercher que j'admire. Merci Lucas pour ta bonhomie, tes clins d'œil charmeurs auxquels il a été dur de résister pendant 3 ans, et merci d'avoir insisté pour me faire découvrir les plaisirs de l'escalade. Merci Grace d'être toujours aussi chaleureux, aussi souriant et attentionné, de ne pas hésiter à venir dans notre bureau pour poser des questions sur les Random Forest ou simplement pour prendre des nouvelles. Asante sana ! Merci Mathis pour ton approche très optimiste de la vie, ton amour de la musique et du soleil, pour les bonnes adresses près du canal Saint-Martin. Merci Paul d'avoir été une agréable compagnie à Vienne et de nous avoir aussi chaleureusement accueillis à dîner avec Camila. Merci Yazid d'avoir apporté, pour un temps trop court, de l'humour, de l'intelligence et une belle soutenance de thèse au LPSM. Merci Claire d'être souvent passée prolonger nos pauses, d'être aussi engagée contre les injustices et de partager cet engagement avec moi. Merci Adeline de m'avoir si bien accueilli au LPSM et dans notre bureau malheureusement trop succinctement partagé, d'être toujours aussi pleine d'énergie et de bonne humeur.

Merci aux camarades qui aux creux des lits font des rêves d'un monde meilleur, à ceux, dont je tairais le nom, qui m'ont accompagné en manifestation défendre le rejet de la LPPR, le rejet de la réforme des retraites, la démission du gouvernement actuel, l'abolition des banques, du capitalisme, du travail, et l'avènement d'un règne éclairé de doctorants, sans contre-pouvoir.

Un mot aussi, pour mes illustres prédécesseurs dans les pas desquels j'inscris les miens en

soutenant ma thèse, Gloria, Nicklas, Thibault, Joseph et Kimia, que j'aurais souhaité connaître plus longtemps.

Que restera-t-il une fois la bouteille finie, une fois que la particularité de chaque goût se sera progressivement diluée dans le vaste brouillard de la mémoire, et que plusieurs condensations successives auront concentré toutes les sensations de la dégustation en quelques images, quelques sonorités, quelques visages, en quelques gouttes qui formeront dans mon esprit l'essence de "ma thèse au LPSM" ? Des réminiscences joyeuses, l'image d'un sourire et l'idée d'avoir passé ici trois des plus belles années de ma vie.

Enfin il reste un être au LPSM qui m'est singulier et qui incarne mes souvenirs les plus heureux de ces années. Merci Camila d'avoir supporté mon snobisme, mon espagnol trébuchant, ma samba bancale et mes Nocturnes boiteux, mes critiques et mes râlements intempestifs, mes multiples maladies rares et mes cancers de l'estomac, du dos, des ongles et des tendons ; merci d'être venue partager avec moi la froideur des légumes anglais et la chaleur de la cuisine indienne, à Londres et à Dishoom, de m'avoir pris dans ta valise pour découvrir l'imposante et la magnifique Vienne, d'avoir partagé avec moi tous ces beaux moments dans les meilleurs restaurants et bars à cocktail de Paris (souvent situés près du 14 rue Bouchardon), d'avoir admiré avec moi les plus belles vierges à l'enfant et les plus belles collines du monde, en Toscane ; merci de m'avoir cédé la place de ton chat dans ta chambre, d'avoir entretenu en permanence le stock de pesto et de parmesan de chez toi ; merci d'avoir pris soin de moi, merci d'avoir partagé avec moi ces jours que ta présence a rendus plus joyeux, merci, en somme, pour ton sourire et ton amour qui ont coloré ma vie.

Tout au long de ma vie, j'ai rencontré des gens qui m'ont accepté et aimé tel que j'étais, avec lesquels j'ai ri, pleuré, appris, joué au foot, bu des verres, étudié à la bibliothèque et joué aux jeux vidéos ; des êtres intelligents, bienveillants, drôles, attentionnés, généreux, ouverts, passionnés, en un mot, exceptionnels, qui m'ont apporté beaucoup de bonheur et d'émerveillement, et que j'ai la chance et la fierté de compter aujourd'hui parmi mes amis les plus chers.

Yasmine, Solange, Hugo, je pense d'abord à vous en écrivant les lignes qui précèdent. Je ne sais comment décrire proprement ni le plaisir que je prends à passer du temps avec vous, ni toute l'admiration que j'ai pour vous, pour vos immenses qualités qui font de vous les amis avec lesquels j'aimerais égoïstement tout faire et auxquels j'aimerais tout raconter, qui font de vous des médecins, avocats et Administrateurs de l'État brillants et des sources intarissables de joie et d'inspiration, pour moi et pour vos proches. Merci pour votre désir de partager avec moi tout ce que vous apprenez et ce que vous accomplissez, avec intelligence et modestie. Votre présence à mes côtés est une des raisons premières pour lesquelles j'apprécie autant la vie. Vous êtes pour moi ce que sont la boussole et la gourde remplie de bon vin à l'explorateur perdu au milieu du désert.

Afin de m'assurer un poste de ministre de l'Enseignement supérieur, de la recherche et de la vérité vraie, je commencerai par clamer toute l'admiration que j'ai pour mon ami de lycée Hugo Roussel, énarque, cuisinier, marathonien, juriste, clown à ses heures perdues, et, surtout, futur Président de la République. Hugo, je ne vais pas m'épancher sur ton CV qui ferait rougir de honte ce manuscrit et sur lequel il faudrait écrire un livre entier, mais j'aimerais quand même insister sur ton exceptionnelle rigueur, tes capacités à étudier et à t'investir à fond dans tout ce qui te plaît, des finances publiques à la cuisine, et à réussir tout ce que tu entreprends. Merci de montrer l'exemple vertueux, de m'avoir traîné aux examens de HSK3, puis de HSK4 (gardons les 5 et 6 pour nous occuper à la retraite), aux conférences du Louvre (trop vite avortées pour cause de pandémie), à la conférence Olivaint, de m'enseigner des astuces de cuisine. Merci de me faire toujours autant rire et merci de t'investir autant dans notre amitié.

Yasmine, j'ai l'impression que nous avons eu une connexion particulière et très rapide dès la seconde, qui tient sûrement à cet indicible dans ton attitude qui entraîne les gens à te confier leurs désirs secrets et leurs regrets cachés pour trouver une juste compassion et des conseils éclairés. Merci pour ta capacité à toujours aborder posément un problème ou une situation et à réussir à faire un pas de côté pour l'envisager sous un nouvel angle, qui a souvent apporté un éclairage nouveau sur ma vie et qui contribuera à faire de toi une excellente avocate dont je saurai exploiter les qualités (SCI Chalet des Thoules, BackMarket, on vous aura un jour !). Merci de rire à mes blagues douteuses, merci pour ta tolérance et ta bienveillance qui te permettent de me comprendre même lorsque je suis perdu dans les abîmes les plus sombres.

Solange, bien que tu sois la seule dont je ne puisse tirer des bénéfices directs au travers du métier (pourquoi avoir choisi, parmi les milliers d'organes et de parties du corps sur lesquelles on peut se spécialiser en médecine, la seule que je ne possède pas, l'utérus ?), je vais quand même chanter tes louanges — qui n'en seront que plus sincères. Avant tout, j'aimerais souligner ton énergie, ta force, ta soif de profiter de la vie qui te maintiennent debout, souriante et prête à donner de toi-même après un trajet de 86 heures entre Dax et Sète en passant par Strasbourg à vélo, par Munich en bus, Marseille en train, Montpellier en bateau puis Sète en blablacar, après deux gardes de nuit d'affilée à l'hôpital, après trois jours de festival sauvage de techno-psy-trance au milieu de la forêt. Merci pour ta passion inépuisable de la vie et de l'humain, qui sans doute t'a conduite à choisir l'un des métiers les plus altruistes et les plus contraignants, qui te pousse à t'intéresser à tout, à lutter contre les injustices, à créer du lien social et, plus important encore, des situations favorables à la création de lien social (je parle bien de l'installation de guirlandes dans tous les endroits que tu visites), à t'intéresser aux émotions des autres et à si bien les comprendre.

Solange, Yasmine, Hugo, je tiens tout autant à vous individuellement qu'à la magnifique alchimie qui se dégage de notre groupe. Puisse-t-elle durer éternellement.

Merci Rémi de n'avoir jamais abandonné ton naturel, dont les conséquences parfois désastreuses ne sont aujourd'hui que des souvenirs très drôles, qui a toujours apporté une bouffée d'air frais, de légèreté et d'intelligence spontanée à ton entourage. Merci d'avoir été aussi attentif, aussi attentionné, merci pour ton humour, merci pour les infinies heures de jeu et de discussion que nous avons passées à Paris, au 10, au 36, rue Bouchardon, à Banyuls, au moulin, au chalet, à Ydra, pour toutes ces heures que nous avons passées, souvent jusqu'à très tard dans la nuit, à s'amuser, à festoyer et à découvrir les richesses et les beautés que le monde a à offrir lorsqu'il est exploré en compagnie d'un ami. Merci d'avoir été, avec Pierre, ma seconde famille, merci d'avoir été pour moi un frère.

Merci Rami d'être toujours aussi amical, accueillant et énergique, merci pour ton humour sans égale qui nous fait tous beaucoup rire et qui révèle une compréhension très fine de la nature humaine, de nos désirs, de nos peurs et de notre mythologie collective.

Merci Elise, Lila, Noa, Rémi, Rami, Alfred, Arthur, Louis, la "bande de Répu" de m'accueillir toujours aussi chaleureusement et amicalement que si l'on venait de se quitter la veille, merci pour votre naturel, merci de m'avoir aidé à grandir aussi bien entouré, merci de faire encore éclore ces bourgeons d'amitié, de jeunesse, de joie, d'intelligence et de convictions qui ont si bien fleuri pendant notre enfance et que j'ai de plus en plus de mal à cultiver en me confrontant à l'aridité du monde.

Merci Alice pour ton approche tranquille et optimiste du monde qui te permet d'affronter, avec humour, les déconvenues du quotidien et la détresse de se retrouver seule dans un supermarché anglais où les deux ingrédients qui règnent en maître et qui imposent leur goût à tous les autres sont le froid et le plastique. Merci de partager avec moi tes sensibilités pianistiques et olfactives, merci de toujours chérir notre amitié, malgré la distance, depuis maintenant plus de 10 ans.

Clément, Chloé, notre rencontre a été le fruit d'un hasard (une pandémie, un confinement, un billet d'avion pris pour le lendemain, un attrait commun pour la Tanzanie, un vol de portefeuille,

etc) et elle aurait dû durer quelques jours, puis quelques petites semaines et elle semble désormais partie pour se prolonger à jamais. Ces mois passés à Zanzibar en votre compagnie et avec celle de Léonard et Léa, puis de Paul et d'Erlend, resteront gravés dans ma mémoire au temple de mes souvenirs les plus chers. Ce livret est malheureusement trop court pour les évoquer tous et je ne mentionnerai que ces nombreuses heures, qui s'étiraient jusqu'à se fondre dans la chaleur et l'humidité ambiante, passées à la table de chez Shah, à discuter, à apprendre à se connaître, à travailler, à étudier le swahili avec Mwalimu, à déguster les délicieux chapatis rapportés du Lookman ou des crevettes à la sauce magique, à somnoler tranquillement, sous le ventilateur et dans la moiteur de l'après-midi. Pour rendre hommage à ce séjour, il faudrait aussi parler des verres de jus devant le coucher du soleil, du sable blanc de Padje, des soirées glauques du Tatoo, des journées de pêche, des ruelles sinueuses de StoneTown, de la beauté du swahili, des morsures du soleil, des maux d'estomac, des pluies diluviennes, etc. Clément, merci d'arriver à créer aussi facilement un espace de convivialité où l'on se sent à l'aise en toute situation, que ce soit pour rire, pour faire la fête ou pour discuter, et qui a grandement contribué à ce que l'on puisse se rapprocher aussi rapidement. Merci pour ton engagement politique et écologique sans lequel nous n'aurions pas profité de ces 43 heures de voyages en train et en bus pour nous conduire en Roumanie et pour nous y en ramener. Chloé, merci pour ton entrain à toute épreuve, ton envie d'apprendre et ton ouverture d'esprit, ta capacité à plier le monde pour qu'il corresponde mieux à tes désirs et à ceux de tes proches, merci pour ta force de décision et de cohésion sans laquelle nous n'aurions pas fait la moitié de toutes nos activités.

Merci Alexandre pour ton amitié, ta gentillesse et ta tolérance, merci pour toutes ces petites anecdotes et imitations sorties du tréfonds de ton immense mémoire qui mettent toujours judicieusement en relation les tumultes du présent avec les richesses du passé. Merci Alex, Bertrand et Nathalie d'être d'aussi agréables voisins et compagnons de dîner ou de voyage, à Paris, dans le quartier le plus bobo de la capitale, à la mer, dans ce charmant petit village de la côte Atlantique dont nous tairons le nom pour ne pas y attirer les foules, ou à la montagne, au "chalet", où nous sommes réunis par la chaleur de la cheminée et par l'odeur de l'abondance qui fond doucement sur les appareils à raclette.

Si je suis la personne que je suis aujourd'hui, si j'ai quelques convictions, quelques qualités et un sourire à présenter à mon entourage et à la communauté humaine dans son ensemble, c'est en premier lieu grâce à ma famille.

Mes parents, mon frère, comment vous dire merci pour tout l'amour que vous m'avez donné ? Je suis si ému en écrivant ces lignes que je ne sais pas par où commencer, sinon par vous dire que sans vous auprès de moi, il y aurait un vide immense dans ma vie.

Maman, papa, je devrais dire d'abord que j'ai reçu tout ce dont j'avais besoin pour m'épanouir en tant qu'enfant, adolescent et adulte heureux de vivre, heureux de se sentir aimé et bien entouré. Que j'ai trouvé, sans avoir eu à le chercher, à la maison, un lieu où raconter mon dernier exploit sportif, ma dernière bonne note obtenue à l'école, mon dernier mal-être, mon dernier avis politique, mon dernier chagrin, mon dernier fantasme sur la technologie sauveuse du peuple, ma dernière lecture, mon dernier morceau de piano, ma dernière soirée, bref, où j'ai pu tout dire, sans crainte d'être jugé, face à des oreilles attentives et des yeux bienveillants. Ces centaines, ces milliers et ces millions d'heures de discussion, à apprendre, à me tromper, à écouter, à rire, à pleurer, à m'énerver, à se disputer, à me faire gronder, à négocier, à expliquer, constituent les fondations et le socle de mon esprit et ce sont d'elles que découlent mes convictions, mes idées, mes désirs, mes connaissances, mes valeurs, en somme, ma personne toute entière. Papa, Maman, je n'aurais pas pu souhaiter meilleure éducation et je vous suis infiniment reconnaissant pour tout ce que vous avez fait et tout ce que vous continuez à faire pour moi, pour Léonard, pour tout

ce que vous nous avez apporté d'amour et d'enseignements. Je souhaite que cela dure toujours.

Merci Mamie de penser autant à nous, de te soucier de savoir si nous avons assez de brioche pour le goûter et de nous avoir apporté de quoi nous en acheter, merci de te préoccuper de notre santé, de nos études (qui touchent enfin au but), merci d'être toujours aussi aimante et aussi heureuse de nous voir.

Merci Laurent, Corinne, Christian, Emma, Bixente pour tous ces joyeux déjeuners et dîners, pour tous ces moments partagés avec la mamie, dans la bonne humeur et la bienveillance.

Dear Shari, I am so very happy to have met you, thank you for giving me the warmest welcome I have ever received from anyone. Thank you for sharing with me a portion of the immense love you hold — a love that makes you shine, brings joy to all around you, and has taught me how to grow into a better human being.

Léa, merci de t'être installée aussi naturellement dans la famille, auprès de Léonard. La liste de nos souvenirs communs ne cesse de s'allonger et à chaque fois que tu étais là, en vacances, à la maison, à un dîner, à un verre, à Zanzibar, au moulin, j'ai l'image d'un moment heureux auquel ta présence a grandement contribué.

Léonard, tu es mon frère, mon meilleur ami, mon plus bel interlocuteur, ma fierté, mon compagnon de jeu, celui qui me comprendra et qui me supportera toujours quoi qu'il advienne, l'être qui m'est le plus cher.

Nous voici déjà arrivés à la conclusion de ces remerciements et de cette aventure pompeusement nommée doctorat. Avant de vous laisser vous délecter des quelque 223 pages que contient ce manuscrit, j'aimerais remercier une dernière fois les êtres qui me sont chers, qui sont, en somme, tous les êtres qui ont partagé avec moi leur goût de la vie, toutes celles et ceux qui m'ont souri ces trois dernières années et dans le sourire desquels j'ai trouvé des marques affectueuses de joie, d'intelligence, et de bienveillance. Merci.

When Random Forests Meet Neural Networks
A Finite-Sample Analysis

Abstract

In essence, this Ph.D. strives to fathom the crossroads of traditional tree-based methods and modern neural architectures, exploring potential synergies, benefits, and theoretical underpinnings from a statistical perspective. The theoretical setting is generally that of non-parametric regression with finite samples. Two pieces of work (Chapters 2 and 3) involve the Deep Forest algorithm (DF, Zhou et al. 2017), which stacks Random Forests (RF) in a Neural Network (NN) fashion. We theoretically analyse the benefit of stacking trees in a simplified DF architecture (Chapter 2), while numerically we use pre-trained DF, among other tree-based methods, to initialize NN training and thereby boost their performances (Chapter 3). In a further development, we examine the behaviour of RF algorithms in the interpolation regime, thus extending the study of interpolating estimators (such as neural networks and kernel methods) to random forests. Rates of convergence are established for interpolating median RF, and the influence of interpolation on the prediction performances is also measured through the volume of the interpolation zone, characterized for interpolating Breiman forests (Chapter 4).
Finally, we present an ongoing implementation work consisting in training neural networks with different objectives inspired from the PAC-Bayes framework in order to reach faster optimisation and better generalisation performances.

Keywords: random forests, statistical learning, neural networks, interpolation, finite sample

Laboratoire de Probabilités, Statistique et Modélisation
Sorbonne Université – Campus Pierre et Marie Curie – 4 place Jussieu – 75005 Paris – France

Contents

Chapter 1

Introduction

Many objects and concepts with complex relationships appear in this work. We present here the main ones who are, by order of importance, i) Random Forests (RF), ii) Neural Networks (NN) iii) interpolation and regularization.

Preliminaries - A Bit of Learning

This work is dedicated to the study of estimators in the context of **supervised learning**. This framework corresponds to the most usual tasks of machine learning: given some labelled data (such as classified images), our aim is to develop an algorithm to estimate the relationship between the input features and the corresponding label in order to make predictions on unlabeled data based on inputs only.

Supervised Learning In the context of supervised learning, we observe i.i.d. instances $(X_i, Y_i)_{1 \leqslant i \leqslant n}$ of a random (i.e. complex) phenomenon (X, Y) of unknown distribution with $X \in \mathcal{X} \subset \mathbb{R}^d$ and $Y \in \mathcal{Y} \subset \mathbb{R}$. The variable Y is often referred to as the *output* or *answer*, which is considered to be real-valued in this work (either continuous or discrete). Our goal is to predict Y given the *input* X (also called the *input variables*). In statistical terms, we want to *estimate* the conditional quantities $\mathbb{E}[Y|X = x]$ in regression, or $\mathbb{P}(Y|X = x)$ in classification, for all $x \in \mathcal{X}$. In order to achieve this goal, we introduce estimators, generically written $f_n : \mathcal{X} \to \mathcal{Y}$, whose construction depends on the data $(X_i, Y_i)_{1 \leqslant i \leqslant n}$.

Non-parametric Regression Consider the simple framework of nonparametric regression estimation, in which an input random vector $X \in \mathbb{R}^d$ is observed, and the goal is to predict the random response $Y \in \mathbb{R}$. Typically, we have

$$Y = f^\star(X) + \varepsilon \tag{1.1}$$

where $f^\star : \mathbb{R}^d \to \mathbb{R}$ and ε is a random noise independent of X verifying $\mathbb{E}[\varepsilon] = 0$ and $\mathbb{E}\left[\varepsilon^2\right] = \sigma^2 > 0$. To determine how good an estimator f_n is, we measure the distance between the prediction of the estimator and the original label with the L^2 cost averaged on the data, and we call this distance the *risk* of f_n:

$$\text{Risk}(f_n) := \mathbb{E}\left[(Y - f_n(X))^2\right].$$

However, there is an inextricable randomness in Y that we cannot hope to predict. Therefore, we write:

$$\begin{aligned}\text{Risk}(f_n) &= \mathbb{E}\left[(f^\star(X) + \varepsilon - f_n(X))^2\right] \\ &= \underbrace{\mathbb{E}\left[(f^\star(X) - f_n(X))^2\right]}_{\mathcal{R}(f_n)} + \mathbb{E}\left[\varepsilon^2\right]\end{aligned}$$

where $\mathcal{R}(f_n)$ is called the *excess risk* of f_n. Our target is thus the best estimator possible without taking into account the additional noise ε. It is called the *Bayes estimator* and writes $f^\star(x) = \mathbb{E}[Y|X = x]$. We also refer to it as the *regression function*.

Our goal is thus to make the best possible use of the i.i.d. data points $(X_i, Y_i)_{1 \leqslant i \leqslant n}$ distributed as (X, Y) in order to construct an estimator $f_n : \mathbb{R}^d \to \mathbb{R}$ of the function $f^\star$.

Estimator Properties A good indicator of the quality of an estimator is the excess risk defined above. Such an excess risk can be decomposed into two parts:

$$\mathbb{E}\left[(f_n(X) - f^\star(X))^2\right] = \underbrace{\mathbb{E}\left[(\mathbb{E}[f_n(X)|X] - f^\star(X))^2\right]}_{\text{bias}} + \underbrace{\mathbb{E}\left[(f_n(X) - \mathbb{E}[f_n(X)|X])^2\right]}_{\text{variance}}. \tag{1.2}$$

1. The *bias* quantifies how close the estimator can be from $f^\star$ in expectation. In general, the more complex the estimator, the lower the bias.
2. The *variance*, which measures the sensitivity of the estimator to the noise of the training data. A priori, the more complex the estimator, the higher the variance (although modern algorithms challenge this paradigm, as we will see in Section 1.3).

We should expect from a well-behaved estimator that, given an infinite amount of data, both its variance and bias converge to 0. This is formalized in the following definition of *consistency*:

Definition 1.0.1. *An estimator f_n is said to be* ***consistent*** [1] *if* $\lim_{n\to\infty} \mathcal{R}(f_n) = 0$.

Consistency is one of the most fundamental properties for an estimator. One of the goals of the statistician is therefore to prove consistency and compute convergence rates under hypotheses that are as general as possible. This property as well as the convergence rates often allow us to better understand the behavior of the algorithm at hand and eventually to refine it. A traditional proof technique consists in upper-bounding the bias and the variance separately in order to prove consistency. In the following, we will very often discuss specific estimators from the bias-variance perspective.

1.1 Random Forests

Although simple in appearance, the Random Forests (RF) algorithm achieves great performances and raises many theoretical questions that resonate with the fundamentals of statistics. We first draw an overview of these algorithms before diving into their theoretical challenges.

[1] Note that several types of consistency exist. We are mostly interested in $\mathbb{L}^2$ consistency, which we simply call consistency in order to lighten the notations.

1.1.1 Presentation

Introduced in the early 2000s, Random Forests (Breiman 2001a) (RF) are still among the most popular machine learning algorithms. They are particularly efficient to deal with tabular data in a supervised setting (both regression and classification), and are applied to a wide range of applications (Díaz-Uriarte et al. 2006; Prasad et al. 2006; Chen et al. 2012; Belgiu et al. 2016). The simplicity of the RF design and hyperparameter tuning, as well as its powerful predictive performances, are key ingredients to its success. The algorithm relies on a "divide and conquer" principle: sample fractions of the data, grow a randomized tree predictor on each subsample, then aggregate the predictors together. Many variations of RF algorithms were built over the years; unless specified otherwise, the generic term RF refers to Breiman's original algorithm, detailed below. As our theoretical studies focus on the regression setting, we present the algorithm in this case. We start by presenting the decision tree construction, at the core of the RF algorithm.

Decision Tree Breiman RF use specific decision trees (DT) called Classification And Regression Trees (CART, Breiman et al. 1984). CART recursively partitions the input space $\mathcal{X}$ with hyperplanes, parallel to one of the axes. Each node of the tree thus takes the form of a hyper-rectangular node included in $\mathcal{X}$, the *root cell*. Starting from the root, at each step of the construction, a cell is split into two parts. It goes on until a stopping criterion is reached (for instance, a pre-specified number of splits). More precisely, given a cell $A = \prod_{\ell=1}^{d}[a_\ell, b_\ell]$, in order to compute the next split, we compute the optimal couple feature/split-value (j_A, z_A) satisfying:

$$(j_A, z_A) = \underset{(j,z)\in\mathcal{E}_A}{\arg\max}\, L_A(j, z), \tag{1.3}$$

where $\mathcal{E}_A = \{(j, z) : j \in \{1, \dots, d\}, z \in (a_j, b_j)\}$ is the set of eligible split in the cell A and where L_A measures a decrease in impurity (the variance in regression, the Gini criterion or the entropy in classification). In regression, denoting $X^{(j)}$ the $j-$th coordinate of X, L writes for all A, j, z:

$$\begin{aligned} L_A(j, z) = &\frac{1}{N_n(A)} \sum_{i=1}^{n} (Y_i - \bar{Y}_A)^2 \mathbb{1}_{X_i \in A} \\ &- \frac{1}{N_n(A)} \sum_{i=1}^{n} \left((Y_i - \bar{Y}_{A_L} \mathbb{1}_{X_i \in A_L} - \bar{Y}_{A_R} \mathbb{1}_{X_i \in A_R} \right)^2 \mathbb{1}_{X_i \in A}, \end{aligned}$$

where $A_L = \{X \in A : X^{(j)} < z\}$, $A_R = \{X \in A : X^{(j)} \geqslant z\}$, and $\bar{Y}_A$ (resp., $\bar{Y}_{A_L}$, $\bar{Y}_{A_R}$) is the average of the Y_i's belonging to A (resp., A_L , A_R), with the convention $0/0 = 0$. Finally, for a given point x, the prediction of a tree at the point x is made by averaging the values Y_i for which the X_i fall into the leaf of x (see eq. 1.4). We provide a schematic view of a DT in Figure 1.1.

Mathematical Definition of RF A random forest is a predictor consisting of a collection of M randomized regression trees. Randomness is introduced in the tree construction to ensure that the trees are diverse enough for the averaging to be efficient. For the j-th tree in the family, the predicted value at the query point X is denoted by $f_n(X; \Theta_j, \mathcal{D}_n)$, where $\Theta_1, \dots, \Theta_M$ are independent random variables, distributed the same as a generic random variable Θ and independent of $\mathcal{D}_n$. In practice, the variable Θ is used to resample the training set prior to the growing of individual trees and to select the successive directions for splitting—more precise definitions

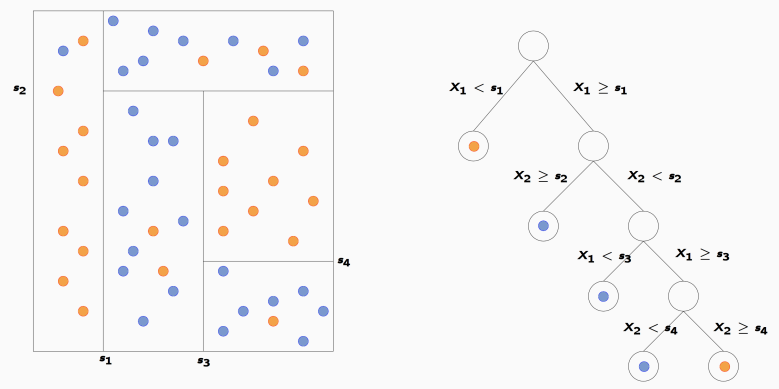


Figure 1.1: Schematic view of a decision tree.

will be given later. In mathematical terms, the j-th tree estimate takes the form

$$f_n(X;\Theta_j,\mathcal{D}_n) = \sum_{i\in\mathcal{D}_n^\star(\Theta_j)} \frac{\mathbb{1}_{X_i\in A_n(X;\Theta_j,\mathcal{D}_n)}Y_i}{N_n(X;\Theta_j,\mathcal{D}_n)}, \tag{1.4}$$

where $\mathcal{D}_n^\star(\Theta_j)$ is the set of data points selected prior to the tree construction, $A_n(X;\Theta_j,\mathcal{D}_n)$ is the cell containing X, and $N_n(X;\Theta_j,\mathcal{D}_n)$ is the number of (preselected) points that fall into $A_n(X;\Theta_j,\mathcal{D}_n)$. At this stage, we note that the trees are combined to form the (finite) forest estimate

$$f_{M,n}(X;\Theta_1,\ldots,\Theta_M,\mathcal{D}_n) = \frac{1}{M}\sum_{j=1}^{M} f_n(X;\Theta_j,\mathcal{D}_n). \tag{1.5}$$

An illustration of a RF is presented on Figure 1.2. To lighten the notations, we drop the dependencies on $\mathcal{D}_n$.

By letting M tend to infinity, according to the law of large numbers, we obtain the **infinite RF**:

$$f_{\infty,n}(x) = \mathbb{E}_\Theta[f_n(x,\Theta)].$$

Such object is easier to study theoretically than the finite forest, while being close to finite RF when M is large (the distance between finite and infinite RF typically decreases as $1/M$, see (Scornet 2016a) for more details). Here, $\mathbb{E}_\Theta$ denotes the expectation w.r.t. Θ, conditional on $\mathcal{D}_n$.

Tree randomness As stated above, in order to diversify the trees of a random forest, we introduce two complementary processes of randomization. The **bootstrap** method involves randomly sampling n observations with replacement from the dataset for each tree, essentially subsampling rows if the data is seen as a $\mathbb{R}^{n\times d}$ matrix. Conversely, **feature subsampling** selects a random subset of features at each node for optimal splitting, akin to column subsampling in matrix representation. Note that this process is repeated for each node of each tree, unlike bootstrap, which is done only once for each tree, at the beginning of their construction. We draw a small review of the theoretical analysis of these processes in the next section.

Parameters Several libraries in many programming languages implement the RF algorithm (see, for instance, `Scikit-learn` in Python (Pedregosa et al. 2011) or the `randomForest` package

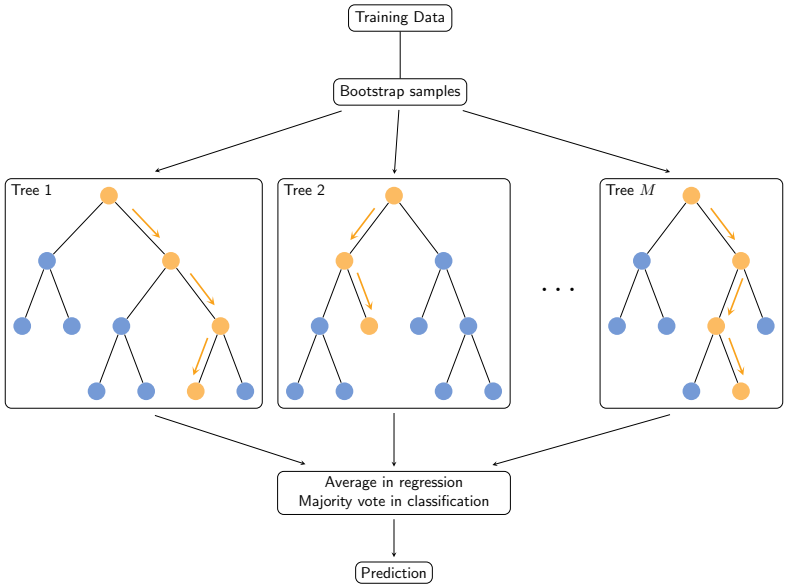


Figure 1.2: Schematic view of a RF.

in `R`). We will mention the parameters as they are named in the famous library `Scikit-learn` (Pedregosa et al. 2011).

- **Number of Trees.** The first thing to set is the number of trees used in the forest. It can be made arbitrarily large (even infinite in our theoretical studies) and is usually limited by computing performances. The influence of the number of trees is well understood, it has been studied in, e.g., Genuer et al. 2010; Díaz-Uriarte et al. 2006; Scornet 2016a. Roughly, from a prediction performance perspective, the more trees the better.

- **Stopping Criterion.** The second parameter to define is the **stopping criterion** that is applied to all the trees of the RF. It is possible to set this criterion in several ways, for instance by fixing a limiting **depth** for each tree, a minimum number of points in each leaf, a minimum impurity/variance value in each leaf, etc. As we discuss later, this parameter has a lot of influence on the behavior of the RF: it determines the convergence rates of upper bounds on the excess risk, for instance. As depth increases, the number of leaves increases as well, and the number of points contained in each leaf decreases: the bias decreases but the variance increases for a tree (the averaging effect is less efficient in each leaf).

- **Max-Features.** The parameter `max-features` determines, in each node, how many directions are randomly subsampled from $\{1, ..., d\}$ for the computation of the best split. We denote it as an integer $m \in \{1, ..., d\}$. The lower m, the more random the construction of a tree and the more diverse the RF.

- **Bootstrap parameters.** Finally, a few parameters also calibrate the bootstrap procedure. In particular, `max-samples` sets the number of points sampled to train each tree. Note that we can also easily deactivate bootstrap in order to train each tree on the whole dataset, as we will do in several theoretical studies of RF.

Intuition Statistical wisdom advocates the optimization of a bias-variance trade-off as a fundamental principle to design a good estimator. Regarding each RF tree, this could *a priori* mean enforcing each leaf to contain a certain amount of data points, or equivalently, limiting the depth of each tree. Limiting the depth increases the bias, but as the prediction of a tree involves averaging the Y_i's in each leaf, it helps to reduce the variance. However, in practice, RF are often built with fully-grown trees (it is the default setting in the `scikit-learn` library) and exhibit good performances in high-depth regimes. The latter strategy can be viewed as starting from a strong learner (a fully grown tree) with a low bias as well as high noise sensitivity, and then introducing regularization processes that reduce the variance while preserving a low bias.

The following section studies several aspects of the bias-variance tradeoff in RF, as well as the regularization processes that allow RF (even those composed of fully-grown trees) to reduce their sensitivity to data noise.

1.1.2 Theoretical Insights and their Empirical Consequences

Several decades have elapsed since the introduction of Classification and Regression Trees (CART) and Random Forests (RF), yet a comprehensive mathematical understanding of their behavior remains elusive. Random Forests, in particular, pose a range of theoretical challenges, and in this section, the focus is on the aspects of consistency. Establishing necessary and sufficient conditions for the consistency of these algorithms is an open problem. In the subsequent paragraphs, we outline a few steps toward addressing this issue.

CART

First, with regard to proving the consistency of a single CART, the main difficulty lies in the strong dependence of the CART construction on the data distribution. As CART's construction takes into account the positions X_i's and the values Y_i's, analyzing its behavior requires several assumptions on both the distribution of $Y|X$ and that of X. We examine these desirable hypotheses following a standard bias-variance decomposition obtained by applying Jensen inequality:

$$\mathbb{E}\left[(f_n(X,\Theta)-f^\star(X))^2\right] \leqslant 2\,\underbrace{\mathbb{E}\left[\left(\sum_{i=1}^n W_i(f^\star(X_i)-f^\star(X))\right)^2\right]}_{\text{bias}} + 2\,\underbrace{\mathbb{E}\left[\left(\sum_{i=1}^n W_i(Y_i-f^\star(X))\right)^2\right]}_{\text{variance}} \tag{1.6}$$

where $W_i = \frac{\mathbb{1}_{X\in A_n(X_i,\Theta)}Y_i}{N_n(X_i,\Theta)}$.

Bias In order to obtain convergence of the bias, one strategy is to ensure that the variation of $f^\star$ within each leaf can be made arbitrary small. To that end, one could assume that $f^\star$ is continuous (or uniformly continuous (Scornet et al. 2015)), has bounded derivatives (Klusowski 2021a) or bounded variations (Klusowski 2021b). However, this is not sufficient to ensure that the bias vanishes. Thus, a second assumption is needed to guarantee that the leaves of CART become small enough on directions where the variation of $f^\star$ is not negligible. Consequently, on these directions, CART should make enough splits to reduce the size of the leaves. To this aim, it is necessary (but not sufficient) that the depth k_n grows to infinity. The size of the leaf is sufficiently reduced when we add a final hypothesis: in the case of *additive models*, thanks to feature disentanglement, CART performs enough splits on each important direction so that it achieves low variations of $f^\star$ in each leaf when the depth verifies $k_n = o(\log_2(n/\log(nd)))$

(Klusowski 2021b) or is such that the number of leaves t_n satisfies $t_n(\log n)^9 = o(n)$ (Scornet et al. 2015). These results were later extended in Elie-Dit-Cosaque et al. 2022 who identified a broader class of functions, which we call the "CART-friendly" class in the sequel, for which this condition is verified, and CART consistency is obtained (see Definition 4.1 in Elie-Dit-Cosaque et al. 2022). Finally, we also mention the first consistency proof of CART, where it is directly supposed that the leaves shrink as n grows (Breiman et al. 1984). Other consistency results on more general data-dependent partitioning algorithms (including CART) also require assumptions from which we can deduce cell shrinking as n grows (Lugosi et al. 1996).

Variance Regarding the variance, we first recall that the predictions of CART in each leaf are made by averaging all the Y_i's contained in the leaf. In a leaf, the variance decreases toward 0 if and only if the number of data points is sufficient (i.e. grows to infinity) in order to average the noise out. This situation is opposite to the one we faced when controlling the bias, in which the leaves needed to be as small as possible, thus leading to a low number of observations per leaf. Therefore, it is necessary to limit the depth and to make assumptions that guarantee a minimum number of points in each leaf. The hypothesis considered in Wager et al. 2015 (Definition 1) perfectly answers our needs: it assumes that each child node contains a minimum portion of the data points belonging to the parent node and that the terminal nodes all contain a minimum number of points. Wager et al. 2015 also require the use of *weakly-dependent* features. With these hypotheses, they obtain a control on the estimation error of any tree of order $O(\sqrt{\log(d)\log(n)/n_{\text{leaf}}})$ with high probability, with n_{leaf} the minimum number of samples per leaf. Convergence of the estimation error was also obtained without assumptions on the behavior of the tree: in the case of additive models, (Scornet et al. 2015; Klusowski 2021b) for instance, with an extension to the "CART-friendly" class introduced above (Elie-Dit-Cosaque et al. 2022). Indeed, for a limited depth, the estimation error of CART can be controlled via the hypotheses introduced in Nobel 1996, namely a subexponential growth of their shattering number, a restricted number of cells.

Consistency The above discussion enlightens the role of the depth parameter k_n as a lever to control the bias/variance trade-off. The higher k_n, the lower the bias, the greater the variance. In order to obtain the consistency of CART, one should calibrate k_n such that (i) it tends to infinity to decrease the bias towards 0 and (ii) the number of points in each leaf tends to infinity so that the variance is reduced to 0. This is typically achieved when k_n is of the order $\alpha \log n$, with $\alpha \in (0, 1)$ (the tree roughly contains n^α leaves, each leaf containing $n^{1-\alpha}$ data points). Following this general comment, Scornet et al. 2015 proves the first consistency result in the case of additive models; it was later extended to a more general class of functions in Elie-Dit-Cosaque et al. 2022. Recently, Klusowski 2021b proved consistency of CART and obtained the first (and only, to the best of our knowledge) consistency rates. Note that these rates also hold in a high-dimensional setting. More precisely, in the case of the additive models, Klusowski 2021b proves that the excess risk of a CART f_n verifies

$$\mathcal{R}(f_n) \leqslant \frac{||f^\star||_{\text{TV}}}{k_n + 3} + C\frac{2^{k_n}\log(nd)}{n}$$

where C is a constant and $||\cdot||_{\text{TV}}$ indicates the total variation norm. This bound is optimal when $k_n = \frac{1}{2}\log_2(n)$. This is the first result providing consistency rates for CART, and it remains to know whether this rate is improvable or not. Indeed, it is of the order $O((\log n)^{-1})$ which is very slow: it is possible to obtain speed in $O(\sqrt{(\log(n)d)/n})$ with similar assumptions (Tan et al. 2019). This matter is discussed in Section 4.3 of Klusowski 2021b. The interested reader can find a summary of the work on CART consistency in Klusowski 2021b Section 1.1.

Random Forests - Quantifying the Randomization Effects

The idea of averaging several randomized estimators, referred to as *bagging*, is a few years older than the conception of RF (Breiman 1996). The benefits of bagging over a single learner were soon identified (Bühlmann et al. 2002) and it was proved in Biau et al. 2008 that it is possible to build a consistent estimator from the aggregation of non-consistent 1-NN which individually have a high variance.

Regarding RF, the benefits of averaging several randomized CART are clearly observed in practice, but hard to quantify mathematically. Several consistency proofs of RF rely on conditions that already guarantee the consistency of a single DT: they remain in a regime where the number of samples per leave tend to infinity (Scornet et al. 2015; Elie-Dit-Cosaque et al. 2022). Proving RF consistency without relying on that of CART requires levering the RF randomization processes, bootstrap and feature subsampling, which are quite difficult to analyze even individually. We discuss their effects on variance reduction in the two following paragraphs.

Feature Subsampling In order to better understand the effect of feature subsampling, several papers study the consistency of several kinds of RF when bootstrap is turned off. As Breiman RF are difficult to study theoretically, we begin with the analysis of a simpler model, the **Centered Random Forests** (CRF), which are built independently of the data (in each node, the cut is performed at the middle along a random direction). When a RF is built depending on the X_i's only we call it *semi-adaptive* and when it is built independently of the data we call it *non-adaptive* (it is sometimes called *purely random forest* in the literature).

To the best of our knowledge, Proposition 2 of Biau 2012b is one of the first results exhibiting a vanishing estimation error, in a high depth regime, with bootstrap off, in the case of CRF (see also Breiman 2004). It was recently improved in Klusowski 2021a where nearly-optimal convergence rates for non-adaptive RF with bootstrap off are obtained. The upper bound on the risk of the CRF $f_{\infty,n}^{\mathrm{CRF}}$ reads (in a slightly simplified manner):

$$\mathcal{R}(f_{\infty,n}^{\mathrm{CRF}}) \lesssim 2^{2k_n \log_2(1-\frac{1}{2^d})} + \frac{2^{k_n}}{n}(\log_2 k_n)^{-\frac{d-1}{2}} + e^{-n2^{-k_n}/2} \tag{1.7}$$

This bound is minimal when $k_n = (n(\log_2^{d-1} n)^{1/2})^{1-\alpha}$ and $\alpha = 2\log_2(1-p/2)/(2\log_2(1-p/2)-1)$, according to a classical bias-variance trade-off. Remarkably, the *estimation error* still converges to 0 in the regime $k_n = \log_2 n$ although it deteriorates the convergence rate. The variance reduction thus directly benefits from the feature subsampling effect, but it is still not as efficient as limiting tree depth. More precisely, compared to a single centered tree, the improvement on the estimation error is in $O((\log_2 k_n)^{-\frac{d-1}{2}})$. As shown in Lin et al. 2006 (and in Theorem 5 of Klusowski 2021b), it is only improvable up to $O((\log_2 k_n)^{-(d-1)})$ for any RF whose construction does not depend on the Y_i's, in particular CRF. Indeed, the prediction of a RF at a given point only depends on a few of its nearest neighbors (potentially chosen adaptively). Consequently, the averaging effect is limited by the number of such neighbors, which scales as $(\log n)^{d-1}$ if the density of X is bounded away from 0 in $[0,1]^d$.

Remark 1.1.1. *Although lower bounded in the case of non-adaptive RF, the reduction of variance due to feature subsampling can reach statistical optimality when combined with a wise split scheme, as used in Mondrian RF (Mourtada et al. 2020). In a Mondrian tree (Lakshminarayanan et al. 2014), a cell is split at a random time which depends on its perimeter (the greater, the bigger the split probability) and the direction to split over is randomly chosen depending on the sizes of sides of the cell (the greater the length of the side, the higher the probability). Mourtada et al. show that Mondrian RF achieve minimax rates on*

the class of s-Hölder functions for $s \in (0, 2]$. Moreover, when $s \in (1, 2]$, these rates cannot be reached by a single tree, which emphasizes the role played by the feature subsampling process.

Regarding the feature subsampling process, one parameter plays a key role: the `max-features` parameter m, which indicates in each node how many directions are sampled as split candidates for the best-split search. In the case of CRF, the choice of the direction is completely random, which corresponds to $m = 1$. Setting $m > 1$ is only pertinent when dealing with adaptive random trees, for which the splitting direction can be chosen in an adaptive manner, for which the theoretical analysis is more complicated. An empirical analysis of the performance of a RF w.r.t. the Signal-to-Noise Ratio (SNR) of the data is provided in Mentch et al. 2019. Their findings are in line with the mathematical intuition that the lower the SNR, the more regularization is needed, the lower m should be. They also provide some insights on how feature subsampling reduces the variance of an aggregation of randomized Ordinary-Least-Square (OLS) linear estimators: it causes a regularization process similar to training an OLS with $\mathbb{L}^2$-penalty. In details, if f_B is the averaging of B OLS estimators, each one being computed on a random subset of features m among d features, then,

$$f_B \xrightarrow[B\to\infty]{} \frac{m}{d} f_{\text{OLS}}$$

where f_{OLS} is the standard OLS estimator computed on all d features. We clearly see here the influence of the `max-features` parameter, and it would be interesting to quantify a similar phenomenon in the case of RF.

Another peculiar result from Kobak et al. 2020 suggests that the regularization effect of feature subsampling is enhanced by simply artificially adding noisy features to the input data: it shows that adding random noise features to the data X, computing an OLS on the augmented data and selecting only d features of the estimator accounts for applying a ridge penalty to a classical OLS estimator when the number of noisy features tends to infinity. Mentch et al. 2022 empirically shows that augmenting the data with noisy features contributes to regularizing RF as well.

Bootstrap To the best of our knowledge, the benefits of any data subsampling process (including bootstrap) have not been quantified yet in the case of Breiman RF. Nevertheless, the effects of data subsampling *without replacement* are pretty well understood for Quantile RF or Median RF (which cut at a given quantile of the sample of each node along a randomly chosen direction) independently of the Y_i's. Scornet 2016a provides a first result on quantile RF consistency, where the variance reduction only stems from the data subsampling process. In particular, RF consistency is obtained even when trees are fully grown and individually inconsistent (although convergence rates are not obtained, as the proof relies on Stone's theorem). Consistency rates were later obtained in Duroux et al. 2018, where it is proved that controlling the tree depth or the data subsampling results in the same variance reduction. Indeed, reducing the subsampling size mechanically limits the depth of each DT, which does not grow more leaves than the number of samples. In detail, the upper bound on the excess risk writes:

$$\mathcal{R}(f_{n,\infty}) \lesssim \frac{2^{k_n}}{n} + d\left(1 - \frac{3}{4d}\right)^{k_n}$$

It can be minimized by either optimizing the depth k_n and neglecting the subsampling effect ($a_n = n$), or by fixing $k = \log_2 a_n$ (fully-grown trees) and calibrating the subsampling rate $a_n = n^{\frac{\log(1-3d/4)}{\log 2 - \log(1-3d/4)}}$. This rate was improved for Median RF in Klusowski 2021a, and proved to be minimax only when $d = 1$.

Consistency of Breiman RF Overall, to the best of our knowledge, for a fixed dimension d, the only consistency results on Breiman RF (without rates) can be found in Scornet et al. 2015; Elie-Dit-Cosaque et al. 2022, mentioned above. Very recently, (Chi et al. 2022) took another step forward by proving consistency of the original Breiman RF in a high-dimensional setting and computing rates. The only non-standard assumption, with regard to that of non-parametric regression, is a Sufficient Impurity Decrease (SID) condition which guarantees that each split lessens the impurity of the current node sufficiently. Denoting $\alpha > 1$ the SID parameter and $b \in (0, 1)$ the proportion of bootstrap samples for each tree, their bound (Theorem 1) roughly writes:

$$\mathcal{R}(f_{\infty,n}) \lesssim \alpha(\lceil bn \rceil)^{-\eta} + (1 - \frac{m}{d\alpha})^{k_n} + (\lceil bn \rceil)^{-\delta+c} \tag{1.8}$$

with $k_n \leqslant c \log_2(bn + 1)$, $c \leqslant 1/4$ $\eta < 1/8$, $2\eta < \delta < 1/4$ and C a constant. The second term on the right-side of Equation (1.8) corresponds to the bias term and its behavior is in line with the common observation that the higher k_n, the lower the bias and the higher m, the more the impurity of each tree decreases at each step of the construction, the better the bias. The first term is part of both the variance and bias and is mostly non-informative: dependencies on m and k_n are hidden and as the computations are made per tree for a fixed number of samples, they do not take advantage of the bootstrap effect on the variance reduction. The last term is, according to the authors, a technical residual that corresponds to the impossibility to control the discrepancy between the estimate $\mathbb{E}\left[\frac{1}{N_A}\sum_{X_i \in A} Y_i\right]$ and $\mathbb{E}[f^\star(X)|X \in A]$ when the cell A is too small; it also limits the depth of the RF by enforcing $c \leqslant \delta$.

Conclusion

In a nutshell, we summarize the main steps taken toward a theory of RF consistency and point out a few directions to conduct further research:

- **Bootstrap.** The effect of data subsampling without replacement is well understood in the case of semi-adaptive RF (see Scornet 2016a, Duroux et al. 2018). From a theoretical point of view, its regularizing effect is high, as reducing the sample size accounts for limiting tree depth. However, this is not exactly the bootstrap process, and it has yet to be quantified in the case of Breiman RF.

- **Feature subsampling.** To the best of our knowledge, the only quantification of variance reduction due to this effect is shown in the upper bound of the CRF estimation error in Proposition 2 of Biau 2012b and Theorem 2 of Klusowski 2021a. Our recent work Arnould et al. 2023 provides an analysis in the case of Median RF (see Chapter 4). The benefits of feature subsampling on the variance reduction of non- and semi-adaptive RF are limited by the number of nearest neighbors at each point (Lin et al. 2006). Full adaptivity (CART) allows the algorithm to build a *weighted* average of the values of the nearest neighbors. We think that quantifying the benefits of a weighted average over a simple average on the estimation error for simple cases could shed some light on the feature subsampling effect.

- **Overall understanding of RF.** In general, the *bias* part of the excess risk is pretty well understood in the case of decision trees, as it is directly linked to the size of a leaf. Then, the extension to RF via averaging is pretty straightforward. One exception is the geometry of CART, which remains hard to deal with, and extending the work of Scornet et al. 2015, Elie-Dit-Cosaque et al. 2022 or Chi et al. 2022 could allow us to better understand what conditions a function should satisfy in order to guarantee that the leaves of a CART shrink sufficiently.

In Elie-Dit-Cosaque et al. 2022, the authors offer a characterization of a particular set of functions termed as "CART-friendly". For a function f to be classified as "CART-friendly", it must satisfy the condition that if the expected value $z \in \mathbb{R}^{d-1} \mapsto \mathbb{E}[f_j(z, X^{(j)})\mathbb{1}_{X^{(j)} \in A_j}]$ remains constant along the j-th side A_j of any given rectangle $A = \prod A_j$, then f itself must also remain constant throughout the entirety of that rectangle. Elie et al. have demonstrated that certain functions, including additive and product functions, inherently belong to this "CART-friendly" class. By achieving an explicit understanding of the "CART-friendly" class, we would be able to gain a deeper comprehension of the conditions under which both CART and Random Forests (RF) are anticipated to be consistent. Regarding the *variance* of Breiman RF, the picture becomes a bit blurry. As detailed above, the effects of both randomization processes have yet to be well analyzed individually before being put together and/or being compared.

Finally, we mention that the theoretical study of RF is not limited to the consistency aspect. Among other angles of study we find variable importance, asymptotic normality or the connection with kernel methods. The interested reader can find a starting point to these subjects in Biau et al. 2016 or in Criminisi et al. 2012.

1.2 Neural Networks

In this section, we succinctly overview neural network algorithms, beginning with an examination of fundamental principles and design considerations, followed by an introduction to the central mathematical challenges posed by neural networks.

1.2.1 Presentation

Neural Networks (NN) are among the most famous machine learning algorithms. Their recent successes at solving vision and natural language processing tasks made them famous far beyond the machine learning community. A key element of their success is that they combine simple conception and training principles (composing non-linear parametric transformations and learning their parameters via Stochastic Gradient Descent (SGD)) with very flexible architectures.

Basic Principles Although more generic presentations of the algorithm can be found (Shrestha et al. 2019), we restrain ourselves to Multi-Layer Perceptrons (MLP, Rumelhart et al. 1986) in a supervised setting.

Architecture A NN is composed of several ($L \in \mathbb{N}$) layers, each layer ℓ representing a composition of a non-linear function $\phi : \mathbb{R} \to \mathbb{R}$ applied element-wise and an affine function $f_\ell : x \in \mathbb{R}^{d_{\ell-1}} \to W_\ell x + b_\ell \in \mathbb{R}^{d_\ell}$ where $W_\ell \in \mathbb{R}^{d_{\ell-1} \times d_\ell}$ and $b_\ell \in \mathbb{R}^{d_\ell}$. The widths of the layers correspond to the dimension of the *latent spaces* in which they inject the data $d_0, ..., d_L$. Obviously, $d_0 = d$ and d_L are fixed by the dimension of the input X and output Y. Overall, a neural network estimator f_n writes

$$f_n(x) = W_L(\phi \odot (W_{L-1}(...(\phi \odot (W_0 x + b_0))...) + b_{L-1}) + b_L, \tag{1.9}$$

see Figure 1.3 for a schematic view. We concatenate all the *parameters* of the network $\{(W_0, b_0), \dots, (W_L, b_L)\}$ into a vector $\theta \in \mathbb{R}^p$ where $p = \sum_{\ell=1}^{L-1}(d_\ell d_{\ell+1} + d_\ell) + d_0 d_1$ and denote $f_{n,\theta}$ the corresponding NN. We sometimes refer to the parameters as the *weights* and the *biases* of the NN. The design of a NN involves the choice of numerous *hyper-parameters* such

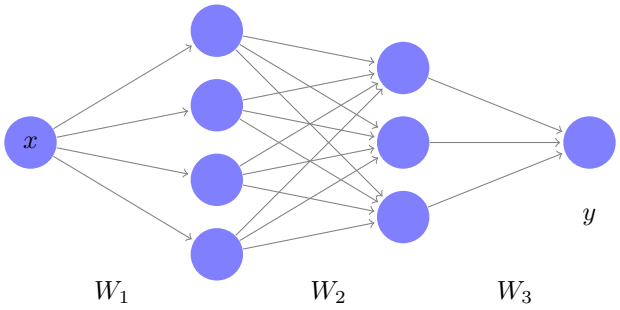


Figure 1.3: Schematic view of a 3-layer neural network.

as the layer type (e.g., convolutional, recurrent, fully connected, ..., see paragraph "data type" below), the number of neurons per layer, the number of layers (also referred to as "depth"), etc. Hyperparameter selection (or search) is a whole area of research that is beyond the scope of the current work.

Data Type One of the big advantages of NN is that their architecture can adapt to different types of data, in particular images and text. Their empirical success was illustrated first and foremost by *Convolutional Neural Networks* (CNN, LeCun et al. 1995) on the ImageNet dataset (Deng et al. 2009). Convolutional operations indeed fit very well to image processing, as they are invariant to translation of the input signal. Other kinds of networks such as *Recurrent Neural Networks* (RNN, Rumelhart et al. 1985) or *transformers* (Vaswani et al. 2017) were also specially designed to process sequential or temporal data. They leverage the sequential structure by using memory processes and/or refined inner-product-like operations. In contrast, there is no *natural* NN architecture that matches *tabular data*. By default, the MLP architecture is still largely used due to its generalist nature. Improving the performances of MLP on tabular data is challenging, and several levers can be considered: changes of hyperparameters, optimization schemes, or initialization methods. In this work, we focus mostly on the initialization aspect, detailed below.

Training Recall that ideally, we would like to find the neural network minimizing the theoretical risk. However, the data distribution is unknown (and so is the theoretical risk), and we only have access to a finite training sample, $\mathcal{D}_n = \{(X_1, Y_1), ..., (X_n, Y_n)\}$ of i.i.d. copies of a generic pair $(X, Y) \in \mathbb{R}^d \times \mathcal{Y}$, to do so. Therefore, we learn a NN predictor by minimizing the empirical risk (surrogate of the risk), that measures the performance of a NN f_n on the training set $\mathcal{D}_n$,i.e.,

$$r_n(f_n) := \frac{1}{n} \sum_{i=1}^{n} \ell(f(X_i), Y_i)$$

where ℓ is either the quadratic cost in a regression setting or the cross-entropy loss in a classification setting. In order to minimize the training loss of a NN with respect to the NN parameters, the most usual training procedure consists of using Stochastic Gradient Descents (SGD, Ruder 2016). This is an iterative procedure where, at each step t, a subset $\mathcal{S}_t$ (mini-batch) of the data is randomly selected, and the parameters are updated as follows:

$$\theta_{t+1} = \theta_t - \eta \frac{1}{|\mathcal{S}_t|} \sum_{(X_i, Y_i) \in \mathcal{S}_t} \nabla_\theta \ell(f_{n,\theta}(X_i), Y_i) \tag{1.10}$$

with $\eta \in \mathbb{R}$ the learning rate. There exists an abundant literature of studies and variations of this algorithm (see Ruder 2016; Bottou et al. 2018 for a review). The stochastic gradient algorithm requires an initial value θ_0 as an input. Due to the non-convexity of the problem to be solved, the initialization is a delicate issue.

Initialization Initialization is a crucial phase of the design of a NN as it will impact its optimization behavior and, ultimately, the quality of the minimum obtained at the end of training. Indeed, we are minimizing non-convex objectives based on algorithms conceived to deal with convex problems, so it is very delicate to ensure convergence independently of the starting point. As NN are very complex methods (often referred to as *black-box models*), it is hard to precisely measure the impact of initialization on both their optimization behavior and their generalization performances. The choice of the initialization scheme is thus often guided by empirical precepts and mathematical intuition. We distinguish two kinds of initialization methods: random and deterministic. A review on initialization of NN can be found in Section 4.2 of Sun 2020.

The success of NN is partially due to several widely used random initialization methods, e.g., Glorot et al. 2010; He et al. 2015. For each layer, the weights are generally drawn according to uniform or normal distributions with a variance decreasing w.r.t. the width of the layer. For instance, a normalized Glorot initialization of a layer ℓ (Glorot et al. 2010) corresponds to a uniform distribution

$$\mathcal{U}\left(\left[-\sqrt{\frac{6}{d_{\ell-1}+d_\ell}}, \sqrt{\frac{6}{d_{\ell-1}+d_\ell}}\right]\right).$$

The goal of such a scheme is to maintain a constant signal amplitude throughout the network which allows stable gradient back-propagation. The use of a uniform distribution instead of a normal distribution or any other distribution is not clearly motivated.

It is also possible to initialize NN weights in a deterministic way. A typical method consists in training a NN on another similar dataset and retrieving the weights to use them as a starting point for the initial task on the original dataset (Zhuang et al. 2020).Another idea is to benefit from the training of another kind of estimator on the original dataset (such as tree-based methods) in order to initialize an MLP. This approach was considered in Welbl 2014 and Biau et al. 2019. Remark that this scheme can constrain the NN architecture.

1.2.2 Challenges

The huge and sudden success of NN 10 years ago challenged our mathematical understanding of many aspects of machine learning and statistics, the main ones being generalization, optimization, and approximation. We stress the non-exhaustive character of this list: NN have been the sources of an uncountable number of publications over the last 15 years, and the study of NN involves many fields. It is hard to enumerate them all concisely, and we focus on the "main" ones considered from a statistical perspective: i) approximation and ii) generalization. We refer the reader to Sun 2020 for a thorough presentation of optimization, or to Belkin 2021 for a high-level introduction to overparameterized NN.

Approximation The first universal approximation results on NN in the 1990s (Cybenko 1989; Hornik et al. 1989) (Pinkus 1999 provides a detailed review of such results) suggested that NN needed an exponential number of parameters (w.r.t. the dimension of the input space) in order to become *universal approximators*. The benefits of using several hidden layers compared to a single layer were still unclear. In practice, NN with many layers and (relatively) few parameters per layer still achieve low training error (low bias). Many articles try to close the gap between

practice and theory by demonstrating approximation rates that seem closer to reality in terms of number of parameters, dimension of the input space and hypothesis on $f^\star$. The problem of evaluating the approximation power of NN can be addressed w.r.t. the *regularity* of the regression function.

The first approximation results were based on typical regularity hypothesis in the sense of functional analysis, i.e., *smoothness* hypothesis (including $f^\star$ being continuous, Lipschitz or $\mathbb{L}^p$ ($p \geqslant 1$) on a compact set of $\mathbb{R}^d$, etc.) Under these assumptions, the number of parameters required by NN to approximate any $f^\star$ at precision ε is typically in $O(\varepsilon^d)$ (Pinkus 1999).

Another line of search consists of looking for *regularities* that neural networks can leverage in order to improve their approximation rates (w.r.t. their number of parameters). Three families of such *regularities* can be identified: i) a *hierarchical* structure ii) a *sparse* structure and iii) a *NN-like* continuous structure. The *hierarchical structure* can refer to a signal that is composed of different levels of expression such as text (letters, words, phrases, etc.) and can be modelled as a composition of different functions. *Sparsity* is a more traditional hypothesis that statisticians consider. It can be seen as a way to model the fact that the signal is null in a part of the space. One way to express it is to define $f^\star$ as living in a submanifold of dimension d' living in an ambient space of dimension $d > d'$. Finally, the last class mostly refers to the class of *Barron functions* (Barron 1994), which can be seen as a continuous extension of NN in the space of functions. As explained in Petersen 2020, under such assumptions, it is possible to obtain much better rates that are independent of the ambient dimension of the input space.

Generalization Overparametrized NN are able to generalize well (reach a low error on a test set) despite having a zero training error, i.e. perfectly fitting the training set. This observation, empirical at first, is currently being studied by statisticians, who sometimes refer to it as the *implicit regularization* occurring during NN training. We discuss in more details in Section 1.3 of this type of favorable behavior that overparametrized / interpolating predictors can exhibit and that requires generally a particular theoretical analysis. Another way of analyzing the generalization error of overparameterized NN is to adopt a PAC-Bayes approach.

In one sentence, PAC-Bayes allows one to upper bound the excess risk of a probability distribution Q on the parameters θ of a NN by a sum of i) the empirical risk of Q and ii) a Kullback-Leibler (KL) divergence between Q and a reference distribution Q_0. Remarkably, the first non-vacuous generalization bounds on NN were obtained a few years ago in Dziugaite et al. 2017. After that, many efforts have been conducted to improve the PAC-Bayes bounds in order to close the gap between theory and practice that we will present in Chapter 5. Here, we simply guide the interested reader to the great introduction of P. Alquier (Alquier 2023).

1.3 Interpolation - Regularization

In the previous sections, we have introduced NN and RF, two powerful estimators that have the ability to *perfectly fit* the data, i.e. to reach a loss equal to 0 on the training set. This phenomenon is called *interpolation*, and the purpose of this section is to provide a different light on the behavior of several estimators through the prism of interpolation. We first look at NN which are historically the first successful interpolators, then at kernel methods which attain minimax rates in interpolation regimes under mild assumptions, and finally at RF which are at the core of the present work. Nice and friendly introductions to this subject can be found in Belkin 2021; Bartlett et al. 2021.

1.3.1 Motivation - Neural Networks

The paradigm stating that high model complexity leads to bad generalization capacity has recently been challenged by the behavior of NN: deeper and larger NN still empirically exhibit high predictive performances (Goodfellow et al. 2016). NN are indeed the first algorithms to achieve a very low training loss (close to 0) on large datasets while maintaining a low *generalization gap*, i.e. a small difference between the training and the testing losses. In that sense, they avoid *overfitting* the data and the traditional *bias-variance tradeoff* falls short to explain their good results and to help designing this kind of estimator. The question raised by this observation can be expressed as follows: how does an overparametrized NN *interpolating* the training data still maintains good generalization performances? This question is vast and is explored in, e.g., Bartlett et al. 2021. We focus here on a specific aspect, that is, what *regularization processes* allow very complex NN with *low bias* to preserve a *low variance*?

Implicit regularization Consider a typical regression model as introduced in Equation (1.1) with

$$Y = f^\star(X) + \varepsilon$$

where ε is a random variable independent of all (X_i, Y_i) that verifies $\mathbb{E}[\varepsilon] = 0$, $\mathbb{E}\left[\varepsilon^2\right] = \sigma^2$. If the model is overparametrized, it can easily fit the data, and at locations X_i, the prediction of is close to $Y_i = f(X_i) + \varepsilon_i$ which is noisy. The fact that overparametrized NN preserve a good generalization score, despite interpolating the training data, means that its global sensitivity to the noise is very low. In other words, *regularization processes* are strong enough to constrain the noise sensitivity locally and allow the NN to properly estimate the signal in most of the space. This phenomenon is often referred to as *implicit regularization* as, *a priori*, NN are trained to minimize the empirical risk without explicit complexity penalty. This phenomenon seems driven by two key components: the hyperparameters of the NN (i.e., width, depth, etc.) and the optimization algorithm (SGD).

Indeed, as shown on Figure 1.4, increasing the number of parameters can decrease the generalization error. It means that, as the number of parameters increases, a stronger regularization process occurs. Either better minima appear (in terms of low complexity, such as quadratic norm), or finding them via SGD becomes easier, or both options take place at the same time.

Regarding the first option, as observed in Belkin 2021, as the size of the functional class of NN increases, the norm of the minimal interpolating NN within this class directly decreases (for any norm or any complexity measure). Therefore, the bigger the class, the better the optimal NN in terms of minimal norm or minimal complexity.

Regarding the second option, according to Belkin 2021, one of the key roles achieved by overparameterization is to release the constraints on the learning problem which eases the search for minimal-complexity estimators (e.g., in the sense of lowest norm). To give a simplistic illustration, in the under-parameterized regime, increasing the generalization capacity of a linear estimator can be done via the addition of a quadratic penalty to the objective. It writes

$$\min_{\beta \in \mathbb{R}^d} ||X\beta - y||_2^2 + \lambda ||\beta||_2^2$$

with $\lambda > 0$ to tune according to a classical bias/variance trade-off. However, in the overparameterized regime, the constraint $X\beta = y$ is easily satisfied, and we can simply look for

$$\min_{\beta \in \mathbb{R}^d s.t. X\beta = y} ||\beta||_2^2.$$

And indeed, it can be easily shown that in the simple cases of overparameterized linear regression,

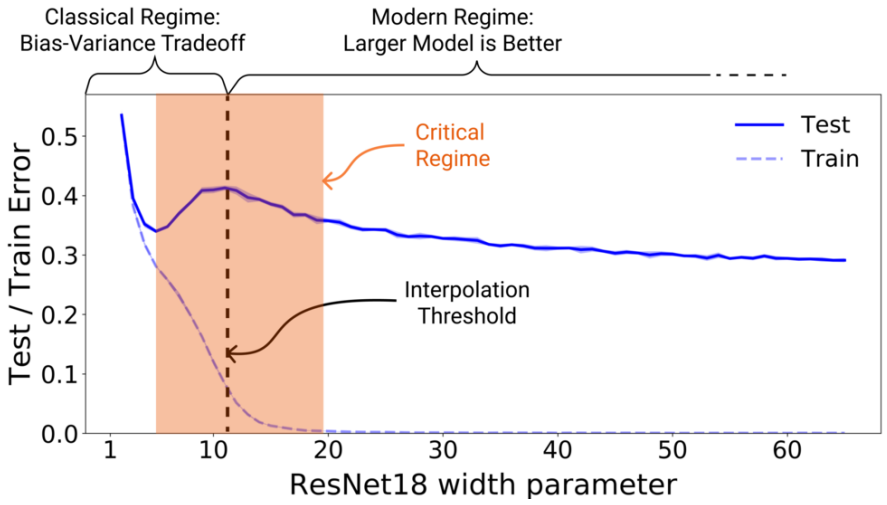


Figure 1.4: Illustration of the double-descent phenomenon: adding parameters to a NN increase its performances after the interpolation threshold. From Nakkiran et al. 2021.

training a linear estimator with GD applied to the quadratic loss yields the minimal-norm estimator (see, e.g., Bartlett et al. 2021).

Several papers try to extend similar results to other settings, including shallow NN. As these results fall outside the scope of this work, we do not examine them further and refer the interested reader to Bach et al. 2021. Finally, remark that the impact of over-parameterization of NN goes far beyond the generalization aspect. In particular, it affects their optimization behavior, as discussed in Sun 2020; Belkin 2021.

1.3.2 Kernel Methods

Kernel methods provide great insights on the behavior of interpolating estimators. They have been theoretically studied in Belkin et al. 2019b and Devroye et al. 1998 for instance. In Belkin et al. 2019b, the authors prove that interpolating singular-kernel methods reach minimax rates in a regression setting under standard assumptions on the regression function. The kernel used is of the form $K : x \to \frac{\mathbb{1}_{||x|| \leq 1}}{||x||}$ which is exploding in 0. Plugging this kernel into a Nadaraya-Watson-type estimator yields

$$f_h(x) := \frac{\sum_{i=1}^n Y_i K\left(\frac{x - X_i}{h}\right)}{\sum_{i=1}^n K\left(\frac{x - X_i}{h}\right)} \tag{1.11}$$

where h is a *window* parameter which sets the averaging level (the greater h, the greater the averaging effect). This interpolating estimator can reach minimax-rates by calibrating h properly. As illustrated in Figure 1.5, the estimator is globally smooth (thanks to the averaging effect) and locally deviates from the signal in order to perfectly fit the training data (thanks to its singular property). Therefore, the estimator is only affected locally by the data noise, which does not deteriorate its convergence to the signal (minimax rate).

In contrast to standard underparametrized methods which suffer from the curse of dimensionality, it is suggested by several authors that regularization of kernel methods could benefit from higher-dimensional settings. For instance, consistency of Laplace kernel method was disproved

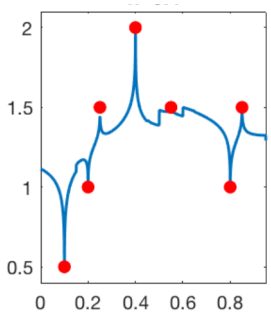


Figure 1.5: Interpolating kernel estimator (from Belkin et al. 2019b).

if the input dimension is constant in Rakhlin et al. 2019. This idea is also supported by Liang et al. 2020a where it is shown that, under several assumptions on the geometry of the data, as d tends to infinity, a phenomenon of implicit regularization starts to occur and the variance decreases toward 0. In Belkin et al. 2019b as well, the variance of the kernel estimator decreases exponentially fast towards 0 w.r.t. the dimension d. The benefits of high dimension are also mentioned in several other contexts (e.g., in terms of computation (Kainen 1997) or optimization (Huang et al. 2020)) and are sometimes referred to as the "blessing of dimensionality" (Kainen 1997; Belkin et al. 2018). To elucidate this phenomenon, we remark that the disparity between "local" and "global" scales augment exponentially with respect to dimensionality. Subsequently, as by nature, the data appears in the form of several discrete points, data noise only manifests at a local level, and a regularized estimator can effectively confine the noise's impact to an assortment of neighborhoods, which contract exponentially with d.

A good illustration of this effect can be found in Belkin et al. 2018 where a simplicial interpolator (a variation of the 1-nearest neighbor estimator) reaches near-optimal convergence rates in a high-dimensional setting. This behavior is in line with the smooth-spiky decomposition mentioned in Bartlett et al. 2021 or in Wyner et al. 2017 for RF and boosting methods: they argue that interpolating estimators can be seen as a sum of a *smooth* term capturing most of the signal and a *spiky* one locally deviating from the signal to interpolate the data.

Finally, we also mention Wang et al. 2022 who recently proved consistency of interpolating kernel methods, defined on Riemannian manifolds, whose kernels can be written as weighted random partition kernels on the sphere.

1.3.3 Random Forests

Understanding the behavior of interpolating RF is a natural question, as RF are usually designed with deep trees (see the default versions in Scikit-learn (Pedregosa et al. 2011) or `randomForest` in **R**) and still yield good performances in generalization. Furthermore, as explained in 1.1.2, in the interpolating (or high-depth) regime, the variance reduction is only due to the randomization processes of the RF. Therefore, proving any variance reduction in this regime (or even better, consistency) is of particular interest as it enlightens the benefits we get from averaging randomized trees in a RF fashion. Finally, from a more mathematical perspective, it also globally improves our understanding of interpolating methods with respect to the behavior of complex/regularized methods and to the smooth/spiky decomposition as introduced in Bartlett et al. 2021. For instance, Wyner et al. 2017 argues that the interpolation property is a key element explaining the good performances of RF. Their empirical analysis relates to the smooth-spiky decomposition already discussed in the case of NN and kernel methods. They advocate that deep (and spiky) RF are

more robust to the noise compared to shallow (and "smooth") RF as their high complexity could allow them to circumscribe noise sensitivity locally, as illustrated on Figure 1.6. This hypothesis is difficult to assess theoretically, as it would require a tight control over the dependence of the estimation error on the depth of the RF k_n.

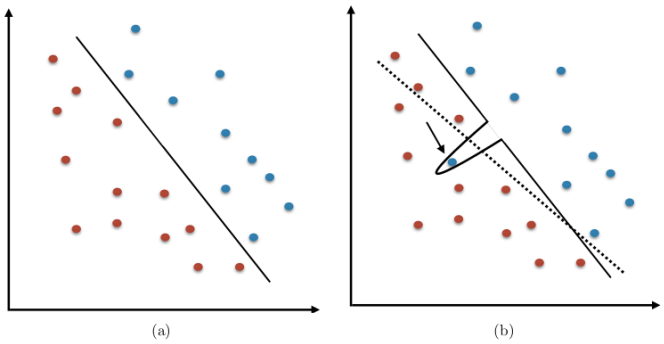


Figure 1.6: Abstract illustration of the benefits of interpolation. (a) SVM estimator. (b) An interpolating estimator robust to data noise (filled lines) and a classical SVM sensitive to data noise (dashed lines). From Wyner et al. 2017.

1.3.4 The Interpolation Prism - Open Questions

After this brief presentation of the behavior of some interpolating estimators, it seems natural to wonder if interpolation is intrinsically linked to their good performances or if they perform well *despite* perfectly fitting the training data. To clarify our ideas, we begin with a short summary of this section and introduce a short comparison with the standard underparametrized regime.

1. **NN.** Overparameterized NN trained with SGD benefit from an *implicit bias* and reach *good minima* (in the sense of low complexity). Here, interpolation is a direct consequence of overparameterization which, intuitively, eases the optimization problem w.r.t. the search of a good minimum obtained via SGD. In practice, it seems that increasing the number of parameters can improve both the train and test errors (see, e.g., the performances of different NN on ImageNet w.r.t. their number of parameters [link]).

2. **Kernel Methods.** Singular kernel estimators are a great illustration of smooth-spiky interpolators. Their noise robustness proceeds from their ability to deviate locally from the signal (and thus interpolate). Although they can achieve minimax rates on the class of β-Hölder functions ($\beta \in (0, 2]$), it is unclear whether the interpolating power of the kernel benefits the estimator or not. Indeed, as shown in Larry Wasserman's course on density estimation [link] , non-singular kernel estimators can also reach minimax rates under the same hypotheses.

3. **RF.** The design of deep RF is also a good case-study of interpolating estimators: from a low-bias CART highly sensitive to data noise, a regularization process (diversification and averaging) is introduced to diminish the variance while maintaining a low bias.

Can we draw a bigger picture from these elements, that is, can we establish a comparison between the classical and interpolation regimes that is not specific to a kind of estimator? An eventual path toward answering this question would be to introduce a measure to quantify either

the regularization of the algorithm or its complexity. This obviously requires a proper definition of complexity and/or of regularization based on, for instance, a specific norm, how many points are averaged to make a prediction, etc. W.r.t. this quantity, the double-descent curve would be a single descent, also observed in the case of RF or kernel methods as well.

Nevertheless, to design good estimators, regularizing a very complex method seems very efficient, maybe more than optimizing a bias-variance tradeoff based on a limitation of the complexity of the estimator.

1.4 Summary of Contributions

Two of my works (Chapters 2 and 3) study or use the Deep Forest algorithm (Zhou et al. 2017). Deep Forest (DF) is a RF-based algorithm that stacks RF in a NN fashion, each RF of each layer taking as input the outputs of the RF of the previous layer. The improvements of this estimator, compared to a single RF, raise the question of how DT/RF benefit from the prediction of previous DT/RF. We investigate this problem in Chapter 2. In Chapter 3, we try to improve the performances of MLP on tabular data by leveraging pretrained tree-based methods (including DF) to initialize the network.

Finally, in Chapter 4, we examine the behavior of RF algorithm in the *interpolation regime*, thus extending the study of interpolating estimators (neural networks and kernel methods) to random forests.

The first three chapters have been published in international conferences on machine learning. The last chapter is dedicated to an ongoing work that has not been published yet. A more detailed description of each chapter is to be found below.

- **Chapter 2** focuses on studying the DF algorithm both empirically and theoretically. This work, published in ICML 21 (Arnould et al. 2021), has been conducted under the supervision of my Ph.D. advisors Claire Boyer and Erwan Scornet.
- **Chapter 3** presents an initialization scheme for NN derived from tree-based methods in order to improve the generalization performances and optimization of the NN. This work has been published in ICLR 2023 (Lutz et al. 2022). It is a joint work with Patrick Lutz (Boston University), under the supervision of my Ph.D. advisors.
- **Chapter 4** is a theoretical study of RF consistency in the interpolation regime. It has been published in AISTATS 2023 (Arnould et al. 2023). This work was carried out under the supervision of my Ph.D. advisors.
- **Chapter 5** presents an attempt to leverage PAC-Bayes objective to improve the generalization of NN. This work is currently supervised by Benjamin Guedj (UCL), it has not been submitted for publication yet.

1.4.1 Analysis of the Deep Forest Algorithm

Chapter 2 focuses on the study of the Deep Forest estimator. More precisely, we analyze the benefit of combining trees in network architectures, both theoretically and numerically. As DF performance has already been validated by the literature (Zhou et al. 2017), the main goals of our study are (i) to quantify the potential benefits of DF over RF, and (ii) to understand the mechanisms at work in such complex architectures. We show in particular that much lighter configuration can be on par with DF default configuration, leading to a drastic reduction of

the number of parameters in few cases. For most datasets, considering DF with two layers is already an improvement over the basic RF algorithm. However, the performance of the overall method is highly dependent on the structure of the random forests present in the first layer of the DF architecture, which leads to instability problems. By establishing tight lower and upper bounds on the risk, we prove that a shallow tree-network may outperform an individual tree in the specific case of a well-structured dataset if the first encoding tree is rich enough. This is a first step to understand the interest of extracting features from trees, and more generally the benefit of tree networks.

1.4.2 Initialization of NN from Tree-Based Methods

In Chapter 3, we propose a new method to initialize a potentially deep MLP for learning tasks with tabular data. Indeed, as discussed in Section 1.2.1, the performances of NN on tabular data are lacking compared to other data types, and we leverage the strong performances of tree-based methods to improve them. Our method consists in first training a tree-based predictor (RF, GBDT or Deep Forest) and then using its translation into an MLP as initialization for the first two layers, the deeper ones being randomly initialized. With the subsequent standard GD training, this procedure is shown to outperform the widely used uniform initialization of MLP (Paszke et al. 2019). It improves the final generalization score of the MLP and accelerates the training process. Initializing the first few layers of the MLP with the translation of the tree-based method and initializing randomly the deeper layers proved to be very successful. This supports the idea that, in our method, the (first) tree-based initialized layers act as relevant feature extractors that allow the MLP to detect the dependence structure in the inputs.

1.4.3 Theoretical Study of Interpolating RF

Following the introduction of *interpolating* estimators in Section 1.3, we study in Chapter 4 the trade-off between interpolation and consistency, in the context of regression, for different types of RF. We prove theoretically that interpolation regimes and consistency cannot be achieved simultaneously for non-adaptive centered RF. The major problem arises from empty cells in tree partitions. We then study a more refined version of the CRF, the Kernel Random Forest (KeRF), built by averaging over all connected data points. By neglecting empty cells, this method is consistent for larger tree depths, but does not meet the exact interpolation requirement. Since adaptivity seems to be the cornerstone to conciliate interpolation and consistency, we study the interpolating Median RF, which is proved to be consistent in the exact interpolation regime. For the first time, it is shown that the averaging effect of the feature randomization inside RF (without bootstrap) is sufficient to "average the noise out" (interpolating trees being sensitive to the noise), i.e. to decrease the variance toward 0. The bias of interpolating trees can still be classically controlled. Numerical experiments show that Breiman RF are consistent when exactly interpolating, i.e. when the whole data set is used to build each fully-grown tree (no bootstrap). It seems that the key randomization mechanism at work in RF is sufficient to achieve consistency in spite of interpolation. Finally, we prove that the volume of the interpolation zone (where the noise sensitivity is maximum) for an infinite Breiman RF tends to 0 at an exponential rate in the dimension d. This supports the idea that the decay of the interpolation volume could be fast enough to retrieve consistency despite interpolation.

1.4.4 PAC-Bayes Objective for NN Training

This ongoing work is presented in Chapter 5 and has not yet been published. PAC-Bayes is a mathematical framework used to derive generalization bounds. It is based on the work of

Perez-Ortiz et al. 2021 which introduces PAC-Bayes-inspired training objectives to optimize (probabilistic) neural networks to obtain generalization guarantees. Our objectives are two-folds: i) improving the generalization capacities of NN directly on the training set by reducing the empirical generalization gap and ii) studying the curvature of NN hessian (via its eigenvalues) when minimizing PAC-Bayes objectives. The last point is motivated by the belief that flatter minima generalize better, as discussed in Section 1.2.2.

Bibliography of the current chapter

Alquier, Pierre (2023). *User-friendly introduction to PAC-Bayes bounds*. arXiv: 2110 . 11216 [stat.ML].

Arnould, Ludovic, Claire Boyer, and Erwan Scornet (2021). "Analyzing the tree-layer structure of Deep Forests". In: *International Conference on Machine Learning*. PMLR, pp. 342–350.

— (2023). "Is interpolation benign for random forest regression?" In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 5493–5548.

Bach, Francis and Lenaic Chizat (2021). *Gradient Descent on Infinitely Wide Neural Networks: Global Convergence and Generalization*. arXiv: 2110.08084 [cs.LG].

Barron, Andrew R (1994). "Approximation and estimation bounds for artificial neural networks". In: *Machine learning* 14, pp. 115–133.

Bartlett, Peter L, Andrea Montanari, and Alexander Rakhlin (2021). "Deep learning: a statistical viewpoint". In: *arXiv preprint arXiv:2103.09177*.

Belgiu, Mariana and Lucian Drăguţ (2016). "Random forest in remote sensing: A review of applications and future directions". In: *ISPRS journal of photogrammetry and remote sensing* 114, pp. 24–31.

Belkin, Mikhail (2021). "Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation". In: *Acta Numerica* 30, pp. 203–248.

Belkin, Mikhail, Daniel Hsu, and Partha Mitra (2018). "Overfitting or perfect fitting? risk bounds for classification and regression rules that interpolate". In: *arXiv preprint arXiv:1806.05161*.

Belkin, Mikhail, Alexander Rakhlin, and Alexandre B Tsybakov (2019b). "Does data interpolation contradict statistical optimality?" In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 1611–1619.

Biau, G., L. Devroye, and G. Lugosi (2008). "Consistency of random forests and other averaging classifiers". In: *Journal of Machine Learning Research* 9.Sep, pp. 2015–2033.

Biau, Gérard (2012b). "Analysis of a random forests model". In: *The Journal of Machine Learning Research* 13, pp. 1063–1095.

Biau, Gérard and Erwan Scornet (2016). "A random forest guided tour". In: *Test* 25.2, pp. 197–227.

Biau, Gérard, Erwan Scornet, and Johannes Welbl (2019). "Neural random forests". In: *Sankhya A* 81.2, pp. 347–386.

Bottou, Léon, Frank E Curtis, and Jorge Nocedal (2018). "Optimization methods for large-scale machine learning". In: *SIAM review* 60.2, pp. 223–311.

Breiman, Leo (1996). "Bagging predictors". In: *Machine learning* 24, pp. 123–140.

— (2001a). "Random forests". In: *Machine learning* 45.1, pp. 5–32.

— (2004). "Consistency for a simple model of random forests". In: *University of California at Berkeley. Technical Report* 670.

Breiman, Leo, Jerome Friedman, Charles J Stone, and Richard A Olshen (1984). *Classification and regression trees*. CRC press.

Bühlmann, Peter and Bin Yu (2002). "Analyzing bagging". In: *The annals of Statistics* 30.4, pp. 927–961.

Chen, Xi and Hemant Ishwaran (2012). "Random forests for genomic data analysis". In: *Genomics* 99.6, pp. 323–329.

Chi, Chien-Ming, Patrick Vossler, Yingying Fan, and Jinchi Lv (2022). "Asymptotic properties of high-dimensional random forests". In: *The Annals of Statistics* 50.6, pp. 3415–3438.

Criminisi, Antonio, Jamie Shotton, Ender Konukoglu, et al. (2012). "Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning". In: *Foundations and trends® in computer graphics and vision* 7.2–3, pp. 81–227.

Cybenko, G. (Dec. 1989). "Approximation by superpositions of a sigmoidal function". In: *Mathematics of Control, Signals and Systems* 2.4, pp. 303–314. ISSN: 1435-568X. DOI: 10.1007/BF02551274. URL: https://doi.org/10.1007/BF02551274.

Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei (2009). "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee, pp. 248–255.

Devroye, Luc, Laszlo Györfi, and Adam Krzyżak (1998). "The Hilbert kernel regression estimate". In: *Journal of Multivariate Analysis* 65.2, pp. 209–227.

Díaz-Uriarte, Ramón and Sara Alvarez de Andrés (2006). "Gene selection and classification of microarray data using random forest". In: *BMC bioinformatics* 7, pp. 1–13.

Duroux, Roxane and Erwan Scornet (2018). "Impact of subsampling and tree depth on random forests". In: *ESAIM: Probability and Statistics* 22, pp. 96–128.

Dziugaite, Gintare Karolina and Daniel M Roy (2017). "Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data". In: *arXiv preprint arXiv:1703.11008*.

Elie-Dit-Cosaque, Kevin and Véronique Maume-Deschamps (2022). "Random forest estimation of conditional distribution functions and conditional quantiles". In: *Electronic Journal of Statistics* 16.2, pp. 6553–6583.

Genuer, Robin, Jean-Michel Poggi, and Christine Tuleau-Malot (2010). "Variable selection using random forests". In: *Pattern recognition letters* 31.14, pp. 2225–2236.

Glorot, Xavier and Yoshua Bengio (2010). "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, pp. 249–256.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep learning*. MIT press.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2015). "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". In: *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034.

Hornik, Kurt, Maxwell Stinchcombe, and Halbert White (1989). "Multilayer feedforward networks are universal approximators". In: *Neural networks* 2.5, pp. 359–366.

Huang, W Ronny et al. (2020). "Understanding generalization through visualizations". In.

Kainen, Paul C. (1997). "Utilizing Geometric Anomalies of High Dimension: When Complexity Makes Computation Easier". In: *Computer Intensive Methods in Control and Signal Processing: The Curse of Dimensionality*. Ed. by Miroslav Kárný and Kevin Warwick. Boston, MA: Birkhäuser Boston, pp. 283–294. ISBN: 978-1-4612-1996-5. DOI: 10.1007/978-1-4612-1996-5_18. URL: https://doi.org/10.1007/978-1-4612-1996-5_18.

Klusowski, Jason M. (2021a). "Sharp analysis of a simple model for random forests". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 757–765.

— (2021b). "Universal consistency of decision trees in high dimensions". In: *arXiv preprint arXiv:2104.13881*.

Kobak, Dmitry, Jonathan Lomond, and Benoit Sanchez (2020). "The optimal ridge penalty for real-world high-dimensional data can be zero or negative due to the implicit ridge regularization". In: *The Journal of Machine Learning Research* 21.1, pp. 6863–6878.

Lakshminarayanan, Balaji, Daniel M Roy, and Yee Whye Teh (2014). "Mondrian forests: Efficient online random forests". In: *Advances in neural information processing systems* 27.

LeCun, Yann, Yoshua Bengio, et al. (1995). "Convolutional networks for images, speech, and time series". In: *The handbook of brain theory and neural networks* 3361.10, p. 1995.

Liang, Tengyuan and Alexander Rakhlin (2020a). "Just interpolate: Kernel "ridgeless" regression can generalize". In.

Lin, Yi and Yongho Jeon (2006). "Random forests and adaptive nearest neighbors". In: *Journal of the American Statistical Association* 101.474, pp. 578–590.

Lugosi, Gábor and Andrew Nobel (1996). "Consistency of data-driven histogram methods for density estimation and classification". In: *The Annals of Statistics* 24.2, pp. 687–706.

Lutz, Patrick, Ludovic Arnould, Claire Boyer, and Erwan Scornet (2022). "Sparse tree-based initialization for neural networks". In: *arXiv preprint arXiv:2209.15283*.

Mentch, Lucas and Siyu Zhou (2019). "Randomization as regularization: a degrees of freedom explanation for random forest success". In: *arXiv preprint arXiv:1911.00190*.

— (2022). "Getting better from worse: Augmented bagging and a cautionary tale of variable importance". In: *Journal of Machine Learning Research* 23.224, pp. 1–32.

Mourtada, Jaouad, Stéphane Gaïffas, and Erwan Scornet (2020). "Minimax optimal rates for Mondrian trees and forests". In: *The Annals of Statistics* 48.4, pp. 2253–2276.

Nakkiran, Preetum, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever (2021). "Deep double descent: Where bigger models and more data hurt". In: *Journal of Statistical Mechanics: Theory and Experiment* 2021.12, p. 124003.

Nobel, Andrew (1996). "Histogram regression estimation using data-dependent partitions". In: *The Annals of Statistics* 24.3, pp. 1084–1105.

Paszke, Adam et al. (2019). "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., pp. 8024–8035.

Pedregosa, F. et al. (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Perez-Ortiz, Maria, Omar Rivasplata, Emilio Parrado-Hernandez, Benjamin Guedj, and John Shawe-Taylor (2021). "Progress in Self-Certified Neural Networks". In: *arXiv preprint arXiv:2111.07737*.

Petersen, Philipp Christian (2020). "Neural network theory". In: *University of Vienna*.

Pinkus, Allan (1999). "Approximation theory of the MLP model in neural networks". In: *Acta numerica* 8, pp. 143–195.

Prasad, Anantha M, Louis R Iverson, and Andy Liaw (2006). "Newer classification and regression tree techniques: bagging and random forests for ecological prediction". In: *Ecosystems* 9, pp. 181–199.

Rakhlin, Alexander and Xiyu Zhai (2019). "Consistency of interpolation with Laplace kernels is a high-dimensional phenomenon". In: *Conference on Learning Theory*. PMLR, pp. 2595–2623.

Ruder, Sebastian (2016). "An overview of gradient descent optimization algorithms". In: *arXiv preprint arXiv:1609.04747*.

Rumelhart, David E, Geoffrey E Hinton, James L McClelland, et al. (1986). "A general framework for parallel distributed processing". In: *Parallel distributed processing: Explorations in the microstructure of cognition* 1.45-76, p. 26.

Rumelhart, David E, Geoffrey E Hinton, and Ronald J Williams (1985). *Learning internal representations by error propagation*. Tech. rep. California Univ San Diego La Jolla Inst for Cognitive Science.

Scornet, Erwan (2016a). "On the asymptotics of random forests". In: *Journal of Multivariate Analysis* 146, pp. 72–83.

Scornet, Erwan, Gérard Biau, and Jean-Philippe Vert (2015). "Consistency of random forests". In: *The Annals of Statistics* 43.4, pp. 1716–1741.

Shrestha, Ajay and Ausif Mahmood (2019). "Review of deep learning algorithms and architectures". In: *IEEE access* 7, pp. 53040–53065.

Sun, Ruo-Yu (2020). "Optimization for deep learning: An overview". In: *Journal of the Operations Research Society of China* 8.2, pp. 249–294.

Tan, Zhiqiang and Cun-Hui Zhang (2019). "Doubly penalized estimation in additive regression with high-dimensional data". In.

Vaswani, Ashish et al. (2017). "Attention is all you need". In: *arXiv preprint arXiv:1706.03762*.

