Learning deep kernel networks : application to efficient and robust structured prediction

Tamim El Ahmad

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2024IPPAT024

Thèse de doctorat

Learning Deep Kernel Networks: Application to Efficient and Robust Structured Prediction

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Télécom Paris

École doctorale n°626 Ecole Doctorale de l'Institut Polytechnique de Paris (ED IP Paris)
Spécialité de doctorat : Informatique, données, IA

Thèse présentée et soutenue à Palaiseau, le 09/07/2024, par

TAMIM EL AHMAD

Composition du Jury :

Stephan Clémençon Professor, Télécom Paris	Président/Examinateur
Hachem Kadri Associate Professor, Université Aix-Marseille	Rapporteur
Bharath Sriperumbudur Associate Professor, Pennsylvania State University	Rapporteur
Julien Mairal Senior Researcher, Inria Grenoble	Examinateur
Anna Korba Assistant Professor, ENSAE Paris	Examinatrice
Zoltán Szabó Professor, London School of Economics	Examinateur
Florence d'Alché-Buc Professor, Télécom Paris	Directrice de thèse
Pierre Laforgue Quantitative Researcher, Capital Fund Management	Co-encadrant de thèse
Céline Brouard Researcher, INRAE Toulouse	Invitée

Contents

Remerciements

Je souhaite commencer par remercier celles et ceux qui m'ont accompagné et soutenu durant ces années de doctorat.

Merci tout d'abord à Florence, pour la confiance que tu m'as accordé en me recrutant en thèse après un stage difficile en pleine période COVID, j'ai énormément appris durant cette thèse grâce à toi. Je tiens également à te remercier pour la détermination dont tu as fait preuve dans des moments qui ont été difficiles pour toi. J'en profite pour te remercier d'avoir proposé à Pierre de m'encadrer, et à toi Pierre d'avoir accepté, c'est peut-être la meilleure décision prise durant cette thèse ! Je me souviendrai pendant longtemps (toujours ?) de cette première deadline ICML, tu as toujours été de bon conseil et d'un soutien indispensable. Je suis ravi d'avoir été encadré par un ami.

I would like to thank Hachem Kadri and Bharath Sriperumbudur for their very thorough review of this manuscript. I also thank Zoltán Szabó, Julien Mairal, Anna Korba, and Stephan Clémençon for being part of my PhD committee. Many thanks also go to Céline Brouard for accepting the invitation to my defense.

Merci à Luc et Junjie, cette thèse n'aurait clairement pas été possible sans vos expertises sans faille en théorie des noyaux et en réseaux de neurones. Travailler avec vous a été un réel plaisir et tout comme avec Pierre, je suis ravi d'avoir travaillé avec des amis. Courage à Junjie pour tes derniers mois en thèse, je ne me fais pas de souci vu la ténacité que tu as déployé pendant notre projet qui n'était vraiment pas facile.

Merci à Emilia, Marc et Jr, je suis à la fois très heureux de voir le bout de cette thèse, mais déjà nostalgique de tous nos moments passés ensemble. Je suis de tout coeur avec vous, Emilia et Jr, pour vos derniers mois de thèse, vous allez y arriver ! Vous êtes, tous.tes les trois, les plus belles rencontres que j'ai faites pendant cette thèse et ça n'a pas de prix !

Merci à tous les gens du labo, ancien.ne.s comme nouvelles.aux, pour toutes les discussions diverses et variées, et moments passés ensemble: Dimitri, Iyad, Ikhlas, Mathilde, Nathan, Anass, Jérémy, Joël, Quentin, Lilian, Arturo... Pour celles et ceux qui n'ont pas encore soutenu: courage, ça va le faire !

Merci à mes ami.e.s que j'aime de tout mon coeur, la famille que j'ai choisie.

Merci à ma nièce Nayla et mon neveu Abdallah, je vous souhaite de vous épanouir et je suis sûr que vous y arriverez avec Raja et Chadi. Tous ces week-ends passés en votre compagnie à Munich étaient un réel plaisir. Merci à toute ma famille au Liban, j'espère qu'on pourra bientôt retourner chez nous dans le sud. J'ai une pensée pour ma grand-mère, qu'elle repose en paix.

Enfin merci à Khaled, Nour, Papa et Maman, sans vous je ne serais rien, cette thèse est autant la mienne que la vôtre.

Les derniers mois de cette thèse ont été marqués par une tragédie en Palestine, que peu de mots sont en mesure de qualifier, et qui n'a toujours pas atteint son terme au moment où j'écris ces lignes. J'en profite donc pour t'imiter, Papa, et je dédie cette thèse à tous les enfants du Liban et de Palestine, faites que votre avenir s'éclaircisse.