Wager, Stefan and Guenther Walther (2015). "Adaptive concentration of regression trees, with application to random forests". In: *arXiv preprint arXiv:1503.06388*.

Wang, Yutong and Clayton D Scott (2022). "Consistent Interpolating Ensembles via the Manifold-Hilbert Kernel". In: *arXiv preprint arXiv:2205.09342*.

Welbl, Johannes (2014). "Casting random forests as artificial neural networks (and profiting from it)". In: *German Conference on Pattern Recognition*. Springer, pp. 765–771.

Wyner, Abraham J, Matthew Olson, Justin Bleich, and David Mease (2017). "Explaining the success of adaboost and random forests as interpolating classifiers". In: *The Journal of Machine Learning Research* 18.1, pp. 1558–1590.

Zhou, Z and J. Feng (2017). "Deep Forest: Towards An Alternative to Deep Neural Networks". In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 3553–3559.

Zhuang, Fuzhen et al. (2020). "A comprehensive survey on transfer learning". In: *Proceedings of the IEEE* 109.1, pp. 43–76.

Chapter 2

Analyzing Deep Forest

Abstract

Random forests on the one hand, and neural networks on the other hand, have met great success in the machine learning community for their predictive performance. Combinations of both have been proposed in the literature, notably leading to the so-called deep forests (DF, Zhou et al. 2017). In this paper, our aim is not to benchmark DF performances, but to investigate instead their underlying mechanisms. Additionally, DF architecture can be generally simplified into more simple and computationally efficient shallow forests networks. Despite some instability, the latter may outperform standard predictive tree-based methods. We exhibit a theoretical framework in which a shallow tree network is shown to enhance the performance of classical decision trees. In such a setting, we provide tight theoretical lower and upper bounds on its excess risk. These theoretical results show the interest of tree-network architectures for well-structured data provided that the first layer, acting as a data encoder, is rich enough.

2.1 Introduction

Deep Neural Networks (DNNs) are among the most widely used machine learning algorithms. They are composed of parameterized differentiable non-linear modules trained by gradient-based methods, which rely on the backpropagation procedure. Their performance mainly relies on layer-by-layer processing as well as feature transformation across layers. Training neural networks usually requires complex hyper-parameter tuning (Bergstra et al. 2011) and a huge amount of data. Although DNNs recently achieved great results in many areas, they remain very complex to handle and unstable to input noise (Zheng et al. 2016).

Recently, several attempts have been made to consider networks with non-differentiable modules. Among them the Deep Forest (DF) algorithm (Zhou et al. 2017), which uses Random Forests (RF, Breiman 2001a) as neurons, has received a lot of attention in recent years in various applications such as hyperspectral image processing (Liu et al. 2020), medical imaging (Sun et al. 2020), drug interactions (Su et al. 2019; Zeng et al. 2020) or even fraud detection (Zhang et al. 2019).

Since the DF procedure stacks multiple layers, each one being composed of complex nonparametric RF estimators, the rationale behind the procedure remains quite obscure. However DF methods exhibit impressive performances in practice, suggesting that stacking RFs and extracting features from these estimators at each layer is a promising way to leverage on the RF performance

in the neural network framework. The goal of this paper is not an exhaustive empirical study of prediction performances of DF (see Zhou et al. 2017) but rather to understand how stacking trees in a network fashion may result in competitive infrastructure.

Related Works. Different manners of stacking trees exist (see Ghods et al. 2020 for a general survey on stacking methods), as the Forwarding Thinking Deep Random Forest (FTDRF), proposed by Miller et al. 2017, for which the proposed network contains trees which directly transmit their output to the next layer (contrary to Deep Forest in which their output is first averaged before being passed to the next layer). A different approach by Feng et al. 2018 consists in rewriting tree gradient boosting as a simple neural network whose layers can be made arbitrary large depending on the boosting tree structure. The resulting estimator is more simple than DF but does not leverage on the ensemble method properties of random forests.

In order to prevent overfitting and to lighten the model, several ways to simplify DF architecture have been investigated. Pang et al. 2018 considers RF whose complexity varies through the network, and combines it with a confidence measure to pass high confidence instances directly to the output layer. Other directions towards DF architecture simplification are to play on the nature of the RF involved (Berrouachedi et al. 2019b, using Extra-Trees instead of Breiman's RF), on the number of RF per layer (Jeong et al. 2020, implementing layers of many forests with few trees), or even on the number of features passed between two consecutive layers (Su et al. 2019) by relying on an importance measure to process only the most important features at each level. The simplification can also occur once the DF architecture is trained, as in Kim et al. 2020 selecting in each forest the most important paths to reduce the network time- and memory-complexity. Approaches to increase the approximation capacity of DF have also been proposed by adjoining weights to trees or to forests in each layer (Utkin et al. 2017; Utkin et al. 2020), replacing the forest by more complex estimators (cascade of ExtraTrees Berrouachedi et al. 2019a), or by combining several of the previous modifications notably incorporating data preprocessing (Guo et al. 2018). Overall, the related works on DF exclusively represent algorithmic contributions without a formal understanding of the driving mechanisms at work inside the forest cascade.

Contributions. In this paper, we analyze the benefit of combining trees in network architecture both theoretically and numerically. As the performances of DF have already been validated by the literature (see Zhou et al. 2017), the main goals of our study are (i) to quantify the potential benefits of DF over RF, and (ii) to understand the mechanisms at work in such complex architectures. We show in particular that much lighter configuration can be on par with DF default configuration, leading to a drastic reduction of the number of parameters in few cases. For most datasets, considering DF with two layers is already an improvement over the basic RF algorithm. However, the performance of the overall method is highly dependent on the structure of the first random forests, which leads to stability issues. By establishing tight lower and upper bounds on the risk, we prove that a shallow tree-network may outperform an individual tree in the specific case of a well-structured dataset if the first encoding tree is rich enough. This is a first step to understand the interest of extracting features from trees, and more generally the benefit of tree networks.

Agenda. DF are formally described in Section 2.2. Section 2.3 is devoted to the numerical study of DF, by evaluating the influence of the number of layers in DF architecture, by showing that shallow sub-models of one or two layers perform the best, and finally by understanding the influence of tree depth in cascade of trees. Section 2.4 contains the theoretical analysis of the shallow centered tree network. For reproducibility purposes, all codes together with all experimental procedures are to be found in the supplementary materials.

2.2 Deep Forests

2.2.1 Description

Deep Forest (Zhou et al. 2017) is a hybrid learning procedure in which random forests are used as the elementary components (neurons) of a neural network. Each layer of DF is composed of an assortment of Breiman's forests and Completely-Random Forests (CRF, Fan et al. 2003) and trained one by one. In a classification setting, each forest of each layer outputs a class probability distribution for any query point x, corresponding to the distribution of the labels in the node containing x. At a given layer, the distributions output by all forests of this layer are concatenated, together with the raw data. This new vector serves as input for the next DF layer. This process is repeated for each layer and the final classification is performed by averaging the forest outputs of the best layer (without raw data) and applying the `argmax` function. The overall architecture is depicted in Figure 2.1.

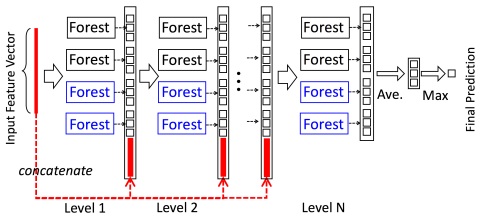


Figure 2.1: Deep Forest architecture (the scheme is taken from Zhou et al. 2017).

2.2.2 DF Hyperparameters

Deep Forests contain an important number of tuning parameters. Apart from the traditional parameters of random forests, DF architecture depends on the number of layers, the number of forests per layer, the type and proportion of random forests to use (Breiman or CRF). In Zhou et al. 2017, the default configuration is set to 8 forests per layer, 4 CRF and 4 RF, 500 trees per forest (other forest parameters are set to `sk-learn` Pedregosa et al. 2011 default values), and layers are added until 3 consecutive layers do not show score improvement.

Due to their large number of parameters and the fact that they use a complex algorithm as elementary bricks, DF consist in a potential high-capacity procedure. However, as a direct consequence, the numerous parameters are difficult to estimate (requiring specific tuning of the optimization process) and need to be stored which leads to high prediction time and large memory consumption. Besides, the layered structure of this estimate, and the fact that each neuron is replaced by a powerful learning algorithm makes the whole prediction hard to properly interpret.

As already pointed out, several attempts to lighten the architecture have been conducted. In this paper, we will propose and assess the performance of a lighter DF configuration on tabular datasets.

Remark 2.2.1. *DF was first designed to classify images. To do so, a pre-processing network called Multi Grained Scanning (MGS) based on convolutions is first applied to the original images. Then the Deep Forest algorithm runs with the newly created features as inputs.*

2.3 Refined Numerical Analysis of DF Architectures

In order to understand the benefit of using a complex architecture like Deep Forests, we compare different configurations of DF on six datasets in which the output is binary, multi-class or continuous, see Table 2.1 for description. All classification datasets belong to the UCI repository, the two regression ones are Kaggle datasets (Housing data and Airbnb Berlin 2020)[1]. Note that the Fashion Mnist features are built using the Multi Grained Scanning process from the DF original article (Zhou et al. 2017) (see S1.3 for the encoding details).

Dataset	Type (Nb of classes)	Train/Val/Test Size	Dim
Adult	Class. (2)	26048/ 6512/ 16281	14
Higgs	Class. (2)	120000/ 28000/ 60000	28
Fashion Mnist	Class (10)	24000/ 6000/ 8000	260
Letter	Class. (26)	12800/ 3200/ 4000	16
Yeast	Class. (10)	830/ 208/ 446	8
Airbnb	Regr.	73044/ 18262/ 39132	13
Housing	Regr.	817/ 205/ 438	61

Table 2.1: Description of the datasets.

In what follows, we propose a light DF configuration. We show that our light configuration performance is comparable to the performance of the default DF architecture of Zhou et al. 2017, thus questioning the relevance of deep models. Therefore, we analyze the influence of the number of layers in DF architectures, showing that DF improvements mostly rely on the first layers of the architecture. To gain insights about the quality of the new features created by the first layer, we consider a shallow tree network for which we evaluate the performance as a function of the first-tree depth.

2.3.1 Towards DF Simplification

Setting. We compare the performances of the following DF architectures on the datasets summarized in Table 2.1:

1. the default setting of DF, described in Section 2.2;
2. the best DF architecture obtained by grid-searching over the number of forests per layer, the number of trees per forest and the maximum depth of each tree. The selected architecture is chosen with respect to the performances achieved on validation datasets;
3. a new light DF architecture, composed of 2 layers, 2 forests per layer (one RF and one CRF) with only 50 trees of depth 30 trained only once;
4. the first layer of the best DF;
5. the first layer of the light DF;
6. a “Flattened best DF as RF" which consists in one RF with as many trees as in the best DF with similar forest parameters (refer to Supplementary Materials S1.2 and Table S3 for details);
7. a “Flattened light DF as RF" which corresponds to one RF with as many trees as in the light DF with similar forest parameters.

[1]https://www.kaggle.com/raghavs1003/airbnb-berlin-2020
https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data

Results. Results are presented in Figures 2.2 and 2.3. Each bar plot respectively corresponds to the average accuracy or the average R^2 score over 10 tries for each test dataset; the error bars stand for accuracy or R^2 standard deviation. The description of the resulting best DF architecture for each dataset is given in Table S3 (Supplementary Materials).

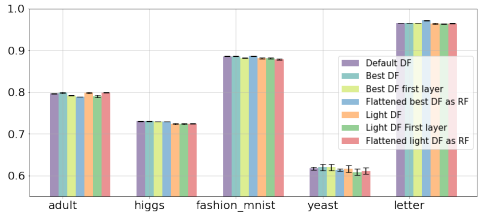


Figure 2.2: Accuracy of different DF architectures for classification datasets (10 runs per bar plot).

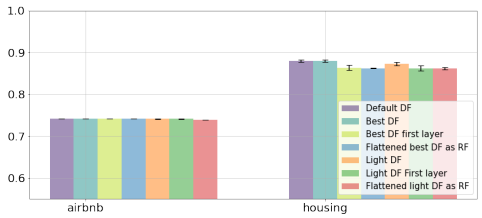


Figure 2.3: R^2 score of different DF architectures for regression datasets (10 runs per bar plot).

As highlighted in Figure 2.2, the performance of the light configuration for classification datasets is comparable to the default and the best configurations' one, while being much more computationally efficient: faster to train, faster at prediction, cheaper in terms of memory (see Table S2 in the Supplementary Materials for a comparison of computing time and memory consumption). Moreover, except on the Letter dataset, the DF performs better than its RF equivalent. The results for the Letter dataset can be explained by the fact that the CRFs within the DF are outperformed by Breiman RFs in this specific case. Overall, for classification tasks, the small performance enhancement of Deep Forests (Default or Best DF) over our light configuration should be assessed in the light of their additional complexity. This questions the usefulness of stacking several layers made of many forests, resulting in a heavy architecture. We further propose an in-depth analysis of the role of each layer to the global DF performance.

2.3.2 Tracking the Best Sub-Model

Setting. On all the previous datasets, we train a DF architecture by specifying the maximal number p of layers. Unspecified hyper-parameters are set to default value (see Section 2.2). For each p, we consider the truncated sub-models composed of layer 1, layer 1-2, ..., layer 1-p, where layer 1-p is the original DF with p layers. For each value of p, we consider the previous nested sub-models with $1, 2, \ldots, p$ layers, and compute the predictive accuracy of the best sub-model.

Results. We only display results for the Adult dataset in Figure 2.4 (all the other datasets show similar results, see Section S1.5 of the Supplementary Materials). The score (accuracy or R^2-score) corresponds to the result on the test dataset. We observe that, over 10 runs, adding layers to the Deep Forest seems not to significantly change the accuracy score. Even if the variance changes

by adding layers, we are not able to detect any pattern, which suggests that the variance of the procedure performance is unstable with respect to the maximal number of layers.

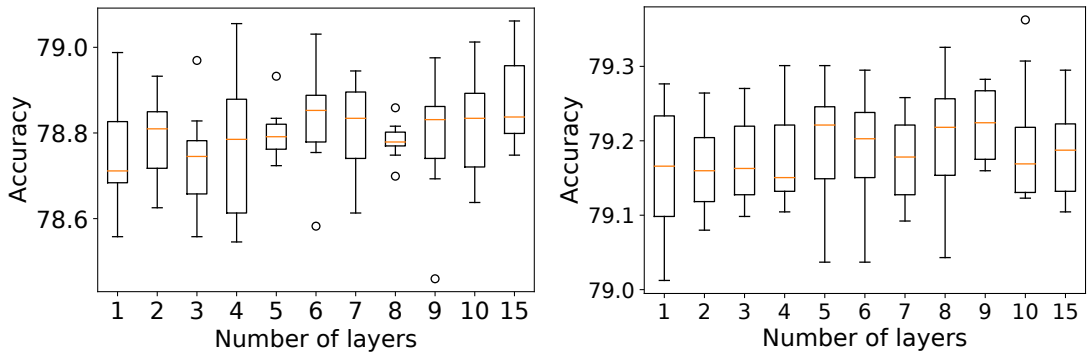


Figure 2.4: Adult dataset. Boxplots over 10 runs of the accuracy of a DF sub-model with 1 (Breiman) forest by layer (left) or 4 forests (2 Breiman, 2 CRF) by layer (right), depending on the maximal number of layers of the global DF model.

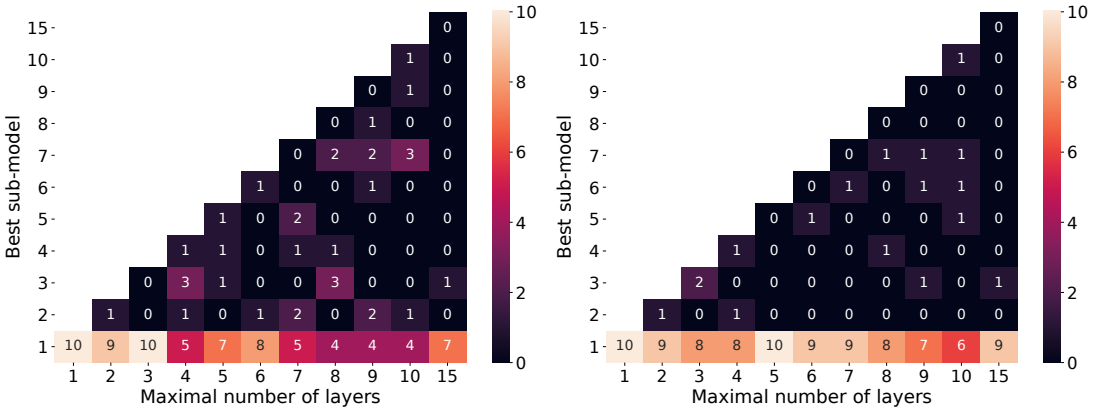


Figure 2.5: Adult dataset. Heatmap counting the optimal layer index over 10 tries of a default DF with 1 (Breiman) forest per layer (left) or 4 forests (2 Breiman, 2 CRF) per layer (right), with respect to the maximal number of layers. The number corresponding to (n, m) on the x- and y-axes indicates how many times out of 10 the layer m is optimal when running a cascade network with a maximal number n of layers.

Globally, we observe that the sub-models with one or two layers often lead to the best performance (see Figure 2.5 for the Adult dataset and Supplementary Materials S1.5). When the dataset is small (Letter or Yeast), the sub-model with only one layer (i.e. a standard RF or an aggregation of RFs) is almost always optimal since a single RF with no maximum depth constraint already overfits on most of these datasets. Therefore the second layer, building upon the predictions of the first layer, entails overfitting as well, therefore leading to no improvement of the overall model. Besides, one can explain the predominance of small sub-models by the weak additional flexibility created by each layer: on the one hand, each new feature vector size corresponds to the number of classes times the number of forests which can be small with respect to the number

of input features; on the other hand, the different forests within one layer are likely to produce similar probability outputs, especially if the number of trees within each forest is large. The story is a bit different for the Housing dataset, for which the best submodel is between 2 and 6. As noticed before, this may be the result of the frustratingly simple representation of the new features created at each layer. Eventually, these numerical experiments corroborate the relevance of shallow DF as the light configuration proposed in the previous section.

We note that adding forests in each layer decreases the number of layers needed to achieve a pre-specified performance. This is surprising and is opposed to the common belief that in Deep Neural Networks, adding layers is usually better than adding neurons in each layer.

We can conclude from the empirical results that the first two layers convey the performance enhancement in DF. Contrary to NNs, depth is not an important feature of DFs. The following studies thus focus on two-layer architectures which are deep enough to reproduce the improvement of deeper architectures over single RFs.

2.3.3 A Precise Understanding of Depth Enhancement

In order to finely grasp the influence of tree depth in DF, we study a simplified version: a shallow CART tree network, composed of two layers, with one CART per layer.

Setting. In such an architecture, the first-layer tree is fitted on the training data. For each sample, the first-layer tree outputs a probability distribution (or a value in a regression setting), which is referred to as "encoded data" and given as input to the second-layer tree, with the raw features as well. For instance, considering binary classification data with classes 0 and 1, with raw features (x_1, x_2, x_3), the input of the second-layer tree is a 5-dimensional feature vector $(x_1, x_2, x_3, p_0, p_1)$, with p_0 (resp. p_1) the predicted probabilities by the first-layer tree for the class 0 (resp. 1).

For each dataset of Table 2.1, we first determine the optimal depth $k^\star$ of a single CART tree via 3-fold cross validation. Then, for a given first-layer tree with a fixed depth, we fit a second-layer tree, allowing its depth to vary. We then compare the resulting shallow tree networks in three different cases: when the (fixed) depth of the first tree is (i) less than $k^\star$, (ii) equal to $k^\star$, and (iii) larger than $k^\star$. We add the optimal single tree performance to the comparison.

Results. Results are displayed in Figure 2.6 for the Adult dataset only (see Supplementary Materials S1.4 for the results on the other datasets). Specifically noticeable in Figure 2.6 (top), the tree network architecture can introduce performance instability when the second-layer tree grows (e.g. when the latter is successively of depth 7, 8 and 9).

Furthermore, when the encoding tree is not deep enough (top), the second-layer tree improves the accuracy until it approximately reaches the optimal depth $k^\star$. In this case, the second-layer tree compensates for the poor encoding, but cannot improve over a single tree with optimal depth $k^\star$. Conversely, when the encoding tree is more developed than an optimal single tree (bottom) - overfitting regime, the second-layer tree may not lead to any improvement, or worse, may degrade the performance of the first-layer tree.

On all datasets, the second-layer tree is observed to always make its first cut over the new features (see Figure 2.7 and Supplementary Materials).

In the case of binary classification, a single cut of the second-layer tree along a new feature yields to gather all the leaves of the first tree, predicted respectively as 0 and 1, into two big leaves, therefore reducing the predictor variance (cf. Figure 2.6 (middle and bottom)). Furthermore, when considering multi-label classification with n_{classes}, the second-layer tree must cut over at least n_{classes} features to recover the partition of the first tree (see Figure S15). Similarly, in the

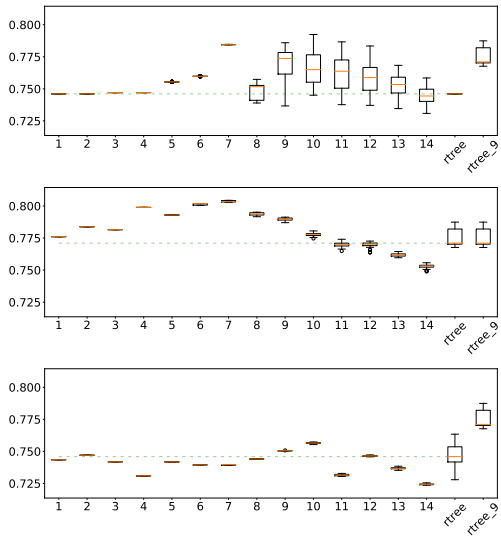


Figure 2.6: Adult dataset. Accuracy on the test dataset of a two-layer tree architecture w.r.t. the second-layer tree depth, when the first-layer (encoding) tree is of depth 2 (top), 9 (middle), and 15 (bottom). `rtree` is a single tree of respective depth 2 (top), 9 (middle), and 15 (bottom), applied on raw data. For this dataset, the optimal depth of a single tree is 9 and the tree with the optimal depth is depicted as `rtree_9` in each plot. The green dashed line indicates the median score of the `rtree`. All boxplots are obtained by 10 different runs.

regression case, the second tree needs to perform a number of splits equal to the number of leaves of the first tree in order to recover the partition of the latter.

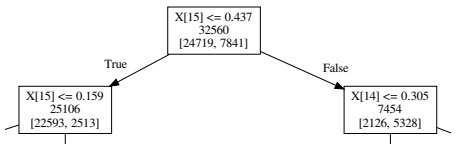


Figure 2.7: Adult dataset. Focus on the first levels of the second-layer tree structure when the first layer tree is of depth 9 (optimal depth). Raw features range from X[0] to X[13], X[14] and X[15] are the features built by the first-layer tree.

In Figure 2.6 (middle), one observes that with a first-layer tree of optimal depth, the second-layer tree may outperform an optimal single tree, by improving both the average accuracy and its variance. We aim at theoretically quantifying this performance gain in the next section.

2.4 Theoretical Study of a Shallow Tree Network

In this section, we focus on the theoretical analysis of a simplified tree network. Our aim is to exhibit settings in which a tree network outperforms a single tree. Recall that the second layer of a tree network gathers tree leaves of the first layer with similar distributions. For this reason, we believe that a tree network is to be used when the dataset has a very specific structure, in which

the same link between the input and the output can be observed in different subareas of the input space. Such a setting is described in Section 2.4.2

To make the theoretical analysis possible, we study centered trees (see Definition 2.4.1) instead of CART. Indeed, studying the original CART algorithm is still nowadays a real challenge and analyzing stacks of CART seems out-of-reach in short term. As highlighted by the previous empirical analysis, we believe that the results we establish theoretically are shared by DF. All proofs are postponed to the Supplementary Materials.

2.4.1 The Network Architecture

We assume to have access to a dataset $\mathcal{D}_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$ of i.i.d. copies of the generic pair (X, Y) with X living in $[0,1]^d$ and $Y \in \{0,1\}$ being the label associated to X.

Notations. Given a decision tree, we denote by $L_n(X)$ the leaf of the tree containing X and $N_n(L_n(X))$ the number of data points falling into $L_n(X)$. The prediction of such a tree at point X is given by

$$f_n(X) = \frac{1}{N_n(L_n(X))} \sum_{X_i \in L_n(X)} Y_i$$

with the convention $0/0 = 0$, i.e. the prediction for X in a leaf with no observations is arbitrarily set to zero.

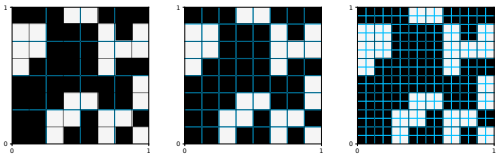
A shallow centered tree network. We want to theoretically analyze the benefits of stacking trees. To do so, we focus on two trees in cascade and will try to determine, in particular, the influence of the first (encoding) tree on the performance of the whole tree network. To catch the variance reduction property of tree networks already emphasized in the previous section, we consider a regression setting: let $f^\star(x) = \mathbb{E}[Y|X = x]$ be the regression function and for any function f, its quadratic risk is defined as $\mathcal{R}(f) = \mathbb{E}[(f(X) - f^\star(X))^2]$, where the expectation is taken over $(X, Y, \mathcal{D}_n)$. We emphasize that although $Y \in \{0,1\}$, we are not interested in classifying Y but rather to estimate the regression function.

Definition 2.4.1 (Shallow centered tree network)**.** *The shallow tree network consists of two trees in cascade:*

- (***Encoding layer***) *The first-layer tree is a* cycling centered tree of depth k. *It is built independently of the data by splitting recursively on each variable, at the center of the cells. The first cut is made along the first coordinate, the second along the second coordinate, etc. The tree construction is stopped when exactly k cuts have been made. For each point X, we extract the empirical mean $\bar{Y}_{L_n(X)}$ of the outputs Y_i falling into the leaf $L_n(X)$ and we pass the new feature $\bar{Y}_{L_n(X)}$ to the next layer, together with the original features X.*
- (***Output layer***) *The second-layer tree is a* centered tree of depth k' *for which a cut can be performed at the center of a cell along a raw feature (as done by the encoding tree) or along the new feature $\bar{Y}_{L_n(X)}$. In this latter case, two cells corresponding to $\{\bar{Y}_{L_n(X)} < 1/2\}$ and $\{\bar{Y}_{L_n(X)} \geqslant 1/2\}$ are created.*

The resulting predictor composed of the two trees in cascade, of respective depth k and k', trained on the data $(X_1, Y_1), \dots, (X_n, Y_n)$ is denoted by $f_{k,k',n}$.

The two cascading trees can be seen as two layers of trees, hence the name of the shallow tree network. Note in particular that $f_{k,0,n}(X)$ is the prediction given by the first encoding tree only and outputs, as a classical tree, the mean of the Y_i's falling into a leaf containing X.



(a) Depth 4 (b) Depth 6 (c) Depth 8

Figure 2.8: Arbitrary chessboard data distribution for $k^\star = 6$ and $N_{\mathcal{B}} = 40$ black cells (p is not displayed here). Partition of the (first) encoding tree of depth 4, 6, 8 (from left to right) is displayed in blue. The optimal depth of a single centered tree for this chessboard distribution is 6.

2.4.2 Problem Setting

Data generation. The data X is assumed to be uniformly distributed over $[0,1]^d$ and $Y \in \{0,1\}$. Let $k^\star$ be a multiple of d and let $p \in (1/2, 1]$. We build a regular partition of the space with *cells* $C_1, \dots, C_{2^{k^\star}}$ of generic form

$$\prod_{k=1}^{d} \left[\frac{i_k}{2^{k^\star/d}}, \frac{i_k+1}{2^{k^\star/d}} \right),$$

for $i_1, ..., i_d \in \{0, ..., 2^{k^\star/d} - 1\}$. We arbitrary assign a color (black or white) to each cell, which has a direct influence on the distribution of Y in the cell. More precisely, for x in a given cell C,

$$\mathbb{P}[Y = 1 | X = x] = \begin{cases} p & \text{if } C \text{ is a black cell,} \\ 1-p & \text{if } C \text{ is a white one.} \end{cases} \tag{2.1}$$

We define $\mathcal{B}$ (resp. $\mathcal{W}$) as the union of black (resp. white) cells and $N_{\mathcal{B}} \in \{0, \dots, 2^{k^\star}\}$ (resp. $N_{\mathcal{W}}$) as the number of black (resp. white) cells. Note that $N_{\mathcal{W}} = 2^{k^\star} - N_{\mathcal{B}}$. The location and the numbers of the black and white cells are arbitrary. This distribution corresponds to a *generalized chessboard* structure. The whole distribution is thus parameterized by $k^\star$ ($2^{k^\star}$ is the total number of cells), p and $N_{\mathcal{B}}$. Examples of this distribution are depicted in Figures 2.8 and 2.9 for different configurations and $d = 2$.

Why such a structured setting? The data distribution introduced above is highly structured, which can be seen as a restrictive study setting. However, the generalized chessboard is nothing but a discretized quantification of the regression function r using only 2 values (see Equation (2.1)). Going further than quantification towards general discretization does not seem appropriate for tree networks. To see this, consider a more general distribution such as

$$\mathbb{P}[Y = 1 | X = x] = P_{ij} \text{ when } x \in C_{ij},$$

where P_{ij} is a random variable drawn uniformly in $[0,1]$.

Lemma 2.4.2. *Assume that the data follows the generalized chessboard distribution described above with parameter $k^\star$, $N_{\mathcal{B}}$ and p. Suppose that $k \geqslant k^\star$. In the infinite sample setting, the risks of a single tree and*

a shallow tree network are given by $\mathcal{R}(f_{k,0,\infty}) = 0$ *and*

$$\mathcal{R}(f_{k,1,\infty}) \geqslant \frac{1}{48}\Big(1 - \frac{8}{2^{k^\star} - 1}\Big) + \frac{1}{2^{2^{k^\star}}}\frac{9}{24}.$$

Lemma 2.4.2 highlights the fact that a tree network has a positive bias, which is not the case for a single tree. Besides, by letting $k^\star$ tend to infinity (that is the size of the cells tends to zero), the above chessboard distribution boils down to a very generic classification framework. In this latter case, the tree network performs poorly since its risk is lower bounded by $1/48$. In short, when the data distribution is disparate across the feature space, the averaging performed by the second tree leads to a biased regressor. Note that Lemma 2.4.2 involves a shallow tree network, performing only one cut on the second layer. But similar conclusions could be drawn for a deeper second-layer tree, until its depth reaches $k^\star$. Indeed, considering $f_{k,k^\star,\infty}$ would result in an unbiased regressor, with comparable performances as of a single tree, while being much more complex.

Armed with Lemma 2.4.2, we believe that the intrinsic structure of DF and tree networks makes them useful to detect similar patterns spread across the feature space. This makes the generalized chessboard distribution particularly well suited for analyzing such behavior. The risk of a shallow tree network in the infinite sample regime for the generalized chessboard distribution is studied in Lemma 2.4.3.

Lemma 2.4.3. *Assume that the data follows the generalized chessboard distribution described above with parameter* $k^\star$, $N_{\mathcal{B}}$ *and* p. *In the infinite sample regime, the following holds for the shallow tree network* $f_{k,k',n}$ (*Definition 2.4.1*).

1. ***Shallow encoding tree.*** *Let* $k < k^\star$. *The risk of the shallow tree network is minimal for all configurations of the chessboard if the second-layer tree is of depth* $k' \geqslant k^\star$ *and if the* $k^\star$ *first cuts are performed along raw features only.*

2. ***Deep encoding tree.*** *Let* $k \geqslant k^\star$. *The risk of the shallow tree network is minimal for all configurations of the chessboard if the second-layer tree is of depth* $k' \geqslant 1$ *and if the first cut is performed along the new feature* $\bar{Y}_{L_n(X)}$.

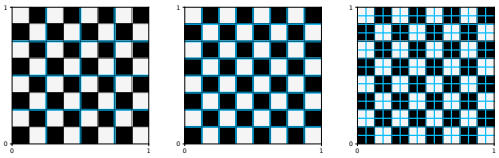
In the infinite sample regime, Lemma 2.4.3 shows that the pre-processing is useless when the encoding tree is shallow ($k < k^\star$): the second tree cannot leverage on the partition of the first one and needs to build a finer partition from zero.

Lemma 2.4.3 also provides an interesting perspective on the second-layer tree which either acts as a copy of the first-layer tree or can simply be of depth one.

Remark 2.4.4. *The results established in Lemma 2.4.3 for centered-tree networks also empirically hold for CART ones* (*see Figures 2.6,S12,S15,S17,S19,S21: (i) the second-layer CART trees always make their first cut on the new feature and always near* $1/2$; *(ii) if the first-layer CART is biased, then the first cuts of the second-layer tree will not improve the accuracy of the first tree and the improvement of the deeper cuts is not significant* (*see Figure 2.6* (*top*)); *(iii) if the first-layer CART is developed enough, then the second-layer CART acts as a variance reducer* (*see Figure 2.6* (*middle*)).

2.4.3 Main Results

Building on Lemma 2.4.2 and 2.4.3, we now focus on a shallow network whose second-layer tree is of depth one, and whose first cut is performed along the new feature $\bar{Y}_{L_n(X)}$ at $1/2$. Two main regimes of training can be therefore identified when the first tree is either shallow ($k < k^\star$) or deep ($k \geqslant k^\star$).



(a) Depth 4 (b) Depth 6 (c) Depth 8

Figure 2.9: Chessboard data distribution for $k^\star = 6$ and $N_{\mathcal{B}} = 2^{k^\star - 1}$. Partition of the (first) encoding tree of depth 4, 6, 8 (from left to right) is displayed in blue. The optimal depth of a single centered tree for this chessboard distribution is 6.

In the first regime ($k < k^\star$), to establish precise non-asymptotics bounds, we study the balanced chessboard distribution (see Figure 2.9). Such a distribution has been studied in the unsupervised literature, in order to generate distribution for X via copula theory (Ghosh et al. 2002; Ghosh et al. 2009) or has been mixed with other distribution in the RF framework (Biau et al. 2008). Intuitively, this is a worst-case configuration for centered trees in terms of bias. Indeed, if $k < k^\star$, each leaf contains the same number of black and white cells. Therefore in expectation the mean value of the leaf is $1/2$ which is non informative.

Proposition 2.4.5 (Risk of a single tree and a shallow tree network when $k < k^\star$)**.** *Assume that the data is drawn according to a balanced chessboard distribution with parameters $k^\star$, $N_{\mathcal{B}} = 2^{k^\star - 1}$ and $p > 1/2$ (see Figure 2.9).*

1. *Consider a single tree $f_{k,0,n}$ of depth $k \in \mathbb{N}^\star$. We have,*

$$\mathcal{R}(f_{k,0,n}) \leqslant \left(p - \frac{1}{2}\right)^2 + \frac{2^k}{2(n+1)} + \frac{(1-2^{-k})^n}{4};$$

and

$$\mathcal{R}(f_{k,0,n}) \geqslant \left(p - \frac{1}{2}\right)^2 + \frac{2^k}{4(n+1)} + \frac{(1-2^{-k})^n}{4}\left(1 - \frac{2^k}{n+1}\right).$$

2. *Consider the shallow tree network $f_{k,1,n}$. We have*

$$R(f_{k,1,n}) \leqslant \left(p - \frac{1}{2}\right)^2 + \frac{2^{k/2+3}(p - \frac{1}{2})}{\sqrt{\pi n}} + \frac{7 \cdot 2^{2k+2}}{\pi^2(n+1)}(1 + \varepsilon_{k,p}) + \frac{p^2 + (1-p)^2}{2}\left(1 - 2^{-k}\right)^n$$

where $\varepsilon_{k,p} = o(2^{-k/2})$ uniformly in p, and

$$R(f_{k,1,n}) \geqslant \left(p - \frac{1}{2}\right)^2.$$

First, note that our bounds are tight in both cases ($k < k^\star$ and $k \geqslant k^\star$) since the rates of the upper bounds match that of the lower ones. The first statement in Proposition 2.4.5 quantifies the bias of a single tree of depth $k < k^\star$: the term $(p - 1/2)^2$ appears in both the lower and upper bounds, which means that no matter how large the training set is, the risk of the tree does not tend to zero. The shallow tree network suffers from the same bias term as soon as the first-layer tree is not deep enough. Here, the flaws of the first-layer tree transfer to the whole network. In

all bounds, the term $(1-2^{-k})^n$ corresponding to the probability of X falling into an empty cell is classic and cannot be eliminated for centered trees, whose splitting strategy is independent of the dataset.

Proposition S1 in the Supplementary Materials extends the previous result to the case of a random chessboard, in which each cell has a probability of being black or white. The same phenomenon is observed: the bias of the first layer tree is not reduced, even in the infinite sample regime.

In the second regime ($k \geqslant k^\star$), the tree network may improve over a single tree as shown in Proposition 2.4.6.

Proposition 2.4.6 (Risk of a single tree and a shallow tree network when $k \geqslant k^\star$)**.** *Consider a generalized chessboard with parameters $k^\star$, $N_{\mathcal{B}}$ and $p > 1/2$.*

1. *Consider a single tree $f_{k,0,n}$ of depth $k \in \mathbb{N}^\star$. We have*

$$\mathcal{R}(f_{k,0,n}) \leqslant \frac{2^k p(1-p)}{n+1} + \left(p^2 + (1-p)^2\right) \frac{(1-2^{-k})^n}{2},$$

and

$$\mathcal{R}(f_{k,0,n}) \geqslant \frac{2^{k-1} p(1-p)}{n+1} + \left(p^2 + (1-p)^2 - \frac{2^k p(1-p)}{n+1}\right) \frac{(1-2^{-k})^n}{2}.$$

2. *Consider the shallow tree network $f_{k,1,n}$. Letting*

$$\bar{p}_{\mathcal{B}}^2 = \left(\frac{N_{\mathcal{B}}}{2^{k^\star}} p^2 + \frac{2^{k^\star} - N_{\mathcal{B}}}{2^{k^\star}} (1-p)^2\right) (1-2^{-k})^n,$$

we have

$$\mathcal{R}(f_{k,1,n}) \leqslant 2 \cdot \frac{p(1-p)}{n+1} + \frac{2^{k+2} \varepsilon_{n,k,p}}{n} + \bar{p}_{\mathcal{B}}^2,$$

where $\varepsilon_{n,k,p} = n(1 - \frac{1-e^{-2(p-\frac{1}{2})^2}}{2^k})^n$, and for all $n \geqslant 2^{k+1}(k+1)$,

$$R(f_{k,1,n}) \geqslant \frac{2p(1-p)}{n} - \frac{2^{k+3}(1-\rho_{k,p})^n}{n} + \bar{p}_{\mathcal{B}}^2,$$

where $0 < \rho_{k,p} < 1$ depends only on p and k.

Proposition 2.4.6 shows that there exists a benefit from using this network when the first-layer tree is deep enough. In this case, the risk of the shallow tree network is $O(1/n)$ whereas that of a single tree is $O(2^k/n)$. In presence of complex and highly structured data (large $k^\star$ and similar distribution in different areas of the input space), the shallow tree network benefits from a variance reduction phenomenon by a factor 2^k. These theoretical bounds are numerically assessed in the Supplementary Materials (see Figures S35 to S40) showing their tightness for a particular choice of the chessboard configuration.

Finally, note that although the dimension d does not explicitly appear in our bounds, it is closely related to $k^\star$. Indeed, in high dimensions, modelling the regression function requires a finer partition, hence a direct relation of the form $k^\star \gg d$. Therefore, obtaining an unbiased estimator with a reduced variance as in Proposition 2.4.5 is more stringent in high dimensions, since it requires to choose $k \geqslant k^\star \gg d$.

2.5 Conclusion

In this paper, we study both numerically and theoretically DF and its elementary components. We show that stacking layers of trees (and forests) may improve the predictive performance of the algorithm. However, based on the empirical study, it seems that most of the improvements rely on the first DF-layers. We show that the performance of a shallow tree network (composed of single CART) depends on the depth of the first-layer tree. When the first-layer tree is deep enough, the second-layer tree may build upon the new features created by the first tree by acting as a variance reducer.

To quantify this phenomenon, we propose a first theoretical analysis of a shallow tree network (composed of centered trees). Our study exhibits the crucial role of the first (encoding) layer: if the first-layer tree is biased, then the entire shallow network inherits this bias, otherwise the second-layer tree acts as a good variance reducer. One should note that this variance reduction cannot be obtained by averaging many trees, as in RF structure: the variance of an averaging of centered trees with depth k is of the same order as one of these individual trees (Biau 2012a; Klusowski 2018), whereas two trees in cascade (the first one of depth k and the second of depth 1) may lead to a variance reduction by a 2^k factor. This highlights the benefit of tree-layer architectures over standard ensemble methods. We thus believe that this first theoretical study of this shallow tree network paves the way of the mathematical understanding of DF.

First-layer trees, and more generally the first layers in DF architecture, can be seen as data-driven encoders. More precisely, the first layers in DF create an automatic embedding of the data, building on the specific conditional relation between the output and the inputs, therefore potentially improving the performance of the overall structure. Since preprocessing is nowadays an important part of all machine learning pipelines, we believe that our analysis is interesting beyond the framework of DF.

S1 Additional Figures

S1.1 Computation Times for Section 2.3

	Yeast	Housing	Letter	Adult	Airbnb	Higgs
Default DF time	13m19s	9m38s	20m31	13m57s	23m23s	43m53s
Light DF time	7s	6s	8s	8s	10s	13s
Default DF MC (MB)	11	6	174	139	166	531
Light DF MC (MB)	5	4	109	72	100	318

Table S2: Comparing the time and memory consumption of DF and Light DF.

S1.2 Table of Best Configurations, Supplementary to Section 2.3.2

Dataset	Best configuration hyperparam.	Optimal sub-model
Adult	6 forests, 20 trees, max depth 30	2
Higgs	10 forests, 280 trees, max depth None	2
Fashion Mnist	8 forests, 500 trees, max depth None (default)	2
Letter	8 forests, 500 trees, max depth None (default)	1
Yeast	6 forests, 200 trees, max depth 30	1
Airbnb	10 forests, 400 trees, max depth None	1
Housing	8 forests, 280 trees, max depth 100	14

Table S3: Details of the best configurations obtained in Figures 2.2 and 2.3.

To find the best configuration, we ran a grid search over the following parameters : number of forests per layer (from 2 to 10) , number of trees per forest (from 30 to 1000), max depth of each tree (from 5 to 100 plus None).

S1.3 Fashion Mnist MGS Encoding

The Fashion Mnist dataset was encoded using the MGS process with two forests, one Breiman RF, one CRF, both of them having 150 trees, 10 samples per leaf minimum and other parameters set to default. Three windows were used of sizes/strides. Then we apply a mean pooling process of size (3,3) to each created filter.

S1.4 Additional Figures to Section 2.3.3

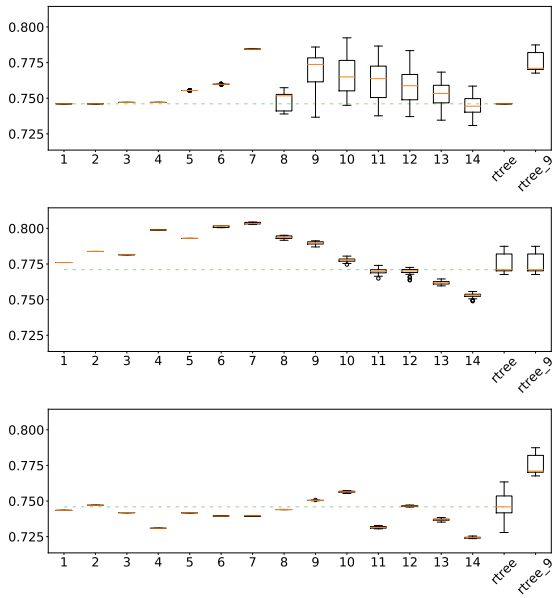


Figure S10: Adult dataset. Accuracy of a two-layer tree architecture w.r.t. the second-layer tree depth, when the first-layer (encoding) tree is of depth 2 (top), 9 (middle), and 15 (bottom). `rtree` is a single tree of respective depth 2 (top), 9 (middle), and 15 (bottom), applied on raw data. For this dataset, the optimal depth of a single tree is 9 and the tree with the optimal depth is depicted as `rtree_9` in each plot. The green dashed line indicates the median score of the `rtree`. All boxplots are obtained by 10 different runs.

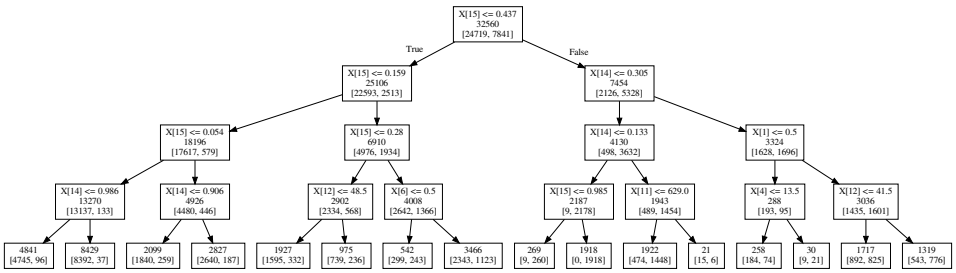


Figure S11: Adult dataset. Second-layer tree structure of depth 4 when the first-layer tree is of depth 9 (optimal depth). Raw features range from X[0] to X[13], X[14] and X[15] are the features built by the first-layer tree.

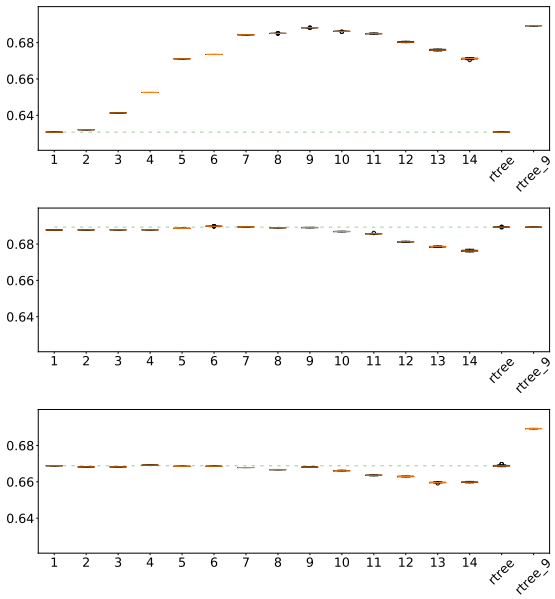


Figure S12: Higgs dataset. Accuracy of a two-layer tree architecture w.r.t. the second-layer tree depth, when the first-layer (encoding) tree is of depth 2 (top), 9 (middle), and 15 (bottom). `rtree` is a single tree of respective depth 2 (top), 9 (middle), and 15 (bottom), applied on raw data. For this dataset, the optimal depth of a single tree is 9 and the tree with the optimal depth is depicted as `rtree_9` in each plot. The green dashed line indicates the median score of the `rtree`. All boxplots are obtained by 10 different runs.

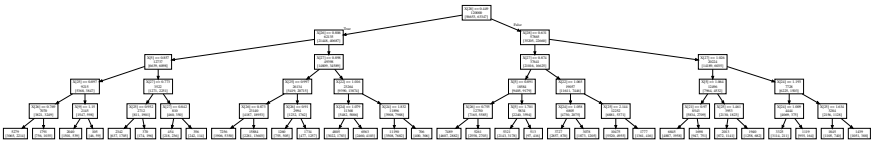


Figure S13: Higgs dataset. Second-layer tree structure of depth 5 when the first-layer tree is of depth 2 (low depth). Raw features range from X[0] to X[13], X[14] and X[15] are the features built by the first-layer tree.

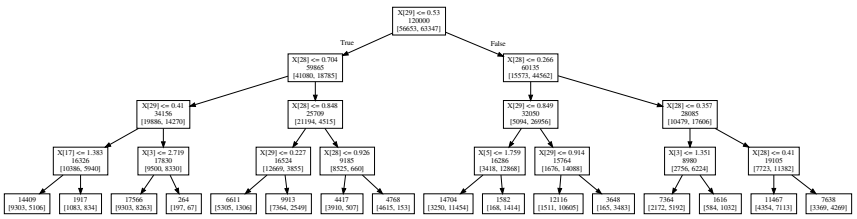


Figure S14: Higgs dataset. Second-layer tree structure of depth 4 when the first-layer tree is of depth 9 (optimal depth). Raw features range from X[0] to X[27], X[28] and X[29] are the features built by the first-layer tree.

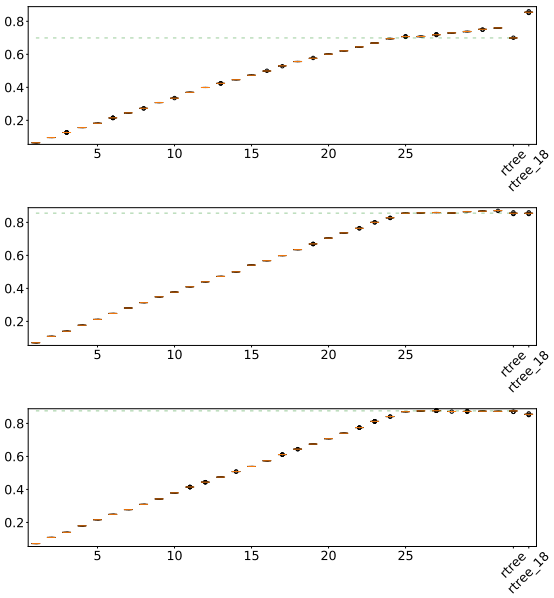


Figure S15: Letter dataset. Accuracy of a two-layer tree architecture w.r.t. the second-layer tree depth, when the first-layer (encoding) tree is of depth 10 (top), 18 (middle), and 26 (bottom). `rtree` is a single tree of respective depth 10 (top), 18 (middle), and 26 (bottom), applied on raw data. For this dataset, the optimal depth of a single tree is 18 and the tree with the optimal depth is depicted as `rtree_18` in each plot. The green dashed line indicates the median score of the `rtree`. All boxplots are obtained by 10 different runs.

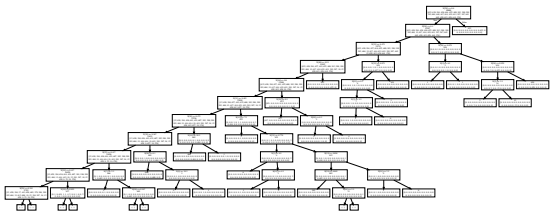


Figure S16: Letter dataset. Second-layer tree structure of depth 30 when the first-layer tree is of depth 18 (optimal depth). We only show the first part of the tree up to depth 10. Raw features range from X[0] to X[15]. The features built by the first-layer tree range from X[16] to X[41].

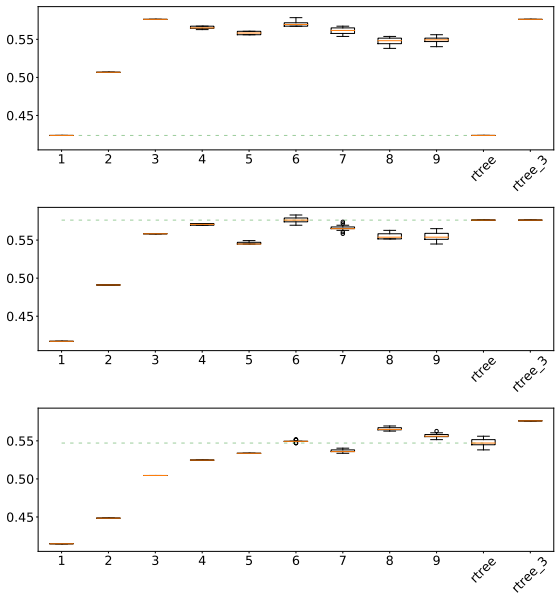


Figure S17: Yeast dataset. Accuracy of a two-layer tree architecture w.r.t. the second-layer tree depth, when the first-layer (encoding) tree is of depth 1 (top), 3 (middle), and 8 (bottom). `rtree` is a single tree of respective depth 1 (top), 3 (middle), and 8 (bottom), applied on raw data. For this dataset, the optimal depth of a single tree is 3 and the tree with the optimal depth is depicted as `rtree_3` in each plot. The green dashed line indicates the median score of the `rtree`. All boxplots are obtained by 10 different runs.

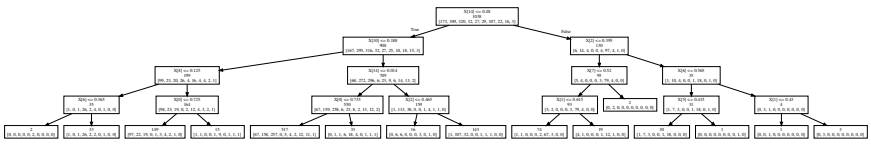


Figure S18: Yeast dataset. Second-layer tree structure of depth 4 when the first-layer tree is of depth 3 (optimal depth). Raw features range from X[0] to X[7]. The features built by the first-layer tree range from X[8] to X[17].

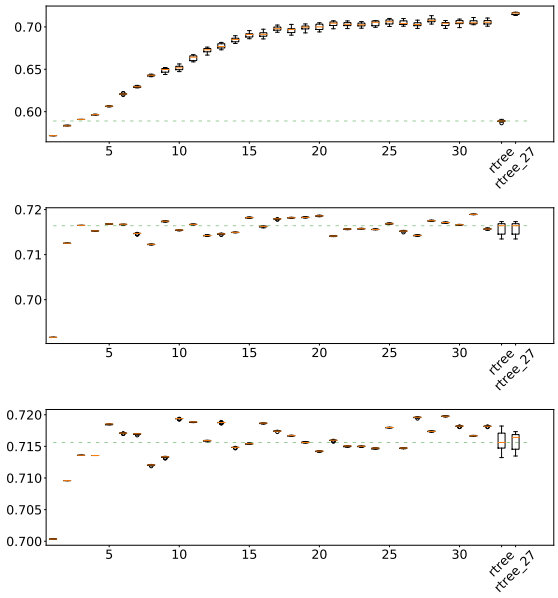


Figure S19: Airbnb dataset. R^2 score of a two-layer tree architecture w.r.t. the second-layer tree depth, when the first-layer (encoding) tree is of depth 10 (top), 27 (middle), and 32 (bottom). `rtree` is a single tree of respective depth 10 (top), 27 (middle), and 32 (bottom), applied on raw data. For this dataset, the optimal depth of a single tree is 27 and the tree with the optimal depth is depicted as `rtree_27` in each plot. The green dashed line indicates the median score of the `rtree`. All boxplots are obtained by 10 different runs.

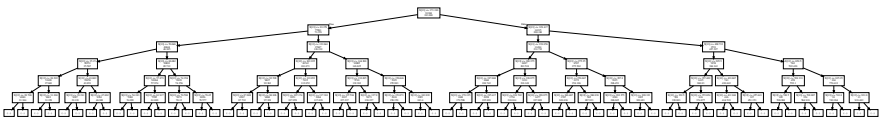


Figure S20: Airbnb dataset. Second-layer tree structure of depth 28 when the first-layer tree is of depth 26 (optimal depth). We only show the first part of the tree up to depth 5. Raw features range from X[0] to X[12], X[13] is the feature built by the first-layer tree.

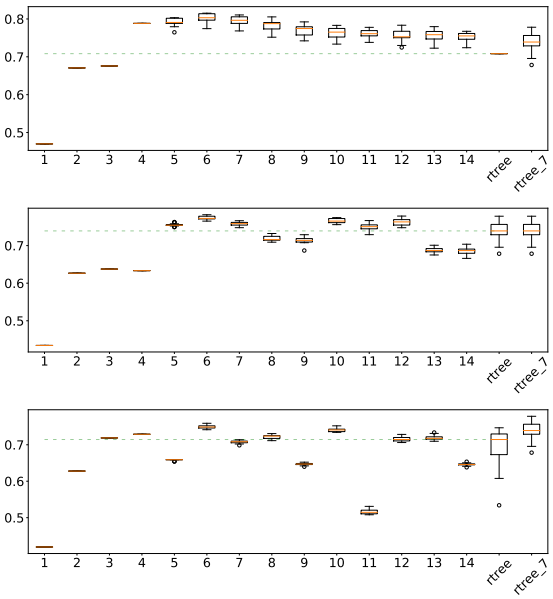


Figure S21: Housing dataset. R^2 score of a two-layer tree architecture w.r.t. the second-layer tree depth, when the first-layer (encoding) tree is of depth 3 (top), 7 (middle), and 12 (bottom). `rtree` is a single tree of respective depth 3 (top), 7 (middle), and 12 (bottom), applied on raw data. For this dataset, the optimal depth of a single tree is 9 and the tree with the optimal depth is depicted as `rtree_7` in each plot. The green dashed line indicates the median score of the `rtree`. All boxplots are obtained by 10 different runs.

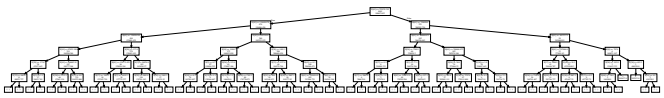


Figure S22: Housing dataset. Second-layer tree structure of depth 10 when the first-layer tree is of depth 7 (optimal depth). We only show the first part of the tree up to depth 5. Raw features range from X[0] to X[60], X[61] is the feature built by the first-layer tree.

S1.5 Additional Figures to Section 2.3.2

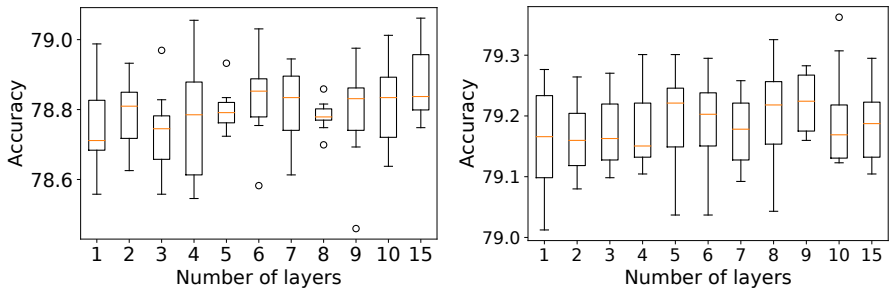


Figure S23: Adult dataset. Boxplots over 10 tries of the accuracy of a DF with 1 forest by layer (left) or 4 forests by layer (right), with respect to the DF maximal number of layers.

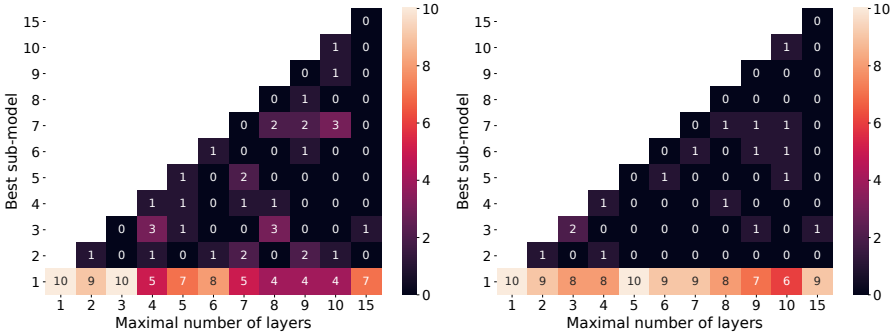


Figure S24: Adult dataset. Heatmap counting the index of the sub-optimal model over 10 tries of a default DF with 1 (Breiman) forest per layer (left) or 4 forests (2 Breiman, 2 CRF) per layer (right), with respect to the maximal number of layers.

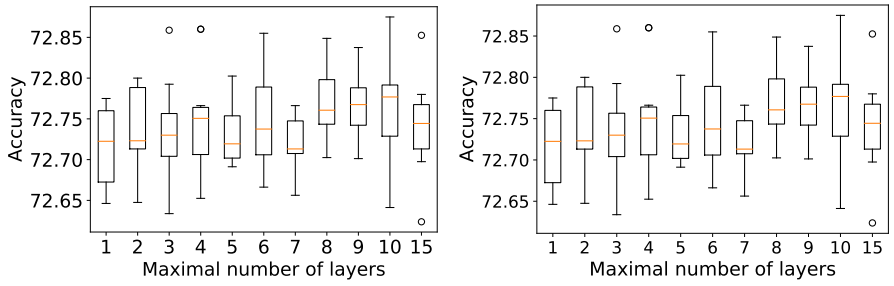


Figure S25: Higgs dataset. Boxplots over 10 tries of the accuracy of a DF with 1 forest by layer (left) or 4 forests by layer (right), with respect to the DF maximal number of layers.

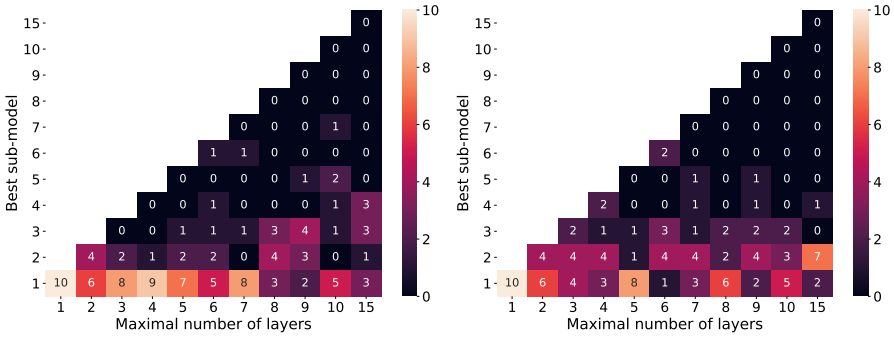


Figure S26: Higgs dataset. Heatmap counting the index of the sub-optimal model over 10 tries of a default DF with 1 (Breiman) forest per layer (left) or 4 forests (2 Breiman, 2 CRF) per layer (right), with respect to the maximal number of layers.

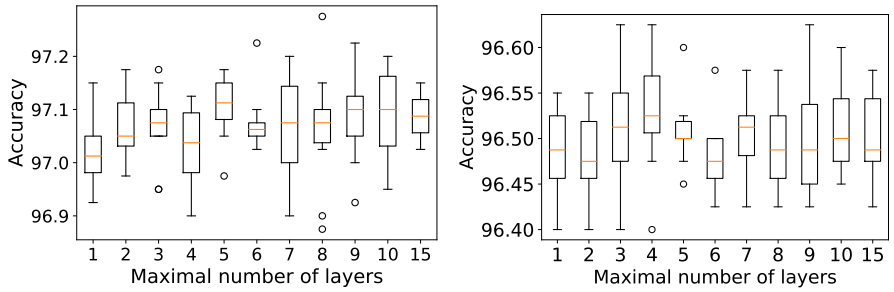


Figure S27: Letter dataset. Boxplots over 10 tries of the accuracy of a DF with 1 forest by layer (left) or 4 forests by layer (right), with respect to the DF maximal number of layers.

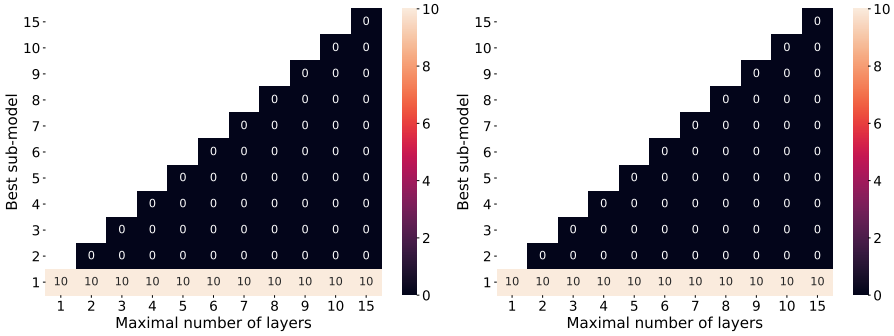


Figure S28: Letter dataset. Heatmap counting the index of the sub-optimal model over 10 tries of a default DF with 1 (Breiman) forest per layer (left) or 4 forests (2 Breiman, 2 CRF) per layer (right), with respect to the maximal number of layers.

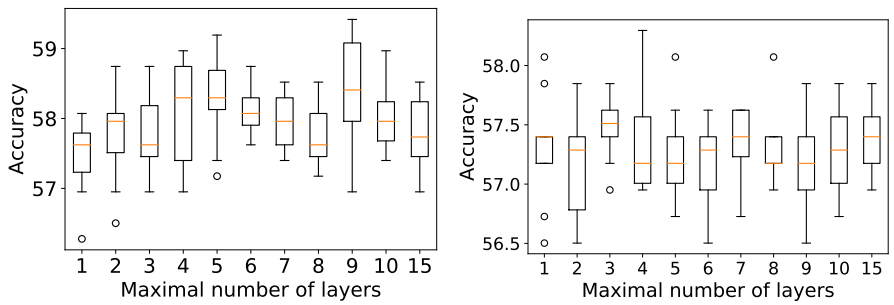


Figure S29: Yeast dataset. Boxplots over 10 tries of the accuracy of a DF with 1 forest by layer (left) or 4 forests by layer (right), with respect to the DF maximal number of layers.

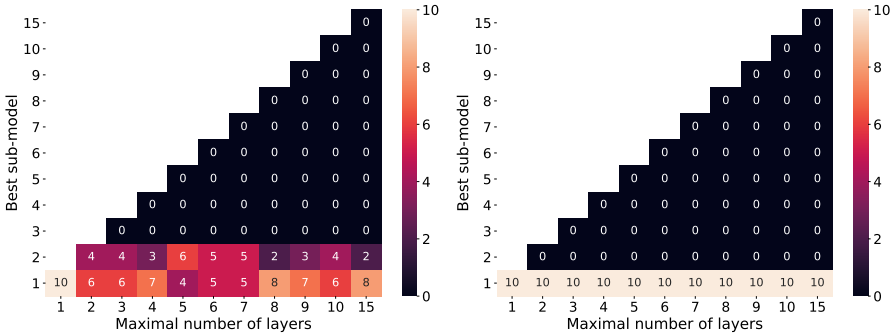


Figure S30: Yeast dataset. Heatmap counting the index of the sub-optimal model over 10 tries of a default DF with 1 (Breiman) forest per layer (left) or 4 forests (2 Breiman, 2 CRF) per layer (right), with respect to the maximal number of layers.

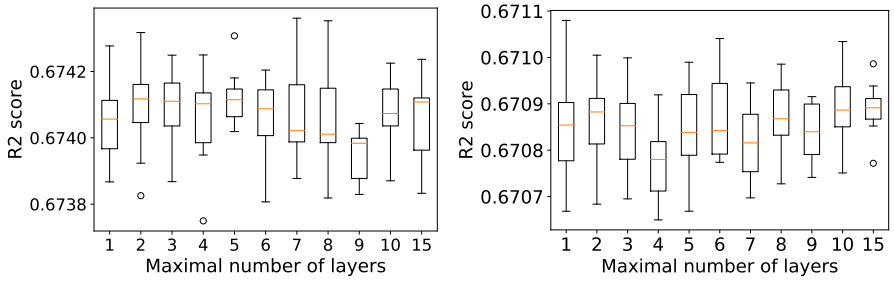


Figure S31: Airbnb dataset. Boxplots over 10 tries of the accuracy of a DF with 1 forest by layer (left) or 4 forests by layer (right), with respect to the DF maximal number of layers.

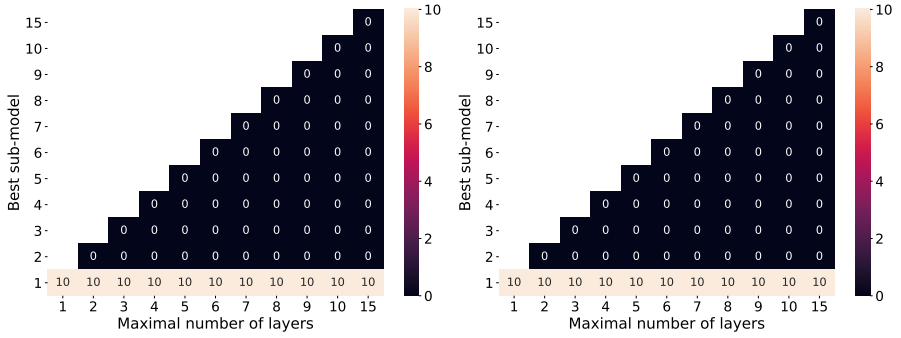


Figure S32: Airbnb datase. Heatmap counting the index of the sub-optimal model over 10 tries of a default DF with 1 (Breiman) forest per layer (left) or 4 forests (2 Breiman, 2 CRF) per layer (right), with respect to the maximal number of layers.

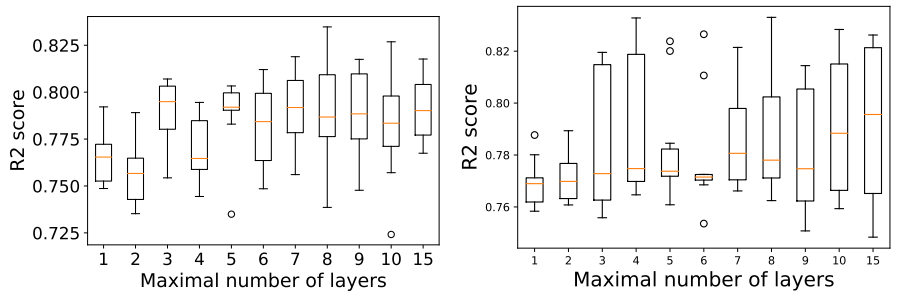


Figure S33: Housing dataset. Boxplots over 10 tries of the accuracy of a DF with 1 forest by layer (left) or 4 forests by layer (right), with respect to the DF maximal number of layers.

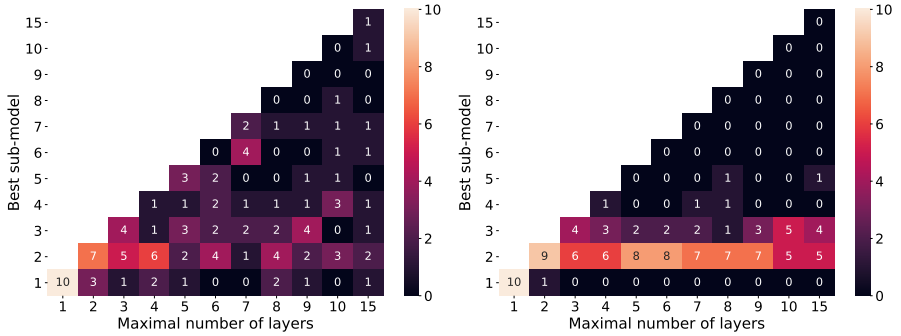


Figure S34: Housing dataset. Heatmap counting the index of the sub-optimal model over 10 tries of a default DF with 1 (Breiman) forest per layer (left) or 4 forests (2 Breiman, 2 CRF) per layer (right), with respect to the maximal number of layers.

S1.6 Additional Figures to Section 2.4

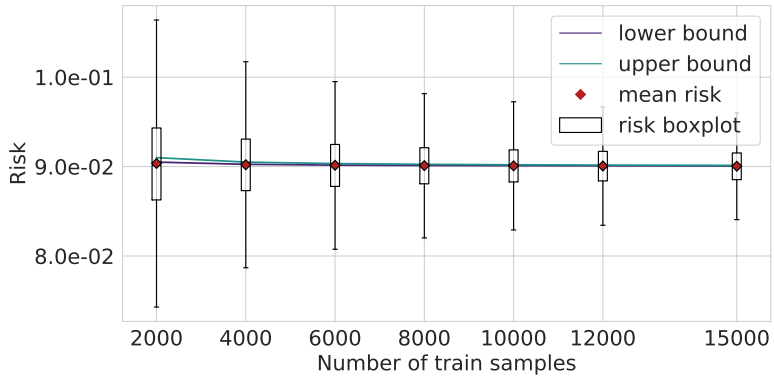


Figure S35: Illustration of the theoretical bounds for a single tree of Proposition 2.4.5 1. for a chessboard with parameters $k^\star = 4$, $N_{\mathcal{B}} = 2^{k^\star - 1}$, and $p = 0.8$. The single tree is of depth $k = 2$. We draw a sample of size n (x-axis), and a single tree $f_{k,0,n}$ is fitted for which the theoretical risk is evaluated. Each boxplot is built out of 20 000 repetitions. The outliers are not shown for the sake of presentation.

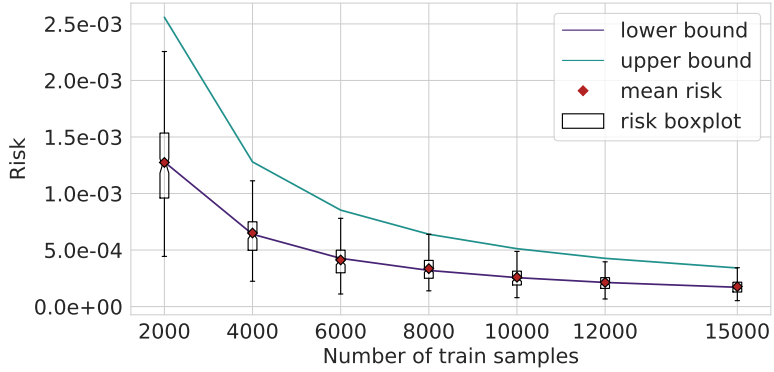


Figure S36: Illustration of the theoretical bounds for a single tree of Proposition 2.4.6 1. for a chessboard with parameters $k^\star = 4$, $N_{\mathcal{B}} = 2^{k^\star - 1}$ and $p = 0.8$. The single tree is of depth $k = 4$. We draw a sample of size n (x-axis), and a single tree $f_{k,0,n}$ is fitted for which the theoretical risk is evaluated. Each boxplot is built out of 20 000 repetitions. The outliers are not shown for the sake of presentation.

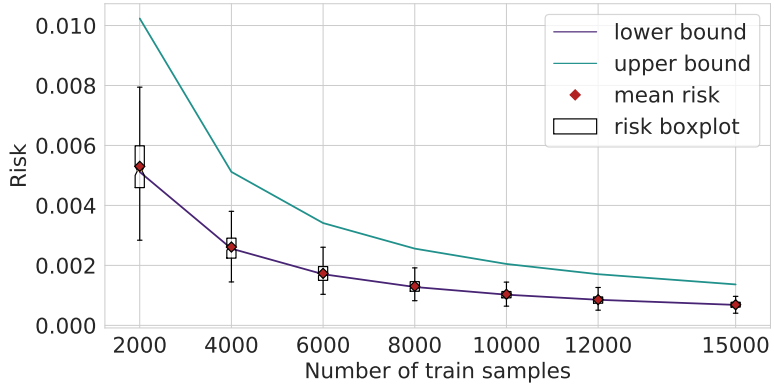


Figure S37: Illustration of the theoretical bounds for a single tree of Proposition 2.4.6 1. for a chessboard with parameters $k^\star = 4$, $N_{\mathcal{B}} = 2^{k^\star - 1}$ and $p = 0.8$. The single tree is of depth $k = 6$. We draw a sample of size n (x-axis), and a single tree $f_{k,0,n}$ is fitted for which the theoretical risk is evaluated. Each boxplot is built out of 20 000 repetitions. The outliers are not shown for the sake of presentation.

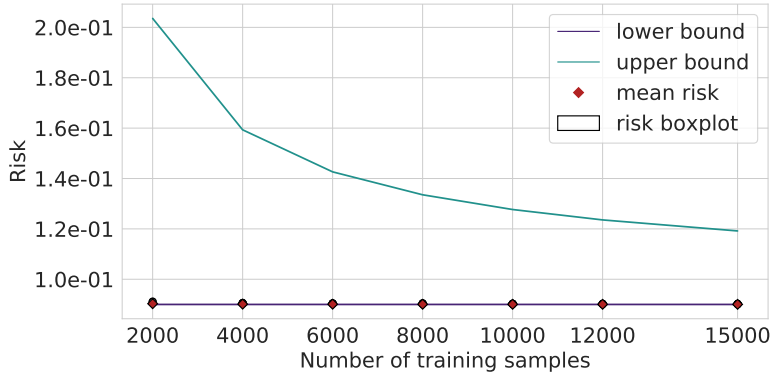


Figure S38: Illustration of the theoretical bounds for a shallow tree network of Proposition 2.4.5 2. for a chessboard with parameters $k^\star = 4$, $N_{\mathcal{B}} = 2^{k^\star - 1}$ and $p = 0.8$. The first-layer tree is of depth $k = 2$. We draw a sample of size n (x-axis), and a single tree $f_{k,0,n}$ is fitted for which the theoretical risk is evaluated. Each boxplot is built out of 20 000 repetitions. The outliers are not shown for the sake of presentation. Note that in such a case, the theoretical lower bound is constant and equal to the bias term.

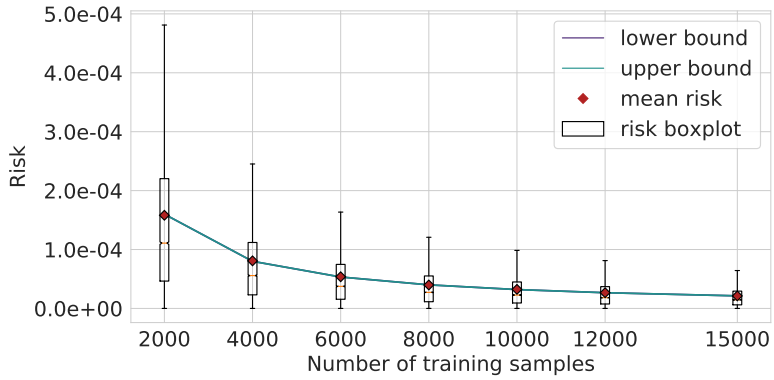


Figure S39: Illustration of the theoretical bounds for a shallow tree network of Proposition 2.4.6 2. for a chessboard with parameters $k^\star = 4$, $N_{\mathcal{B}} = 2^{k^\star - 1}$ and $p = 0.8$. The first-layer tree is of depth $k = 4$. We draw a sample of size n (x-axis), and a single tree $f_{k,0,n}$ is fitted for which the theoretical risk is evaluated. Each boxplot is built out of 20 000 repetitions. The outliers are not shown for the sake of presentation. Note that in such a case, the theoretical lower bound is constant and equal to the bias term. Note that the lower bound and the upper bound are merged.

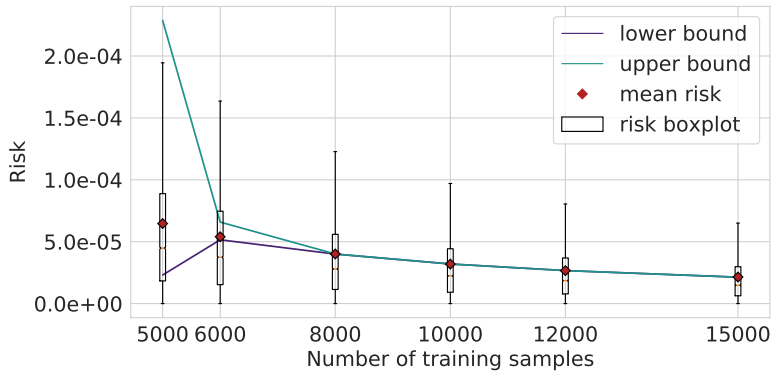


Figure S40: Illustration of the theoretical bounds for a shallow tree network of Proposition 2.4.6 2. for a chessboard with parameters $k^\star = 4$, $N_{\mathcal{B}} = 2^{k^\star - 1}$ and $p = 0.8$. The first-layer tree is of depth $k = 6$. We draw a sample of size n (x-axis), and a single tree $f_{k,0,n}$ is fitted for which the theoretical risk is evaluated. Each boxplot is built out of 20 000 repetitions. The outliers are not shown for the sake of presentation. Note that in such a case, the theoretical lower bound is constant and equal to the bias term.

S2 Technical Results on Binomial Random Variables

Lemma S1. *Let Z be a binomial $\mathfrak{B}(n,p)$, $p \in (0,1]$, $n > 0$. Then,*

1. $$\frac{1-(1-p)^n}{(n+1)p} \leqslant \mathbb{E}\left[\frac{\mathbb{1}_{Z>0}}{Z}\right] \leqslant \frac{2}{(n+1)p}$$
2. $$\mathbb{E}\left[\frac{1}{1+Z}\right] \leqslant \frac{1}{(n+1)p}$$
3. $$\mathbb{E}\left[\frac{1}{1+Z^2}\right] \leqslant \frac{3}{(n+1)(n+2)p^2}$$
4. $$\mathbb{E}\left[\frac{\mathbb{1}_{Z>0}}{\sqrt{Z}}\right] \leqslant \frac{2}{\sqrt{np}}$$

5. *Let k be an integer $\leqslant n$. Then,*
$$\mathbb{E}\left[Z \mid Z \geqslant k\right] = np + (1-p)k\frac{\mathbb{P}\left(Z=k\right)}{\sum_{i=k}^{n}\mathbb{P}\left(Z=i\right)}.$$

6. *Let Z be a binomial $\mathfrak{B}(n,\frac{1}{2})$, $n > 0$. Then,*
$$\mathbb{E}\left[Z \mid Z \leqslant \lfloor\frac{n+1}{2}\rfloor - 1\right] \geqslant \frac{n}{2} - \left(\frac{\sqrt{n}}{\sqrt{\pi}} + \frac{2\sqrt{2n}}{\pi\sqrt{2n+1}}\right).$$

7. *Let Z be a binomial $\mathfrak{B}(n,\frac{1}{2})$, $n > 0$. Then,*
$$\mathbb{E}\left[Z \mid Z \geqslant \lfloor\frac{n+1}{2}\rfloor\right] \leqslant \frac{n}{2} + 1 + \frac{1}{\sqrt{\pi(n+1)}}.$$

Proof. The reader may refer to the Lemma 11 of Biau 2012a to see the proof of 2., 3. and the right-hand side of 1. The left-hand side inequality of 1 can be found in the Section 1 of Cribari-Neto et al. 2000.

4. The first two inequalities rely on simple analysis :

$$\begin{aligned}\mathbb{E}\left[\frac{\mathbb{1}_{Z>0}}{\sqrt{Z}}\right] &\leqslant \mathbb{E}\left[\frac{2}{1+\sqrt{Z}}\right] \\ &\leqslant \mathbb{E}\left[\frac{2}{\sqrt{1+Z}}\right].\end{aligned}$$

To go on, we adapt a transformation from Section 2 of Cribari-Neto et al. 2000 to our setting:

$$\begin{aligned}\mathbb{E}\left[\frac{2}{\sqrt{1+Z}}\right] &= \frac{2}{\Gamma(1/2)}\int_0^\infty \frac{e^{-t}}{\sqrt{t}}\mathbb{E}\left[e^{-tZ}\right]dt \\ &= \frac{2}{\Gamma(1/2)}\int_0^\infty \frac{e^{-t}}{\sqrt{t}}(1-p+pe^{-t})^n dt \\ &= \frac{2}{\Gamma(1/2)}\int_0^{-\log(1-p)} g(r)e^{-rn}dr,\end{aligned}$$

with $g(r) := p^{-1}e^{-r}\left(-\log(1+\frac{1-e^{-r}}{p})\right)^{-1/2}$ after the change of variable $(1-p+pe^{-t}) = e^{-r}$.

Let's prove that

$$g(r) \leqslant \frac{1}{\sqrt{rp}}. \tag{2.2}$$

It holds that $\log(1+x) \leqslant \frac{2x}{2+x}$ when $-1 < x \leqslant 0$, therefore

$$g(r)^2 = p^{-2}e^{-2r}\left(-\log(1+\frac{1-e^{-r}}{p})\right)^{-1} \leqslant p^{-2}e^{-2r}\frac{2p+e^{-r}-1}{2(1-e^{-r})}.$$

Furthermore,

$$\begin{aligned}2p &\geqslant 2p\left(e^{-r}+re^{-2r}\right) \\ &\geqslant 2p\left(e^{-r}+re^{-2r}\right) + r\left(e^{-3r}-e^{-2r}\right) \\ &= re^{-2r}(2p-1+e^{-r}) + 2pe^{-r},\end{aligned}$$

and then dividing by rp^2,

$$\frac{2}{rp}(1-e^{-r}) \geqslant \frac{1}{p^2}e^{-2r}(2p-1+e^{-r}) \qquad \Longleftrightarrow \qquad \frac{1}{rp} \geqslant p^{-2}e^{-2r}\frac{2p+e^{-r}-1}{2(1-e^{-r})},$$

which proves (2.2).

Equation (2.2) leads to

$$\mathbb{E}\left[\frac{2}{\sqrt{1+Z}}\right] \leqslant \frac{2}{\Gamma(1/2)}\int_0^{-\log(1-p)} \frac{1}{\sqrt{pr}}e^{-rn}dr. \tag{2.3}$$

Note that $\Gamma(1/2) = \sqrt{\pi}$. After the change of variable $u = \sqrt{rn}$, we obtain :

$$\mathbb{E}\left[\frac{2}{\sqrt{1+Z}}\right] \leqslant \frac{4}{\sqrt{np\pi}}\int_0^{\sqrt{-n\log(1-p)}} e^{-u^2}du \leqslant \frac{4}{\sqrt{np\pi}}\int_0^\infty e^{-u^2}du \leqslant \frac{2}{\sqrt{np}}$$

which ends the proof of (iv).

5.(a) We recall that $p = 1/2$. An explicit computation of the expectation yields :

$$\begin{aligned}
\mathbb{E}\left[Z \mid Z < \lfloor \frac{n+1}{2} \rfloor\right] &= \frac{1}{\mathbb{P}\left(Z \leqslant \lfloor \frac{n+1}{2} \rfloor - 1\right)} \sum_{i=1}^{\lfloor \frac{n+1}{2} \rfloor - 1} \frac{i}{2^n} \binom{n}{i} \\
&= \frac{2}{1} \frac{n}{2^n} \left(\frac{2^n}{2} - \frac{1}{2} \binom{n-1}{\frac{n-1}{2}} \right) \mathbb{1}_{n\%2=1} \\
&+ \frac{n}{\frac{1}{2} - \frac{1}{2}\mathbb{P}\left(Z = n/2\right)} \left(\sum_{i=1}^{n/2} i \binom{n}{i} - \frac{n}{2} \binom{n}{n/2} \right) \frac{\mathbb{1}_{n\%2=0}}{2^n} \\
&= n \left(\frac{1}{2} - \frac{1}{2^n} \binom{n-1}{\frac{n-1}{2}} \right) \mathbb{1}_{n\%2=1} + \frac{n \cdot \mathbb{1}_{n\%2=0}}{1 - \mathbb{P}\left(Z = n/2\right)} \left(\frac{1}{2} - \frac{1}{2^n} \binom{n}{n/2} \right).
\end{aligned}$$

We use that for all $m \in 2\mathbb{N}^*$,

$$\binom{m}{m/2} \leqslant \frac{2^m}{\sqrt{\pi(m/2 + 1/4)}} \tag{2.4}$$

and

$$\frac{1}{1 - \mathbb{P}\left(Z = m/2\right)} \geqslant 1 + \frac{\sqrt{2}}{\sqrt{\pi n}}$$

where the last inequality can be obtained via a series expansion at $n = \infty$. Replacing the terms by their bounds, we have :

$$\begin{aligned}
\mathbb{E}\left[Z \mid Z < \lfloor \frac{n+1}{2} \rfloor\right] &\geqslant n \left(\left(\frac{1}{2} - \frac{1}{\sqrt{\pi(2m-1)}} \right) \mathbb{1}_{n\%2=1} \right. \\
&\left. + \left(1 + \frac{\sqrt{2}}{\sqrt{\pi n}} \right) \left(\frac{1}{2} - \frac{2}{\sqrt{\pi(2n+1)}} \right) \mathbb{1}_{n\%2=0} \right) \\
&\geqslant n \left(\frac{1}{2} - \frac{1}{\sqrt{n\pi}} - \frac{2\sqrt{2}}{\pi\sqrt{n(2n+1)}} \right) \\
&\geqslant \frac{n}{2} + \sqrt{n} \left(\frac{1}{\sqrt{\pi}} - \frac{2\sqrt{2}}{\pi} \sqrt{(2n+1)} \right)
\end{aligned}$$

which ends the proof of this item (v)(a).

5.(b) We also begin with an explicit computation of the expectation :

$$\begin{aligned}\mathbb{E}\left[Z \mid Z \geqslant \lfloor\frac{n+1}{2}\rfloor\right] &= \frac{1}{\mathbb{P}\left(Z \geqslant \lfloor\frac{n+1}{2}\rfloor\right)} \sum_{i=\lfloor\frac{n+1}{2}\rfloor}^{n} \frac{i}{2^n}\binom{n}{i} \\ &= \frac{2}{1}\frac{1}{2^n}\left(2^{n-2} + 2^{n-1} + \frac{1}{2}\binom{n-1}{\frac{n-1}{2}}\right)\mathbb{1}_{n\%2=1} \\ &+ \frac{n}{\frac{1}{2}+\frac{1}{2}\mathbb{P}\left(Z=n/2\right)}\left(\sum_{i=\lfloor\frac{n+1}{2}\rfloor}^{n} i\binom{n}{i}\right)\frac{\mathbb{1}_{n\%2=0}}{2^n} \\ &= \left(\frac{n}{2}+1+\frac{1}{2^n}\binom{n-1}{\frac{n-1}{2}}\right)\mathbb{1}_{n\%2=1} + \frac{n\cdot\mathbb{1}_{n\%2=0}}{1+\mathbb{P}\left(Z=n/2\right)}\left(\frac{1}{2}+\frac{1}{2^n}\binom{n}{n/2}\right).\end{aligned}$$

The computation of the upper bound relies on the following inequalities : $\forall m \in 2\mathbb{N}^*$,

$$\binom{2m}{m} \leqslant \frac{2^{2m}}{\sqrt{\pi(m+1/4)}} \tag{2.5}$$

as well as

$$\frac{1}{1+\mathbb{P}\left(Z=n/2\right)} \leqslant 1 - \frac{\sqrt{2}}{\sqrt{\pi n}} + \frac{2}{\pi n}$$

where the last bound can be found via a series expansion at $n=\infty$. Replacing all terms by their bound and simplifying roughly gives the result. □

Lemma S2 (Uniform Bernoulli labels: risk of a single tree)**.** *Let K be a compact in $\mathbb{R}^d, d \in \mathbb{N}$. Let $X, X_1, ..., X_n, n \in N^*$ be i.i.d. random variables uniformly distributed over K, $Y, Y_1, ..., Y_n$ i.i.d Bernoulli variables of parameter $p \in [0,1]$ which can be considered as the labels of $X, X_1, ..., X_n$. We denote by $f_{0,k,n}, k \in \mathbb{N}^*$ a single tree of depth k. Then we have, for all $k \in N^*$,*

(*i*)

$$\mathbb{E}\left[(f_{0,0,n}(X) - f^\star(X))^2\right] = \frac{p(1-p)}{n} \tag{2.6}$$

(*ii*)

$$2^k \cdot \frac{p(1-p)}{n} + \left(p^2 - \frac{2^k}{n}\right)(1-2^{-k})^n \leqslant \mathbb{E}\left[(f_{0,k,n}(X) - f^\star(X))^2\right] \leqslant 2^{k+1}\cdot\frac{p(1-p)}{n} + p^2(1-2^{-k})^n \tag{2.7}$$

Proof. (i) In the case $k=0$, $f_{0,0,n}$ simply computes the mean of all the (Y_i)'s over K:

$$\mathbb{E}\left[(f_{0,0,n}(X) - f^\star(X))^2\right] = \mathbb{E}\left[\left(\frac{1}{n}\sum_i Y_i - p\right)^2\right] \tag{2.8}$$

$$= \mathbb{E}\left[\frac{1}{n^2}\sum_i (Y_i - p)^2\right] \qquad (Y_i \text{ independent}) \tag{2.9}$$

$$= \frac{p(1-p)}{n}. \tag{2.10}$$

(ii)

$$\mathbb{E}\left[(f_{0,k,n}(X) - f^\star(X))^2\right] = \mathbb{E}\left[\left(\frac{1}{N_n(L_n(X))}\sum_{X_i \in L_n(X)} Y_i - p\right)^2 \mathbb{1}_{N_n(L_n(X))>0}\right] \tag{2.11}$$

$$+ p^2 \mathbb{P}\left(N_n(L_n(X)) = 0\right)$$

$$= \mathbb{E}\left[\frac{\mathbb{1}_{N_n(L_n(X))>0}}{N_n(L_n(X))^2}\sum_{X_i \in L_n(X)} (Y_i - p)^2\right] + p^2 \mathbb{P}\left(N_n(L_n(X)) = 0\right) \tag{2.12}$$

$$= p(1-p)\mathbb{E}\left[\frac{\mathbb{1}_{N_n(L_n(X))>0}}{N_n(L_n(X))}\right] + p^2(1-2^{-k})^n \tag{2.13}$$

Noticing that $N_n(L_n(X))$ is a binomial $\mathfrak{B}(n, \frac{1}{2^k})$, we obtain the upper bound using Lemma S1 (i) :

$$\mathbb{E}\left[\frac{\mathbb{1}_{N_n(L_n(X))>0}}{N_n(L_n(X))}\right] \leqslant 2 \cdot \frac{2^k}{n} \tag{2.14}$$

the lower bound is immediately obtained by applying Lemma S1, (i):

$$\mathbb{E}\left[\frac{\mathbb{1}_{N_n(L_n(X))>0}}{N_n(L_n(X))}\right] \geqslant \frac{2^k}{n}\left(1 - (1-2^{-k})^n\right) \tag{2.15}$$

□

S3 Proof of Lemma 2.4.2

Recall that $\mathcal{B}$ (resp. $\mathcal{W}$) is the union of black (resp. white) cells of the dataset.

Note that

$$\mathbb{E}[(f_{k,1,\infty}(X) - f^\star(X))^2] = \mathbb{E}[(f_{k,1,\infty}(X) - f^\star(X))^2 \mathbb{1}_{X\in\mathcal{B}}] + \mathbb{E}[(f_{k,1,\infty}(X) - f^\star(X))^2 \mathbb{1}_{X\in\mathcal{W}}]. \tag{2.16}$$

Now, we analyze the first term in Equation (2.16). We have

$$\mathbb{E}[(f_{k,1,\infty}(X) - f^\star(X))^2 \mathbb{1}_{X\in\mathcal{B}}] = \mathbb{E}[\mathbb{E}[(f_{k,1,\infty}(X) - f^\star(X))^2 \mathbb{1}_{X\in\mathcal{B}}|N_\mathcal{B}]] \tag{2.17}$$

$$= \sum_{i,j} \mathbb{E}[\mathbb{E}[\Big(\frac{1}{N_\mathcal{B}}\sum_{i',j'\in\mathcal{B}} p_{i'j'} - p_{ij}\Big)^2 \mathbb{1}_{X\in C_{ij}\cap\mathcal{B}}|N_\mathcal{B}]]$$

$$= \sum_{i,j} \mathbb{E}\left[\mathbb{E}[\mathbb{1}_{X\in C_{ij}}\mathbb{1}_{X\in\mathcal{B}}\mathbb{1}_{N_\mathcal{B}>0}\Big(\frac{1}{N_\mathcal{B}}\sum_{i',j'\in\mathcal{B}} p_{i'j'} - p_{ij}\Big)^2 |N_\mathcal{B}]\right]$$

$$+ \sum_{i,j} \mathbb{E}\left[\mathbb{E}\left[\mathbb{1}_{X\in C_{i,j}\cap\mathcal{B}}\mathbb{1}_{N_\mathcal{B}=0} p_{i,j}^2 \mid N_\mathcal{B}\right]\right]. \tag{2.18}$$

We begin with the second term in Equation (2.18). We have, for all i, j,

$$\mathbb{E}\left[\mathbb{E}\left[\mathbb{1}_{X\in C_{i,j}\cap\mathcal{B}}\mathbb{1}_{N_\mathcal{B}=0}p_{i,j}^2 \mid N_\mathcal{B}\right]\right] = \mathbb{E}\left[\mathbb{1}_{X\in C_{i,j}}\mathbb{1}_{N_\mathcal{B}=0}\mathbb{E}\left[p_{i,j}^2\mathbb{1}_{X\in\mathcal{B}} \mid X, N_\mathcal{B}\right]\right] \tag{2.19}$$

$$= \mathbb{E}\left[\mathbb{1}_{X\in C_{i,j}}\mathbb{1}_{N_\mathcal{B}=0}\mathbb{E}\left[p_{i,j}^2\mathbb{1}_{p_{i,j}\geqslant\frac{1}{2}}\right]\right]. \tag{2.20}$$

As $p_{i,j}$ is drawn uniformly in $[0,1]$,

$$\begin{aligned}\mathbb{E}\left[p_{i,j}^2\mathbb{1}_{p_{i,j}\geqslant\frac{1}{2}}\right] &= \mathbb{E}\left[p_{i,j}^2 \mid p_{i,j}\geqslant\frac{1}{2}\right]\mathbb{P}\left(p_{i,j}\geqslant\frac{1}{2}\right)\\ &= \frac{7}{24}.\end{aligned}$$

Therefore,

$$\mathbb{E}\left[\mathbb{E}\left[\mathbb{1}_{X\in C_{i,j}\cap\mathcal{B}}\mathbb{1}_{N_\mathcal{B}=0}p_{i,j}^2 \mid N_\mathcal{B}\right]\right] = \frac{7}{24}\mathbb{P}\left(X\in C_{i,j}\right)\mathbb{P}\left(N_\mathcal{B}=0\right) \tag{2.21}$$

$$= \frac{1}{2^{2^{k^\star}}}\frac{7}{24}. \tag{2.22}$$

Regarding the first term of Equation (2.18),

$$\mathbb{E}[\mathbb{1}_{X\in C_{ij}}\mathbb{1}_{X\in\mathcal{B}}\mathbb{1}_{N_\mathcal{B}>0}\Big(\frac{1}{N_\mathcal{B}}\sum_{i',j'\in\mathcal{B}}p_{i'j'}-p_{ij}\Big)^2|N_\mathcal{B}]$$

$$= \mathbb{E}\left[\mathbb{1}_{X\in C_{ij}}\mathbb{E}\left[\mathbb{1}_{X\in\mathcal{B}}\mathbb{1}_{N_\mathcal{B}>0}\left(\frac{1}{N_\mathcal{B}}\sum_{i',j'}(p_{i',j'}-p_{i,j})\right)^2\Big|X,N_\mathcal{B}\right]\right] \tag{2.23}$$

$$= \mathbb{E}\left[\mathbb{1}_{X\in C_{ij}}\right]\mathbb{E}\left[\mathbb{1}_{p_{i,j}\geqslant\frac{1}{2}}\mathbb{1}_{N_\mathcal{B}>0}\left(\frac{1}{N_\mathcal{B}}\sum_{i',j'\in\mathcal{B}}(p_{i',j'}-p_{i,j})\right)^2\Big|N_\mathcal{B}\right] \tag{2.24}$$

$$\begin{aligned}= \mathbb{E}\left[\mathbb{1}_{X\in C_{ij}}\right]\mathbb{E}\Bigg[\frac{1}{N_\mathcal{B}^2}\Bigg(&\sum_{\substack{i',j',i'',j''\in\mathcal{B}\\(i',j')\neq(i'',j'')}}(p_{i'j'}-p_{ij})(p_{i''j''}-pij)\\ &+\sum_{\substack{i',j'\in\mathcal{B},\\(i',j'\neq(i,j))}}(p_{i'j'}-p_{ij})^2\Bigg)\Big|N_\mathcal{B},N_\mathcal{B}>0,p_{i,j}\geqslant\frac{1}{2}\Bigg]\cdot\mathbb{P}\left(p_{i,j}\geqslant\frac{1}{2}\cap N_\mathcal{B}>0\right).\end{aligned} \tag{2.25}$$

Recall that p_{ij} is drawn uniformly over $[0,1]$. Therefore, $\mathbb{P}\left(p_{i,j}\geqslant\frac{1}{2}\cap N_\mathcal{B}>0\right)=\mathbb{P}\left(p_{i,j}\geqslant\frac{1}{2}\right)=\frac{1}{2}$.

Thus,

$$\begin{aligned}
&\mathbb{E}[\mathbb{1}_{X\in C_{ij}}\mathbb{1}_{X\in\mathcal{B}}\mathbb{1}_{N_\mathcal{B}>0}\Big(\frac{1}{N_\mathcal{B}}\sum_{i',j'\in\mathcal{B}}p_{i'j'}-p_{ij}\Big)^2|N_\mathcal{B}]\\
&=\frac{1}{2^{k^\star}}\mathbb{E}\left[\frac{1}{N_\mathcal{B}^2}(N_\mathcal{B}-1)(N_\mathcal{B}-2)\mathrm{Var}(p_{i,j}\mid p_{i,j}\geqslant\frac{1}{2})\right.\\
&\qquad\left.+\sum_{\substack{i',j'\in\mathcal{B},\\(i',j'\neq(i,j))}}2\mathrm{Var}(p_{i,j}\mid p_{i,j}\geqslant\frac{1}{2})\big|N_\mathcal{B},N_\mathcal{B}>0,p_{i,j}\geqslant\frac{1}{2}\right]\\
&=\frac{1}{2^{k^\star}}\mathbb{E}\left[\frac{1}{N_\mathcal{B}^2}\left((N_\mathcal{B}-1)(N_\mathcal{B}-2)\frac{1}{48}+2\frac{1}{48}(N_\mathcal{B}-1)\right)\big|N_\mathcal{B},N_\mathcal{B}>0\right]\mathbb{P}\left(p_{i,j}\geqslant\frac{1}{2}\cap N_\mathcal{B}>0\right)\\
&=\frac{1}{2^{k^\star+1}}\mathbb{E}\left[\frac{1}{48N_\mathcal{B}^2}\left(N_\mathcal{B}^2-N_\mathcal{B}\right)\mid N_\mathcal{B},N_\mathcal{B}>0\right]\\
&=\frac{1}{48\cdot2^{k^\star+1}}\left(1-\mathbb{E}\left[\frac{1}{N_\mathcal{B}}\mid N_\mathcal{B},N_\mathcal{B}>0\right]\right).
\end{aligned}$$

We now have:

$$\begin{aligned}
&\mathbb{E}\left[\mathbb{E}[\mathbb{1}_{X\in C_{ij}}\mathbb{1}_{X\in\mathcal{B}}\mathbb{1}_{N_\mathcal{B}>0}\Big(\frac{1}{N_\mathcal{B}}\sum_{i',j'\in\mathcal{B}}p_{i'j'}-p_{ij}\Big)^2|N_\mathcal{B}]\right]\\
&=\mathbb{E}\left[\frac{1}{48\cdot2^{k^\star+1}}\left(1-\mathbb{E}\left[\frac{1}{N_\mathcal{B}}\mid N_\mathcal{B},N_\mathcal{B}>0\right]\right)\right]\\
&=\frac{1}{48\cdot2^{k^\star+1}}\left(1-\mathbb{E}\left[\frac{1}{N_\mathcal{B}}\mid N_\mathcal{B}>0\right]\right).
\end{aligned}$$

Notice that $N_\mathcal{B}$ is a binomial variable of parameters $2^{k^\star},1/2$. Thus we can apply Lemma S1 to deduce

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{N_\mathcal{B}}\mid N_\mathcal{B}>0\right]&=\mathbb{E}\left[\frac{\mathbb{1}_{Z>0}}{Z}\right]\frac{1}{\mathbb{P}\left(Z>0\right)}\\
&\leqslant\frac{4}{2^{k^\star}+1}\frac{1}{\mathbb{P}\left(Z>0\right)}
\end{aligned}$$

Moreover, as $\mathbb{P}\left(Z>0\right)\geqslant\frac{1}{2}$, we have:

$$\mathbb{E}\left[\mathbb{E}[\mathbb{1}_{X\in C_{ij}}\mathbb{1}_{X\in\mathcal{B}}\mathbb{1}_{N_\mathcal{B}>0}\Big(\frac{1}{N_\mathcal{B}}\sum_{i',j'\in\mathcal{B}}p_{i'j'}-p_{ij}\Big)^2|N_\mathcal{B}]\right]\geqslant\frac{1}{48\cdot2^{k^\star+1}}\left(1-\frac{8}{2^{k^\star}+1}\right)$$

In the end, the first term of Equation (2.16) verifies

$$\mathbb{E}[(f_{k,1,\infty}(X)-f^\star(X))^2\mathbb{1}_{X\in\mathcal{B}}]\geqslant\frac{1}{2}\Big(\frac{1}{48}\Big(1-\frac{8}{2^{k^\star}-1}\Big)\Big)+\frac{1}{2^{2^{k^\star}}}\frac{7}{24}.$$

Similar computations show that the second term of Equation (2.16) verifies:

$$\begin{aligned}\mathbb{E}[(f_{k,1,\infty}(X) - f^\star(X))^2 \mathbb{1}_{X\in\mathcal{W}}]. &\geqslant \frac{1}{2}\Big(\frac{1}{48}\Big(1 - \frac{8}{2^{k^\star}-1}\Big)\Big) + \mathbb{E}\left[\mathbb{E}\left[\mathbb{1}_{X\in C_{ij}\cap\mathcal{W}}\mathbb{1}_{N_\mathcal{W}=0}p_{ij}^2 \mid N_\mathcal{W}\right]\right] \\ &\geqslant \frac{1}{2}\Big(\frac{1}{48}\Big(1 - \frac{8}{2^{k^\star}-1}\Big)\Big) + \frac{1}{2^{k^\star}}\frac{1}{2^{2^{k^\star}}}\mathbb{E}\left[p_{ij}^2 \mid p_{ij} < \frac{1}{2}\right] \qquad (2.26) \\ &\geqslant \frac{1}{2}\Big(\frac{1}{48}\Big(1 - \frac{8}{2^{k^\star}-1}\Big)\Big) + \frac{1}{2^{2^{k^\star}}}\frac{1}{12}. \qquad (2.27)\end{aligned}$$

All in all, we have

$$\mathbb{E}[(f_{k,1,\infty}(X) - f^\star(X))^2] \geqslant \frac{1}{48}\Big(1 - \frac{8}{2^{k^\star}-1}\Big) + \frac{1}{2^{2^{k^\star}}}\frac{9}{24}.$$

S4 Proof of Lemma 2.4.3

First, note that since we are in an infinite sample regime, the risk of our estimators is equal to their bias term. We can thus work with the true distribution instead of a finite data set.

1. The risk of a second-layer tree cutting k' times, $k' \geqslant k^\star$ along the raw features equals 0 (thus being minimal) as each leaf is included in a cell. We now exhibit one configuration for which any second-layer tree of depth $k' < k^\star$ is biased. We consider the balanced chessboard with parameters $k^\star$, $N_\mathcal{B} = 2^{k^\star-1}$ and p, defined in Proposition 2.4.5 and shown in Figure 2.9. For all $k < k^\star$, each leaf of the first tree contains exactly half black and half white cells, thus predicting $1/2$ and having a risk of $(p - \frac{1}{2})^2$. Therefore a second-layer tree building on raw features only would predict $1/2$ everywhere and would also be biased. If the second-layer tree performs a cut on the new feature provided by the first-layer tree, it creates two leaves: all the leaves where the prediction of the first tree is greater than or equal to $1/2$ are gathered in the right leaf, all the other leaves are gathered in the left leaf. The left leaf is empty and the prediction of the second-layer tree is also $1/2$ everywhere. Any new cut along the new feature would create one leaf predicting $1/2$ on $[0,1]^2$ and other leaves being empty. In any case, the second-layer tree is biased. Thus the minimal risk for all configurations is obtained by a second-layer tree of depth $k' \geqslant k^\star$ which cuts along the raw features only.

2. When $k \geqslant k^\star$, the first tree is unbiased since each of its leaves is included in only one chessboard data cell. Splitting on the new feature in the second-layer tree induces a separation between cells for which $\mathbb{P}[Y = 1|X \in C] = p$ and cells for which $\mathbb{P}[Y = 1|X \in C] = 1 - p$ since $p \neq 1/2$. Taking the expectation of Y on these two regions leads to a shallow tree network of risk zero.

S5 Proof of Proposition 2.4.5

S5.1 Proof of statement 1.: Risk of a Single Tree

Recall that if a cell is empty, the tree prediction in this cell is set (arbitrarily) to zero. Thus,

$$\begin{aligned}
&\mathbb{E}\left[(f_{k,0,n}(X) - f^\star(X))^2\right] \\
&= \mathbb{E}\left[(f_{k,0,n}(X) - f^\star(X))^2 \mathbb{1}_{N_n(L_n(X))>0}\right] + \mathbb{E}\left[(f^\star(X))^2 \mathbb{1}_{N_n(L_n(X))=0}\right], \qquad (2.28)\\
&= \mathbb{E}\left[\left(\frac{1}{N_n(L_n(X))}\sum_{X_i\in L_n(X)} Y_i - f^\star(X)\right)^2 \mathbb{1}_{N_n(L_n(X))>0}\right] + \mathbb{E}\left[(f^\star(X))^2 \mathbb{1}_{N_n(L_n(X))=0}\right], \qquad (2.29)
\end{aligned}$$

where

$$\begin{aligned}
\mathbb{E}\left[(f^\star(X))^2 \mathbb{1}_{N_n(L_n(X))=0}\right] &= \mathbb{E}\left[(f^\star(X))^2 \mathbb{1}_{N_n(L_n(X))=0}\mathbb{1}_{X\in\mathcal{B}}\right] + \mathbb{E}\left[(f^\star(X))^2 \mathbb{1}_{N_n(L_n(X))=0}\mathbb{1}_{X\in\mathcal{W}}\right] \\
&= \left(\frac{p^2}{2} + \frac{(1-p)^2}{2}\right)\mathbb{P}\left(N_n(L_n(X)) = 0\right) \qquad (2.30)\\
&= (p^2 + (1-p)^2)\frac{(1-2^{-k})^n}{2}. \qquad (2.31)
\end{aligned}$$

We now study the first term in (2.29), by considering that X falls into $\mathcal{B}$ (the same computation holds when X falls into $\mathcal{W}$). Letting (X', Y') a generic random variable with the same distribution as (X, Y), one has

$$\begin{aligned}
&\mathbb{E}\left[\left(\frac{1}{N_n(L_n(X))}\sum_{X_i\in L_n(X)} Y_i - p\right)^2 \mathbb{1}_{N_n(L_n(X))>0}\mathbb{1}_{X\in\mathcal{B}}\right] \qquad (2.32)\\
&= \frac{1}{2}\mathbb{E}\left[\left(\frac{1}{N_n(L_n(X))}\sum_{X_i\in L_n(X)} (Y_i - \mathbb{E}\left[Y'|X'\in L_n(X)\right])\right)^2 \mathbb{1}_{N_n(L_n(X))>0}\right] \qquad (2.33)\\
&\qquad + \mathbb{E}\left[\left(\mathbb{E}\left[Y'|X'\in L_n(X)\right] - p\right)^2 \mathbb{1}_{X\in\mathcal{B}}\mathbb{1}_{N_n(L_n(X))>0}\right] \\
&= \frac{1}{2}\mathbb{E}\left[\frac{\mathbb{1}_{N_n(L_n(X))>0}}{N_n(L_n(X))^2}\mathbb{E}\left[\left(\sum_{X_i\in L_n(X)} (Y_i - \mathbb{E}\left[Y'|X'\in L_n(X)\right])\right)^2 \mid N_n(L_n(X))\right]\right] \\
&\qquad + \frac{1}{2}\left(p - \frac{1}{2}\right)^2 \mathbb{P}\left(N_n(L_n(X)) > 0\right), \qquad (2.34)
\end{aligned}$$

where we used the fact that $\mathbb{E}\left[Y'|X'\in L_n(X)\right] = 1/2$ as in any leaf there is the same number of black and white cells. Moreover, conditional to $N_n(L_n(X))$, $\sum_{X_i\in L_n(X)} Y_i$ is a binomial random variable with parameters $\mathfrak{B}(N_n(L_n(X)), \frac{1}{2})$. Hence we obtain :

$$\mathbb{E}\left[\frac{\mathbb{1}_{N_n(L_n(X))>0}}{N_n(L_n(X))^2}\mathbb{E}\left[\left(\sum_{X_i\in L_n(X)}(Y_i-\mathbb{E}\left[Y'|X'\in L_n(X)\right])\right)^2|N_n(L_n(X))\right]\right] \tag{2.35}$$

$$=\frac{1}{4}\mathbb{E}\left[\frac{\mathbb{1}_{N_n(L_n(X))>0}}{N_n(L_n(X))}\right]. \tag{2.36}$$

The same computation holds when X falls into $\mathcal{W}$. Indeed, the left-hand side term in (2.148) is unchanged, as for the right-hand side term, note that $(\frac{1}{2}-p)^2=(\frac{1}{2}-(1-p))^2$. Consequently,

$$\mathbb{E}\left[\left(\frac{1}{N_n(L_n(X))}\sum_{X_i\in L_n(X)}Y_i-f^\star(X)\right)^2\mathbb{1}_{N_n(L_n(X))>0}\right] \tag{2.37}$$

$$=\frac{1}{4}\mathbb{E}\left[\frac{\mathbb{1}_{N_n(L_n(X))>0}}{N_n(L_n(X))}\right]+\left(p-\frac{1}{2}\right)^2(1-(1-2^{-k})^n). \tag{2.38}$$

Injecting (2.38) into (2.29), we have

$$\mathbb{E}\left[(f_{k,0,n}(X)-f^\star(X))^2\right] \tag{2.39}$$

$$=\frac{1}{4}\mathbb{E}\left[\frac{\mathbb{1}_{N_n(L_n(X))>0}}{N_n(L_n(X))}\right]+\left(p-\frac{1}{2}\right)^2(1-(1-2^{-k})^n)+(p^2+(1-p)^2)\frac{(1-2^{-k})^n}{2} \tag{2.40}$$

$$=\frac{1}{4}\mathbb{E}\left[\frac{\mathbb{1}_{N_n(L_n(X))>0}}{N_n(L_n(X))}\right]+\left(p-\frac{1}{2}\right)^2+\left(p^2+(1-p)^2-2\left(p-\frac{1}{2}\right)^2\right)\frac{(1-2^{-k})^n}{2} \tag{2.41}$$

$$=\frac{1}{4}\mathbb{E}\left[\frac{\mathbb{1}_{N_n(L_n(X))>0}}{N_n(L_n(X))}\right]+\left(p-\frac{1}{2}\right)^2+\frac{(1-2^{-k})^n}{4}. \tag{2.42}$$

Noticing that $N_n(L_n(X))$ is a binomial random variable $\mathfrak{B}(n,\frac{1}{2^k})$, we obtain the upper and lower bounds with Lemma S1 (i):

$$\mathbb{E}\left[\frac{\mathbb{1}_{N_n(L_n(X))>0}}{N_n(L_n(X))}\right]\leqslant\frac{2^{k+1}}{n+1}, \tag{2.43}$$

and,

$$\mathbb{E}\left[\frac{\mathbb{1}_{N_n(L_n(X))>0}}{N_n(L_n(X))}\right]\geqslant\left(1-(1-2^{-k})^n\right)\frac{2^k}{n+1}. \tag{2.44}$$

Gathering all the terms gives the result,

$$\mathbb{E}\left[(f_{k,0,n}(X)-f^\star(X))^2\right]\leqslant\left(p-\frac{1}{2}\right)^2+\frac{2^k}{2(n+1)}+\frac{(1-2^{-k})^n}{4}$$

and

$$\mathbb{E}\left[(f_{k,0,n}(X)-f^\star(X))^2\right]\geqslant\left(p-\frac{1}{2}\right)^2+\frac{2^k}{4(n+1)}+\frac{(1-2^{-k})^n}{4}\left(1-\frac{2^k}{n+1}\right).$$

S5.2 Proof of statement 2.: Risk of a Shallow Tree Network

Let $k \in \mathbb{N}$. Denote by $\mathcal{L}_k = \{L_{i,k}, i = 1, \dots, 2^k\}$ the set of all leaves of the encoding tree (of depth k). We let $\mathcal{L}_{\tilde{\mathcal{B}}_k}$ be the set of all cells of the encoding tree containing at least one observation, and such that the empirical probability of Y being equal to one in the cell is larger than $1/2$, i.e.

$$\tilde{\mathcal{B}}_k = \cup_{L \in \mathcal{L}_{\tilde{\mathcal{B}}_k}} \{x, x \in L\}$$

$$\mathcal{L}_{\tilde{\mathcal{B}}_k} = \{L \in \mathcal{L}_k, N_n(L) > 0, \frac{1}{N_n(L)} \sum_{X_i \in L} Y_i \geqslant \frac{1}{2}\}.$$

Similarly,

$$\mathcal{L}_{\tilde{\mathcal{W}}_k} = \{L \in \mathcal{L}_k, N_n(L) > 0, \frac{1}{N_n(L)} \sum_{X_i \in L} Y_i < \frac{1}{2}\}.$$

and

$$\tilde{\mathcal{W}}_k = \cup_{L \in \mathcal{L}_{\tilde{\mathcal{W}}_k}} \{x, x \in L\}$$

Proof of 2. (Upper-Bound)

Recall that $k < k^\star$. In this case, each leaf of the encoding tree contains half black square and half white square (see Figure 2.9a). Hence, the empirical probability of Y being equal to one in such leaf is close to $1/2$. Recalling that our estimate is $f_{k,1,n}$, we have

$$\begin{aligned}
&\mathbb{E}\left[(f_{k,1,n}(X) - f^\star(X))^2\right] \\
&= \mathbb{E}\left[(f_{k,1,n}(X) - p)^2 \mathbb{1}_{X \in \mathcal{B}} \mathbb{1}_{X \in \tilde{\mathcal{B}}_k}\right] + \mathbb{E}\left[(f_{k,1,n}(X) - p)^2 \mathbb{1}_{X \in \mathcal{B}} \mathbb{1}_{X \in \tilde{\mathcal{W}}_k}\right] \\
&+ \mathbb{E}\left[(f_{k,1,n}(X) - (1-p))^2 \mathbb{1}_{X \in \mathcal{W}} \mathbb{1}_{X \in \tilde{\mathcal{B}}_k}\right] + \mathbb{E}\left[(f_{k,1,n}(X) - (1-p))^2 \mathbb{1}_{X \in \mathcal{W}} \mathbb{1}_{X \in \tilde{\mathcal{W}}_k}\right] \\
&+ \mathbb{E}\left[(f_{k,1,n}(X) - p)^2 \mathbb{1}_{X \in \mathcal{B}} (1 - \mathbb{1}_{X \in \tilde{\mathcal{B}}_k} - \mathbb{1}_{X \in \tilde{\mathcal{W}}_k})\right] \\
&+ \mathbb{E}\left[(f_{k,1,n}(X) - (1-p))^2 \mathbb{1}_{X \in \mathcal{W}} (1 - \mathbb{1}_{X \in \tilde{\mathcal{B}}_k} - \mathbb{1}_{X \in \tilde{\mathcal{W}}_k})\right]
\end{aligned} \tag{2.45}$$

Note that $X \notin \tilde{\mathcal{B}}_k \cup \tilde{\mathcal{W}}_k$ is equivalent to X belonging to an empty cell. Besides, the prediction is null by convention in an empty cell. Therefore, the sum of the last two terms in (2.45) can be written as

$$\mathbb{E}\left[p^2 \mathbb{1}_{X \in \mathcal{B}} \mathbb{1}_{N_n(C_n(X))=0})\right] + \mathbb{E}\left[(1-p)^2 \mathbb{1}_{X \in \mathcal{W}} \mathbb{1}_{N_n(C_n(X))=0})\right] = \frac{p^2 + (1-p)^2}{2} \left(1 - \frac{1}{2^k}\right)^n. \tag{2.46}$$

To begin with we focus on the first two terms in (2.45). We deal with the last two terms at the very end as similar computations are conducted.

$$\begin{aligned}
&\mathbb{E}\left[(f_{k,1,n}(X)-p)^2\mathbb{1}_{X\in\mathcal{B}}\mathbb{1}_{X\in\tilde{\mathcal{B}}_k}\right]+\mathbb{E}\left[(f_{k,1,n}(X)-p)^2\mathbb{1}_{X\in\mathcal{B}}\mathbb{1}_{X\in\tilde{\mathcal{W}}_k}\right]\\
&=\mathbb{E}\left[\mathbb{E}\left[\left(\frac{1}{N_n(\tilde{\mathcal{B}}_k)}\sum_{X_i\in\tilde{\mathcal{B}}_k}Y_i-p\right)^2\Big|\tilde{\mathcal{B}}_k\right]\mathbb{P}\left(X\in\tilde{\mathcal{B}}_k,X\in\mathcal{B}|\tilde{\mathcal{B}}_k\right)\right]\\
&\quad+\mathbb{E}\left[\mathbb{E}\left[\left(\frac{1}{N_n(\tilde{\mathcal{W}}_k)}\sum_{X_i\in\tilde{\mathcal{W}}_k}Y_i-p\right)^2\Big|\tilde{\mathcal{W}}_k\right]\mathbb{P}\left(X\in\tilde{\mathcal{W}}_k,X\in\mathcal{B}|\tilde{\mathcal{W}}_k\right)\right]. \qquad (2.47)
\end{aligned}$$

Regarding the left-hand side term in (2.47),

$$\mathbb{E}\left[\left(\frac{1}{N_n(\tilde{\mathcal{B}}_k)}\sum_{X_i\in\tilde{\mathcal{B}}_k}Y_i-p\right)^2\Big|\tilde{\mathcal{B}}_k\right]\leqslant\left(p-\frac{1}{2}\right)^2, \qquad (2.48)$$

since $p>1/2$ and, by definition of $\tilde{\mathcal{B}}_k$,

$$\sum_{X_i\in\tilde{\mathcal{B}}_k}Y_i\geqslant N_n(\tilde{\mathcal{B}}_k)/2.$$

Now, regarding the right-hand side term in (2.47), we let

$$Z_{\tilde{\mathcal{W}}_k}=\mathbb{E}\left[\sum_{X_i\in\tilde{\mathcal{W}}_k}Y_i\mid N_1,...,N_{2^k},\tilde{\mathcal{W}}_k\right],$$

where $N_1,...,N_{2^k}$ denote the number of data points falling in each leaf $L_1,\dots,L_{2^k}$ of the encoding tree. Hence,

$$\begin{aligned}
&\mathbb{E}\left[\left(\frac{1}{N_n(\tilde{\mathcal{W}}_k)}\sum_{X_i\in\tilde{\mathcal{W}}_k}Y_i-p\right)^2\Big|\tilde{\mathcal{W}}_k\right]\\
&=\mathbb{E}\left[\frac{1}{N_n(\tilde{\mathcal{W}}_k)^2}\mathbb{E}\left[\left(\sum_{X_i\in\tilde{\mathcal{W}}_k}Y_i-Z_{\tilde{\mathcal{W}}_k}\right)^2+\left(Z_{\tilde{\mathcal{W}}_k}-N_n(\tilde{\mathcal{W}}_k)p\right)^2\right.\right. \qquad (2.49)\\
&\left.\left.+2\left(\sum_{X_i\in\tilde{\mathcal{W}}_k}Y_i-Z_{\tilde{\mathcal{W}}_k}\right)\left(Z_{\tilde{\mathcal{W}}_k}-N_n(\tilde{\mathcal{W}}_k)p\right)\mid N_1,...,N_{2^k},\tilde{\mathcal{W}}_k\right]\Big|\tilde{\mathcal{W}}_k\right] \qquad (2.50)
\end{aligned}$$

The cross-term is null according to the definition of $Z_{\tilde{\mathcal{W}}_k}$, and since $(Z_{\tilde{W}_k}-N_n(\tilde{W}_k)p)$ is

$(N_1, ..., N_{2^k}, \tilde{\mathcal{W}}_k)$-measurable. Therefore,

$$\begin{aligned}
&\mathbb{E}\left[\left(\frac{1}{N_n(\tilde{\mathcal{W}}_k)}\sum_{X_i\in\tilde{\mathcal{W}}_k} Y_i - p\right)^2 \Big| \tilde{\mathcal{W}}_k\right] \\
&= \mathbb{E}\left[\frac{1}{N_n(\tilde{\mathcal{W}}_k)^2}\mathbb{E}\left[\left(\sum_{X_i\in\tilde{\mathcal{W}}_k} Y_i - Z_{\tilde{\mathcal{W}}_k}\right)^2 \mid N_1, ..., N_{2^k}, \tilde{\mathcal{W}}_k\right] \Big| \tilde{\mathcal{W}}_k\right] \qquad (2.51)\\
&\quad + \mathbb{E}\left[\frac{1}{N_n(\tilde{\mathcal{W}}_k)^2}\mathbb{E}\left[\left(Z_{\tilde{\mathcal{W}}_k} - N_n(\tilde{\mathcal{W}}_k)p\right)^2 \mid N_1, ..., N_{2^k}, \tilde{\mathcal{W}}_k\right] \Big| \tilde{\mathcal{W}}_k\right] \\
&= I_n + J_n, \qquad (2.52)
\end{aligned}$$

where I_n and J_n can be respectively identified as variance and bias terms. Indeed,

$$\mathbb{E}\left[\left(\sum_{X_i\in\tilde{\mathcal{W}}_k} Y_i - Z_{\tilde{\mathcal{W}}_k}\right)^2 \mid N_1, ..., N_{2^k}, \tilde{\mathcal{W}}_k\right]$$

is the variance of a binomial random variable $B(N_n(\tilde{\mathcal{W}}_k), \frac{1}{2})$ conditioned to be lower or equal to $N_n(\tilde{\mathcal{W}}_k)/2$. According to Technical Lemma S1, we have

$$I_n \leqslant \frac{1}{4}\mathbb{E}\left[\frac{\mathbb{1}_{N_n(\tilde{\mathcal{W}}_k)>0}}{N_n(\tilde{\mathcal{W}}_k)\mathbb{P}\left(B(N_n(\tilde{\mathcal{W}}_k), 1/2) \leqslant N_n(\tilde{\mathcal{W}}_k)/2\right)}\Big|\tilde{\mathcal{W}}_k\right] \leqslant \frac{1}{2}\mathbb{E}\left[\frac{\mathbb{1}_{N_n(\tilde{\mathcal{W}}_k)>0}}{N_n(\tilde{\mathcal{W}}_k)}\Big|\tilde{\mathcal{W}}_k\right]. \qquad (2.53)$$

Regarding J_n,

$$\begin{aligned}
Z_{\tilde{\mathcal{W}}_k} - N_n(\tilde{\mathcal{W}}_k)p &= \mathbb{E}\left[\sum_{X_i\in\tilde{\mathcal{W}}_k} Y_i \mid N_1, ..., N_{2^k}, \tilde{\mathcal{W}}_k\right] - N_n(\tilde{\mathcal{W}}_k)p \qquad (2.54)\\
&= \mathbb{E}\left[\sum_{j=1}^{2^k}\sum_{X_i\in L_j} Y_i \mathbb{1}_{L_j\subset\tilde{\mathcal{W}}_k} \mid N_1, ..., N_{2^k}, \tilde{\mathcal{W}}_k\right] - N_n(\tilde{\mathcal{W}}_k)p \qquad (2.55)\\
&= \sum_{j=1}^{2^k}\left(\mathbb{E}\left[\sum_{X_i\in L_j} Y_i \mid N_1, ..., N_{2^k}, \tilde{\mathcal{W}}_k\right] - pN_j\right)\mathbb{1}_{L_j\subset\tilde{\mathcal{W}}_k}, \qquad (2.56)
\end{aligned}$$

since $\mathbb{1}_{L_j\subset\tilde{\mathcal{W}}_k}$ is $\tilde{\mathcal{W}}_k$ -measurable and $N_n(\tilde{\mathcal{W}}_k) = \sum_{i=1}^{2^k} N_j$. Noticing that

$$\mathbb{E}\left[\sum_{X_i\in L_j} Y_i \mid N_1, ..., N_{2^k}, \tilde{\mathcal{W}}_k\right] = \mathbb{E}\left[\sum_{X_i\in L_j} Y_i \mid N_j, \tilde{\mathcal{W}}_k\right], \qquad (2.57)$$

we deduce

$$Z_{\tilde{\mathcal{W}}_k} - N_n(\tilde{\mathcal{W}}_k)p = \sum_{j=1}^{2^k} \left(\mathbb{E}\left[\sum_{X_i \in L_j} Y_i \mid N_j, \tilde{\mathcal{W}}_k \right] - N_j p \right) \mathbb{1}_{L_j \subset \tilde{\mathcal{W}}_k} \tag{2.58}$$

and

$$(Z_{\tilde{\mathcal{W}}_k} - N_n(\tilde{\mathcal{W}}_k)p)^2 = \left(\sum_{j=1}^{2^k} f_j \mathbb{1}_{L_j \subset \tilde{\mathcal{W}}_k} \right)^2 \tag{2.59}$$

with $f_j = \left(N_j p - \mathbb{E}\left[\sum_{X_i \in L_j} Y_i \mid N_j, \tilde{\mathcal{W}}_k \right] \right)$. For all j, such that $L_j \subset \tilde{\mathcal{W}}_k$, $\mathbb{E}\left[\sum_{X_i \in L_j} Y_i \mid N_j, \tilde{\mathcal{W}}_k \right]$ is a binomial random variable $\mathfrak{B}(N_n(\tilde{\mathcal{W}}_k), \frac{1}{2})$ conditioned to be lower or equal to $N_n(\tilde{\mathcal{W}}_k)/2$. Using Lemma S1 (6), we obtain :

$$f_j \leqslant N_j \left(p - \frac{1}{2} \right) + \sqrt{N_j} \left(\frac{1}{\sqrt{\pi}} + \frac{2\sqrt{2}}{\pi\sqrt{(2n+1)}} \right) \tag{2.60}$$

$$\leqslant N_j \left(p - \frac{1}{2} \right) + \sqrt{N_j} + \frac{2}{\pi}. \tag{2.61}$$

Therefore,

$$(Z_{\tilde{\mathcal{W}}_k} - N_n(\tilde{\mathcal{W}}_k)p)^2 \leqslant \left(N_n(\tilde{\mathcal{W}}_k) \left(p - \frac{1}{2} \right) + \sum_{j=1}^{2^k} \sqrt{N_j} \mathbb{1}_{L_j \subset \tilde{\mathcal{W}}_k} + \frac{2^{k+1}}{\pi} \right)^2 \tag{2.62}$$

$$\leqslant \left(N_n(\tilde{\mathcal{W}}_k) \left(p - \frac{1}{2} \right) + 2^{k/2} \sqrt{N_n(\tilde{\mathcal{W}}_k)} + \frac{2^{k+1}}{\pi} \right)^2, \tag{2.63}$$

since, according to Cauchy-Schwarz inequality,

$$\sum_{j=1}^{2^k} \sqrt{N_j} \mathbb{1}_{L_j \subset \tilde{\mathcal{W}}_k} \leqslant 2^{k/2} N_n(\tilde{\mathcal{W}}_k)^{1/2}. \tag{2.64}$$

Overall

$$J_n \leqslant \mathbb{E}\left[\frac{1}{N_n(\tilde{\mathcal{W}}_k)^2} \mathbb{E}\left[\left(N_n(\tilde{\mathcal{W}}_k) \left(p - \frac{1}{2} \right) + 2^{k/2} N_n(\tilde{\mathcal{W}}_k)^{1/2} + \frac{2^{k+1}}{\pi} \right)^2 \mid N_1, ..., N_{2^k}, \tilde{\mathcal{W}}_k \right] \Big| \tilde{\mathcal{W}}_k \right] \tag{2.65}$$

$$\begin{aligned} &\leqslant \left(p - \frac{1}{2} \right)^2 + 2^k \mathbb{E}\left[\frac{\mathbb{1}_{N_n(\tilde{\mathcal{W}}_k)>0}}{N_n(\tilde{\mathcal{W}}_k)} \Big| \tilde{\mathcal{W}}_k \right] \\ &+ \frac{2^{2k+2}}{\pi^2} \mathbb{E}\left[\frac{\mathbb{1}_{N_n(\tilde{\mathcal{W}}_k)>0}}{N_n(\tilde{\mathcal{W}}_k)^2} \Big| \tilde{\mathcal{W}}_k \right] + 2^{k/2+1} \left(p - \frac{1}{2} \right) \mathbb{E}\left[\frac{\mathbb{1}_{N_n(\tilde{\mathcal{W}}_k)>0}}{N_n(\tilde{\mathcal{W}}_k)^{1/2}} \Big| \tilde{\mathcal{W}}_k \right] \\ &+ \frac{2^{k+2}}{\pi} \left(p - \frac{1}{2} \right) \mathbb{E}\left[\frac{\mathbb{1}_{N_n(\tilde{\mathcal{W}}_k)>0}}{N_n(\tilde{\mathcal{W}}_k)} \Big| \tilde{\mathcal{W}}_k \right] + \frac{2^{\frac{3k}{2}+2}}{\pi} \mathbb{E}\left[\frac{\mathbb{1}_{N_n(\tilde{\mathcal{W}}_k)>0}}{N_n(\tilde{\mathcal{W}}_k)^{3/2}} \Big| \tilde{\mathcal{W}}_k \right]. \end{aligned} \tag{2.66}$$

All together, we obtain

$$\begin{aligned}
&I_n + J_n \\
&\leqslant \left(p - \frac{1}{2}\right)^2 + \left(2^k + \frac{1}{2} + \frac{2^{k+2}}{\pi}\left(p - \frac{1}{2}\right)\right)\mathbb{E}\left[\frac{\mathbb{1}_{N_n(\tilde{\mathcal{W}}_k)>0}}{N_n(\tilde{\mathcal{W}}_k)}\Big|\tilde{\mathcal{W}}_k\right] + \frac{2^{2k+2}}{\pi^2}\mathbb{E}\left[\frac{\mathbb{1}_{N_n(\tilde{\mathcal{W}}_k)>0}}{N_n(\tilde{\mathcal{W}}_k)^2}\Big|\tilde{\mathcal{W}}_k\right] \\
&\quad + 2^{k/2+1}\left(p - \frac{1}{2}\right)\mathbb{E}\left[\frac{\mathbb{1}_{N_n(\tilde{\mathcal{W}}_k)>0}}{N_n(\tilde{\mathcal{W}}_k)^{1/2}}\Big|\tilde{\mathcal{W}}_k\right] + \frac{2^{\frac{3k}{2}+2}}{\pi}\mathbb{E}\left[\frac{\mathbb{1}_{N_n(\tilde{\mathcal{W}}_k)>0}}{N_n(\tilde{\mathcal{W}}_k)^{3/2}}\Big|\tilde{\mathcal{W}}_k\right]
\end{aligned}$$

We apply Lemma S1(i)(iv) to $N_n(\tilde{\mathcal{W}}_k)$ which is a binomial $\mathfrak{B}(n, p')$ where $p' = \mathbb{P}(X \in \tilde{\mathcal{W}}_k|\tilde{\mathcal{W}}_k)$:

$$\mathbb{E}\left[\frac{\mathbb{1}_{N_n(\tilde{\mathcal{W}}_k)>0}}{N_n(\tilde{\mathcal{W}}_k)}\Big|\tilde{\mathcal{W}}_k\right] \leqslant \frac{2}{(n+1)p'},$$

$$\mathbb{E}\left[\frac{\mathbb{1}_{N_n(\tilde{\mathcal{W}}_k)>0}}{N_n(\tilde{\mathcal{W}}_k)^{1/2}}\Big|\tilde{\mathcal{W}}_k\right] \leqslant \frac{2}{\sqrt{n \cdot p'}}.$$

We deduce that

$$I_n + J_n \leqslant (p - \frac{1}{2})^2 + \frac{2^{k/2+2}(p - \frac{1}{2})}{\sqrt{\pi n \cdot p'}} + \frac{2}{(n+1)\cdot p'}\left(2^k + \frac{1}{2} + \frac{2^{k+2}}{\pi} + \frac{2^{3k/2+2}}{\pi\sqrt{\pi}} + 3 \cdot \frac{2^{2k+2}}{\pi^2}\right).$$

Finally,

$$\begin{aligned}
&\mathbb{E}\left[(f_{k,1,n}(X) - p)^2 \mathbb{1}_{X\in\mathcal{B}}\mathbb{1}_{X\in\tilde{\mathcal{B}}_k}\right] + \mathbb{E}\left[(f_{k,1,n}(X) - p)^2 \mathbb{1}_{X\in\mathcal{B}}\mathbb{1}_{X\in\tilde{\mathcal{W}}_k}\right] \\
&\leqslant \left(p - \frac{1}{2}\right)^2 \mathbb{P}\left(X \in \tilde{\mathcal{B}}_k, X \in \mathcal{B}\right) + \mathbb{E}\left[(I_n + J_n)\mathbb{P}\left(X \in \tilde{\mathcal{W}}_k, X \in \mathcal{B}|\tilde{\mathcal{W}}_k\right)\right]
\end{aligned}$$

Since for all $\tilde{\mathcal{B}}_k$, there is exactly the same number of black cells and white cells in $\tilde{\mathcal{B}}_k$, we have

$$\mathbb{P}\left(X \in \tilde{\mathcal{W}}_k, X \in \mathcal{B}|\tilde{\mathcal{W}}_k\right) = \frac{\mathbb{P}\left(X \in \tilde{\mathcal{W}}_k|\tilde{\mathcal{W}}_k\right)}{2},$$

yielding

$$\mathbb{E}\left[(f_{k,1,n}(X) - p)^2 \mathbb{1}_{X\in\mathcal{B}}\mathbb{1}_{X\in\tilde{\mathcal{B}}_k}\right] + \mathbb{E}\left[(f_{k,1,n}(X) - p)^2 \mathbb{1}_{X\in\mathcal{B}}\mathbb{1}_{X\in\tilde{\mathcal{W}}_k}\right] \tag{2.67}$$

$$\leqslant \frac{1}{2}\left(p - \frac{1}{2}\right)^2 + \frac{2^{k/2+1}(p - \frac{1}{2})}{\sqrt{\pi n}} + \frac{1}{(n+1)}\left(2^k + \frac{1}{2} + \frac{2^{k+2}}{\pi} + \frac{2^{3k/2+2}}{\pi\sqrt{\pi}} + 3 \cdot \frac{2^{2k+2}}{\pi^2}\right) \tag{2.68}$$

$$\leqslant \frac{1}{2}\left(p - \frac{1}{2}\right)^2 + \frac{2^{k/2+1}(p - \frac{1}{2})}{\sqrt{\pi n}} + \frac{3 \cdot 2^{2k+2}}{(n+1)\pi^2}(1 + \varepsilon_1(k)) \tag{2.69}$$

where $\varepsilon_1(k) = \frac{\pi^2}{3\cdot 2^{(2k+2)}}\left(2^k + \frac{1}{2} + \frac{2^{k+2}}{\pi} + \frac{2^{3k/2+2}}{\pi\sqrt{\pi}}\right)$.

The two intermediate terms of (2.45) can be similarly bounded from above. Indeed,

$$\mathbb{E}\left[(f_{k,1,n}(X)-(1-p))^2\mathbb{1}_{X\in\mathcal{W}}\mathbb{1}_{X\in\tilde{\mathcal{B}}_k}\right]+\mathbb{E}\left[(f_{k,1,n}(X)-(1-p))^2\mathbb{1}_{X\in\mathcal{W}}\mathbb{1}_{X\in\tilde{\mathcal{W}}_k}\right] \tag{2.70}$$

$$\begin{aligned}&=\mathbb{E}\left[\mathbb{E}\left[\left(\frac{1}{N_n(\tilde{\mathcal{B}}_k)}\sum_{X_i\in\tilde{\mathcal{B}}_k}Y_i-(1-p)\right)^2\Big|\tilde{\mathcal{B}}_k\right]\mathbb{P}\left(X\in\tilde{\mathcal{B}}_k,X\in\mathcal{W}|\tilde{\mathcal{B}}_k\right)\right]\\&\quad+\mathbb{E}\left[\mathbb{E}\left[\left(\frac{1}{N_n(\tilde{\mathcal{W}}_k)}\sum_{X_i\in\tilde{\mathcal{W}}_k}Y_i-(1-p)\right)^2\Big|\tilde{\mathcal{W}}_k\right]\mathbb{P}\left(X\in\tilde{\mathcal{W}}_k,X\in\mathcal{W}|\tilde{\mathcal{W}}_k\right)\right],\end{aligned} \tag{2.71}$$

where, by definition of $\tilde{\mathcal{W}}_k$,

$$\mathbb{E}\left[\left(\frac{1}{N_n(\tilde{\mathcal{W}}_k)}\sum_{X_i\in\tilde{\mathcal{W}}_k}Y_i-(1-p)\right)^2\Big|\tilde{\mathcal{W}}_k\right]\leqslant\left(p-\frac{1}{2}\right)^2.$$

The first term in (2.71) can be treated similarly as above:

$$\begin{aligned}&\mathbb{E}\left[\left(\frac{1}{N_n(\tilde{\mathcal{B}}_k)}\sum_{X_i\in\tilde{\mathcal{B}}_k}Y_i-(1-p)\right)^2\Big|\tilde{\mathcal{B}}_k\right]\\&=\mathbb{E}\left[\frac{1}{N_n(\tilde{\mathcal{B}}_k)^2}\mathbb{E}\left[\left(\sum_{X_i\in\tilde{\mathcal{B}}_k}Y_i-Z_{\tilde{\mathcal{B}}_k}\right)^2\mid N_1,...,N_{2^k},\tilde{\mathcal{B}}_k\right]\Big|\tilde{\mathcal{B}}_k\right]\\&\quad+\mathbb{E}\left[\frac{1}{N_n(\tilde{\mathcal{B}}_k)^2}\mathbb{E}\left[\left(Z_{\tilde{\mathcal{B}}_k}-N_n(\tilde{\mathcal{B}}_k)(1-p)\right)^2\mid N_1,...,N_{2^k},\tilde{\mathcal{B}}_k\right]\Big|\tilde{\mathcal{B}}_k\right]\\&=I_n'+J_n',\end{aligned} \tag{2.72}$$

where

$$Z_{\tilde{\mathcal{B}}_k}=\mathbb{E}\left[\sum_{X_i\in\tilde{\mathcal{B}}_k}Y_i\mid N_1,...,N_{2^k},\tilde{\mathcal{B}}_k\right],$$

and the cross-term in (2.72) is null according to the definition of $Z_{\tilde{\mathcal{B}}_k}$. Regarding I_n', note that

$$\mathbb{E}\left[\left(\sum_{X_i\in\tilde{\mathcal{B}}_k}Y_i-Z_{\tilde{\mathcal{B}}_k}\right)^2\mid N_1,...,N_{2^k},\tilde{\mathcal{B}}_k\right]$$

is the variance of a binomial random variable $B(N_n(\tilde{\mathcal{B}}_k),\frac{1}{2})$ conditioned to be strictly larger than $N_n(\tilde{\mathcal{B}}_k)/2$. According to Technical Lemma S1, we have

$$I_n'\leqslant\frac{1}{4}\mathbb{E}\left[\frac{\mathbb{1}_{N_n(\tilde{\mathcal{B}}_k)>0}}{N_n(\tilde{\mathcal{B}}_k)\mathbb{P}\left(B(N_n(\tilde{\mathcal{B}}_k),1/2)>N_n(\tilde{\mathcal{B}}_k)/2\right)}\Big|\tilde{\mathcal{B}}_k\right]\leqslant\mathbb{E}\left[\frac{\mathbb{1}_{N_n(\tilde{\mathcal{B}}_k)>0}}{N_n(\tilde{\mathcal{B}}_k)}\Big|\tilde{\mathcal{B}}_k\right]. \tag{2.73}$$

To obtain the last inequality, notice that

$$\mathbb{P}\left(B(N_n(\tilde{\mathcal{B}}_k),1/2) > N_n(\tilde{\mathcal{B}}_k)/2\right) = \frac{1}{2} - \frac{1}{2}\mathbb{P}\left(B(N_n(\tilde{\mathcal{B}}_k),1/2) = N_n(\tilde{\mathcal{B}}_k)/2\right)$$
$$\geqslant \frac{1}{2}\left(1 - \frac{1}{\sqrt{\pi(n/2+1/4)}}\right) \geqslant \frac{1}{4}$$

as soon as $n \geqslant 4$.
Regarding J_n', we have

$$\mathbb{E}\left[\frac{1}{N_n(\tilde{\mathcal{B}}_k)^2}\mathbb{E}\left[\left(Z_{\tilde{\mathcal{B}}_k} - N_n(\tilde{\mathcal{B}}_k)(1-p)\right)^2 \mid N_1,...,N_{2^k},\tilde{\mathcal{B}}_k\right]\right] \tag{2.74}$$
$$= \mathbb{E}\left[\frac{1}{N_n(\tilde{\mathcal{B}}_k)^2}\mathbb{E}\left[\left(\sum_{i=1}^{2^k}\left(\mathbb{E}\left[\sum_{X_i\in L_i} Y_i \mid N_j,\tilde{\mathcal{B}}_k\right] - N_j(1-p)\right)\mathbb{1}_{L_j\subset\tilde{\mathcal{B}}_k}\right)^2 \mid N_1,...,N_{2^k},\tilde{\mathcal{B}}_k\right]\right]. \tag{2.75}$$

For all j, such that $L_j \subset \tilde{\mathcal{B}}_k$, $\mathbb{E}\left[\sum_{X_i\in L_j} Y_i \mid N_j,\tilde{\mathcal{B}}_k\right]$ is a binomial random variable $\mathfrak{B}(N_j,\frac{1}{2})$ conditioned to be larger than $\lfloor (N_j+1)/2\rfloor$. Then, according to Technical Lemma (7)

$$\mathbb{E}\left[\sum_{X_i\in L_j} Y_i \mid N_j,\tilde{\mathcal{B}}_k\right] \leqslant \frac{N_j}{2} + 1 + \frac{1}{\sqrt{\pi(N_j+1)}}.$$

Hence,

$$\mathbb{E}\left[\sum_{X_i\in L_i} Y_i \mid N_j,\tilde{\mathcal{B}}_k\right] - N_j(1-p) \leqslant N_j(p-\frac{1}{2}) + 1 + \frac{1}{\sqrt{\pi(N_j+1)}} \tag{2.76}$$
$$\leqslant N_j\left(p-\frac{1}{2}\right) + \sqrt{N_j} + \frac{2}{\pi}, \tag{2.77}$$

for $N_j \geqslant 1$. Thus,

$$\mathbb{E}\left[\frac{1}{N_n(\tilde{\mathcal{B}}_k)^2}\mathbb{E}\left[\left(Z_{\tilde{\mathcal{B}}_k} - N_n(\tilde{\mathcal{B}}_k)(1-p)\right)^2 \mid N_1,...,N_{2^k},\tilde{\mathcal{B}}_k\right]\right] \tag{2.78}$$
$$\leqslant \mathbb{E}\left[\frac{1}{N_n(\tilde{\mathcal{B}}_k)^2}\mathbb{E}\left[\left(\sum_{i=1}^{2^k}\left(N_j\left(p-\frac{1}{2}\right) + \sqrt{N_j} + \frac{2}{\pi}\right)\mathbb{1}_{L_j\subset\tilde{\mathcal{B}}_k}\right)^2 \mid N_1,...,N_{2^k},\tilde{\mathcal{B}}_k\right]\right] \tag{2.79}$$
$$\leqslant \mathbb{E}\left[\frac{1}{N_n(\tilde{\mathcal{B}}_k)^2}\mathbb{E}\left[\left(N_n(\tilde{\mathcal{B}}_k)\left(p-\frac{1}{2}\right) + 2^{k/2}\sqrt{N_n(\tilde{\mathcal{B}}_k)} + \frac{2^{k+1}}{\pi}\right)^2 \mid N_1,...,N_{2^k},\tilde{\mathcal{B}}_k\right]\right]. \tag{2.80}$$

All together, we obtain

$$I'_n + J'_n \leqslant \left(p - \frac{1}{2}\right)^2 + \left(2^k + 1 + \frac{2^{k+2}}{\pi}\left(p - \frac{1}{2}\right)\right)\mathbb{E}\left[\frac{\mathbb{1}_{N_n(\tilde{\mathcal{B}}_k)>0}}{N_n(\tilde{\mathcal{B}}_k)}\middle|\tilde{\mathcal{B}}_k\right] + \frac{2^{2k+2}}{\pi^2}\mathbb{E}\left[\frac{\mathbb{1}_{N_n(\tilde{\mathcal{B}}_k)>0}}{N_n(\tilde{\mathcal{B}}_k)^2}\middle|\tilde{\mathcal{B}}_k\right]$$
$$+ 2^{k/2+1}\left(p - \frac{1}{2}\right)\mathbb{E}\left[\frac{\mathbb{1}_{N_n(\tilde{\mathcal{B}}_k)>0}}{N_n(\tilde{\mathcal{B}}_k)^{1/2}}\middle|\tilde{\mathcal{B}}_k\right] + \frac{2^{\frac{3k}{2}+2}}{\pi}\mathbb{E}\left[\frac{\mathbb{1}_{N_n(\tilde{\mathcal{B}}_k)>0}}{N_n(\tilde{\mathcal{B}}_k)^{3/2}}\middle|\tilde{\mathcal{B}}_k\right]$$

The computation is similar to (2.67), with $p'' = \mathbb{P}\left(X \in \tilde{\mathcal{B}}_k \mid \tilde{\mathcal{B}}_k\right)$:

$$I_n + J_n \leqslant \left(p - \frac{1}{2}\right)^2 + \frac{2^{k/2+3}(p - \frac{1}{2})}{\sqrt{\pi n \cdot p''}} + \left(2^k + 1 + \frac{2^{k+2}}{\pi}\left(p - \frac{1}{2}\right) + \frac{2^{3k/2+2}}{\pi} + \frac{2^{2k+2}}{\pi^2}\right)\frac{2}{(n+1)p''}$$
$$\leqslant \left(p - \frac{1}{2}\right)^2 + \frac{2^{k/2+3}(p - \frac{1}{2})}{\sqrt{\pi n \cdot p''}} + \frac{2^{2k+3}}{\pi^2(n+1)p''}(1 + \varepsilon_2(k))$$

with $\varepsilon_2(k) = \frac{\pi^2}{2^{(2k+3)}}\left(2^k + 1 + \frac{2^{k+2}}{\pi}(p - 1/2) + \frac{2^{3k/2+2}}{\pi}\right)$. Finally,

$$\mathbb{E}\left[(f_{k,1,n}(X) - (1-p))^2 \mathbb{1}_{X\in\mathcal{W}}\mathbb{1}_{X\in\tilde{\mathcal{B}}_k}\right] + \mathbb{E}\left[(f_{k,1,n}(X) - (1-p))^2 \mathbb{1}_{X\in\mathcal{W}}\mathbb{1}_{X\in\tilde{\mathcal{W}}_k}\right]$$
$$\leqslant \mathbb{E}\left[(I'_n + J'_n)\mathbb{P}\left(X \in \mathcal{W}, X \in \tilde{\mathcal{B}}_k | \tilde{\mathcal{B}}_k\right)\right] + \left(p - \frac{1}{2}\right)^2 \mathbb{P}\left(X \in \mathcal{W}, X \in \tilde{\mathcal{W}}_k\right)$$
$$\leqslant \mathbb{E}\left[\left(\left(p - \frac{1}{2}\right)^2 + \frac{2^{k/2+3}(p - \frac{1}{2})}{\sqrt{\pi n \cdot p''}} + \frac{2^{2k+3}}{\pi^2(n+1)p''}(1 + \varepsilon_2(k))\right)\mathbb{P}\left(X \in \mathcal{W}, X \in \tilde{\mathcal{B}}_k | \tilde{\mathcal{B}}_k\right)\right]$$
$$+ \left(p - \frac{1}{2}\right)^2 \mathbb{P}\left(X \in \mathcal{W}, X \in \tilde{\mathcal{W}}_k\right).$$

Since for all $\tilde{\mathcal{B}}_k$, there is exactly the same number of black cells and white cells in $\tilde{\mathcal{B}}_k$, we have

$$\mathbb{P}\left(X \in \mathcal{W}, X \in \tilde{\mathcal{B}}_k | \tilde{\mathcal{B}}_k\right) = \frac{p''}{2},$$

yielding

$$\mathbb{E}\left[(f_{k,1,n}(X) - (1-p))^2 \mathbb{1}_{X\in\mathcal{W}}\mathbb{1}_{X\in\tilde{\mathcal{B}}_k}\right] + \mathbb{E}\left[(f_{k,1,n}(X) - (1-p))^2 \mathbb{1}_{X\in\mathcal{W}}\mathbb{1}_{X\in\tilde{\mathcal{W}}_k}\right]$$
$$\leqslant \frac{1}{2}\left(p - \frac{1}{2}\right)^2 + \frac{2^{k/2+2}(p - \frac{1}{2})}{\sqrt{\pi n}} + \frac{2^{2k+3}}{2 \cdot \pi^2(n+1)}(1 + \varepsilon_2(k)). \tag{2.81}$$

Gathering (2.46), (2.69) and (2.81), we have

$$\mathbb{E}\left[(f_{k,1,n}(X) - f^\star(X))^2\right] \leqslant \left(p - \frac{1}{2}\right)^2 + \frac{2^{k/2+3}(p - \frac{1}{2})}{\sqrt{\pi n}} + \frac{7 \cdot 2^{2k+2}}{\pi^2(n+1)}(1 + \varepsilon(k))$$
$$+ \frac{p^2 + (1-p)^2}{2}\left(1 - \frac{1}{2^k}\right)^n$$

where $\varepsilon(k) = \frac{6\varepsilon_1(k) + \varepsilon_2(k)}{7}$.

Proof of 2. (Lower-Bound)

First, note that $f_{k,1,n}$ is constant on each element C of the partition built by the first layer tree. Therefore, for all C,denoting $\mathcal{D}_n$ the dataset $(X_1, Y_1), ...(X_n, Y_n)$,

$$\mathbb{E}\left[(f_{k,1,n}(X) - f^\star(X))^2)\middle|X \in C, \mathcal{D}_n\right] \geqslant Var\left(f^\star(X))^2\middle|X \in C\right). \tag{2.82}$$

As the first layer tree is shallow, each of its leaf contains several squares of the dataset and $\mathbb{E}\left[f^\star(X)\right] = 1/2$. Therefore,

$$\mathbb{E}\left[(f_{k,1,n}(X) - f^\star(X))^2)\middle|X \in C, \mathcal{D}_n\right] \geqslant \mathbb{E}\left[(f^\star(X) - \frac{1}{2})^2)\middle|X \in C, \mathcal{D}_n\right] \tag{2.83}$$

$$\geqslant (p - \frac{1}{2})^2. \tag{2.84}$$

Overall,

$$\mathbb{E}\left[(f_{k,1,n}(X) - f^\star(X))^2)\right] \geqslant (p - \frac{1}{2})^2. \tag{2.85}$$

S6 Proof of Proposition 2.4.6

S6.1 Proof of statement 1.: Risk of a Single Tree

As in the precedent proof, we distinguish the case where the cell containing X might be empty, in such a case the tree will predict 0:

$$\begin{aligned}
&\mathbb{E}\left[(f_{k,0,n}(X) - f^\star(X))^2)\right] \\
&\quad = \mathbb{E}\left[(f_{k,0,n}(X) - f^\star(X))^2 \mathbb{1}_{N_n(L_n(X))>0}\right] + \mathbb{E}\left[(f^\star(X))^2 \mathbb{1}_{N_n(L_n(X))=0}\right] && (2.86)\\
&\quad = \mathbb{E}\left[(f_{k,0,n}(X) - f^\star(X))^2 \mathbb{1}_{N_n(L_n(X))>0}\right] + (p^2 + (1-p)^2)\frac{(1-2^{-k})^n}{2}. && (2.87)
\end{aligned}$$

We denote by $L_1, ..., L_{2^k}$ the leaves of the tree. Let $b \in \{1, \dots, 2^k\}$ such that L_b belongs to $\mathcal{B}$. We have

$$\begin{aligned}
&\mathbb{E}\left[(f_{k,0,n}(X) - p)^2) \mathbb{1}_{X\in\mathcal{B}} \mathbb{1}_{N_n(L_n(X))>0}\right] \\
&\quad = \sum_{L_j \subset \mathcal{B}} \mathbb{E}\left[\left(\frac{\mathbb{1}_{N_n(L_j)>0}}{N_n(L_j)} \sum_{X_i \in L_j} (Y_i - p)\right)^2 \mathbb{1}_{X\in L_j}\right] && (2.88)\\
&\quad = \frac{2^k}{2} \cdot \mathbb{E}\left[\left(\frac{\mathbb{1}_{N_n(L_b)>0}}{N_n(L_b)} \sum_{X_i \in L_b} (Y_i - p)\right)^2\right] \mathbb{P}\left(X \in L_b\right) && (2.89)\\
&\quad = \frac{1}{2}\mathbb{E}\left[\left(\frac{\mathbb{1}_{N_n(L_b)>0}}{N_n(L_b)} \sum_{X_i \in L_b} (Y_i - p)\right)^2\right] && (2.90)\\
&\quad = \frac{1}{2}\mathbb{E}\left[\frac{\mathbb{1}_{N_n(L_b)>0}}{N_n(L_b)^2} \mathbb{E}\left[\left(\sum_{X_i \in L_b} (Y_i - p)\right)^2 | N_n(L_b)\right]\right] && (2.91)\\
&\quad = \frac{1}{2}\mathbb{E}\left[\frac{\mathbb{1}_{N_n(L_b)>0}}{N_n(L_b)^2} \mathbb{E}\left[\sum_{X_i \in L_b} (Y_i - p)^2 | N_n(L_b)\right]\right] \quad \text{(by independence of the } Y_i\text{)} && (2.92)\\
&\quad = \frac{1}{2}\mathbb{E}\left[\frac{\mathbb{1}_{N_n(L_b)>0}}{N_n(L_b)} p(1-p)\right]. && (2.93)
\end{aligned}$$

Remark that the above computation holds when $X \in \mathcal{W}$ after replacing p by $(1-p)$, $\mathcal{B}$ by $\mathcal{W}$ and L_b by L_w: indeed when Y is a Bernoulli random variable, Y and $1-Y$ have the same variance. Hence, using Equation (2.87), the computation in (2.93) and its equivalence for $\mathcal{W}$, we obtain

$$\begin{aligned}
&\mathbb{E}\left[(f_{k,0,n}(X) - f^\star(X))^2)\right] \\
&= \frac{1}{2}\mathbb{E}\left[\frac{\mathbb{1}_{N_n(L_b)>0}}{N_n(L_b)} p(1-p)\right] + \frac{1}{2}\mathbb{E}\left[\frac{\mathbb{1}_{N_n(L_w)>0}}{N_n(L_w)} p(1-p)\right] + (p^2 + (1-p)^2)\frac{(1-2^{-k})^n}{2} \\
&= p(1-p)\mathbb{E}\left[\frac{\mathbb{1}_{N_n(L_w)>0}}{N_n(L_w)}\right] + (p^2 + (1-p)^2)\frac{(1-2^{-k})^n}{2},
\end{aligned}$$

since $N_n(L_b)$ and $N_n(L_w)$ are both binomial random variables $\mathfrak{B}(n, \frac{1}{2^k})$. Therefore we can con-

clude using Lemma S1 (i):

$$\mathbb{E}\left[(f_{k,0,n}(X) - f^\star(X))^2)\right] \leqslant \frac{2^k p(1-p)}{n+1} + \left(p^2 + (1-p)^2\right)\frac{(1-2^{-k})^n}{2}$$

and

$$\mathbb{E}\left[(f_{k,0,n}(X) - f^\star(X))^2)\right] \geqslant \frac{2^{k-1} p(1-p)}{n+1} + \left(p^2 + (1-p)^2 - \frac{2^k p(1-p)}{n+1}\right)\frac{(1-2^{-k})^n}{2}.$$

S6.2 Proof of Statement 2.: Risk of a Shallow Tree Network

Let $k \in \mathbb{N}$. Denote by $\mathcal{L}_k = \{L_i, i = 1, \ldots, 2^k\}$ the set of all leaves of the encoding tree (of depth k). We let $\mathcal{L}_{\tilde{\mathcal{B}}_k}$ be the set of all cells of the encoding tree containing at least one observation, and such that the empirical probability of Y being equal to one in the cell is larger than $1/2$, i.e.

$$\tilde{\mathcal{B}}_k = \cup_{L \in \mathcal{L}_{\tilde{\mathcal{B}}_k}} \{x, x \in L\}$$

$$\mathcal{L}_{\tilde{\mathcal{B}}_k} = \{L \in \mathcal{L}_k, N_n(L) > 0, \frac{1}{N_n(L)} \sum_{X_i \in L} Y_i \geqslant \frac{1}{2}\}.$$

Accordingly, we let the part of the input space corresponding to $\mathcal{L}_{\tilde{\mathcal{B}}_k}$ as

$$\tilde{\mathcal{B}}_k = \cup_{L \in \mathcal{L}_{\tilde{\mathcal{B}}_k}} \{x, x \in L\}$$

Similarly,

$$\mathcal{L}_{\tilde{\mathcal{W}}_k} = \{L \in \mathcal{L}_k, N_n(L) > 0, \frac{1}{N_n(L)} \sum_{X_i \in L} Y_i < \frac{1}{2}\}.$$

and

$$\tilde{\mathcal{W}}_k = \cup_{L \in \mathcal{L}_{\tilde{\mathcal{W}}_k}} \{x, x \in L\}$$

Proof of 2. (Upper-Bound)

Recall that $k \geqslant k^\star$. In this case, each leaf of the encoding tree is included in a chessboard cell. As usual,

$$\mathbb{E}\left[(f_{k,1,n}(X) - f^\star(X))^2)\right] = \mathbb{E}\left[(f_{k,1,n}(X) - f^\star(X))^2 \mathbb{1}_{N_n(L_n(X))>0}\right] + \frac{p^2 + (1-p)^2}{2}\left(1 - \frac{1}{2^k}\right)^n.$$

Note that

$$
\begin{aligned}
&\mathbb{E}\left[(f_{k,1,n}(X)-f^\star(X))^2 \mathbb{1}_{N_n(L_n(X))>0}\right] \\
&= \mathbb{E}\left[\left(\frac{1}{N_n(\tilde{\mathcal{B}}_k)}\sum_{X_i\in\tilde{\mathcal{B}}_k} Y_i - p\right)^2 \mathbb{1}_{X\in\mathcal{B}}\mathbb{1}_{X\in\tilde{\mathcal{B}}_k}\right] + \mathbb{E}\left[\left(\frac{1}{N_n(\tilde{\mathcal{W}}_k)}\sum_{X_i\in\tilde{\mathcal{W}}_k} Y_i - p\right)^2 \mathbb{1}_{X\in\mathcal{B}}\mathbb{1}_{X\in\tilde{\mathcal{W}}_k}\right] \\
&+ \mathbb{E}\left[\left(\frac{1}{N_n(\tilde{\mathcal{B}}_k)}\sum_{X_i\in\tilde{\mathcal{B}}_k} Y_i - (1-p)\right)^2 \mathbb{1}_{X\in\mathcal{W}}\mathbb{1}_{X\in\tilde{\mathcal{B}}_k}\right] \\
&+ \mathbb{E}\left[\left(\frac{1}{N_n(\tilde{\mathcal{W}}_k)}\sum_{X_i\in\tilde{\mathcal{W}}_k} Y_i - (1-p)\right)^2 \mathbb{1}_{X\in\mathcal{W}}\mathbb{1}_{X\in\tilde{\mathcal{W}}_k}\right] \\
&\leqslant \frac{1}{2}\mathbb{E}\left[\left(\frac{1}{N_n(\tilde{\mathcal{B}}_k)}\sum_{X_i\in\tilde{\mathcal{B}}_k} Y_i - p\right)^2 \mathbb{1}_{N_n(\tilde{\mathcal{B}}_k)>0}\right] + \mathbb{E}\left[\mathbb{1}_{X\in\mathcal{W},X\in\tilde{\mathcal{B}}_k}\right] \\
&+ \frac{1}{2}\mathbb{E}\left[\left(\frac{1}{N_n(\tilde{\mathcal{W}}_k)}\sum_{X_i\in\tilde{\mathcal{W}}_k} Y_i - (1-p)\right)^2 \mathbb{1}_{N_n(\tilde{\mathcal{W}}_k)>0}\right] + \mathbb{E}\left[\mathbb{1}_{X\in\mathcal{B},X\in\tilde{\mathcal{W}}_k}\right]. \qquad (2.94)
\end{aligned}
$$

Let L be a generic cell. The thourth term in (2.94) can be upper-bounded as follows:

$$
\begin{aligned}
\mathbb{E}\left[\mathbb{1}_{X\in\mathcal{B},X\in\tilde{\mathcal{W}}_k}\right] &= \sum_{j=1}^{2^k}\mathbb{E}\left[\mathbb{1}_{X\in L_j}\mathbb{1}_{L_j\subset\tilde{\mathcal{W}}_k\cap\mathcal{B}}\right] &(2.95)\\
&= \sum_{j=1}^{2^k}\mathbb{E}\left[\mathbb{1}_{X\in L_j}\right]\mathbb{E}\left[\mathbb{1}_{L_j\subset\tilde{\mathcal{W}}_k\cap\mathcal{B}}\right] &(2.96)\\
&= \mathbb{P}\left(L_j\subset\tilde{\mathcal{W}}_k\cap\mathcal{B}\right). &(2.97)
\end{aligned}
$$

by symmetry. Now,

$$\begin{aligned}
&\mathbb{P}\left(L \subset \tilde{\mathcal{W}}_k \cap L \subset \mathcal{B}\right) \\
&= \mathbb{P}\left(\left(\frac{1}{N_n(L)} \sum_{X_i \in L} \mathbb{1}_{Y_i=0} > \frac{1}{2}\right) \bigcap (L \subset \mathcal{B})\right) && (2.98)\\
&\leqslant \mathbb{E}\left[\mathbb{P}\left(\left(\frac{1}{N_n(L)} \sum_{X_i \in L, L \subset \mathcal{B}} \mathbb{1}_{Y_i=0} - (1-p) \geqslant \frac{1}{2} - (1-p)\right) \bigcap (L \subset \mathcal{B})\right)\right] && (2.99)\\
&\leqslant \mathbb{E}\left[e^{-2N_n(L)(p-\frac{1}{2})^2}\right] && (2.100)\\
&\qquad \text{(according to Hoeffding's inequality)} \\
&= \prod_{i=1}^{n} \mathbb{E}\left[e^{-2(p-\frac{1}{2})^2 \mathbb{1}_{X_i \in L}}\right] && (2.101)\\
&\qquad \text{(by independence of } X_i\text{'s)} \\
&= \left(\frac{e^{-2(p-\frac{1}{2})^2}}{2^k} + 1 - \frac{1}{2^k}\right)^n. && (2.102)
\end{aligned}$$

Consequently,

$$\mathbb{E}\left[\mathbb{1}_{X \in \mathcal{B}, X \in \tilde{\mathcal{W}}_k}\right] \leqslant \left(\frac{e^{-2(p-\frac{1}{2})^2}}{2^k} + 1 - \frac{1}{2^k}\right)^n.$$

Similar calculations show that

$$\begin{aligned}
\mathbb{E}\left[\mathbb{1}_{X \in \mathcal{W}, X \in \tilde{\mathcal{B}}_k}\right] &= \mathbb{P}\left(L \subset \tilde{\mathcal{B}}_k \cap L \subset \mathcal{W}\right) \\
&\leqslant \left(\frac{e^{-2(p-\frac{1}{2})^2}}{2^k} + 1 - \frac{1}{2^k}\right)^n. && (2.103)
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\mathbb{E}\left[(f_{k,1,n}(X) - f^\star(X))^2)\right] \\
&\leqslant \frac{1}{2}\mathbb{E}\left[\left(\frac{1}{N_n(\tilde{\mathcal{B}}_k)} \sum_{X_i \in \tilde{\mathcal{B}}_k} Y_i - p\right)^2 \mathbb{1}_{N_n(\tilde{\mathcal{B}}_k)>0}\right] \\
&+ \frac{1}{2}\mathbb{E}\left[\left(\frac{1}{N_n(\tilde{\mathcal{W}}_k)} \sum_{X_i \in \tilde{\mathcal{W}}_k} Y_i - (1-p)\right)^2 \mathbb{1}_{N_n(\tilde{\mathcal{W}}_k)>0}\right] \\
&\quad + \left(\frac{e^{-2(p-\frac{1}{2})^2}}{2^k} + 1 - \frac{1}{2^k}\right)^n + \frac{p^2 + (1-p)^2}{2}\left(1 - \frac{1}{2^k}\right)^n. && (2.104)
\end{aligned}$$

Now, the first term in (2.104) can be written as

$$\mathbb{E}\left[\left(\frac{1}{N_n(\tilde{\mathcal{B}}_k)}\sum_{X_i\in\tilde{\mathcal{B}}_k} Y_i - p\right)^2 \mathbb{1}_{N_n(\tilde{\mathcal{B}}_k)>0}\right] \tag{2.105}$$

$$= \mathbb{E}\left[\left(\frac{1}{N_n(\tilde{\mathcal{B}}_k)}\sum_{X_i\in\tilde{\mathcal{B}}_k} Y_i - p\right)^2 \mathbb{1}_{N_n(\tilde{\mathcal{B}}_k)>0}\mathbb{1}_{\mathcal{B}=\tilde{\mathcal{B}}_k}\right]$$

$$+ \mathbb{E}\left[\left(\frac{1}{N_n(\tilde{\mathcal{B}}_k)}\sum_{X_i\in\tilde{\mathcal{B}}_k} Y_i - p\right)^2 \mathbb{1}_{N_n(\tilde{\mathcal{B}}_k)>0}\mathbb{1}_{\mathcal{B}\neq\tilde{\mathcal{B}}_k}\right] \tag{2.106}$$

$$\leqslant \mathbb{E}\left[\left(\frac{1}{N_n(\tilde{\mathcal{B}}_k)}\sum_{X_i\in\tilde{\mathcal{B}}_k} Y_i - p\right)^2 \mathbb{1}_{N_n(\tilde{\mathcal{B}}_k)>0}\mathbb{1}_{\mathcal{B}=\tilde{\mathcal{B}}_k}\right] + \mathbb{P}\left(\mathcal{B}\neq\tilde{\mathcal{B}}_k\right) \tag{2.107}$$

Now, using a union bound, we obtain

$$\mathbb{P}\left(\mathcal{B}\neq\tilde{\mathcal{B}}_k\right) \leqslant \sum_{L_j\subset\mathcal{B}} \mathbb{P}\left(L_j\not\subset\tilde{\mathcal{B}}_k\right) + \sum_{L_j\subset\mathcal{W}} \mathbb{P}\left(L_j\subset\tilde{\mathcal{B}}_k\right) \tag{2.108}$$

$$\leqslant 2^k\cdot\mathbb{P}\left(L\not\subset\tilde{\mathcal{B}}_k\cap L\subset\mathcal{B}\right) + 2^k\cdot\mathbb{P}\left(L\subset\tilde{\mathcal{B}}_k\cap L\subset\mathcal{W}\right) \tag{2.109}$$

$$\leqslant 2^{k+1}\left(\frac{e^{-2(p-\frac{1}{2})^2}}{2^k} + 1 - \frac{1}{2^k}\right)^n, \tag{2.110}$$

according to (2.102) and (2.103). Additionally, the left term in (2.107) satisfies

$$\mathbb{E}\left[\left(\frac{1}{N_n(\tilde{\mathcal{B}}_k)}\sum_{X_i\in\tilde{\mathcal{B}}_k} Y_i - p\right)^2 \mathbb{1}_{N_n(\tilde{\mathcal{B}}_k)>0}\mathbb{1}_{\mathcal{B}=\tilde{\mathcal{B}}_k}\right] \leqslant \mathbb{E}\left[\left(\frac{1}{N_n(\mathcal{B})}\sum_{X_i\in\mathcal{B}} Y_i - p\right)^2 \mathbb{1}_{N_n(\mathcal{B})>0}\right] \tag{2.111}$$

$$\leqslant \mathbb{E}\left[\frac{\mathbb{1}_{N_n(\mathcal{B})>0}}{N_n(\mathcal{B})^2}\left(\sum_{X_i\in\mathcal{B}} Y_i - pN_n(\mathcal{B})\right)^2\right] \tag{2.112}$$

$$= p(1-p)\mathbb{E}\left[\frac{\mathbb{1}_{N_n(\mathcal{B})>0}}{N_n(\mathcal{B})}\right], \tag{2.113}$$

noticing that the square term of (2.112) is nothing but the conditional variance of a binomial distribution $B(N_n(\mathcal{B}), p)$. By Lemma S1 (i) on $N_n(\mathcal{B})$ which is a binomial random variable $B(n,p)$ with $p = 1/2$ (exactly half of the cells are black),

$$\mathbb{E}\left[\left(\frac{1}{N_n(\tilde{\mathcal{B}}_k)}\sum_{X_i\in\tilde{\mathcal{B}}_k} Y_i - p\right)^2 \mathbb{1}_{N_n(\tilde{\mathcal{B}}_k)>0}\mathbb{1}_{N_n(\tilde{\mathcal{B}}_k)>0}\right] \leqslant \frac{2p(1-p)}{n+1}.$$

Hence

$$\mathbb{E}\left[\left(\frac{1}{N_n(\tilde{\mathcal{B}}_k)}\sum_{X_i\in\tilde{\mathcal{B}}_k} Y_i - p\right)^2 \mathbb{1}_{\mathcal{B}=\tilde{\mathcal{B}}_k}\right] \leqslant \frac{2p(1-p)}{n+1} + 2^{k+1}\left(\frac{e^{-2(p-\frac{1}{2})^2}}{2^k} + 1 - \frac{1}{2^k}\right)^n. \quad (2.114)$$

Similarly,

$$\mathbb{E}\left[\left(\frac{1}{N_n(\tilde{\mathcal{W}}_k)}\sum_{X_i\in\tilde{\mathcal{W}}_k} Y_i - (1-p)\right)^2 \mathbb{1}_{N_n(\tilde{\mathcal{W}}_k)>0}\right] \leqslant \frac{2p(1-p)}{n+1} + 2^{k+1}\left(\frac{e^{-2(p-\frac{1}{2})^2}}{2^k} + 1 - \frac{1}{2^k}\right)^n. \quad (2.115)$$

Finally, injecting (2.114) and (2.115) into (2.104), we finally get

$$\begin{aligned}\mathbb{E}\left[(f_{k,1,n}(X) - f^\star(X))^2)\right] \leqslant \frac{p^2+(1-p)^2}{2}\left(1-\frac{1}{2^k}\right)^n + 2^{k+1}\cdot\left(\frac{e^{-2(p-\frac{1}{2})^2}}{2^k} + 1 - \frac{1}{2^k}\right)^n \\ + \frac{2p(1-p)}{n+1} + \left(\frac{e^{-2(p-\frac{1}{2})^2}}{2^k} + 1 - \frac{1}{2^k}\right)^n,\end{aligned}$$

which concludes this part of the proof.

Proof of 2. (Lower Bound)

We have

$$\mathbb{E}\left[(f_{k,1,n}(X) - f^\star(X))^2)\right] = \mathbb{E}\left[(f_{k,1,n}(X) - f^\star(X))^2 \mathbb{1}_{N_n(L_n(X))>0}\right] + \left(\frac{p^2+(1-p)^2}{2}\right)\left(1-\frac{1}{2^k}\right)^n,$$

where

$$\begin{aligned}&\mathbb{E}\left[(f_{k,1,n}(X) - f^\star(X))^2 \mathbb{1}_{N_n(L_n(X))>0}\right] \\ &\geqslant \mathbb{E}\left[\left(\frac{1}{N_n(\tilde{\mathcal{B}}_k)}\sum_{X_i\in\tilde{\mathcal{B}}_k} Y_i - p\right)^2 \mathbb{1}_{X\in\mathcal{B}}\mathbb{1}_{X\in\tilde{\mathcal{B}}_k}\mathbb{1}_{N_n(\tilde{\mathcal{B}}_k)>0}\mathbb{1}_{\mathcal{B}=\tilde{\mathcal{B}}_k}\right] \\ &\quad + \mathbb{E}\left[\left(\frac{1}{N_n(\tilde{\mathcal{W}}_k)}\sum_{X_i\in\tilde{\mathcal{W}}_k} Y_i - (1-p)\right)^2 \mathbb{1}_{X\in\mathcal{W}}\mathbb{1}_{X\in\tilde{\mathcal{W}}_k}\mathbb{1}_{N_n(\tilde{\mathcal{W}}_k)>0}\mathbb{1}_{\mathcal{W}=\tilde{\mathcal{W}}_k}\right] \\ &\geqslant \mathbb{P}(X\in\mathcal{B})\,\mathbb{E}\left[\left(\frac{1}{N_n(\tilde{\mathcal{B}}_k)}\sum_{X_i\in\tilde{\mathcal{B}}_k} Y_i - p\right)^2 \mathbb{1}_{\mathcal{B}=\tilde{\mathcal{B}}_k}\mathbb{1}_{N_n(\tilde{\mathcal{B}}_k)>0}\right] \\ &\quad + \mathbb{P}(X\in\mathcal{W})\,\mathbb{E}\left[\left(\frac{1}{N_n(\tilde{\mathcal{W}}_k)}\sum_{X_i\in\tilde{\mathcal{W}}_k} Y_i - (1-p)\right)^2 \mathbb{1}_{\mathcal{W}=\tilde{\mathcal{W}}_k}\mathbb{1}_{N_n(\tilde{\mathcal{W}}_k)>0}\right]. \quad (2.116)\end{aligned}$$

The first expectation term line (2.116) can be written as

$$\begin{aligned}
&\mathbb{E}\left[\left(\frac{1}{N_n(\tilde{\mathcal{B}}_k)}\sum_{X_i\in\tilde{\mathcal{B}}_k} Y_i - p\right)^2 \mathbb{1}_{\mathcal{B}=\tilde{\mathcal{B}}_k}\mathbb{1}_{N_n(\tilde{\mathcal{B}}_k)>0}\right] \\
&= \mathbb{P}\left(\mathcal{B}=\tilde{\mathcal{B}}_k\right)\mathbb{E}\left[\left(\frac{1}{N_n(\mathcal{B})}\sum_{X_i\in\mathcal{B}} Y_i - p\right)^2 |\mathcal{B}=\tilde{\mathcal{B}}_k\right] \qquad (2.117)
\end{aligned}$$

According to (2.110),

$$\mathbb{P}\left(\mathcal{B}=\tilde{\mathcal{B}}_k\right) \geqslant 1-2^k\cdot\left(1+\frac{e^{-2(p-\frac{1}{2})^2}-1}{2^k}\right)^n. \qquad (2.118)$$

Similarly,

$$\mathbb{P}\left(\mathcal{W}=\tilde{\mathcal{W}}_k\right) \geqslant 1-2^k\cdot\left(1+\frac{e^{-2(p-\frac{1}{2})^2}-1}{2^k}\right)^n.$$

Furthermore,

$$\begin{aligned}
&\mathbb{E}\left[\left(\frac{1}{N_n(\mathcal{B})}\sum_{X_i\in\mathcal{B}} Y_i - p\right)^2 \middle| \mathcal{B}=\tilde{\mathcal{B}}_k\right] \\
&= \mathbb{E}\left[\frac{1}{N_n(\mathcal{B})^2}\mathbb{E}\left[\left(\sum_{X_i\in\mathcal{B}} Y_i - N_n(\mathcal{B})p\right)^2 \middle| N_1,\ldots N_{2^k}, \mathcal{B}=\tilde{\mathcal{B}}_k\right] \middle| \mathcal{B}=\tilde{\mathcal{B}}_k\right] \qquad (2.119)
\end{aligned}$$

where we let $Z = \sum_{X_i \in \mathcal{B}} Y_i$. A typical bias-variance decomposition yields

$$\mathbb{E}\left[\left(\sum_{X_i \in \mathcal{B}} Y_i - N_n(\mathcal{B})p\right)^2 \middle| N_1, ...N_{2^k}, \mathcal{B} = \tilde{\mathcal{B}}_k\right] \tag{2.120}$$

$$= \mathbb{E}\left[\left(Z - \mathbb{E}\left[Z \mid N_1, ...N_{2^k}, \tilde{\mathcal{B}}_k = \mathcal{B}\right]\right)^2\right.$$

$$\left. + \left(\mathbb{E}\left[Z \mid N_1, ...N_{2^k}, \tilde{\mathcal{B}}_k = \mathcal{B}\right] - N_n(\mathcal{B})p\right)^2 \mid N_1, ...N_{2^k}, \tilde{\mathcal{B}}_k = \mathcal{B}\right] \tag{2.121}$$

$$\geqslant \mathbb{E}\left[\left(Z - \mathbb{E}\left[Z \mid N_1, ...N_{2^k}, \tilde{\mathcal{B}}_k = \mathcal{B}\right]\right)^2 \mid N_1, ...N_{2^k}, \tilde{\mathcal{B}}_k = \mathcal{B}\right]$$

$$= \mathbb{E}\left[\left(\sum_{L_j \subset \mathcal{B}} Z_j - \mathbb{E}\left[Z_j \mid N_j, L_j \subset \tilde{\mathcal{B}}_k\right]\right)^2 \mid N_1, ...N_{2^k}, \tilde{\mathcal{B}}_k = \mathcal{B}\right] \tag{2.122}$$

$$= \sum_{L_j \subset \mathcal{B}} \mathbb{E}\left[\left(Z_j - \mathbb{E}\left[Z_j \mid N_j, L_j \subset \tilde{\mathcal{B}}_k\right]\right)^2 \mid N_j, L_j \subset \tilde{\mathcal{B}}_k\right]$$

$$+ 2 \sum_{L_i, L_j \subset \mathcal{B}, L_i \neq L_j} \mathbb{E}\left[\left(Z_i - \mathbb{E}\left[Z_i \mid N_i, L_i \subset \tilde{\mathcal{B}}_k\right]\right)\left(Z_j - \mathbb{E}\left[Z_j \mid N_j, L_j \subset \tilde{\mathcal{B}}_k\right]\right) \mid N_1, ...N_{2^k}, \tilde{\mathcal{B}}_k = \mathcal{B}\right] \tag{2.123}$$

$$= \sum_{L_j \subset \mathcal{B}} \mathbb{E}\left[\left(Z_j - \mathbb{E}\left[Z_j \mid N_j, L_j \subset \tilde{\mathcal{B}}_k\right]\right)^2 \mid N_j, L_j \subset \tilde{\mathcal{B}}_k\right]. \tag{2.124}$$

with $Z_j = \sum_{X_i \in L_j} Y_i$, and $L_1, \ldots, L_{2^k}$ the leaves of the first layer tree. Note that $Z_j | N_j, L_j \subset \mathcal{B}$ are i.i.d binomial variable $\mathfrak{B}(N_j, p)$. In (2.122) and (2.123), we used that that given a single leaf $L_j \subset \mathcal{B}$, $\mathbb{E}\left[Z_j \mid N_1, ...N_{2^k}, \tilde{\mathcal{B}}_k = \mathcal{B}\right] = \mathbb{E}\left[Z_j \mid N_j, L_j \subset \tilde{\mathcal{B}}_k\right]$. To obtain (2.124), we used that conditional to $N_1, ...N_{2^k}, \tilde{\mathcal{B}}_k = \mathcal{B}$, Z_i and Z_j are independent. Therefore the double sum equals 0.

Let j be an integer in $\{1, ..., 2^k\}$,

$$\mathbb{E}\left[\left(Z_j - \mathbb{E}\left[Z_j \mid N_j, L_j \subset \tilde{\mathcal{B}}_k\right]\right)^2 \mid N_j, L_j \subset \tilde{\mathcal{B}}_k\right] \tag{2.125}$$

$$= \mathbb{E}\left[Z_j^2 \mid N_j, L_j \subset \tilde{\mathcal{B}}_k\right] - \mathbb{E}\left[Z_j \mid N_j, L_j \subset \tilde{\mathcal{B}}_k\right]^2 \tag{2.126}$$

$$\geqslant \mathbb{E}\left[Z_j^2 \mid N_j\right] - \mathbb{E}\left[Z_j \mid N_j, L_j \subset \tilde{\mathcal{B}}_k\right]^2 \tag{2.127}$$

$$= N_j p(1-p) + N_j^2 p^2 - \left(N_j p + \frac{N_j}{2}(1-p) \frac{\mathbb{P}\left(Z_j = \frac{N_j}{2} \mid N_j\right)}{\sum_{i=\frac{N_j}{2}}^{N_j} \mathbb{P}\left(Z_j = i\right)} \right)^2 \tag{2.128}$$

$$\geqslant N_j(1-p)\left(p - N_j(1-p)\mathbb{P}\left(Z_j = \frac{N_j}{2} \mid N_j\right)^2 - 2N_j p \cdot \mathbb{P}\left(Z_j = \frac{N_j}{2} \mid N_j\right)\right) \tag{2.129}$$

$$\geqslant N_j(1-p)\left(p - \frac{N_j(1-p)}{\pi\left(\frac{N_j}{2} + \frac{1}{4}\right)}\left(4p(1-p)\right)^{N_j} - \frac{2N_j}{\sqrt{\pi\left(\frac{N_j}{2} + \frac{1}{4}\right)}}\left(4p(1-p)\right)^{N_j/2}\right) \tag{2.130}$$

$$\geqslant N_j p(1-p) - \left(\frac{2(1-p)^2}{\pi} + 2\sqrt{2}(1-p)\right) \cdot N_j^{3/2} \cdot \left(4p(1-p)\right)^{N_j/2}. \tag{2.131}$$

We deduced Line (2.127) from the fact that Z_j^2 is a positive random variable, (2.128) from Lemma (S1) (5), Line (2.129) from the fact that $p > 1/2$ and Line (2.130) from the inequality (2.4) on the binomial coefficient. Injecting (2.123) and (2.131) into (2.119) yields

$$\mathbb{E}\left[\left(\frac{1}{N_n(\mathcal{B})}\sum_{X_i \in \mathcal{B}} Y_i - p\right)^2 \middle| \mathcal{B} = \tilde{\mathcal{B}}_k\right]$$

$$\geqslant \mathbb{E}\left[\frac{1}{N_n(\mathcal{B}_k)^2}\sum_{L_j \subset \mathcal{B}}\left(N_j p(1-p) - \left(\frac{2(1-p)^2}{\pi} + 2\sqrt{2}(1-p)\right) \cdot N_j^{3/2} \cdot \left(4p(1-p)\right)^{N_j/2}\right) \middle| \mathcal{B} = \tilde{\mathcal{B}}_k\right]$$

$$\geqslant \mathbb{E}\left[\frac{p(1-p)}{N_n(\mathcal{B})} \mid \mathcal{B} = \tilde{\mathcal{B}}_k\right] - \left(\frac{2(1-p)^2}{\pi} + 2\right)\sum_{L_j \subset \mathcal{B}} \mathbb{E}\left[\left(4p(1-p)\right)^{N_j/2} \mid \mathcal{B} = \tilde{\mathcal{B}}_k\right] \tag{2.132}$$

$$\geqslant p(1-p)\mathbb{E}\left[\frac{1}{N_n(\mathcal{B})} \mid \mathcal{B} = \tilde{\mathcal{B}}_k\right] - 3 \cdot 2^{k-1}\mathbb{E}\left[\left(4p(1-p)\right)^{N_b/2} \mid \mathcal{B} = \tilde{\mathcal{B}}_k\right] \tag{2.133}$$

where the last inequality relies on the fact that the $N_j, L_j \subset \mathcal{B}$ are i.i.d, with $b \in 1, ..., 2^k$ be the index of a cell included in $\mathcal{B}$. N_j is a binomial random variable $\mathfrak{B}(n, 2^{-k})$.

$$\mathbb{E}\left[(4p(1-p))^{N_j/2} \mid \mathcal{B} = \tilde{\mathcal{B}}_k\right] \leqslant \mathbb{E}\left[(4p(1-p))^{N_j/2}\right] \frac{1}{\mathbb{P}\left(\mathcal{B} = \tilde{\mathcal{B}}_k\right)} \tag{2.134}$$

$$= \left(\sqrt{4p(1-p)} \cdot 2^{-k} + (1 - 2^{-k})\right)^n \frac{1}{\mathbb{P}\left(\mathcal{B} = \tilde{\mathcal{B}}_k\right)}. \tag{2.135}$$

From the inequality Line (2.118), we deduce that as soon as $n \geqslant \frac{(k+1)\log(2)}{\log(2^k) - \log(e^{-2(p-1/2)^2} - 1 + 2^k)}$,

$$\frac{1}{\mathbb{P}\left(\mathcal{B} = \tilde{\mathcal{B}}_k\right)} \leqslant 2. \tag{2.136}$$

Therefore,

$$\mathbb{E}\left[(4p(1-p))^{N_j/2} \mid \mathcal{B} = \tilde{\mathcal{B}}_k\right] \leqslant 2\left(\sqrt{4p(1-p)} \cdot 2^{-k} + (1 - 2^{-k})\right)^n. \tag{2.137}$$

Moreover,

$$\mathbb{E}\left[\frac{1}{N_n(\mathcal{B})} | \mathcal{B} = \tilde{\mathcal{B}}_k\right] \geqslant \frac{1}{\mathbb{E}\left[N_n(\mathcal{B}) | \mathcal{B} = \tilde{\mathcal{B}}_k\right]} \tag{2.138}$$

$$\geqslant \frac{\mathbb{P}\left(\mathcal{B} = \tilde{\mathcal{B}}_k\right)}{\mathbb{E}\left[N_n(\mathcal{B})\right]} \tag{2.139}$$

$$\geqslant \frac{2}{n} - \frac{2^{k+1}}{n}\left(1 + \frac{e^{-2(p-\frac{1}{2})^2} - 1}{2^k}\right)^n \tag{2.140}$$

where the last inequality comes from the probability bound line (2.118) and the fact that $N_n(\mathcal{B})$ is a binomial random variable $\mathfrak{B}(n, 1/2)$.

Finally,

$$\mathbb{E}\left[\left(\frac{1}{N_n(\mathcal{B})} \sum_{X_i \in \mathcal{B}} Y_i - p\right)^2 | \mathcal{B} = \tilde{\mathcal{B}}_k\right] \tag{2.141}$$

$$\geqslant \frac{2p(1-p)}{n} - 3 \cdot 2^k \left(1 - 2^{-k}\left(1 - \sqrt{4p(1-p)}\right)\right)^n - \frac{2^{k+1}p(1-p)}{n}\left(1 + \frac{e^{-2(p-\frac{1}{2})^2} - 1}{2^k}\right)^n. \tag{2.142}$$

Similarly, regarding the second term of (2.116), note that $\mathbb{P}\left(\tilde{\mathcal{B}}_k = \mathcal{B}\right) = \mathbb{P}\left(\tilde{\mathcal{W}}_k = \mathcal{W}\right)$ and

$$\begin{aligned}
&\mathbb{E}\left[\left(\sum_{X_i\in\mathcal{W}} Y_i - N_n(\mathcal{W})(1-p)\right)^2 \middle| N_n(\mathcal{W}), \mathcal{W} = \tilde{\mathcal{W}}_k\right] \\
&= \mathbb{E}\left[\left(\sum_{X_i\in\mathcal{W}} \mathbb{1}_{Y_i=0} - N_n(\mathcal{W})p\right)^2 \middle| N_n(\mathcal{W}), \mathcal{W} = \tilde{\mathcal{W}}_k\right].
\end{aligned}$$

Thus we can adapt the above computation to this term :

$$\begin{aligned}
&\mathbb{E}\left[\left(\frac{1}{N_n(\mathcal{W})}\sum_{X_i\in\mathcal{W}} Y_i - p\right)^2 \middle| \mathcal{W} = \tilde{\mathcal{W}}_k\right] \qquad (2.143)\\
&\geqslant \frac{2p(1-p)}{n} - 3\cdot 2^k\left(1 - 2^{-k}\left(1-\sqrt{4p(1-p)}\right)\right)^n - \frac{2^{k+1}p(1-p)}{n}\left(1 + \frac{e^{-2(p-\frac{1}{2})^2}-1}{2^k}\right)^n.
\end{aligned}$$

Rearranging all terms proves the result :

$$\begin{aligned}
&\mathbb{E}\left[(f_{k,1,n}(X) - f^\star(X))^2\right] \geqslant \left(\frac{2p(1-p)}{n} - 2^{k+2}\cdot\left(1 - 2^{-k}\left(1-\sqrt{4p(1-p)}\right)\right)^n\right.\\
&\left. - \frac{2^{k+1}p(1-p)}{n}\cdot\left(1 + \frac{e^{-2(p-\frac{1}{2})^2}-1}{2^k}\right)^n\right)\left(1 - 2^k\cdot\left(1 + \frac{e^{-2(p-\frac{1}{2})^2}-1}{2^k}\right)^n\right)\\
&+ \frac{p^2+(1-p)^2}{2}\left(1-\frac{1}{2^k}\right)^n\\
&\geqslant \frac{2p(1-p)}{n} - 2^{k+2}\cdot\left(1 - 2^{-k}\left(1-\sqrt{4p(1-p)}\right)\right)^n - \frac{2^{k+1}p(1-p)}{n}\cdot\left(1 + \frac{e^{-2(p-\frac{1}{2})^2}-1}{2^k}\right)^n\\
&- \frac{2^{k+1}p(1-p)}{n}\cdot\left(1 + \frac{e^{-2(p-\frac{1}{2})^2}-1}{2^k}\right)^n + \frac{p^2+(1-p)^2}{2}\left(1-\frac{1}{2^k}\right)^n\\
&\geqslant \frac{2p(1-p)}{n} - 2^{k+2}\cdot\left(1 - 2^{-k}\left(1-\sqrt{4p(1-p)}\right)\right)^n - \frac{2^{k+2}p(1-p)}{n}\cdot\left(1 - \frac{1-e^{-2(p-\frac{1}{2})^2}}{2^k}\right)^n\\
&+ \frac{p^2+(1-p)^2}{2}\left(1-\frac{1}{2^k}\right)^n\\
&\geqslant \frac{2p(1-p)}{n} - \frac{2^{k+3}\cdot(1-\rho_{k,p})^n}{n} + \frac{p^2+(1-p)^2}{2}\left(1-\frac{1}{2^k}\right)^n
\end{aligned}$$

where

$$\rho_{k,p} = 2^{-k}\min\left(1-\sqrt{4p(1-p)}, 1 - e^{-2(p-\frac{1}{2})^2}\right).$$

Note that, since $p > 1/2$, $0 < \rho_{k,p} < 1$.

Lemma S1. *Let S be a positive random variable. For any real-valued $\alpha \in [0,1]$, for any $n \in \mathbb{N}$,*

$$\mathbb{P}\left(S \leqslant \alpha n\right) \mathbb{V}[S | S \leqslant \alpha n] \leqslant \mathbb{V}[S]$$

Proof. We start by noticing that:

$$\begin{aligned} A_n &= \mathbb{P}\left(S > \alpha n\right) \mathbb{E}\left[\left(S - \mathbb{E}\left[S \mid S > \alpha n\right]\right)^2 \mid S > \alpha n\right] \\ &\quad + \mathbb{P}\left(S \leqslant \alpha n\right) \mathbb{E}\left[\left(S - \mathbb{E}\left[S \mid S \leqslant \alpha n\right]\right)^2 \mid S \leqslant \alpha n\right] \\ &\leqslant \mathbb{P}\left(S > \alpha n\right) \mathbb{E}\left[\left(S - a\right)^2 \mid S > \alpha n\right] + \mathbb{P}\left(S \leqslant \alpha n\right) \mathbb{E}\left[\left(S - b\right)^2 \mid S \leqslant \alpha n\right] \end{aligned}$$

for any $(a, b) \in \mathbb{R}^2$.

Then,

$$\begin{aligned} A_n &\leqslant \mathbb{P}\left(S > \alpha n\right) \mathbb{E}\left[\left(S - a\right)^2 \mid S > \alpha n\right] + \mathbb{P}\left(S \leqslant \alpha n\right) \mathbb{E}\left[\left(S - a\right)^2 \mid S \leqslant \alpha n\right] \\ &= \mathbb{E}\left[\left(S - a\right)^2\right] \end{aligned}$$

for any $a \in \mathbb{R}$.

Choosing $a = \mathbb{E}\left[S\right]$, we obtain

$$A_n \leqslant \mathbb{V}[S].$$

Therefore,

$$\mathbb{P}\left(S \leqslant \alpha n\right) \mathbb{V}[S \mid S \leqslant \alpha n] \leqslant \mathbb{V}[S].$$

□

S7 Extended Results for a Random Chessboard

Proposition S1 (Risk of a single tree and a shallow tree network when $k < k^\star$)**.** *Let $N \in \{1, ..., 2^{k^\star}\}$. We consider the data distribution defined by a random chessboard with i.i.d. cells such that for each cell $C_i, i \in \{1, ..., 2^{k^\star}\}$*

$$\mathbb{P}\left(C_i \subset \mathcal{B}\right) = \frac{N}{2^{k^\star}}$$

and $\mathbb{P}\left(C_i \subset \mathcal{W}\right) = 1 - \frac{N}{2^{k^\star}}$. Notice that the (random) numbers $N_{\mathcal{W}}$ and $N_{\mathcal{B}}$ of white and black cells satisfy $0 \leqslant N_{\mathcal{W}} = 2^{k^\star} - N_{\mathcal{B}} \leqslant 2^{k^\star}$. We study the risk of the shallow tree network $f_{k,1,n}$.

1. *Consider a single tree $f_{k,0,n}$ of depth $k \in \mathbb{N}^\star$,*

$$\begin{aligned} \mathcal{R}(f_{k,0,n}) \leqslant 4(p - \frac{1}{2})^2 \frac{N}{2^{k^\star}} \left(1 - \frac{N}{2^{k^\star}}\right) \left(1 + \frac{1}{2^{k^\star - k}}\right) + \frac{2^{k-1}}{n+1} \\ + \left((1-p)^2 - \frac{N}{2^{k^\star}}(1-2p)\right)(1 - 2^{-k})^n \end{aligned}$$

and

$$\mathcal{R}(f_{k,0,n}) \geqslant 4(p - \frac{1}{2})^2 \frac{N}{2^{k^\star}} \left(1 - \frac{N}{2^{k^\star}}\right) + \frac{2^k}{n+1}(1-p)^2 + C_{k^\star,k,N,p}(1 - 2^{-k})^n$$

where $C_{k^\star,k,N,p} = (1-p)^2 - \frac{N}{2^{k^\star}}(1-2p) - \frac{(1-p)^2 2^k}{n+1} - 4(p-\frac{1}{2})^2 \frac{N}{2^{k^\star}} \left(1 - \frac{N}{2^{k^\star}}\right) \left(1 + \frac{1}{2^{k^\star-k}}\right)$.

2. *Consider the shallow tree network* $f_{k,1,n}$*, in the infinite sample regime,*

$$\mathcal{R}(f_{k,1,n}) \geqslant \left(p - \frac{1}{2}\right)^2 \min\left(1 - \frac{N}{2^{k^\star}}, \frac{N}{2^{k^\star}}\right)^2$$

and

$$\mathcal{R}(f_{k,1,n}) \leqslant 4\left(p - \frac{1}{2}\right)^2 \left(1 - \frac{N}{2^{k^\star}}\right) \frac{N}{2^{k^\star}} + p^2 \min\left(\frac{N}{2^{k^\star}}, 1 - \frac{N}{2^{k^\star}}\right).$$

S8 Proof of Proposition S1

S8.1 First Statement: Risk of a Single Tree

To see the definitions of $\tilde{\mathcal{B}}$ and $\tilde{\mathcal{W}}$ refer to the notations of the second statement of the proof of Proposition 2.4.5, in Appendix S6.2.

Recall that $k < k^\star$, meaning that a tree leaf may contain black and white cells. If a cell is empty, the tree prediction in this cell is set (arbitrarily) to zero. Thus,

$$\begin{aligned}
&\mathbb{E}\left[(f_{k,0,n}(X) - f^\star(X))^2\right] \\
&= \mathbb{E}\left[(f_{k,0,n}(X) - f^\star(X))^2 \mathbb{1}_{N_n(L_n(X))>0}\right] + \mathbb{E}\left[(f^\star(X))^2 \mathbb{1}_{N_n(L_n(X))=0}\right] \qquad (2.144) \\
&= \mathbb{E}\left[\left(\frac{1}{N_n(L_n(X))} \sum_{X_i \in L_n(X)} Y_i - f^\star(X)\right)^2 \mathbb{1}_{N_n(L_n(X))>0}\right] + \mathbb{E}\left[(f^\star(X))^2 \mathbb{1}_{N_n(L_n(X))=0}\right], \qquad (2.145)
\end{aligned}$$

where the expectation is taken over the distribution of the chessboard, (X, Y) and the dataset $(X_i, Y_i)_{1 \leqslant i \leqslant n}$. Besides,

$$\begin{aligned}
\mathbb{E}\left[(f^\star(X))^2 \mathbb{1}_{N_n(L_n(X))=0}\right] &= \mathbb{E}\left[(f^\star(X))^2 \mathbb{1}_{N_n(L_n(X))=0} \mathbb{1}_{X \in \mathcal{B}}\right] + \mathbb{E}\left[(f^\star(X))^2 \mathbb{1}_{N_n(L_n(X))=0} \mathbb{1}_{X \in \mathcal{W}}\right] \\
&= ((1-p)^2 - \frac{N_{\mathcal{B}}}{2^{k^\star}}(1-2p))(1-2^{-k})^n \qquad (2.146)
\end{aligned}$$

We now study the first term in (2.145), by considering that X falls into $\mathcal{B}$ (the same computation holds when X falls into $\mathcal{W}$). We denote $|L_n(X) \cap \mathcal{B}||$ (resp. $|L_n(X) \cap \mathcal{W}|$) the number of black (resp. white) cells included in the cell containing X. Letting (X', Y') generic random variables

with the same distribution as (X, Y), one has

$$\mathbb{E}\left[\left(\frac{1}{N_n(L_n(X))}\sum_{X_i\in L_n(X)} Y_i - p\right)^2 \mathbb{1}_{N_n(L_n(X))>0}\mathbb{1}_{X\in\mathcal{B}}\right] \tag{2.147}$$

$$= \mathbb{P}(X\in\mathcal{B})\,\mathbb{E}\left[\left(\frac{1}{N_n(L_n(X))}\sum_{X_i\in L_n(X)}\left(Y_i - \mathbb{E}\left[Y'\middle|X'\in L_n(X), |L_n(X)\cap\mathcal{B}|\right]\right)\right)^2 \mathbb{1}_{N_n(L_n(X))>0}\right]$$
$$+ \mathbb{P}(X\in\mathcal{B})\,\mathbb{E}\left[\left(\mathbb{E}\left[Y'\middle|X'\in L_n(X), |L_n(X)\cap\mathcal{B}|\right] - p\right)^2 \mathbb{1}_{N_n(L_n(X))>0}\right] \tag{2.148}$$

$$= \mathbb{P}(X\in\mathcal{B})\cdot$$
$$\mathbb{E}\left[\frac{\mathbb{1}_{N_n(L_n(X))>0}}{N_n(L_n(X))^2}\mathbb{E}\left[\left(\sum_{X_i\in L_n(X)}\left(Y_i - \mathbb{E}\left[Y'\middle|X'\in L_n(X), |L_n(X)\cap\mathcal{B}|\right]\right)\right)^2 \mid N_n(L_n(X)), |L_n(X)\cap\mathcal{B}|\right]\right]$$
$$+ \mathbb{P}(X\in\mathcal{B})\,\mathbb{P}(N_n(L_n(X))>0)\,\mathbb{E}\left[\left(\mathbb{E}\left[Y'\middle|X'\in L_n(X), |L_n(X)\cap\mathcal{B}|\right] - p\right)^2\right]$$
$$= \mathbb{P}(X\in\mathcal{B})\left(\mathbb{E}\left[\frac{\mathbb{1}_{N_n(L_n(X))>0}}{N_n(L_n(X))^2}\cdot V_{\mathcal{B}}\right] + \beta_{\mathcal{B}}\right) \tag{2.149}$$

where

$$\beta_{\mathcal{B}} = \mathbb{P}(N_n(L_n(X))>0)\,\mathbb{E}\left[\left(\mathbb{E}\left[Y'\middle|X'\in L_n(X), |L_n(X)\cap\mathcal{B}|\right] - p\right)^2\right] \tag{2.150}$$

and

$$V_{\mathcal{B}} = \mathbb{E}\left[\left(\sum_{X_i\in L_n(X)}\left(Y_i - \mathbb{E}\left[Y'\middle|X'\in L_n(X), |L_n(X)\cap\mathcal{B}|\right]\right)\right)^2 \mid N_n(L_n(X)), |L_n(X)\cap\mathcal{B}|\right] \tag{2.151}$$

Similarly we define β_W and $V_{\mathcal{W}}$ by replacing in the expressions (2.150) and (2.151) $\mathcal{B}$ by $\mathcal{W}$ so that:

$$\mathbb{E}\left[(f_{k,0,n}(X) - f^\star(X))^2\right]$$
$$= \mathbb{P}(X\in\mathcal{B})\left(\mathbb{E}\left[\frac{\mathbb{1}_{N_n(L_n(X))>0}}{N_n(L_n(X))^2}\cdot V_{\mathcal{B}}\right] + \beta_{\mathcal{B}}\right) + \mathbb{P}(X\in\mathcal{W})\left(\mathbb{E}\left[\frac{\mathbb{1}_{N_n(L_n(X))>0}}{N_n(L_n(X))^2}\cdot V_{\mathcal{W}}\right] + \beta_{\mathcal{W}}\right)$$
$$+ ((1-p)^2 - \frac{N_{\mathcal{B}}}{2^{k^\star}}(1-2p))(1-2^{-k})^n. \tag{2.152}$$

This last expression can be read as a bias-variance decomposition. We already know the probability to be in a non-empty cell, see (2.146), then

$$\mathbb{P}(X\in\mathcal{B}) = \mathbb{E}\left[\mathbb{P}(X\in\mathcal{B}|N_{\mathcal{B}})\right] = \mathbb{E}\left[\frac{N_{\mathcal{B}}}{2^{k^\star}}\right] = \frac{N}{2^{k^\star}}.$$

We now make explicit the terms in Equation (2.152) starting with the bias term $\beta_{\mathcal{B}}$:

$$\mathbb{E}\left[Y'\middle|X' \in L_n(X), |L_n(X)\cap\mathcal{B}|\right] = \frac{p\cdot|L_n(X)\cap\mathcal{B}| + (1-p)|L_n(X)\cap\mathcal{W}|}{2^{k^\star-k}} \tag{2.153}$$

$$= (1-p) + \frac{|L_n(X)\cap\mathcal{B}|}{2^{k^\star-k}}(2p-1) \tag{2.154}$$

$$= p + \frac{|L_n(X)\cap\mathcal{W}|}{2^{k^\star-k}}(1-2p), \tag{2.155}$$

where $|L_n(X)\cap\mathcal{B}|$ stands for the number of black cells in $L_n(X)$. In the same way, $|L_n(X)\cap\mathcal{W}|$ stands for the number of white cells in $L_n(X)$. Hence,

$$\mathbb{E}\left[\left(\mathbb{E}\left[Y'\middle|X' \in L_n(X), |L_n(X)\cap\mathcal{B}|\right] - p\right)^2\right] = \frac{4(p-\frac{1}{2})^2}{2^{2(k^\star-k)}}\mathbb{E}\left[|L_n(X)\cap\mathcal{W}|^2\right] \tag{2.156}$$

Note that $|L_n(X)\cap\mathcal{W}||N_{\mathcal{B}} \sim \mathfrak{B}(2^{k^\star-k}, 1-N/2^{k^\star})$. Thus, we have

$$\mathbb{E}\left[|L_n(X)\cap\mathcal{W}|^2\right] = \frac{N}{2^k}\left(1-\frac{N}{2^{k^\star}}\right) + \frac{1}{2^{2k}}(2^{k^\star}-N)^2. \tag{2.157}$$

Therefore,

$$\mathbb{E}\left[\left(\mathbb{E}\left[Y'\middle|X' \in L_n(X), |L_n(X)\cap\mathcal{B}|\right] - p\right)^2\right] = \frac{4(p-\frac{1}{2})^2}{2^{2(k^\star-k)}}\left(\frac{N}{2^k}\left(1-\frac{N}{2^{k^\star}}\right) + \frac{1}{2^{2k}}(2^{k^\star}-N)^2\right). \tag{2.158}$$

Similar computations show that when $X \in \mathcal{W}$,

$$\mathbb{E}\left[\left(\mathbb{E}\left[Y'\middle|X' \in L_n(X), |L_n(X)\cap\mathcal{W}|\right] - (1-p)\right)^2\right] = \frac{4(p-\frac{1}{2})^2}{2^{2(k^\star-k)}}\left(\frac{N}{2^k}\left(1-\frac{N}{2^{k^\star}}\right) + \frac{N^2}{2^{2k}}\right). \tag{2.159}$$

We deduce from Equations (2.158) and (2.159) that

$$\begin{aligned}\beta_{\mathcal{B}} + \beta_{\mathcal{W}} = \frac{4(p-\frac{1}{2})^2}{2^{2(k^\star-k)}}&\mathbb{P}\left(N_n(L_n(X)) > 0\right)\left(\mathbb{P}\left(X\in\mathcal{B}\right)\left(\frac{N}{2^k}\left(1-\frac{N}{2^{k^\star}}\right) + \frac{1}{2^{2k}}(2^{k^\star}-N)^2\right)\right.\\ &\left. + \mathbb{P}\left(X\in\mathcal{W}\right)\left(\frac{N}{2^k}\left(1-\frac{N}{2^{k^\star}}\right) + \frac{N^2}{2^{2k}}\right)\right)\\ = 4(p-\frac{1}{2})^2&\frac{N}{2^{k^\star}}\left(1-\frac{N}{2^{k^\star}}\right)\left(1+\frac{1}{2^{k^\star-k}}\right)\left(1-(1-2^{-k})^n\right).\end{aligned} \tag{2.160}$$

Clearly,

$$\beta_{\mathcal{B}} + \beta_{\mathcal{W}} \geqslant 4(p-\frac{1}{2})^2\frac{N}{2^{k^\star}}\left(1-\frac{N}{2^{k^\star}}\right)\left(1-(1-2^{-k})^n\right). \tag{2.161}$$

Now we compute the variance term $V_{\mathcal{B}}$. Letting $Z = \sum_{X_i \in L_n(X)} Y_i$,

$$Z | N_n(L_n(X)), |L_n(X) \cap \mathcal{B}| \sim \mathfrak{B}(N_n(L_n(X)), p')$$

where $p' = (1-p) + \frac{|L_n(X) \cap \mathcal{B}|}{2^{k^\star - k}}(2p-1)$ (see Equations (2.153) to (2.155)). Therefore, recall that $V_{\mathcal{B}}$ is nothing but the variance of the binomial random variable Z conditional on $|L_n(X) \cap \mathcal{B}|$ defined in Equation (2.151), consequently

$$V_{\mathcal{B}} = N_n(L_n(X)) p'(1-p'). \tag{2.162}$$

By independence of $N_n(L_n(X))$ and $|L_n(X) \cap \mathcal{B}|$, we can write that

$$\mathbb{E}\left[\frac{\mathbb{1}_{N_n(L_n(X))>0}}{N_n(L_n(X))^2} \cdot V_{\mathcal{B}}\right] = \mathbb{E}\left[\frac{\mathbb{1}_{N_n(L_n(X))>0}}{N_n(L_n(X))}\right] \mathbb{E}\left[\underbrace{p'(1-p)'}_{(1-p)^2 \leqslant p'(1-p') \leqslant 1/4}\right]. \tag{2.163}$$

From Technical Lemma S1, we deduce that

$$\frac{2^k}{n+1}\left(1-(1-2^{-k})^n\right) \leqslant \mathbb{E}\left[\frac{\mathbb{1}_{N_n(L_n(X))>0}}{N_n(L_n(X))}\right] \leqslant \frac{2^{k+1}}{n+1}.$$

Hence,

$$\frac{2^k}{n+1}\left(1-(1-2^{-k})^n\right)(1-p)^2 \leqslant \mathbb{E}\left[\frac{\mathbb{1}_{N_n(L_n(X))>0}}{N_n(L_n(X))} V_{\mathcal{B}}\right] \leqslant \frac{2^{k-1}}{n+1}. \tag{2.164}$$

By symmetry, $V_{\mathcal{W}}$ is also the variance of a binomial random variable with parameters $N_n(L_n(X))$, $1-p'$ conditional on $|L_n(X) \cap \mathcal{W}|$. Thus $V_{\mathcal{B}} = V_{\mathcal{W}}$. To conclude, combining Equations (2.152), (2.161) and (2.164) leads to

$$\mathcal{R}(f_{k,0,n}(X)) \leqslant 4(p-\frac{1}{2})^2\left(1-\frac{N}{2^{k^\star}}(1-\frac{N}{2^{k^\star}})\right) + \frac{2^{k-1}}{n+1} + ((1-p)^2 - \frac{N}{2^{k^\star}}(1-2p))(1-2^{-k})^n$$

and

$$\begin{aligned}\mathcal{R}(f_{k,0,n}(X)) \geqslant\ & 4(p-\frac{1}{2})^2\left(1-\frac{N}{2^{k^\star}}(2-\frac{N}{2^{k^\star}})\right)\frac{2^k}{n+1}\left(1-(1-2^{-k})^n\right)(1-p)^2 \\ & + ((1-p)^2 - \frac{N}{2^{k^\star}}(1-2p))(1-2^{-k})^n.\end{aligned}$$

S8.2 Second Statement: Risk of a Shallow Tree Network

Recall that we are in the infinite sample regime and that $k < k^\star$.

$$\begin{aligned}&\mathbb{E}\left[(f_{k,1,n}(X) - f^\star(X)^2\right] \\ &= \mathbb{E}\left[(f_{k,1,n}(X)-p)^2 \mathbb{1}_{X \in \mathcal{B} \cap \tilde{\mathcal{B}}}\right] + \mathbb{E}\left[(f_{k,1,n}(X)-(1-p))^2 \mathbb{1}_{X \in \mathcal{W} \cap \tilde{\mathcal{W}}}\right] \\ &\quad + \mathbb{E}\left[(f_{k,1,n}(X)-(1-p))^2 \mathbb{1}_{X \in \mathcal{W} \cap \tilde{\mathcal{B}}}\right] + \mathbb{E}\left[(f_{k,1,n}(X)-p)^2 \mathbb{1}_{X \in \mathcal{B} \cap \tilde{\mathcal{W}}}\right].\end{aligned} \tag{2.165}$$

We begin with the computation of the first term.

$$\mathbb{E}\left[(f_{k,1,n}(X)-p)^2 \mathbb{1}_{X\in\mathcal{B}\cap\tilde{\mathcal{B}}}\right] = \mathbb{P}\left(X\in\mathcal{B}\cap\tilde{\mathcal{B}}\right)\mathbb{E}\left[(f_{k,1,n}(X)-p)^2 \mid X\in\mathcal{B}\cap\tilde{\mathcal{B}}\right] \tag{2.166}$$

$$= \mathbb{P}\left(X\in\mathcal{B}\cap\tilde{\mathcal{B}}\right)\mathbb{E}\left[\left(\mathbb{E}\left[Y' \mid X'\in\tilde{\mathcal{B}}\right]-p\right)^2 \mid X\in\mathcal{B}\cap\tilde{\mathcal{B}}\right]. \tag{2.167}$$

Regarding the probability term,

$$\mathbb{P}\left(X\in\mathcal{B}\cap\tilde{\mathcal{B}}\right) = \mathbb{P}\left(X\in\mathcal{B}\right)\mathbb{P}\left(X\in\tilde{\mathcal{B}} \mid X\in\mathcal{B}\right) \tag{2.168}$$

$$\leqslant \frac{N}{2^{k^\star}}. \tag{2.169}$$

We denote by $B_1, ..., B_{2^k}$ the number of black cells in the leaves $L_1, ..., L_{2^k}$. Then,

$$\begin{aligned}
\mathbb{E}\left[Y' \mid X'\in\tilde{\mathcal{B}}\right] &= \mathbb{E}\left[(1-p)+(2p-1)\frac{\sum_{i=1}^{2^k} B_i \mathbb{1}_{L_i\subset\tilde{\mathcal{B}}}}{|\tilde{\mathcal{B}}|}\right] \\
&= (1-p)+(2p-1)\mathbb{E}\left[\sum_{i=1}^{2^k}\frac{\mathbb{1}_{L_i\subset\tilde{\mathcal{B}}}}{|\tilde{\mathcal{B}}|}\mathbb{E}\left[B_i\middle||\tilde{\mathcal{B}}|\right]\right] \\
&= (1-p)+(2p-1)\mathbb{E}\left[\frac{B_j}{2^{k^\star-k}} \mid B_j \geqslant \frac{|L_j|}{2}\right]
\end{aligned}$$

where L_j is a leaf included in $\tilde{\mathcal{B}}$. Moreover,

$$\begin{aligned}
\mathbb{E}\left[B_j \mid B_j \geqslant \frac{|L_j|}{2}\right] &= \frac{N}{2^k}+\left(1-\frac{N}{2^{k^\star}}\right)\left(2^{k^\star-k-1}-1\right)\frac{\mathbb{P}\left(B_j = 2^{k^\star-k-1}-1\right)}{\mathbb{P}\left(B_j \geqslant 2^{k^\star-k-1}-1\right)} \\
&\leqslant \frac{N}{2^k}+\left(1-\frac{N}{2^{k^\star}}\right)2^{k^\star-k-1}.
\end{aligned}$$

Therefore,

$$\mathbb{E}\left[Y' \mid X'\in\tilde{\mathcal{B}}\right] \leqslant (1-p)+(2p-1)\frac{1}{2}\left(1+\frac{N}{2^{k^\star}}\right)$$

and

$$\mathbb{E}\left[\left(\mathbb{E}\left[Y' \mid X'\in\tilde{\mathcal{B}}\right]-p\right)^2 \mid X\in\mathcal{B}\cap\tilde{\mathcal{B}}\right] \geqslant (p-\frac{1}{2})^2\left(1-\frac{N}{2^{k^\star}}\right)^2. \tag{2.170}$$

To compute the upper bound, note that

$$\mathbb{E}\left[B_j \mid B_j \geqslant \frac{|L_j|}{2}\right] \geqslant \mathbb{E}\left[B_j\right] = \frac{N}{2^k}.$$

Thus,

$$\mathbb{E}\left[\left(\mathbb{E}\left[Y' \mid X' \in \tilde{\mathcal{B}}\right] - p\right)^2 \mid X \in \mathcal{B} \cap \tilde{\mathcal{B}}\right] \leqslant 4\left(p - \frac{1}{2}\right)^2 \left(1 - \frac{N}{2^{k^\star}}\right)^2. \tag{2.171}$$

We adapt the previous computations to the term $\mathbb{E}\left[(f_{k,1,n}(X) - (1-p))^2 \mathbb{1}_{X \in \mathcal{W} \cap \tilde{\mathcal{W}}}\right]$ from Equation (2.165). We have

$$\mathbb{E}\left[(f_{k,1,n}(X) - (1-p))^2 \mid X \in \mathcal{W} \cap \tilde{\mathcal{W}}\right] \geqslant (p - \frac{1}{2})^2 \frac{N^2}{2^{2k^\star}} \tag{2.172}$$

and

$$\mathbb{E}\left[(f_{k,1,n}(X) - (1-p))^2 \mid X \in \mathcal{W} \cap \tilde{\mathcal{W}}\right] \leqslant 4\left(p - \frac{1}{2}\right)^2 \frac{N^2}{2^{2k^\star}} \tag{2.173}$$

Moreover, note that

$$\mathbb{E}\left[(f_{k,1,n}(X) - p)^2 \mathbb{1}_{X \in \mathcal{B} \cap \tilde{\mathcal{W}}}\right] \leqslant p^2 \mathbb{P}\left(X \in \mathcal{B} \cap \tilde{\mathcal{W}}\right) \tag{2.174}$$

and

$$\mathbb{E}\left[(f_{k,1,n}(X) - p)^2 \mathbb{1}_{X \in \mathcal{B} \cap \tilde{\mathcal{W}}}\right] \geqslant \left(p - \frac{1}{2}\right)^2 \mathbb{P}\left(X \in \mathcal{B} \cap \tilde{\mathcal{W}}\right). \tag{2.175}$$

Similarly,

$$\mathbb{E}\left[(f_{k,1,n}(X) - p)^2 \mathbb{1}_{X \in \mathcal{W} \cap \tilde{\mathcal{B}}}\right] \leqslant p^2 \mathbb{P}\left(X \in \mathcal{W} \cap \tilde{\mathcal{B}}\right) \tag{2.176}$$

and

$$\mathbb{E}\left[(f_{k,1,n}(X) - p)^2 \mathbb{1}_{X \in \mathcal{W} \cap \tilde{\mathcal{B}}}\right] \geqslant \left(p - \frac{1}{2}\right)^2 \mathbb{P}\left(X \in \mathcal{W} \cap \tilde{\mathcal{B}}\right). \tag{2.177}$$

Gathering Equation (2.165) and Equations (2.170) to (2.177) yields

$$\begin{aligned}
\mathbb{E}\left[(f_{k,1,n}(X) - f^\star(X)^2\right] &\geqslant (p - \frac{1}{2})^2 \left(1 - \frac{N}{2^{k^\star}}\right)^2 \mathbb{P}\left(X \in \mathcal{B} \cap \tilde{\mathcal{B}}\right) + (p - \frac{1}{2})^2 \frac{N^2}{2^{2k^\star}} \mathbb{P}\left(X \in \mathcal{W} \cap \tilde{\mathcal{W}}\right) \\
&\quad + \left(p - \frac{1}{2}\right)^2 \mathbb{P}\left(X \in \mathcal{W} \cap \tilde{\mathcal{B}}\right) + \left(p - \frac{1}{2}\right)^2 \mathbb{P}\left(X \in \mathcal{B} \cap \tilde{\mathcal{W}}\right) \\
&\geqslant \left(p - \frac{1}{2}\right)^2 \min\left(1 - \frac{N}{2^{k^\star}}, \frac{N}{2^{k^\star}}\right)^2
\end{aligned}$$

as well as

$$
\begin{aligned}
&\mathbb{E}\left[(f_{k,1,n}(X) - f^\star(X)^2\right] \\
&\leqslant 4\left(p-\frac{1}{2}\right)^2\left(1-\frac{N}{2^{k^\star}}\right)^2\mathbb{P}\left(X\in\mathcal{B}\cap\tilde{\mathcal{B}}\right) + 4\left(p-\frac{1}{2}\right)^2\frac{N^2}{2^{2k^\star}}\mathbb{P}\left(X\in\mathcal{W}\cap\tilde{\mathcal{W}}\right) \\
&\quad + p^2\mathbb{P}\left(X\in\mathcal{W}\cap\tilde{\mathcal{B}}\right) + p^2\mathbb{P}\left(X\in\mathcal{B}\cap\tilde{\mathcal{W}}\right) \\
&\leqslant 4\left(p-\frac{1}{2}\right)^2\left(1-\frac{N}{2^{k^\star}}\right)^2\frac{N}{2^{k^\star}}\mathbb{P}\left(X\in\tilde{\mathcal{B}}\mid X\in\mathcal{B}\right) \\
&+ 4\left(p-\frac{1}{2}\right)^2\frac{N^2}{2^{2k^\star}}\left(1-\frac{N}{2^{k^\star}}\right)\mathbb{P}\left(X\in\tilde{\mathcal{W}}\mid X\in\mathcal{W}\right) \\
&\quad + p^2\left(1-\frac{N}{2^{k^\star}}\right)\mathbb{P}\left(X\in\tilde{\mathcal{B}}\mid X\in\mathcal{W}\right) + p^2\frac{N}{2^{k^\star}}\mathbb{P}\left(X\in\tilde{\mathcal{W}}\mid X\in\mathcal{B}\right) \\
&\leqslant 4\left(p-\frac{1}{2}\right)^2\left(1-\frac{N}{2^{k^\star}}\right)^2\frac{N}{2^{k^\star}} + 4\left(p-\frac{1}{2}\right)^2\frac{N^2}{2^{2k^\star}}\left(1-\frac{N}{2^{k^\star}}\right) \\
&\quad + p^2\left(1-\frac{N}{2^{k^\star}}\right)\mathbb{P}\left(X\in\tilde{\mathcal{B}}\right) + p^2\frac{N}{2^{k^\star}}\mathbb{P}\left(X\in\tilde{\mathcal{W}}\right) \\
&\leqslant 4\left(p-\frac{1}{2}\right)^2\left(1-\frac{N}{2^{k^\star}}\right)\frac{N}{2^{k^\star}} + p^2\max\left(\frac{N}{2^{k^\star}}1-\frac{N}{2^{k^\star}}\right).
\end{aligned}
$$

Bibliography of the current chapter

Bergstra, J. S., R. Bardenet, Y. Bengio, and K. Balázs (2011). "Algorithms for Hyper-Parameter Optimization". In: *Advances in Neural Information Processing Systems 24*. Ed. by J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger. Curran Associates, Inc., pp. 2546–2554.

Berrouachedi, A., R. Jaziri, and G. Bernard (2019a). "Deep Cascade of Extra Trees". In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, pp. 117–129.

— (2019b). "Deep Extremely Randomized Trees". In: *International Conference on Neural Information Processing*. Springer, pp. 717–729.

Biau, G. (2012a). "Analysis of a random forests model". In: *The Journal of Machine Learning Research* 13.1, pp. 1063–1095.

Biau, G., L. Devroye, and G. Lugosi (2008). "Consistency of random forests and other averaging classifiers". In: *Journal of Machine Learning Research* 9.Sep, pp. 2015–2033.

Breiman, Leo (2001a). "Random forests". In: *Machine learning* 45.1, pp. 5–32.

Cribari-Neto, F., N. L. Garcia, and K. LP Vasconcellos (2000). "A note on inverse moments of binomial variates". In: *Brazilian Review of Econometrics* 20.2, pp. 269–277.

Fan, Wei, Haixun Wang, Philip S Yu, and Sheng Ma (2003). "Is random model better? on its accuracy and efficiency". In: *Third IEEE International Conference on Data Mining*. IEEE, pp. 51–58.

Feng, Ji, Yang Yu, and Zhi-Hua Zhou (2018). "Multi-layered gradient boosting decision trees". In: *Advances in neural information processing systems*, pp. 3551–3561.

Ghods, Alireza and Diane J Cook (2020). "A survey of deep network techniques all classifiers can adopt". In: *Data Mining and Knowledge Discovery*, pp. 1–42.

Ghosh, Soumyadip and Shane G Henderson (2002). "Chessboard distributions and random vectors with specified marginals and covariance matrix". In: *Operations Research* 50.5, pp. 820–834.

— (2009). "Patchwork distributions". In: *Advancing the Frontiers of Simulation*. Springer, pp. 65–86.

Guo, Yang, Shuhui Liu, Zhanhuai Li, and Xuequn Shang (2018). "BCDForest: a boosting cascade deep forest model towards the classification of cancer subtypes based on gene expression data". In: *BMC bioinformatics* 19.5, p. 118.

Jeong, Mira, Jaeyeal Nam, and Byoung Chul Ko (2020). "Lightweight Multilayer Random Forests for Monitoring Driver Emotional Status". In: *IEEE Access* 8, pp. 60344–60354.

Kim, S., M. Jeong, and B. C. Ko (2020). "Interpretation and Simplification of Deep Forest". In: *arXiv preprint arXiv:2001.04721*.

Klusowski, Jason M. (2018). "Sharp analysis of a simple model for random forests". In: *arXiv preprint arXiv:1805.02587*.

Liu, B. et al. (2020). "Morphological Attribute Profile Cube and Deep Random Forest for Small Sample Classification of Hyperspectral Image". In: *IEEE Access* 8, pp. 117096–117108.

Miller, Kevin, Chris Hettinger, Jeffrey Humpherys, Tyler Jarvis, and David Kartchner (May 2017). "Forward Thinking: Building Deep Random Forests". In.

Pang, M., K. Ting, P. Zhao, and Z. Zhou (2018). "Improving Deep Forest by Confidence Screening". In: *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 1194–1199.

Pedregosa, F. et al. (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Su, R., X. Liu, L. Wei, and Q. Zou (2019). "Deep-Resp-Forest: A deep forest model to predict anti-cancer drug response". In: *Methods* 166, pp. 91–102.

Sun, L. et al. (2020). "Adaptive Feature Selection Guided Deep Forest for COVID-19 Classification With Chest CT". In: *IEEE Journal of Biomedical and Health Informatics* 24.10, pp. 2798–2805.

Utkin, L. V and M. A Ryabinin (2017). "Discriminative metric learning with deep forest". In: *arXiv preprint arXiv:1705.09620*.

Utkin, L. V and K. D Zhuk (2020). "Improvement of the Deep Forest Classifier by a Set of Neural Networks". In: *Informatica* 44.1.

Zeng, X. et al. (2020). "Network-based prediction of drug–target interactions using an arbitrary-order proximity embedded deep forest". In: *Bioinformatics* 36.9, pp. 2805–2812.

Zhang, Y. et al. (2019). "Distributed deep forest and its application to automatic detection of cash-out fraud". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 10.5, pp. 1–19.

Zheng, S., Y. Song, T. Leung, and I. Goodfellow (2016). "Improving the robustness of deep neural networks via stability training". In: *Proceedings of the ieee conference on computer vision and pattern recognition*, pp. 4480–4488.

Zhou, Z and J. Feng (2017). "Deep Forest: Towards An Alternative to Deep Neural Networks". In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 3553–3559.

Chapter 3

Tree Sparse NN Initialization

Abstract

Dedicated neural network (NN) architectures have been designed to handle specific data types (such as CNN for images or RNN for text), which ranks them among state-of-the-art methods for dealing with these data. Unfortunately, no architecture has been found for dealing with tabular data yet, for which tree ensemble methods (tree boosting, random forests) usually show the best predictive performances. In this work, we propose a new sparse initialization technique for (potentially deep) multilayer perceptrons (MLP): we first train a tree-based procedure to detect feature interactions and use the resulting information to initialize the network, which is subsequently trained via standard gradient descent (GD) strategies. Numerical experiments on several tabular data sets show the benefits of this new, simple and easy-to-use method, both in terms of generalization capacity and computation time, compared to default MLP initialization and even to existing complex deep learning solutions. In fact, this wise MLP initialization raises the performances of the resulting NN methods to that of gradient boosting on tabular data. Besides, such initializations are able to preserve the sparsity of weights introduced in the first layers of the network throughout the training, which emphasizes that the first layers act as a sparse feature extractor (like convolutional layers in CNN).

3.1 Introduction

Neural networks are now widely used in many domains of machine learning, in particular when dealing with very structured data. They indeed provide state-of-the-art performances for applications with images or text. However, neural networks still perform poorly on tabular inputs, for which tree ensemble methods remain the gold standards (Grinsztajn et al. 2022). The goal of this paper is to improve the performances of the former by using the strengths of the latter.

Tree ensemble methods Tree-based methods are widely used in the ML community, especially for processing tabular data. Two main approaches exist depending on whether the tree building process is parallel (e.g. Random Forest, RF, see Breiman 2001a) or sequential (e.g. Gradient Boosting Decision Trees, GBDT, see Friedman 2001). In these tree ensemble procedures, the final prediction relies on averaging predictions of randomized decision trees, coding for particular partitions of the input space. The two most successful and most widely used implementations of

these methods are XGBoost and LightGBM (see Chen et al. 2016; Ke et al. 2017) which both rely on the sequential GBDT approach.

Neural networks Neural Networks (NN) are efficient methods to unveil the patterns of spatial or temporal data, such as images (Krizhevsky et al. 2012) or texts (Liu et al. 2016). Their performance results notably from the fact that several architectures directly encode relevant structures in the input: convolutional neural networks (CNN, LeCun et al. 1995) use convolutions to detect spatially-invariant patterns in images, and recurrent neural networks (RNN, Rumelhart et al. 1985) use a hidden temporal state to leverage the natural order of a text. However, a dedicated *natural* architecture has yet to be introduced to deal with tabular data. Indeed, designing such an architecture would require to detect and leverage the structure of the relations between variables, which is much easier for images or text (spatial or temporal correlation) than for tabular data (unconstrained covariance structure).

NN initialization and training In the absence of a suitable architecture for handling tabular data, the Multi-Layer Perceptron (MLP) architecture (Rumelhart et al. 1986) remains the obvious choice due to its generalist nature. Apart from the large number of parameters, one difficulty of MLP training arises from the non-convexity of the loss function (see, e.g., Sun 2020). In such situations, the initialization of the network parameters (weights and biases) are of the utmost importance, since it can influence both the optimization stability and the quality of the minimum found. Typically, such initializations are drawn according to independent uniform distributions with a variance decreasing w.r.t. the size of the layer (He et al. 2015). Therefore, one may wonder how to capitalize on methods that are inherently capable of recognizing patterns in tabular data (e.g., tree-based methods) to propose a new NN architecture suitable for tabular data and an initialization procedure that leads to faster convergence and better generalization performance.

3.1.1 Related Works

How MLP can be used to handle tabular data remains unclear, especially since a corresponding prior in the MLP architecture adapted to the correlations of the input is not obvious, to say the least. Indeed, none of the existing NN architectures can consistently match the performance of state-of-the-art tree-based predictors on tabular data (Shwartz-Ziv et al. 2022; Gorishniy et al. 2021; and in particular Table 2 in Borisov et al. 2021).

Self-attention architectures Specific NN architectures have been proposed to deal with tabular data. For example, TabNet (Arik et al. 2021) uses a sequential self-attention structure to detect relevant features and then applies several networks for prediction. SAINT (Somepalli et al. 2021), on the other hand, uses a two-dimensional attention structure (on both features and samples) organized in several layers to extract relevant information which is then fed to a classical MLP. These methods typically require a large amount of data, since the self-attention layers and the output network involve numerous MLP.

Trees and neural networks Several solutions have been proposed to leverage the correspondence between tree-based methods and NN, in order to develop more efficient models for processing tabular data. For example, TabNN (Ke et al. 2018) first trains a GBDT on the available data, then extracts a group of features per individual tree, compresses the resulting groups, and uses a tailored Recursive Encoder based on the structure of these groups (with an initialization based on the tree leaves). Therefore, TabNN employs pre-trained tree-based methods to design more efficient NN. Conversely, Sethi 1990 Brent 1991, and later Welbl 2014, Richmond et al. 2015

and Biau et al. 2019 propose to translate decision trees into very specific MLP (made of 3 layers) and use GD training to improve upon the original tree-based method. Such procedures can be seen as a way to relax and generalize the partition geometry produced by trees and their aggregation. To our knowledge, such translations have not been used to boost the training of *general* NN architectures.

3.1.2 Contributions

In this work, we propose a new method to initialize a potentially deep MLP for learning tasks with tabular data. Our method consists in first training a tree-based predictor (RF, GBDT or Deep Forest, see Section 3.2.1) and then using its translation into an MLP as initialization for the first two layers, the deeper ones being randomly initialized. With subsequent standard GD training, this procedure is shown to outperform the widely used uniform initialization of MLP -default initialization in Pytorch Paszke et al. 2019) as follows.

1. **Improved performances.** For tabular data, the predictive performances of the MLP after training are improved compared to MLP that use a random initialization. Our procedure also outperforms more complex deep learning procedures based on self-attention and is on par with classical tree-based methods (such as XGBoost).

2. **Faster optimization.** The optimization following a tree-based initialization is boosted in the sense that it enjoys a faster convergence towards a (better) empirical minimum: a tree-based initialization results in faster training of the MLP.

Initializing the first few layers of the MLP with the translation of the tree-based method and initializing randomly the deeper layers is the most successful initialization scheme that we experimented. This supports the idea that in our method, the (first) tree-based initialized layers act as relevant feature extractors that allow the MLP to detect correlations in the inputs. In this context, our approach is dedicated on improving the performance of standard MLP models; therefore it is conceptually different from pre-existing procedures also relying on the translation of tree-based models into NN: (Biau et al. 2019) aim at fine-tuning tree-based methods using a very specific neural network framework (made of only 3 layers). We, on the other hand, use tree-based methods to carefully initialize certain layers of a generic MLP, which is then substantially trained using standard GD strategies.

Outline In Section 3.2, we introduce the predictors in play and describe how tree-based methods can be translated into MLP.

The core of our analysis is contained in Section 3.3, where we describe in detail the MLP initialization process and provide extensive numerical evaluations showing the benefits of this method.

3.2 Equivalence Between Trees and MLP

Consider the classical setting of supervised learning in which we are given a set of input/output samples $\{(X_i, Y_i)\}_{i=1}^n$ drawn i.i.d. from some unknown joint distribution. Our goal is to construct a (MLP) function to predict the output from the input. To do so, we leverage the translation of tree-based methods into MLP.

3.2.1 Presentation of the Predictors in Play

Tree-based methods We consider three different tree ensemble methods: Random Forests (RF), Gradient Boosting Decision Trees (GBDT) and Deep Forests (DF). They all share the same base component: the Decision Tree (DT, see Breiman et al. 1984 for details). We call its terminal nodes *leaf nodes*, which correspond to the cells of the final tree partition. RF (Breiman 2001b) is a predictor consisting of a collection of independently trained and randomized trees. Its final prediction is made by averaging the predictions of all its DT in regression or by a majority vote in classification. GBDT (Friedman 2001) aims at minimizing a prediction loss function by successively aggregating DT that approximate the opposite gradient of that loss function (see Chen et al. 2016 for details on XGBoost). DF (Zhou et al. 2017) is a hybrid learning procedure in which random forests are used as elementary components (neurons) of a neural-network-like architecture (see Figure S5 and Appendix S1 for details).

Multilayer Perceptron (MLP) The multilayer perceptron is a predictor consisting of a composition of multiple affine functions, with (potentially different) nonlinear activation functions between them. Standard activation functions include, for instance, the rectified linear unit or the hyperbolic tangent. Deep MLP are a much richer class of predictors than tree-based methods which build simple partitions of the space and output piecewise constant predictions. Therefore, any of the tree-based models presented above can be approximated and in fact exactly rewritten as an MLP as follows.

3.2.2 An Exact Translation of Tree-Based Methods into MLP

From decision tree to 3-layer MLP Recall that a decision tree codes for a partition of the input space in as many parts as there are leaf nodes in the tree. Given an input x, we can identify the leaf where x falls by examining for each hyperplane of the partition whether x falls on the right or left side of the hyperplane. The prediction is then made by averaging the outputs of all the training samples falling into the leaf of x. A DT can be thus translated into a highly sparse 3-layer MLP:

1. The first layer contains a number of neurons equal to the number of hyperplanes in the partition, each neuron encoding by ± 1 whether x falls on the left or right side of the hyperplane.

2. The second layer contains a number of neurons equal to the number of leaves in the DT. Based on the first layer, it identifies in which leaf x falls and outputs a vector with a single 1 at the leaf position and -1 everywhere else.

3. The last layer contains a single output neuron that returns the tree prediction. Its weights encode the average output of all training samples for each leaf of the tree.

This procedure is explained in detail and formally in Biau et al. 2019 and in Appendix S2.

From RF/GBDT to 3-layer MLP Although RF and GBDT are constructed in different ways, they both average multiple DT predictions to give the final result. Thus, to translate a RF or a GBDT into an MLP, we simply turn each tree into a 3-layer MLP as described above, and concatenate all the obtained networks to form a wider 3-layer MLP. When concatenating, we set all weights between the MLP translations of the different trees to 0, since the trees do not interact with each other in predicting the target value for a new feature vector. The step in which the responses of the different trees are averaged can be combined with the third layer of the individual tree

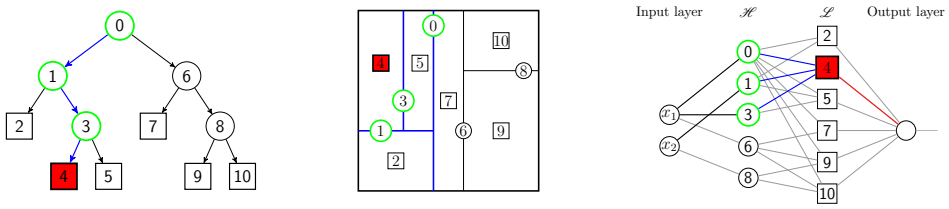


Figure 3.1: (from Biau et al. 2019). Illustration of a decision tree, its induced feature space partition and its corresponding MLP translation on a problem with 2 input variables.

translations, resulting in a final MLP translation with a total of three layers. Consider a RF with M trees of depth k, the resulting MLP will contain $M * (2^k - 1 + 2^k + 1)$ neurons which can be very wide if the depth of each tree is high.

From Deep Forests to deeper MLP A Deep Forest is a cascade of Random Forests. As such, it can be translated into an MLP containing the MLP translations of the different RF in cascade, resulting in a deeper and wider MLP (note that the obtained MLP has a number of layers that is a multiple of 3).Furthermore, in the Deep Forest architecture, the input vector is concatenated to the output of each intermediate layer. To mimic these skip connections in the MLP, we add additional neurons to each layer, except for the last three, which encode an identity mapping. Appendix S1 gives more insights into DF and their MLP translations. In particular, perfect translation of a DF suffers from numerical instabilities due to the replication of catastrophic cancellations (the deeper the DF, the greater their amplitude, cf Appendix S4). This does not impact the sequel of the study, which relies on MLP approximations introduced in Section 3.2.2.

3.2.3 Relaxing Tree-Based Translation to Allow Gradient Descent Training

As shown in the previous section, one can construct an MLP that exactly reproduces a tree-based predictor. However this translation involves (i) piecewise constant activation functions (sign) and (ii) different activation functions in a same layer (sign and identity when translating DF). These constraints can hinder the MLP training, which relies on GD strategies (requiring differentiability), as well as efficient implementation tricks, given that automatic differentiation libraries only support one activation function per layer. Therefore, given a pre-trained tree-based predictor (RF, GBDT or DF), we aim at relaxing its translation into a MLP, mimicking its behavior as closely as possible but in a compatible way with standard NN training.

From tree-based methods to differentiable MLP To do so, Welbl 2014; Biau et al. 2019 consider the differentiable $\tanh$ activation, well suited for approximating both the sign and identity functions. Indeed, this can be achieved by multiplying or dividing the output of a neuron by a large constant before applying the function $\tanh$ and rescaling the result accordingly if necessary, i.e. for large enough $a, c > 0$, $\text{sign}(x) \approx \tanh(ax)$ and $x \approx c \tanh\left(\frac{x}{c}\right)$.

However, we cannot choose a arbitrarily large as this would make gradients vanish during the network optimization (the function being flat on most of the space), and hinder training. We therefore introduce 4 hyper-parameters for the MLP encoding of any tree-based method that regulate the degree of approximation for the activation functions after the first, second and third layers of a decision tree translation, as well as for the identity mapping, respectively denoted by `strength01`, `strength12`, `strength23` and `strength_id`.

Hyperparameter choice The use of the tanh activation function involves extra hyper-parameters. We study the influence of each one, by making them vary in some range (keeping the others fixed to 10^{10}, resulting in an almost perfect approximation of the sign and identity functions), see Appendix S4.1 for details. Our analysis shows that increasing the hyperparameters beyond some limit value is no longer beneficial (as the activation functions are already perfectly approximated) and, across multiple data sets, these limit values are similar. We also exhibit relevant search spaces that will allow us to find optimal HP values for each application.

3.3 A New Initialization Method for MLP Training

In this section, we study the impact of tree-based initialization methods for MLP training when dealing with tabular data. The latter empirically proves to be always preferable to standard random initialization and makes MLP a competitive predictor for tabular data. Our code is publically available at `https://github.com/LutzPatrick/SparseTreeBasedInit`.

3.3.1 Our Proposal

Random initialization is the most common technique for initializing MLP prior to stochastic gradient training. It consists in setting all layer parameters to random values of small magnitude centered at 0. More precisely, all parameter values of the j-th layer are uniformly drawn in $[-1/\sqrt{d_j}, 1/\sqrt{d_j}]$ where d_j is the layer input dimension; this is the default behavior of most MLP implementations such as `nn.Linear` in PyTorch (Paszke et al. 2019).

We introduce new ways of initializing an MLP for learning with tabular data, by leveraging the recasting of tree-based methods in a neural network fashion:

- **RF/GBDT initialization.** First, a RF/GBDT is fitted to the training data and transformed into a 3-layer neural network, following the procedure described in Section 3.2. The first two layers of this network are used to initialize the first two layers of the network of interest. Thus, upon initialization, these first two layers encode the RF/GBDT partition. The parameters of the third and all subsequent layers are randomly initialized as described above. See Figure S7 in Appendix S3 for an illustration.
- **DF initialization.** Similarly as above, a Deep Forest (DF) using ℓ forest layers is first fitted to the training data. The first $3\ell - 1$ layers of the MLP are then initialized using the first $3\ell - 1$ layers of the MLP encoding of this pre-trained DF. The parameters of the 3ℓ-th and all subsequent layers are randomly initialized as explained above.

These tree-based initialization techniques may seem far-fetched at first glance, but they are actually consistent with recent approaches to adapting Deep Learning models for tabular data. The key to interpreting them is to think of the first (tree-based initialized) layers of the MLP as a feature extractor that produces an abstract representation of the input data (in fact, this is a vector encoding the tree-based predictor's space partition in which the observation lies). The subsequent randomly initialized layers, once trained, then perform the prediction task based on this abstract representation.

3.3.2 Experimental Setup

Datasets & learning tasks We compare prediction performances on a total of 10 datasets: 3 regression datasets (Airbnb, Diamonds and Housing), 5 binary classification datasets (Adult, Bank, Blastchar, Heloc, Higgs) and 2 multi-class classification datasets (Covertype and Volkert).

We mostly chose data sets that are used for benchmarking in relevant literature: Adult, Heloc, Housing, Higgs and Covertype are used by Borisov et al. 2021 and Bank, Blastchar and Volkert are used by Somepalli et al. 2021. Moreover, we add Airbnb and Diamonds to balance the different types of prediction tasks. The considered datasets are all medium-sized (10–60k observations) except for Covertype and Higgs (approx. 500k observations). Details about the datasets are given in Appendix S5.1.

Predictors We consider the following tree-based predictors: Random Forest (RF), Deep Forest (DF Zhou et al. 2019) and XGBoost (denoted by GBDT, Chen et al. 2016). The latter usually achieves state-of-the-art performances on tabular data sets (see, e.g., Shwartz-Ziv et al. 2022; Gorishniy et al. 2021; Borisov et al. 2021). We also consider deep learning approaches: MLP with default uniform initialization (MLP rand. init.) or tree-based initialization (resp. MLP RF init., MLP GBDT init. and MLP DF init.); and a transformer architecture SAINT Somepalli et al. 2021. This complex architecture is specifically designed for applications on tabular data and includes self-attention and inter-sample attention layers that extract feature correlations that are then passed on to an MLP. For regression and classification tasks, we use the mean-squared error (MSE) and cross-entropy loss for NN training, respectively. We choose SAINT as a baseline model as it is reported to outperform all other NN predictors on most of our data sets (all except Airbnb and Diamonds, see Borisov et al. 2021; Somepalli et al. 2021).

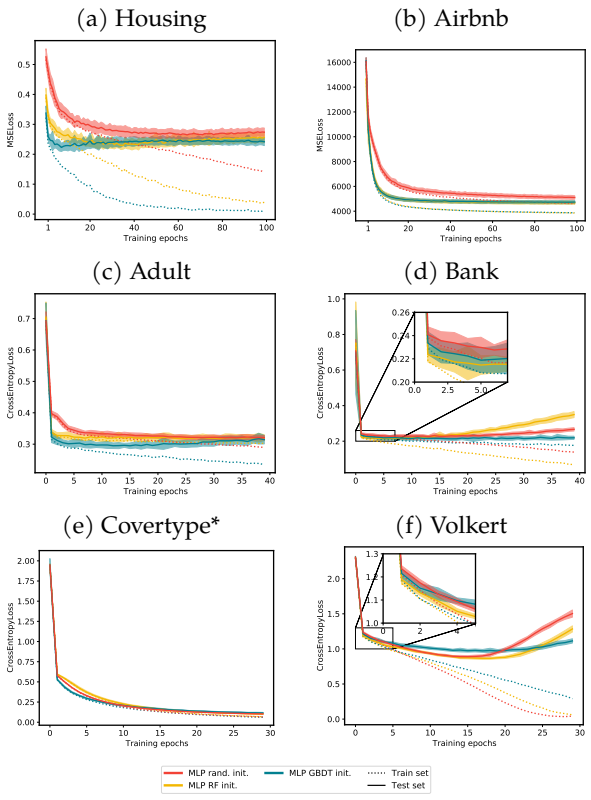


Figure 3.2: Optimization behaviour of randomly, RF and GBDT initialized MLP evaluated over a 5 times repeated (stratified) 5-fold of each data set, according to Protocol P1. The lines and shaded areas report the mean and standard deviation. *evaluation on a single 5-fold cross validation.

Parameter optimization All NN are trained using the Adam optimizer (Kingma et al. 2014). All hyper-parameters (HP) are determined empirically using the optuna library (Akiba et al. 2019) for Bayesian optimization.

For most HP, we use the default search spaces of Borisov et al. 2021. For all HP tuning the tree-to-MLP translation, we have identified relevant search spaces (see Appendix S4.1). An overview of all search spaces used for each method and the HP selected for experimental protocol P2 can be found in Appendix S5.5. The quantity minimized during HP tuning is the model's validation loss, and the smallest validation loss that occurred during training for MLP-based models.

3.3.3 A Better MLP Initialization for a Better Optimization

In this subsection, the optimization of standard MLP is shown to benefit from the proposed initialization technique. Experiments have been conducted on 6 out of the 10 data sets.

Experimental protocol 1 (P1) To obtain comparable optimization processes, we ensure that all MLP-related hyper-parameters (width, depth, learning rate), are identical for all the MLP regardless of the initialization scheme. These HP are chosen to maximize the predictive performance of the standard randomly initialized MLP. All HP related to the initialization technique (HP for the tree-based predictors and their translation) are optimized independently for each tree-based initialization.

Results Figure 3.2 shows that for most data sets, the use of tree-based initialization methods for MLP training provides a *faster convergence* towards *a better minimum* (in terms of generalization) than random initialization. This is all the more remarkable since Protocol P1 has been calibrated in favor of random initialization. Among tree-based initializers, GBDT initializations outperform or are on par with RF initializations in terms of the optimization behavior on all regression and binary classification problems. However, for multi-class classification problems, the advantages of tree-based initialization seem to be limited. This is probably due to the fact that the MLP architecture at play is tailored for random initialization, being thus too restrictive for tree-based initializers. Experiments presented in Appendix S5.3 with fixed arbitrary widths corroborate this idea: in this case, the RF initialization is beneficial for the optimization process. For the Adult, Bank, and Volkert data sets, Figure 3.2 also shows the performance of each method at initialization. None of these procedures leads to a better MLP performance at initialization (due to both the non-exact translation from trees to MLP and to the additional randomly initialized layers), but rather help guiding the MLP in its learning process.

3.3.4 A Better MLP Initialization for a Better Generalization

In this subsection, tree-based initialization methods are shown to systematically improve the predictive power of neural networks compared to random initialization. We compare our procedure to the predictors described in Section 3.3.2, but also to 3 other NN techniques: one close to the default uniform initialization (Xavier init., see Glorot et al. 2010), one using random orthogonal matrices (LUSV init., see Mishkin et al. 2015) and the winning ticket lottery strategy (WT prun., see Frankle et al. 2018), which is a pruning method used during training to end up with a sparse NN. The reader may refer to Appendix S5.4 for more details about these three techniques.

Experimental protocol 2 (P2) Each MLP is trained on 100 epochs, but with HP tuned depending on the initialization technique. For maximum comparability, the optimization budget is strictly the same for all methods (100 "optuna" iterations each, where one optuna iteration consists

Model \ Data set	Housing	Airbnb	Diamonds	Adult	Bank	Blastchar	Heloc	Higgs	Covertype	Volkert
	MSE ↓	MSE ↓ ($x10^3$)	MSE ↓ ($x10^{-3}$)	AUC ↑ (in %)	AUC ↑ (in %)	AUC ↑ (in %)	AUC ↑ (in %)	AUC ↑ (in %)	Acc. ↑ (in %)	Acc. ↑ (in %)
Random Forest	0.263±0.009	5.39±0.13	9.80±0.35	91.6±0.3	92.8±0.3	**84.5±1.2**	91.3±0.6	80.4±0.1	83.6±0.1	64.2±0.3
GBDT	**0.208±0.010**	**4.71±0.15**	**7.38±0.28**	**92.7±0.3**	**93.3±0.3**	**84.7±1.0**	**92.1±0.4**	82.8±0.1	**97.0±0.0**	71.3±0.4
Deep Forest	0.225±0.008	**4.68±0.16**	8.23±0.29	91.8±0.3	92.9±0.2	**83.7±1.2**	90.3±0.5	81.2±0.0*	92.4±0.1*	66.3±0.4
MLP rand. init.	0.258±0.011	5.07±0.16	15.5±12.5	90.5±0.4	91.0±0.3	81.4±1.2	80.1±0.1	83.2±0.3	96.7±0.0	72.2±0.4
MLP Xavier init.	0.263±0.012	5.05±0.17	12.4±6.19	90.5±0.5	90.8±0.5	81.7±1.3	79.9±1.1	82.8±0.1	96.8±0.0	72.1±0.4
MLP LUSV init.	0.295±0.018	4.99±0.14	14.1±5.00	90.5±0.5	90.2±0.5	**84.3±1.2**	79.9±0.9	81.7±0.1	96.5±0.0	70.8±0.5
MLP WT prun.	0.248±0.011	5.26±2.11	9.83±5.07	90.6±0.4	90.9±0.5	**84.4±1.2**	89.6±0.7	82.9±0.1	**97.0±0.0**	71.5±0.4
MLP RF init.	0.222±0.009	**4.66±0.16**	7.93±0.22	92.1±0.3	92.4±0.4	**84.4±1.2**	**91.7±0.4**	**83.6±0.1**	96.7±0.0	**74.1±0.4**
MLP GBDT init.	**0.206±0.007**	**4.70±0.09**	8.15±0.35	92.2±0.3	92.5±0.3	**84.6±1.2**	91.5±0.6	83.0±0.0	96.2±0.0	73.5±0.5
MLP DF init.	0.234±0.016	**4.81±0.13**	8.28±0.24	91.9±0.4	92.2±0.3	**84.2±1.0**	91.4±0.6	83.3±0.1*	94.5±0.3*	71.3±0.5
SAINT	0.258±0.011	**4.81±0.15**	17.7±3.83	91.6±0.3	92.2±0.4	**84.0±0.8**	90.2±0.7	**83.7±0.1***	96.6±0.1*	70.1±0.4

Table 3.1: Best scores and their standard deviations for Protocol 2. For each data set, predictors performing at least as well as the best over all (resp. best DL) score up to its standard deviation are highlighted in **bold** (resp. underlined). All scores are based on a 5 times repeated (stratified) 5-fold cross validation. For each model, HP have been chosen via the "optuna" library with 100 iterations. See Appendix S5.4 for a comparison with literature results. *score based on a simple 5-fold cross val.

of one hold-out validation). In particular, when using a tree-based initializer, we use 25 HP optimization iterations to find optimal HP for the tree-based predictor, fix these HPs, and then use the remaining 75 iterations to determine optimal HP for the MLP. For all NN approaches, the model with the best performance on the validation set during training is kept (using the classical *early-stopping* procedure). Performances are measured via the MSE for regression, the AUROC score (AUC) for binary classification and the accuracy (Acc.) for multi-class classification, averaging 5 runs of 5-fold cross-validation.

Results Table 3.1 shows that RF or GBDT initialization strictly outperform random initialization, in terms of final generalization performance, for all data sets except Covertype (for which performances are similar). They also systematically achieve better results than the LUSV and Xavier init. and are better on all but 2 datasets than the WT pruning procedure which is a more refined procedure. Additionally, the MLP using both RF and GBDT initialization techniques outperform SAINT on all medium-sized data sets and fall short on large data sets (Higgs and Covertype).

Despite its simplicity , the proposed method (based on RF or GBDT) is on par with GBDT on half of the data sets, ranking MLP as relevant predictors for tabular data. Note that the GBDT used for initialization of the MLP is way less powerful than the best one found here (see details in Tables S10 and S11). This shows that our procedure produces, with a relatively low initialization cost, powerful MLP relevant for tabular data. Among the tree-based initializers, RF is on par with or outperforms GBDT initialization on all data sets but Housing. DF initialization, for its part, cannot compete with RF and GBDT initialization, despite showing some improvement over the random one (except for Covertype and Volkert). This underlines that injecting prior information via tree-based methods into the first layers of a MLP is, among all the aforementioned methods, the best way to improve its performance.

The interested reader may find a comparison of the optimization procedures of all MLP methods and SAINT (Figure S13) and tables summarizing all HP (Tables S10 and S11) in Appendix S5.5. We remark that tree-based initializers generally bring into play wider networks with similar depths (fixed width of 2048 and adaptive depth between 4 and 10) compared to MLP with default initialization. Yet, for most data sets, the overall procedure is computationally more

efficient than state-of-the-art deep learning architectures like SAINT, both in terms of number of parameters and training time (see Tables S6-S8 in Appendix S5.4).

3.3.5 Analyzing Key Elements of the New Initialization Methods

Influence of the MLP width We mainly use standard search spaces from (Borisov et al. 2021) to determine the optimal hyperparameters for each model. However, the MLP width is an exception to this. The standard search spaces used in the literature usually involve MLP with a few hundred neurons per layer (e.g. up to 100 neurons in Borisov et al. 2021); yet, in this work, we consider MLP with a width up to 2048 neurons. Large MLP are actually very beneficial for tree-based initialization methods as they allow the use of more expressive tree-based models in the initialization step.

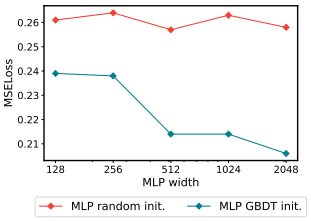


Figure 3.3: Influence of width on the generalization performance for random and GBDT initializations. Mean values over 5 times repeated 5-fold cross-validation on Housing.

Figure 3.3 compares the performance of an MLP with random/GDBT initializations and various widths. There is no gain in prediction by using wider (thus more complex) NN, when randomly initialized. This is corroborated by the results of Table S4: for all regression and binary classification data sets, the performance of our (potentially much wider) MLP with random initialization is consistently close to the literature values, and only increases for multi-class classification tasks. However, an MLP initialized with GBDT significantly benefits from enlarging the NN width (justifying a fixed width of 2048 for tree-based initialized MLP). This confirms the idea that tree-based initialization helps to reveal relevant features to the MLP, all the more as the width increases, and by doing so, boosts the MLP performance after training.

Performance of the initializer Another interesting step in unraveling the essence of the new initialization method is to understand which characteristics of a tree-based model are relevant to its success as an initializer. Undoubtedly, its predictive accuracy plays an important role, but does this aspect alone suffice to characterize the success of the new initialization method? Figure S14 compares the predictive performance of different RF/GBDT initializers and the performance of the respective MLP after training. As the figure illustrates, a better performance of the tree-based predictor used for initialization does not always lead to a better performance of the MLP after training (see Airbnb and Volkert). This observation suggests that other aspects, such as the expressiveness of the feature interactions captured by the initializer, the structure it induces on the MLP or the weight distributions of the initializer, must also play a significant role in the initialization method's success.

MLP sparsity Finally, we investigate the structure that tree-based initialization induces on the MLP *after* training. Figure 3.4 shows the weight distributions of the three first and the last layers before and after MLP training, for random, RF and GBDT initializations on Housing (see Appendix S5.7 for more data sets). It indicates that the weight distribution on the first two layers change significantly during training when the MLP is randomly initialized: the weights are uniformly distributed at epoch 0 but appear to be Gaussian after training. When RF or GBDT initializers are used instead, the weights of the first two layers are sparsely distributed at epoch 0 by construction, and their distribution is preserved during training (notice the logarithmic y-axis for these plots in Figure 3.4). Note that the (uniform) distribution of the weights in other layers is also preserved through training (third and last lines of Figure 3.4). This means that

our initialization technique, in combination with SGD optimization strategies, introduces an *implicit regularization* of the NN optimization: the sparse structure of the initialization (on first layers) is maintained. This is very similar to the CNN architecture (constrained by design), a very successful class of NN designed for image processing. Besides, the weight distributions are not squeezed towards zero during learning when sparse initialization is used, preventing poor generalization performances according to previous works (Neal 2012; Blundell et al. 2015).

Looking at Figure 3.4, one may get the impression that the weights in the first layers remain unchanged during GD training, and that ultimately no learning takes place in these layers. However, numerical experiments (see Appendix S5.4) show that the weights of *all* layers are modified during learning; the first two layers actually undergo the greatest changes.

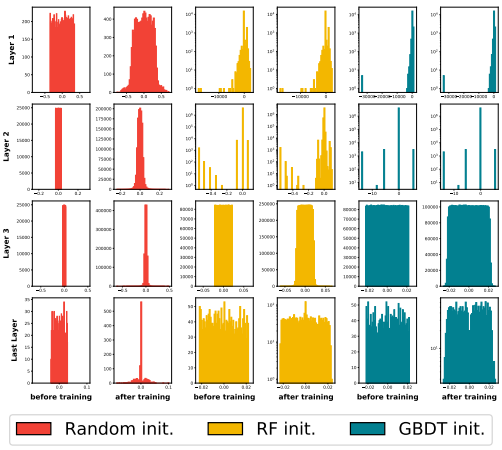


Figure 3.4: Histograms of the first three and last layers' weights before and after the MLP training on Housing. Comparison between random, RF and GBDT initializations.

3.4 Conclusion and Future Work

This work builds upon the permeability that exists between tree methods and neural networks, in particular how the former can help the latter during training, with tabular inputs. We proposed new methods for smartly initializing the first layers of standard MLP using pre-trained tree-based methods. The sparsity of this initialization is preserved during training, which shows that it encodes relevant correlations between the data features. Among deep learning methods, such initializations of MLP always improve the performance compared to the widely used random initialization, and provide an easy-to-use and more efficient alternative to SAINT (attention-based method) for tabular data. The performance of this wisely-initialized MLP is remarkably approaching that of XGBoost, which so far reigns supreme for learning tasks on tabular data.

Limitations & future work While our procedure is quite generic, some restrictions are noticeable. First, our analysis only allows to initialize neural networks with $\tanh$ activation functions; removing this limitation by considering ReLU is a good avenue for future work. Besides, while quite reasonable, our initialization is more time-consuming than the random (default) one. Moreover, we need to further investigate the benefits of our initialization method on very large data sets. Finally, another interesting direction could be using the efficient hyperparameter search in tree-based methods to automatically determine a good default NN architecture.

S1 Details on Deep Forest (DF) and its Translation

The layers of DF are composed of an assortment of Breiman's Random Forests and Completely-Random Forests (CRF, Fan et al. 2003) and are trained one after another in a cascade manner. At a given layer, the outputs of all forests are concatenated, together with the raw input data. This new vector serves as input for the next DF layer. This process is repeated for each layer and the final output is obtained by averaging the forest outputs of the best layer (without raw data).

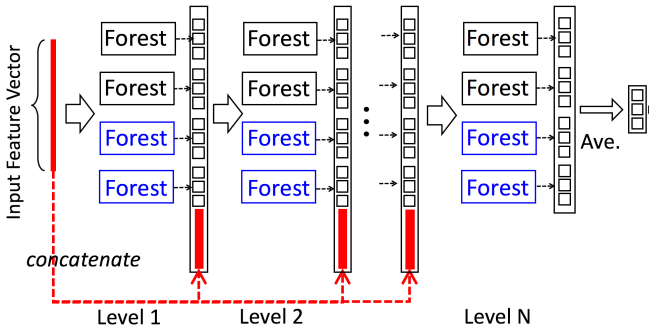


Figure S5: Illustration of the Deep Forest cascade structure for a classification problem with 3 classes. Each level of the cascade consists of two Breiman RFs (black) and two completely random forests (blue). The original input feature vector is concatenated to the output of each intermediate layer. Figure taken from (Zhou et al. 2019).

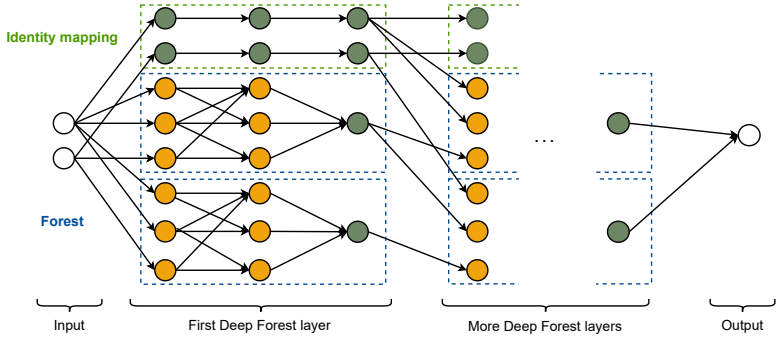


Figure S6: Illustration of the MLP translation of a Deep Forest. Yellow nodes use the $x \mapsto 2.1\mathbb{1}_{\{x>0\}} - 1$ activation function and green nodes use the identity activation function.

S2 Details of the Translation of a Decision Tree into an MLP

Recall that a decision tree codes for a partition of the input space in as many parts as there are leaf nodes in the tree. To know in which partition cell an input feature vector $x \in \mathbb{R}^d$ falls into, we move in the tree from the root to the corresponding leaf using simple rules: at each m-th inner node, x is passed onto the left child node if its i_m-th coordinate is less than or equal to some threshold t_m, and to the right child node otherwise. The decision rule at each inner node of the tree introduces a split of the feature space into two subsets $\mathcal{H}_m^- = \{x \in \mathbb{R}^d \mid x^{(i_m)} \leqslant t_m\}$ and $\mathcal{H}_m^+ = \{x \in \mathbb{R}^d \mid x^{(i_m)} > t_m\}$. Consistent with how the MLP translation works, we intentionally define $\mathcal{H}_m^-$ and $\mathcal{H}_m^+$ such that at each inner node m, $\mathcal{H}_m^- \cup \mathcal{H}_m^+ = \mathbb{R}^d$. Let N be the number of inner nodes of the decision tree; note that the decision tree has exactly $N + 1$ leaf nodes, since it is by definition a complete binary tree, see Figure 3.1 for an illustration. For a leaf node $\ell \in \{1, \ldots, N+1\}$ of the tree, let $\mathcal{P}_\ell^- \subset \{1, \ldots, N\}$ (respectively $\mathcal{P}_\ell^+$) be the set of all inner nodes whose left (respectively right) subtree contains ℓ, that is, $\mathcal{P}_\ell^+ \cup \mathcal{P}_\ell^-$ is the set of all parent nodes

of ℓ. Then, the decision tree sorts an observation $x \in \mathbb{R}^d$ into its leaf $\mathcal{R}_\ell$ if and only if

$$x \in \mathcal{R}_\ell = \left(\bigcap_{m \in \mathcal{P}_\ell^-} \mathcal{H}_m^- \right) \cap \left(\bigcap_{m \in \mathcal{P}_\ell^+} \mathcal{H}_m^+ \right). \tag{3.1}$$

In fact, $\{\mathcal{R}_\ell\}_{\ell \in \mathcal{L}}$ is the feature space partition coded by the tree, see Figure 3.1 for an example. Finally, the tree returns the average response of all training samples that fall into the same leaf as the input data; let us call a_ℓ the average response of all training samples in $\mathcal{R}_\ell$. The final prediction of the decision tree g can therefore be expressed as

$$g(x) = \sum_{\ell=1}^{N+1} a_\ell \mathbb{1}_{\{x \in \mathcal{R}_\ell\}}.$$

Let us now explore how an MLP can be designed to reproduce the prediction of a decision tree. Consider an MLP of depth 3 with N neurons on the first layer. For each inner node $m \in \{1, \dots, N\}$, the m-th neuron of the first layer indicates on which side of the split introduced by this inner node a given feature vector lies: it equals -1 if the feature vector lies in $\mathcal{H}_m^-$ and $+1$ if it lies in $\mathcal{H}_m^+$. This can be achieved applying the following affine transformation and a sign activation function to the feature vector,

$$A_1 : x \in \mathbb{R}^d \mapsto x^{(i_m)} - t_m \quad \text{and} \quad \varphi_1 : x \mapsto \begin{cases} -1 & \text{if } x \leqslant 0 \\ 1 & \text{if } x > 0. \end{cases}$$

The second layer of the 3-layer MLP has $N+1$ neurons. For each leaf node $\ell \in \{1, \dots, N+1\}$, the ℓ-th neuron of the second layer indicates whether a given feature vector $x \in \mathbb{R}^d$ lies in $\mathcal{R}_\ell$ or not: it equals $+1$ if $x \in \mathcal{R}_\ell$ and -1 if $x \notin \mathcal{R}_\ell$. Using equation (3.1), this can be achieved by applying the following affine transformation and a sign activation function to the output of the first layer,

$$A_2 : x \in \mathbb{R}^N \mapsto \sum_{m \in \mathcal{P}_\ell^+} x^{(m)} - \sum_{m \in \mathcal{P}_\ell^-} x^{(m)} - \left| \mathcal{P}_\ell^+ \cup \mathcal{P}_\ell^- \right| + \frac{1}{2} \quad \text{and} \quad \varphi_2 : x \mapsto \begin{cases} -1 & \text{if } x \leqslant 0 \\ 1 & \text{if } x > 0. \end{cases}$$

The last layer of the MLP contains a single output neuron that returns the tree prediction. Using the output of the second layer, this can be achieved by applying the following affine transformation and an identity activation function,

$$A_3 : x \in \mathbb{R}^{N+1} \mapsto \frac{1}{2} \left(\sum_{\ell=1}^{N+1} x^{(\ell)} a_\ell + \sum_{\ell=1}^{N+1} a_\ell \right) \quad \text{and} \quad \varphi_3 : x \mapsto x \tag{3.2}$$

where a_ℓ is the average response of all training samples in $\mathcal{R}_\ell$. Note that $\{a_\ell\}_{\ell=1}^{N+1}$ is a set of real numbers in regression problems and a set of probability vectors representing class distributions in classification problems. An illustration of the MLP translation of a decision tree is shown in Figure 3.1. This translation procedure is explained, for example, in Biau et al. 2019 with more details.

S3 Illustration of our Initialisation Method

We provide below an illustration (Figure S7) showing how the whole MLP is initialised using both the tree-based method for the first layers and the random initialisation for the deeper layers.

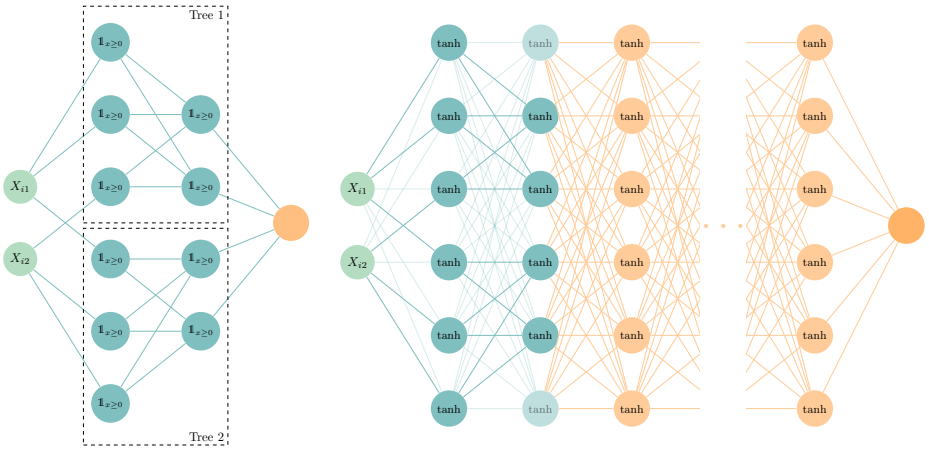


Figure S7: Illustration of the initialization technique on an MLP with 2 inputs and 1 output. In (a), a pre-trained tree-based method composed of 2 trees is represented in a NN fashion involving indicator functions as activation functions. In (b), an MLP of arbitrary depth and involving tanh activation functions is represented at initialization: the weights of the first two layers are initialized using the information captured in (a) (note that all connections marked in transparent blue are initialized to 0). The weights of the subsequent layers are randomly initialized (orange).

S4 Detail on the MLP Translation Accuracy

Recasting a Deep Forest into a deep MLP using our method may suffer from numerical instabilities altering the predictive behaviour. This is due to a phenomenon of catastrophic cancellation, more likely to occur with deep MLP translations. This is explained in the following section.

S4.1 On the Choice of Hyper-Parameters

In Section 3.2.3, four hyper-parameters were introduced to approximate the sign and identity functions through the layers of an elementary MLP.

We address here the choice of the HPs and propose an optimal range for these parameters in the sense that they are as small as possible while guaranteeing a faithful MLP translation.

We focus on the analysis of deep forest translation, as the structure of all other tree-based methods can be seen as a truncated variant of a deep forest. The deep forest is trained and translated into an MLP on each data set (see Section 3.2) for different values of the HPs. To identify the influence of each HP, we make them vary in some range while the other three HPs are fixed to 10^{10}, resulting in an almost perfect approximation of the respective sign and identity functions. Figure S8 shows the predictive performance of a deep forest and its MLP translation playing with different HPs.

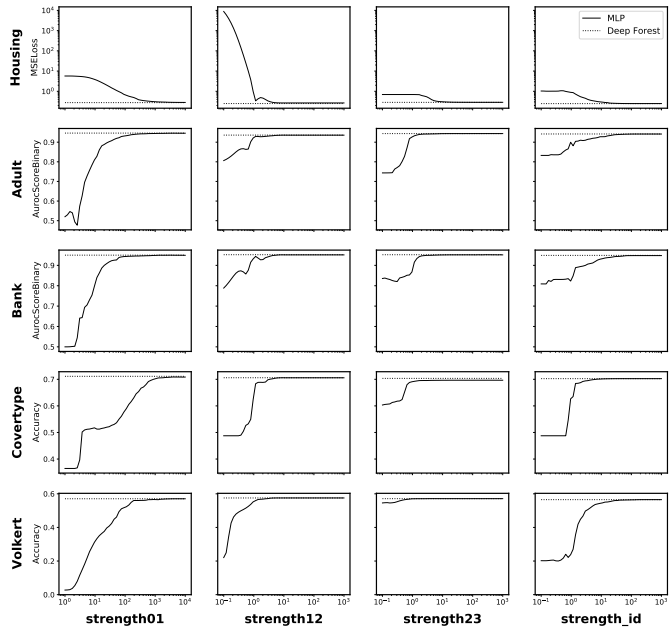


Figure S8: Comparison of the performance of a trained deep forest and its neural network encoding. Deep forest architecture: maximal depth of 8 per tree, 8 trees per forest, 1 forests per layer, 3 layers.

Figure S8 shows in particular that

(i) increasing the HPs beyond some limit value is no longer beneficial as the activation functions are already perfectly approximated;

(ii) across multiple data sets, these limit values are similar.

One could note that the coefficients in the first layer of a decision tree translation should be of a larger order of magnitude than those corresponding to the other activation functions to achieve an accurate translation. To give some insight into why this is the case, recall that the m-th neuron of the first layer determines whether the input vector belongs to $\mathcal{H}_m^-$ or $\mathcal{H}_m^+$, and note that its outputs can be of arbitrarily small size because the vector can be arbitrarily close to the decision boundaries. Note also that an MLP translation would better compromise on translation accuracy to ensure sufficient gradient flow. Based on these observations, we remark that choosing the HP of the following orders allows for maximum gradient flow while still providing an accurate translation: `strength01` $\in [1, 10^4]$, `strength12` $\in [10^{-2}, 10^2]$, `strength23` $\in [10^{-2}, 10^2]$ and `strength_id` $\in [10^{-2}, 10^2]$. This will actually help us later on to calibrate the search spaces when empirically tuning these HPs for each data set.

S4.2 A Fundamental Numerical Instability of the Neural Network Encoding

The encoding of a decision tree by a neural network proposed in Section 3.2.3 is numerically unstable, i.e., it does not necessarily yield the same result as the tree itself, even when using the original, non-approximated activation functions. This is the result of a catastrophic cancellation that occurs within the MLP translation. The term catastrophic cancellation describes

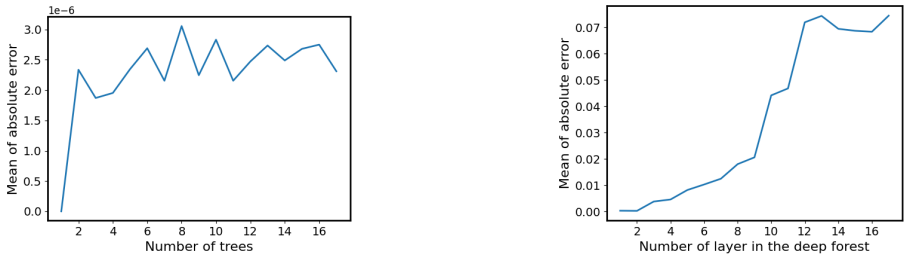
the remarkable loss of precision that occurs when two nearly equal numbers are numerically subtracted. For example, take the numbers $a = 1$ and $b = 10^{-10}$, and perform the computation $(a + b) - a$ on a machine with limited precision, say to 8 significant digits. The machine will return $(a+b)-a = 1-1 = 0$, although this result is clearly not correct. This phenomenon occurs in the third layer of the MLP encoding, see equation (3.2). The two sums calculated in this layer are almost equal in magnitude but have opposite signs, resulting in a catastrophic cancellation that has a greater impact the more partitions of the input space the decision tree uses, i.e. the deeper it is.

Figure S9 illustrates the effect of this phenomenon, comparing the mean approximation error between a simple decision tree and its neural network encoding on the airbnb data set. In Figure S9a, the result at the output layer of the tree was replaced by the exact training mean of the corresponding decision tree partition, compensating for the catastrophic cancellation. No such compensation was done for Figure S9b. This shows the grave implications of this instability: the mean error grows exponentially with the depth of an individual tree.

(a) replacing the output layer's result with the exact training mean of the corresp. tree partition

(b) using the output layer's result with catastrophic cancellation

Figure S9: Illustration of the fundamental numerical instability of the decision tree encoding.



(a) Random Forest.

(b) Deep Forest

Figure S10: Effects of numerical instabilities on more complex tree-based predictors. Airbnb data set. Random Forests are composed of trees of depth 7. Deep forest architecture: tree depth of 7, 5 trees per forest, 1 forest per layer and a variable number of layers.

Although the errors introduced by this phenomenon may not be large for a given decision tree, they might accumulate when several such trees are composed, for example in Random or Deep Forests. Figure S10 compares the mean approximation error between Random/Deep Forests of different complexities and their corresponding neural net encoding on the Airbnb

data set. It shows that the composition of several trees in a cascade manner, as performed by the Deep Forest, leads to a stronger amplification of their individual inaccuracies than the parallel composition of trees, as performed by the Random Forest. This result is to be expected because decision trees composed in parallel do not influence each other's predictions, whereas in a cascade architecture the results of the first layer of decision trees affect the input of the subsequent layers and inaccuracies can thus develop stronger effects.

We note that this catastrophic cancellation can be easily circumvented by introducing an additional layer. If this maps the output of the second layer from $\{-1, 1\}$ to $\{0, 1\}$, the last layer could then simply multiply each of these outputs by the average response of a partition set. However, Figure S10 also shows that the error introduced by the catastrophic cancellation remains relatively small, except for deep forests with many layers. Therefore, we did not immediately address this issue and planned to fall back on this analysis if the MLP coding did not produce the expected results later in our analysis. However, this somewhat imprecise MLP coding worked well for all our purposes.

S5 Supplements to Numerical Evaluations

S5.1 Data sets

Data sets description In the sequel, we run numerical experiments on 10 real-world, heterogeneous, tabular data sets, all but two of which have already been used to benchmark deep learning methods, see Borisov et al. 2021; Somepalli et al. 2021. The chosen data sets represent a variety of different learning tasks and sample sizes. Tables S2 & S3 respectively give links to the platforms storing the data sets (four of them are available on the UCI Machine Learning Repository **ML_Repo**) and an overview of their main properties.

Data set	Link
Housing	Scikit-learn
Airbnb	Inside Airbnb
Diamond	OpenML
Adult	UCI Machine Learning Repository
Bank	UCI Machine Learning Repository
Blastchar	Kaggle
Heloc	FICO
Higgs	UCI Machine Learning Repository
Covertype	UCI Machine Learning Repository
Volkert	AutoML

Table S2: Links to data sets.

	Housing	Airbnb	Diamonds	Adult	Bank	Blastchar	Heloc	Higgs	Covertype	Volkert
Dataset size	20 640	119 268	53 940	32 561	45 211	7 043	9 871	550 000	581 012	58 310
# Num. features	8	10	6	6	7	3	21	27	44	147
# Cat. features	0	3	3	8	9	17	2	1	10	0
Task	Regr.	Regr.	Regr.	Classif.	Classif.	Classif.	Classif.	Classif.	Classif.	Classif.
# Classes	-	-	-	2	2	2	2	2	7	10

Table S3: Main properties of the data sets.

The Housing data set contains U.S. Census household attributes and the associated learning task is to predict the median house value for California districts (**Housing**). The Airbnb data set is provided by the company itself and holds attributes on different Airbnb listings in Berlin, such as the location of the apartment, the number of reviews, etc. The goal is to predict the price of each listing. Similarly, the diamond data set contains characteristics of different diamonds (e.g., carat weight or cut quality), and the goal is to predict the price of a diamond. The Adult data set contains Census information on adults (over 16-year olds) and its prediction task is to determine whether a person earns over $50k a year. The Bank data set is related with direct marketing campaigns (phone calls) of a Portuguese banking institution, the classification goal is to predict whether the client will subscribe a term deposit. The Blastchar data set features information on customers of a fictional company that provides phone and internet services. The classification goal is to predict whether a customer cancels their contract in the upcoming month. The Heloc data set contains personal and credit record information on people that recently took on a line of credit, the classification task being to predict whether they will repay this credit within 2 years. On the Higgs data set (**baldi2014searching**), the classification problem is to distinguish between signal processes that produce Higgs bosons and background processes that do not. For this purpose, it contains kinematic properties measured by the particle detectors in the accelerator that have been produced using Monte Carlo simulations. The Covertype data set contains cartographic variables on forest cells and it's task is to predict the forest cover type. Finally, for the Volkert data set, different patches of the same size have been cut from images that belong to 10 different landscape scenes (coast, forest, mountain, plain, etc.). Each observation contains visual descriptors of one patch, the goal of this classification problem is to find the landscape type of the original picture.

S5.2 Implementation Details

RFs are implemented using `sklearn`'s `RandomForestRegressor` and `RandomForestClassifier` classes with default configuration for all parameters that are not mentioned explicitly. DFs are implemented using the `ForestLayer` library (Zhou et al. 2019) and GBDTs are implemented using the `XGBoost` library (Chen et al. 2016). MLPs are implemented and trained with `pytorch`, using the mean-squared error and the cross entropy as objective function for regression and classification problems respectively. The SAINT model is implemented using the library provided by Somepalli et al. 2021.

All methods are trained on a 32 GB RAM machine using 12 Intel Core i7-8700K CPUs, and one NVIDIA GeForce RTX 2080 GPU when possible (only the GDBT and MLP implementations including SAINT use the GPU). Hyper-parameter searches are parallelized on up to 4 of these machines.

Hyper-parameter optimization We tune all hyper-parameters using the `optuna` library (Akiba et al. 2019) with a fixed number of iterations for all models. In this context, an *iteration* corresponds to a set of hyper-parameters whose performance is evaluated with respect to a given method. The `optuna` library uses Bayesian optimization and, in particular, the tree-structured Parzen estimator model (**TPE**) to determine the parameters to be explored at each iteration of hyper-parameter optimization. This approach has been reported to outperform random search for hyper-parameter optimization (**Bayesian_optim_better_than_random_search**).

Data pre-processing Machine learning pipelines often include pre-processing transformations of the input data before the training phase, a necessary step, especially when using neural networks (**bishop1995neural**). We follow the pre-processing that is used in Borisov et al. 2021

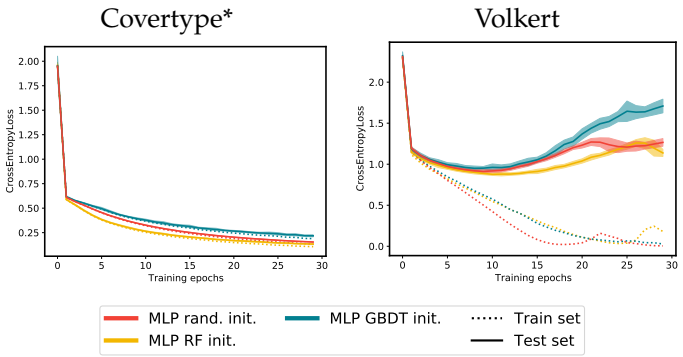


Figure S11: Optimization behaviour of randomly, RF and GBDT initialized MLP and SAINT evaluated over a 5 times repeated (stratified) 5-fold of each data set, according to Protocol P1, but where the MLP width is fixed to 2048 for all methods. The lines and shaded areas report the mean and standard deviation. *evaluation on a single 5-fold cross validation.

and Somepalli et al. 2021. Hence, we normalize all continuous input features to zero mean and unit variance. This corresponds to linearly transform the input features as follows

$$\tilde{\mathbf{x}}_{\cdot j} = \frac{\mathbf{x}_{\cdot j} - \mu}{\sigma}$$

where $\mathbf{x}_{\cdot j}$ is the j-th continuous feature of either train, validation or test observations, μ and σ are the mean and standard deviation calculated over the train set only. This way we assure that no information from the validation or test sets is used in the normalization step. Moreover, all categorical features are label encoded, i.e. each level of a categorical variable is replaced with an integer in $\{1, \ldots, \#\text{ levels}\}$.

S5.3 Working with an Arbitrary Width in P1 (Optimization Behaviour)

Figure S11 shows the optimization behaviour of the randomly, RF and GBDT initialized MLP on the multi-class classification problems. Note that in contrast to Figure 3.2 in this setting, which is less restrictive for RF initialization, this method does indeed lead to *a faster convergence* and *a better minimum* (in terms of generalization).

However, for these multi-class classification problems, the GBDT initialization tends to deteriorate the optimization compared to RF or random initialization methods. Indeed, RF are genuinely multiclassification predictors whose splits are built using all output classes simultaneously whereas splits in GBDT are only built following a one-vs-all strategy. This implies that, with a fixed budget of splits (and therefore of neurons), RF are likely to be more versatile than GBDT.

S5.4 Additional Material for Protocol P2 (Generalization Behaviour)

Details About Additional NN Training Techniques

In Protocol 2, we assess the performances of generalization of the proposed methods, of the predictors described in Section 3.3.2, but also on three additional NN techniques:

1. the Xavier initialization (Glorot et al. 2010) corresponds to a rescaled uniform initialization

$U \sim \mathcal{U}\left[\pm\frac{\sqrt{6}}{\sqrt{n_{j+1}+n_j}}\right]$, where n_j are the number of neurons in layer j. This random initialization is very close to the one used by default in this paper and simply denoted "random init";

2. the layer-sequential unit-variance orthogonal initialization (LSUV) (Mishkin et al. 2015) consists in a simple initialization that combines elements of (Glorot et al. 2010) and (**saxe2013exact**). In a first step, the weights of each layer are initialized as random orthogonal matrices. Then, the variance in the outputs on each layer on the training data is scaled close to 1 by repeatedly dividing the layer's weights by the empirically determined standard deviation. Although targeted to Computer Vision applications, this approach seems easily adaptable for our case;

3. the winning ticket network pruning (Frankle et al. 2018) is more a simplification approach of the NN architecture during training than an initialization technique. That being said, it remains interesting to compare this strategy to the one developed in the paper, as the winning ticket network pruning enforces NN sparsity during training. This can be indeed put in parallel to the sparsity of the first layers introduced by the proposed initialization and preserved during training. The principle is to train a randomly initialized network, pruning it to obtain a sparse NN with similar performance and then re-train the sparse network a second time using the same instance of random initialization as before. These steps are repeated a certain number of times. The winning ticket network pruning is therefore computationally very intense and has to the best of our knowledge only been studied on medium-sized data sets. We thus use a slightly different procedure than (Frankle et al. 2018) to determine winning tickets. First of all, we allocate at most N training epochs to determining a winning ticket where N is the number of epochs during the final model training itself. This fixed number of training epochs is then distributed among n pruning rounds, each of which consists in training the model (for N/n epochs), pruning it, and resetting all non-pruned weights to their initial (random) coefficients. This approach takes the same time as one-shot pruning but proves to be more efficient.

Extension of Table 3.1 (Best Performances)

Table S4 provides a comparison of the performances obtained by ourselves and the literature (where available) for each model. Notice that our results are broadly consistent with those in the literature, with two exceptions. First, our random initialized MLP tends to perform better than in the literature, which can be explained by the fact that we use a much larger search space than usual for the MLP width (see Section 3.3.5 for a discussion on this). Second, our performance on Higgs is significantly lower than in the literature. This can be explained by the fact that we only include 5% of the original data set's observations in our analysis due to hardware limitations that do not allow us to train large MLP on 11M samples.

Benefits of Training the Feature Extractor via Gradient Descent

In Section 3.3, we demonstrated ways in which our initialization method can be beneficial for MLP training, resulting in faster convergence towards better minima (in the sens of generalization). A natural question that might arise in this context is whether translating the tree-based method into a MLP framework is actually beneficial. After all, one could be tempted to directly use the tree-based method as a feature pre-processing (without translating it into an MLP) and feed the resulting features into an MLP. In this case, the MLP would be trained via gradient descent *without* the feature extraction. However, it turns out that (i) the weights on the sparse feature extraction

Model \ Data set	Housing (†)		Airbnb	Diamonds	Covertype (†)		Volkert (§)	
	MSE ↓		MSE ↓ $\times 10^3$	MSE ↓ $\times 10^{-3}$	Accuracy ↑ in %		Accuracy ↑ in %	
	perf. in literature	our results	our results	our results	perf. in literature	our results	perf. in literature	our results
Random Forest	0.272±0.006	0.263±0.009	5.39±0.13	9.80±0.35	78.1±0.1	83.6±0.1	66.3±1.3	64.2±0.3
GBDT	0.206±0.005	**0.208±0.010**	**4.71±0.15**	**7.38±0.28**	97.3±0.0	**97.0±0.0**	69.0±0.5	71.3±0.4
Deep Forest	-	0.225±0.008	**4.68±0.16**	8.23±0.29	-	92.4±0.1*	-	66.3±0.4
MLP rand. init.	0.263±0.008	0.258±0.011	5.07±0.16	15.5±12.5	91.0±0.4	96.7±0.0	63.0±1.56	72.2±0.4
MLP RF init.	-	0.222±0.009	**4.66±0.16**	7.93±0.22	-	96.7±0.0	-	**74.1±0.4**
MLP GBDT init.	-	**0.206±0.007**	**4.70±0.09**	8.15±0.35	-	96.2±0.0	-	73.5±0.5
MLP DF init.	-	0.234±0.016	**4.81±0.13**	8.28±0.24	-	94.5±0.3*	-	71.3±0.5
SAINT	0.226±0.004	0.258±0.011	**4.81±0.15**	17.7±3.83	96.3±0.1	96.6±0.1*	70.1±0.6	70.1±0.4

Model \ Data set	Adult (†)		Bank (§)		Blastchar (§)		Heloc (†)		Higgs (†)	
	AUC ↑ in %		AUC ↑ in %		AUC ↑ in %		AUC ↑ in %		AUC ↑ in %	
	perf. in literature	our results	perf. in literature	our results	perf. in literature	our results	perf. in literature	our results	perf. in literature	our results
Random Forest	91.7±0.2	91.6±0.3	89.1±0.3	92.8±0.3	80.6±0.7	**84.5±1.2**	90.0±0.2	91.3±0.6	79.7±0.0	80.4±0.1
GBDT	92.8±0.1	**92.7±0.3**	93.0±0.2	**93.3±0.3**	81.8±0.3	**84.7±1.0**	92.2±0.0	**92.1±0.4**	85.9±0.0	82.8±0.1
Deep Forest	-	91.8±0.3	-	92.9±0.2	-	**83.7±1.2**	-	90.3±0.5	-	81.2±0.0*
MLP rand. init.	90.3±0.2	90.5±0.4	91.5±0.2	91.0±0.3	59.6±0.3	81.4±1.2	80.3±0.1	80.1±0.1	85.6±0.0	83.2±0.3
MLP RF init	-	92.1±0.3	-	92.4±0.4	-	**84.4±1.2**	-	**91.7±0.4**	-	**83.6±0.1**
MLP GBDT init.	-	92.2±0.3	-	92.5±0.3	-	**84.6±1.2**	-	91.5±0.6	-	83.0±0.0
MLP DF init.	-	91.9±0.4	-	92.2±0.3	-	**84.2±1.0**	-	91.4±0.6	-	83.3±0.1*
SAINT	91.6±0.4	91.6±0.3	93.3±0.1	92.2±0.4	84.7±0.3	**84.0±0.8**	90.7±0.2	90.2±0.7	88.3±0.0	**83.7±0.1***

Table S4: Best scores for Protocol P2. For each data set, our best overall score is highlighted in **bold** and our best Deep Learning score is underlined. Our scores are based on 5 times repeated (stratified) 5-fold cross validation. For each of our models, HP were selected via the `optuna` library (100 iterations). Sources for literature values: Borisov et al. 2021 (†) and Somepalli et al. 2021 (§). *score based on a single 5-fold cross validation.

layer are indeed modified during gradient descent optimization and (ii) training the feature extractor via gradient descent largely contributes to the competitive generalization performance of our initialization method.

Figure S12 shows the histograms of the differences between all MLP parameters at initialization (RF strategy) and after training. As the histograms indicate, the weights in all layer throughout the MLP are modified during training. In particular, the weights of the first two (RF initialized) layers are not stationary but change to a large extent.

Table S5 shows the generalization performance of MLP initialized with the RF strategy, and compares the two scenarios in which the parameters of the first two layers (that is, the feature extraction layers built using the RF) are modified or frozen during MLP training. These results show that training the feature extraction layers is essential for the success of our initialization method.

Number of Parameters of Best Neural Networks

In Table S6, we compare the number of parameters of each NN method. Although the tree-based initialised MLP contain more parameters than the randomly initialized ones, the former are mostly sparse and the execution times are close (see Table S7). Finally note that the number of parameters of the RF/GBDT init. MLP is globally on par with that of SAINT (sometimes more, sometimes less) but for a smaller execution times (Table S7) and mostly better performances (Table S4).

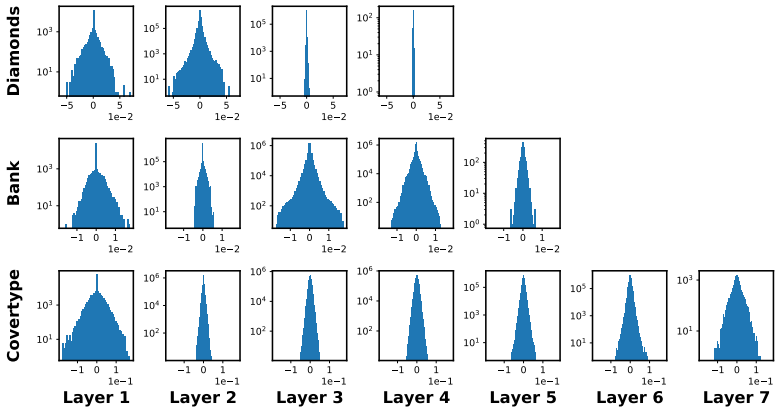


Figure S12: Histograms of the difference between all MLP parameters at initialization (RF strategy) and after training. Three data sets have been chosen for illustrative proposes. The behaviour in the light of our analysis (see S5.4) is similar on the 7 other data sets.

Model \ Data set	Housing	Airbnb	Diamonds	Adult	Bank	Blastchar	Heloc	Higgs	Covertype	Volkert
	MSE ↓	MSE ↓ ($x10^3$)	MSE ↓ ($x10^{-3}$)	AUC ↑ (in %)	AUC ↑ (in %)	AUC ↑ (in %)	AUC ↑ (in %)	AUC ↑ (in %)	Acc. ↑ (in %)	Acc. ↑ (in %)
MLP rand. init.	0.258±0.011	5.07±0.16	15.5±12.5	90.5±0.4	91.0±0.3	81.4±1.2	80.1±0.1	83.2±0.3	96.7±0.0	72.2±0.4
MLP RF init. frozen	0.262±0.018	14.5±2.71	13.7±1.48	91.1±0.3	90.9±0.5	84.4±0.9	91.0±0.6	75.9±0.2	92.2±0.2	69.4±0.5
MLP RF init.	0.222±0.009	4.66±0.16	7.93±0.22	92.1±0.3	92.4±0.4	84.4±1.2	91.7±0.4	83.6±0.1	96.7±0.0	74.1±0.4

Table S5: Best scores for Protocol 2. The scores are based on 5 times repeated (stratified) 5-fold cross validation. MLP RF init. frozen refers to the MLP RF init. model where the parameters of the first two layers (that are initialized using the Random Forest) are frozen during training, that is, they are kept at their initial values.

Model \ Data set	Housing	Airbnb	Diamonds	Adult	Bank	Blastchar	Heloc	Higgs	Covertype	Volkert
MLP rand. init.	2.47M	1.86M	363k	1.09M	52.4K	13.1M	11.3M	11.6M	1.14M	9.03M
MLP RF init.	33.6M	12.6M	8.42M	29.4M	8.43M	25.2M	16.8M	4.26M	21.1M	17.1M
MLP GBDT init.	8.41M	12.6M	12.6M	33.6M	8.43M	16.8M	25.2M	8.46M	4.32M	21.3M
MLP DF init.	88.1M	34.0M	59.3M	42.0M	46.2M	34.36M	25.8M	43.2M	57.6M	34.1M
SAINT	56.8M	27.0M	53.1M	7.20M	6.12M	322M	98.2M	43.2M	6.44M	169M

Table S6: Comparison of the number of parameters for each model.

Model \ Data set	Housing	Airbnb	Diamonds	Adult	Bank	Blastchar	Heloc	Higgs	Covertype	Volkert
MLP rand. init.	5.96 (32)	91.9 (98)	13.3 (31)	11.8 (37)	21.6 (62)	6.78 (34)	14.3 (60)	467 (32)	312 (69)	12.3 (31)
MLP RF init.	37.8 (29)	131 (44)	26.8 (25)	17.2 (19)	21.5 (23)	8.58 (18)	6.61 (15)	253(39)	2040 (91)	28.3 (25)
MLP GBDT init.	22.9 (49)	280 (95)	53.6 (37)	34.5 (31)	7.47 (3)	7.76 (7)	8.54 (8)	63.0 (5)	437 (66)	52.8 (37)
MLP DF init.	233 (72)	360 (48)	182 (31)	99.4 (54)	105 (26)	29.0 (52)	14.7 (19)	3280 (76)	5580 (95)	181 (31)
SAINT	81.9 (37)	640 (83)	394 (84)	15.6 (11)	52.7 (32)	7.23 (2)	51.0 (31)	2310 (19)	6580 (97)	394 (84)

Table S7: Comparison of the execution time in seconds for model initialization and training until the best validation lost is reached. The number of training epochs is indicated in parentheses.

Model \ Data set	Housing	Airbnb	Diamonds	Adult	Bank	Blastchar	Heloc	Higgs	Covertype	Volkert
MLP rand. init.	0.00/5.96 (32)	0.00/91.9 (98)	0.00/13.3 (31)	0.00/11.8 (37)	0.00/21.6 (62)	0.00/6.78 (34)	0.00/14.3 (60)	0.00/467 (32)	0.00/312 (69)	0.00/12.3 (31)
MLP RF init.	2.21/35.6 (29)	1.87/129 (44)	2.20/24.6 (25)	1.16/16.0 (19)	1.89/19.6 (23)	2.81/5.77 (18)	1.78/4.83 (15)	6.31/247 (39)	3.67/2040 (91)	3.70/24.6 (25)
MLP GBDT init.	5.35/17.5 (49)	4.19/276 (95)	4.65/48.9 (37)	2.32/32.2 (31)	4.27/3.20 (3)	6.07/1.69 (7)	4.65/3.89 (8)	4.18/58.8 (5)	2.30/435 (66)	3.88/48.9 (37)
MLP DF init.	15.1/218 (72)	5.31/355 (48)	7.19/175 (31)	8.36/91 (54)	9.25/96 (26)	6.64/22.3 (52)	5.64/9.04 (19)	18.9/3260 (76)	11.2/5570 (95)	5.87/175 (31)

Table S8: Comparison of the execution time in seconds for MLP initialization/training until the best validation lost is reached. The number of training epochs is indicated in parentheses. A value of 0.00 indicates running times smaller than 5×10^{-3} seconds.

Comparison of the Execution Times of the Best Neural Networks

Table S7 presents a comparison of the execution times of the training of different NN methods using the hyper-parameters determined by the protocol P2. For each model, the total training time (initialization + gradient descent optimization) is given, measured up to the point where the best validation loss is reached ("early stopping"). It shows that RF/GBDT initialized MLP train faster than SAINT and a bit slower than randomly initialized MLP. For completeness, Table S8 gives the execution time for the initialization and training step separately.

Optimization Behaviour

For completeness, Figure S13 shows the optimization behaviour of the randomly, RF and GBDT initialized MLP as well as SAINT under the Protocol P2.

S5.5 Hyper-Parameter Detting

Search Spaces

Table S9 shows the HP search spaces that were used to determine an optimal HP setting. The same search spaces were used for the experimental protocols P1 and P2. Note that, in Table S9, n_classes corresponds to the number of classes for classification problems and is 1 for regression problems. Furthermore, the different search spaces given for SAINT were used for smaller/larger data sets, where a data set qualifies as smaller if it has less that 50 explanatory variables.

Experimental Protocol P2

Tables S10 and S11 show the HP setting used for the experimental protocol P2. For the search spaces and descriptions of the function of each HP see Table S9.

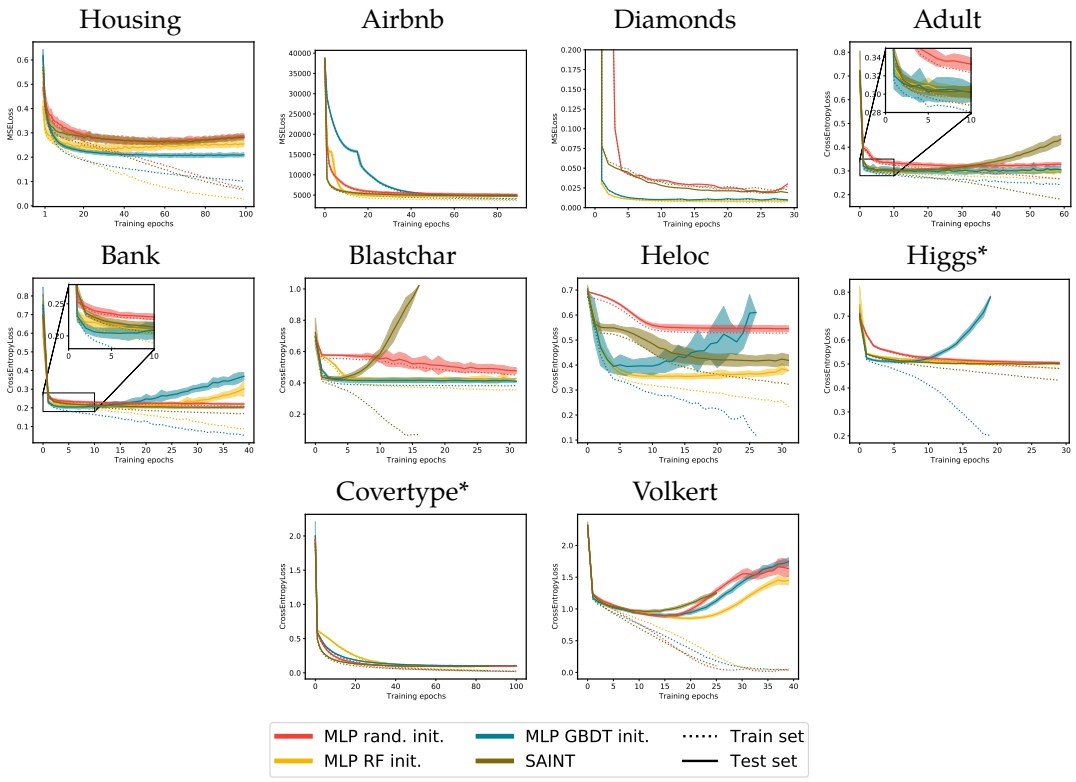


Figure S13: Optimization behaviour of randomly, RF and GBDT initialized MLP and SAINT evaluated over a 5 times repeated (statisfied) 5-fold of each data set, according to Protocol P2. The lines and shaded areas report the mean and standard deviation. *evaluation on a single 5-fold cross validation.

Method	Parameter	Search space	Function
Random Forests	max_depth	$\{1, \ldots, 12\}$	see here
	n_estimators	$\{1000\}$	
	max_features	$[0, 1]$	
GBDT	max_depth	{1,...,12}	see here
	n_estimators	$\{1000\}$	
	reg_alpha	$[10^{-8}, 1]$	
	reg_lambda	$[10^{-8}, 1]$	
	learning_rate	$[0.01, 0.3]$	
Deep Forest	forest_depth	$\{1, 2, 3\}$	Number of Deep Forest layers
	n_forests	$\{1\}$	Number of forests per Deep Forest layer
	n_estimators	$\{1000\}$	RF parameters, see here
	max_depth	$\{1, \ldots, 12\}$	
	max_features	$[0, 1]$	
MLP random init.	learning_rate	$[10^{-6}, 10^{-1}]$	learning rate of SGD training
	depth	$\{1, \ldots, 10\}$	number of layer
	width	$\{1, \ldots, 2048\}$	number of neurons per layer
	epochs	$\{100\}$	number of SGD training epochs
	batch_size	$\{256\}$	batch size of SGD training
MLP RF init.	max_depth	$\{1, \ldots, 11\}$	Parameters of the RF initializer, see here
	n_estimators	$2048/2^{\text{max\_depth}}$	
	max_features	$[0, 1]$	
	learning_rate	$[10^{-6}, 10^{-1}]$	learning rate of SGD training
	depth	$\{3, \ldots, 10\}$	number of layer
	width	$\{2048\}$	number of neurons per layer
	epochs	$\{100\}$	number of SGD training epochs
	batch_size	$\{256\}$	batch size of SGD training
	strength01	$[1, 10^4]$	MLP translation parameters, see Section 3.2.3
	strength12	$[0.01, 100]$	
MLP GBDT init.	max_depth	{1,...,11}	Parameters of the GBDT initializer, see here
	n_estimators	$2048/(\text{n\_classes} \cdot 2^{\text{max\_depth}})$	
	reg_alpha	$[10^{-8}, 1]$	
	reg_lambda	$[10^{-8}, 1]$	
	learning_rate_GBDT	$[0.01, 0.3]$	
	learning_rate	$[10^{-6}, 10^{-1}]$	learning rate of SGD training
	depth	$\{3, \ldots, 10\}$	number of layer
	width	$\{2048\}$	number of neurons per layer
	epochs	$\{100\}$	number of SGD training epochs
	batch_size	$\{256\}$	batch size of SGD training
	strength01	$[1, 10^4]$	MLP translation parameters, see Section 3.2.3
	strength12	$[0.01, 100]$	
MLP DF init.	forest_depth	$\{1, 2, 3\}$	Number of Deep Forest layers
	n_forests	$\{1\}$	Number of forests per Deep Forest layer
	n_estimators	$2048/2^{\text{max\_depth}}$	
	max_depth	$\{1, \ldots, 12\}$	RF parameters, see here
	max_features	$[0, 1]$	
	learning_rate	$[10^{-6}, 10^{-1}]$	learning rate of SGD training
	depth	$\{3, \ldots, 10\}$	number of layer
	width	$\{2048\}$	number of neurons per layer
	epochs	$\{100\}$	number of SGD training epochs
	batch_size	$\{256\}$	batch size of SGD training
	strength01	$[1, 10^4]$	
	strength12	$[0.01, 100]$	MLP translation parameters, see Section 3.2.3
	strength23	$[0.01, 100]$	
	strength_id	$[0.01, 100]$	
SAINT	epochs	$\{100\}$	number of SGD training epochs
	batch_size	$\{256\}/\{64\}$	batch size of SGD training
	dim	$[32, 64, 128]/[8, 16]$	number of neurons per layer in attention block
	depth	$\{1, 2, 3\}$	number of layers in each attention block
	heads	$\{2, 4, 8\}$	number of head in each attention layer
	dropout	$\{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$	dropout used during SGD training

Table S9: Hyper-parameter search spaces used for numerical evaluations.

Method	Parameter	Housing	Airbnb	Adult	Bank	Covertype	Volkert
Random Forests	max_depth	12	12	11	12	12	12
	n_estimators	1000	1000	1000	1000	1000	1000
	max_features	0.437	0.623	0.596	0.943	0.811	0.688
GBDT	max_depth	12	9	6	7	11	10
	n_estimators	1000	1000	1000	1000	1000	1000
	reg_alpha	0.305	4.60×10^{-6}	2.39×10^{-5}	1.52×10^{-4}	0.728	4.47×10^{-6}
	reg_lambda	1.13×10^{-2}	1.75×10^{-8}	1.35×10^{-6}	1.07×10^{-3}	6.51×10^{-4}	1.71×10^{-6}
	learning_rate	3.82×10^{-2}	0.238	1.08×10^{-2}	1.34×10^{-2}	0.181	0.107
Deep Forest	forest_depth	4	9	2	2	9	3
	n_forests	1	1	1	1	1	1
	n_estimators	1000	1000	1000	1000	1000	1000
	max_depth	5	12	11	9	12	12
	max_features	0.361	0.410	0.166	0.206	0.218	0.134
MLP random init.	learning_rate	9.01×10^{-4}	4.21×10^{-4}	2.07×10^{-4}	1.1×10^{-4}	1.15×10^{-4}	2.29×10^{-4}
	depth	4	4	4	4	4	6
	width	1100	959	1175	856	738	1482
	epochs	100	100	100	100	100	100
	batch_size	256	256	256	256	256	256
MLP RF init.	max_depth	8	10	8	8	10	8
	n_estimators	8	2	8	8	2	8
	max_features	0.442	0.321	0.613	0.650	0.897	0.825
	learning_rate	1.04×10^{-4}	1.72×10^{-4}	1.55×10^{-5}	1.01×10^{-4}	1.04×10^{-5}	1.45×10^{-4}
	depth	10	5	5	4	7	6
	width	2048	2048	2048	2048	2048	2048
	epochs	100	100	100	100	100	100
	batch_size	256	256	256	256	256	256
	strength01	1090	668	537	71.4	13.7	1.02
	strength12	0.0749	1.09	62.7	34.5	1.05×10^{-2}	5.53×10^{-2}
MLP GBDT init.	max_depth	3	4	4	4	8	4
	n_estimators	256	128	128	128	1	12
	reg_alpha	1.30×10^{-7}	1.10×10^{-2}	1.26×10^{-8}	0.413	1.33×10^{-2}	6.76×10^{-6}
	reg_lambda	1.57×10^{-7}	9.52×10^{-4}	7.85×10^{-4}	7.48×10^{-3}	0.643	1.99×10^{-7}
	learning_rate_GBDT	0.211	0.297	0.202	0.285	0.112	0.272
	learning_rate	1.11×10^{-5}	1.97×10^{-5}	4.77×10^{-5}	6.22×10^{-4}	6.19×10^{-5}	1.63×10^{-4}
	depth	4	5	6	4	3	7
	width	2048	2048	2048	2048	2048	2048
	epochs	100	100	100	100	100	100
	batch_size	256	256	256	256	256	256
	strength01	575	7830	132	20.5	7280	4.08
	strength12	5.60	0.461	66.0	5.52	93.4	7.11×10^{-2}
MLP DF init.	forest_depth	6	3	3	2	3	2
	n_forests	1	1	1	1	1	1
	n_estimators	16	2	64	32	2	8
	max_depth	7	10	5	6	10	8
	max_features	0.350	0.598	0.992	0.322	0.633	0.342
	learning_rate	1.04×10^{-5}	6.67×10^{-5}	1.54×10^{-5}	3.08×10^{-5}	1.58×10^{-5}	2.31×10^{-4}
	depth	23	10	9	13	15	9
	width	2048	2048	2048	2048	2048	2048
	epochs	100	100	100	100	100	100
	batch_size	256	256	256	256	256	256
	strength01	515	36.6	41.0	15.5	51.6	1.41
	strength12	0.162	0.242	10.6	0.213	0.124	0.154
	strength23	1.94	10.4	47.8	1.94	4.26×10^{-2}	0.149
	strength_id	3.63×10^{-2}	6.34×10^{-2}	7.44	2.75×10^{-2}	5.09×10^{-2}	3.69
SAINT	epochs	100	100	100	100	100	100
	batch_size	256	256	256	256	64	256
	dim	128	64	32	32	8	16
	depth	3	2	2	1	2	2
	heads	2	8	2	8	4	8
	dropout	0.2	0	0.4	0.8	0.5	0.8

Table S10: Hyper-parameters used for the experimental protocol P2.

Method	Parameter	Diamonds	Blastchar	Heloc	Higgs
Random Forests	max_depth	12	6	9	12
	n_estimators	1000	1000	1000	1000
	max_features	0.967	0.547	0.607	0.577
GBDT	max_depth	7	1	1	11
	n_estimators	1000	1000	1000	1000
	reg_alpha	0.341	7.15×10^{-7}	0.123	2.29×10^{-8}
	reg_lambda	5.15×10^{-4}	1.59×10^{-7}	1.44×10^{-2}	0.391
	learning_rate	9.17×10^{-2}	1.48×10^{-2}	0.282	2.46×10^{-2}
Deep Forest	forest_depth	4	7	10	3
	n_forests	1	1	1	1
	n_estimators	1000	1000	1000	1000
	max_depth	12	2	4	12
	max_features	0.454	0.641	0.196	0.163
MLP random init.	learning_rate	2.35×10^{-4}	1.05×10^{-4}	1.14×10^{-6}	2.26×10^{-5}
	depth	9	8	8	9
	width	1011	1475	1369	1284
	epochs	100	100	100	100
	batch_size	256	256	256	256
MLP RF init.	max_depth	10	5	7	9
	n_estimators	2	64	16	4
	max_features	0.904	0.425	0.728	0.670
	learning_rate	6.67×10^{-5}	5.07×10^{-6}	7.33×10^{-6}	2.17×10^{-5}
	depth	4	8	6	3
	width	2048	2048	2048	2048
	epochs	100	100	100	100
	batch_size	256	256	256	256
	strength01	19.8	4500	331	1.43
	strength12	0.420	42.9	1.06	0.329
MLP GBDT init.	max_depth	3	1	3	5
	n_estimators	256	1024	256	64
	reg_alpha	4.56×10^{-2}	1.63×10^{-5}	6.21×10^{-7}	2.58×10^{-6}
	reg_lambda	6.17×10^{-4}	2.19×10^{-4}	3.03×10^{-4}	3.20×10^{-6}
	learning_rate_GBDT	0.214	4.72×10^{-2}	8.42×10^{-2}	0.290
	learning_rate	8.94×10^{-5}	5.60×10^{-6}	4.54×10^{-4}	1.36×10^{-4}
	depth	5	6	8	4
	width	2048	2048	2048	2048
	epochs	100	100	100	100
	batch_size	256	256	256	256
	strength01	3870	4690	6550	4780
	strength12	56.6	21.0	31.8	0.423
MLP DF init.	forest_depth	3	2	2	3
	n_forests	1	1	1	1
	n_estimators	4	128	64	8
	max_depth	9	4	5	8
	max_features	0.695	0.516	0.280	0.572
	learning_rate	2.04×10^{-5}	2.00×10^{-6}	1.91×10^{-5}	9.33×10^{-6}
	depth	16	10	8	12
	width	2048	2048	2048	2048
	epochs	100	100	100	100
	batch_size	256	256	256	256
	strength01	21.0	93.0	97.8	1.12
	strength12	0.119	20.0	0.987	9.22×10^{-2}
	strength23	5.34×10^{-2}	0.283	27.1	0.207
	strength_id	0.358	0.475	9.70	0.152
SAINT	epochs	100	100	100	100
	batch_size	256	256	256	64
	dim	64	128	64	16
	depth	3	3	3	2
	heads	4	8	2	8
	dropout	0.2	0.5	0.8	0.8

Table S11: Hyper-parameters used for the experimental protocol P2.

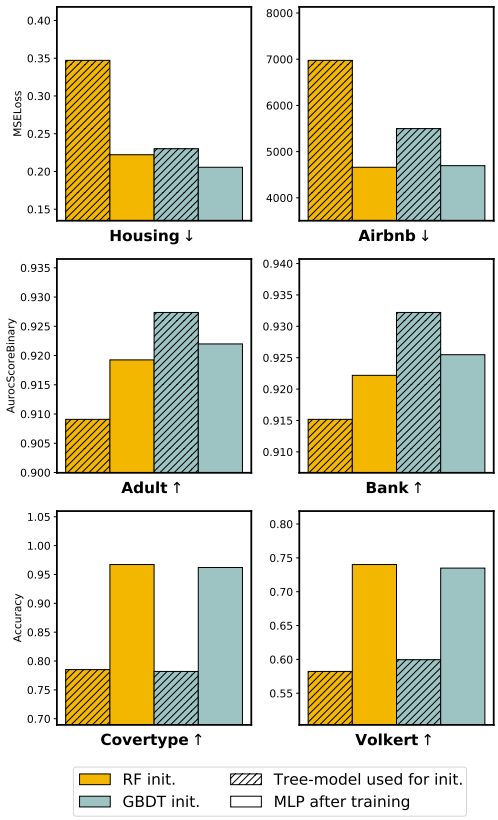


Figure S14: Comparison of the performance of the RF and GBDT models used for initialization and the final performance of the corresponding MLPs.

S5.6 Performances of Tree-Based Methods Used for Initialisation of MLP

Figure S14 compares the performance of RF and GBDT models and the performance of optimized MLP, initialized with RF and GBDT respectively. We can notice that the difference in performance between GBDT and RF does not systematically turn into the same difference in performance for the corresponding trained networks. This suggests that beyond their respective performances, the very structures of RF and GBDT predictors play an important role in the final MLP performances.

S5.7 Additional Figures to Section 3.3.5 (Analyzing key elements of the new initialization methods)

Figures S15 and S16 show the same histograms as Figure 3.4 evaluated on the other data set considered in protocol P2. Note the logarithmic y-axis for the first two RF and GBDT initialized layers.

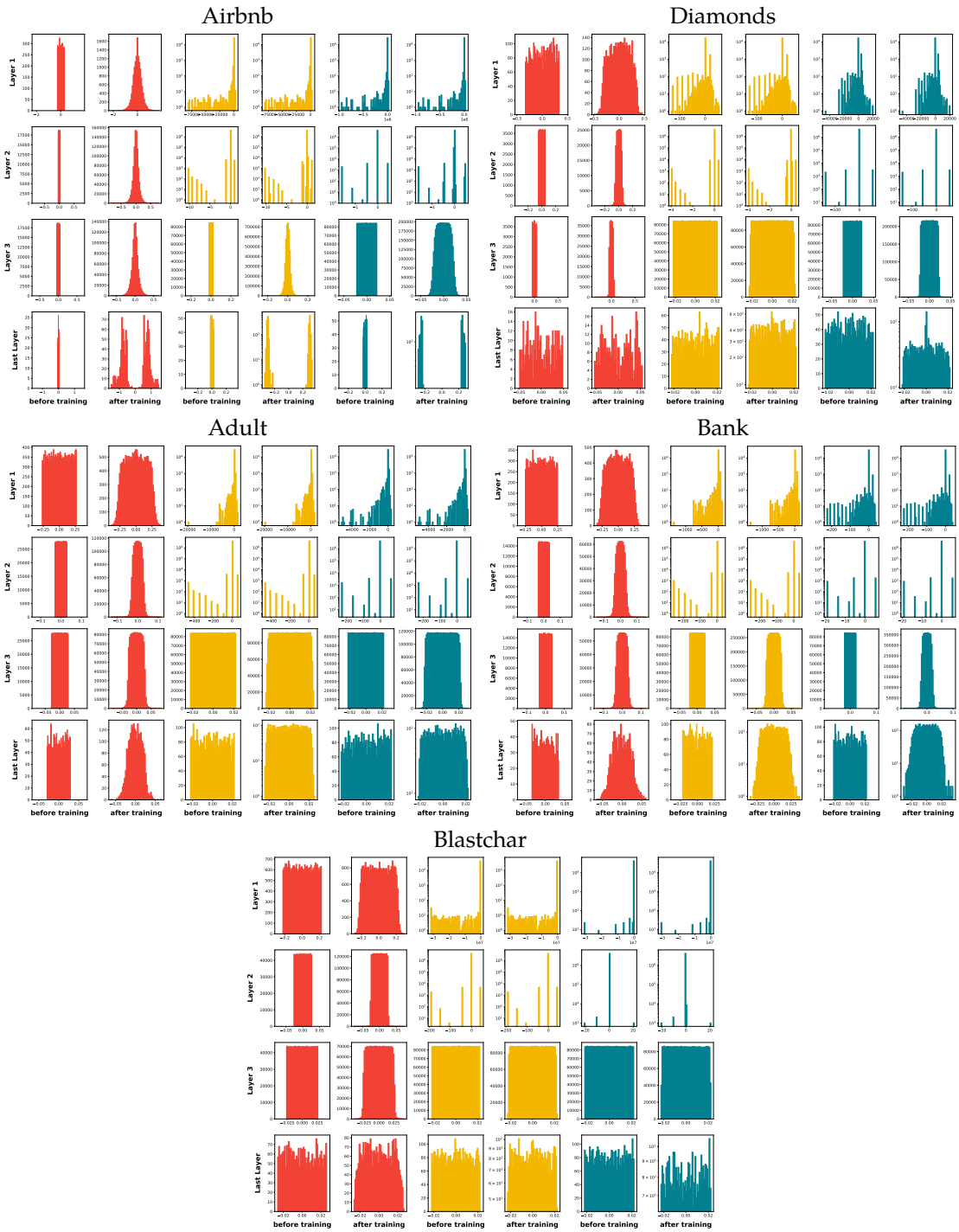


Figure S15: Histograms of the first three first and the last layers' weights before and after the MLP training on the Airbnb, Diamonds, Adult, Bank and Blastchar data sets. Comparison between random, RF and GBDT initializations.

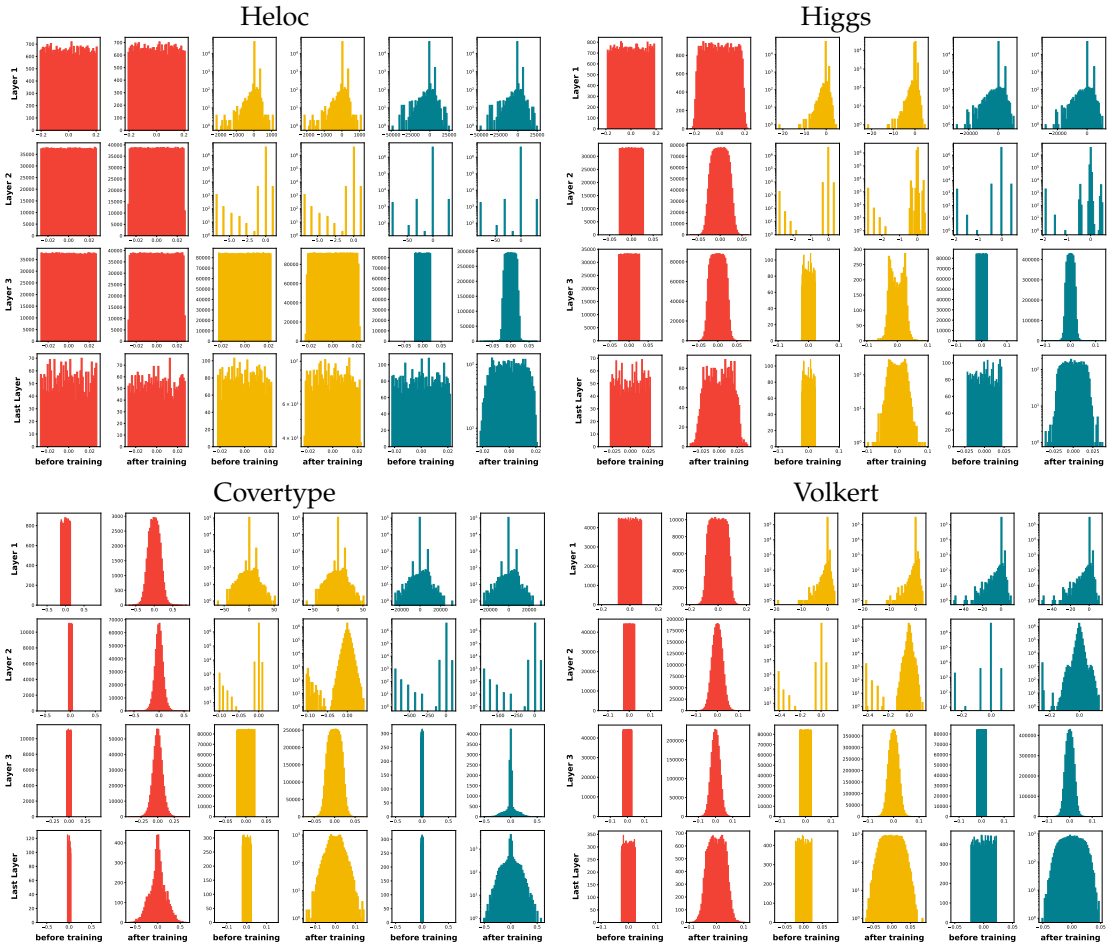


Figure S16: Histograms of the first three first and the last layers' weights before and after the MLP training on the Heloc, Higgs, Covertype and Volkert data sets. Comparison between random, RF and GBDT initializations.

Bibliography of the current chapter

Akiba, Takuya, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama (2019). "Optuna: A Next-generation Hyperparameter Optimization Framework". In: *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Arik, Sercan Ö and Tomas Pfister (2021). "Tabnet: Attentive interpretable tabular learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 8, pp. 6679–6687.

Biau, Gérard, Erwan Scornet, and Johannes Welbl (2019). "Neural random forests". In: *Sankhya A* 81.2, pp. 347–386.

Blundell, Charles, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra (2015). "Weight uncertainty in neural network". In: *International conference on machine learning*. PMLR, pp. 1613–1622.

Borisov, Vadim, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci (2021). *Deep Neural Networks and Tabular Data: A Survey*. DOI: 10.48550/ARXIV.2110.01889. URL: https://arxiv.org/abs/2110.01889.

Breiman, Leo (2001a). "Random forests". In: *Machine learning* 45.1, pp. 5–32.

— (2001b). "Random Forests". In: *Machine Learning* 45.1, pp. 5–32. ISSN: 1573-0565. DOI: 10.1023/A:1010933404324. URL: https://doi.org/10.1023/A:1010933404324.

Breiman, Leo, Jerome Friedman, Charles J Stone, and Richard A Olshen (1984). *Classification and regression trees*. CRC press.

Brent, Richard P (1991). "Fast training algorithms for multilayer neural nets". In: *IEEE Transactions on Neural Networks* 2.3, pp. 346–354.

Chen, Tianqi and Carlos Guestrin (2016). "Xgboost: A scalable tree boosting system". In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794.

Fan, Wei, Haixun Wang, Philip S Yu, and Sheng Ma (2003). "Is random model better? on its accuracy and efficiency". In: *Third IEEE International Conference on Data Mining*. IEEE, pp. 51–58.

Frankle, Jonathan and Michael Carbin (2018). "The lottery ticket hypothesis: Finding sparse, trainable neural networks". In: *arXiv preprint arXiv:1803.03635*.

Friedman, Jerome H (2001). "Greedy function approximation: a gradient boosting machine". In: *Annals of statistics*, pp. 1189–1232.

Glorot, Xavier and Yoshua Bengio (2010). "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, pp. 249–256.

Gorishniy, Yury, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko (2021). *Revisiting Deep Learning Models for Tabular Data*. DOI: 10.48550/ARXIV.2106.11959. URL: https://arxiv.org/abs/2106.11959.

Grinsztajn, Léo, Edouard Oyallon, and Gaël Varoquaux (2022). "Why do tree-based models still outperform deep learning on tabular data?" In: *arXiv preprint arXiv:2207.08815*.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2015). "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". In: *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034.

Ke, Guolin, Jia Zhang, Zhenhui Xu, Jiang Bian, and Tie-Yan Liu (2018). "TabNN: A universal neural network solution for tabular data". In.

Ke, Guolin et al. (2017). "Lightgbm: A highly efficient gradient boosting decision tree". In: *Advances in neural information processing systems* 30.

Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25.

LeCun, Yann, Yoshua Bengio, et al. (1995). "Convolutional networks for images, speech, and time series". In: *The handbook of brain theory and neural networks* 3361.10, p. 1995.

Liu, Pengfei, Xipeng Qiu, and Xuanjing Huang (2016). "Recurrent neural network for text classification with multi-task learning". In: *arXiv preprint arXiv:1605.05101*.

Mishkin, Dmytro and Jiri Matas (2015). "All you need is a good init". In: *arXiv preprint arXiv:1511.06422*.

Neal, Radford M (2012). *Bayesian learning for neural networks*. Vol. 118. Springer Science & Business Media.

Paszke, Adam et al. (2019). "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., pp. 8024–8035.

Richmond, David L, Dagmar Kainmueller, Michael Y Yang, Eugene W Myers, and Carsten Rother (2015). "Relating cascaded random forests to deep convolutional neural networks for semantic segmentation". In: *arXiv preprint arXiv:1507.07583*.

Rumelhart, David E, Geoffrey E Hinton, James L McClelland, et al. (1986). "A general framework for parallel distributed processing". In: *Parallel distributed processing: Explorations in the microstructure of cognition* 1.45-76, p. 26.

Rumelhart, David E, Geoffrey E Hinton, and Ronald J Williams (1985). *Learning internal representations by error propagation*. Tech. rep. California Univ San Diego La Jolla Inst for Cognitive Science.

Sethi, Ishwar Krishnan (1990). "Entropy nets: from decision trees to neural networks". In: *Proceedings of the IEEE* 78.10, pp. 1605–1613.

Shwartz-Ziv, Ravid and Amitai Armon (2022). "Tabular data: Deep learning is not all you need". In: *Information Fusion* 81, pp. 84–90. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2021.11.011. URL: https://www.sciencedirect.com/science/article/pii/S1566253521002360.

Somepalli, Gowthami, Micah Goldblum, Avi Schwarzschild, C. Bayan Bruss, and Tom Goldstein (2021). *SAINT: Improved Neural Networks for Tabular Data via Row Attention and Contrastive Pre-Training*. DOI: 10.48550/ARXIV.2106.01342. URL: https://arxiv.org/abs/2106.01342.

Sun, Ruo-Yu (2020). "Optimization for deep learning: An overview". In: *Journal of the Operations Research Society of China* 8.2, pp. 249–294.

Welbl, Johannes (2014). "Casting random forests as artificial neural networks (and profiting from it)". In: *German Conference on Pattern Recognition*. Springer, pp. 765–771.

Zhou, Z and J. Feng (2017). "Deep Forest: Towards An Alternative to Deep Neural Networks". In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 3553–3559.

Zhou, Zhi-Hua and Ji Feng (Jan. 2019). "Deep forest". In: *National Science Review* 6, pp. 74–86. DOI: 10.1093/nsr/nwy108.

Chapter 4

RF Interpolation

Abstract

Statistical wisdom suggests that very complex models, interpolating training data, will be poor at predicting unseen examples. Yet, this aphorism has been recently challenged by the identification of benign overfitting regimes, specially studied in the case of parametric models: generalization capabilities may be preserved despite model high complexity. While it is widely known that fully-grown decision trees interpolate and, in turn, have bad predictive performances, the same behavior is yet to be analyzed for Random Forests (RF). In this paper, we study the trade-off between interpolation and consistency for several types of RF algorithms. Theoretically, we prove that interpolation regimes and consistency cannot be achieved simultaneously for several non-adaptive RF. Since adaptivity seems to be the cornerstone to bring together interpolation and consistency, we study interpolating Median RF which are proved to be consistent in the interpolating regime. This is the first result conciliating interpolation and consistency for RF, highlighting that the averaging effect introduced by feature randomization is a key mechanism, sufficient to ensure the consistency in the interpolation regime and beyond. Numerical experiments show that Breiman's RF are consistent while exactly interpolating, when no bootstrap step is involved. We theoretically control the size of the interpolation area, which converges fast enough to zero, giving a necessary condition for exact interpolation and consistency to occur in conjunction.

4.1 Introduction

Random Forests (RF, Breiman 2001a) have proven to be very efficient algorithms, especially on tabular data sets. As any machine learning (ML) algorithm, Random Forests and Decision Trees have been analyzed and used according to the overfitting-underfitting trade-off. Regularization parameters have been introduced in order to control the variance while still reducing the bias. For instance, one can increase the variety of the constructed trees (by playing either with bootstrap samples or feature subsampling) or control the tree structure (by limiting either the number of points falling within each leaf or the maximum depth of all trees).

However, the paradigm stating that high model complexity leads to bad generalization capacity has been recently challenged: in particular, deeper and larger neural networks still empirically exhibit high predictive performances (Goodfellow et al. 2016). In such situations, overfitting can be qualified as "benign": complex models, possibly leading to interpolation of the

training examples, still generalize well on unseen data (Bartlett et al. 2021).

Regarding parametric methods, benign overfitting has been exhibited and well understood in linear regression (Bartlett et al. 2020; Tsigler et al. 2020; Liang et al. 2020b) and investigated in the context of neural networks (Belkin et al. 2019a). Many researchers currently study the *implicit bias* or *implicit regularization* of stochastic gradient (SGD) strategies used during neural network training: the optimization of an over-parametrized one-hidden-layer neural network via SGD will converge to a minimum of minimal norm with good generalization properties in a regression setting (Bach et al. 2021), or with maximal margin in a classification setting (Chizat et al. 2020).

Regarding non-parametric methods, practitioners have noticed the good performances of high-depth RFs for a long time (by default, several ML libraries such as the popular Scikit-Learn grow trees until pure leaves are reached). More recently, the use of interpolating (or very deep) trees for boosting and bagging methods has been discussed by Tang et al. 2018 and Wyner et al. 2017. While Tang et al. 2018 criticize the relevancy of interpolating random forests, Wyner et al. 2017 believe that the *self-averaging* process at hand in RF (or in boosting methods) also produces an implicit regularization that prevents the interpolating algorithm from overfitting. Note that the regularization properties of RF have also been studied in the light of their complexity (Buschjäger et al. 2021) and tree depth (Zhou et al. 2021). This phenomenon can be put in parallel with the results proved in Devroye et al. 1998 and Belkin et al. 2019b where they show that an interpolating kernel method using a singular kernel (similar to $K(x) = ||x||^{-\alpha}\mathbb{1}_{||x||\leqslant 1}$) is consistent, reaching minimax convergence rate for β-Hölder regular functions. More recently, Wang et al. 2022 showed the consistency of interpolating kernel methods, defined on Riemannian manifolds, whose kernels can be written as weighted random partition kernels on the sphere (similarly to the kernel random forest methods defined in Section 4.4).

Contributions and outline In this paper, we study the trade-off between interpolation and consistency in the context of regression, for different types of RF:

- Centered RF (Section 4.3). We prove theoretically that interpolation regimes and consistency cannot be achieved simultaneously for non-adaptive centered RF . The major problem arises from empty cells in tree partitions. Therefore, we also study a slightly modified Centered RF that does not take into account empty cells;

- Kernel RF (Section 4.4). We then study a more refined version of the CRF, the Kernel Random Forest (KeRF), built by averaging over all connected data points. By neglecting empty cells, this method is consistent for larger tree depths, but does not meet the exact interpolation requirement yet;

- Median RF (Section 4.5). Since adaptivity seems to be the cornerstone to conciliate interpolation and consistency, we study the interpolating Median RF, which is proved to be consistent in the exact interpolation regime . For the first time, it is shown that the averaging effect of the feature randomization inside RF (without boostrap) is sufficient to "average the noise out" (interpolating trees being sensitive to the noise), i.e. to decrease the variance towards 0. The bias of interpolating trees can be still classically controlled;

- Breiman RF (Section 4.6). Numerical experiments show that Breiman RF are consistent when exactly interpolating, i.e. when the whole data set is used to build each fully-grown tree (no bootstrap). It seems that the key randomization mechanism at work in RF is sufficient to reach consistency in spite of interpolation. Finally, we prove that the volume of the interpolation zone (where noise sensitivity is maximum) for an infinite Breiman RF tends to 0 at an exponential rate in the dimension d. This supports the idea that the decay of the interpolation volume could be fast enough to retrieve consistency despite interpolation.

		Conditions for consistency			
		Regardless of the noise scenario		In a noisy scenario	
		Managing the empty cells issue	Controlling the bias	Controlling the variance	Decreasing volume of the interpolation zone
Mean interpolation regime (non-adaptive RF)	Centered RF	×	✓	✓	
	Void-free CRF	✓	✓	?	
	Centered KeRF	✓	✓	✓	
Exact interpolation (semi-adaptive and adaptive RF)	Median RF	✓	✓	✓	✓
	Breiman RF	✓	?	?	✓

Figure 4.1: Summary of theoretical contributions.

Please refer to Figure 4.1 for an overview of theoretical contributions. All proofs and details on numerical experiments are given in Appendix S2 and S3.

4.2 Setting

Framework In a general non-parametric regression framework, we assume to be given a *training set* $\mathcal{D}_n := ((X_1, Y_1), ..., (X_n, Y_n))$, composed of i.i.d. copies of the generic random variable (X, Y), where the input X is assumed *throughout the paper* to be uniformly distributed over $[0,1]^d$, and $Y \in \mathbb{R}$ is the output. The underlying model is assumed to satisfy $Y = f^\star(X) + \varepsilon$, where $f^\star(x) = \mathbb{E}[Y|X = x]$ is the regression function and ε a random noise satisfying, almost surely, $\mathbb{E}[\varepsilon|X] = 0$ and $\mathbb{V}[\varepsilon|X] \leqslant \sigma^2 < \infty$, for some $\sigma^2 \geqslant 0$. Given an input $x \in [0,1]^d$, the goal is to estimate the associated response $f^\star(x)$. We measure the performance of an estimator f_n via its *excess risk*, defined as $\mathcal{R}(f_n) := \mathbb{E}\left[(f_n(X) - f^\star(X))^2\right]$, and its consistency property.

Definition 4.2.1 (Consistency)**.** *An estimator f_n is **consistent** when* $\lim_{n\to\infty} \mathcal{R}(f_n) = 0$.

Estimator A Random Forest (RF) is a predictor consisting of a collection of M randomized trees (see Breiman et al. 1984for details about decision trees). To build a forest, we generate $M \in \mathbb{N}^\star$ independent random variables $(\Theta_1, \ldots, \Theta_M)$, distributed as a generic random variable Θ, independent of $\mathcal{D}_n$. In our setting, Θ_j actually represents the successive random splitting directions and the resampling data mechanism in the j-th tree. The predicted value at the query point x given by the j-th tree is defined as

$$f_n(x, \Theta_j) = \sum_{i=1}^{n} \frac{\mathbb{1}_{X_i \in A_n(x,\Theta_j)} Y_i}{N_n(x, \Theta_j)} \mathbb{1}_{N_n(x,\Theta_j)>0} \,,$$

where $A_n(x, \Theta_j)$ is the cell containing x and $N_n(x, \Theta_j)$ is the number of points falling into $A_n(x, \Theta_j)$. The (finite) forest estimate then results from the aggregation of M trees:

$$f_{M,n}(x, \mathbf{\Theta}_M) = \frac{1}{M} \sum_{m=1}^{M} f_n(x, \Theta_m) \,,$$

where $\mathbf{\Theta}_M := (\Theta_1, ..., \Theta_M)$. By letting M tending to infinity, we can consider the *infinite* forest estimate, $f_{\infty,n}(x) = \mathbb{E}_\Theta[f_n(x, \Theta)]$, which has also played an important role in the theoretical

understanding of random forests (see Scornet 2016afor more details). Here, $\mathbb{E}_\Theta$ denotes the expectation w.r.t. Θ, conditional on $\mathcal{D}_n$.

Several random forests have been proposed depending on the type of randomness they contain (what Θ represents) and the type of decision trees they aggregate. Breiman forest is one of the most widely used RF, which exhibits excellent predictive performances. Unfortunately, its behavior is difficult to theoretically analyze, because of the numerous complex mechanisms involved in the predictive process (data resampling, data-dependent splits, split randomization). Therefore, in this paper, we simultaneously study the consistency and interpolation properties of different simplified versions of RF, both adaptive (i.e. when trees are built in a data-dependent manner) and non-adaptive.

All forests include a *depth* parameter, denoted k_n, which limits the maximum length of each branch in a tree, thus limiting the number of leaves (up to 2^{k_n}). In this work, we analyze how the tuning of k_n allows us to adjust the *consistency* and *interpolation* characteristics of the forest. The classical notion of (exact) interpolation is defined below.

Definition 4.2.2 ((Exact) interpolation)**.** *An estimator f_n is said to interpolate if for all training data (X_i, Y_i), we have $f_n(X_i) = Y_i$ almost surely.*

Recall that the prediction of a single tree at a point x is given by the average of all Y_i such that X_i is contained in the leaf of x. Therefore, each tree within a forest can be parameterized in order to interpolate: it is sufficient to grow the tree until pure leaves (i.e. leaves containing labels of the same values) are reached. In any regression model with continuous random noise, we have $Y_i \neq Y_j$ for all $i \neq j$ almost surely. Therefore, an interpolating tree is a tree that contains at most one point per leaf.

As the final prediction of the random forest is made by averaging the predictions of all its trees, if all trees interpolate, the random forest interpolates as well. Consequently, throughout all the theoretical analysis, we consider RF built without sub-sampling: each tree is built using the whole dataset instead of bootstrap samples as in standard RF. We will discuss the empirical effect of bootstrap in Section 4.6.

Remark 4.2.3. *In a classification setting, it is possible to obtain pure leaves with more than one point per cell (see Mentch et al. 2019 for more details).*

4.3 Centered RF

We start our analysis of interpolation and consistency of RF with the simple yet widely studied Centered Random Forest (CRF, see Biau 2012b). CRF are ensemble methods said to be non-adaptive since trees are built independently of the data: at each step of a centered tree construction, a feature is uniformly chosen among all possible d features and the split along the chosen feature is made at the center of the current cell. Then, the trees are aggregated to produce a CRF. Although simpler, the study of the mechanisms at hand in non-adaptive RF already provides good insights about the inner behaviour of more general RF.

4.3.1 Interpolation in CRF

Lemma 4.3.1. *The CRF $f_{M,n}^{\mathrm{CRF}}$ interpolates if and only if all trees that form the CRF interpolate.*

Since CRF construction is non-adaptive, it is impossible to enforce exactly one observation per leaf. Hence trees do not interpolate and in turn, the interpolation regime (Definition 4.2.2) cannot be satisfied for CRF. This leads us to examine a weaker notion of interpolation in probability.

Proposition 4.3.2 (Probability of interpolation for a centered tree)**.** *Denote $\mathcal{I}_T$ the event "a centered tree of depth k_n interpolates the training data". Then, for all $n \geqslant 3$, fixing $k_n = \lfloor \log_2(\alpha_n n) \rfloor$, with $\alpha_n \in \mathbb{N} \setminus \{0, 1\}$, one has*

$$e^{-\frac{n}{\alpha_n - 1}} \leqslant \mathbb{P}(\mathcal{I}_T) \leqslant e^{-\frac{n}{2(\alpha_n+1)}}.$$

According to Proposition 4.3.2, the probability that a tree interpolates tends to one if and only if $k_n = \lfloor \log_2(\alpha_n n) \rfloor$ with $\alpha_n = \omega(n)$[1]. Consequently, the regime $\alpha_n = \omega(n)$ completely characterizes the interpolation of a centered tree. Proposition 4.3.2 can be in turn used to control the interpolation probability of a centered RF.

Corollary 4.3.3 (Probability of interpolation for a CRF)**.** *We denote by $\mathcal{I}_F$ the event "a centered forest $f^{\mathrm{CRF}}_{M,n}(., \mathbf{\Theta}_M)$ interpolates". Then, for $k_n = \lfloor \log_2(\alpha_n n) \rfloor$ with $\alpha_n \geqslant 1$,*

$$\mathbb{P}(\mathcal{I}_F) \leqslant e^{-\frac{n}{2(\alpha_n+1)}}. \tag{4.1}$$

Therefore, the condition $\alpha_n = \omega(n)$ (corresponding to the interpolation of a single centered tree with high probability) is necessary to ensure that w.h.p., the RF interpolates. Our analysis stresses that a tree depth of at least $k_n = 2\log_2(n)$ is required to obtain tree/forest interpolation.

In fact, choosing k_n of the order of $\log_2(n)$ characterizes another type of interpolation regime. To see this, consider a centered tree of depth k, whose leaves are denoted $L_1, \ldots, L_{2^k}$. The number of points falling into the leaf L_i is denoted $N_n(L_i)$. Since X is uniformly distributed over $[0, 1]^d$, then, for all $i = 1, \ldots, 2^k$,

$$\mathbb{P}(X \in L_i) = \frac{1}{2^k} \quad \text{and} \quad \mathbb{E}[N_n(L_i)] = \frac{n}{2^k}. \tag{4.2}$$

Definition 4.3.4 (Mean interpolation regime)**.** *A CRF $f^{\mathrm{CRF}}_{M,n}$ satisfies the* mean interpolation regime *when each tree of $f_{M,n}$ has at least n leaves, i.e. if and only if $k_n \geqslant \log_2 n$.*

By Equation (4.2), the mean interpolation regime implies that for all leaves L_i, $\mathbb{E}[N_n(L_i)] \leqslant 1$: one could say that trees interpolate in expectation, in the mean interpolation regime.

4.3.2 Inconsistency of the Standard CRF

In both interpolation regimes (mean and in probability), trees need to be very deep, with a growing number of empty cells as n tends to infinity, eventually damaging the consistency of the overall CRF.

Proposition 4.3.5. *Suppose that $\mathbb{E}\left[f^\star(X)^2\right] > 0$ and set $\alpha > 0$. Then the infinite Centered Random Forest $f^{\mathrm{CRF}}_{\infty,n}$ of depth $k_n \geqslant \log_2 \alpha n$ is inconsistent.*

Proposition 4.3.5 emphasizes the poor generalization capacities of the interpolating CRF (under any interpolating regime), which could be expected given its non-adaptive construction. Indeed, the non-consistency of the CRF stems from the fact that the probability for a random point X to fall in an empty cell does not converge to zero, introducing an irreducible bias in the excess risk.

[1] i.e. α_n asymptotically dominates n.

4.3.3 Consistency of Void-Free CRF under the Mean Interpolation Regime

Since limiting the impact of empty cells seems crucial for consistency, we study a CRF that averages over non-empty cells only, which we call the *Void-Free CRF*. Note that predictions in empty leaves are arbitrary set to 0. Denoting $\Lambda_n(x, \mathbf{\Theta}_M)$ the number of non-empty leaves containing x in the forest with trees $\Theta_1, \ldots, \Theta_M$, the void-free CRF is written as

$$f_{M,n}^{\mathrm{VF}}(x, \mathbf{\Theta}_M) = \frac{1}{\Lambda_n(x, \mathbf{\Theta}_M)} \sum_{m=1}^{M} f_n(x, \Theta_m) \mathbb{1}_{N_n(x,\Theta_m)>0}.$$

The problematic terms that arise in the theoretical derivations of classical CRF vs. void-free CRF are of different natures: the probability $\mathbb{P}(N_n(X, \mathbf{\Theta}_M) = 0)$ of falling into an empty leaf in a random tree of an (infinite) CRF compared to the probability $\mathbb{P}[\forall m \in \{1, \ldots, M\}, N_n(X, \Theta_m) = 0]$ of falling into empty leaves in all trees in the (infinite) CRF. Lemma 4.3.6 below controls this last term.

Lemma 4.3.6. *Consider a finite void-free CRF $f_{M,n}^{\mathrm{VF}}(\cdot, \mathbf{\Theta}_M)$ of depth $k \in \mathbb{N}$. Let $x \in [0,1]^d$ and denote $\mathcal{E}_{M,n}(x)$ the event "for all $m \in \{1, \ldots, M\}, N_n(x, \Theta_m) = 0$". Then,*

$$\mathbb{P}(\mathcal{E}_{M,n}(x)) \leqslant e^{-\frac{kn}{2^{k+1}}} + e^{-Md^{-k}}. \tag{4.3}$$

Consequently, if $k = \lfloor \log_2(n) \rfloor$ and $M_n = \omega(n^{\log_2 d})$, then $\lim_{n\to\infty} \mathbb{P}(\mathcal{E}_{M_n,n}(x)) = 0$.

As previously, the infinite void-free CRF is defined as $f_{\infty,n}^{\mathrm{VF}}(x) = \mathbb{E}_\Theta[f_n(x, \Theta) | N_n(x, \Theta) > 0]$.

Theorem 4.3.7. *Assume that $f^\star$ has bounded partial derivatives. Then, the infinite void-free-CRF of depth $k = \lfloor \log_2 n \rfloor$ is consistent in a noiseless setting ($\sigma = 0$), and, for all $n > 1$,*

$$\mathcal{R}\left(f_{\infty,n}^{\mathrm{VF}}(X)\right) \leqslant C_d \left(\frac{n}{\log_2 n}\right)^{2\log_2\left(1-\frac{1}{2d}\right)} + (C_d + 2)\, n^{-1/(2\ln 2)},$$

where $C_d = 4d\left(\sum_{j=1}^d ||\partial f_j^\star||_\infty^2\right)$.

The overall rate is of order $O\left(n^{2\log(1-1/2d)}\right)$ which is a typical approximation rate for CRF, see Klusowski 2021a. As a matter of fact, Theorem 4.3.7 highlights that empty cells do not limit the performance of the void-free-CRF in the mean interpolation regime.

However, this construction introduces a conditioning over $N_n(x, \Theta) > 0$ that prevents us from bounding the variance in the case of noisy samples. Therefore, in the next section, we analyze Centered Kernel RF (KeRF) with a different aggregation rule (empty cells still being neglected).

4.4 Centered Kernel RF

As formalized in Geurts et al. 2006 and developed in Arlot et al. 2014, slightly modifying the aggregation rule of tree estimates provides a kernel-type estimator. Instead of averaging the predictions of all centered trees, the construction of a Kernel RF (KeRF) is performed by growing all centered trees and then averaging along all points contained in the leaves in which x falls, i.e.

$$f_{M,n}^{\mathrm{KeRF}}(x, \mathbf{\Theta}_M) := \frac{\sum_{i=1}^n Y_i \sum_{m=1}^M \mathbb{1}_{X_i \in A_n(x,\Theta_m)}}{\sum_{i=1}^n \sum_{m=1}^M \mathbb{1}_{X_i \in A_n(x,\Theta_m)}}.$$

One of the benefits of this construction is to limit the influence of empty cells, which can be harmful both for consistency and interpolation (see Section 4.3). As earlier, the infinite KeRF is defined as,

$$f_{\infty,n}^{\mathrm{KeRF}}(x) = \frac{\sum_{i=1}^n Y_i K_n(x, X_i)}{\sum_{i=1}^n K_n(x, X_i)},$$

where $K_n(x,z) = \mathbb{P}_\Theta\left[z \in A_n(x,\Theta)\right]$ is the probability that x and z are in the same cell w.r.t. a tree built according to Θ (see Scornet 2016b for details).

Interpolation conditions Since KeRF aggregates centered trees as CRF (but in a different way), the results of Section 4.3 can be extended to KeRF: (i) the mean interpolation regime is met for centered trees (hence for KeRF) when $k_n \geqslant \log_2 n$; (ii) a necessary condition to attain the KeRF interpolation in probability is $k_n > 2\log_2(n)$. One can note that the depths required for both interpolation regimes are still large, leading to as many empty cells for KeRF as for classical CRF but the aggregation rule is such that they are not taken into account in KeRF predictions, which gives hope that consistency could be preserved.

Consistency We study the convergence of the centered KeRF under the *mean interpolation regime*. To this end, we consider extra hypotheses on the noise and on the regularity of $f^\star$.

Theorem 4.4.1. *Assume that $f^\star$ is Lipschitz continuous and that the additive noise ε is a centered Gaussian variable independent from X with finite variance σ^2. Then, the risk of the infinite centered KeRF of depth $k_n = \lfloor \log_2(n) \rfloor$ verifies, for all $d > 5$, for all n large enough,*

$$\mathcal{R}(f_{\infty,n}^{\mathrm{KeRF}}) \leqslant 8L^2 d^2 n^{2\log_2(1-\frac{1}{d})} + C_d (\log_2 n)^{-\frac{d-5}{6}} (\log_2(\log_2 n))^{d/3},$$

where $C_d > 0$ is a constant dependent on σ^2 and made explicit in the proof.

Theorem 4.4.1 states that the infinite centered KeRF estimator is consistent as soon as $d > 5$, with a slow convergence rate of $\log(n)^{-(d-5)/6}$. The proof is based on the general paradigm of bias-variance trade-off and is adapted from Scornet 2016b. At first sight, one might think that the rate becomes better as the dimension d increases. However, the constant term highly depends on the dimension, so that the established bound should be regarded for a fixed d.

Choosing $k_n = \lfloor \log_2(n) \rfloor$ in Theorem 4.4.1 allows us to have a mean interpolation regime concomitant with consistency for KeRF, therefore highlighting that consistency and mean interpolation are compatible. This is not the case for CRF for which the mean interpolation regime forbids convergence (Proposition 4.3.5). If a "mean" overfitting regime is benign for the consistency of KeRF, it seems to be nonetheless malignant for the convergence rate. Indeed, Lin et al. 2006 provides a lower bound on the convergence rate of a deep non-adaptive RF (such as the CRF), scaling in $(\log n)^{-(d-1)}$. This leads us to believe that the convergence rate we obtain in Theorem 4.4.1 is marginally improvable.

Interpolation of kernel estimators has been recently studied with singular kernel by Belkin et al. 2019b. Since KeRF are kernel estimators, one can wonder how sharp is our bound (Theorem 4.4.1) compared to that of Belkin et al. 2019b, which is minimax. Due to the spikiness of the singular kernel studied in Belkin et al. 2019b, interpolation arises for any kernel bandwidth. The latter can be then tuned to reach minimax rates of consistency. The story is totally different for KeRF since interpolation occurs only for specific tree depths $k_n \geqslant \log(n)$ (where the depth parameter is closely related to the bandwidth of classical kernel estimates). Less latitude for choosing the depth then leads to sub-optimal rates of consistency (see Theorem 4.4.1). Of course, a better rate

of consistency in $O(n^{1/(3+d\log 2)})$ could be obtained as in Scornet 2016b when optimizing this depth parameter, but leaving the interpolation world.

We numerically assess the performance of KeRF in the mean interpolation regime (see Appendix S3).

4.5 Semi-Adaptive RF: Median RF

So far, consistency has been analyzed in the mean interpolation regime. What about consistency with exact RF interpolation? To analyze this phenomenon, we thus introduce semi-adaptive RF, Median RF, whose constructions depend on the training inputs X_i's (and not on the outputs Y_i's).

The Median RF, studied e.g. in Duroux et al. 2018; Klusowski 2021a, is composed of median trees that first randomly choose the direction to cut over and then cut at the median of the data points contained in the current cells. In our analysis, for any cell containing n_c observations, the median is set as the middle of the segment of two consecutive order statistics: $X_{(n_c/2)}$ and $X_{(n_c/2+1)}$ for an even number of observations, $X_{(\frac{n_c-1}{2})}$ and $X_{(\frac{n_c+1}{2})}$ otherwise.

4.5.1 Consistency

In order to obtain consistency for an adaptive RF, one needs to control two terms: the bias and the variance terms. On the one hand, the bias is roughly controlled by the diameter of the leaf times the supremum of the derivatives of $f^\star$ in the leaf. In the interpolating regime, the depth is maximum so the diameter of the leaf is minimum and therefore the bias is smoothly upper bounded.

On the other hand, a "low" depth regime is usually required to control the variance term, so that each leaf of each tree contains an infinite number of points when n tends to $+\infty$. This directly "averages the noise out" and decreases the variance towards 0 within each tree. However, in the interpolating case, each leaf contains only one point and we can only rely on the averaging effect of the RF, induced by the random splitting mechanism, to upper bound the variance. Studying the effect of the random splitting mechanism in full generality remains challenging. However, as increasing the dimension also increases the diversity of the trees within the RF, it should naturally be easier to control the variance of an interpolating Median RF in an asymptotic high-dimensional setting, as we prove and discuss in Appendix S2.5.

The following theorem establishes the consistency of the interpolating Median RF in the general setting of noisy data and fixed input dimension.

Theorem 4.5.1. *Suppose that $f^\star$ has bounded partial derivatives and that n is a power of two. Then, the infinite interpolating Median RF $f_{\infty,n}^{\mathrm{MedRF}}$ is consistent and verifies:*

$$\mathcal{R}\left(f_{\infty,n}^{\mathrm{MedRF}}\right) \leqslant C_1 d \left(\sum_{\ell=1}^{d} ||\partial_\ell f^\star||_\infty^2\right) \left(1-\frac{3}{4d}\right)^{\log_2 n} + \sigma^2 C_{2,d} (\log_2 n)^{-(d-1)/2},$$

where C_1 and $C_{2,d}$ are explicit constants, the former being independent of the dimension d (see the proof for the exact computations).

The control of the bias term follows the general approach used in Duroux et al. 2018 with substantial technical refinements. On the other hand, we propose a more general approach for the control of the variance inspired by Biau 2012b; Klusowski 2021a, where we derive explicit bounds specifically designed for Median RF. Note that the consistency achieved by Median RF

cannot be obtained for CRF under the interpolation regime due to the non-negligible probability of falling into empty cells (see Proposition 4.3.2).

Theorem 4.5.1 is the first result ensuring consistency of RF despite exact interpolation. It is even more impressive considering that bootstrap is off so that the averaging process in the RF is only due to feature subsampling. More specifically, when dealing with interpolating trees, the variance reduction does not come from averaging many points in the leaf of a given tree anymore (since the tree depth is no longer limited), but results from averaging single points from the leaves of different trees.

If interpolation remains compatible with consistency in the case of Median RF, it nevertheless damages the convergence rate. Indeed, it has been proved that, for all α small enough, the convergence rate of Median RF with trees of depth $k = (1-\alpha)\log_2(n)$ is $n^{-\alpha}$ (see Theorem 3 in Klusowski 2021a). In the case of interpolating Median RF, Theorem 4.5.1 highlights a phase transition when $k = \log_2(n)$, as the convergence rate is driven by the variance term, which is of order $(\log_2 n)^{-(d-1)/2}$. While being very slow, this rate is close to the lower bound $(\log_2 n)^{-(d-1)}$ established for non-adaptive interpolating RF (Lin et al. 2006). Actually, by assuming $\log_2(n) \geqslant d$, our proof can be directly modified so that our upper bound matches the lower bound of Lin et al. 2006 (using the second statement of Lemma S.1 in (Klusowski 2021a) instead of the first one).

Note that Theorem 4.5.1 does not contradict Proposition 1 in Tang et al. 2018, as the condition therein is not proved to be satisfied for interpolating median RF (nor for interpolating CRF).

We also provide numerical experiments (resp. Section 4.5.2 and S3.1) that illustrate the consistency of the interpolating Median RF.

4.5.2 Volume of the Interpolation Area

In this section, we aim at quantifying the volume of the interpolation area of a Median RF, which is a prerequisite for the RF consistency. To pursue our analysis, we first give a rigorous definition of the interpolation area.

Definition 4.5.2. *The **interpolation area** is the subspace of $[0,1]^d$ where the forest prediction depends only on one training point. For a given forest $f_{M,n}(.,\mathbf{\Theta}_M)$, the interpolation area is denoted by*[2]

$$\mathcal{A}(f_{M,n}(.,\mathbf{\Theta}_M)) = \left\{ x \in [0,1]^d, \exists! X_i \in \mathcal{D}_n, X_i \in \bigcap_{m=1}^{M} A_n(x,\Theta_m) \right\}.$$

The interpolation zone is highly dependent on both the geometry of the training points X_i's and the construction of the trees. Analyzing the interpolation area for a finite Median RF turns out to be quite a challenging task. Therefore, we focus our study on the *core interpolation area* $\mathcal{A}_{min}$ written as

$$\mathcal{A}_{min} = \bigcap_{M\in\mathbb{N},\mathbf{\Theta}_M} \mathcal{A}(f_{M,n}(.,\mathbf{\Theta}_M)).$$

The area $\mathcal{A}_{min}$ is the intersection of the interpolation zones of all possible forests, or equivalently of a forest containing all possible trees (and therefore all possible cuts). As an example note that in the case of median trees, every cut may occur with a positive probability. Therefore, $\mathcal{A}_{min}$ matches the volume of the interpolation area of an infinite Median RF. In the following proposition, we control the Lebesgue measure (denoted by μ) of the core interpolation area $\mathcal{A}_{min}^{\mathrm{MedRF}}$ of an infinite Median RF.

[2] the symbol $\exists!$ means "there exists a unique".

Proposition 4.5.3. *For all $n \geqslant 2$, for all $d \geqslant 2$, consider an infinite Median RF. Then,*

$$\mathbb{E}_{\mathcal{D}_n}\left[\mu(\mathcal{A}_{min}^{\mathrm{MedRF}})\right] \leqslant 2\left(\frac{2}{n}\right)^{d-1}.$$

The volume of the core interpolation area of an infinite Median RF tends to 0 polynomially in n and exponentially in d.

Remark 4.5.4. *Apart from a very restricted zone, the prediction of an infinite Median RF mostly relies on more than one training point. More specifically, this is a necessary condition for consistency: the volume of the area where the prediction involves only a finite number of points (a fortiori the interpolation zone) should tend to 0. Indeed, by decomposing the risk as $R(f_n(X)\mathbb{1}_{X\in\mathcal{A}_{min}^{\mathrm{MedRF}}}) + R(f_n(X)\mathbb{1}_{X\notin\mathcal{A}_{min}^{\mathrm{MedRF}}})$, the first term is at least of the order $\sigma^2\mu(\mathcal{A}_{min}^{\mathrm{MedRF}})$. Therefore, it is not possible to cancel out the noise of the training dataset when only a finite number of points is used for the prediction. The noise in such an area remains of order σ^2. Proposition 4.5.3 portends the predominant self-averaging property of adaptive RF, and hence underpins the idea of good capabilities of Median RF in interpolation regimes.*

4.6 Breiman RF

The widely-used Breiman RF is composed of several CART (Breiman et al. 1984), each one trained on a bootstrap sample, and for which the successive splitting directions and thresholds are chosen at each step (among a random subset of directions) in order to minimize the CART criterion. Breiman RF exhibit excellent predictive performance even if their adaptivity to the data remains a real hurdle to their theoretical analysis.

From the interpolation perspective, each CART being trained on a bootstrap sample, the RF interpolation is not ensured when considering fully-grown trees. Indeed, a tree cannot interpolate a point that is not chosen in the bootstrap step. For this reason, we focus our study on the volume of interpolation areas for Breiman RF without bootstrap and then analyze their empirical behavior in interpolating regimes through a battery of numerical experiments.

Interpolation As a Breiman RF is built using both the X_i's and the Y_i's, it is difficult to determine the depth necessary to reach the interpolation state. Depending on the data, the latter can be of the order $k \approx \log_2(n)$ in the best case, if each cut creates approximately two groups of the same size), or $k \approx n$ in the worst case, if only one point is separated from the others at each step (low signal-to-noise ratios situations, see e.g., Ishwaran 2015). Note that by omitting the bootstrap in the RF construction, the interpolation of Breiman RF directly results from aggregating fully-grown trees.

Volume of the interpolation zone As shown in the next proposition, the volume of the core interpolation area of Breiman RF tends to 0 as n tends to infinity.

Proposition 4.6.1. *Consider an infinite Breiman forest constructed without bootstrap with each tree fully developed. Suppose that for a given configuration of the training data, all cuts have a probability strictly greater than 0 to appear. Then, the volume of the minimal interpolation zone verifies*

$$\mathbb{E}\left[\mu(\mathcal{A}_{min})\right] \leqslant \frac{1}{n^{d-1}}\left(1-2^{-n}\right)^d.$$

Similarly to the Median RF, the bound on the interpolation volume for a Breiman forest enjoys the same order of decay, improved by a constant exponential in the dimension. Since predictions

cannot be accurate in the interpolation area in a noisy setting, it is necessary that the volume of this area decreases to zero in order to ensure the RF consistency (see Remark 4.5.4). Proposition 4.6.1 therefore suggests the good generalization properties of Breiman RF in interpolation regimes, as several training points are mostly used for prediction.

Setting the number of eligible features for splitting to 1 is sufficient to ensure the hypothesis on cuts in Proposition 4.6.1: one can obtain a tree in which all splits are performed along a single direction. Note that this result could be extended to any RF composed of DT where each cut has a non-zero probability to appear.

In Appendix S3, we numerically evaluate the volume of the interpolation zone and compare it to the theoretical bounds in Proposition 4.6.1.

Empirical study of consistency We now present an empirical study of Breiman RF consistency in interpolation regimes. In the theoretical analysis, we have focused on a specific type of Breiman RF (without bootstrap and a *max-features* parameter equal to 1). We now examine the characteristics of Breiman forests with their default parameters and study the regularization processes that limit the noise sensitivity in the interpolation regime. In order to reach a better estimation of the regression function, Breiman RF average several CARTs while introducing randomness in the construction of each tree to diversify them. The first randomization comes from the bootstrap: each tree is trained on a bootstrap sample (selecting n observations out of the n original ones, with replacement). The other randomization results from a random selection of splitting directions: at each node, a subset of $\{1, \dots, d\}$ of size max-features is randomly selected and the CART criterion is optimized along these directions only (setting *max-features* to 1 provides the maximum diversity whereas setting it to d results in the construction of a unique tree).

The benefit of these two aspects in the construction of the Breiman RF is numerically analyzed when using interpolating Breiman trees. In Figure 4.2, we measure the excess risk of two RFs with 2000 trees and max-depth= `None`, where for the first one, bootstrap is used and the max-features parameter is set to 1, whereas the second one excludes bootstrap and sets the max-features parameter to $\lceil d/3 \rceil$ (default value in `randomForest` in `R`).

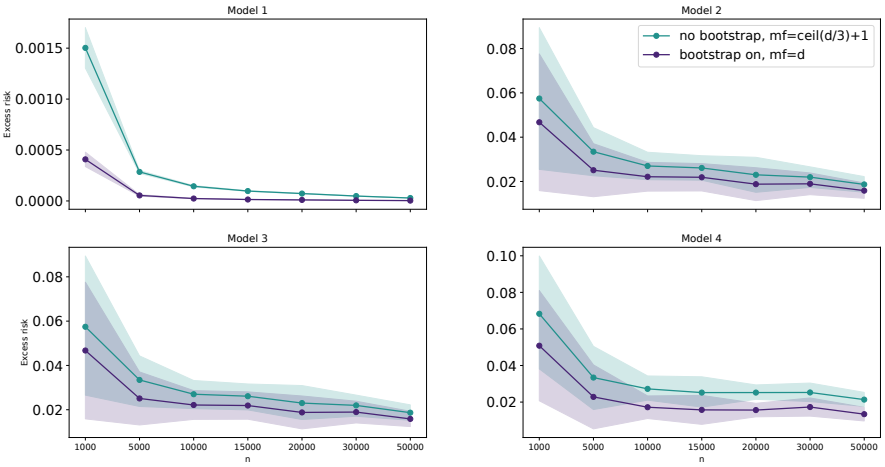


Figure 4.2: Consistency of two Breiman RF: excess risk w.r.t. sample size n. Mean over 10 tries (bold lines) and mean $\pm$ std (filled zone), when using 2000 trees per forest, and max-depth=None. See Appendix S3 for the model definitions.

In Figure 4.2, we observe that the excess risk decreases to 0 for all models and for both

forests. Indeed, each randomizing process alone induces enough diversity across trees for the self-averaging property to be efficient, resulting in the consistency of the overall forests (see also Scornet 2016a; Mentch et al. 2019; Mourtada et al. 2020 for insights about tree diversity in random forests).

However, when using bootstrap, consistency comes at the cost of leaving the interpolation regime, as only $2/3$ of the data are used in average to build each tree (see Figures S18, S19 in Section S3.2 for more details about the forest non-interpolation). In regards of this internal sampling selection, the aggregation of interpolating bagged trees results in smoothing the decision process of the entire forest, providing thereby a consistent but not interpolating estimate.

In turn, Breiman RF built with *max-features*$= \lceil d/3 \rceil$ seems consistent while preserving its interpolating behavior. Within this configuration, the final RF still interpolates the data but the volume of the interpolation zone is very small as shown in Figure S16. This is in line with the vision of a *locally spiky* estimator developed in Wyner et al. 2017 and Bartlett et al. 2021. Indeed, the influence of the averaging effect is locally null near the data training points, but increases with the distance from these points. Note that bootstrap and feature subsampling act differently. Bootstrap smoothens predictions by averaging different observations, even at points of the training set, which leads to an empty interpolation area. On the other hand, feature subsampling increases tree partition diversity, which reduces but does not annihilate the interpolation area of the overall forest.

Remark 4.6.2. *One of the advantage of using deep (interpolating) trees, compared to shallow ones, is that it allows the RF to build more diversified trees. Indeed, the number of possible trees roughly grows exponentially with regard to the depth (also depending on n, d and the* `max-features` *parameter). Especially when* `max-features` *is low, this should improve the averaging effect of the RF which is of particular interest when dealing with noisy data.*

In this regard, Breiman RF with *max-features*$= \lceil d/3 \rceil$ are similar to interpolating *spiky* non-singular kernel methods, as studied in Belkin et al. 2019b, except for the leeway allowed for the hyperparameters tuning. Indeed, as underlined for non-adaptive centered forests, the depth k_n (i.e. the tuned parameter) is constrained to a strict range to ensure both consistency and interpolation. This is not the case for singular kernel methods, as they interpolate regardless of the window parameter value.

4.7 Conclusion

In this paper, we study both empirically and theoretically the tradeoff between interpolation and consistency of different types of random forests: when dealing with non-adaptive RF (CRF), empty cells prevent consistency; so that aggregating only non-empty leaves (void-free CRF) leads to convergence rates, only in a noiseless scenario. In a noisy setting, the kernel RF aggregates leaves differently (also avoiding empty ones). For kernel RF, we establish a (slow) consistency rate in the mean interpolation regime. We then study semi-adaptive RF that are closer to those used in practice and that present the advantage of being able to exactly interpolate the training data. The convergence of the median RF in the exact interpolation regime is established, showing the power of such architecture (even when used without bootstrap). Our study also shows that a prerequisite for consistency is that the minimal interpolation zone tends to zero as n tends to infinity. We theoretically analyze this quantity for median and Breiman forests, emphasizing that interpolation might occur in conjunction with consistency if the volume of such areas vanishes fast enough. An experimental study supports the concomitance of consistency and interpolation in Breiman RF, when no bootstrap step is involved.

Contrary to Nadaraya-Watson methods involving singular kernels that interpolate regardless of the bandwidth parameter, RF interpolate only for a specific choice of the depth, thus restricting the regime in which interpolation and consistency occur in concordance. Overall, most simple RF versions were relevant to study RF consistency when the tree depth was limited but are not actually sufficient to handle deeper trees corresponding to interpolation regimes. For adaptive forests, increasing the tree depth towards the interpolation regime results in a reduced bias, and the variance reduction phenomenon only results from the split randomization effect. The higher the dimension, the more diversified the trees, the stronger the averaging effect and the variance reduction. Analyzing the strength of this phenomenon, which highly depends on the very shape of tree partitions, is the cornerstone to prove the consistency of adaptive RF in a general regression setting. We believe that interpolation remains benign for the consistency of adaptive RF, but can damage their convergence rate (this was the case for KeRF in the mean interpolation regime and for Median RF in the exact interpolation regime), at least when bootstrap is not used.

The analysis of the interpolation zone of RF introduced in this article is an important tool for the understanding of RF prediction in interpolation regimes. Indeed the volume of the interpolation area is actually a roundabout way to measure the diversity in the constructed trees: if this volume is high, all trees end up building similar partitions. This diversity measure could also be used as a regularization tool to reduce the RF complexity by keeping only the most uncorrelated trees (in terms of partition) in a PCA fashion.

S1 Summary of Contributions

		Conditions for consistency			
		Regardless of the noise scenario		In a noisy scenario	
		Managing the empty cells issue	Controlling the bias	Controlling the variance	Decreasing volume of the interpolation zone
Mean interpolation regime (non-adaptive RF)	Centered RF	✗	✓	✓	
	Void-free CRF	✓	✓	?	
	Centered KeRF	✓	✓	✓	
Exact interpolation (semi-adaptive and adaptive RF)	Median RF	✓	✓	✓	✓
	Breiman RF	✓	?	?	✓

Figure S3: Summary of theoretical contributions

S2 Proofs

S2.1 Reminders and Notations

Tree and RF estimator: We recall the prediction of the given by the j-th tree of the RF at point x:

$$f_n(x, \Theta_j) = \sum_{i=1}^{n} \frac{\mathbb{1}_{X_i \in A_n(x,\Theta_j)} Y_i}{N_n(x, \Theta_j)} \mathbb{1}_{N_n(x,\Theta_j)>0} \, ,$$

where $A_n(x, \Theta_j)$ is the cell containing x and $N_n(x, \Theta_j)$ is the number of points falling into $A_n(x, \Theta_j)$. It is also written as follows:

$$f_n(x, \Theta_j) = \sum_{i=1}^{n} W_{ni}(x, \Theta_j) Y_i,$$

where $W_{ni}(x, \Theta_j) = \frac{\mathbb{1}_{X_i \in A_n(x,\Theta_j)}}{N_n(x,\Theta_j)} \mathbb{1}_{N_n(x,\Theta_j)>0}$. The (finite) forest estimate then results from the aggregation of M trees:

$$f_{M,n}(x, \mathbf{\Theta}_M) = \frac{1}{M} \sum_{m=1}^{M} f_n(x, \Theta_m) \, ,$$

where $\mathbf{\Theta}_M := (\Theta_1, ..., \Theta_M)$.

S2.2 Proofs of Section 4.3 (Centered RF)

Proof of Lemma 4.3.1 (Link between tree and forest interpolation)

First, it is clear that if all trees of a forest interpolate, the forest interpolates. Now, suppose that the forest $f_{M,n}^{\mathrm{CRF}}$ interpolates a training point $X_s, s \in \{1, \dots, n\}$. Then, by definition of $f_{M,n}^{\mathrm{CRF}}$,

$$\begin{aligned} f_{M,n}^{\mathrm{CRF}}(X_s, \boldsymbol{\Theta_M}) &= \frac{1}{M} \sum_{j=1}^{M} \sum_{i=1}^{n} Y_i W_{ni}(X_s, \Theta_j) \\ &= \sum_{i=1}^{n} Y_i \left(\frac{1}{M} \sum_{j=1}^{M} W_{ni}(X_s, \Theta_j) \right) \\ &= Y_s, \end{aligned}$$

where $W_{ni}(X_s, \Theta_j) := \frac{\mathbb{1}_{X_i \in A_n(X_s, \Theta_j)}}{N_n(X_s, \Theta_j)} \mathbb{1}_{N_n(X_s, \Theta_j) > 0}$. Consequently,

$$f_{M,n}^{\mathrm{CRF}}(X_s, \boldsymbol{\Theta_M}) = Y_s \tag{4.4}$$

$$\iff \quad Y_s \left(\frac{1}{M} \sum_{j=1}^{M} W_{ns}(X_s, \Theta_j) - 1 \right) + \sum_{i \neq s} Y_i \left(\frac{1}{M} \sum_{j=1}^{M} W_{ni}(X_s, \Theta_j) \right) = 0. \tag{4.5}$$

For (4.5) to hold almost surely, it is necessary that it holds conditional on $X_1, \dots, X_n, \Theta_1, \dots, \Theta_M$. Since, for all $j \in \{1, \dots, M\}$, the terms $W_{ni}(X_s, \Theta_j)$ are measurable with respect to $X_1, \dots, X_n, \Theta_1, \dots, \Theta_M$ and Y_s is independent of $(Y_i, i \neq s)$ given $X_1, \dots, X_n, \Theta_1, \dots, \Theta_M$, equality (4.5) leads to, for all $i \neq s$,

$$\frac{1}{M} \sum_{j=1}^{M} W_{ns}(X_s, \Theta_j) = 1, \quad \text{and} \quad \frac{1}{M} \sum_{j=1}^{M} W_{ni}(X_s, \Theta_j) = 0. \tag{4.6}$$

Since all weights $W_{ni}(X, \Theta)$ take values in $[0, 1]$, we have, for all $j \in \{1, \dots, M\}$ and for all $i \neq s$

$$W_{ns}(X_s, \Theta_j) = 1 \quad \text{and} \quad W_{ni}(X_s, \Theta_j) = 0. \tag{4.7}$$

Finally, for all $j \in \{1, \dots, M\}$, the prediction of the jth tree at X_s is given by

$$f_n^{\mathrm{CRF}}(X_s, \Theta_j) = \sum_{i=1}^{n} W_{ni}(X_s, \Theta_j) Y_i \tag{4.8}$$

$$= Y_s, \tag{4.9}$$

and therefore all trees of the forest interpolate the point X_s.

Proof of Proposition 4.3.2 (Probability of interpolation for a centered tree)

As all the leaves have the same volume and the data points are independent and uniformly distributed, having at most one point per leaf is equivalent to distribute n balls into 2^k boxes containing at most one point with $2^k \geqslant n$ as can be seen on Figure S4. Recalling that $\mathcal{I}_T$ is the

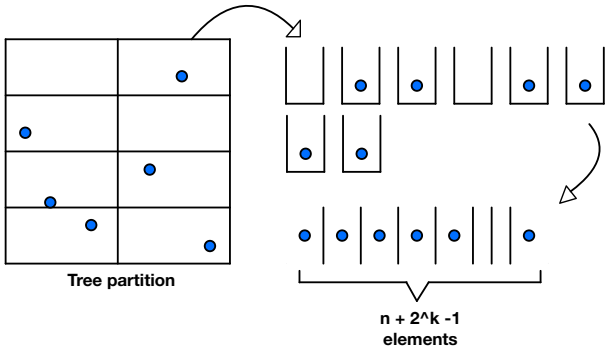


Figure S4: Computing the interpolation probability (depth $k = 3$, $n = 6$)

event "a centered tree of depth k_n interpolates the training data", we have

$$\begin{aligned}\mathbb{P}(\mathcal{I}_T) &= \frac{\binom{2^k}{n}}{\binom{n+2^k-1}{n}} \\ &= \frac{2^k!}{(2^k-n)!n!}\frac{n!(2^k-1)!}{(n+2^k-1)!} \\ &= \frac{2^k \times (2^k-1) \times \ldots \times (2^k-n+1)}{(2^k+n-1) \times (2^k+n-2) \times \ldots 2^k}.\end{aligned}$$

If we have $k = \log_2(\alpha_n n) \in \mathbb{N}$, we have

$$\mathbb{P}(\mathcal{I}_T) = \frac{\alpha_n n}{(\alpha_n+1)n-1} \cdot \frac{\alpha_n n - 1}{(\alpha_n+1)n-2} \cdots \frac{(\alpha_n-1)n+1}{\alpha_n n}.$$

In the general case where $k = \lfloor \log_2(\alpha_n n) \rfloor$, that is $\alpha_n n/2 \leqslant 2^k \leqslant \alpha_n n$, we can lower bound the probability of the event $\mathcal{I}_T$ as

$$\begin{aligned}\mathbb{P}(\mathcal{I}_T) = \frac{2^k \times (2^k-1) \times \ldots \times (2^k-n+1)}{(2^k+n-1) \times (2^k+n-2) \times \ldots 2^k} &\geqslant \left(\frac{2^k-n+1}{2^k+n-1}\right)^n \geqslant \left(\frac{2^k-n}{2^k+n}\right)^n \\ &\geqslant \exp\left(n \log\left(\frac{2^k-n}{2^k+n}\right)\right) \geqslant \exp\left(n \log\left(1 - \frac{2n}{2^k+n}\right)\right) \geqslant \exp\left(-n\left(\frac{2}{\frac{2^k}{n}-1}\right)\right) \\ &\geqslant \exp\left(-\left(\frac{4n}{\alpha_n - 2}\right)\right)\end{aligned}$$

since $\log(1-x) \geqslant -x/(1-x)$ and provided that $\alpha_n > 2$ for the last inequality. To upper bound the probability, note that, for all $r \in \{1, \ldots, \lfloor n/2 \rfloor\}$

$$\frac{2^k-n+r}{2^k+n-r} \leqslant \frac{2^k-n+\frac{n}{2}}{2^k+n-\frac{n}{2}-1} \leqslant \frac{2^k-\frac{n}{2}}{2^k+\frac{n}{2}-1},$$

and, for all $r \in \{1, ..., n\}$,

$$\frac{2^k - n + r}{2^k + n - r} \leqslant 1.$$

Therefore, one can also upper bound the probability as

$$\begin{aligned}
\mathbb{P}\left(\mathcal{I}_T\right) &= \frac{2^k \times (2^k - 1) \times \ldots \times (2^k - n + 1)}{(2^k + n - 1) \times (2^k + n - 2) \times \ldots 2^k} \\
&\leqslant \left(\frac{2^k - \frac{n}{2}}{2^k + \frac{n}{2} - 1}\right)^{\lfloor n/2 \rfloor} \\
&\leqslant \exp\left(\left\lfloor \frac{n}{2} \right\rfloor \log\left(1 - \frac{n-1}{2^k + \frac{n}{2} - 1}\right)\right) \\
&\leqslant \exp\left(-\left\lfloor \frac{n}{2} \right\rfloor \left(\frac{\frac{n}{2}}{2^k + \frac{n}{2} - 1}\right)\right) \\
&\leqslant \exp\left(-\left\lfloor \frac{n}{2} \right\rfloor \left(\frac{\frac{1}{2}}{\frac{2^k}{n} + \frac{1}{2}}\right)\right) \\
&\leqslant \exp\left(-\left\lfloor \frac{n}{2} \right\rfloor \left(\frac{1}{2\alpha_n + 1}\right)\right),
\end{aligned}$$

for all $n \geqslant 2$. Finally, for all $n \geqslant 2$, and for all $\alpha_n > 2$,

$$\exp\left(-\frac{4n}{\alpha_n - 2}\right) \leqslant \mathbb{P}\left(\mathcal{I}_T\right) \leqslant \exp\left(-\left\lfloor \frac{n}{2} \right\rfloor \left(\frac{1}{2\alpha_n + 1}\right)\right).$$

Proof of Corollary 4.3.3 (Probability of interpolation for a CRF)

As it is necessary for all trees to interpolation for the forest to interpolate, the probability that the forest interpolates is smaller than the probability that a single tree interpolates.

Proof of Proposition 4.3.5 (CRF inconsistency)

Let $f_{\infty,n}^{\mathrm{CRF}}$ be an infinite CRF with each tree of depth $k_n \geqslant \log_2(\alpha_n n)$, that is each tree has at least $\alpha_n n$ leaves, with $\alpha_n n > 1$. Let X be uniformly distributed on $[0,1]^d$. We write $\bar{f}_{n,\infty}^{\mathrm{CRF}}(X) = \mathbb{E}\left[f_{\infty,n}^{\mathrm{CRF}}(X) | X, X_1, ..., X_n\right]$. Then, denoting $\mathcal{E}$ the event "$N_{n,\infty}(X) = 0$" (or equivalently, "X falls

into a non-empty leaf"),

$$\mathcal{R}(f^{\mathrm{CRF}}_{\infty,n}(X)) = \mathbb{E}\left[\left(f^{\mathrm{CRF}}_{\infty,n}(X) - f^\star(X)\right)^2\right] \tag{4.10}$$
$$\geqslant \mathbb{E}\left[\left(\bar{f}^{\mathrm{CRF}}_{n,\infty}(X) - f^\star(X)\right)^2\right] \tag{4.11}$$
$$= \mathbb{E}\left[\left(\sum_{i=1}^n \mathbb{E}_\Theta\left[W_{ni}(X,\Theta) f^\star(X_i)\right] - \left(\mathbb{1}_{\mathcal{E}} + \mathbb{1}_{\mathcal{E}^c}\right) f^\star(X)\right)^2\right] \tag{4.12}$$
$$= \mathbb{E}\left[\left(\mathbb{1}_{\mathcal{E}^c}\sum_{i=1}^n \mathbb{E}_\Theta\left[W_{ni}(X,\Theta)\left(f^\star(X_i) - f^\star(X)\right)\right] - \mathbb{1}_{\mathcal{E}} f^\star(X)\right)^2\right] \tag{4.13}$$
$$\geqslant \mathbb{E}\left[f^\star(X)^2 \mathbb{1}_{\mathcal{E}}\right] \tag{4.14}$$
$$\geqslant \mathbb{E}\left[f^\star(X)^2 \mathbb{P}\left(\mathcal{E}|X\right)\right]. \tag{4.15}$$

Besides,

$$\mathbb{P}\left(\mathcal{E}|X\right) = \mathbb{P}\left(N_{n,\infty}(X) = 0|X\right) \tag{4.16}$$
$$\geqslant \left(1 - \frac{1}{\alpha_n n}\right)^n, \tag{4.17}$$

and as $\log(1 - 1/x) \geqslant -\frac{1}{x-1}$ for $x > 1$,

$$\left(1 - \frac{1}{\alpha_n n}\right)^n = e^{n\log\left(1-\frac{1}{\alpha_n n}\right)} \tag{4.18}$$
$$\geqslant e^{-\frac{n}{\alpha_n n - 1}}. \tag{4.19}$$

Thus,

$$\mathcal{R}(f^{\mathrm{CRF}}_{\infty,n}(X)) \geqslant e^{-\frac{n}{\alpha_n n-1}} \mathbb{E}\left[f^\star(X)^2\right], \tag{4.20}$$

which tends to 0 if and only if α_n tends to zero as n tends to infinity. Since, by assumptions, α_n does not tend to zero and $\mathbb{E}\left[f^\star(X)^2\right] > 0$, the infinite CRF is inconsistent.

Proof of Lemma 4.3.6 (Probability of falling into an empty cell of the void-free CRF)

Recall that $\mathcal{E}_{M,n}(x)$ is the event "for all $m \in \{1,\dots,M\}, N_n(x,\Theta_m) = 0$". We have

$$\mathcal{E}_{M,n}(x) = \bigcap_{j=1}^M \left\{N_n(x,\Theta_j) = 0\right\}. \tag{4.21}$$

Given a dataset, we distinguish two situations: either x falls into an area where it cannot be connected to a point X_i for any tree, or the dataset is such that x could be connected to a point X_i for a certain configuration of cuts within a tree. We write $\mathcal{E}_{1,n}(x)$ the ($\mathcal{D}_n$-measurable) event $\{\forall\theta, N_n(x,\theta) = 0\}$. Consequently, we have $\mathcal{E}_{1,n}(x)^c = \{\exists\,\theta, N_n(x,\theta) \neq 0\}$. Using these notations,

we obtain

$$\mathbb{P}\left(\mathcal{E}_{M,n}(x)\right) = \mathbb{P}\left(\mathcal{E}_{M,n}(x) \cap \mathcal{E}_{1,n}(x)\right) + \mathbb{P}\left(\mathcal{E}_{M,n}(X) \cap \mathcal{E}_{1,n}(x)^c\right) \tag{4.22}$$

$$= \mathbb{P}\left(\mathcal{E}_{1,n}(x)\right) + \mathbb{P}\left(\mathcal{E}_{M,n}(x) \cap \mathcal{E}_{1,n}(x)^c\right) \tag{4.23}$$

where the first probability term of the second line is a probability taken over $\mathcal{D}_n$ only, since $\mathcal{E}_{1,n}(x)$ does not depend on Θ. We control this probability thanks to the following Lemma.

Lemma S1. *For all $x \in [0,1]^d$, we let $\mathcal{E}_{1,n}(x)$ be the event $\{\forall \theta, N_n(x,\theta) = 0\}$. Then, we have*

$$\mathbb{P}\left(\mathcal{E}_{1,n}(x)\right) \leqslant e^{-\frac{n}{2^{k+1}}}.$$

Proof. Let $x \in [0,1]^d$. The event $\mathcal{E}_{1,n}(x)$ happens if all the points of the dataset fall into parts of the space that cannot connect to x for any tree. In order to compute its probability, we compute the size of the *connection area of* x for trees of depth k, denoted

$$Z_{c,k}(x) = \left\{z \in [0,1]^d \ : \exists \theta, z \in A_n(x,\theta)\right\}. \tag{4.24}$$

We recall that trees are built independently from the dataset and that all cuts are made in the middle of the current node for a uniformly chosen feature at each step. We denote $A(k_1, ..., k_d, x)$ the cell of x obtained by cutting k_j times along feature $X^{(j)}$ for all $j \in \{1, \dots, d\}$. Then, the volume of the connection area $Z_{c,k}$ of x is

$$\mu(Z_{c,k}(x)) = \mu\left(\bigcup_{\substack{0 \leqslant k_1, \dots, k_d \leqslant k \\ \sum_j k_j = k}} A(k_1, ..., k_d, x)\right) \tag{4.25}$$

$$\geqslant \mu\left(\bigcup_{\substack{0 \leqslant k_1, k_2 \leqslant k \\ k_1 + k_2 = k}} A(k_1, k_2, 0, ..., 0, x)\right). \tag{4.26}$$

By σ-additivity of μ,

$$\mu\left(\bigcup_{\substack{0 \leqslant k_1, k_2 \leqslant k \\ k_1 + k_2 = k}} A(k_1, k_2, 0, ..., 0, x)\right)$$

$$= \mu\Big(A(k, 0, ..., 0, x)\Big) + \sum_{j=1}^{k} \mu\left(A(k-j, j, 0, ..., 0, x) \setminus \bigcup_{\ell=0}^{j-1} A(k-\ell, \ell, 0, ..., 0, x)\right). \tag{4.27}$$

Given the shape of the cells $A(k-j, j, 0, ..., 0, x)$, for all $j \in \{1, \dots, d\}$, we have (see Figure S5)

$$A(k-j, j, 0, ..., 0, x) \setminus \bigcup_{\ell=0}^{j-1} A(k-\ell, \ell, 0, ..., 0, x)$$

$$= A(k-j, j, 0, ..., 0, x) \setminus A(k-j+1, j-1, 0, ..., 0, x). \tag{4.28}$$

Furthermore, note that, for all $j \in \{1, \dots, d\}$, the volume of each cell $A(k-j+1, j-1, 0, ..., 0, x)$ is 2^{-k} (since k cuts have been performed). Therefore, for all $j \in \{1, \dots, k\}$,

1. $\mu(A(k-j,j,0,...,0,x)) = \mu(A(k-j+1,j-1,0,...,0,x)) = 2^{-k}$
2. $\mu\big((A(k-j,j,0,...,0,x) \cap A(k-j+1,j-1,0,...,0,x)\big) = \frac{\mu(A(k-j,j,0,...,0,x))}{2}$ as can be seen on Figure S5.

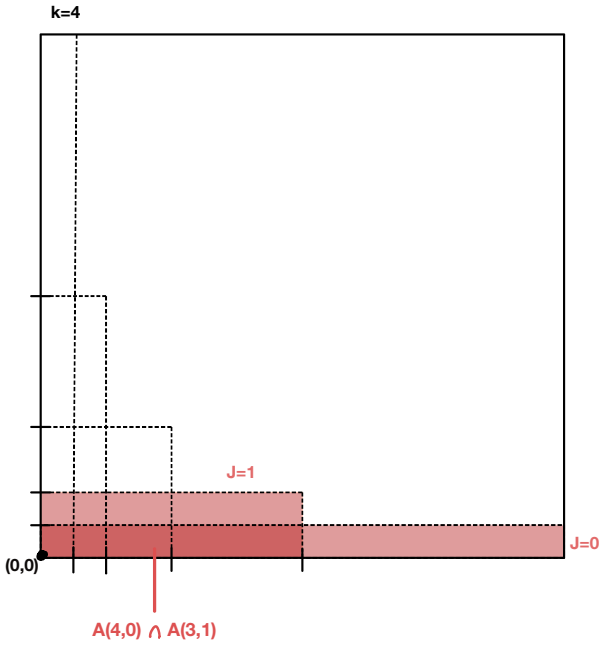


Figure S5: Volume of leaf intersection $\mu\big((A(k-j,j,x) \cap A(k-j+1,j-1,x)\big)$ in dimension 2 with $x = (0,0)$, $k = 4$ cuts and $j \in \{0,1\}$.

We deduce from these facts that, for all j,

$$\mu\big(A(k-j,j,0,...,0,x) \setminus A(k-j+1,j-1,0,...,0,x)\big) = \frac{\mu(A(k-j,j,0,...,0,x))}{2} \tag{4.29}$$

$$= 2^{-(k+1)} \tag{4.30}$$

Hence, combining equations (4.27), (4.28) and (4.29), we have

$$\mu\left(\bigcup_{\substack{0\leqslant k_1,k_2\leqslant k \\ k_1+k_2=k}} A(k_1,k_2,0,...,0,x)\right) = 2^{-k} + k2^{-(k+1)}. \tag{4.31}$$

Consequently, using inequality (4.26),

$$\mu(Z_{c,k}(x)) \geqslant k2^{-(k+1)}. \tag{4.32}$$

Finally, as the X_i's are uniformly distributed on $[0,1]^d$ and $\mathcal{E}_{1,n}(x)$ is realized when none of

the X_is fall into $Z_{c,k}(x)$,

$$\begin{aligned}
\mathbb{P}\left(\mathcal{E}_{1,n}(x)\right) &= \mathbb{P}\left(\forall i \in \{1,\dots,n\}, X_i \notin Z_{c,k}(x)\right) && (4.33)\\
&= \left(1-\mu(Z_{c,k}(x))\right)^n && (4.34)\\
&\leqslant \left(1-k2^{-(k+1)}\right)^n && (4.35)\\
&= e^{n\log(1-k2^{-(k+1)})} && (4.36)\\
&\leqslant e^{-\frac{kn}{2^{k+1}}}. && (4.37)
\end{aligned}$$

□

Regarding the second term of (4.23), we have

$$\begin{aligned}
\mathbb{P}\left(\mathcal{E}_{M,n}(x)\cap\mathcal{E}_{2,n}(x)\right) &= \mathbb{P}\left(\left(\bigcap_{j=1}^{M} N_n(x,\Theta_j)=0\right)\bigcap\left(\exists i\in\{1,\dots,n\}, X_i\in Z_{c,k}(x)\right)\right) && (4.38)\\
&= \mathbb{E}\left[\mathbb{E}\left[\mathbb{1}_{\exists i\in\{1,\dots,n\}, X_i\in Z_{c,k}(x)}\mathbb{1}_{\bigcap_{j=1}^{M} N_n(x,\Theta_j)=0}|\mathcal{D}_n\right]\right] && (4.39)\\
&= \mathbb{E}\left[\mathbb{1}_{\exists i\in\{1,\dots,n\}, X_i\in Z_{c,k}(x)}\mathbb{P}\left(\bigcap_{j=1}^{M} N_n(x,\Theta_j)=0|\mathcal{D}_n\right)\right] && (4.40)\\
&= \mathbb{E}\left[\mathbb{1}_{\exists i\in\{1,\dots,n\}, X_i\in Z_{c,k}(x)}(1-p_n)^M\right] && (4.41)
\end{aligned}$$

where $p_n = \mathbb{P}_\Theta\left(N_n(x,\Theta)>0|\mathcal{D}_n\right)$ and where the last line is obtained by independence of the Θ_j's conditionally on $\mathcal{D}_n$. Note that, if $\exists i\in\{1,\dots,n\}, X_i\in Z_{c,k}(x)$, then $p_n \geqslant d^{-k}$ since a tree connects x and a point in $Z_{c,k}(x)$ with probability at least d^{-k} (i.e. by choosing the right cut at each step). Hence,

$$\mathbb{1}_{\exists i\in\{1,\dots,n\}, X_i\in Z_{c,k}(x)}(1-p_n)^M \leqslant (1-d^{-k})^M, \tag{4.42}$$

which leads to

$$\begin{aligned}
\mathbb{P}\left(\mathcal{E}_{M,n}(x)\cap\mathcal{E}_{1,n}(x)^c\right) &\leqslant \left(1-d^{-k}\right)^M && (4.43)\\
&\leqslant e^{-Md^{-k}}. && (4.44)
\end{aligned}$$

Finally, gathering Lemma S1 and inequality (4.44) yields

$$\mathbb{P}\left(\mathcal{E}_{M,n}(x)\right) \leqslant e^{-\frac{kn}{2^{k+1}}} + e^{-Md^{-k}}. \tag{4.45}$$

Proof of Proposition 4.3.7 (Consistency of void-free-CRF in a noiseless setting)

Recall that, in a noiseless setting (that is, for all i, $Y_i = f^\star(X_i)$), the risk of the Void-free CRF can be written as

$$\begin{aligned}
&\mathbb{E}\left[\left(f^{\mathrm{VF}}_{\infty,n}(X)-f^\star(X)\right)^2\right]\\
&= \mathbb{E}\left[\left(\frac{\mathbb{1}_{\mathbb{P}_\Theta(N_n(X,\Theta)>0)>0}}{\mathbb{P}_\Theta\left(N_n(X,\Theta)>0\right)}\sum_{i=1}^{n} f^\star(X_i)\mathbb{E}_\Theta[W_{ni}(X,\Theta)\mathbb{1}_{N_n(X,\Theta)>0}]-f^\star(X)\right)^2\right].
\end{aligned}$$

We decompose $f^\star(X)$ as

$$f^\star(X) = \left(\mathbb{1}_{\mathbb{P}_\Theta(N_n(X,\Theta)>0)>0} + \mathbb{1}_{\mathbb{P}_\Theta(N_n(X,\Theta)>0)=0}\right) f^\star(X)$$

in order to write

$$\begin{aligned}
&\mathbb{E}\left[\left(\frac{\mathbb{1}_{\mathbb{P}_\Theta(N_n(X,\Theta)>0)>0}}{\mathbb{P}_\Theta\left(N_n(X,\Theta>0)\right)}\sum_{i=1}^n f^\star(X_i)\mathbb{E}_\Theta[W_{ni}(X,\Theta)\mathbb{1}_{N_n(X,\Theta)>0}] - f^\star(X)\right)^2\right] \\
&= \mathbb{E}\left[\left(\frac{\mathbb{1}_{\mathbb{P}_\Theta(N_n(X,\Theta)>0)>0}}{\mathbb{P}_\Theta\left(N_n(X,\Theta>0)\right)}\sum_{i=1}^n \left(f^\star(X_i)-f^\star(X)\right)\mathbb{E}_\Theta[W_{ni}(X,\Theta)\mathbb{1}_{N_n(X,\Theta)>0}]\right.\right. \\
&\left.\left. -f^\star(X)\mathbb{1}_{\mathbb{P}_\Theta(N_n(X,\Theta)>0)=0}\right)^2\right] \\
&\leqslant 2\mathbb{E}\left[\left(\frac{\mathbb{1}_{\mathbb{P}_\Theta(N_n(X,\Theta)>0)>0}}{\mathbb{P}_\Theta\left(N_n(X,\Theta>0)\right)}\sum_{i=1}^n \left(f^\star(X_i)-f^\star(X)\right)\mathbb{E}_\Theta[W_{ni}(X,\Theta)\mathbb{1}_{N_n(X,\Theta)>0}]\right)^2\right] \\
&\quad + 2\mathbb{E}\left[\left(f^\star(X)\mathbb{1}_{\mathbb{P}_\Theta(N_n(X,\Theta)>0)=0}\right)^2\right] \qquad (4.46)
\end{aligned}$$

The second term of the last inequality verifies

$$\mathbb{E}\left[\left(f^\star(X)\mathbb{1}_{\mathbb{P}_\Theta(N_n(X,\Theta)>0)=0}\right)^2\right] \leqslant ||f^\star||_\infty^2 \mathbb{P}\left(\mathbb{P}_\Theta\left(N_n(X,\Theta)>0\right)=0\right). \qquad (4.47)$$

The event $\{\mathbb{P}_\Theta\left(N_n(X,\Theta)>0\right)=0\}$ is $(X,\mathcal{D}_n)$-measurable, it corresponds to the situation where for any θ, $N_n(X,\theta)=0$, i.e. the dataset is such that it is impossible for a tree to connect X with one of the X_i's. This probability is controlled by Lemma S1:

$$\mathbb{P}\left(\mathbb{P}_\Theta\left(N_n(X,\Theta)>0\right)=0\right) \leqslant e^{-\frac{kn}{2^{k+1}}}.$$

Denoting by $\mu\left(A_n^{(j)}(x,\Theta)\right)$ the length of the jth side of the cell containing x and following a computation from Klusowski 2021a,

$$\begin{aligned}
&\sum_{i=1}^n W_{ni}(X,\Theta)|f^\star(X)-f^\star(X_i)|\mathbb{1}_{N_n(X,\Theta)>0} \\
&\leqslant \sum_{i=1}^n W_{ni}(X,\Theta)\left(\sum_{j=1}^d ||\partial_j f^\star||_\infty |X_i^{(j)}-X^{(j)}|\right)\mathbb{1}_{N_n(X,\Theta)>0} \qquad (4.48) \\
&\leqslant \sum_{i=1}^n W_{ni}(X,\Theta)\mathbb{1}_{N_n(X,\Theta)>0}\sum_{j=1}^d ||\partial_j f^\star||_\infty (b_j-a_j) \qquad (4.49) \\
&\leqslant \mathbb{1}_{N_n(X,\Theta)>0}\sum_{j=1}^d ||\partial_j f^\star||_\infty \mu\left(A_n^{(j)}(X,\Theta)\right). \qquad (4.50)
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\mathbb{E}\left[\left(f_{\infty,n}^{\mathrm{VF}}(X)-f^\star(X)\right)^2\right]\\
&\leqslant 2\mathbb{E}\left[\left(\frac{1}{\mathbb{P}_\Theta(N_n(X,\Theta)>0)}\sum_{j=1}^d ||\partial_j f||_\infty \mathbb{E}_\Theta\left[\mathbb{1}_{N_n(X,\Theta)>0}\mu\left(A_n^{(j)}(X,\Theta)\right)\right]\right)^2\right]\\
&\quad + 2e^{-\frac{kn}{2^{k+1}}} \qquad (4.51)\\
&\leqslant 2d\sum_{j=1}^d ||\partial f_j^\star||_\infty^2 \mathbb{E}\left[\frac{1}{\mathbb{P}_\Theta(N_n(X,\Theta)>0)^2}\mathbb{E}_\Theta\left[\mathbb{1}_{N_n(X,\Theta)>0}\mu\left(A_n^{(j)}(X,\Theta)\right)\right]^2\right]\\
&\quad + 2e^{-\frac{kn}{2^{k+1}}}. \qquad (4.52)
\end{aligned}$$

Note that the length $\mu\left(A_n^{(j)}(X,\Theta)\right)$ of the j-th side of the cell $A_n(X,\Theta)$ and the event $\{N_n(X,\Theta)>0\}$ are not independent conditional on $X_1,...,X_n,X$. Indeed, given the geometry of the dataset, it is possible that cutting along the jth direction isolates X from the dataset. Therefore its length should be computed conditional on the event $\{N_n(X,\Theta)>0\}$.

To this aim, we denote for all $\kappa\in\mathbb{N}$, $A_{n,\kappa}(X,\Theta)$ the cell containing X at depth κ in a centered tree built with the extra randomness Θ. Conditional on $N_n(X,\Theta)>0$, the jth direction can be chosen to split along if and only if it does not isolate X from the points of the dataset. Thus, we denote by $E_{n,\kappa}(j,X,\Theta)$ the event "In a centered tree built with the randomized cuts Θ, at depth κ, splitting the cell containing X along the jth direction does not isolate X". Then,

$$\begin{aligned}
\mathbb{E}_\Theta\left[\mathbb{1}_{N_n(X,\Theta)>0}\mu\left(A_n^{(j)}(X,\Theta)\right)\right] &= \mathbb{E}_\Theta\left[\mathbb{1}_{N_n(X,\Theta)>0}\mu\left(A_n^{(j)}(X,\Theta)\right)\left(\mathbb{1}_{E_{n,\kappa}(j,X,\Theta)^c}+\mathbb{1}_{E_{n,\kappa}(j,X,\Theta)}\right)\right] \qquad (4.53)\\
&\leqslant \mathbb{E}_\Theta\left[\mathbb{1}_{N_n(X,\Theta)>0}\mathbb{1}_{E_{n,\kappa}(j,X,\Theta)^c}\right] \qquad (4.54)\\
&\quad + \mathbb{E}_\Theta\left[\mathbb{1}_{N_n(X,\Theta)>0}\mathbb{1}_{E_{n,\kappa}(j,X,\Theta)}\mu\left(A_n^{(j)}(X,\Theta)\right)\right], \qquad (4.55)
\end{aligned}$$

since $\mu\left(A_n^{(j)}(X,\Theta)\right)\leqslant 1$. We denote $A_{n,\kappa}^{(j),\mathrm{left}}(X,\Theta)$ (resp. $A_{n,\kappa}^{(j),\mathrm{right}}(X,\Theta)$) the left (resp. right) daughter of the cell $A_{n,\kappa}(X,\Theta)$ that has been split along the jth direction (note that the whole cell is considered here, not only the projection on the j-th side). Then,

$$\begin{aligned}
&\mathbb{E}_\Theta\left[\mathbb{1}_{N_n(X,\Theta)>0}\mathbb{1}_{E_{n,\kappa}(j,X,\Theta)^c}\right]\\
&= \mathbb{P}_\Theta\left(E_{n,\kappa}(j,X,\Theta)^c\middle|N_n(X,\Theta)>0\right)\mathbb{P}_\Theta\left(N_n(X,\Theta)>0\right) \qquad (4.56)\\
&= \mathbb{P}_\Theta\left(\left(N_n(A_{n,\kappa}^{(j),\mathrm{left}}(X,\Theta))=0\right)\cap\left(X\in A_{n,\kappa}^{(j),\mathrm{right}}(X,\Theta)\right)\middle|N_n(X,\Theta)>0\right)\mathbb{P}_\Theta\left(N_n(X,\Theta)>0\right)\\
&\quad + \mathbb{P}_\Theta\left(\left(N_n(A_{n,\kappa}^{(j),\mathrm{right}}(X,\Theta))=0\right)\cap\left(X\in A_{n,\kappa}^{(j),\mathrm{left}}(X,\Theta)\right)\middle|N_n(X,\Theta)>0\right)\mathbb{P}_\Theta\left(N_n(X,\Theta)>0\right) \qquad (4.57)\\
&\leqslant 2\mathbb{P}_\Theta\left(N_n(A_{n,\kappa}^{(j),\mathrm{left}}(X,\Theta))=0\middle|N_n(X,\Theta)>0\right)\mathbb{P}_\Theta\left(N_n(X,\Theta)>0\right). \qquad (4.58)
\end{aligned}$$

Moreover,

$$\begin{aligned}&\mathbb{E}_\Theta\left[\mathbb{1}_{N_n(X,\Theta)>0}\mathbb{1}_{E_{n,\kappa}(j,X,\Theta)}\mu\left(A_n^{(j)}(X,\Theta)\right)\right]\\&\leqslant\mathbb{E}_\Theta\left[\mu\left(A_n^{(j)}(X,\Theta)\right)\big|E_{n,\kappa}(j,X,\Theta),N_n(X,\Theta)>0\right]\mathbb{P}_\Theta\left(N_n(X,\Theta)>0\right).\end{aligned}\tag{4.59}$$

Denoting $K_{j,\kappa}(X,\Theta)$ the number of splits made on feature j up to depth κ to produce the cell containing X, we obtain

$$\begin{aligned}&\mathbb{E}_\Theta\left[\mu\left(A_n^{(j)}(X,\Theta)\right)\big|E_{n,\kappa}(j,X,\Theta),N_n(X,\Theta)>0\right]\\&\leqslant\mathbb{E}_\Theta\left[2^{-K_{j,\kappa}(X,\Theta)}\big|E_{n,\kappa}(j,X,\Theta),N_n(X,\Theta)>0\right].\end{aligned}\tag{4.60}$$

We denote $\delta_j(X,\Theta)\in\{0,1\}^k$ the vector indicating at which depth the jth direction is chosen for splitting, that is $\delta_{j,\ell}(X,\Theta)=1$ if and only if the jth feature is used for splitting at depth ℓ. We have

$$K_{j,\kappa}(X,\Theta)=\sum_{\ell=1}^{\kappa}\delta_{j,\ell}(X,\Theta).$$

For $\ell=1,\dots,\kappa$, the random variables $\delta_{j,\ell}(X,\Theta)$ are distributed as Bernoulli random variables. Conditional on $E_{n,\kappa}(j,X,\Theta)$ and $N_n(X,\Theta)>0$, we know that for all $\ell=1,\dots,\kappa$, the jth direction was eligible for splitting at level ℓ. Therefore, the probability of selecting the jth direction at any level $1\leqslant\ell\leqslant\kappa$, is $p_\ell\geqslant 1/d$ (at worst, all variables are eligible for splitting, leading to $p_\ell=1/d$). Besides, conditional on $E_{n,\kappa}(j,X,\Theta)$ and $N_n(X,\Theta)>0$, the random variables $\delta_{j,\ell}(X,\Theta)$ are independent by construction of the centered forest. Indeed, conditional on $E_{n,\kappa}(j,X,\Theta)$ and $N_n(X,\Theta)>0$, the jth direction can be chosen up to depth κ (independence is broken only when the direction cannot be chosen at a given depth as the following one will not be chosen either). Then,

$$\mathbb{E}_\Theta\left[2^{-K_{j,\kappa}(X,\Theta)}\big|E_{n,\kappa}(j,X,\Theta),N_n(X,\Theta)>0\right]=\prod_{\ell=1}^{\kappa}\mathbb{E}_\Theta\left[2^{-\delta_{j,\ell}(X,\Theta)}\big|E_{n,\kappa}(j,X,\Theta),N_n(X,\Theta)>0\right]\tag{4.61}$$

$$=\prod_{\ell=1}^{\kappa}\left(\frac{p_\ell}{2}+(1-p_\ell)\right)\tag{4.62}$$

$$\leqslant\left(1-\frac{1}{2d}\right)^{\kappa}.\tag{4.63}$$

Therefore, injecting Equations (4.58) and (4.63) into (4.55), we get

$$\frac{\mathbb{E}_\Theta\left[\mathbb{1}_{N_n(X,\Theta)>0}\mu\left(A_n^{(j)}(X,\Theta)\right)\right]}{\mathbb{P}_\Theta\left(N_n(X,\Theta)>0\right)}\leqslant 2\mathbb{P}_\Theta\left(N_n(A_{n,\kappa}^{(j),\text{left}}(X,\Theta))=0\big|N_n(X,\Theta)>0\right)+\left(1-\frac{1}{2d}\right)^{\kappa},\tag{4.64}$$

which implies

$$\left(\frac{\mathbb{E}_\Theta\left[\mathbb{1}_{N_n(X,\Theta)>0}\mu\left(A_n^{(j)}(X,\Theta)\right)\right]}{\mathbb{P}_\Theta\left(N_n(X,\Theta)>0\right)}\right)^2$$
$$\leqslant 4\mathbb{P}_\Theta\left(N_n(A_{n,\kappa}^{(j),\text{left}}(X,\Theta))=0\middle|N_n(X,\Theta)>0\right)+2\left(1-\frac{1}{2d}\right)^{2\kappa}, \tag{4.65}$$

using $(a+b)^2 \leqslant 2a^2+2b^2 \leqslant 2a^2+2b$ if $b \leqslant 1$. Plugging-in this expression into (4.52) leads to

$$\mathbb{E}\left[\left(f_{\infty,n}^{\text{VF}}(X)-f^\star(X)\right)^2\right] \leqslant 4d\sum_{j=1}^d ||\partial f_j^\star||_\infty^2\left(1-\frac{1}{2d}\right)^{2\kappa}+2e^{-\frac{kn}{2^{k+1}}}$$
$$+8d\sum_{j=1}^d ||\partial f_j^\star||_\infty^2\mathbb{P}\left(N_n(A_{n,\kappa}^{(j),\text{left}}(X,\Theta))=0\middle|N_n(X,\Theta)>0\right). \tag{4.66}$$

Then,

$$\mathbb{P}\left(N_n(A_{n,\kappa}^{(j),\text{left}}(X,\Theta))=0\middle|N_n(X,\Theta)>0\right) \tag{4.67}$$
$$=\mathbb{E}\left[\mathbb{P}\left(N_n(A_{n,\kappa}^{(j),\text{left}}(X,\Theta))=0\middle|N_n(X,\Theta)>0,N_n(A_{n,\kappa}(X,\Theta)),X,\Theta\right)|N_n(X,\Theta)>0\right]$$
$$=\mathbb{E}\left[2^{-N_n(A_{n,\kappa}(X,\Theta))}|N_n(X,\Theta)>0\right] \tag{4.68}$$
$$\leqslant 2\mathbb{E}\left[2^{-N_n(A_{n,\kappa}(X,\Theta))}\right]. \tag{4.69}$$

The last line is obtained by making the expectation explicit and noting that $\mathbb{P}\left(N_n(X,\Theta)>0\right)^{-1} \leqslant 1/(1-e^{-1}) \leqslant 2$. Furthermore, conditional on X,Θ, $N_n(A_{n,\kappa}(X,\Theta))$ is distributed as a binomial of parameters n and $\mu\left(A_{n,\kappa}(X,\Theta)\right)=2^{-\kappa}$. Thus,

$$\mathbb{P}\left(N_n(A_{n,\kappa}^{(j),\text{left}}(X,\Theta))=0\middle|N_n(X,\Theta)>0\right) \leqslant 2\mathbb{E}\left[2^{-N_n(A_{n,\kappa}(X,\Theta))}\right] \tag{4.70}$$
$$\leqslant 2\mathbb{E}\left[\mathbb{E}\left[2^{-N_n(A_{n,\kappa}(X,\Theta))}|X,\Theta\right]\right] \tag{4.71}$$
$$\leqslant 2\left(1-\frac{\mu\left(A_{n,\kappa}(X,\Theta)\right)}{2}\right)^n \tag{4.72}$$
$$=2\left(1-2^{-\kappa-1}\right)^n \tag{4.73}$$
$$\leqslant 2\exp\left(-\frac{n}{2^{\kappa+1}}\right). \tag{4.74}$$

Overall,

$$\mathbb{E}\left[\left(f_{\infty,n}^{\text{VF}}(X)-f^\star(X)\right)^2\right]$$
$$\leqslant 4d\left(\sum_{j=1}^d ||\partial f_j^\star||_\infty^2\right)\left(\left(1-\frac{1}{2d}\right)^{2\kappa}+4\exp\left(-\frac{n}{2^{\kappa+1}}\right)\right)+2\exp\left(-\frac{kn}{2^{k+1}}\right). \tag{4.75}$$

Choosing $\kappa = \log_2(n) - \log_2(\log_2(n))$, that is $2^\kappa = n/(\log_2(n))$, we obtain

$$\exp\left(2\kappa \log\left(1 - \frac{1}{2d}\right)\right) + 4\exp\left(-\frac{n}{2^{\kappa+1}}\right) \leqslant \left(\frac{n}{\log_2 n}\right)^{2\log_2\left(1-\frac{1}{2d}\right)} + 4n^{-1/(2\ln 2)}. \tag{4.76}$$

Consequently, recalling that $k = \lfloor \log_2(n) \rfloor$,

$$\begin{aligned}
&\mathbb{E}\left[\left(f_{\infty,n}^{\mathrm{VF}}(X) - f^\star(X)\right)^2\right] \\
&\leqslant 4d\left(\sum_{j=1}^d ||\partial f_j^\star||_\infty^2\right)\left(\left(\frac{n}{\log_2 n}\right)^{2\log_2\left(1-\frac{1}{2d}\right)} + 4n^{-1/(2\ln 2)}\right) + 2n^{-1/(2\ln 2)} \qquad (4.77)\\
&\leqslant C_d\left(\frac{n}{\log_2 n}\right)^{2\log_2\left(1-\frac{1}{2d}\right)} + (C_d+2)\, n^{-1/(2\ln 2)}, \qquad (4.78)
\end{aligned}$$

where $C_d = 4d\left(\sum_{j=1}^d ||\partial f_j^\star||_\infty^2\right)$.

S2.3 Proofs of Section 4.4 (Theorem 4.4.1)

In this section, we prove the consistency of the infinite KeRF estimator in the mean interpolating regime (Theorem 4.4.1). We follow the proof given in Scornet 2016b and first present two of its results.

Lemma S2 (Scornet 2016b). *Let $k \in \mathbb{N}$ and consider an infinite centered random forest of depth k. Then, for all $x, z \in [0,1]^d$,*

$$K_k(x,z) = \sum_{\substack{k_1,\dots,k_d \\ \sum_{\ell=1}^d k_\ell = k}} \frac{k!}{k_1!\dots k_d!}\left(\frac{1}{d}\right)^k \prod_{j=1}^d \mathbb{1}_{\lceil 2^{k_j} x^{(j)} \rceil = \lceil 2^{k_j} z^{(j)} \rceil}.$$

Theorem S3 (Scornet 2016b). *Let $f^\star$ be a L-Lipschitz function. Then, for all k,*

$$\sup_{x\in[0,1]^d}\left|\frac{\int_{[0,1]^d} k_k(x,z) f^\star(z)\mathrm{d}z_1\dots\mathrm{d}z_d}{\int_{[0,1]^d} k_k(x,z)\mathrm{d}z_1\dots\mathrm{d}z_d} - f^\star(x)\right| \leqslant Ld\left(1-\frac{1}{2d}\right)^k.$$

Proof of Theorem 4.4.1. Let $x \in [0,1]^d$ and recall that

$$f_{\infty,n}^{\mathrm{KeRF}}(x) = \frac{\sum_{i=1}^n Y_i K_k(x, X_i)}{\sum_{i=1}^n K_k(x, X_i)}.$$

Thus, letting

$$A_n(x) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{Y_i K_k(x,X_i)}{\mathbb{E}\left[K_k(x,X)\right]} - \frac{\mathbb{E}\left[Y K_k(x,X)\right]}{\mathbb{E}\left[K_k(x,X)\right]}\right),$$
$$B_n(x) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{K_k(x,X_i)}{\mathbb{E}\left[K_k(x,X)\right]} - 1\right),$$
$$\text{and } M_n(x) = \frac{\mathbb{E}\left[Y K_k(x,X)\right]}{\mathbb{E}\left[K_k(x,X)\right]},$$

the estimate $f_{\infty,n}^{\mathrm{KeRF}}(x)$ can be rewritten as

$$f_{\infty,n}^{\mathrm{KeRF}}(x) = \frac{M_n(x) + A_n(x)}{1 + B_n(x)},$$

which leads to

$$f_{\infty,n}^{\mathrm{KeRF}}(x) - f^\star(x) = \frac{M_n(x) - f^\star(x) + A_n(x) - B_n(x) f^\star(x)}{1 + B_n(x)}.$$

According to Theorem S3, we have

$$\begin{aligned}
|M_n(x) - f^\star(x)| &= \left|\frac{\mathbb{E}\left[f^\star(X)K_k(x,X)\right]}{\mathbb{E}\left[K_k(x,X)\right]} + \frac{\mathbb{E}\left[\varepsilon K_k(x,X)\right]}{\mathbb{E}\left[K_k(x,X)\right]} - f^\star(x)\right| \\
&\leqslant \left|\frac{\mathbb{E}\left[f^\star(X)K_k(x,X)\right]}{\mathbb{E}\left[K_k(x,X)\right]} - f^\star(x)\right| \\
&\leqslant C\left(1 - \frac{1}{2d}\right)^k,
\end{aligned}$$

where $C = Ld$. Take $\alpha \in]0, 1/2]$. Let $\mathcal{C}_\alpha(x)$ be the event $\big\{|A_n(x)| \leqslant \alpha\big\} \cap \big\{|B_n(x)| \leqslant \alpha\big\}$. On the event $\mathcal{C}_\alpha(x)$, we have

$$\begin{aligned}
|f_{\infty,n}^{\mathrm{KeRF}}(x) - f^\star(x)|^2 &\leqslant 8|M_n(x) - f^\star(x)|^2 + 8|A_n(x) - B_n(x) f^\star(x)|^2 \\
&\leqslant 8C^2\left(1 - \frac{1}{2d}\right)^{2k} + 8\alpha^2(1 + \|f^\star\|_\infty)^2.
\end{aligned}$$

Thus,

$$\mathbb{E}[|f_{\infty,n}^{\mathrm{KeRF}}(x) - f^\star(x)|^2 \mathbb{1}_{\mathcal{C}_\alpha(x)}] \leqslant 8C^2\left(1 - \frac{1}{2d}\right)^{2k} + 8\alpha^2(1 + \|f^\star\|_\infty)^2. \tag{4.79}$$

Consequently, to find an upper bound on the rate of consistency of $f_{\infty,n}^{\mathrm{KeRF}}$, we just need to

upper bound

$$\begin{aligned}
\mathbb{E}\Big[|f_{\infty,n}^{\mathrm{KeRF}}(x) - f^\star(x)|^2 \mathbb{1}_{\mathcal{C}_\alpha^c(x)}\Big] &\leqslant \mathbb{E}\Big[\Big|\max_{1\leqslant i\leqslant n}|Y_i| + |f^\star(x)|\Big|^2 \mathbb{1}_{\mathcal{C}_\alpha^c(x)}\Big] \\
&\quad \text{(since } f_{\infty,n}^{\mathrm{KeRF}} \text{ is a local averaging estimate)} \\
&\leqslant \mathbb{E}\Big[\Big|2\|f^\star\|_\infty + \max_{1\leqslant i\leqslant n}|\varepsilon_i|\Big|^2 \mathbb{1}_{\mathcal{C}_\alpha^c(x)}\Big] \\
&\leqslant \left(\mathbb{E}\left[2\|f^\star\|_\infty + \max_{1\leqslant i\leqslant n}|\varepsilon_i|\right]^4 \mathbb{P}\left[\mathcal{C}_\alpha^c(x)\right]\right)^{1/2} \\
&\quad \text{(by Cauchy-Schwarz inequality)} \\
&\leqslant \left(\left(16\|f^\star\|_\infty^4 + 8\mathbb{E}\Big[\max_{1\leqslant i\leqslant n}|\varepsilon_i|\Big]^4\right)\mathbb{P}\left[\mathcal{C}_\alpha^c(x)\right]\right)^{1/2}.
\end{aligned}$$

According to Lemma S5, there exists a constant $C' > 0$ such that, for all n,

$$\mathbb{E}\Big[\max_{1\leqslant i\leqslant n}\varepsilon_i^4\Big] \leqslant C'\sigma^4(\log n)^2. \tag{4.80}$$

Thus, there exists C'' such that, for all $n > 1$,

$$\mathbb{E}\Big[|f_{\infty,n}^{\mathrm{KeRF}}(x) - f^\star(x)|^2 \mathbb{1}_{\mathcal{C}_\alpha^c(x)}\Big] \leqslant C''\sigma^2(\log n)(\mathbb{P}\left[\mathcal{C}_\alpha^c(x)\right])^{1/2}. \tag{4.81}$$

The last probability $\mathbb{P}\left[\mathcal{C}_\alpha^c(x)\right]$ can be upper bounded by using Chebyshev's inequality. Indeed, with respect to $A_n(x)$,

$$\begin{aligned}
\mathbb{P}\big[|A_n(x)| > \alpha\big] &\leqslant \frac{1}{n\alpha^2}\mathbb{E}\left[\frac{YK_k(x,X)}{\mathbb{E}\left[K_k(x,X)\right]} - \frac{\mathbb{E}\left[YK_k(x,X)\right]}{\mathbb{E}\left[K_k(x,X)\right]}\right]^2 \\
&\leqslant \frac{1}{n\alpha^2}\frac{1}{(\mathbb{E}\left[K_k(x,X)\right])^2}\mathbb{E}\Big[Y^2K_k(x,X)^2\Big] \\
&\leqslant \frac{2}{n\alpha^2}\frac{1}{(\mathbb{E}\left[K_k(x,X)\right])^2}\Big(\mathbb{E}\Big[f^\star(X)^2K_k(x,X)^2\Big] \\
&\qquad + \mathbb{E}\Big[\varepsilon^2K_k(x,X)^2\Big]\Big) \\
&\leqslant \frac{2(\|f^\star\|_\infty^2 + \sigma^2)}{n\alpha^2}\frac{\mathbb{E}\left[K_k(x,X)^2\right]}{(\mathbb{E}\left[K_k(x,X)\right])^2} \qquad (4.82) \\
&= \frac{C_0}{n\alpha^2}\frac{\mathbb{E}\left[K_k(x,X)^2\right]}{(\mathbb{E}\left[K_k(x,X)\right])^2} \qquad (4.83)
\end{aligned}$$

with $C_0 = 2(\|f^\star\|_\infty^2 + \sigma^2)$ a constant. Meanwhile with respect to $B_n(x)$, we obtain, still by Chebyshev's inequality,

$$\mathbb{P}\big[|B_n(x)| > \alpha\big] \leqslant \frac{1}{n\alpha^2}\frac{\mathbb{E}\left[K_k(x,X)^2\right]}{(\mathbb{E}\left[K_k(x,X)\right])^2} \tag{4.84}$$

which matches the control made by Scornet 2016b. Consequently,

$$\mathbb{P}\left[\mathcal{C}_\alpha^c(x)\right] \leqslant \mathbb{P}\big[|A_n(x)| > \alpha\big] + \mathbb{P}\big[|B_n(x)| > \alpha\big] \tag{4.85}$$

$$\leqslant \frac{C_0+1}{n\alpha^2}\frac{\mathbb{E}\left[K_k(x,X)^2\right]}{(\mathbb{E}\left[K_k(x,X)\right])^2}. \tag{4.86}$$

Besides, for all $x \in [0,1]^d$, for all k, $\mathbb{E}\left[K_k^{cc}(x,X)\right] = \frac{1}{2^k}$ (see in Scornet 2016b the proof of theorem VI.1 p.11). Since $K_k(x,X) \leqslant 1$, we know that

$$\mathbb{E}\left[K_k^{cc}(x,X)\right] = \frac{1}{2^k} \;\geqslant\; \mathbb{E}\left[K_k^{cc}(x,X)^2\right] \;\geqslant\; (\mathbb{E}\left[K_k^{cc}(x,X)\right])^2 = \frac{1}{2^{2k}}, \tag{4.87}$$

which leads to

$$\mathbb{P}\left[\mathcal{C}_\alpha^c(x)\right] \leqslant 2^{2k}\left(\frac{C_0+1}{n\alpha^2}\right)\mathbb{E}\left[K_k(x,X)^2\right], \tag{4.88}$$

but to pursue, we need a tighter upper bound on $\mathbb{E}\left[K_k^{cc}(x,X)^2\right]$ than that obtained from (4.87). Such a control is provided in Lemma S4 below, which is original, and departs from the work of Scornet 2016b.

Lemma S4. *For all $d \geqslant 2$, for all k large enough, for all $x \in [0,1]^d$,*

$$\mathbb{E}\left[K_k^{cc}(x,X)^2\right] \leqslant 2^{-k}k^{-\frac{d-1}{2}}\left(C_1 + C_2\left(\log_2(k)\right)^d\right), \tag{4.89}$$

where

$$C_1 = 1 + \frac{2d^{d/2}}{(4\pi)^{(d-1)/2}} \quad \textit{and} \quad C_2 = 5^d\left(\frac{d-1}{2}\right)^d. \tag{4.90}$$

Proof of Lemma S4. From Lemma S2, we know that

$$\mathbb{E}\left[K_k^{cc}(x,X)^2\right] = \mathbb{E}\left[\left(\sum_{\substack{k_1,\ldots,k_d \\ \sum_{j=1}^d k_j = k}} \frac{k!}{k_1!\ldots k_d!}\left(\frac{1}{d}\right)^k \prod_{j=1}^d \mathbb{1}_{\lceil 2^{k_j}x^{(j)}\rceil = \lceil 2^{k_j}X^{(j)}\rceil}\right)^2\right]. \tag{4.91}$$

Developing the square within the expectation, we obtain two terms, the first one A being the sum of squares and the second one, B, being the cross-product terms. The first term A takes the form

$$A := \mathbb{E}\left[\sum_{\substack{k_1,\ldots,k_d \\ \sum_{j=1}^d k_j = k}} \left(\frac{k!}{k_1!\ldots k_d!}\right)^2\left(\frac{1}{d}\right)^{2k}\prod_{j=1}^d \mathbb{1}_{\lceil 2^{k_j}x^{(j)}\rceil = \lceil 2^{k_j}x^{(j)}\rceil}\right] \tag{4.92}$$

$$= \sum_{\substack{k_1,\ldots,k_d \\ \sum_{j=1}^d k_j = k}} \left(\frac{k!}{k_1!\ldots k_d!}\right)^2\left(\frac{1}{d}\right)^{2k}\prod_{j=1}^d \mathbb{P}\left(\lceil 2^{k_j}x^{(j)}\rceil = \lceil 2^{k_j}X^{(j)}\rceil\right). \tag{4.93}$$

Note that, for all j, $\mathbb{P}\left(\lceil 2^{k_j} x^{(j)} \rceil = \lceil 2^{k_j} X^{(j)} \rceil\right) = 2^{-k_j}$, and $\prod_{j=1}^{d} 2^{-k_j} = 2^{-k}$. Therefore,

$$A = \sum_{\substack{k_1,\dots,k_d \\ \sum_{j=1}^d k_j = k}} \left(\frac{k!}{k_1!\dots k_d!}\right)^2 \left(\frac{1}{d}\right)^{2k} 2^{-k}. \tag{4.94}$$

Thanks to Richmond et al. 2009, we know that, for all $d \geqslant 2$,

$$\sum_{\substack{k_1,\dots,k_d \\ \sum_{j=1}^d k_j = k}} \left(\frac{k!}{k_1!\dots k_d!}\right)^2 \underset{k\to+\infty}{\sim} \frac{d^{2k+d/2}}{(4\pi k)^{(d-1)/2}}. \tag{4.95}$$

Therefore, for all k large enough, we have

$$\sum_{\substack{k_1,\dots,k_d \\ \sum_{j=1}^d k_j = k}} \left(\frac{k!}{k_1!\dots k_d!}\right)^2 \leqslant \frac{2d^{2k+d/2}}{(4\pi k)^{(d-1)/2}}. \tag{4.96}$$

Thus, letting $C_1 = 2d^{d/2}/(4\pi)^{(d-1)/2}$, for all k large enough,

$$A \leqslant C_1 2^{-k} k^{-(d-1)/2}. \tag{4.97}$$

Regarding the second term B,

$$B := \mathbb{E}\left[\sum_{\substack{(k_1,\dots,k_d) \\ \neq (\ell_1,\dots,\ell_d), \\ \sum_{j=1}^d k_j = \sum_{j=1}^d \ell_j = k}} \frac{k!}{k_1!\dots k_d!}\frac{k!}{\ell_1!\dots \ell_d!}\left(\frac{1}{d}\right)^{2k} \prod_{j=1}^{d} \mathbb{1}_{\lceil 2^{k_j} x^{(j)} \rceil = \lceil 2^{k_j} X^{(j)} \rceil} \mathbb{1}_{\lceil 2^{\ell_j} x^{(j)} \rceil = \lceil 2^{\ell_j} X^{(j)} \rceil}\right] \tag{4.98}$$

$$= \sum_{\substack{(k_1,\dots,k_d) \\ \neq (\ell_1,\dots,\ell_d), \\ \sum_{j=1}^d k_j = \sum_{j=1}^d \ell_j = k}} \frac{k!}{k_1!\dots k_d!}\frac{k!}{\ell_1!\dots \ell_d!}\left(\frac{1}{d}\right)^{2k} \mathbb{P}\left(\bigcap_{j=1}^{d} \left(\left(\lceil 2^{k_j} x^{(j)} \rceil = \lceil 2^{k_j} X^{(j)} \rceil\right) \cap \left(\lceil 2^{\ell_j} x^{(j)} \rceil = \lceil 2^{\ell_j} X^{(j)} \rceil\right)\right)\right).$$

A small computation yields

$$\mathbb{P}\left(\bigcap_{j=1}^{d}\left((\lceil 2^{k_j}x^{(j)}\rceil = \lceil 2^{k_j}X^{(j)}\rceil)\cap(\lceil 2^{\ell_j}x^{(j)}\rceil = \lceil 2^{\ell_j}X^{(j)}\rceil)\right)\right)$$

$$= \mathbb{P}\left(\bigcap_{j=1}^{d}\lceil 2^{\ell_j}x^{(j)}\rceil = \lceil 2^{\ell_j}X^{(j)}\rceil \,\middle|\, \forall j, \lceil 2^{k_j}x^{(j)}\rceil = \lceil 2^{k_j}X^{(j)}\rceil\right)2^{-k} \tag{4.99}$$

$$= 2^{-k}\prod_{j=1}^{d}\mathbb{P}\left(\lceil 2^{\ell_j}x^{(j)}\rceil = \lceil 2^{\ell_j}X^{(j)}\rceil \,\middle|\, \lceil 2^{k_j}x^{(j)}\rceil = \lceil 2^{k_j}X^{(j)}\rceil\right) \tag{4.100}$$

$$= 2^{-k}2^{-\sum_{j=1}^{d}(\ell_j-k_j)\mathbb{1}_{\ell_j\geqslant k_j}} \tag{4.101}$$

$$= 2^{-\sum_{j=1}^{d}k_j(\mathbb{1}_{\ell_j\geqslant k_j}+\mathbb{1}_{\ell_j<k_j})-\sum_{j=1}^{d}(\ell_j-k_j)\mathbb{1}_{\ell_j\geqslant k_j}} \tag{4.102}$$

$$= 2^{-\sum_{j=1}^{d}k_j\mathbb{1}_{\ell_j<k_j}-\sum_{j=1}^{d}\ell_j\mathbb{1}_{\ell_j\geqslant k_j}} \tag{4.103}$$

$$= 2^{-\sum_{j=1}^{d}\max(k_j,\ell_j)}. \tag{4.104}$$

Therefore,

$$B = \left(\frac{1}{d}\right)^{2k}\sum_{\substack{(k_1,\dots,k_d)\\ \neq(\ell_1,\dots,\ell_d),\\ \sum_{j=1}^{d}k_j=\sum_{j=1}^{d}\ell_j=k}}\frac{k!}{k_1!\dots k_d!}\frac{k!}{\ell_1!\dots\ell_d!}\left(\frac{1}{2}\right)^{\sum_{j=1}^{d}\max(k_j,\ell_j)}. \tag{4.105}$$

$$= \left(\frac{1}{d}\right)^{2k}\sum_{\substack{(k_1,\dots,k_d)\\ \neq(\ell_1,\dots,\ell_d),\\ \sum_{j=1}^{d}k_j=\sum_{j=1}^{d}\ell_j=k}}\frac{k!}{k_1!\dots k_d!}\frac{k!}{\ell_1!\dots\ell_d!}\left(\frac{1}{2}\right)^{k+\frac{1}{2}\sum_{j=1}^{d}|k_j-\ell_j|} \tag{4.106}$$

$$= \left(\frac{1}{2d^2}\right)^{k}\sum_{\substack{(k_1,\dots,k_d)\\ \neq(\ell_1,\dots,\ell_d),\\ \sum_{j=1}^{d}k_j=\sum_{j=1}^{d}\ell_j=k}}\frac{k!}{k_1!\dots k_d!}\frac{k!}{\ell_1!\dots\ell_d!}\left(\frac{1}{2}\right)^{\frac{1}{2}\sum_{j=1}^{d}|k_j-\ell_j|}. \tag{4.107}$$

For all $q>0$, define the set $\mathcal{K}_q = \{\boldsymbol{\ell}=(\ell_1,\dots,\ell_d), \boldsymbol{k}=(k_1,\dots,k_d) | \sum_{j=1}^{d}|k_j-\ell_j| \geqslant 2q\}$, so that

$$B = \left(\frac{1}{2d^2}\right)^{k}\sum_{\substack{(\boldsymbol{k},\boldsymbol{\ell})\in\mathcal{K}_q\\ \boldsymbol{\ell}\neq\boldsymbol{k}\\ \sum_{j=1}^{d}k_j=\sum_{j=1}^{d}\ell_j=k}}\frac{k!}{k_1!\dots k_d!}\frac{k!}{\ell_1!\dots\ell_d!}\left(\frac{1}{2}\right)^{\frac{1}{2}\sum_{j=1}^{d}|k_j-\ell_j|}$$

$$+\left(\frac{1}{2d^2}\right)^{k}\sum_{\substack{(\boldsymbol{k},\boldsymbol{\ell})\notin\mathcal{K}_q\\ \boldsymbol{\ell}\neq\boldsymbol{k}\\ \sum_{j=1}^{d}k_j=\sum_{j=1}^{d}\ell_j=k}}\frac{k!}{k_1!\dots k_d!}\frac{k!}{\ell_1!\dots\ell_d!}\left(\frac{1}{2}\right)^{\frac{1}{2}\sum_{j=1}^{d}|k_j-\ell_j|}$$

$$= B_1 + B_2. \tag{4.108}$$

Regarding B_1, we have

$$B_1 \leqslant \left(\frac{1}{2d^2}\right)^k \sum_{\substack{(\boldsymbol{k},\boldsymbol{\ell})\in\mathcal{K}_q \\ \boldsymbol{\ell}\neq\boldsymbol{k} \\ \sum_{j=1}^d k_j=\sum_{j=1}^d \ell_j=k}} \frac{k!}{k_1!\dots k_d!}\frac{k!}{\ell_1!\dots\ell_d!}2^{-q} \tag{4.109}$$

$$\leqslant \left(\frac{1}{2d^2}\right)^k 2^{-q} \left(\sum_{\boldsymbol{k},\sum_{j=1}^d k_j=k} \frac{k!}{k_1!\dots k_d!}\right)\left(\sum_{\boldsymbol{\ell},\sum_{j=1}^d \ell_j=k} \frac{k!}{\ell_1!\dots\ell_d!}\right) \tag{4.110}$$

$$\leqslant 2^{-k-q}, \tag{4.111}$$

as

$$\sum_{\boldsymbol{k},\sum_{j=1}^d k_j=k} \frac{k!}{k_1!\dots k_d!} = d^k. \tag{4.112}$$

We now define, for all $\boldsymbol{k}$, $\mathcal{K}_q(\boldsymbol{k}) := \{\boldsymbol{\ell}=(\ell_1,\dots,\ell_d), \sum_{j=1}^d \ell_j = k, \sum_{j=1}^d |k_j-\ell_j| \geqslant 2q\}$. Regarding B_2, we have

$$B_2 \leqslant \left(\frac{1}{2d^2}\right)^k \sum_{\substack{\boldsymbol{k},\boldsymbol{\ell}\notin\mathcal{K}_q \\ \boldsymbol{\ell}\neq\boldsymbol{k} \\ \sum_{j=1}^d k_j=\sum_{j=1}^d \ell_j=k}} \frac{k!}{k_1!\dots k_d!}\frac{k!}{\ell_1!\dots\ell_d!} \tag{4.113}$$

$$= \left(\frac{1}{2d^2}\right)^k \sum_{\boldsymbol{k},\sum_{j=1}^d k_j=k} \frac{k!}{k_1!\dots k_d!} \sum_{\substack{\boldsymbol{\ell}\notin\mathcal{K}_q(\boldsymbol{k}) \\ \boldsymbol{\ell}\neq\boldsymbol{k} \\ \sum_{j=1}^d k_j=\sum_{j=1}^d \ell_j=k}} \frac{k!}{\ell_1!\dots\ell_d!}. \tag{4.114}$$

$$\tag{4.115}$$

Note that for all $\boldsymbol{\ell}$, $\frac{k!}{\ell_1!\dots\ell_d!}$ is maximal when $\max_i \ell_i$ is minimal. Therefore, for all $k \geqslant 2d$,

$$\frac{k!}{\ell_1!\dots\ell_d!} = \frac{k!}{\Gamma(\ell_1+1)\dots\Gamma(\ell_d+1)} \tag{4.116}$$

$$\leqslant \frac{k!}{\Gamma(\lfloor k/d\rfloor+1)\dots\Gamma(\lfloor k/d\rfloor+1)} \tag{4.117}$$

$$\leqslant \frac{k!}{\Gamma(k/d)^d}. \tag{4.118}$$

Using an inequality from Batir 2008, we obtain

$$\frac{k!}{\Gamma(k/d)^d} \leqslant \frac{k^{k+1/2}e^{-k}}{k^k d^{-k} e^{-k} k^{d/2}}$$
$$\leqslant d^k k^{-(d-1)/2}.$$

Overall, for all $k \geqslant 2d$,

$$B_2 \leqslant \left(\frac{1}{2d}\right)^k \sum_{\boldsymbol{k}, \sum_{j=1}^d k_j = k} \frac{k!}{k_1! \dots k_d!} \sum_{\substack{\boldsymbol{\ell} \notin \mathcal{K}_q(\boldsymbol{k}) \\ \boldsymbol{\ell} \neq \boldsymbol{k} \\ \sum_{j=1}^d k_j = \sum_{j=1}^d \ell_j = k}} k^{-(d-1)/2} \tag{4.119}$$

$$\leqslant k^{-(d-1)/2} \left(\frac{1}{2d}\right)^k \sum_{\boldsymbol{k}, \sum_{j=1}^d k_j = k} \frac{k!}{k_1! \dots k_d!} \mathrm{Card}(\mathcal{K}_q(\boldsymbol{k})). \tag{4.120}$$

We now want to upper bound the cardinal of $\mathcal{K}_q(\boldsymbol{k})$. Denoting by $B_{L_1}(0, 2q)$ the ball of radius $2q$ with respect to the L_1 norm, note that

$$\mathrm{Card}(\mathcal{K}_q(\boldsymbol{k})) \leqslant \mathrm{Card}(\{x \in \mathbb{N}^d \cap B_{L_1}(\boldsymbol{k}, 2q)\}) \tag{4.121}$$

$$\leqslant \mathrm{Card}(\{x \in \mathbb{N}^d \cap B_{L_1}(0, 2q)\}). \tag{4.122}$$

Since,

$$B_{L_1}(0, c) \subset B_{L_\infty}(0, c) \subset B_{L_\infty}(0, \lceil c \rceil),$$

we have,

$$\begin{aligned} \mathrm{Card}(\mathcal{K}_q(\boldsymbol{k})) &\leqslant \mathrm{Card}(\{x \in \mathbb{N}^d \cap B_{L_\infty}(0, \lceil 2q \rceil)\}) \\ &\leqslant (2\lceil 2q \rceil + 1)^d \\ &\leqslant (4q + 3)^d . \end{aligned}$$

Thus, we have, for all $k \geqslant 2d$,

$$B_2 \leqslant k^{-(d-1)/2} \left(\frac{1}{2d}\right)^k (4q+3)^d \sum_{\boldsymbol{k}, \sum_{j=1}^d k_j = k} \frac{k!}{k_1! \dots k_d!} \tag{4.123}$$

$$\leqslant k^{-(d-1)/2} (4q+3)^d 2^{-k}, \tag{4.124}$$

as

$$\sum_{\boldsymbol{k}, \sum_{j=1}^d k_j = k} \frac{k!}{k_1! \dots k_d!} = d^k. \tag{4.125}$$

Finally, for all q, we have

$$B = B_1 + B_2 \tag{4.126}$$

$$\leqslant 2^{-k-q} + k^{-(d-1)/2} (4q+3)^d 2^{-k}. \tag{4.127}$$

Let $q = \left(\frac{d-1}{2}\right) \log_2(k)$. For all $q \geqslant 3$, that is for all $k \geqslant 2^{6/(d-1)}$, and for all $k \geqslant 2d$,

$$B \leqslant 2^{-k} \left(k^{-\frac{d-1}{2}} + k^{-(d-1)/2} C_2 \left(\log_2(k)\right)^d\right), \tag{4.128}$$

where

$$C_2 = 5^d \left(\frac{d-1}{2} \right)^d . \tag{4.129}$$

Finally, for all k large enough

$$\begin{aligned} \mathbb{E}\left[K_k^{cc}(x,X)^2\right] &\leqslant A + B_1 + B_2 && (4.130)\\ &\leqslant 2^{-k} k^{-\frac{d-1}{2}} \left(C_1 + 1 + C_2 \left(\log_2(k)\right)^d \right). && (4.131) \end{aligned}$$

□

According to inequality (4.88) and Lemma S4, we have, for all k large enough

$$\mathbb{P}\left[\mathcal{C}_\alpha^c(x)\right] \leqslant \frac{C_0+1}{n\alpha^2} 2^k k^{-\frac{d-1}{2}} \left(C_1 + C_2 \left(\log_2(k)\right)^d \right). \tag{4.132}$$

Consequently, according to inequality (4.81), we obtain, for all k large enough

$$\begin{aligned} &\mathbb{E}\Big[|f_{\infty,n}^{\mathrm{KeRF}}(x) - f^\star(x)|^2 \mathbb{1}_{\mathcal{C}_\alpha^c(x)}\Big] \\ &\leqslant C'' \sigma^2 \log n \left(\frac{C_0+1}{n\alpha^2} 2^k k^{-\frac{d-1}{2}} \left(C_1 + C_2 \left(\log_2(k)\right)^d \right) \right)^{1/2} \\ &\leqslant C'' \sigma^2 (C_0+1)^{1/2} (\max(C_1,C_2))^{1/2} \frac{\log n}{n^{1/2}\alpha} 2^{k/2} k^{-\frac{d-1}{4}} \left(\left(1 + \left(\log_2(k)\right)^d \right) \right)^{1/2} \\ &\leqslant C_3 \frac{\log n}{n^{1/2}\alpha} 2^{k/2} k^{-\frac{d-1}{4}} \left(\log_2(k)\right)^{d/2}, \end{aligned}$$

where $C_3 = C'' \sigma^2 (C_0+1)^{1/2} (2\max(C_1,C_2))^{1/2}$. Then using inequality (4.79), for all k large enough

$$\begin{aligned} &\mathbb{E}\Big[f_{\infty,n}^{\mathrm{KeRF}}(x) - f^\star(x)\Big]^2 \\ &\leqslant \mathbb{E}\Big[|f_{\infty,n}^{\mathrm{KeRF}}(x) - f^\star(x)|^2 \mathbb{1}_{\mathcal{C}_\alpha(x)}\Big] + \mathbb{E}\Big[|f_{\infty,n}^{\mathrm{KeRF}}(x) - f^\star(x)|^2 \mathbb{1}_{\mathcal{C}_\alpha^c(x)}\Big] \\ &\leqslant 8L^2 d^2 \left(1 - \frac{1}{2d}\right)^{2k} + 8\alpha^2 (1 + \|f^\star\|_\infty)^2 \\ &\quad + C_3 \sigma^2 (\log n) \frac{2^{k/2}}{\alpha n^{1/2}} k^{-\frac{d-1}{4}} (\log_2 k)^{d/2}. \end{aligned}$$

Optimizing the right hand side in α, that is choosing

$$\alpha^3 = (\log n) \frac{2^{k/2}}{n^{1/2}} k^{-\frac{d-1}{4}} (\log_2 k)^{d/2} \frac{C_3}{8(1+\|f^\star\|_\infty)^2}, \tag{4.133}$$

we get

$$\mathbb{E}\Big[f_{\infty,n}^{\mathrm{KeRF}}(x) - f^\star(x)\Big]^2 \leqslant 8L^2 d^2 \left(1 - \frac{1}{2d}\right)^{2k} + 4C_3^{2/3} (1+\|f^\star\|_\infty)^{2/3} (\log n)^{2/3} \frac{2^{k/3}}{n^{1/3}} k^{-\frac{d-1}{6}} (\log_2 k)^{d/3}.$$

Choosing $k_n = \log_2(n)$, we obtain, for all n large enough,

$$\mathbb{E}\Big[f_{\infty,n}^{\mathrm{KeRF}}(x) - f^\star(x)\Big]^2 \leqslant 8L^2d^2n^{2\log_2(1-\frac{1}{d})} + 4C_3^{2/3}(1+\|f^\star\|_\infty)^{2/3}(\log n)^{2/3}(\log_2 n)^{-\frac{d-1}{6}}(\log_2(\log_2 n))^{d/3}. \quad (4.134)$$

Finally,

$$\mathbb{E}\Big[f_{\infty,n}^{\mathrm{KeRF}}(x) - f^\star(x)\Big]^2 \leqslant 8L^2d^2n^{2\log_2(1-\frac{1}{d})} + C_4(\log_2 n)^{-\frac{d-5}{6}}(\log_2(\log_2 n))^{d/3}. \quad (4.135)$$

with

$$C_4 = 18\times 2^{2/3}\times(\log 2)^{2/3}C''^{2/3}(\|f^\star\|_\infty^2+\sigma^2+1)(\max(C_1,C_2))^{1/3}. \quad (4.136)$$

□

Lemma S5. *Consider n i.i.d. random variables $\varepsilon_1,\dots,\varepsilon_n$, distributed as $\mathcal{N}(0,1)$. Then, for all $n \geqslant 21$,*

$$\mathbb{E}\Big[\max_{1\leqslant i\leqslant n}\varepsilon_i^4\Big] \leqslant 32e(\log n)^2.$$

Proof. We have, for all $p \geqslant 1$,

$$\mathbb{E}\left[\max_{1\leqslant i\leqslant n}|\varepsilon_i|^4\right] \leq \left(\mathbb{E}\left[\max_{1\leqslant i\leqslant n}|\varepsilon_i|^{4p}\right]\right)^{1/p} \leqslant \left(\mathbb{E}\left[\sum_{i=1}^n|\varepsilon_i|^{4p}\right]\right)^{1/p}, \quad (4.137)$$

using Jensen's inequality (by concavity of $x\mapsto x^{1/p}$ for $p\geqslant 1$). The p-th moment of a Gaussian variable $\mathcal{N}(0,1)$ can be computed as follows

$$\mathbb{E}\left[|\varepsilon_1|^p\right] = \int_0^\infty \mathbb{P}\left[|\varepsilon|^p \geqslant u\right]\mathrm{d}u \quad (4.138)$$

$$= \int_0^\infty \mathbb{P}\left[|\varepsilon| \geqslant t\right]pt^{p-1}\mathrm{d}t \quad (4.139)$$

$$\leqslant \int_0^\infty 2\exp(-t^2/2)pt^{p-1}\mathrm{d}t, \quad (4.140)$$

using classical tail inequalities for Gaussian variables. Now, setting $s = t^2/2$ and recalling that $\Gamma(z) = \int_0^\infty \exp(-t)t^{z-1}\mathrm{d}t$, we have

$$\int_0^\infty 2\exp(-t^2/2)pt^{p-1}\mathrm{d}t = 2p\int_0^\infty \exp(-s)(2s)^{\frac{p-2}{2}}\mathrm{d}s \quad (4.141)$$

$$= 2p2^{\frac{p-2}{2}}\Gamma(p/2). \quad (4.142)$$

According to Theorem 2.2 in Batir 2008, we have, for all $x>0$

$$\Gamma(x+1) < \sqrt{2\pi}x^x\exp(-x)\left(x^2+\frac{x}{3}+\frac{1}{18}\right)^{1/4}. \quad (4.143)$$

Let

$$f : x \mapsto \exp(-x)\left(x^2 + \frac{x}{3} + \frac{1}{18}\right), \tag{4.144}$$

one can show that f is non-increasing on $[1/2, \infty)$. Thus, for all $x \geqslant 1/2$,

$$\begin{aligned}
\Gamma(x+1) &< \sqrt{2\pi} x^x f(1/2)^{1/4} && (4.145)\\
&< \sqrt{2\pi} x^x \exp(-1/2)\left(\frac{1}{2}\right)^{1/4} && (4.146)\\
&< 2x^x. && (4.147)
\end{aligned}$$

Hence, for all $p \geqslant 3$,

$$\mathbb{E}\left[|\varepsilon_1|^p\right] \leqslant 4p 2^{\frac{p-2}{2}} (p/2)^{p/2}, \tag{4.148}$$

which leads to

$$\begin{aligned}
\mathbb{E}\left[\max_{1 \leqslant i \leqslant n} |\varepsilon_i|^4\right] &\leqslant \left(\mathbb{E}\left[\sum_{i=1}^n |\varepsilon_i|^{4p}\right]\right)^{1/p} && (4.149)\\
&\leqslant n^{1/p}\left(16p 2^{\frac{4p-2}{2}} (2p)^{2p}\right)^{1/p} && (4.150)\\
&\leqslant 16 n^{1/p} p^2 \left(\frac{p}{2}\right)^{1/p} && (4.151)\\
&\leqslant 32 n^{1/p} p^2. && (4.152)
\end{aligned}$$

Choosing $p = \log n$ yields, for all $n \geqslant e^3$,

$$\mathbb{E}\left[\max_{1 \leqslant i \leqslant n} |\varepsilon_i|^4\right] \leqslant 32e(\log n)^2. \tag{4.153}$$

□

S2.4 Proofs of Section 4.5 (Semi-Adaptive Forests)

Lemma S6. *For all $\alpha \in [0, 1)$, the depth k_n^{AdaCT} of a semi-adaptive centered tree verifies*

$$\lim_{n \to \infty} \mathbb{P}\left(k_n^{\mathrm{AdaCT}}(X, \Theta) \in [\log_2(n) \pm \log_2^{1-\alpha}(n)]\right) = 1.$$

Lemma S6 states that the asymptotic behavior of $k_n^{\mathrm{AdaCT}}(X, \Theta)$ is equivalent to $\log_2 n$ up to a negligible factor. The $\log(n)$ equivalent matches the condition for the mean interpolation regime in the case of CRF exhibited in Section 4.3.

Proof of Lemma S6

For all $0 \leqslant j \leqslant k$, we let $A_{j,n}(X,\Theta)$ be the cell containing X in the tree truncated at level j. Similarly, we let $N_{j,n}(X,\Theta)$ the number of observations in this cell. Then,

$$\begin{aligned}\mathbb{P}\left(k_n(X,\Theta) \geqslant k\right) &= \mathbb{P}\left(N_{k-1,n}(X,\Theta) \geqslant 2\right) && (4.154)\\ &= \mathbb{E}\left[\mathbb{P}\left(N_{k-1,n}(X,\Theta) \geqslant 2 | X,\Theta\right)\right] && (4.155)\\ &= 1 - \left(1 - \frac{1}{2^{k-1}}\right)^n - \frac{n}{2^{k-1}}\left(1 - \frac{1}{2^{k-1}}\right)^{n-1}. && (4.156)\end{aligned}$$

Using the inequality $\log(1-x) \leqslant -x$ for all $x \in [0,1)$ yields,

$$\begin{aligned}\mathbb{P}\left(k_n(X,\Theta) \geqslant k\right) &\geqslant 1 - \exp\left(-\frac{n}{2^{k-1}}\right) - \frac{n}{2^{k-1}}\exp\left(-\frac{n-1}{2^{k-1}}\right) && (4.157)\\ &\geqslant 1 - \left(1 + \frac{n}{2^{k-1}}\right)\exp\left(-\frac{n}{2^{k-1}}\right). && (4.158)\end{aligned}$$

Letting $k = (1-\varepsilon_n)\log_2(n)$ in (4.158) yields

$$\mathbb{P}\left(k_n(X,\Theta) \geqslant k\right) \geqslant 1 - (1 + 2n^{\varepsilon_n})\exp\left(-2n^{\varepsilon_n}\right). \tag{4.159}$$

Note that, setting $\varepsilon_n = c_1(\log_2 n)^{-\alpha}$ for any $\alpha \in [0,1)$ implies that

$$n^{\varepsilon_n} = \exp\left(\varepsilon_n \log n\right) \tag{4.160}$$

tends to infinity. Therefore, for all $c_1 > 0$ and all $\alpha_i n[0,1)$,

$$\lim_{n\to\infty} \mathbb{P}\left(k_n(X,\Theta) \geqslant \log_2(n) - c_1(\log_2 n)^{-\alpha}\right) = 1. \tag{4.161}$$

Besides,

$$\begin{aligned}\mathbb{P}\left(k_n(X,\Theta) \leqslant k\right) &= 1 - \mathbb{P}\left(k_n(X,\Theta) > k\right) && (4.162)\\ &= \left(1 - \frac{1}{2^k}\right)^n - \frac{n}{2^k}\left(1 - + \frac{1}{2^k}\right)^{n-1}. && (4.163)\end{aligned}$$

Using the inequality $\log(1-x) \geqslant -x/(1-x)$ for all $x \in [0,1)$, we have

$$\begin{aligned}\mathbb{P}\left(k_n(X,\Theta) \leqslant k\right) &\geqslant \exp\left(-\frac{n}{2^k - 1}\right) + \frac{n}{2^k}\exp\left(-\frac{n-1}{2^k-1}\right) && (4.164)\\ &\geqslant \left(1 + \frac{n}{2^k}\right)\exp\left(-\frac{n}{2^k - 1}\right). && (4.165)\end{aligned}$$

Letting $k = (1+\varepsilon_n)\log_2(n)$ in (4.165) yields

$$\begin{aligned}\mathbb{P}\left(k_n(X,\Theta) \geqslant k\right) &\geqslant \left(1 + 2n^{-\varepsilon_n}\right)\exp\left(-\frac{n}{n^{1+\varepsilon_n} - 1}\right) && (4.166)\\ &\geqslant \left(1 + 2n^{-\varepsilon_n}\right)\exp\left(-\frac{n^{-\varepsilon_n}}{1 - \frac{1}{n^{1+\varepsilon_n}}}\right), && (4.167)\end{aligned}$$

which tends to 1 for the choice $\varepsilon_n = c_2(\log_2 n)^{-\alpha}$, for any $\alpha \in [0,1)$ and any $c_2 > 0$.

Proof of Theorem 4.5.1 (Consistency of Median RF)

Preliminary results In all the preliminary results, we use the fact that the spacing between two consecutive order statistics, that originate from an i.i.d. sample uniformly distributed on $[0,1]$ of size n_j is distributed as a beta distribution $Beta(1,n_j)$. We also recall that, for all α,β,

$$\mathbb{V}\left[\mathcal{B}(\alpha,\beta)\right] = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)} \quad \text{and} \quad \mathbb{E}\left[\mathcal{B}(\alpha,\beta)\right] = \frac{\alpha}{\alpha+\beta}. \tag{4.168}$$

Lemma S7 (Control of a cell side of a fully-developed median RF)**.** *Assume that $n \geqslant 16$ is a power of two. For all $x \in [0,1]^d$, for all $\ell \in \{1,\dots,d\}$ and depth $k \in \mathbb{N}^*$, with $k \leqslant \lfloor \log_2 n \rfloor$, we have*

$$\mathbb{E}\left[\mu\left(A_{k,n}^{(\ell)}(x,\Theta)\right)^2\right] \leqslant C_1\left(1-\frac{3}{4d}\right)^k, \tag{4.169}$$

with $C_1 \leqslant 256\exp\left(\frac{42+\sqrt{5}}{2-\sqrt{2}}\right)$.

Proof of Lemma S7. Fix $x \in [0,1]^d$. For all ℓ, let $\boldsymbol{\delta}_\ell(x,\Theta)$ be the vector whose components are defined as $\delta_{j,\ell}(x,\Theta) = 1$ if the j-th cut is made along direction ℓ and 0 otherwise. Without loss of generality, we let $\ell = 1$ and fix $x \in [0,1]^d$. For all $j \in \{0,\dots,k\}$, we denote $A_{j,n}^{(1)}(x,\Theta)$ the cell containing x at level j, projected onto the first direction, and $n_j = n2^{-j}$ the number of observations falling into this cell.

Recall that we consider the median forest in which splits are performed at the middle of two consecutive order statistics in a cell, so that each resulting cell contains exactly the same number of observations. With these notations in mind, we want to upper bound, for all j,

$$\mathbb{E}\left[\mu\left(A_{j,n}^{(1)}(x,\Theta)\right)^2 | \boldsymbol{\delta}_1(x,\Theta)\right],$$

where, for now, the split randomization $\boldsymbol{\delta}_1(x,\Theta)$ is considered fixed and may be omitted in the notations. Let us fix $j \leqslant k-1$, define

$$A_{j,n}^{(1)}(x,\Theta) = [M_{1,j}, M_{2,j}],$$

and assume that the next cut is made along the first axis at position M_j. Then,

$$\begin{aligned}
&\mu\left(A_{j+1,n}^{(1)}(x,\Theta)\right)^2 \\
&= (M_j - M_{1,j})^2 \mathbb{1}_{x\in[M_{1,j},M_j]} + (M_{2,j} - M_j)^2 \mathbb{1}_{x\in[M_j,M_{2,j}]} && (4.170)\\
&= (M_j - M_{1,j})^2 + \left((M_{2,j} - M_j)^2 - (M_j - M_{1,j})^2\right) \mathbb{1}_{x\in[M_j,M_{2,j}]} && (4.171)\\
&= (M_j - M_{1,j})^2 + (M_{1,j} + M_{2,j} - 2M_j)\,(M_{2,j} - M_{1,j}) \mathbb{1}_{x\in[M_j,M_{2,j}]}. && (4.172)
\end{aligned}$$

We denote $X'_1, ..., X'_{n_j}$ the points contained in the cell $A_{j,n}^{(1)}(x,\Theta)$. Note that the second term

in (4.172) can be decomposed as

$$
\begin{aligned}
&(M_{1,j} + M_{2,j} - 2M_j)\,(M_{2,j} - M_{1,j})\mathbb{1}_{x\in[M_j,M_{2,j}]} \\
&= \left(X'_{(1)} + X'_{(n_j)} - X'_{(n_j/2)} - X'_{(n_j/2+1)} + M_{1,j} - X'_{(1)} + M_{2,j} - X'_{(n_j)}\right)(M_{2,j} - M_{1,j})\mathbb{1}_{x\in[M_j,M_{2,j}]} \qquad (4.173)\\
&= \left(\frac{X'_{(1)} + X'_{(n_j)}}{2} - X'_{(n_j/2)} + \frac{X'_{(1)} + X'_{(n_j)}}{2} - X'_{(n_j/2+1)} + M_{1,j} - X'_{(1)} + M_{2,j} - X'_{(n_j)}\right) \\
&\quad \times (M_{2,j} - M_{1,j})\mathbb{1}_{x\in[M_j,M_{2,j}]} \qquad (4.174)\\
&\leqslant \left(\frac{X'_{(1)} + X'_{(n_j)}}{2} - X'_{(n_j/2)} + \frac{X'_{(1)} + X'_{(n_j)}}{2} - X'_{(n_j/2+1)} + M_{2,j} - X'_{(n_j)}\right)(M_{2,j} - M_{1,j}). \qquad (4.175)
\end{aligned}
$$

Injecting (4.175) into (4.172), taking the expectation and using Cauchy-Schwarz inequality leads to

$$
\begin{aligned}
\mathbb{E}\left[\mu\left(A^{(1)}_{j+1,n}(x,\Theta)\right)^2\right] \leqslant\ & \mathbb{E}\left[(M_j - M_{1,j})^2\right] \\
&+ \left(\mathbb{E}\left[\left(\frac{X'_{(1)} + X'_{(n_j)}}{2} - X'_{(n_j/2)}\right)^2\right]\mathbb{E}\left[(M_{2,j} - M_{1,j})^2\right]\right)^{1/2} \\
&+ \left(\mathbb{E}\left[\left(\frac{X'_{(1)} + X'_{(n_j)}}{2} - X'_{(n_j/2+1)}\right)^2\right]\mathbb{E}\left[(M_{2,j} - M_{1,j})^2\right]\right)^{1/2} \\
&+ \left(\mathbb{E}\left[\left(M_{2,j} - X'_{(n_j)}\right)^2\right]\mathbb{E}\left[(M_{2,j} - M_{1,j})^2\right]\right)^{1/2}. \qquad (4.176)
\end{aligned}
$$

Considering the second term, we have

$$
\begin{aligned}
&\mathbb{E}\left[\left(\frac{X'_{(1)} + X'_{(n_j)}}{2} - X'_{(n_j/2)}\right)^2\right] \\
&= \mathbb{E}\left[\left(\frac{X'_{(n_j)} - X'_{(1)}}{2} - (X'_{(n_j/2)} - X'_{(1)})\right)^2\right] \qquad (4.177)\\
&= \mathbb{E}\left[(X'_{(n_j)} - X'_{(1)})^2\mathbb{E}\left[\left(\frac{1}{2} - \frac{(X'_{(n_j/2)} - X'_{(1)})}{(X'_{(n_j)} - X'_{(1)})}\right)^2 | X'_{(1)}, X'_{(n_j)}\right]\right]. \qquad (4.178)
\end{aligned}
$$

where

$$
\frac{(X'_{(n_j/2)} - X'_{(1)})}{(X'_{(n_j)} - X'_{(1)})} | X'_{(1)}, X'_{(n_j)} \sim \mathcal{B}\left(\frac{n_j}{2} - 1, \frac{n_j}{2}\right),
$$

with $\mathbb{E}[\mathcal{B}(\frac{n_j}{2} - 1, \frac{n_j}{2})] = \frac{n_j - 2}{2(n_j - 1)}$.

Thus,

$$\mathbb{E}\left[\left(\frac{1}{2}-\frac{(X'_{(n_j/2)}-X'_{(1)})}{(X'_{(n_j)}-X'_{(1)})}\right)^2 |X'_{(1)},X'_{(n_j)}\right] = \left(\frac{1}{2}-\frac{n_j-2}{2(n_j-1)}\right)^2 + \mathbb{V}\left[\mathcal{B}\left(\frac{n_j}{2}-1,\frac{n_j}{2}\right)\right] \tag{4.179}$$

$$= \frac{1}{4(n_j-1)^2}+\frac{1}{4}\frac{n_j-2}{(n_j-1)^2} \tag{4.180}$$

$$= \frac{1}{4(n_j-1)}. \tag{4.181}$$

Consequently,

$$\mathbb{E}\left[\left(\frac{X'_{(1)}+X'_{(n_j)}}{2}-X'_{(n_j/2)}\right)^2\right] = \frac{1}{4(n_j-1)}\mathbb{E}\left[(X'_{(n_j)}-X'_{(1)})^2\right] \tag{4.182}$$

$$\leqslant \frac{1}{4(n_j-1)}\mathbb{E}\left[(M_{2,j}-M_{1,j})^2\right]. \tag{4.183}$$

Similarly,

$$\mathbb{E}\left[\left(\frac{X'_{(1)}+X'_{(n_j)}}{2}-X'_{(n_j/2+1)}\right)^2\right] = \frac{1}{4(n_j-1)}\mathbb{E}\left[(X'_{(n_j)}-X'_{(1)})^2\right] \tag{4.184}$$

$$\leqslant \frac{1}{4(n_j-1)}\mathbb{E}\left[(M_{2,j}-M_{1,j})^2\right]. \tag{4.185}$$

By Lemma S8,

$$\mathbb{E}\left[\left(M_{2,j}-X'_{(n_j)}\right)^2\right] \leqslant \frac{5}{(n_j-1)^2}\mathbb{E}\left[(M_{2,j}-M_{1,j})^2\right].$$

Gathering all previous inequalities into (4.172) yields

$$\mathbb{E}\left[\mu\left(A^{(1)}_{j+1,n}(x,\Theta)\right)^2\right] \leqslant \mathbb{E}\left[(M_j-M_{1,j})^2\right]+\frac{1}{\sqrt{n_j-1}}\mathbb{E}\left[(M_{2,j}-M_{1,j})^2\right] + \frac{\sqrt{5}}{(n_j-1)}\mathbb{E}\left[(M_{2,j}-M_{1,j})^2\right]. \tag{4.186}$$

Considering the first term in (4.186), we have

$$(M_j-M_{1,j})^2 = \left(\frac{X'_{(n_j/2)}+X'_{(n_j/2+1)}}{2}-X'_{(1)}+X'_{(1)}-M_{1,j}\right)^2 \tag{4.187}$$

$$\leqslant \left(X'_{(n_j/2+1)}-X'_{(1)}+X'_{(1)}-M_{1,j}\right)^2 \tag{4.188}$$

$$\leqslant \left(X'_{(n_j/2+1)}-X'_{(1)}\right)^2+\left(X'_{(1)}-M_{1,j}\right)^2+2\left(X'_{(n_j/2+1)}-X'_{(1)}\right)\left(X'_{(1)}-M_{1,j}\right).$$

Taking the expectation and using Cauchy-Schwarz inequality, we obtain

$$\mathbb{E}\left[(M_j - M_{1,j})^2\right] \leqslant \mathbb{E}\left[\left(X'_{(n_j/2+1)} - X'_{(1)}\right)^2\right] + \mathbb{E}\left[\left(X'_{(1)} - M_{1,j}\right)^2\right] + 2\left(\mathbb{E}\left[\left(X'_{(n_j/2+1)} - X'_{(1)}\right)^2\right]\mathbb{E}\left[\left(X'_{(1)} - M_{1,j}\right)^2\right]\right)^{1/2}. \tag{4.189}$$

Now,

$$\mathbb{E}\left[\left(X'_{(n_j/2+1)} - X'_{(1)}\right)^2\right] = \mathbb{E}\left[\left(X'_{(n_j)} - X'_{(1)}\right)^2 \mathbb{E}\left[\left(\frac{X'_{(n_j/2+1)} - X'_{(1)}}{X'_{(n_j)} - X'_{(1)}}\right)^2 | X'_{(1)}, X'_{(n_j)}\right]\right], \tag{4.190}$$

where

$$\mathbb{E}\left[\left(\frac{X'_{(n_j/2+1)} - X'_{(1)}}{X'_{(n_j)} - X'_{(1)}}\right)^2 | X'_{(1)}, X'_{(n_j)}\right] = \mathbb{E}\left[\mathcal{B}\left(\frac{n_j}{2}, \frac{n_j}{2} - 1\right)^2\right] \tag{4.191}$$

$$= \mathbb{V}\left[\mathcal{B}\left(\frac{n_j}{2}, \frac{n_j}{2} - 1\right)\right] + \left(\mathbb{E}\left[\mathcal{B}\left(\frac{n_j}{2}, \frac{n_j}{2} - 1\right)\right]\right)^2 \tag{4.192}$$

$$= \frac{\frac{n_j}{2}\left(\frac{n_j}{2} - 1\right)}{(n_j - 1)^2 n_j} + \left(\frac{n_j/2}{n_j - 1}\right)^2 \tag{4.193}$$

$$= \frac{1}{4}\frac{n_j - 2}{(n_j - 1)^2} + \left(\frac{1}{2}\frac{n_j}{n_j - 1}\right)^2 \tag{4.194}$$

$$= \frac{1}{4}\frac{n_j^2 + n_j - 2}{(n_j - 1)^2} \tag{4.195}$$

$$\leqslant \frac{1}{4}\frac{(n_j + 1/2)^2}{(n_j - 1)^2}. \tag{4.196}$$

Therefore,

$$\mathbb{E}\left[\left(X'_{(n_j/2+1)} - X'_{(1)}\right)^2\right] \leqslant \frac{1}{4}\frac{(n_j + 1/2)^2}{(n_j - 1)^2}\mathbb{E}\left[\left(X'_{(n_j)} - X'_{(1)}\right)^2\right]. \tag{4.197}$$

Injecting this expression into (4.189), we have

$$\mathbb{E}\left[(M_j - M_{1,j})^2\right] \leqslant \frac{1}{4}\frac{(n_j + 1/2)^2}{(n_j - 1)^2}\mathbb{E}\left[\left(X'_{(n_j)} - X'_{(1)}\right)^2\right] + \mathbb{E}\left[\left(X'_{(1)} - M_{1,j}\right)^2\right] + \frac{(n_j + 1/2)}{(n_j - 1)}\left(\mathbb{E}\left[\left(X'_{(n_j)} - X'_{(1)}\right)^2\right]\mathbb{E}\left[\left(X'_{(1)} - M_{1,j}\right)^2\right]\right)^{1/2}. \tag{4.198}$$

According to Technical Lemma S8, we have

$$\mathbb{E}\left[\left(X'_{(1)} - M_{1,j}\right)^2\right] \leqslant \frac{5}{(n_j - 1)^2}\mathbb{E}\left[M_{2,j} - M_{1,j}\right].$$

Hence,

$$\begin{aligned}
&\mathbb{E}\left[(M_j - M_{1,j})^2\right] \\
&\leqslant \frac{1}{4}\frac{(n_j+1/2)^2}{(n_j-1)^2}\mathbb{E}\left[(M_{2,j} - M_{1,j})^2\right] + \frac{5}{(n_j-1)^2}\mathbb{E}\left[(M_{2,j} - M_{1,j})^2\right] \\
&\quad + \frac{(n_j+1/2)}{(n_j-1)}\left(\mathbb{E}\left[(M_{2,j} - M_{1,j})^2\right]\frac{5}{(n_j-1)^2}\mathbb{E}\left[(M_{2,j} - M_{1,j})^2\right]\right)^{1/2} && (4.199)\\
&\leqslant \left(\frac{1}{4}\frac{(n_j+1/2)^2}{(n_j-1)^2} + \frac{5}{(n_j-1)^2} + \frac{(n_j+1/2)\sqrt{5}}{(n_j-1)^2}\right)\mathbb{E}\left[(M_{2,j} - M_{1,j})^2\right] && (4.200)\\
&\leqslant \frac{1}{4}\frac{(n_j+1/2)^2}{(n_j-1)^2}\left(1 + \frac{20}{(n_j+1/2)^2} + \frac{4\sqrt{5}}{(n_j+1/2)}\right)\mathbb{E}\left[(M_{2,j} - M_{1,j})^2\right] && (4.201)\\
&\leqslant \frac{1}{4}\left(1 + \frac{3}{2(n_j-1)}\right)^2\left(1 + \frac{20}{(n_j+1/2)^2} + \frac{4\sqrt{5}}{(n_j+1/2)}\right)\mathbb{E}\left[(M_{2,j} - M_{1,j})^2\right] && (4.202)\\
&\leqslant \frac{1}{4}\left(1 + \frac{9}{2(n_j-1)}\right)\left(1 + \frac{20}{(n_j+1/2)^2} + \frac{4\sqrt{5}}{(n_j+1/2)}\right)\mathbb{E}\left[(M_{2,j} - M_{1,j})^2\right], && (4.203)
\end{aligned}$$

for all $n_j \geqslant 4$, since $(1+x)^2 \leqslant 1 + 3x$ if $x \leqslant 1$.

Consequently,

$$\begin{aligned}
&\mathbb{E}\left[(M_j - M_{1,j})^2\right] \\
&\leqslant \frac{1}{4}\left(1 + \frac{9}{2(n_j-1)}\right)\left(1 + \frac{30}{n_j-1}\right)\mathbb{E}\left[(M_{2,j} - M_{1,j})^2\right] && (4.204)\\
&\leqslant \frac{1}{4}\left(1 + \frac{69}{2(n_j-1)} + \frac{90}{(n_j-1)^2}\right)\mathbb{E}\left[(M_{2,j} - M_{1,j})^2\right] && (4.205)\\
&\leqslant \frac{1}{4}\left(1 + \frac{35+6}{n_j-1}\right)\mathbb{E}\left[(M_{2,j} - M_{1,j})^2\right] && (4.206)\\
&\leqslant \frac{1}{4}\left(1 + \frac{41}{n_j-1}\right)\mathbb{E}\left[(M_{2,j} - M_{1,j})^2\right], && (4.207)
\end{aligned}$$

for all $n_j \geqslant 16$. Recall that, until now, we have fixed $\boldsymbol{\delta}_1(x,\Theta)$ and omitted the explicit conditioning in the proof to lighten notations. Thus, plugging-in the previous inequality into (4.186) yields, for all $n_j \geqslant 16$,

$$\begin{aligned}
&\mathbb{E}\left[\mu\left(A^{(1)}_{j+1,n}(x,\Theta)\right)^2 |\boldsymbol{\delta}_1(x,\Theta)\right] \\
&\leqslant \frac{1}{4}\left(1 + \frac{41}{n_j-1}\right)\mathbb{E}\left[(M_{2,j} - M_{1,j})^2 |\boldsymbol{\delta}_1(x,\Theta)\right] \\
&\quad + \frac{1}{\sqrt{n_j-1}}\mathbb{E}\left[(M_{2,j} - M_{1,j})^2 \big|\boldsymbol{\delta}_1(x,\Theta)\right] + \frac{\sqrt{5}}{(n_j-1)}\mathbb{E}\left[(M_{2,j} - M_{1,j})^2 |\boldsymbol{\delta}_1(x,\Theta)\right] && (4.208)\\
&\leqslant \frac{1}{4}\left(1 + \frac{42+\sqrt{5}}{\sqrt{n_j-1}}\right)\mathbb{E}\left[\mu\left(A^{(1)}_{j,n}(x,\Theta)\right)^2 |\boldsymbol{\delta}_1(x,\Theta)\right]. && (4.209)
\end{aligned}$$

Recall that $\boldsymbol{\delta}_1(x,\Theta)$ is the vector whose components are defined as $\delta_{j,1}(x,\Theta)=1$ if the j-th cut is made along the first direction and 0 otherwise. We let $K_1 = \|\boldsymbol{\delta}_1(x,\Theta)\|_1$ be the number of times the first direction is split. By induction, we have

$$\mathbb{E}\left[\mu\left(A_{k,n}^{(1)}(x,\Theta)\right)^2\right] = \mathbb{E}\left[\mathbb{E}\left[\mu\left(A_{k,n}^{(1)}(x,\Theta)\right)^2 |\boldsymbol{\delta}_1(x,\Theta)\right]\right] \tag{4.210}$$

$$\leqslant \mathbb{E}\left[\prod_{\substack{j:\delta_{j,1}=1,\\ j\leqslant k-4}} \frac{1}{4}\left(1+\frac{42+\sqrt{5}}{\sqrt{n_j-1}}\right)\right] \tag{4.211}$$

$$\leqslant 4^4\mathbb{E}\left[4^{-K_1}\prod_{\substack{j:\delta_{j,1}=1,\\ j\leqslant k-4}} \left(1+\frac{42+\sqrt{5}}{\sqrt{n_j-1}}\right)\right]. \tag{4.212}$$

The product can be upper bounded as follows, with $C = 42+\sqrt{5}$,

$$\log\left(\prod_{j,\delta_{j,l}=1,j\leqslant k-4}\left(1+\frac{C}{\sqrt{n_j-1}}\right)\right) \leqslant \log\left(\prod_{\substack{j:\delta_{j,1}=1,\\ j\leqslant k-4}}\left(1+\frac{C}{\sqrt{n_{j+1}}}\right)\right) \tag{4.213}$$

$$= \sum_{\substack{j:\delta_{j,1}=1,\\ j\leqslant k-4}} \log\left(1+\frac{C\sqrt{2}\cdot 2^{j/2}}{n^{1/2}}\right) \tag{4.214}$$

$$\leqslant \frac{C\sqrt{2}}{n^{1/2}}\sum_{j=0}^{k-4} 2^{j/2} \tag{4.215}$$

$$\leqslant \frac{C\sqrt{2}}{\sqrt{2}-1}\frac{2^{(k-3)/2}}{n^{1/2}} \tag{4.216}$$

$$\leqslant \frac{C}{2\sqrt{2}-2}. \tag{4.217}$$

Thus,

$$\mathbb{E}\left[\mu\left(A_{k,n}^{(1)}(x,\Theta)\right)^2\right] \leqslant 4^4\exp\left(\frac{C}{2\sqrt{2}-2}\right)\mathbb{E}\left[4^{-K_1}\right]. \tag{4.218}$$

Since $K_1 \sim \text{Bin}(k,1/d)$, we have

$$\mathbb{E}\left[4^{-K_1}\right] = \left(1-\frac{1}{d}+\frac{1}{4d}\right)^k \tag{4.219}$$

$$= \left(1-\frac{3}{4d}\right)^k. \tag{4.220}$$

Finally,

$$\mathbb{E}\left[\mu\left(A_{k,n}^{(1)}(x,\Theta)\right)^2\right] \leqslant 4^4\exp\left(\frac{C}{2\sqrt{2}-2}\right)\left(1-\frac{3}{4d}\right)^k, \tag{4.221}$$

with $C = 42 + \sqrt{5}$.

□

Lemma S8 (Technical Lemma). *1. Let $x \in [0,1]^d$ and consider the cell $A_{n,j}(x,\Theta)$ containing x at depth $j \leqslant k-1$. W.l.o.g. restrict the study to the one-dimensional cell $A^{(1)}_{n,j}(x,\Theta)$ corresponding to the cell $A_{n,j}(x,\Theta)$ along the first dimension only, and set $A^{(1)}_{n,j}(x,\Theta) = [M_{1,j}; M_{2,j}]$. The one-dimensional cell $A^{(1)}_{n,j}(x,\Theta)$ contains n_j points denoted $X'_1, \dots X'_{n_j}$ (random subsample of the initial training sample). Call $X'_{(1)}, \dots, X'_{(n_j)}$ the ordered version of $X'_1, \dots X'_{n_j}$. Then,*

$$\mathbb{E}\left[\left(X'_{(1)} - M_{1,j}\right)^2\right] \leqslant \frac{5}{(n_j-1)^2}\mathbb{E}\left[\left(M_{2,j} - M_{1,j}\right)^2\right],$$

and

$$\mathbb{E}\left[\left(M_{2,j} - X'_{(n_j)}\right)^2\right] \leqslant \frac{5}{(n_j-1)^2}\mathbb{E}\left[\left(M_{2,j} - M_{1,j}\right)^2\right].$$

2. Consider now the cell $A_{n,j}(X_1,\Theta)$ containing X_1 at depth $j \leqslant k-1$. W.l.o.g. restrict the study to the one-dimensional cell $A^{(1)}_{n,j}(X_1,\Theta)$ corresponding to the cell $A_{n,j}(X_1,\Theta)$ along the first dimension only, and set $A^{(1)}_{n,j}(X_1,\Theta) = [M_{1,j}; M_{2,j}] = [M_{1,j}(X_1,\Theta); M_{2,j}(X_1,\Theta)]$. The one-dimensional cell $A^{(1)}_{n,j}(X_1,\Theta)$ contains n_j points denoted $\{X'_1, \dots X'_{n_j-1}\} \cup \{X_1\}$ (random subsample of the initial training sample containing X_1 and projected on the first axis). Call $X'_{(1)}, \dots, X'_{(n_j-1)}$ the ordered version of $X'_1, \dots X'_{n_j-1}$. Then,

$$\mathbb{E}\left[X'_{(1)} - M_{1,j} | X_1\right] \leqslant \frac{1}{n_j}\mathbb{E}\left[(M_{2,j} - M_{1,j}) | X_1\right],$$

and

$$\mathbb{E}\left[M_{2,j} - X'_{(n_j-1)} | X_1\right] \leqslant \frac{1}{n_j}\mathbb{E}\left[(M_{2,j} - M_{1,j}) | X_1\right].$$

Proof of Lemma S8.

Notations W.l.o.g. consider the following development according to the first direction only. Let $x \in [0,1]^d$. Recall that we consider the cell $A^{(1)}_{j,n}(x,\Theta) = [M_{1,j}; M_{2,j}]$ containing x at depth $j \leqslant k-1$. The cut at $M_{1,j}$ (resp. $M_{2,j}$) has been obtained at an anterior depth $j_1 \leqslant j$ (resp. $j_2 \leqslant j$), as the middle of two order statistics of a previous subsample:

$$M_{1,j} = \frac{M_{1,j,-} + M_{1,j,+}}{2} \quad \text{and} \quad M_{2,j} = \frac{M_{2,j,-} + M_{2,j,+}}{2},$$

with $M_{1,j,-} < M_{1,j,+}$ and $M_{2,j,-} < M_{2,j,+}$. The following computations can be also conducted in a similar way when $M_{1,j} = 0$ or $M_{2,j} = 1$. The current cell $A^{(1)}_{j,n}(x,\Theta)$, includes now n_j points of the original training sample, which are denoted by $X'_1, \dots, X'_{n_j}$. Remark that as $M_{1,j,-}$ and $M_{2,j,+}$ refer to anterior order statistics of a previous subsample (including the points $X'_1, \dots, X'_{n_j}$), then $X'_1, \dots, X'_{n_j}$ are i.i.d. uniformly distributed in $[M_{j,1,-}; M_{2,j,+}]$. Denote by $X'_{(1)}, \dots, X'_{(n_j)}$, the ordered statistics of the current subsample $X'_1, \dots, X'_{n_j}$ in $A^{(1)}_{j,n}(x,\Theta)$ for some fixed x.

First statement - Control of $\mathbb{E}[(X'_{(1)} - M_{1,j})^2]$. We have

$$\mathbb{E}\left[\left(X'_{(1)} - M_{1,j}\right)^2\right] \leqslant 2\mathbb{E}\left[(M_{1,j,+} - M_{1,j})^2\right] + 2\mathbb{E}\left[\left(X'_{(1)} - M_{1,j,+}\right)^2\right]. \tag{4.222}$$

Note that, by definition of $M_{1,j}$, the quantity $M_{1,j,+} - M_{1,j}$ corresponds to a half spacing between two points in the cell previously built by cutting on the first direction at depth j_1, denoted $A^{(1)}_{j_1,n}(x,\Theta)$. By construction, the spacings between two consecutive points in $A^{(1)}_{j_1,n}(x,\Theta)$ were the same in distribution. Since points have been removed between $A^{(1)}_{j,n}(x,\Theta)$ and $A^{(1)}_{j_1,n}(x,\Theta)$, the spacings are larger between consecutive points in $A^{(1)}_{j,n}(x,\Theta)$ than between consecutive points in $A^{(1)}_{j_1,n}(x,\Theta)$. This leads to

$$M_{1,j,+} - M_{1,j} = \frac{M_{1,j,+} - M_{1,j,-}}{2} \leqslant \frac{X'_{(2)} - X'_{(1)}}{2}.$$

Therefore, since all variables are bounded,

$$\begin{aligned}\mathbb{E}\left[(M_{1,j,+} - M_{1,j})^2\right] &\leqslant \frac{1}{4}\mathbb{E}\left[\left(X'_{(2)} - X'_{(1)}\right)^2\right] \\ &\leqslant \frac{1}{4}\mathbb{E}\left[(X'_{(n_j)} - X'_{(1)})^2\mathbb{E}\left[\frac{\left(X'_{(2)} - X'_{(1)}\right)^2}{\left(X'_{(n_j)} - X'_{(1)}\right)^2}|X'_{(1)}, X'_{(n_j)}\right]\right].\end{aligned}$$

Regarding the inner expectation,

$$\begin{aligned}\mathbb{E}\left[\frac{\left(X'_{(2)} - X'_{(1)}\right)^2}{\left(X'_{(n_j)} - X'_{(1)}\right)^2}|X'_{(1)}, X'_{(n_j)}\right] &= \mathbb{E}\left[\mathcal{B}\left(1, n_j - 2\right)^2\right] &(4.223)\\ &= \mathbb{V}\left[\mathcal{B}\left(1, n_j - 2\right)\right] + \left(\mathbb{E}\left[\mathcal{B}\left(1, n_j - 2\right)\right]\right)^2 &(4.224)\\ &= \frac{n_j - 2}{(n_j - 1)^2 n_j} + \left(\frac{1}{n_j - 1}\right)^2 &(4.225)\\ &\leqslant \frac{2}{(n_j - 1)^2}. &(4.226)\end{aligned}$$

Finally,

$$\begin{aligned}\mathbb{E}\left[(M_{1,j,+} - M_{1,j})^2\right] &\leqslant \frac{1}{2(n_j - 1)^2}\mathbb{E}\left[\left(X'_{(n_j)} - X'_{(1)}\right)^2\right] &(4.227)\\ &\leqslant \frac{1}{2(n_j - 1)^2}\mathbb{E}\left[(M_{2,j} - M_{1,j})^2\right]. &(4.228)\end{aligned}$$

Regarding the second term in (4.222), we have

$$\begin{aligned}
\mathbb{E}\left[\left(X'_{(1)} - M_{1,j,+}\right)^2\right] &= \mathbb{E}\left[\mathbb{E}\left[\left(X'_{(1)} - M_{1,j,+}\right)^2 | M_{1,j,+}, X_{M_{2,j},-}\right]\right] \\
&= \mathbb{E}\left[\left(M_{2,j,-} - M_{1,j,+}\right)^2 \mathbb{E}\left[\left(\frac{X'_{(1)} - M_{1,j,+}}{M_{2,j,-} - M_{1,j,+}}\right)^2 | M_{1,j,+}, M_{2,j,-}\right]\right] \\
&\leqslant \mathbb{E}\left[\left(M_{2,j,-} - M_{1,j,+}\right)^2 \mathbb{E}\left[\mathcal{B}(1, n_j - 1)^2\right]\right] \\
&\leqslant \frac{2}{n_j^2}\mathbb{E}\left[\left(M_{2,j,-} - M_{1,j,+}\right)^2\right] \\
&\leqslant \frac{2}{n_j^2}\mathbb{E}\left[\left(M_{2,j} - M_{1,j}\right)^2\right].
\end{aligned}$$

Finally,

$$\begin{aligned}
\mathbb{E}\left[\left(X'_{(1)} - M_{1,j}\right)^2\right] &\leqslant \left(\frac{1}{(n_j-1)^2} + \frac{4}{n_j^2}\right)\mathbb{E}\left[\left(M_{2,j} - M_{1,j}\right)^2\right] \\
&\leqslant \frac{5}{(n_j-1)^2}\mathbb{E}\left[\left(M_{2,j} - M_{1,j}\right)^2\right].
\end{aligned}$$

The second point of the first statement can be proved in the exact same manner.

Second statement - Control of $\mathbb{E}[X'_{(1)} - M_{1,j}|X_1]$. In this part, we study the cell $A^{(1)}_{n,j}(X_1, \Theta)$. The cell $A^{(1)}_{n,j}(X_1, \Theta)$ contains n_j data points (including X_1). We denote by $X'_1, \ldots, X'_{n_j-1}$ the observations falling into $A^{(1)}_{n,j}(X_1, \Theta)$, different from X_1. Note that, these $n_j - 1$ observations are still i.i.d. uniformly distributed in $[M_{1,j,-}; M_{2,j,+}]$. We denote by $X'_{(1)}, \ldots, X'_{(n_j-1)'}$ the subsample $X'_1, \ldots, X'_{n_j-1}$. We have

$$M_{2,j} - M_{1,j} = M_{2,j} - X'_{(n_j-1)} + \sum_{q=1}^{n_j-2}\left(X'_{(n_j-q)} - X'_{(n_j-q-1)}\right) + X'_{(1)} - M_{1,j}. \tag{4.229}$$

Thus,

$$\mathbb{E}\left[X'_{(1)} - M_{1,j}|X_1\right] + \mathbb{E}\left[M_{2,j} - X'_{(n_j-1)}|X_1\right] = \mathbb{E}\left[M_{2,j} - M_{1,j}|X_1\right] - (n_j - 2)\mathbb{E}\left[X'_{(2)} - X'_{(1)}|X_1\right]. \tag{4.230}$$

The variables $X'_{(1)}$ and $X'_{(2)}$ being order statistics of a subsample independent of X_1, one gets

$$\mathbb{E}\left[X'_{(2)} - X'_{(1)}|X_1\right] = \mathbb{E}\left[\mathbb{E}\left[\frac{X'_{(2)} - X'_{(1)}}{M_{2,j,+} - M_{1,j,-}}|X_1, M_{1,j,-}, M_{2,j,+}\right](M_{2,j,+} - M_{1,j,-})|X_1\right] \tag{4.231}$$

$$= \mathbb{E}\left[\mathbb{E}\left[\mathcal{B}(1, n_j - 1)\right](M_{2,j,+} - M_{1,j,-})|X_1\right] \tag{4.232}$$

$$= \frac{1}{n_j}\mathbb{E}\left[(M_{2,j,+} - M_{1,j,-})|X_1\right] \tag{4.233}$$

$$\geqslant \frac{1}{n_j}\mathbb{E}\left[(M_{2,j} - M_{1,j})|X_1\right]. \tag{4.234}$$

Finally,

$$\mathbb{E}\left[X'_{(1)} - M_{1,j}|X_1\right] + \mathbb{E}\left[M_{2,j} - X'_{(n_j-1)}|X_1\right] \leqslant \left(1 - \frac{n_j - 2}{n_j}\right)\mathbb{E}\left[(M_{2,j} - M_{1,j})|X_1\right], \tag{4.235}$$

$$= \frac{2}{n_j}\mathbb{E}\left[(M_{2,j} - M_{1,j})|X_1\right], \tag{4.236}$$

and, by symmetry,

$$\mathbb{E}\left[X'_{(1)} - M_{1,j}|X_1\right] \leqslant \frac{1}{n_j}\mathbb{E}\left[(M_{2,j} - M_{1,j})|X_1\right], \tag{4.237}$$

and

$$\mathbb{E}\left[M_{2,j} - X'_{(n_j-1)}|X_1\right] \leqslant \frac{1}{n_j}\mathbb{E}\left[(M_{2,j} - M_{1,j})|X_1\right]. \tag{4.238}$$

□

Lemma S9 (Control of the leaf side and volume of a fully developed median RF)**.** *Assume that $n \geqslant 4$ is a power of two. Consider a median tree of depth k and denote $A_{n,k}(X_1, \Theta)$ the leaf containing X_1. For all $\ell \in \{1, \ldots, d\}$, we denote K_ℓ the number of splits along the ℓ-th direction and $n_j = 2^{-j}$. Let also $\boldsymbol{\delta}_\ell(X_1, \Theta)$ be the vector whose components are defined as $\delta_{j,\ell}(X_1, \Theta) = 1$ if the j-th cut of the cell $A_n^{(\ell)}(X_1, \Theta)$ is made along direction ℓ and 0 otherwise. Then,*

$$\mathbb{E}\left[\mu(A_{n,k}^{(\ell)}(X_1, \Theta))|X_1, \boldsymbol{\delta}_\ell(X_1, \Theta)\right] \leqslant 2^{-K_\ell+2} \prod_{\substack{j:\delta_{j,\ell}=1,\\ j\leqslant k-2}} \left(1 + \frac{2}{\sqrt{n_j - 1}}\right). \tag{4.239}$$

In particular, letting $C_2 = 4\exp(5/(\sqrt{2}-1))$, we have

$$\mathbb{E}\left[\mu(A_{n,k}^{(\ell)}(X_1, \Theta))|X_1, \boldsymbol{\delta}_\ell(X_1, \Theta)\right] \leqslant C_2\, 2^{-K_\ell}, \tag{4.240}$$

and

$$\mathbb{E}\left[\mu(A_{n,k}(X_1, \Theta))|X_1, \boldsymbol{\delta}_1(X_1, \Theta), \ldots, \boldsymbol{\delta}_d(X_1, \Theta)\right] \leqslant C_2 2^{-k}. \tag{4.241}$$

Proof. We write $A_{n,j}^{(\ell)}(X_1, \Theta) = [M_{1,j}, M_{2,j}]$ the cell of the RF containing X_1 along the direction ℓ,

at depth j. To lighten the notations, we omit the dependencies in X_1, in Θ and in ℓ. We also write $X'_1, ..., X'_{n_j}$ the data points contained in the cell $A^{(\ell)}_{n,j}(X_1, \Theta)$, and we denote by $X'_{(1)}, ..., X'_{(n_j-1)}$ the ordered version of $\{X'_1, ..., X'_{n_j}\} \setminus \{X_1\}$. We suppose that the next cut is occurring on the ℓ-th direction and compute the size of the new cell containing X_1, $A^{(\ell)}_{n,j+1}(X_1, \Theta)$, so that 4 different events are possible:

1. X_1 is in the first "part" of the cell, i.e. $X_1 \in \left[M_{1,j}, X'_{(n_j/2-1)}\right]$;
2. X_1 is in the second "part" of the cell, i.e. , $X_1 \in \left[X'_{(n_j/2+1)}, M_{2,j}\right]$;
3. X_1 is in the "middle (left)" of the cell, i.e. $X_1 \in \left[X'_{(n_j/2-1)}, X'_{(n_j/2)}\right]$;
4. X_1 is in the "middle (right)" of the cell, $X_1 \in \left[X'_{(n_j/2)}, X'_{(n_j/2+1)}\right]$.

The length of the following cell can be therefore decomposed with respect to the previous events:

$$
\begin{aligned}
&\mu\left(A^{(\ell)}_{n,j+1}(X_1, \Theta)\right) \\
&= \left(\frac{X'_{(n_j/2-1)} + X'_{(n_j/2)}}{2} - M_{1,j}\right) \mathbb{1}_{X_1 \in \left[M_{1,j}, X'_{(n_j/2-1)}\right]} \\
&\quad + \left(M_{2,j} - \frac{X'_{(n_j/2)} + X'_{(n_j/2+1)}}{2}\right) \mathbb{1}_{X_1 \in \left[X'_{(n_j/2+1)}, M_{2,j}\right]} \\
&\quad + \left(\frac{X_1 + X'_{(n_j/2)}}{2} - M_{1,j}\right) \mathbb{1}_{X_1 \in \left[X'_{(n_j/2-1)}, X'_{(n_j/2)}\right]} \\
&\quad + \left(M_{2,j} - \frac{X'_{(n_j/2)} + X_1}{2}\right) \mathbb{1}_{X_1 \in \left[X'_{(n_j/2)}, X'_{(n_j/2+1)}\right]} && (4.242) \\
&\leqslant \left(X'_{(n_j/2)} - M_{1,j}\right) \mathbb{1}_{X_1 \in \left[M_{1,j}, X'_{(n_j/2-1)}\right]} + \left(M_{2,j} - X'_{(n_j/2)}\right) \mathbb{1}_{X_1 \in \left[X'_{(n_j/2+1)}, M_{2,j}\right]} && (4.243) \\
&\quad + \left(X'_{(n_j/2)} - M_{1,j}\right) \mathbb{1}_{X_1 \in \left[X'_{(n_j/2-1)}, X'_{(n_j/2)}\right]} + \left(M_{2,j} - X'_{(n_j/2)}\right) \mathbb{1}_{X_1 \in \left[X'_{(n_j/2)}, X'_{(n_j/2+1)}\right]} && (4.244) \\
& && (4.245)
\end{aligned}
$$

$$
\begin{aligned}
&\mu\left(A^{(\ell)}_{n,j+1}(X_1,\Theta)\right)\\
&\leqslant \left(X'_{(n_j/2)}-M_{1,j}\right)\mathbb{1}_{X_1\in[M_{1,j},X'_{(n_j/2)}]}+\left(M_{2,j}-X'_{(n_j/2)}\right)\mathbb{1}_{X_1\in[X'_{(n_j/2)},M_{2,j}]} &(4.246)\\
&=\left(X'_{(n_j/2)}-M_{1,j}\right)+2\left(\frac{M_{1,j}+M_{2,j}}{2}-X'_{(n_j/2)}\right)\mathbb{1}_{X_1\in[X'_{(n_j/2)},M_{2,j}]} &(4.247)\\
&=\left(X'_{(n_j/2)}-X'_{(1)}\right)+\left(X'_{(1)}-M_{1,j}\right)+2\left(\frac{X'_{(1)}+X'_{(n_j-1)}}{2}-X'_{(n_j/2)}\right)\mathbb{1}_{X_1\in[X'_{(n_j/2)},M_{2,j}]}\\
&\quad-\left(\left(X'_{(1)}+X'_{(n_j-1)}\right))-(M_{1,j}+M_{2,j})\right)\mathbb{1}_{X_1\in[X'_{(n_j/2)},M_{2,j}]} &(4.248)\\
&\leqslant \left(X'_{(n_j/2)}-X'_{(1)}\right)+\left(X'_{(1)}-M_{1,j}\right)+2\left(\frac{X'_{(1)}+X'_{(n_j-1)}}{2}-X'_{(n_j/2)}\right)\mathbb{1}_{X_1\in[X'_{(n_j/2)},M_{2,j}]}\\
&\quad+(M_{2,j}-X'_{(n_j-1)}). &(4.249)
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
&\mathbb{E}\left[\mu\left(A^{(\ell)}_{n,j+1}(X_1,\Theta)\right)\middle|X_1,\boldsymbol{\delta}_\ell(X_1,\Theta)\right]\\
&\leqslant \mathbb{E}\left[(X'_{(n_j-1)}-X'_{(1)})\mathbb{E}\left[\frac{X'_{(n_j/2)}-X'_{(1)}}{X'_{(n_j-1)}-X'_{(1)}}|X_1,\boldsymbol{\delta}_\ell(X_1,\Theta),X'_{(1)},X'_{(n_j-1)}\right]\middle|X_1,\boldsymbol{\delta}_\ell(X_1,\Theta)\right]\\
&\quad+2\mathbb{E}\left[\mathbb{E}\left[\left|\frac{X'_{(n_j-1)}+X'_{(1)}}{2}-X'_{(n_j/2)}\right|\middle|X_1,\boldsymbol{\delta}_\ell(X_1,\Theta),X'_{(1)},X'_{(n_j-1)}\right]\middle|X_1,\boldsymbol{\delta}_\ell(X_1,\Theta)\right]\\
&\quad+\mathbb{E}\left[X'_{(1)}-M_{1,j}\middle|X_1,\boldsymbol{\delta}_\ell(X_1,\Theta)\right]+\mathbb{E}\left[M_{2,j}-X'_{(n_j-1)}\middle|X_1,\boldsymbol{\delta}_\ell(X_1,\Theta)\right]. &(4.250)
\end{aligned}
$$

Regarding the first term of (4.250), remark that by property of the uniform distribution, conditional on $X'_{(1)}$ and $X'_{(n_j-1)}$, the order statistics between 1 and n_j-1 follow Beta distributions independently from X_1 and $\boldsymbol{\delta}_\ell(X_1,\Theta)$. Therefore,

$$
\frac{X'_{(n_j/2)}-X'_{(1)}}{X'_{(n_j-1)}-X'_{(1)}}|X'_{(1)},X'_{(n_j-1)}\sim\mathcal{B}(n_j/2-1,n_j/2-1),
$$

so that

$$
\mathbb{E}\left[\frac{X'_{(n_j/2)}-X'_{(1)}}{X'_{(n_j-1)}-X'_{(1)}}\middle|X_1,\boldsymbol{\delta}_\ell(X_1,\Theta),X'_{(1)},X'_{(n_j-1)}\right]=\frac{n_j/2-1}{2(n_j/2-1)}=\frac{1}{2}.
$$

Overall the first term of (4.250) verifies

$$\mathbb{E}\left[(X'_{(n_j-1)}-X'_{(1)})\mathbb{E}\left[\frac{X'_{(n_j/2)}-X'_{(1)}}{X'_{(n_j-1)}-X'_{(1)}}\Big|X_1,\boldsymbol{\delta}_\ell,X'_{(1)},X'_{(n_j-1)}\right]\Big|X_1,\boldsymbol{\delta}_\ell(X_1,\Theta)\right]$$

$$=\mathbb{E}\left[\frac{X'_{(n_j-1)}-X'_{(1)}}{2}\Big|X_1,\boldsymbol{\delta}_\ell(X_1,\Theta)\right] \tag{4.251}$$

$$\leqslant\mathbb{E}\left[\frac{M_{2,j}-M_{1,j}}{2}\Big|X_1,\boldsymbol{\delta}_\ell(X_1,\Theta)\right]. \tag{4.252}$$

Regarding the second term of (4.250), we have

$$\mathbb{E}\left[\left|\frac{X'_{(n_j-1)}+X'_{(1)}}{2}-X'_{(n_j/2)}\right|\Big|X_1,\boldsymbol{\delta}_\ell,X'_{(1)},X'_{(n_j-1)}\right]$$

$$=\mathbb{E}\left[\left|\frac{X'_{(n_j-1)}-X'_{(1)}}{2}-\left(X'_{(n_j/2)}-X'_{(1)}\right)\right|\Big|X_1,\boldsymbol{\delta}_\ell,X'_{(1)},X'_{(n_j-1)}\right] \tag{4.253}$$

$$=\left(X'_{(n_j-1)}-X'_{(1)}\right)\mathbb{E}\left[\left|\frac{1}{2}-\frac{X'_{(n_j/2)}-X'_{(1)}}{X'_{(n_j-1)}-X'_{(1)}}\right|\Big|X_1,\boldsymbol{\delta}_\ell,X'_{(1)},X'_{(n_j-1)}\right] \tag{4.254}$$

$$=(X'_{(n_j-1)}-X'_{(1)})\mathbb{E}\left[\left|\frac{1}{2}-\mathcal{B}(n_j/2-1,n_j/2-1)\right|\right] \tag{4.255}$$

$$\leqslant(X'_{(n_j-1)}-X'_{(1)})\sqrt{\mathbb{E}\left[\left|\mathcal{B}(n_j/2-1,n_j/2-1)-\frac{1}{2}\right|^2\right]} \tag{4.256}$$

$$\leqslant\frac{M_{2,j}-M_{1,j}}{2\sqrt{n_j-1}}, \tag{4.257}$$

where the last inequality is simply obtained by computing the variance of a Beta distribution.

Therefore,

$$2\mathbb{E}\left[\mathbb{E}\left[\left|\frac{X'_{(n_j-1)}+X'_{(1)}}{2}-X'_{(n_j/2)}\right|\Big|X_1,\boldsymbol{\delta}_\ell(X_1,\Theta),X'_{(1)},X'_{(n_j-1)}\right]\Big|X_1,\boldsymbol{\delta}_\ell(X_1,\Theta)\right]$$

$$\leqslant\frac{1}{2\sqrt{n_j-1}}\mathbb{E}\left[M_{2,j}-M_{1,j}\Big|X_1,\boldsymbol{\delta}_\ell(X_1,\Theta)\right]. \tag{4.258}$$

The third and fourth terms of (4.250) have the same expression, controlled by Lemma S8:

$$\mathbb{E}\left[X'_{(1)}-M_{1,j}\Big|X_1,\boldsymbol{\delta}_\ell(X_1,\Theta)\right]=\mathbb{E}\left[M_{2,j}-X'_{(n_j-1)}\Big|X_1,\boldsymbol{\delta}_\ell(X_1,\Theta)\right] \tag{4.259}$$

$$\leqslant\frac{1}{n_j}\mathbb{E}\left[M_{2,j}-M_{1,j}\big|X_1,\boldsymbol{\delta}_\ell(X_1,\Theta)\right]. \tag{4.260}$$

Finally, gathering (4.252), (4.258) and (4.260) yields

$$\mathbb{E}\left[\mu\left(A^{(\ell)}_{n,j+1}(X_1,\Theta)\right)\middle|X_1,\boldsymbol{\delta}_\ell(X_1,\Theta)\right] \leqslant \mathbb{E}\left[M_{2,j}-M_{1,j}\middle|X_1,\boldsymbol{\delta}_\ell(X_1,\Theta)\right]\left(\frac{1}{2}+\frac{1}{2\sqrt{n_j-1}}+\frac{2}{n_j}\right) \tag{4.261}$$

$$\leqslant \frac{1}{2}\left(1+\frac{5}{\sqrt{n_j-1}}\right)\mathbb{E}\left[M_{2,j}-M_{1,j}\middle|X_1,\boldsymbol{\delta}_\ell(X_1,\Theta)\right] \tag{4.262}$$

$$= \frac{1}{2}\left(1+\frac{5}{\sqrt{n_j-1}}\right)\mathbb{E}\left[\mu\left(A^{(\ell)}_{n,j}(X_1,\Theta)\right)\middle|X_1,\boldsymbol{\delta}_\ell(X_1,\Theta)\right] \tag{4.263}$$

for all $n_j \geqslant 4$. An iterative product yields

$$\mathbb{E}\left[\mu\left(A^{(\ell)}_{n,k}(X_1,\Theta)\right)\middle|X_1,\boldsymbol{\delta}_\ell(X_1,\Theta)\right] \leqslant \mathbb{E}\left[\prod_{\substack{j:\delta_{j,\ell}=1,\\ j\leqslant k-2}}\frac{1}{2}\left(1+\frac{5}{\sqrt{n_j-1}}\right)\middle|X_1,\boldsymbol{\delta}_\ell(X_1,\Theta)\right] \tag{4.264}$$

$$= \prod_{\substack{j:\delta_{j,\ell}=1,\\ j\leqslant k-2}}\frac{1}{2}\left(1+\frac{5}{\sqrt{n_j-1}}\right) \tag{4.265}$$

$$= 2^{-K_\ell+2}\prod_{\substack{j:\delta_{j,\ell}=1,\\ j\leqslant k-2}}\left(1+\frac{5}{\sqrt{n_j-1}}\right), \tag{4.266}$$

which proves the first statement. Recalling that $n_j = n2^{-j}$,

$$\sum_{j=0}^{k}\log\left(1+\frac{5\sqrt{4/3}}{\sqrt{n}}2^{j/2}\right) \leqslant \frac{5\sqrt{4/3}}{\sqrt{n}}\frac{2^{(k+1)/2}-1}{\sqrt{2}-1} \tag{4.267}$$

$$\leqslant \frac{5\sqrt{4/3}}{\sqrt{2}-1}\frac{2^{(\log_2 n)/2}}{\sqrt{n}} \tag{4.268}$$

$$= \frac{5\sqrt{4/3}}{\sqrt{2}-1}, \tag{4.269}$$

we have

$$\mathbb{E}\left[\mu\left(A^{(\ell)}_{n,k}(X_1,\Theta)\right)\middle|X_1,\boldsymbol{\delta}_\ell(X_1,\Theta)\right] \leqslant 2^{-K_\ell+2}\exp\left(\frac{5\sqrt{4/3}}{\sqrt{2}-1}\right), \tag{4.270}$$

which proves the second statement. Note that

$$\mathbb{E}\left[\mu\left(A_{n,k}(X_1,\Theta)\right)\big|X_1,\boldsymbol{\delta}_1(X_1,\Theta),\ldots,\boldsymbol{\delta}_d(X_1,\Theta)\right] \tag{4.271}$$

$$=\mathbb{E}\left[\prod_{\ell=1}^{d}\mu\left(A_{n,k}^{(\ell)}(X_1,\Theta)\right)\big|X_1,\boldsymbol{\delta}_1(X_1,\Theta),\ldots,\boldsymbol{\delta}_d(X_1,\Theta)\right] \tag{4.272}$$

$$=\prod_{\ell=1}^{d}\mathbb{E}\left[\mu\left(A_{n,k}^{(\ell)}(X_1,\Theta)\right)\big|X_1,\boldsymbol{\delta}_\ell(X_1,\Theta)\right] \tag{4.273}$$

$$\leqslant\prod_{\ell=1}^{d}\prod_{\substack{j:\delta_{j,\ell}=1,\\ j\leqslant k-2}}\frac{1}{2}\left(1+\frac{5}{\sqrt{n_j-1}}\right) \tag{4.274}$$

$$\leqslant\prod_{j\leqslant k-2}\frac{1}{2}\left(1+\frac{5}{\sqrt{n_j-1}}\right) \tag{4.275}$$

$$\leqslant 4\times 2^{-k}\prod_{j\leqslant k-2}\left(1+\frac{5}{\sqrt{n_j-1}}\right) \tag{4.276}$$

$$\leqslant 4\times 2^{-k}\exp\left(\frac{5\sqrt{4/3}}{\sqrt{2}-1}\right). \tag{4.277}$$

□

S2.5 Proof of the Main Result (Median RF Consistency)

Theorem S10 (Upper bound on the risk of the median forest)**.** *Consider a generic pair (X,Y) of random variables such that $Y=f^\star(X)+\varepsilon$, where $||\partial_\ell f^\star||_\infty^2$ exists for all $\ell\in\{1,\ldots,d\}$, X is uniformly distributed on $[0,1]^d$ and the noise ε satisfies, almost surely, $\mathbb{E}[\varepsilon|X]=0$ and $\mathbb{V}[\varepsilon|X]\leqslant\sigma^2$. Consider $n\geqslant 16$ i.i.d. observations, where n is a power of two, distributed as the generic pair (X,Y). Then, the risk of the infinite median forest trained on this data set satisfies*

$$\mathbb{E}\left[\left(f_{\infty,n}^{\mathrm{MedRF}}(X)-f^\star(X)\right)^2\right]\leqslant C_1 d\left(\sum_{\ell=1}^{d}||\partial_\ell f^\star||_\infty^2\right)\left(1-\frac{3}{4d}\right)^{\log_2 n}+\sigma^2 C_{2,d}(\log_2 n)^{-(d-1)/2}, \tag{4.278}$$

with

$$C_1=1024\exp\left(\frac{42+\sqrt{5}}{2-\sqrt{2}}\right)\quad\textit{and}\quad C_{2,d}=2\left(32\ \exp\left(\frac{5\sqrt{4/3}}{\sqrt{2}-1}\right)\right)^d d^{d/2}. \tag{4.279}$$

In particular, the infinite median forest is consistent, that is

$$\lim_{n\to\infty}\mathbb{E}\left[\left(f_{\infty,n}^{\mathrm{MedRF}}(X)-f^\star(X)\right)^2\right]=0. \tag{4.280}$$

Proof. We begin with a simple bias/variance decomposition:

$$\begin{aligned}
&\mathbb{E}\left[\left(f_{\infty,n}^{\mathrm{MedRF}}(X) - f^\star(X)\right)^2\right] \\
&= \mathbb{E}\left[\left(\mathbb{E}_\Theta\left[\sum_{i=1}^n W_{ni}(X,\Theta) Y_i\right] - f^\star(X)\right)^2\right] \\
&= \mathbb{E}\left[\left(\sum_{i=1}^n \mathbb{E}_\Theta\left[W_{ni}(X,\Theta)\right]\left(f^\star(X_i) + \varepsilon_i\right) - f^\star(X)\right)^2\right] \\
&= \mathbb{E}\left[\left(\sum_{i=1}^n \mathbb{E}_\Theta\left[W_{ni}(X,\Theta)\right]\left(f^\star(X_i) - f^\star(X)\right) + \sum_{i=1}^n \mathbb{E}_\Theta\left[W_{ni}(X,\Theta)\right]\varepsilon_i\right)^2\right] \\
&= \mathbb{E}\left[\left(\sum_{i=1}^n \mathbb{E}_\Theta\left[W_{ni}(X,\Theta)\right]\left(f^\star(X_i) - f^\star(X)\right)\right)^2\right] + \mathbb{E}\left[\left(\sum_{i=1}^n \mathbb{E}_\Theta\left[W_{ni}(X,\Theta)\right]\varepsilon_i\right)^2\right],
\end{aligned}$$

where the penultimate line comes from the fact that

$$\sum_{i=1}^n \mathbb{E}_\Theta\left[W_{ni}(X,\Theta)\right] = \mathbb{E}_\Theta\left[\sum_{i=1}^n W_{ni}(X,\Theta)\right] = 1, \tag{4.281}$$

(since all leaves contain exactly one observation), and the last line results from a null cross product.

Controlling the bias We have,

$$\begin{aligned}
&\mathbb{E}\left[\left(\sum_{i=1}^n \mathbb{E}_\Theta\left[W_{ni}(X,\Theta)\right]\left(f^\star(X_i) - f^\star(X)\right)\right)^2\right] \\
&= \mathbb{E}\left[\left(\mathbb{E}_\Theta\left[\sum_{i=1}^n W_{ni}(X,\Theta)(f^\star(X_i) - f^\star(X))\right]\right)^2\right] && (4.282) \\
&\leqslant \mathbb{E}\left[\left(\sum_{i=1}^n W_{ni}(X,\Theta)(f^\star(X_i) - f^\star(X))\right)^2\right] && (4.283) \\
&\leqslant \mathbb{E}\left[\left(\sum_{i=1}^n \mathbb{1}_{X\in A_n(X_i,\Theta)}(f^\star(X_i) - f^\star(X))\right)^2\right] && (4.284) \\
&\leqslant \mathbb{E}\left[\sum_{i=1}^n \mathbb{1}_{X\in A_n(X_i,\Theta)}\left(f^\star(X_i) - f^\star(X)\right)^2\right], && (4.285)
\end{aligned}$$

because $W_{ni}(X,\Theta) = \mathbb{1}_{X \in A_n(X_i,\Theta)}$ and by applying twice Jensen inequality (third and fifth lines). Noticing that,

$$\begin{aligned}\mathbb{1}_{X \in A_n(X_i,\Theta)} |f^\star(X) - f^\star(X_i)| &\leqslant \sum_{\ell=1}^{d} ||\partial_\ell f^\star||_\infty |X_i^{(\ell)} - X^{(\ell)}| \mathbb{1}_{X \in A_n(X_i,\Theta)} \\ &\leqslant \sum_{\ell=1}^{d} ||\partial_\ell f^\star||_\infty \mu(A_n^{(\ell)}(X,\Theta)) \mathbb{1}_{X \in A_n(X_i,\Theta)},\end{aligned}$$

we get,

$$\begin{aligned}\mathbb{E}\left[\sum_{i=1}^{n} \mathbb{1}_{X \in A_n(X_i,\Theta)} \left(f^\star(X_i) - f^\star(X)\right)^2\right] &\leqslant \mathbb{E}\left[\sum_{i=1}^{n} \mathbb{1}_{X \in A_n(X_i,\Theta)} \left(\sum_{\ell=1}^{d} ||\partial_\ell f^\star||_\infty \mu(A_n^{(\ell)}(X,\Theta))\right)^2\right] \\ &\leqslant \mathbb{E}\left[\left(\sum_{\ell=1}^{d} ||\partial_\ell f^\star||_\infty \mu(A_n^{(\ell)}(X,\Theta))\right)^2\right] \qquad (4.286) \\ &\leqslant \left(\sum_{\ell=1}^{d} ||\partial_\ell f^\star||_\infty^2\right) \sum_{\ell=1}^{d} \mathbb{E}\left[\mu(A_n^{(\ell)}(X,\Theta))^2\right]. \qquad (4.287)\end{aligned}$$

where the last inequality directly results from Cauchy-Schwarz inequality. By Lemma S7, since $k = \lfloor \log_2 n \rfloor$,

$$\left(\sum_{\ell=1}^{d} ||\partial_\ell f^\star||_\infty^2\right) \sum_{\ell=1}^{d} \mathbb{E}\left[\mu(A_n^{(\ell)}(X,\Theta))^2\right] \leqslant Cd \left(\sum_{\ell=1}^{d} ||\partial_\ell f^\star||_\infty^2\right) \left(1 - \frac{3}{4d}\right)^{\log_2 n}, \qquad (4.288)$$

with

$$C = 1024 \exp\left(\frac{42 + \sqrt{5}}{2 - \sqrt{2}}\right). \qquad (4.289)$$

Controlling the variance Following Biau 2012b, the variance term of the median forest writes

$$\begin{aligned}\mathbb{E}\left[\left(\sum_{i=1}^{n} \mathbb{E}_\Theta\left[W_{ni}(X,\Theta)\right] \varepsilon_i\right)^2\right] &= \mathbb{E}\left[\sum_{i=1}^{n} \left(\mathbb{E}_\Theta\left[W_{ni}(X,\Theta)\right]\right)^2 \varepsilon_i^2\right] \qquad (4.290) \\ &= \mathbb{E}\left[\sum_{i=1}^{n} \left(\mathbb{E}_\Theta\left[W_{ni}(X,\Theta)\right]\right)^2 \mathbb{E}\left[\varepsilon_i^2 | X, X_1, \ldots, X_n\right]\right] \qquad (4.291) \\ &\leqslant \mathbb{E}\left[\sum_{i=1}^{n} \left(\mathbb{E}_\Theta\left[W_{ni}(X,\Theta)\right]\right)^2 \sigma^2\right] \qquad (4.292) \\ &\leqslant \sigma^2 n \mathbb{E}\left[\left(\mathbb{E}_\Theta\left[W_{n1}(X,\Theta)\right]\right)^2\right], \qquad (4.293)\end{aligned}$$

where we have used the fact that the cross products are null (since $\mathbb{E}[\varepsilon_i|X_i] = 0$). Since each leaf of the median tree contains exactly one observation, denoting Θ' an i.i.d. copy of Θ, we have

$$\begin{aligned}\left(\mathbb{E}_\Theta\left[W_{n1}(X,\Theta)\right]\right)^2 &= \mathbb{E}_\Theta\left[W_{n1}(X,\Theta)\right]\mathbb{E}_{\Theta'}\left[W_{n1}(X,\Theta')\right]\\ &= \mathbb{E}_{\Theta,\Theta'}\left[W_{n1}(X,\Theta)W_{n1}(X,\Theta')\right]\\ &= \mathbb{E}_{\Theta,\Theta'}\left[\mathbb{1}_{X\in A_n(X_1,\Theta)}\mathbb{1}_{X\in A_n(X_1,\Theta')}\right].\end{aligned}$$

Consequently,

$$\mathbb{E}\left[\left(\sum_{i=1}^n \mathbb{E}_\Theta\left[W_{ni}(X,\Theta)\right]\varepsilon_i\right)^2\right] \leqslant \sigma^2 n\mathbb{E}\left[\mathbb{1}_{X\in A_n(X_1,\Theta)}\mathbb{1}_{X\in A_n(X_1,\Theta')}\right].$$

For all ℓ, we let $A_n^{(\ell)}(X_1,\Theta)$ be the cell $A_n(X_1,\Theta)$ projected onto the ℓ-th dimension. Let also $\boldsymbol{\delta}_\ell(X_1,\Theta)$ be the vector whose components are defined as $\delta_{j,l} = 1$ if the j-th cut of the cell $A_n^{(\ell)}(X_1,\Theta)$ is made along direction ℓ and 0 otherwise. We define similarly $\boldsymbol{\delta}_\ell(X_1,\Theta')$ for the cell $A_n^{(\ell)}(X_1,\Theta')$. We also let $K_\ell = \|\boldsymbol{\delta}_\ell(X_1,\Theta)\|_1$ (resp. K'_ℓ) be the number of times the ℓ-th direction is split in the tree built with Θ (resp. Θ'). Then,

$$\begin{aligned}&\mathbb{E}\left[\left(\sum_{i=1}^n \mathbb{E}_\Theta\left[W_{ni}(X,\Theta)\right]\varepsilon_i\right)^2\right]\\
&\leqslant \sigma^2 n\mathbb{E}\left[\mathbb{1}_{X\in A_n(X_1,\Theta)\cap A_n(X_1,\Theta')}\right]\\
&= \sigma^2 n\mathbb{E}\left[\prod_{\ell=1}^d \mu\left(A_n^{(\ell)}(X_1,\Theta)\cap A_n^{(\ell)}(X_1,\Theta')\right)\right]\\
&= \sigma^2 n\mathbb{E}\left[\mathbb{E}\left[\prod_{\ell=1}^d \mu\left(A_n^{(\ell)}(X_1,\Theta)\cap A_n^{(\ell)}(X_1,\Theta')\right)\middle| X_1, \boldsymbol{\delta}_\ell(X_1,\Theta), \boldsymbol{\delta}_1(X_1,\Theta'),\ldots,\boldsymbol{\delta}_d(X_1,\Theta')\right]\right]\\
&= \sigma^2 n\mathbb{E}\left[\prod_{\ell=1}^d \mathbb{E}\left[\mu\left(A_n^{(\ell)}(X_1,\Theta)\cap A_n^{(\ell)}(X_1,\Theta')\right)\middle| X_1, \boldsymbol{\delta}_\ell(X_1,\Theta), \boldsymbol{\delta}_1(X_1,\Theta'),\ldots,\boldsymbol{\delta}_d(X_1,\Theta')\right]\right]\\
&= \sigma^2 n\mathbb{E}\left[\prod_{\ell=1}^d \mathbb{E}\left[\mu\left(A_n^{(\ell)}(X_1,\Theta)\cap A_n^{(\ell)}(X_1,\Theta')\right)\middle| X_1, \boldsymbol{\delta}_\ell(X_1,\Theta), \boldsymbol{\delta}_\ell(X_1,\Theta')\right]\right].\end{aligned}$$

The penultimate equality is obtained by conditional independence. Indeed, as the data are uniformly distributed, each coordinate follows a uniform distribution independently from the other coordinates. Thus, in a given cell, cutting along another direction preserves the distribution of the data points along direction ℓ (it is akin to removing half of them uniformly). Consequently, when knowing the number of cuts on each direction, having information on the length of the cell along a direction does not provide information on the distribution of the data on other directions. The last equality is deduced from the same argument. All the cuts on other direction than ℓ have the same influence along direction ℓ: they remove half the data points but preserve the distribution of the remaining ones. Therefore, conditional on $\boldsymbol{\delta}_\ell$, the length of the cell along

direction ℓ is independent from $\boldsymbol{\delta}_{\ell'}$ for all $\ell' \neq \ell$ Now,

$$\begin{aligned}
&\mathbb{E}\left[\mu\left(A_n^{(\ell)}(X_1,\Theta)\cap A_n^{(\ell)}(X_1,\Theta')\right)\middle| X_1, \boldsymbol{\delta}_\ell(X_1,\Theta), \boldsymbol{\delta}_\ell(X_1,\Theta')\right] \\
&\leqslant \mathbb{E}\left[\min(\mu(A_n^{(\ell)}(X_1,\Theta)), \mu(A_n^{(\ell)}(X_1,\Theta'))\middle| X_1, \boldsymbol{\delta}_\ell(X_1,\Theta), \boldsymbol{\delta}_\ell(X_1,\Theta')\right] \\
&= \frac{1}{2}\left(\mathbb{E}\left[\mu(A_n^{(\ell)}(X_1,\Theta))\middle| X_1, \boldsymbol{\delta}_\ell(X_1,\Theta)\right] + \mathbb{E}\left[\mu(A_n^{(\ell)}(X_1,\Theta'))\middle| X_1, \boldsymbol{\delta}_\ell(X_1,\Theta')\right]\right) \\
&\quad - \frac{1}{2}\mathbb{E}\left[|\mu(A_n^{(\ell)}(X_1,\Theta)) - \mu(A_n^{(\ell)}(X_1,\Theta'))|\middle| X_1, \boldsymbol{\delta}_\ell(X_1,\Theta), \boldsymbol{\delta}_\ell(X_1,\Theta')\right].
\end{aligned}$$

Moreover,

$$\begin{aligned}
&\mathbb{E}\left[|\mu(A_n^{(\ell)}(X_1,\Theta)) - \mu(A_n'^{(\ell)}(X_1,\Theta))|\middle| X_1, \boldsymbol{\delta}_\ell(X_1,\Theta), \boldsymbol{\delta}_\ell(X_1,\Theta')\right] \\
&= \mathbb{E}\left[|\mu(A_n^{(\ell)}(X_1,\Theta)) - \mu(A_n^{(\ell)}(X_1,\Theta'))|\left(\mathbb{1}_{K_\ell<K'_\ell} + \mathbb{1}_{K_\ell\geqslant K'_\ell}\right)\middle| X_1, \boldsymbol{\delta}_\ell(X_1,\Theta), \boldsymbol{\delta}_\ell(X_1,\Theta')\right] \\
&\geqslant \mathbb{E}\left[\left(\mu(A_n^{(\ell)}(X_1,\Theta)) - \mu(A_n^{(\ell)}(X_1,\Theta'))\right)\middle| X_1, \boldsymbol{\delta}_\ell(X_1,\Theta), \boldsymbol{\delta}_\ell(X_1,\Theta')\right]\mathbb{1}_{K_\ell<K'_\ell} \\
&\quad + \mathbb{E}\left[\left(\mu(A_n^{(\ell)}(X_1,\Theta')) - \mu(A_n^{(\ell)}(X_1,\Theta))\right)\middle| X_1, \boldsymbol{\delta}_\ell(X_1,\Theta), \boldsymbol{\delta}_\ell(X_1,\Theta')\right]\mathbb{1}_{K_\ell\geqslant K'_\ell} \\
&\geqslant \left(\mathbb{E}\left[\mu(A_n^{(\ell)}(X_1,\Theta))|X_1, \boldsymbol{\delta}_\ell(X_1,\Theta)\right] - \mathbb{E}\left[\mu(A_n^{(\ell)}(X_1,\Theta'))|X_1, \boldsymbol{\delta}_\ell(X_1,\Theta')\right]\right)\mathbb{1}_{K_\ell<K'_\ell} \\
&\quad + \left(\mathbb{E}\left[\mu(A_n^{(\ell)}(X_1,\Theta'))|X_1, \boldsymbol{\delta}_\ell(X_1,\Theta')\right] - \mathbb{E}\left[\mu(A_n^{(\ell)}(X_1,\Theta))|X_1, \boldsymbol{\delta}_\ell(X_1,\Theta)\right]\right)\mathbb{1}_{K_\ell\geqslant K'_\ell}.
\end{aligned}$$

Letting $B_\ell = \mathbb{E}\left[\mu(A_n^{(\ell)}(X_1,\Theta))|X_1, \boldsymbol{\delta}_\ell(X_1,\Theta)\right]$ and $B'_\ell = \mathbb{E}\left[\mu(A_n^{(\ell)}(X_1,\Theta'))|X_1, \boldsymbol{\delta}_\ell(X_1,\Theta')\right]$, we have

$$\begin{aligned}
&\mathbb{E}\left[\mu\left(A_n^{(\ell)}(X_1,\Theta)\cap A_n^{(\ell)}(X_1,\Theta')\right)\middle| X_1, \boldsymbol{\delta}_\ell(X_1,\Theta), \boldsymbol{\delta}_\ell(X_1,\Theta')\right] \\
&\leqslant \frac{1}{2}(B_\ell + B'_\ell) - \frac{1}{2}(B_\ell - B'_\ell)\mathbb{1}_{K_\ell<K'_\ell} - \frac{1}{2}(B'_\ell - B_\ell)\mathbb{1}_{K_\ell\geqslant K'_\ell} \\
&\leqslant B_\ell\mathbb{1}_{K_\ell\geqslant K'_\ell} + B'_\ell\mathbb{1}_{K_\ell<K'_\ell}.
\end{aligned}$$

Now, according to Lemma S9, letting $C_2 = 4\exp(5/(\sqrt{2}-1))$, we have $B_\ell \leqslant C_2 2^{-K_\ell}$ and $B'_\ell \leqslant C_2 2^{-K'_\ell}$. Therefore,

$$\begin{aligned}
\mathbb{E}\left[\mu\left(A_n^{(\ell)}(X_1,\Theta)\cap A_n^{(\ell)}(X_1,\Theta')\right)\middle| X_1, \boldsymbol{\delta}_\ell(X_1,\Theta), \boldsymbol{\delta}_\ell(X_1,\Theta')\right] &\leqslant C_2 2^{-K_\ell}\mathbb{1}_{K_\ell\geqslant K'_\ell} + C_2 2^{-K'_\ell}\mathbb{1}_{K_\ell<K'_\ell} \\
&\leqslant C_2 2^{-\max(K_\ell,K'_\ell)}.
\end{aligned}$$

Overall,

$$\mathbb{E}\left[\left(\sum_{i=1}^{n}\mathbb{E}_\Theta\left[W_{ni}(X,\Theta)\right]\varepsilon_i\right)^2\right]$$

$$\leqslant \sigma^2 n\mathbb{E}\left[\prod_{\ell=1}^{d}\mathbb{E}\left[\mu\left(A_n^{(\ell)}(X_1,\Theta)\cap A_n^{(\ell)}(X_1,\Theta')\right)\middle| X_1,\boldsymbol{\delta}_\ell(X_1,\Theta),\boldsymbol{\delta}_\ell(X_1,\Theta')\right]\right] \tag{4.294}$$

$$\leqslant \sigma^2 n\mathbb{E}\left[\prod_{\ell=1}^{d}C_2 2^{-\max(K_\ell,K'_\ell)}\right] \tag{4.295}$$

$$\leqslant \sigma^2 C_2^d n\ \mathbb{E}\left[2^{-\sum_{\ell=1}^{d}\max(K_\ell,K'_\ell)}\right] \tag{4.296}$$

$$\leqslant \sigma^2 C_2^d n\ \ 2^{-k_n}\mathbb{E}\left[2^{-\sum_{\ell=1}^{d}|K_\ell-K'_\ell|}\right], \tag{4.297}$$

since

$$\begin{aligned}\sum_{\ell=1}^{d}\max(K_\ell,K'_\ell) &= \frac{1}{2}\sum_{\ell=1}^{d}K_\ell+\frac{1}{2}\sum_{\ell=1}^{d}K'_\ell+\frac{1}{2}\sum_{\ell=1}^{d}|K_\ell-K'_\ell|\\ &= k_n+\frac{1}{2}\sum_{\ell=1}^{d}|K_\ell-K'_\ell|.\end{aligned}$$

According to Lemma S.1 from Klusowski 2021a (see Supplementary Materials), one has

$$\mathbb{E}\left[2^{-\sum_{\ell=1}^{d}|K_\ell-K'_\ell|}\right]\leqslant\frac{8^d\,d^{d/2}}{k_n^{(d-1)/2}}. \tag{4.298}$$

Finally, combining (4.297) and (4.298), the variance of the median forest is upper bounded by

$$\mathbb{E}\left[\left(\sum_{i=1}^{n}\mathbb{E}_\Theta\left[W_{ni}(X,\Theta)\right]\varepsilon_i\right)^2\right]\leqslant\sigma^2C_2^d n\ 2^{-k_n}\frac{8^d\,d^{d/2}}{k_n^{(d-1)/2}} \tag{4.299}$$

$$\leqslant 2\sigma^2\left(8\ C_2\ d^{1/2}\right)^d(\log_2 n)^{-(d-1)/2}, \tag{4.300}$$

since $k_n=\lfloor\log_2(n)\rfloor$. All in all,

$$\begin{aligned}&\mathbb{E}\left[\left(f_{\infty,n}^{\mathrm{MedRF}}(X)-f^\star(X)\right)^2\right]\\ &\leqslant Cd\left(\sum_{\ell=1}^{d}||\partial_\ell f^\star||_\infty^2\right)\left(1-\frac{3}{4d}\right)^{\log_2 n}+2\sigma^2\left(8\ C_2\ d^{1/2}\right)^d(\log_2 n)^{-(d-1)/2}\end{aligned}$$

with $C_2=4\exp(5/(\sqrt{2}-1))$ and

$$C=1024\exp\left(\frac{42+\sqrt{5}}{2-\sqrt{2}}\right). \tag{4.301}$$

□

Controlling the Variance of an Interpolating Median RF in an Asymptotic High-Dimensional Setting

The following result shows the decrease of the variance of the Median RF under an asymptotic high-dimensional framework. It is also numerically illustrated in Section S3.1.

Proposition S11. *For all $d > \log_2 n$, the variance of the infinite interpolating Median RF $f_{\infty,n}^{\mathrm{MedRF}}$ verifies*

$$V(f_{\infty,n}^{\mathrm{MedRF}}) = \mathbb{E}\left[\left(\sum_{i=1}^{n} \mathbb{E}_\Theta\left[W_{ni}(X,\Theta)\right]\varepsilon_i\right)^2\right] \leqslant \frac{4C_2^2\sigma^2}{n} + 2C_2\sigma^2\left(1-\exp\left(-\frac{\log_2^2 n}{d-\log_2 n}\right)\right),$$

where $C_2 = 4\exp\left(5/(\sqrt{2}-1)\right)$. Suppose that the input dimension d dominates $\log_2^2 n$ asymptotically ($d \gg \log_2^2 n$), then the variance tends to 0 (as n, d tends to infinity), with a rate of the order of $\max(\frac{\log^2 n}{d}, \frac{1}{n})$.

The proof is given below. This results shows that the Median RF benefits from an increase of the dimension as it will improve its averaging effect and help to reduce the variance. Of course, in such a setting, the variance is only one part of the story, and a control on the bias becomes a real hindrance (as the approximation error may explode), unless extra model assumptions are formulated. For instance, consider for any input dimension d the case of a linear model, i.e. $Y = X^\top\theta + \varepsilon$ for $\theta \in \mathbb{R}^d$ and such that $\|\theta\|_2 \leqslant C/\sqrt{d}$, with $C > 0$ a constant.

One can actually show that in such a setting, the bias term remains bounded as n (and d) grows towards infinity (using for example the analysis conducted in the next theorem). This echoes in particular the behavior of ridgeless least squares estimator in modern interpolation regimes (see (Hastie et al. 2022)).

Proof of Proposition S11. A typical bias-variance decomposition yields (see e.g. Biau 2012b)

$$V(f_{\infty,n}^{\mathrm{MedRF}}) \leqslant \sigma^2 n\mathbb{P}\left(X \in A_n(X_1,\Theta) \cap A_n(X_1,\Theta')\right) \tag{4.302}$$

with Θ' an independent copy of Θ. Recalling that the depth is chosen as $k = \lfloor \log_2 n \rfloor$. Consider the event

$$E = E(\Theta,\Theta',X_1,k) := \{\Theta \text{ and } \Theta' \text{ do not cut on common directions on the path to } X_1\}.$$

Denote $M(\Theta, X_1)$ the number of distinct directions chosen by the tree Θ to produce the leaf containing X_1 (upper bounded by $\log_2 n$). Then,

$$\begin{aligned}
\mathbb{P}(E) &\geqslant \mathbb{E}\left[\left(\frac{d-M(\Theta,X_1)}{d}\right)^{\log_2 n}\right] && (4.303)\\
&\geqslant \left(\frac{d-\log_2 n}{d}\right)^{\log_2 n} && (4.304)\\
&= \exp\left(\log_2 n \log\left(1-\frac{\log_2 n}{d}\right)\right) && (4.305)\\
&\geqslant \exp\left(-\frac{\log_2^2 n}{d-\log_2 n}\right), && (4.306)
\end{aligned}$$

using, for all $x \in [0,1)$, $\log(1-x) \geqslant -x/(1-x)$. The above probability tends to 1 as soon as

$d \gg \log_2^2 n$. Then,

$$\begin{aligned}
&\mathbb{P}\left(X \in A_n(X_1,\Theta) \cap A_n(X_1,\Theta')\right) && (4.307)\\
&= \mathbb{P}\left(\{X \in A_n(X_1,\Theta) \cap A_n(X_1,\Theta')\} \cap E\right) + \mathbb{P}\left(\{X \in A_n(X_1,\Theta) \cap A_n(X_1,\Theta')\} \cap E^c\right)\\
&\leqslant \mathbb{P}\left(X \in A_n(X_1,\Theta) | X \in A_n(X_1,\Theta'), E\right) \mathbb{P}\left(X \in A_n(X_1,\Theta')\right) + \mathbb{P}\left(\{X \in A_n(X_1,\Theta)\} \cap E^c\right). && (4.308)
\end{aligned}$$

Applying Lemma S9 (Line (4.241)) yields

$$\begin{aligned}
\mathbb{P}\left(X \in A_n(X_1,\Theta')\right) &= \mathbb{E}\left[\mu\left(A_n(X_1,\Theta')\right)\right] && (4.309)\\
&= \mathbb{E}\left[\mathbb{E}\left[\mu(A_n(X_1,\Theta)) | X_1, \boldsymbol{\delta}_1(X_1,\Theta), \ldots, \boldsymbol{\delta}_d(X_1,\Theta)\right]\right] && (4.310)\\
&\leqslant C_2 2^{-k} && (4.311)
\end{aligned}$$

with $C_2 = 4\exp\left(\frac{5}{\sqrt{2}-1}\right)$. Moreover, conditional on E, $\{X \in A_n(X_1,\Theta)\}$ and $\{X \in A_n(X_1,\Theta')\}$ are independent as Θ and Θ' do not share any common direction on the path to X_1 and therefore the splits in Θ and Θ' are performed on independent sample components (by uniformity of X and X_1). Therefore, also by Lemma S9,

$$\begin{aligned}
\mathbb{P}\left(X \in A_n(X_1,\Theta) | X \in A_n(X_1,\Theta'), E\right) &= \mathbb{E}\left[\mu\left(A_n(X_1,\Theta)\right)\right] && (4.312)\\
&= \mathbb{E}\left[\mathbb{E}\left[\mu(A_n(X_1,\Theta)) | X_1, \boldsymbol{\delta}_1(X_1,\Theta), \ldots, \boldsymbol{\delta}_d(X_1,\Theta)\right]\right] && (4.313)\\
&\leqslant C_2 2^{-k}. && (4.314)
\end{aligned}$$

Similarly, the volume $\mu\left(A_n(X_1,\Theta)\right)$ is independent of the directions chosen to build the leaf, therefore

$$\begin{aligned}
\mathbb{P}\left(\{X \in A_n(X_1,\Theta)\} \cap E^c\right) &= \mathbb{P}\left(X \in A_n(X_1,\Theta)\right) \mathbb{P}\left(E^c\right)\\
&\leqslant C_2 2^{-k}\left(1 - \exp\left(-\frac{\log_2^2 n}{d - \log_2 n}\right)\right).
\end{aligned}$$

Overall,

$$\mathbb{P}\left(X \in A_n(X_1,\Theta) \cap A_n(X_1,\Theta')\right) \leqslant C_2^2 2^{-2k} + C_2 2^{-k}\left(1 - \exp\left(-\frac{\log_2^2 n}{d - \log_2 n}\right)\right)$$

and

$$V(f_{\infty,n}^{\mathrm{MedRF}}) \leqslant C_2^2 n\sigma^2 2^{-2k} + n\sigma^2 C_2\left(1 - \exp\left(-\frac{\log_2^2 n}{d - \log_2 n}\right)\right) 2^{-k}. \tag{4.315}$$

Since $k = \lfloor \log_2 n \rfloor$, we have $2^{-k} \leqslant 2/n$ and

$$V(f_{\infty,n}^{\mathrm{MedRF}}) \leqslant \frac{4C_2^2\sigma^2}{n} + 2C_2\sigma^2\left(1 - e^{-\frac{\log_2^2 n}{d - \log_2 n}}\right). \tag{4.316}$$

□

Proof of Proposition 4.5.3 (Interpolation Volume of Median RF)

It is possible to conduct a one-dimensional analysis and then to extend the result to the multi-dimensional case by a simple multiplication. Indeed all the leaves are determined coordinate per coordinate, therefore the interpolation area is the product of all interpolation areas along each direction.

Let $Z_1, \dots, Z_n$ be n i.i.d. random variables uniformly distributed over $[0, 1]$. As in the infinite Median RF, the univariate trees, i.e., built by cutting along one direction only, appear almost surely. Then, the length of a leaf of such tree is bounded in expectation by $Z_{(k+1)} - Z_{(k-1)}$ where $Z_{(i)}$ indicates the i-the statistical order. Moreover, it is known that $Z_{(k)}$ follows a Beta distribution of parameters $(k, n - k + 1)$. Therefore,

$$\mathbb{E}\left[Z_{(k+1)} - Z_{(k-1)}\right] = \frac{k+1}{n+1} - \frac{k-1}{n+1} \tag{4.317}$$

$$\leqslant \frac{2}{n}. \tag{4.318}$$

Now, as $X_1, ..., X_n$ are i.i.d. and uniformly distributed over $[0, 1]^d$, for any data point $x \in [0, 1]^d$ we simply have that

$$\mathbb{E}\left[\mu(\mathcal{A}_{min,x})\right] \leqslant \frac{2^d}{n^d}.$$

Finally, since by definition all interpolation zones are disjoint and the interpolation area is the union of n interpolation areas, we have

$$\mathbb{E}\left[\mu(\mathcal{A}_{min})\right] \leqslant \frac{2^d}{n^{d-1}}$$

which ends the proof.

S2.6 Proofs of Section 4.6 (Interpolation Volume of Breiman RF)

Proof of Proposition 4.6.1. Before diving into the computations, let us recall two facts about Breiman RF construction. First, in CART, each cut is made at the middle of two consecutive points in a given direction. Second, considering all univariate trees (trees whose splits are performed along one single direction), the probability of cutting between all pairs of successive points along all dimensions is strictly positive. Therefore, for a given point X_i, one can define the minimal interpolation zone around X_i as

$$\mathcal{A}_{min,X_i} := \bigcap_{M \in \mathbb{N}, \boldsymbol{\Theta}_M} \mathcal{A}_{X_i,\boldsymbol{\Theta}_M}. \tag{4.319}$$

The boundaries of this area are given for each direction by the cuts between X_i and its *neighbor points* respectively to the considered direction, as illustrated on Figure S6.

1. The interpolation zone is the union of n interpolation zones, each one containing a single X_i. We denote $\mathcal{A}(m_{M,n}(., \boldsymbol{\Theta}_M)) = \mathcal{A}_{X_1,\boldsymbol{\Theta}_M} \cup ... \cup \mathcal{A}_{X_n,\boldsymbol{\Theta}_M}$ with $\mathcal{A}_{X_i,\Theta_M} = \{x \in [0, 1]^d, m_{M,n}(x, \boldsymbol{\Theta}_M) = Y_i\}$. We begin with a one-dimensional analysis, and consider, without loss of generality, the first variable. We let $Z_1 := X_1^{(1)}, ..., Z_n := X_n^{(1)}$ the first components of the observations $X_1, \dots, X_n$. As $X_1, \dots, X_n$ are i.i.d. and follow a uniform

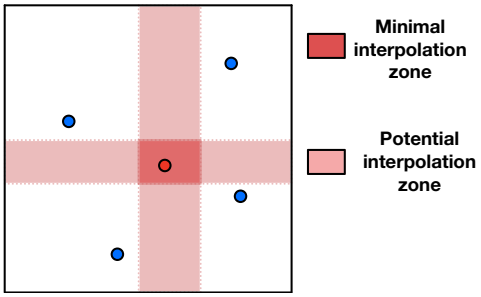


Figure S6: Different interpolation zones of a data point (in red).

distribution over $[0,1]^d$, $Z_1, ..., Z_n$ are i.i.d. and uniformly distributed on $[0,1]$. We consider the interpolation area at $x = Z_n$ and we reason conditional on Z_n in the following. The length (volume) of $\mathcal{A}_{min,x}$ restricted to the first dimension is simply given by the sum of the distance from x to its closest point on the left side and to its closest point on the right side (divided by 2 as the cut are made in the middle of two points). Therefore,

$$\mu(A_{min,x}) = \frac{1}{2}\left(x - \max_{\{Z_i, Z_i<x\}\cup\{0\}} Z_i + \min_{\{Z_i, Z_i>x\}\cup\{1\}} Z_i - x\right). \tag{4.320}$$

All computations are made conditionally on x. Denoting N_x the cardinal of the set $\{Z_i : Z_i < x \text{ with } 1 \leqslant i < n\}$, we have for any $t \in [0, x/2)$,

$$\begin{aligned}
&\mathbb{P}\left(\frac{1}{2}\left(x - \max_{\{Z_i, Z_i<x\}\cup\{0\}} Z_i\right) \leqslant t \,\middle|\, x\right) && (4.321)\\
&= 1 - \mathbb{P}\left(\max_{\{Z_i, Z_i<x\}\cup\{0\}} Z_i < x - 2t \,\middle|\, x\right) && (4.322)\\
&= 1 - \mathbb{E}\left[\mathbb{E}\left[\mathbb{P}\left((Z_{i_1} < x-2t) \cap ... \cap (Z_{i_{N_x}} < x-2t)\middle| N_x, Z_{i_1} < x, ..., Z_{i_{N_x}} < x, x\right)\right] \middle| x\right] && (4.323)\\
&= 1 - \mathbb{E}\left[\mathbb{P}\left(Z_1 < x - 2t | Z_1 \leqslant x, x\right)^{N_x} \middle| x\right] && (4.324)\\
&= 1 - \sum_{k=0}^{n-1} \mathbb{P}\left(N_x = k|x\right) \mathbb{P}\left(Z_1 < x - 2t | Z_1 < x, x\right)^k && (4.325)\\
&= 1 - \sum_{k=0}^{n-1} \mathbb{P}\left(N_x = k|x\right) \left(\frac{x-2t}{x}\right)^k && (4.326)\\
&= 1 - \left((1-x) + x\left(\frac{x-2t}{x}\right)\right)^{n-1} && (4.327)\\
&= 1 - (1-2t)^{n-1} && (4.328)
\end{aligned}$$

where the penultimate equality is obtained by noticing that N_x is a binomial of parameters $(n-1, x)$ and computing its probability-generating function. So for all $t \geqslant 0$,

$$\mathbb{P}\left(\frac{1}{2}\left(x - \max_{\{Z_i, Z_i<x\}\cup\{0\}} Z_i\right) \leqslant t | x\right) = 1 - (1-2t)^{n-1} \mathbb{1}_{t<x/2}.$$

By symmetry,

$$\mathbb{P}\left(\frac{1}{2}\left(\min_{\{Z_i, Z_i>x\}\cup\{1\}} Z_i - x\right) \leqslant t | x\right) = 1 - (1-2t)^{n-1}\mathbb{1}_{t>(1-x)/2}.$$

Overall, using the fact that for any positive variable Z with cumulative function F_Z, $\mathbb{E}[Z] = \int(1-F_Z)$, we have

$$\begin{aligned}\mathbb{E}[\mu(\mathcal{A}_{min,x})|x] &= \int_0^{x/2}(1-2u)^{n-1}du + \int_0^{(1-x)/2}(1-2u)^{n-1}du \\ &= \frac{1}{2n}\left(2-(1-x)^n - x^n\right) \\ &\leqslant \frac{1}{n}\left(1-\frac{1}{2^n}\right).\end{aligned}$$

Now, as $X_1, ..., X_n$ are i.i.d. and uniformly distributed over $[0,1]^d$, for any data point $x \in [0,1]^d$ we simply have that

$$\mathcal{A}_{min,x} = \bigtimes_{j=1}^{d} \mathcal{A}_{min,x^{(j)}}.$$

Therefore,

$$\mathbb{E}[\mu(\mathcal{A}_{min,x})] \leqslant \frac{1}{n^d}\left(1-2^{-n}\right)^d.$$

Finally, since by definition all interpolation zones are disjoint, we have

$$\mathbb{E}[\mu(\mathcal{A}_{min})] \leqslant \frac{1}{n^{d-1}}\left(1-2^{-n}\right)^d.$$

2. It is enough to notice that the minimal interpolation zone is the intersection of all the potential interpolation zones. It is reached when the forest contains all the possible cuts. Then, as the probability of any given cut appearing is strictly greater than 0 by hypothesis, the probability of its appearance in the infinite forest is one. Therefore almost surely, when M grows to infinity, the interpolation zone of the forest reaches the minimal interpolation zone.

□

S3 Experiments

For all experiments, we consider four different regression models, most of which have been already considered in Laan et al. 2007: Model 1 is additive without noise ($d = 2$), Model 2 is polynomial with interactions ($d = 8$), Model 3 is the sum of elementary terms that contain non-polynomial interactions ($d = 6$) and Model 4 ($d = 5$) corresponds to a generalized linear model:

- **Model 1:** $d = 2, Y = 2X_1^2 + \exp(-X_2^2)$
- **Model 2:** $d = 6, Y = X_1^2 + X_2^2 X_3 e^{-|X_4|} + X_5 - X_6 + \mathcal{N}(0, 0.5)$
- **Model 3:** $d = 8, Y = X_1 X_2 + X_3^2 - X_4 X_5 + X_6 X_7 - X_8^2 + \mathcal{N}(0, 0.5)$
- **Model 4:** $d = 5, Y = 1/(1 + \exp(-10(\sum_{i=1}^{d} X_i - 1/2))) + \mathcal{N}(0, 0.05)$
- **Model 5**: $d = 4, Y = -\sin(2X_1 X_2) + X_2^2 + X_3 - e^{X_4} + \mathcal{N}(0, 0.5)$
- **Model 6**: $d = 8, Y = \mathbb{1}_{\{X_1 \geqslant 0\}} + X_2^3 + \mathbb{1}_{\{X_3 + X_5 - X_6 - X_7 - X_8 \geqslant 1\}} + e^{-X_2^2} + \mathcal{N}(0, 0.5)$
- **Model 7**: $d = 4, Y = X_1 + 2(X_2 - 1)^2 + \frac{\sin(2\pi X_3)}{2 - \sin(2\pi X_3)} + 2\sin(2\pi X_4) + 2\cos(2\pi X_4) + 4\sin(2\pi X_4)^2 + 4\cos(2\pi X_4)^2 + \mathcal{N}(0, 0.5)$
- **Model 8**: $d = 4, Y = X_1 + 3X_2^2 - 2e^{X_3} + X_4$.

All the experiments are conducted using Python3. We use Scikit-learn RandomForestRegressor class to implement the Breiman RF model. We coded CRF, KeRF and AdaCRF models ourselves, mainly relying on *numpy* and *joblib* libraries for computation optimisation. Experiments were run on 4 16-cores CPU and took at most a few hours to run.

S3.1 Consistency Experiments

For all consistency experiments, the dataset was divided into a train dataset (80% of the data) and a test dataset (20%) of the data.

The parameters of the estimators were set as follows:

- all RF estimators have 500 *trees* to mimic the behavior of the infinite RF.
- parameter *bootstrap* is set to *False* for all estimators in order preserve the interpolation property, or set to *True* when specified.
- all other parameters are set to default value.

Consistency of KeRF in the Mean Interpolation Regime

We train a centered KeRF (with $M = 500$) of depth fixed to $\lfloor \log_2 n \rfloor + 1$ (mean interpolation regime) for different sample sizes n and evaluate the empirical quadratic risk on the test set.

Results On Figure S7, for all models, the risk decreases toward zero as the number of samples n increases (with slow convergence rates). These numerical results, even though obtained for a finite KeRF with a large number $M = 500$ of centered trees, support the theoretical consistency of the infinite KeRF in the mean interpolation regime (see Theorem 4.4.1).

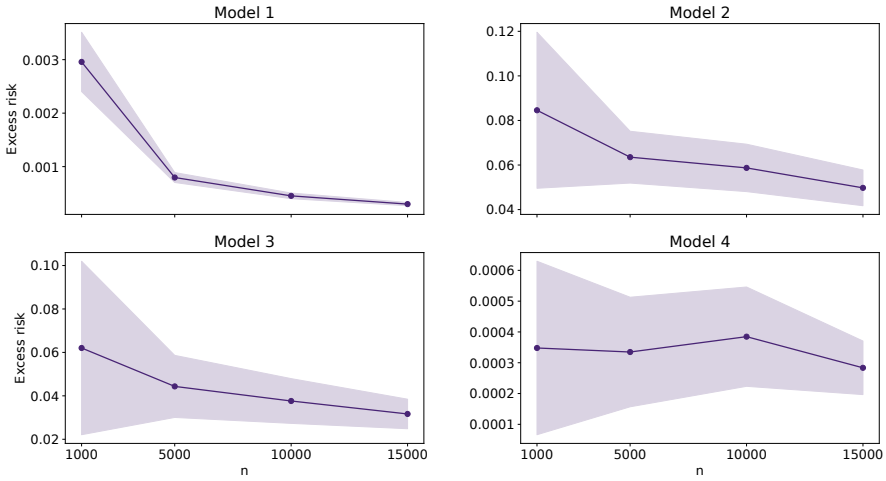


Figure S7: KeRF consistency results: excess risk w.r.t. sample sizes. For each sample size n, the experiment is repeated 30 times: we represent the mean over the 30 tries (bold line) and the mean $\pm$ std (filled zone).

Consistency of Median RF in the Interpolation Regime

We analyze the empirical performances of Median RF in noiseless and noisy settings on the models specified above. For each model, given a training set, we train Median RF (with $M = 500$ trees) until pure leaves are reached, and measure its excess risk on a test set.

Figure S8 shows that the excess risk of a Median RF decreases as n grows. These empirical performances lend support to the idea that Median RF are consistent even with a finite number of trees and beyond the noiseless setting.

Consistency of Breiman RF, Additional Models to Figure 1

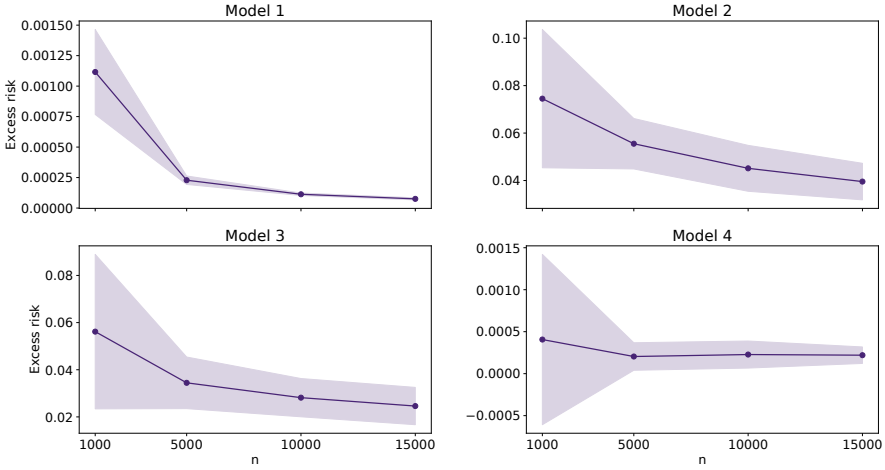


Figure S8: Consistency results for a Median RF with $M = 500$ trees: excess risk w.r.t. the sample size n. For each sample size, the experiment is repeated 30 times: we represent the mean over the 30 tries (bold line) and the mean $\pm$ std (filled zone).

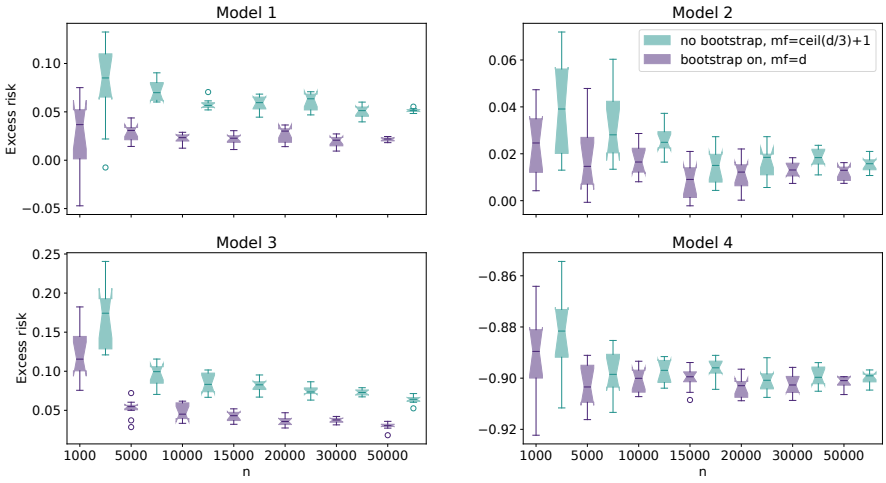


Figure S9: Consistency of Breiman RF: excess risk w.r.t the sample size n. RF parameters: 2000 trees, max-depth set to None, max-features$= 1$. Boxplots over 10 tries.

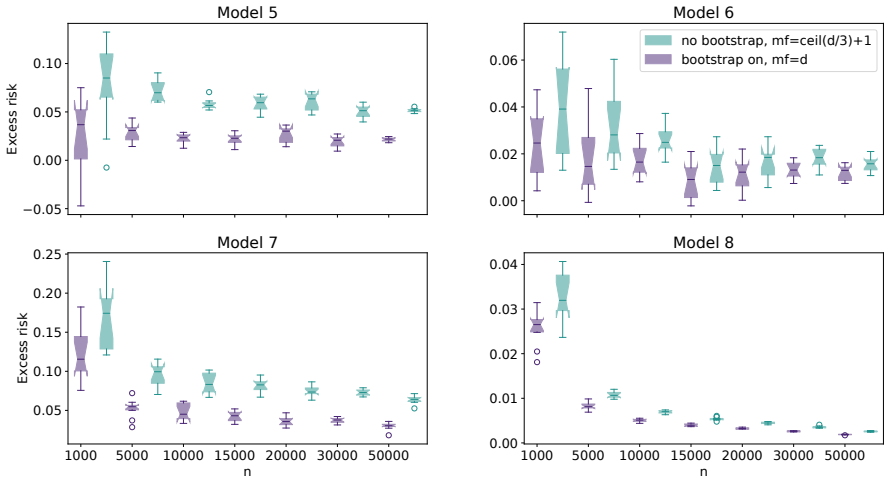


Figure S10: Consistency of Breiman RF: excess risk w.r.t the sample size n. RF parameters: 2000 trees, max-depth set to None, max-features$= 1$. Boxplots over 10 tries.

Consistency of Breiman RF with Max-Feature$= 1$

On Figure S11, we see that the excess risk of a Breiman RF with the max-features parameter set to 1 is decreasing towards 0 as n increases. This RF seems consistent for all models.

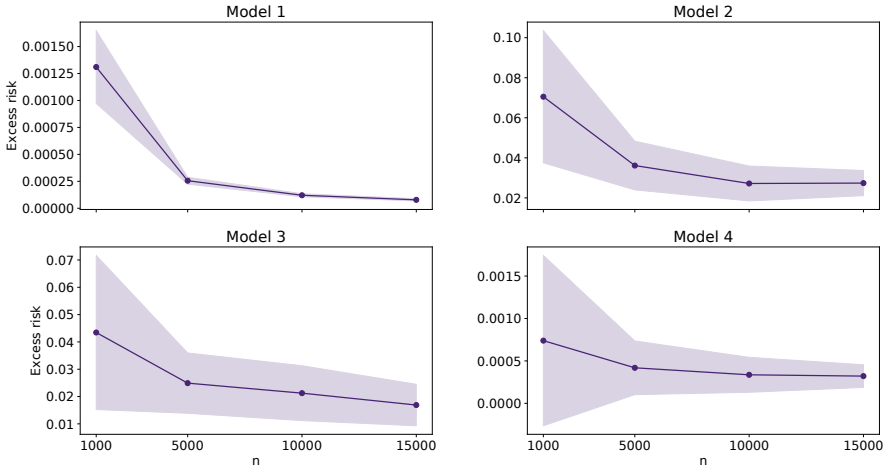


Figure S11: Consistency of Breiman RF: excess risk w.r.t sample size. RF parameters: 500 trees, max-depth set to None, max-features$= 1$, no bootstrap. Mean over 30 tries (doted line) and std (filled zone).

Decrease of the Variance of the Breiman RF in a High-Dimensional Setting

Numerical experiments show the decrease of the variance of interpolating Breiman RF when d increases. The model involves no signal and only noise (with specified variance σ^2).

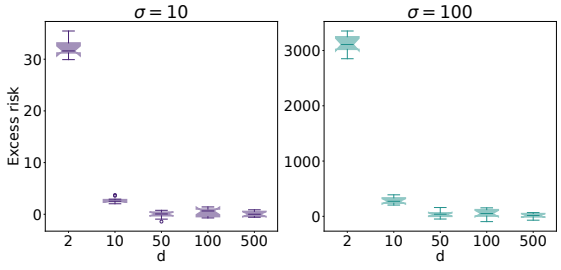


Figure S12: Decrease of the variance of an interpolating Breiman RF with `max-features=1` w.r.t. dimension d. 10 repetitions per boxplot, 5000 training points and 50000 testing points were used for each repetition. The Breiman RF contains 1000 trees.

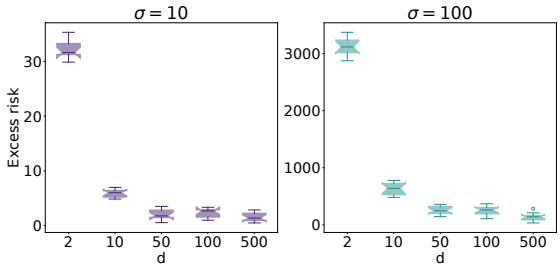


Figure S13: Decrease of the variance of an interpolating Breiman RF with `max-features=`$\lfloor d/3 \rfloor$ w.r.t. dimension d. 10 repetitions per boxplot, 5000 training points and 50000 testing points were used for each repetition. The Breiman RF contains 1000 trees.

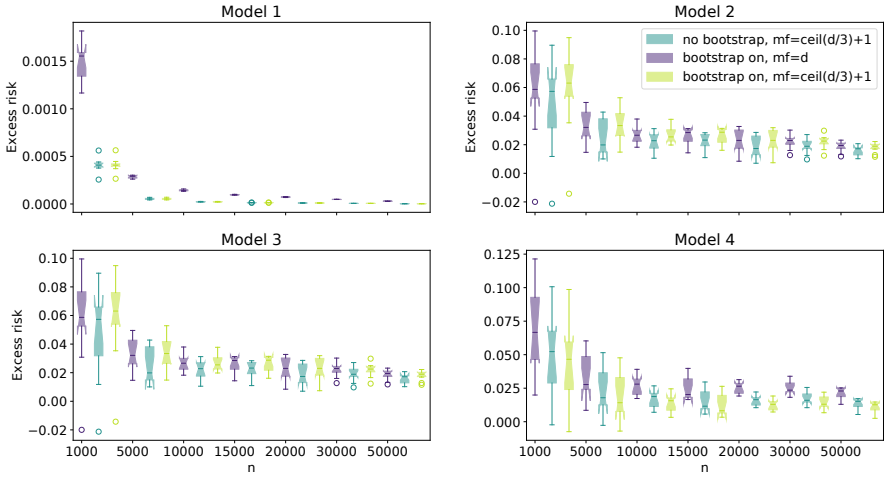


Figure S14: Consistency of Breiman RF: excess risk w.r.t sample size. RF parameters: 2000 trees, max-depth set to None, max-features= 1. Boxplots over 10 tries.

Comparison of Breiman RF with and without Bootstrap

S3.2 Interpolation experiments

Volume of the Interpolation Zone w.r.t Sample Size n

We numerically evaluate the volume of the interpolation area of a Breiman RF (with 5000 trees, see Figure S17 in Appendix S3.2 for details about this choice) when the sample size n increases.

In Figure S15, the volume of the minimal interpolation zone is shown to tend polynomially fast to 0 (linear in the logarithmic scale) for all considered models as the dataset size increases, matching the behavior of the theoretical bound established in Proposition 4.6.1.

One could notice the slight gap between the theoretical and experimental curves, which actually reflects the gap between an infinite forest (for which Proposition 4.6.1 holds) and its approximation by a finite forest (5000 trees here). This gap naturally tends to increase with n (when the number of trees is fixed) as the approximation of the infinite RF by a finite one deteriorates with n.

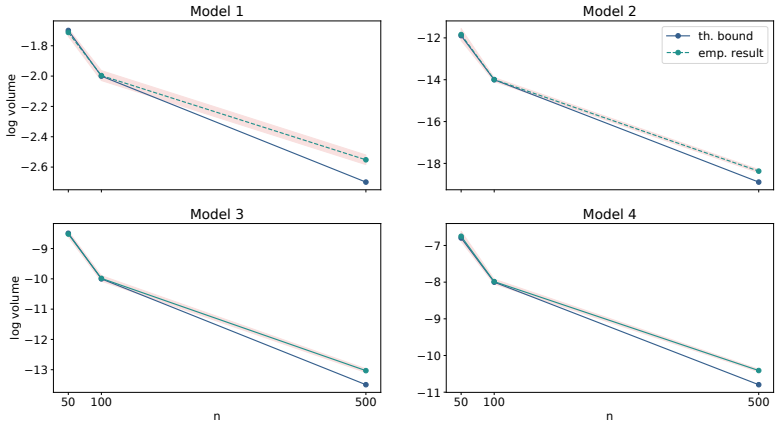


Figure S15: Log volume of the interpolation zone of a Breiman RF with 5000 Trees, max features set to 1, no bootstrap. Mean over 10 tries (red line) and mean $\pm$ std (filled zone). The theoretical bound (Proposition 4.6.1) is represented in green.

Increasing `max-feature` parameter We plot on Figure S16 the log-volume of the interpolation zone of a Breiman RF with the max-features parameter set to $\lceil d/3 \rceil$ (the default value proposed in `R randomForest` package). The volume decreases polynomially in n but slower than when max-features$= 1$ (Figure S15) which is to be expected: choosing max-features$= 1$ should increase the diversity of the splits and therefore reduce the volume of the interpolation zone.

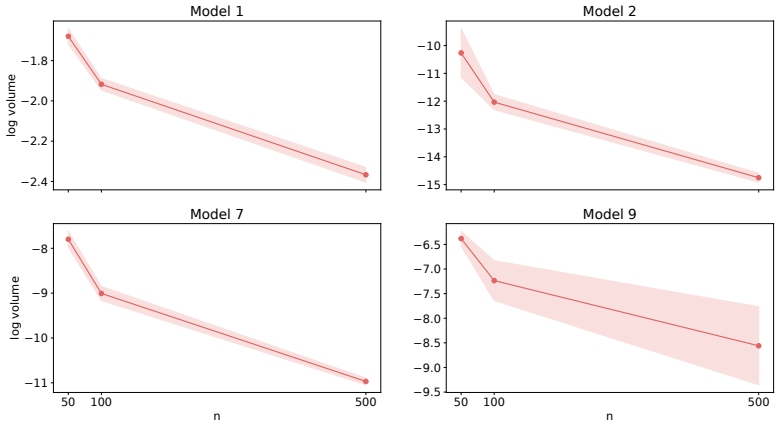


Figure S16: Log volume of Breiman RF interpolation zone w.r.t. sample size n. RF parameters: 500 trees, no bootstrap, max features $= \lceil d/3 \rceil$. Mean over 10 tries (bold line) and std (filled zone).

Volume of the Interpolation Zone w.r.t Number of Trees M

In this section, we empirically measure how fast decreases the volume of the interpolation zone of a Breiman RF when its number of trees M increases, and how close the interpolation zone gets from the minimal interpolation zone.

To this end, for a fixed sample size $n = 500$, we numerically evaluate the volume of the interpolation area when the number M of trees in the forest grows. This volume is anticipated to

be a non-increasing function of M (for $M = 1$, note that the interpolation volume is 1, the volume of $[0, 1]^d$), but its decrease rate highly depends on the data geometry, making its theoretical evaluation difficult. The numerical results in Figure S17 show a fast decay towards zero of the interpolation volume for all models, already tiny from $M = 500$ trees. Furthermore, it seems to converge to the theoretical bound (dotted line) derived in Proposition 4.6.1 for an infinite RF with a max-feature parameter equal to 1.

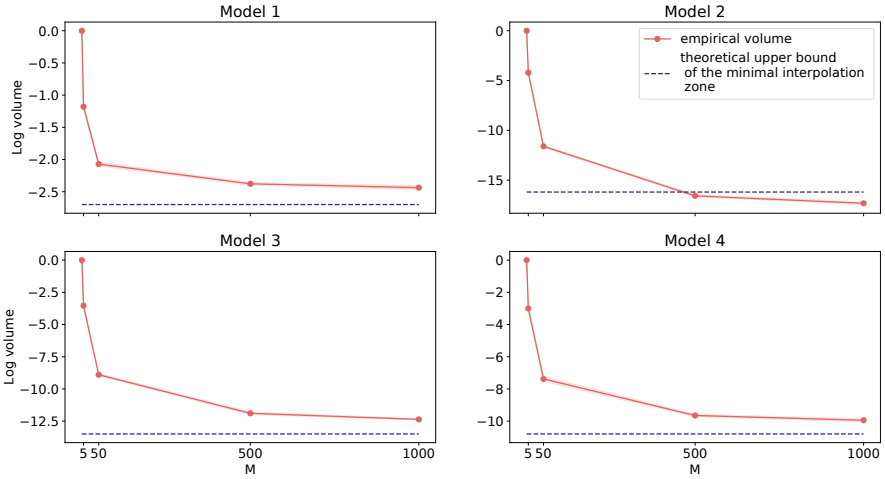


Figure S17: Log volume of Breiman RF interpolation zone w.r.t. the number M of trees. RF parameters: no bootstrap, max features $= 1$. Mean over 10 tries (bold line) and std (filled zone). Sample size $n = 500$.

Analysis of the Interpolation Property of Breiman RF with Bootstrap

In this experiment, we try to measure how close a Breiman RF with bootstrap on is from exactly interpolating (with other parameters being 500 trees, max-depth set to None, max-features$= d$). To this end, we measure the difference between the true train labels (the Y_is) and the predicted ones (the $\hat{Y}_i$s) by computing

$$I_{\text{loss}} := \frac{1}{n} \sum_{i=1}^{n} \frac{|Y_i - \hat{Y}_i|}{Y_i}.$$

The closer is this quantity to 0, the closer is the forest from interpolating. On Figure S18, we plot different quantiles of the above quantity as n varies.

For instance, if we take the 0.8-quantile in red on Figure S18 and look at the upper-right plot (model 2), we read that the I_{loss} roughly equals 0.6 for 80% of the points. This quantity seems globally constant in n. Finally, the quantiles are smaller in the case of a strong signal-to-noise ratio (models 1 and 4) than in the case of a bigger one (models 2 and 3).

On Figure S19, we also plot the quantiles of the I_{loss} for the four different models while the number of trees varies. Adding trees does not significantly change the value of the different quantiles.

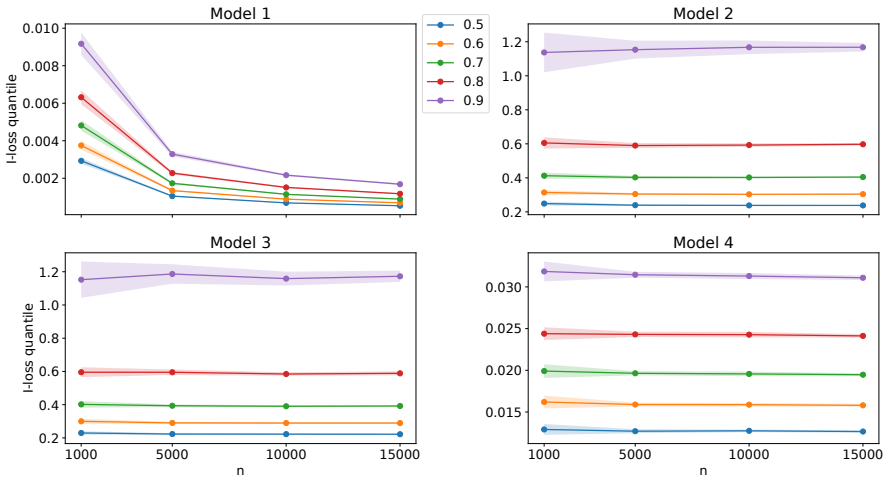


Figure S18: I_{loss} of a Breiman RF w.r.t sample size n. RF parameters: 500 trees, bootstrap on, max-features$= d$, max-depth set to None. Mean over 30 tries (doted lines) and std (filled zones).

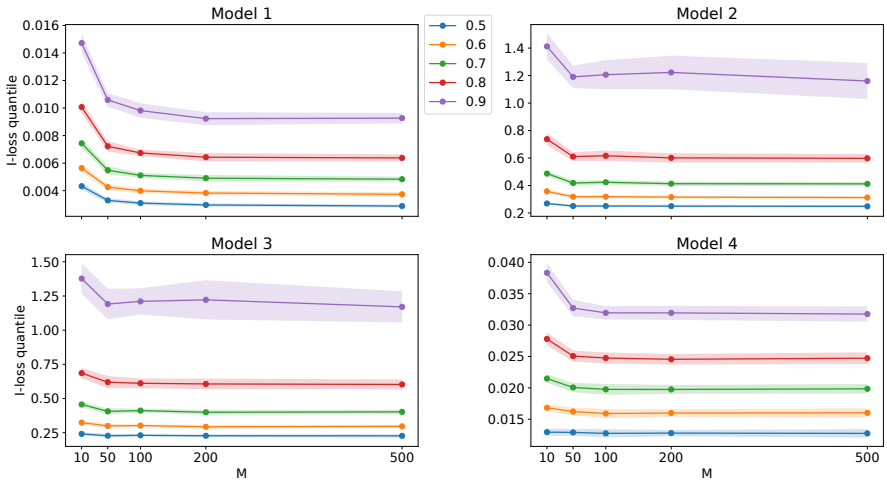


Figure S19: I_{loss} of a Breiman RF w.r.t number of trees. Parameters: bootstrap on, max-features$= d$, max-depth set to None. Sample size $n = 1000$. Mean over 30 tries (doted lines) and std (filled zones).

Bibliography of the current chapter

Arlot, Sylvain and Robin Genuer (2014). "Analysis of purely random forests bias". In: *arXiv preprint arXiv:1407.3939*.

Bach, Francis and Lenaic Chizat (2021). *Gradient Descent on Infinitely Wide Neural Networks: Global Convergence and Generalization*. arXiv: 2110.08084 [cs.LG].

Bartlett, Peter L, Philip M Long, Gábor Lugosi, and Alexander Tsigler (2020). "Benign overfitting in linear regression". In: *Proceedings of the National Academy of Sciences* 117.48, pp. 30063–30070.

Bartlett, Peter L, Andrea Montanari, and Alexander Rakhlin (2021). "Deep learning: a statistical viewpoint". In: *arXiv preprint arXiv:2103.09177*.

Batir, Necdet (2008). "Inequalities for the gamma function". In: *Archiv der Mathematik* 91.6, pp. 554–563.

Belkin, Mikhail, Daniel Hsu, Siyuan Ma, and Soumik Mandal (2019a). "Reconciling modern machine-learning practice and the classical bias–variance trade-off". In: *Proceedings of the National Academy of Sciences* 116.32, pp. 15849–15854.

Belkin, Mikhail, Alexander Rakhlin, and Alexandre B Tsybakov (2019b). "Does data interpolation contradict statistical optimality?" In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 1611–1619.

Biau, Gérard (2012b). "Analysis of a random forests model". In: *The Journal of Machine Learning Research* 13, pp. 1063–1095.

Breiman, Leo (2001a). "Random forests". In: *Machine learning* 45.1, pp. 5–32.

Breiman, Leo, Jerome Friedman, Charles J Stone, and Richard A Olshen (1984). *Classification and regression trees*. CRC press.

Buschjäger, Sebastian and Katharina Morik (2021). "There is no Double-Descent in Random Forests". In: *arXiv preprint arXiv:2111.04409*.

Chizat, Lenaic and Francis Bach (2020). "Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss". In: *Conference on Learning Theory*. PMLR, pp. 1305–1338.

Devroye, Luc, Laszlo Györfi, and Adam Krzyżak (1998). "The Hilbert kernel regression estimate". In: *Journal of Multivariate Analysis* 65.2, pp. 209–227.

Duroux, Roxane and Erwan Scornet (2018). "Impact of subsampling and tree depth on random forests". In: *ESAIM: Probability and Statistics* 22, pp. 96–128.

Geurts, P., D. Ernst, and L. Wehenkel (2006). "Extremely randomized trees". In: *Machine learning* 63.1, pp. 3–42.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep learning*. MIT press.

Hastie, Trevor, Andrea Montanari, Saharon Rosset, and Ryan J. Tibshirani (2022). "Surprises in high-dimensional ridgeless least squares interpolation". In: *The Annals of Statistics* 50.2, pp. 949 –986. DOI: 10.1214/21-AOS2133. URL: https://doi.org/10.1214/21-AOS2133.

Ishwaran, Hemant (2015). "The effect of splitting on random forests". In: *Machine learning* 99.1, pp. 75–118.

Klusowski, Jason M. (2021a). "Sharp analysis of a simple model for random forests". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 757–765.

Laan, Mark J Van der, Eric C Polley, and Alan E Hubbard (2007). "Super learner". In: *Statistical applications in genetics and molecular biology* 6.1.

Liang, Tengyuan, Alexander Rakhlin, and Xiyu Zhai (2020b). "On the multiple descent of minimum-norm interpolants and restricted lower isometry of kernels". In: *Conference on Learning Theory*. PMLR, pp. 2683–2711.

Lin, Yi and Yongho Jeon (2006). "Random forests and adaptive nearest neighbors". In: *Journal of the American Statistical Association* 101.474, pp. 578–590.

Mentch, Lucas and Siyu Zhou (2019). "Randomization as regularization: a degrees of freedom explanation for random forest success". In: *arXiv preprint arXiv:1911.00190*.

Mourtada, Jaouad, Stéphane Gaïffas, and Erwan Scornet (2020). "Minimax optimal rates for Mondrian trees and forests". In: *The Annals of Statistics* 48.4, pp. 2253–2276.

Richmond, Lawrence Bruce and Jeffrey Shallit (2009). "Counting abelian squares". In: *Electronic Journal of Combinatorics*.

Scornet, Erwan (2016a). "On the asymptotics of random forests". In: *Journal of Multivariate Analysis* 146, pp. 72–83.

— (2016b). "Random forests and kernel methods". In: *IEEE Transactions on Information Theory* 62.3, pp. 1485–1500.

Tang, Cheng, Damien Garreau, and Ulrike von Luxburg (2018). "When do random forests fail?" In: *Advances in neural information processing systems* 31.

Tsigler, Alexander and Peter L Bartlett (2020). "Benign overfitting in ridge regression". In: *arXiv preprint arXiv:2009.14286*.

Wang, Yutong and Clayton D Scott (2022). "Consistent Interpolating Ensembles via the Manifold-Hilbert Kernel". In: *arXiv preprint arXiv:2205.09342*.

Wyner, Abraham J, Matthew Olson, Justin Bleich, and David Mease (2017). "Explaining the success of adaboost and random forests as interpolating classifiers". In: *The Journal of Machine Learning Research* 18.1, pp. 1558–1590.

Zhou, Siyu and Lucas Mentch (2021). "Trees, forests, chickens, and eggs: when and why to prune trees in a random forest". In: *arXiv preprint arXiv:2103.16700*.

Chapter 5

PAC-Bayes

Preambulum

We inform the reader that the following material is a work in progress. The content presented herein has not yet been subjected to the rigorous scrutiny that typically accompanies finalized research. Consequently, some results and statements may lack comprehensive proof or thorough validation. We did our best to mention when such results are not rigorously demonstrated and what further experiments will be conducted to make up for the current shortcomings.

5.1 Introduction

The recent success of Neural Networks (NN) is largely due to a combination of overparametrization and artifacts to improve generalization (e.g., dropout layers, see Srivastava et al. 2014). Through Stochastic Gradient Descent (SGD), a NN can be effectively trained to minimize the empirical risk in a supervised setting, often achieving a perfect fit to training data while still generalizing well to unseen data. The mechanisms underpinning this empirical behavior remain poorly understood theoretically. As a consequence, it remains difficult in practice to design NN architectures and/or training procedures that specifically enhance the generalization properties. The main proposal of this work is thus to study a training scheme inspired from the theoretical PAC-Bayes framework to improve the generalization capacities of any NN in a supervised setting.

For many years, statistical theory has advocated the compromise between bias and variance to design good estimators. To sum up, increasing the approximation capacities of a model by adding parameters should also, at some point, increase the variance, directly accountable for poor generalization performances. However, the success of deep and wide NN results from the fact that very complex methods in terms of number of parameters, both interpolate the training data and generalize well (Goodfellow et al. 2016). Since then, many researchers have tried to explain, both empirically and theoretically, the good generalization performances of NN despite their high complexity (see Paragraph Related Works below).

Several attempts to study the generalization capacities of NN exploit the PAC-Bayes theory that allows us to efficiently control the generalization gap, i.e. the difference between the empirical and the theoretical risk. The first non-vacuous bounds on NN were obtained a few years ago in Dziugaite et al. 2017. After that, many efforts have been conducted to improve the PAC-Bayes bounds in order to close the gap between theory and practice (see, e.g., Zhou et al. 2018; Letarte et al. 2019; Lan et al. 2020; Tsuzuku et al. 2020; Clerico et al. 2023).

Taking advantage of this control, in a recent work, Pérez-Ortiz et al. 2021 introduced a new training objective to minimize an upper bound on the theoretical risk, instead of minimizing only the empirical one, in order to improve the guarantees on the generalization performance of a NN. The bound is a typical combination of the empirical risk and a PAC-Bayes penalty, which controls the generalization gap. Their method also leverages the *flatness* of the loss landscape to minimize the PAC-penalty. The connection between the flatness of a minimum and its generalization capacity has drawn a lot of attention lately (see for instance He et al. 2019; Jiang et al. 2019; Keskar et al. 2016) and seems to be a promising direction to both explain the generalization power of overparametrized networks and to develop new training methods (Foret et al. 2020; Du et al. 2021; He et al. 2019).

Contributions Through extensive numerical experiments, we study and extend the training scheme of Pérez-Ortiz et al. 2021. We aim at showing that it improves the generalization power of a NN and also increases the flatness of the loss landscape. Our main contribution would thus be twofold:

1. **Link between generalization and loss flatness.** We would like to verify if adding the PAC-Bayes penalty to the training objective increases the flatness of the loss landscape, thus exhibiting a link between the generalization power of a NN and the flatness of its loss landscape. The former is measured via the top eigenvalues of the hessian of the network parameters, using an approximation method developed in Golmant et al. 2018.

2. **PAC-Bayes penalty for improved generalization.** We would like to use the PAC-Bayes penalty to enhance the generalization performances of the NN measured on a test set. Compared to Pérez-Ortiz et al. 2021, we do not focus on obtaining a bound on the theoretical risk of the network but rather to improve the empirical performances of a NN evaluated on a test set, which also allows us to slightly diverge from the theoretical framework.

Related Works Traditional complexity measures (VC dimension, Rademacher complexity...) fail to capture the good generalization performances of over-parametrized NN (Belkin 2021). On the other hand, the PAC-Bayes framework has shown very promising results in this direction. Since the first non-vacuous bound obtained in 2017 (Dziugaite et al. 2017), many papers have improved the bounds on the generalization gap for several kinds of NN architectures as mentioned above (Zhou et al. 2018; Letarte et al. 2019; Lan et al. 2020; Tsuzuku et al. 2020; Clerico et al. 2023). Other theoretical approaches include the study of benign overfitting in the case of linear regression (Bartlett et al. 2020; Tsigler et al. 2020; Liang et al. 2020b), also investigated in the context of neural networks (Belkin et al. 2019a). The influence of training via SGD on the generalization power of NN is also pointed out and studied by researchers interested in the *implicit bias* or *implicit regularization*: the optimization of an over-parametrized one-hidden-layer neural network via SGD will converge to a minimum of minimal norm with good generalization properties in a regression setting (Bach et al. 2021), or with maximal margin in a classification setting (Chizat et al. 2020). However, these approaches often focus on simplified architectures, which cannot fully explain the behavior of DNN used in practice. On the other hand, based on numerical observations, several empirical measures have been proposed to quantify the generalization properties of NN (Jiang et al. 2019). In the former, they notice that sharpness-based measures seem to be the most correlated to generalization performances. The connection between the flatness of the minima and their generalization power has been studied extensively from both theoretical and empirical perspectives (Dziugaite et al. 2017; Jiang et al. 2019; Keskar et al. 2016). Taking advantage of this connection, very recently, several papers have considered minimizing the sharpness of the loss of NN in order to improve their generalization (namely Sharpness

Aware Minimization procedures). For instance, Foret et al. 2020 minimizes the maximum of the loss in a ball of small radius at each step, instead of minimizing simply the loss. Several variants of this method have been later introduced (see e.g., Kwon et al. 2021; Du et al. 2021). On another note, He et al. 2019 and Izmailov et al. 2018 both propose to average the weights of several NN (either the same network at different steps of the training or different NN) to reach flatter regions, which empirically improves the generalization.

5.2 Training a NN under a PAC-Bayes Objective

5.2.1 Data and Estimators

We consider a classification task in a supervised learning setting. We have a dataset $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ of independent pairs drawn from an unknown distribution P on the space $\mathcal{X} \times \mathcal{Y}$. Typically, $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \{1, ..., K\}$ where K is the number of classes (depending on the dataset at hand). We denote ℓ the cross-entropy loss used to assess the quality of the prediction: $\ell : (\pi(x), y) \to -y \log \pi(x)$ where $\pi(x)$ is a probability vector.

The neural network weights are represented by a vector $\boldsymbol{w} \in \mathbb{R}^p$. In this work, we will consider several kinds of Convolutional Neural Networks (CNN) (detailed in Appendix S1.1). We denote $f_w : \mathcal{X} \to \mathcal{Y}$ the associated neural network. Neural networks are usually trained to minimize the empirical risk $r_n(w) = \frac{1}{n} \sum_{i=1}^{n} \ell(f_w(x_i), y_i)$ via SGD. Its theoretical counterpart, the risk, is denoted $R(w) := \mathbb{E}\left[\ell(f_w(X), Y)\right]$.

In a PAC-Bayes setting, we consider Probabilistic Neural Networks (PNN) which are (data-dependent) probability distributions over the weight space $\mathbb{R}^p$. An estimator can be built in several ways from a PNN: by sampling from the PNN distribution (randomized estimator), by taking the mean of the distribution (mean estimator), etc. The risk of a PNN Q is defined as $R(Q) = \mathbb{E}_{w\sim Q}[R(w)]$ and the empirical risk as $r_n(Q) = \mathbb{E}_{w\sim Q}[r_n(w)]$ (we omit the dependence on the data to lighten the notations). To train a PNN, instead of simply minimizing the empirical risk, we minimize a sum of the empirical risk plus a penalty term which is derived from a PAC-Bayes bound as shown in the next section.

5.2.2 Training Objectives

A PAC-Bayes bound controls the discrepancy between the empirical and theoretical risks with high probability over the data. We first provide an example of a PAC-Bayes bound from McAllester 1999.

Theorem 5.2.1 (Mc Allester, 1999)**.** *Given a prior distribution Q_0, $\delta \in (0, 1)$, then, with probability at least $1 - \delta$ over S, for all Q, we have*

$$R(Q) \leqslant r_n(Q) + \sqrt{\frac{KL(Q||Q_0) + \log\left(\frac{2\sqrt{n}}{\delta}\right)}{2n}}$$

The prior distribution can be arbitrary chosen or built from a subset of the training data as we will see later. The PAC-Bayes framework is quite flexible and allows one to derive a wide variety of bounds. An overview of all the bounds that we use is given in Appendix S1.2. We can use these bounds as new objectives to train a NN instead of implementing a simple empirical risk

minimization, that is

$$\phi(Q) := r_n(Q) + \sqrt{\frac{\mathrm{KL}(Q||Q_0) + \log\left(\frac{2\sqrt{n}}{\delta}\right)}{2n}}$$

becomes the new objective to minimize. Since, with high probability $R(Q) \leqslant \phi(Q)$, minimizing $\phi(Q)$ directly accounts for minimizing an upper bound on the theoretical risk, which is the *true* objective to minimize. In practice, as very powerful NN are used, the empirical risk often reaches 0; therefore the real challenge is to decrease the penalty term, especially the KL part.

5.3 Training Process

Pérez-Ortiz et al. 2021 developed a strategy in order to find a distribution Q minimizing the objective previously introduced. This training scheme is also the one used in this work, and we summarize it here:

1. **Parametrization of Q and Q_0.** In order to parametrize the distributions, we rewrite the weights $w = \mu + \sigma \odot V$ with $\mu \in \mathbb{R}^p$, $\sigma \in \mathbb{R}^p$ and V a random vector of size p following a Gaussian or Laplace distribution P_V. Learning Q accounts for learning its parameters μ and σ.

2. **Choice of the prior Q_0.** In order to decrease the KL term in the bound, Q_0, parametrized by μ_0 and σ_0, has to be chosen specifically. First, we initialize and train a classical (deterministic) network on a subset of the data (e.g., 50% of the data). It provides weights W_0 that will serve as μ_0 for the prior distribution Q_0. Then σ_0 is simply chosen from a grid search (see details below).

3. **Learning Q.** Given the initialization Q_0 with parameters (μ_0, σ_0), we can learn the parameters μ and σ via SGD. As usually done, we apply SGD on random batches of data for a given number of iterations. For each batch, we sample the parameters $w \sim Q$, we compute the loss of the corresponding network on the data batch and then compute the gradient of the objective ϕ w.r.t. μ and σ: $\nabla_\mu \phi = \frac{\partial \phi}{\partial W} + \frac{\partial \phi}{\partial \mu}$ and $\nabla_\sigma \phi = \frac{\partial \phi}{\partial W} \cdot \frac{V}{1+\exp(-\sigma)} + \frac{\partial \phi}{\partial \sigma}$. This computation can be decomposed into two terms, the gradient of the empirical risk and the gradient of the PAC-Bayes penalty. The computation of the gradient of the empirical risk is direct: $\nabla_\mu \ell(f_w(x), y)$ and $\nabla_\sigma \ell(f_w(x), y)$. The KL term being explicit w.r.t. μ and σ (when P_V is a Gaussian or Laplace distribution), we can also compute its gradient w.r.t. the parameters. Considering two one-dimensional Gaussian distributions Q and Q_0, it writes as follows: $\mathrm{KL}(Q||Q_0) = \frac{1}{2}\left(\log(\frac{\sigma_0}{\sigma}) + \frac{(\mu_1-\mu_0)^2}{b_0} + \frac{b_1}{b_0} - 1\right)$.

4. **Re-initialization.** In order to improve the training performances of the network, after a condition is met (number of epochs...), we actualize the prior by setting $Q_0 = Q$. This allows the posterior to move further away from the initial distribution Q_0, an area for which the empirical risk can remain high.

More details, such as the KL computation between two Gaussian distributions, can be found in Pérez-Ortiz et al. 2021.

As we show in the following section, this training scheme reduces the curvature of the loss landscape w.r.t. the weights of the network.

5.4 PAC-Bayes Penalty and Flatness

In this section, we compare the *flatness* of the loss optimization landscape of a neural network trained via empirical risk minimization and one trained with a PAC-Bayes penalty added to the empirical risk.

5.4.1 Evaluating the Optimization Loss Landscape Curvature

In order to measure the curvature of the NN loss optimization landscape, we compute the highest eigenvalues of the Hessian of the loss r_n at point $w \in \mathbb{R}^p$. The Hessian structure provides information on local curvature, and therefore on flatness. This characterization of the flatness of a minimum[1] has been considered in, e.g., Ghorbani et al. 2019; Chaudhari et al. 2019; Foret et al. 2020; Gilmer et al. 2021. When the number of parameters exceeds a few thousands, computing the eigenvalues of the Hessian is cumbersome. We thus rely on an approximation algorithm implemented in Golmant et al. 2018.

5.4.2 Experimental Protocol

We consider a deterministic network `det-net` and a probabilistic one, the latter referring to the posterior net of the pair `prior-net`, `posterior-net`, both having the same architecture. The training parameters are also identical (optimizer, learning-rate, batchsize, etc.). We measure the top eigenvalue of the Hessian of the network parameters for both networks at initialization and at the end of training. The deterministic net is trained in a standard way via SGD to minimize the empirical risk, and the probabilistic one is trained according to the procedure described in Section 5.3. Regarding the probabilistic network, the prior network is trained for a fixed number of epochs and the posterior network is trained with *early stopping*: as soon as the training loss decreases below a given threshold ε, we stop the training. The same early stopping condition is used to train the deterministic network. In this manner, the eigenvalues are evaluated when both networks reach the same training state.

We consider three different kinds of convolutional architectures ordered by size: MobileNet (roughly 200k parameters, Howard et al. 2017), DenseNet (8M parameters, Huang et al. 2017) and ResNet101 (44.5M parameters, He et al. 2016). MobileNet stacks light blocks of convolutional/normalization layers. DenseNet employs dense connectivity by receiving feature maps from all preceding layers to facilitate feature reuse and gradient flow. Finally, ResNet adds residual connections to the classical convolutional layers, allowing the signal to propagate much deeper. Details about these models are given in Appendix S1.1.

In order to show that the decrease of the top eigenvalues is robust to the change of hyperparameters, we run the experiment on MNIST dataset for different training/architecture configurations of MobileNet specified as follows: 3 different depths (1, 2 or 4 mobile blocks), 2 widths (small, large), 3 different learning rates (1e-3, 5e-3, 1e-3), 3 different PAC-Bayes inspired objectives (fquad, bbb and flamb defined in Section S1.2), dropout on (with probability 0.1) or off and a KL penalty set at 1 or 0.1. We also evaluated this protocol on CIFAR for all the networks mentioned.

5.4.3 Results

As can be seen on Figure 5.1, on the MNIST dataset, in average over all the MobileNet architectures, the top eigenvalues of the Hessian decrease when we consider the posterior network and its penalized objective. The prior and deterministic network have the same architecture and training

[1] Other definitions are discussed in Dinh et al. 2017.

objectives, therefore at initialization their top eigenvalues are of the same order. The eigenvalues at initialization are much higher than at the end of training which is to be expected as they fall at a random point which has no reason to be flat over all directions. The posterior network is initialized at the end of the training of the prior network: between "prior end" and "posterior initialization" only the training objective changes and this already brings a decrease of the eigenvalues of the Hessian. Overall, the smallest top eigenvalues are obtained at the end of the training of the posterior network. Similar results are observed for the DenseNet and ResNet NN on both MNIST and CIFAR as can be seen in Section S1.3.

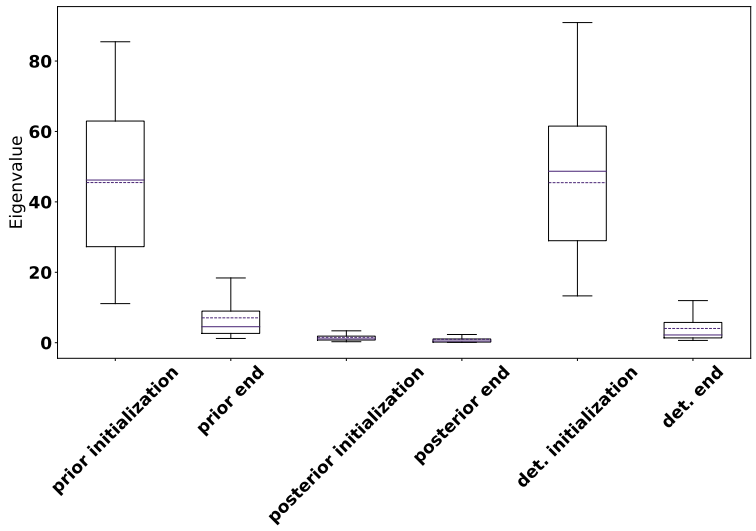


Figure 5.1: NN hessian top eigenvalue of deterministic, prior and posterior MobileNet. For each network, the top eigenvalue is computed at the initialization of the network ("... initialization" on the x axis) and at the end of training ("... end" on the x axis) Boxplot over 3 times, MNIST dataset.

Adding the PAC-penalty, which controls the generalization gap, to the training objective flattens the loss around the minimum reached by the network. Although the bound is based on several theoretical hypotheses that might not be verified in practice (such as i.i.d. data samples), this outcome is still encouraging and sheds further insight into the connection between generalization and the smoothness of the loss landscape. Further experiments should be implemented to explore this connection. In particular, it would be nice to unveil a mathematical link between the value of the generalization penalty and the curvature of the loss landscape. It would require a more theoretical analysis with more experiments on hand-made examples.

In the following section, we try to enhance the generalization power of the networks with the PAC-Bayes penalty.

5.4.4 Generalization Performances under PAC-Bayes Inspired Training

After showing that the PAC-Bayes penalty allows NN to reach flatter minima, we would like to show that it also help to improve the performances of NN on the test set. Generalization performances can be measured through both the generalization score on a test set which was not used during training and the generalization gap, i.e. the difference between the train score and the test score. Due to the sharpness decrease observed in Section 5.4, one could expect pbnet to

have a smaller generalization gap and therefore, for equal train losses, a better test loss. We are currently running experiments to confirm this intuition.

In the following experiment, we compare the generalization performances of a deterministic network and its probabilistic counterpart trained with PAC-Bayes penalty. We train the posterior network according to the training process described in Section 5.3. In order to build the prior, we keep 10% of the data as a validation set: as soon as the network reaches a generalization gap superior to a given threshold, we stop its training and switch to training the pbnet.

As we are interested in obtaining good empirical performances instead of preserving theoretical guarantees, we slightly adapt the training scheme to enhance the performances of the network. Indeed, we observed that the posterior network was hard to optimize far from its initialization point, as the KL part of the loss grows quite fast. Consequently, we update the prior network several times during the training in order to reset at 0 the KL loss between the prior and posterior networks. We also added a penalty to the KL term for all the PAC-Bayes objectives in order to allow the posterior network to explore the parameter space more broadly.

However, the deterministic and posterior networks have the same performances, especially in generalization. Neither the test accuracy nor the generalization gap of pbnet significantly improve.

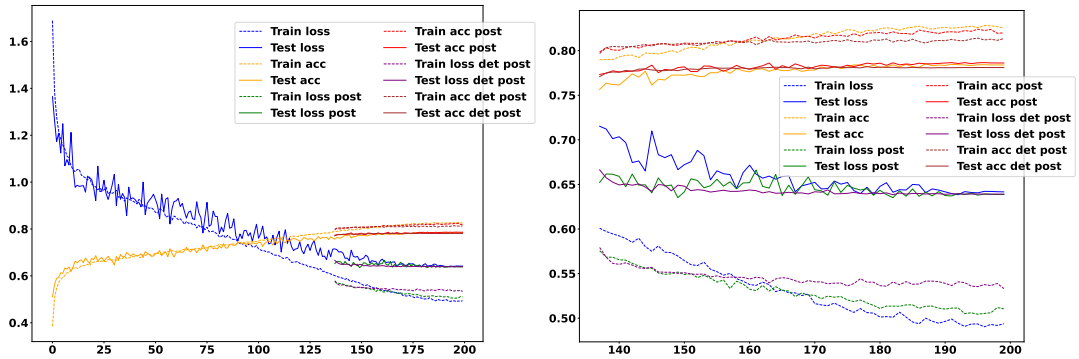


Figure 5.2: Comparison of a deterministic and a posterior MobileNet during training. Left: training from epoch 0 to epoch 220. Right: zoom on epochs 137-220 (beginning of posterior training). CIFAR10 Dataset.

As shown on Figures 5.3, similar results are obtained when we directly train the network according to the previous scheme from epoch 0 (the prior net is simply a random initialization in this case). Similar results are obtained with DenseNet and ResNet as shown in Section S1.4.

5.5 Conclusion and Further Work

Although the results of Section 5.4.3 were promising w.r.t. the connection between the generalization penalty and the loss flatness, it seems that in practice, on real datasets, we do not observe the expected improvement on the test error. This could be explained by the distance taken from the theoretical framework where the PAC upper bound holds. However, when sticking to the theoretical framework, the optimization of the posterior network is much more complicated as the KL term limits the minimization of the empirical risk.

The connection between the PAC penalty and the flatness of the loss landscape seems quite clear empirically, even though it has to be made more rigorous. However, the last link in the chain,

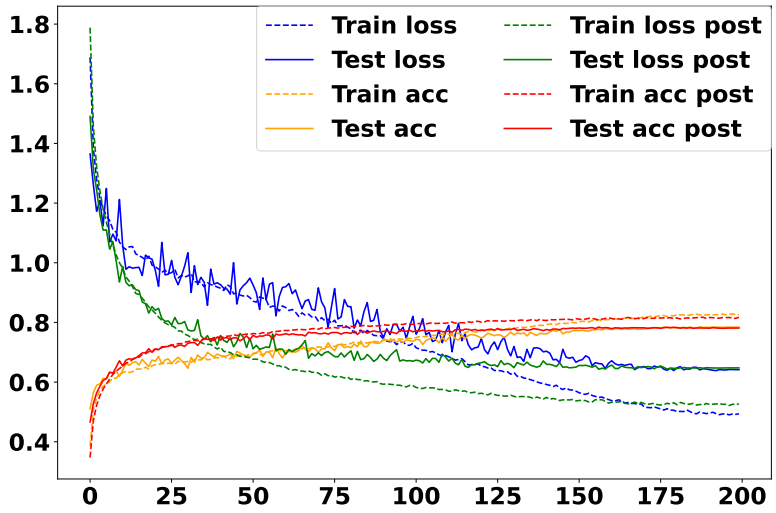


Figure 5.3: Comparison of a deterministic and a posterior MobileNet during training. The posterior network is trained from epoch 0 and the prior is updated every 5 epochs.

the reduction of the generalization gap, remains missing. As in the case of complex, real data, optimizing the posterior network is hard without moving away from the theoretical framework, more experiments should be implemented on a hand-made setting. In particular, it should be tested if for a similar value of the empirical risk, the deterministic and the PAC networks have different values of the loss flatness and/or of the generalization gap.

5.A Appendix

S1.1 Different Kinds of NN

We detail here the architectures of the networks used in the experiments. The DenseNet and ResNet NN refer to densenet121 and resnet101 of the `torchvision.models` package. The MobileNet architecture is slightly modified compared to the original one: `GroupNorm` layers are used instead of `BatchNorm` layers and the lengths of the layers are smaller.

S1.2 PAC-Bayes Objectives

Several objectives can be used to train a NN (Pérez-Ortiz et al. 2021):

$$\phi_{\text{lambda}}(Q,\lambda) := \frac{r_n(Q)}{1-\lambda/2} + \frac{KL(Q||Q_0) + log\left(\frac{2\sqrt{n}}{\delta}\right)}{n\lambda(1-\lambda/2)},$$

$$\phi_{\text{quad}}(Q) := \left(\sqrt{r_n(Q) + \frac{KL(Q||Q_0) + log\left(\frac{2\sqrt{n}}{\delta}\right)}{2n}} + \sqrt{\frac{KL(Q||Q_0) + log\left(\frac{2\sqrt{n}}{\delta}\right)}{2n}}\right)^2$$

and

$$\phi_{\text{bbb}}(Q) := r_n(Q) + \eta \frac{KL(Q||Q_0)}{n}.$$

The last objective, ϕ_{bbb}, is quite handy as it is possible to penalize the KL term with the η parameter.

S1.3 Additional Figures to the Flatness Experiment

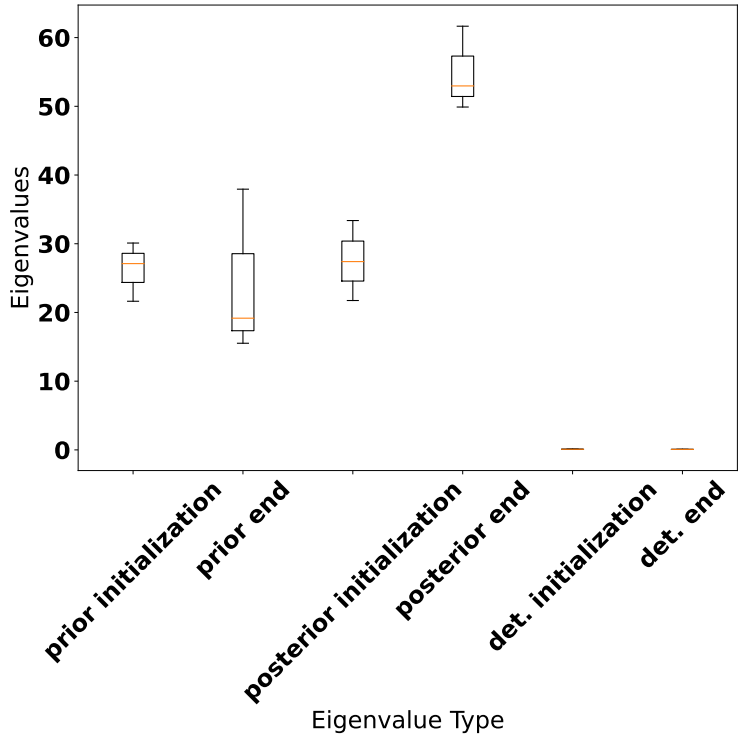


Figure S4: NN hessian top eigenvalue of deterministic, prior and posterior DenseNet. For each network, the top eigenvalue is computed at the initialization of the network ("... initialization" on the x axis) and at the end of training ("... end" on the x axis). Boxplot over 3 times, MNIST dataset.

S1.4 Additional Figures to the Generalization Experiment

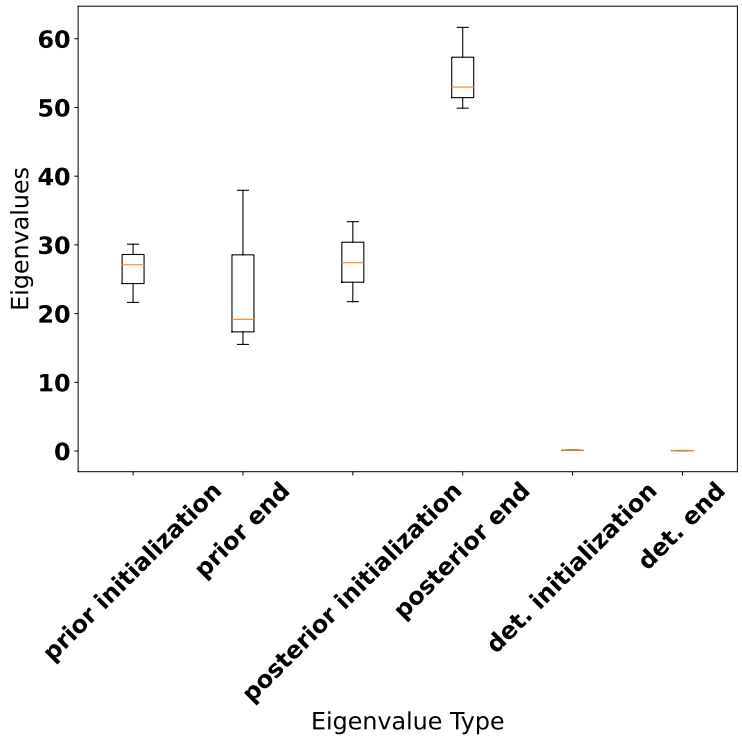


Figure S5: NN hessian top eigenvalue of deterministic, prior and posterior DenseNet. For each network, the top eigenvalue is computed at the initialization of the network ("... initialization" on the x axis) and at the end of training ("... end" on the x axis). Boxplot over 3 times, CIFAR10 dataset.

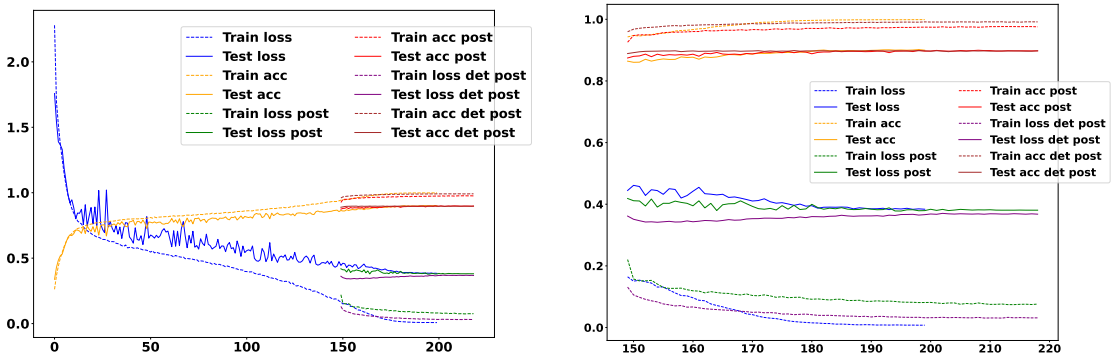


Figure S6: Comparison of a deterministic and a posterior DenseNet during training. Left: training from epoch 0 to epoch 220. Right: zoom on epochs 149-220 (beginning of posterior training). CIFAR10 Dataset.

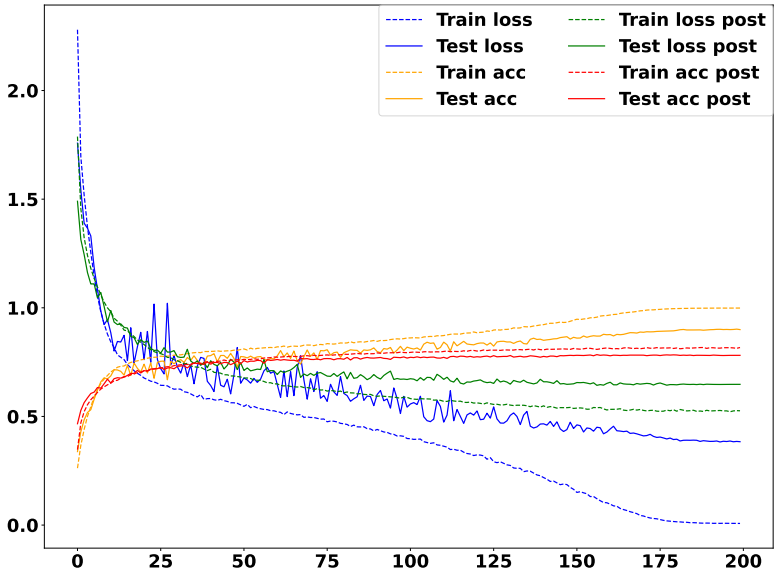


Figure S7: Comparison of a deterministic and a posterior DenseNet during training. The posterior network is trained from epoch 0 and the prior is updated every 5 epochs.

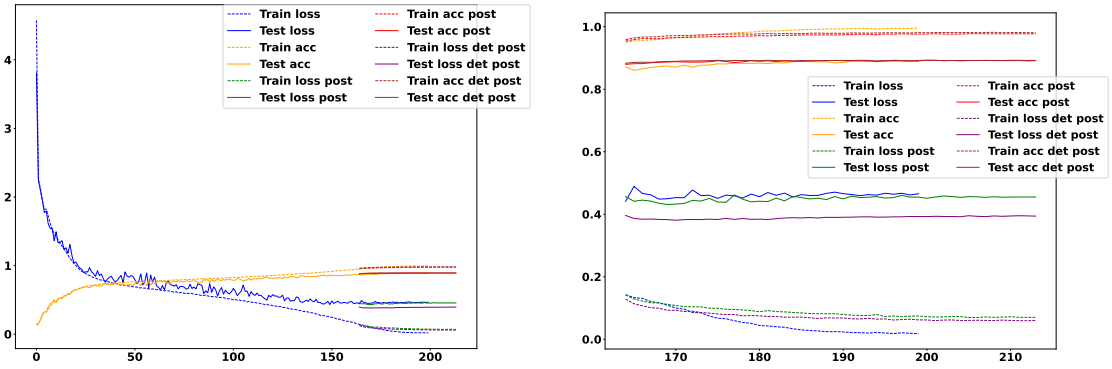


Figure S8: Comparison of a deterministic and a posterior Resnet during training. Left: training from epoch 0 to epoch 220. Right: zoom on epochs 164-220 (beginning of posterior training). CIFAR10 Dataset.

Bibliography of the current chapter

Bach, Francis and Lenaic Chizat (2021). *Gradient Descent on Infinitely Wide Neural Networks: Global Convergence and Generalization*. arXiv: 2110.08084 [cs.LG].

Bartlett, Peter L, Philip M Long, Gábor Lugosi, and Alexander Tsigler (2020). "Benign overfitting in linear regression". In: *Proceedings of the National Academy of Sciences* 117.48, pp. 30063–30070.

Belkin, Mikhail (2021). "Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation". In: *Acta Numerica* 30, pp. 203–248.

Belkin, Mikhail, Daniel Hsu, Siyuan Ma, and Soumik Mandal (2019a). "Reconciling modern machine-learning practice and the classical bias–variance trade-off". In: *Proceedings of the National Academy of Sciences* 116.32, pp. 15849–15854.

Chaudhari, Pratik et al. (2019). "Entropy-sgd: Biasing gradient descent into wide valleys". In: *Journal of Statistical Mechanics: Theory and Experiment* 2019.12, p. 124018.

Chizat, Lenaic and Francis Bach (2020). "Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss". In: *Conference on Learning Theory*. PMLR, pp. 1305–1338.

Clerico, Eugenio, George Deligiannidis, and Arnaud Doucet (2023). "Wide stochastic networks: Gaussian limit and PAC-Bayesian training". In: *International Conference on Algorithmic Learning Theory*. PMLR, pp. 447–470.

Dinh, Laurent, Razvan Pascanu, Samy Bengio, and Yoshua Bengio (2017). "Sharp minima can generalize for deep nets". In: *International Conference on Machine Learning*. PMLR, pp. 1019–1028.

Du, Jiawei et al. (2021). "Efficient sharpness-aware minimization for improved training of neural networks". In: *arXiv preprint arXiv:2110.03141*.

Dziugaite, Gintare Karolina and Daniel M Roy (2017). "Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data". In: *arXiv preprint arXiv:1703.11008*.

Foret, Pierre, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur (2020). "Sharpness-aware minimization for efficiently improving generalization". In: *arXiv preprint arXiv:2010.01412*.

Ghorbani, Behrooz, Shankar Krishnan, and Ying Xiao (2019). "An investigation into neural net optimization via hessian eigenvalue density". In: *International Conference on Machine Learning*. PMLR, pp. 2232–2241.

Gilmer, Justin et al. (2021). "A loss curvature perspective on training instability in deep learning". In: *arXiv preprint arXiv:2110.04369*.

Golmant, Noah, Zhewei Yao, Amir Gholami, Michael Mahoney, and Joseph Gonzalez (Oct. 2018). *pytorch-hessian-eigenthings: efficient PyTorch Hessian eigendecomposition*. Version 1.0. URL: https://github.com/noahgolmant/pytorch-hessian-eigenthings.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep learning*. MIT press.

He, Haowei, Gao Huang, and Yang Yuan (2019). "Asymmetric valleys: Beyond sharp and flat local minima". In: *Advances in neural information processing systems* 32.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Howard, Andrew G et al. (2017). "Mobilenets: Efficient convolutional neural networks for mobile vision applications". In: *arXiv preprint arXiv:1704.04861*.

Huang, Gao, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger (2017). "Densely connected convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708.

Izmailov, Pavel, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson (2018). "Averaging weights leads to wider optima and better generalization". In: *arXiv preprint arXiv:1803.05407*.

Jiang, Yiding, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio (2019). "Fantastic generalization measures and where to find them". In: *arXiv preprint arXiv:1912.02178*.

Keskar, Nitish Shirish, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang (2016). "On large-batch training for deep learning: Generalization gap and sharp minima". In: *arXiv preprint arXiv:1609.04836*.

Kwon, Jungmin, Jeongseop Kim, Hyunseo Park, and In Kwon Choi (2021). "ASAM: Adaptive Sharpness-Aware Minimization for Scale-Invariant Learning of Deep Neural Networks". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, pp. 5905–5914. URL: https://proceedings.mlr.press/v139/kwon21b.html.

Lan, Xinjie, Xin Guo, and Kenneth E Barner (2020). "PAC-Bayesian generalization bounds for multilayer perceptrons". In: *arXiv preprint arXiv:2006.08888*.

Letarte, Gaël, Pascal Germain, Benjamin Guedj, and François Laviolette (2019). "Dichotomize and generalize: PAC-Bayesian binary activated deep neural networks". In: *Advances in Neural Information Processing Systems* 32.

Liang, Tengyuan, Alexander Rakhlin, and Xiyu Zhai (2020b). "On the multiple descent of minimum-norm interpolants and restricted lower isometry of kernels". In: *Conference on Learning Theory*. PMLR, pp. 2683–2711.

McAllester, David A (1999). "PAC-Bayesian model averaging". In: *Proceedings of the twelfth annual conference on Computational learning theory*, pp. 164–170.

Pérez-Ortiz, María, Omar Rivasplata, John Shawe-Taylor, and Csaba Szepesvári (2021). "Tighter risk certificates for neural networks". In: *Journal of Machine Learning Research* 22.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov (2014). "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1, pp. 1929–1958.

Tsigler, Alexander and Peter L Bartlett (2020). "Benign overfitting in ridge regression". In: *arXiv preprint arXiv:2009.14286*.

Tsuzuku, Yusuke, Issei Sato, and Masashi Sugiyama (2020). "Normalized flat minima: Exploring scale invariant definition of flat minima for neural networks using pac-bayesian analysis". In: *International Conference on Machine Learning*. PMLR, pp. 9636–9647.

Zhou, Wenda, Victor Veitch, Morgane Austern, Ryan P Adams, and Peter Orbanz (2018). "Non-vacuous generalization bounds at the imagenet scale: a PAC-bayesian compression approach". In: *arXiv preprint arXiv:1804.05862*.

Bibliography

Akiba, Takuya, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama (2019). "Optuna: A Next-generation Hyperparameter Optimization Framework". In: *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Alquier, Pierre (2023). *User-friendly introduction to PAC-Bayes bounds*. arXiv: 2110 . 11216 [stat.ML].

Arik, Sercan Ö and Tomas Pfister (2021). "Tabnet: Attentive interpretable tabular learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 8, pp. 6679–6687.

Arlot, Sylvain and Robin Genuer (2014). "Analysis of purely random forests bias". In: *arXiv preprint arXiv:1407.3939*.

Arnould, Ludovic, Claire Boyer, and Erwan Scornet (2021). "Analyzing the tree-layer structure of Deep Forests". In: *International Conference on Machine Learning*. PMLR, pp. 342–350.

— (2023). "Is interpolation benign for random forest regression?" In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 5493–5548.

Bach, Francis and Lenaic Chizat (2021). *Gradient Descent on Infinitely Wide Neural Networks: Global Convergence and Generalization*. arXiv: 2110.08084 [cs.LG].

Barron, Andrew R (1994). "Approximation and estimation bounds for artificial neural networks". In: *Machine learning* 14, pp. 115–133.

Bartlett, Peter L, Philip M Long, Gábor Lugosi, and Alexander Tsigler (2020). "Benign overfitting in linear regression". In: *Proceedings of the National Academy of Sciences* 117.48, pp. 30063–30070.

Bartlett, Peter L, Andrea Montanari, and Alexander Rakhlin (2021). "Deep learning: a statistical viewpoint". In: *arXiv preprint arXiv:2103.09177*.

Batir, Necdet (2008). "Inequalities for the gamma function". In: *Archiv der Mathematik* 91.6, pp. 554–563.

Belgiu, Mariana and Lucian Drăguţ (2016). "Random forest in remote sensing: A review of applications and future directions". In: *ISPRS journal of photogrammetry and remote sensing* 114, pp. 24–31.

Belkin, Mikhail (2021). "Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation". In: *Acta Numerica* 30, pp. 203–248.

Belkin, Mikhail, Daniel Hsu, Siyuan Ma, and Soumik Mandal (2019a). "Reconciling modern machine-learning practice and the classical bias–variance trade-off". In: *Proceedings of the National Academy of Sciences* 116.32, pp. 15849–15854.

Belkin, Mikhail, Daniel Hsu, and Partha Mitra (2018). "Overfitting or perfect fitting? risk bounds for classification and regression rules that interpolate". In: *arXiv preprint arXiv:1806.05161*.

Belkin, Mikhail, Alexander Rakhlin, and Alexandre B Tsybakov (2019b). "Does data interpolation contradict statistical optimality?" In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 1611–1619.

Bergstra, J. S., R. Bardenet, Y. Bengio, and K. Balázs (2011). "Algorithms for Hyper-Parameter Optimization". In: *Advances in Neural Information Processing Systems 24*. Ed. by J. Shawe-Taylor,

R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger. Curran Associates, Inc., pp. 2546–2554.

Berrouachedi, A., R. Jaziri, and G. Bernard (2019a). "Deep Cascade of Extra Trees". In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, pp. 117–129.

— (2019b). "Deep Extremely Randomized Trees". In: *International Conference on Neural Information Processing*. Springer, pp. 717–729.

Biau, G. (2012a). "Analysis of a random forests model". In: *The Journal of Machine Learning Research* 13.1, pp. 1063–1095.

Biau, G., L. Devroye, and G. Lugosi (2008). "Consistency of random forests and other averaging classifiers". In: *Journal of Machine Learning Research* 9.Sep, pp. 2015–2033.

Biau, Gérard (2012b). "Analysis of a random forests model". In: *The Journal of Machine Learning Research* 13, pp. 1063–1095.

Biau, Gérard and Erwan Scornet (2016). "A random forest guided tour". In: *Test* 25.2, pp. 197–227.

Biau, Gérard, Erwan Scornet, and Johannes Welbl (2019). "Neural random forests". In: *Sankhya A* 81.2, pp. 347–386.

Blundell, Charles, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra (2015). "Weight uncertainty in neural network". In: *International conference on machine learning*. PMLR, pp. 1613–1622.

Borisov, Vadim, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci (2021). *Deep Neural Networks and Tabular Data: A Survey*. DOI: 10.48550/ARXIV.2110.01889. URL: https://arxiv.org/abs/2110.01889.

Bottou, Léon, Frank E Curtis, and Jorge Nocedal (2018). "Optimization methods for large-scale machine learning". In: *SIAM review* 60.2, pp. 223–311.

Breiman, Leo (1996). "Bagging predictors". In: *Machine learning* 24, pp. 123–140.

— (2001a). "Random forests". In: *Machine learning* 45.1, pp. 5–32.

— (2001b). "Random Forests". In: *Machine Learning* 45.1, pp. 5–32. ISSN: 1573-0565. DOI: 10.1023/A:1010933404324. URL: https://doi.org/10.1023/A:1010933404324.

— (2004). "Consistency for a simple model of random forests". In: *University of California at Berkeley. Technical Report* 670.

Breiman, Leo, Jerome Friedman, Charles J Stone, and Richard A Olshen (1984). *Classification and regression trees*. CRC press.

Brent, Richard P (1991). "Fast training algorithms for multilayer neural nets". In: *IEEE Transactions on Neural Networks* 2.3, pp. 346–354.

Bühlmann, Peter and Bin Yu (2002). "Analyzing bagging". In: *The annals of Statistics* 30.4, pp. 927–961.

Buschjäger, Sebastian and Katharina Morik (2021). "There is no Double-Descent in Random Forests". In: *arXiv preprint arXiv:2111.04409*.

Chaudhari, Pratik et al. (2019). "Entropy-sgd: Biasing gradient descent into wide valleys". In: *Journal of Statistical Mechanics: Theory and Experiment* 2019.12, p. 124018.

Chen, Tianqi and Carlos Guestrin (2016). "Xgboost: A scalable tree boosting system". In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794.

Chen, Xi and Hemant Ishwaran (2012). "Random forests for genomic data analysis". In: *Genomics* 99.6, pp. 323–329.

Chi, Chien-Ming, Patrick Vossler, Yingying Fan, and Jinchi Lv (2022). "Asymptotic properties of high-dimensional random forests". In: *The Annals of Statistics* 50.6, pp. 3415–3438.

Chizat, Lenaic and Francis Bach (2020). "Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss". In: *Conference on Learning Theory*. PMLR, pp. 1305–1338.

Clerico, Eugenio, George Deligiannidis, and Arnaud Doucet (2023). "Wide stochastic networks: Gaussian limit and PAC-Bayesian training". In: *International Conference on Algorithmic Learning Theory*. PMLR, pp. 447–470.

Cribari-Neto, F., N. L. Garcia, and K. LP Vasconcellos (2000). "A note on inverse moments of binomial variates". In: *Brazilian Review of Econometrics* 20.2, pp. 269–277.

Criminisi, Antonio, Jamie Shotton, Ender Konukoglu, et al. (2012). "Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning". In: *Foundations and trends® in computer graphics and vision* 7.2–3, pp. 81–227.

Cybenko, G. (Dec. 1989). "Approximation by superpositions of a sigmoidal function". In: *Mathematics of Control, Signals and Systems* 2.4, pp. 303–314. ISSN: 1435-568X. DOI: 10.1007/BF02551274. URL: https://doi.org/10.1007/BF02551274.

Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei (2009). "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee, pp. 248–255.

Devroye, Luc, Laszlo Györfi, and Adam Krzyżak (1998). "The Hilbert kernel regression estimate". In: *Journal of Multivariate Analysis* 65.2, pp. 209–227.

Díaz-Uriarte, Ramón and Sara Alvarez de Andrés (2006). "Gene selection and classification of microarray data using random forest". In: *BMC bioinformatics* 7, pp. 1–13.

Dinh, Laurent, Razvan Pascanu, Samy Bengio, and Yoshua Bengio (2017). "Sharp minima can generalize for deep nets". In: *International Conference on Machine Learning*. PMLR, pp. 1019–1028.

Du, Jiawei et al. (2021). "Efficient sharpness-aware minimization for improved training of neural networks". In: *arXiv preprint arXiv:2110.03141*.

Duroux, Roxane and Erwan Scornet (2018). "Impact of subsampling and tree depth on random forests". In: *ESAIM: Probability and Statistics* 22, pp. 96–128.

Dziugaite, Gintare Karolina and Daniel M Roy (2017). "Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data". In: *arXiv preprint arXiv:1703.11008*.

Elie-Dit-Cosaque, Kevin and Véronique Maume-Deschamps (2022). "Random forest estimation of conditional distribution functions and conditional quantiles". In: *Electronic Journal of Statistics* 16.2, pp. 6553–6583.

Fan, Wei, Haixun Wang, Philip S Yu, and Sheng Ma (2003). "Is random model better? on its accuracy and efficiency". In: *Third IEEE International Conference on Data Mining*. IEEE, pp. 51–58.

Feng, Ji, Yang Yu, and Zhi-Hua Zhou (2018). "Multi-layered gradient boosting decision trees". In: *Advances in neural information processing systems*, pp. 3551–3561.

Foret, Pierre, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur (2020). "Sharpness-aware minimization for efficiently improving generalization". In: *arXiv preprint arXiv:2010.01412*.

Frankle, Jonathan and Michael Carbin (2018). "The lottery ticket hypothesis: Finding sparse, trainable neural networks". In: *arXiv preprint arXiv:1803.03635*.

Friedman, Jerome H (2001). "Greedy function approximation: a gradient boosting machine". In: *Annals of statistics*, pp. 1189–1232.

Genuer, Robin, Jean-Michel Poggi, and Christine Tuleau-Malot (2010). "Variable selection using random forests". In: *Pattern recognition letters* 31.14, pp. 2225–2236.

Geurts, P., D. Ernst, and L. Wehenkel (2006). "Extremely randomized trees". In: *Machine learning* 63.1, pp. 3–42.

Ghods, Alireza and Diane J Cook (2020). "A survey of deep network techniques all classifiers can adopt". In: *Data Mining and Knowledge Discovery*, pp. 1–42.

Ghorbani, Behrooz, Shankar Krishnan, and Ying Xiao (2019). "An investigation into neural net optimization via hessian eigenvalue density". In: *International Conference on Machine Learning*. PMLR, pp. 2232–2241.

Ghosh, Soumyadip and Shane G Henderson (2002). "Chessboard distributions and random vectors with specified marginals and covariance matrix". In: *Operations Research* 50.5, pp. 820–834.

— (2009). "Patchwork distributions". In: *Advancing the Frontiers of Simulation*. Springer, pp. 65–86.

Gilmer, Justin et al. (2021). "A loss curvature perspective on training instability in deep learning". In: *arXiv preprint arXiv:2110.04369*.

Glorot, Xavier and Yoshua Bengio (2010). "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, pp. 249–256.

Golmant, Noah, Zhewei Yao, Amir Gholami, Michael Mahoney, and Joseph Gonzalez (Oct. 2018). *pytorch-hessian-eigenthings: efficient PyTorch Hessian eigendecomposition*. Version 1.0. URL: https://github.com/noahgolmant/pytorch-hessian-eigenthings.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep learning*. MIT press.

Gorishniy, Yury, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko (2021). *Revisiting Deep Learning Models for Tabular Data*. DOI: 10.48550/ARXIV.2106.11959. URL: https://arxiv.org/abs/2106.11959.

Grinsztajn, Léo, Edouard Oyallon, and Gaël Varoquaux (2022). "Why do tree-based models still outperform deep learning on tabular data?" In: *arXiv preprint arXiv:2207.08815*.

Guo, Yang, Shuhui Liu, Zhanhuai Li, and Xuequn Shang (2018). "BCDForest: a boosting cascade deep forest model towards the classification of cancer subtypes based on gene expression data". In: *BMC bioinformatics* 19.5, p. 118.

Hastie, Trevor, Andrea Montanari, Saharon Rosset, and Ryan J. Tibshirani (2022). "Surprises in high-dimensional ridgeless least squares interpolation". In: *The Annals of Statistics* 50.2, pp. 949 –986. DOI: 10.1214/21-AOS2133. URL: https://doi.org/10.1214/21-AOS2133.

He, Haowei, Gao Huang, and Yang Yuan (2019). "Asymmetric valleys: Beyond sharp and flat local minima". In: *Advances in neural information processing systems* 32.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2015). "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". In: *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034.

— (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Hornik, Kurt, Maxwell Stinchcombe, and Halbert White (1989). "Multilayer feedforward networks are universal approximators". In: *Neural networks* 2.5, pp. 359–366.

Howard, Andrew G et al. (2017). "Mobilenets: Efficient convolutional neural networks for mobile vision applications". In: *arXiv preprint arXiv:1704.04861*.

Huang, Gao, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger (2017). "Densely connected convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708.

Huang, W Ronny et al. (2020). "Understanding generalization through visualizations". In.

Ishwaran, Hemant (2015). "The effect of splitting on random forests". In: *Machine learning* 99.1, pp. 75–118.

Izmailov, Pavel, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson (2018). "Averaging weights leads to wider optima and better generalization". In: *arXiv preprint arXiv:1803.05407*.

Jeong, Mira, Jaeyeal Nam, and Byoung Chul Ko (2020). "Lightweight Multilayer Random Forests for Monitoring Driver Emotional Status". In: *IEEE Access* 8, pp. 60344–60354.

Jiang, Yiding, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio (2019). "Fantastic generalization measures and where to find them". In: *arXiv preprint arXiv:1912.02178*.

Kainen, Paul C. (1997). "Utilizing Geometric Anomalies of High Dimension: When Complexity Makes Computation Easier". In: *Computer Intensive Methods in Control and Signal Processing: The Curse of Dimensionality*. Ed. by Miroslav Kárný and Kevin Warwick. Boston, MA: Birkhäuser Boston, pp. 283–294. ISBN: 978-1-4612-1996-5. DOI: 10.1007/978-1-4612-1996-5_18. URL: https://doi.org/10.1007/978-1-4612-1996-5_18.

Ke, Guolin, Jia Zhang, Zhenhui Xu, Jiang Bian, and Tie-Yan Liu (2018). "TabNN: A universal neural network solution for tabular data". In.

Ke, Guolin et al. (2017). "Lightgbm: A highly efficient gradient boosting decision tree". In: *Advances in neural information processing systems* 30.

Keskar, Nitish Shirish, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang (2016). "On large-batch training for deep learning: Generalization gap and sharp minima". In: *arXiv preprint arXiv:1609.04836*.

Kim, S., M. Jeong, and B. C. Ko (2020). "Interpretation and Simplification of Deep Forest". In: *arXiv preprint arXiv:2001.04721*.

Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

Klusowski, Jason M. (2018). "Sharp analysis of a simple model for random forests". In: *arXiv preprint arXiv:1805.02587*.

— (2021a). "Sharp analysis of a simple model for random forests". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 757–765.

— (2021b). "Universal consistency of decision trees in high dimensions". In: *arXiv preprint arXiv:2104.13881*.

Kobak, Dmitry, Jonathan Lomond, and Benoit Sanchez (2020). "The optimal ridge penalty for real-world high-dimensional data can be zero or negative due to the implicit ridge regularization". In: *The Journal of Machine Learning Research* 21.1, pp. 6863–6878.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25.

Kwon, Jungmin, Jeongseop Kim, Hyunseo Park, and In Kwon Choi (2021). "ASAM: Adaptive Sharpness-Aware Minimization for Scale-Invariant Learning of Deep Neural Networks". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, pp. 5905–5914. URL: https://proceedings.mlr.press/v139/kwon21b.html.

Laan, Mark J Van der, Eric C Polley, and Alan E Hubbard (2007). "Super learner". In: *Statistical applications in genetics and molecular biology* 6.1.

Lakshminarayanan, Balaji, Daniel M Roy, and Yee Whye Teh (2014). "Mondrian forests: Efficient online random forests". In: *Advances in neural information processing systems* 27.

Lan, Xinjie, Xin Guo, and Kenneth E Barner (2020). "PAC-Bayesian generalization bounds for multilayer perceptrons". In: *arXiv preprint arXiv:2006.08888*.

LeCun, Yann, Yoshua Bengio, et al. (1995). "Convolutional networks for images, speech, and time series". In: *The handbook of brain theory and neural networks* 3361.10, p. 1995.

Letarte, Gaël, Pascal Germain, Benjamin Guedj, and François Laviolette (2019). "Dichotomize and generalize: PAC-Bayesian binary activated deep neural networks". In: *Advances in Neural Information Processing Systems* 32.

Liang, Tengyuan and Alexander Rakhlin (2020a). "Just interpolate: Kernel "ridgeless" regression can generalize". In.

Liang, Tengyuan, Alexander Rakhlin, and Xiyu Zhai (2020b). "On the multiple descent of minimum-norm interpolants and restricted lower isometry of kernels". In: *Conference on Learning Theory*. PMLR, pp. 2683–2711.

Lin, Yi and Yongho Jeon (2006). "Random forests and adaptive nearest neighbors". In: *Journal of the American Statistical Association* 101.474, pp. 578–590.

Liu, B. et al. (2020). "Morphological Attribute Profile Cube and Deep Random Forest for Small Sample Classification of Hyperspectral Image". In: *IEEE Access* 8, pp. 117096–117108.

Liu, Pengfei, Xipeng Qiu, and Xuanjing Huang (2016). "Recurrent neural network for text classification with multi-task learning". In: *arXiv preprint arXiv:1605.05101*.

Lugosi, Gábor and Andrew Nobel (1996). "Consistency of data-driven histogram methods for density estimation and classification". In: *The Annals of Statistics* 24.2, pp. 687–706.

Lutz, Patrick, Ludovic Arnould, Claire Boyer, and Erwan Scornet (2022). "Sparse tree-based initialization for neural networks". In: *arXiv preprint arXiv:2209.15283*.

McAllester, David A (1999). "PAC-Bayesian model averaging". In: *Proceedings of the twelfth annual conference on Computational learning theory*, pp. 164–170.

Mentch, Lucas and Siyu Zhou (2019). "Randomization as regularization: a degrees of freedom explanation for random forest success". In: *arXiv preprint arXiv:1911.00190*.

— (2022). "Getting better from worse: Augmented bagging and a cautionary tale of variable importance". In: *Journal of Machine Learning Research* 23.224, pp. 1–32.

Miller, Kevin, Chris Hettinger, Jeffrey Humpherys, Tyler Jarvis, and David Kartchner (May 2017). "Forward Thinking: Building Deep Random Forests". In.

Mishkin, Dmytro and Jiri Matas (2015). "All you need is a good init". In: *arXiv preprint arXiv:1511.06422*.

Mourtada, Jaouad, Stéphane Gaïffas, and Erwan Scornet (2020). "Minimax optimal rates for Mondrian trees and forests". In: *The Annals of Statistics* 48.4, pp. 2253–2276.

Nakkiran, Preetum, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever (2021). "Deep double descent: Where bigger models and more data hurt". In: *Journal of Statistical Mechanics: Theory and Experiment* 2021.12, p. 124003.

Neal, Radford M (2012). *Bayesian learning for neural networks*. Vol. 118. Springer Science & Business Media.

Nobel, Andrew (1996). "Histogram regression estimation using data-dependent partitions". In: *The Annals of Statistics* 24.3, pp. 1084–1105.

Pang, M., K. Ting, P. Zhao, and Z. Zhou (2018). "Improving Deep Forest by Confidence Screening". In: *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 1194–1199.

Paszke, Adam et al. (2019). "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., pp. 8024–8035.

Pedregosa, F. et al. (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Perez-Ortiz, Maria, Omar Rivasplata, Emilio Parrado-Hernandez, Benjamin Guedj, and John Shawe-Taylor (2021). "Progress in Self-Certified Neural Networks". In: *arXiv preprint arXiv:2111.07737*.

Pérez-Ortiz, María, Omar Rivasplata, John Shawe-Taylor, and Csaba Szepesvári (2021). "Tighter risk certificates for neural networks". In: *Journal of Machine Learning Research* 22.

Petersen, Philipp Christian (2020). "Neural network theory". In: *University of Vienna*.

Pinkus, Allan (1999). "Approximation theory of the MLP model in neural networks". In: *Acta numerica* 8, pp. 143–195.

Prasad, Anantha M, Louis R Iverson, and Andy Liaw (2006). "Newer classification and regression tree techniques: bagging and random forests for ecological prediction". In: *Ecosystems* 9, pp. 181–199.

Rakhlin, Alexander and Xiyu Zhai (2019). "Consistency of interpolation with Laplace kernels is a high-dimensional phenomenon". In: *Conference on Learning Theory*. PMLR, pp. 2595–2623.

Richmond, David L, Dagmar Kainmueller, Michael Y Yang, Eugene W Myers, and Carsten Rother (2015). "Relating cascaded random forests to deep convolutional neural networks for semantic segmentation". In: *arXiv preprint arXiv:1507.07583*.

Richmond, Lawrence Bruce and Jeffrey Shallit (2009). "Counting abelian squares". In: *Electronic Journal of Combinatorics*.

Ruder, Sebastian (2016). "An overview of gradient descent optimization algorithms". In: *arXiv preprint arXiv:1609.04747*.

Rumelhart, David E, Geoffrey E Hinton, James L McClelland, et al. (1986). "A general framework for parallel distributed processing". In: *Parallel distributed processing: Explorations in the microstructure of cognition* 1.45-76, p. 26.

Rumelhart, David E, Geoffrey E Hinton, and Ronald J Williams (1985). *Learning internal representations by error propagation*. Tech. rep. California Univ San Diego La Jolla Inst for Cognitive Science.

Scornet, Erwan (2016a). "On the asymptotics of random forests". In: *Journal of Multivariate Analysis* 146, pp. 72–83.

— (2016b). "Random forests and kernel methods". In: *IEEE Transactions on Information Theory* 62.3, pp. 1485–1500.

Scornet, Erwan, Gérard Biau, and Jean-Philippe Vert (2015). "Consistency of random forests". In: *The Annals of Statistics* 43.4, pp. 1716–1741.

Sethi, Ishwar Krishnan (1990). "Entropy nets: from decision trees to neural networks". In: *Proceedings of the IEEE* 78.10, pp. 1605–1613.

Shrestha, Ajay and Ausif Mahmood (2019). "Review of deep learning algorithms and architectures". In: *IEEE access* 7, pp. 53040–53065.

Shwartz-Ziv, Ravid and Amitai Armon (2022). "Tabular data: Deep learning is not all you need". In: *Information Fusion* 81, pp. 84–90. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2021.11.011. URL: https://www.sciencedirect.com/science/article/pii/S1566253521002360.

Somepalli, Gowthami, Micah Goldblum, Avi Schwarzschild, C. Bayan Bruss, and Tom Goldstein (2021). *SAINT: Improved Neural Networks for Tabular Data via Row Attention and Contrastive Pre-Training*. DOI: 10.48550/ARXIV.2106.01342. URL: https://arxiv.org/abs/2106.01342.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov (2014). "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1, pp. 1929–1958.

Su, R., X. Liu, L. Wei, and Q. Zou (2019). "Deep-Resp-Forest: A deep forest model to predict anti-cancer drug response". In: *Methods* 166, pp. 91–102.

Sun, L. et al. (2020). "Adaptive Feature Selection Guided Deep Forest for COVID-19 Classification With Chest CT". In: *IEEE Journal of Biomedical and Health Informatics* 24.10, pp. 2798–2805.

Sun, Ruo-Yu (2020). "Optimization for deep learning: An overview". In: *Journal of the Operations Research Society of China* 8.2, pp. 249–294.

Tan, Zhiqiang and Cun-Hui Zhang (2019). "Doubly penalized estimation in additive regression with high-dimensional data". In.

Tang, Cheng, Damien Garreau, and Ulrike von Luxburg (2018). "When do random forests fail?" In: *Advances in neural information processing systems* 31.

Tsigler, Alexander and Peter L Bartlett (2020). "Benign overfitting in ridge regression". In: *arXiv preprint arXiv:2009.14286*.

Tsuzuku, Yusuke, Issei Sato, and Masashi Sugiyama (2020). "Normalized flat minima: Exploring scale invariant definition of flat minima for neural networks using pac-bayesian analysis". In: *International Conference on Machine Learning*. PMLR, pp. 9636–9647.

Utkin, L. V and M. A Ryabinin (2017). "Discriminative metric learning with deep forest". In: *arXiv preprint arXiv:1705.09620*.

Utkin, L. V and K. D Zhuk (2020). "Improvement of the Deep Forest Classifier by a Set of Neural Networks". In: *Informatica* 44.1.

Vaswani, Ashish et al. (2017). "Attention is all you need". In: *arXiv preprint arXiv:1706.03762*.

Wager, Stefan and Guenther Walther (2015). "Adaptive concentration of regression trees, with application to random forests". In: *arXiv preprint arXiv:1503.06388*.

Wang, Yutong and Clayton D Scott (2022). "Consistent Interpolating Ensembles via the Manifold-Hilbert Kernel". In: *arXiv preprint arXiv:2205.09342*.

Welbl, Johannes (2014). "Casting random forests as artificial neural networks (and profiting from it)". In: *German Conference on Pattern Recognition*. Springer, pp. 765–771.

Wyner, Abraham J, Matthew Olson, Justin Bleich, and David Mease (2017). "Explaining the success of adaboost and random forests as interpolating classifiers". In: *The Journal of Machine Learning Research* 18.1, pp. 1558–1590.

Zeng, X. et al. (2020). "Network-based prediction of drug–target interactions using an arbitrary-order proximity embedded deep forest". In: *Bioinformatics* 36.9, pp. 2805–2812.

Zhang, Y. et al. (2019). "Distributed deep forest and its application to automatic detection of cash-out fraud". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 10.5, pp. 1–19.

Zheng, S., Y. Song, T. Leung, and I. Goodfellow (2016). "Improving the robustness of deep neural networks via stability training". In: *Proceedings of the ieee conference on computer vision and pattern recognition*, pp. 4480–4488.

Zhou, Siyu and Lucas Mentch (2021). "Trees, forests, chickens, and eggs: when and why to prune trees in a random forest". In: *arXiv preprint arXiv:2103.16700*.

Zhou, Wenda, Victor Veitch, Morgane Austern, Ryan P Adams, and Peter Orbanz (2018). "Non-vacuous generalization bounds at the imagenet scale: a PAC-bayesian compression approach". In: *arXiv preprint arXiv:1804.05862*.

Zhou, Z and J. Feng (2017). "Deep Forest: Towards An Alternative to Deep Neural Networks". In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 3553–3559.

Zhou, Zhi-Hua and Ji Feng (Jan. 2019). "Deep forest". In: *National Science Review* 6, pp. 74–86. DOI: 10.1093/nsr/nwy108.

Zhuang, Fuzhen et al. (2020). "A comprehensive survey on transfer learning". In: *Proceedings of the IEEE* 109.1, pp. 43–76.