Abstract

The task of predicting structured objects, e.g. graphs or sequences, is more demanding than the standard supervised regression or classification problems, where the outputs are usually low-dimensional vectors. It has recently attracted a lot of attention in various fields, such as computational biology and chemistry. Such structured spaces are usually high-dimensional, discrete, large, and lack of linear structure, which makes it difficult to design a versatile model, i.e. a model able to deal with various output types within a unified framework, together with strong theoretical foundations.

In this thesis, we focus on surrogate kernel methods, and in particular Input Output Kernel Regression, a versatile and theoretically-funded structured prediction approach leveraging the kernel trick in both the input and output spaces. However, in practice, this method exhibits some flaws. As with other kernel-based methods, IOKR suffers from computational burdens at both the training and inference phases. Moreover, it benefits from a closed-form solution when combined with the squared loss, and it is challenging to employ a wider variety of losses. Finally, it is not efficient in handling complex inputs such as images or texts. Our goal is then to design an OKR model that is: scalable to large datasets, theoretically sound (i.e. for which excess risk bounds can be derived), compatible with a wider variety of losses, and able to learn representations from complex inputs.

In the first part of this thesis, we focus on the input kernel, and introduce a new sub-Gaussian sketching distribution, called the *p-sparsified sketches*, in order to scale-up matrix-valued decomposable kernel machines with generic Lipschitz-continuous losses. Sketching consists in manipulating random linear projections to reduce computational complexity while maintaining good statistical performance. We additionally provide an excess risk bound of the estimator induced by this approach.

In the second part, we introduce *Sketched Input Sketched Output Kernel Regression*, an IOKR-based method that leverages sketching on both the input and output kernels to induce a reduced-rank structured estimator. We derive its excess risk bound with sub-Gaussian or sub-sampling input/output sketches and show that it attains close-to-optimal learning rates. Besides, we demonstrate the strong empirical performance of SISOKR on datasets on which IOKR is intractable.

In the last part, we apply sketching on the output kernel and introduce a deep neural architecture able to predict within the possibly infinite-dimensional output kernel's feature space. Indeed, we compute the basis induced by the eigenfunctions of the sketched output empirical covariance operator, and *Deep Sketched Output Kernel Regression*'s neural network then computes an expansion within this basis and learns its coordinates during training. This unlocks the use of gradient-based methods for

any loss which is the composition of the square loss with a sub-differentiable function, such as standard robust losses, and any neural architectures, such as transformers. Empirical validations of the approach are provided, in particular on a text-to-molecule dataset.

Résumé

La prédiction d'objets structurés, tels que les graphes ou les séquences par exemple, est plus exigeante que les problèmes standards de régressions ou de classification supervisés, dans lesquels les sorties sont généralement des vecteurs de petite dimension. Cette tâche fait l'objet de beaucoup d'attention dans différents domaines, comme la biologie ou la chimie informatique. Les espaces structurés sont en général de grande dimension, discrets, et non-linéaires, ce qui complique la conceptualisation d'un modèle polyvalent, autrement dit un modèle capable de gérer différents types de sorties dans un cadre unifié, tout en bénéficiant de solides fondations théoriques.

Dans cette thèse, nous nous concentrons sur les méthodes à noyaux de substitution, et en particulier à la méthode *Input Output Kernel Regression* (*IOKR*), une approche de prédiction structurée polyvalente et théoriquement fondée utilisant l'astuce du noyau sur les espaces d'entrée et de sortie. Toutefois, cette méthode présente plusieurs limites. En premier lieu, elle souffre de lourds coûts de calcul pendant les phases d'apprentissage et de prédiction. De plus, il n'est pas évident d'utiliser d'autres fonctions de perte que la quadratique (qui lui permet de bénéficier d'une solution explicite). Enfin, elle est confrontée à l'incapacité des noyaux à apprendre des représentations à partir de données d'entrée complexes comme des images ou du texte, contrairement aux réseaux de neurones profonds. Notre objectif est donc de concevoir un modèle utilisant un noyau de sortie passant à l'échelle de grandes bases de données, avec une borne sur son excès de risque, compatible avec une plus grande variété de fonctions de perte et capable d'apprendre des représentations à partir de données d'entrée complexes grâce à l'utilisation de réseaux de neurones profonds.

Tout d'abord, nous travaillons sur le noyau d'entrée, et introduisons une nouvelle distribution de projections aléatoires sous-gaussienne, les *p-sparsified sketches*, afin de passer à l'échelle les machines à noyau matriciel décomposable utilisant des fonctions de perte lipschitziennes. Ces projections aléatoires sont linéaires et vont au-delà du sous-échantillonnage, largement étudié au sein de la littérature des méthodes à noyaux. L'objectif de cette distribution est d'atteindre un équilibre optimal entre l'efficacité calculatoire du sous-échantillonnage et les bonnes performances statistiques des projections gaussiennes. De plus, nous fournissons une borne d'excès de risque de l'estimateur induit par cette approche dans le cadre de la régression à sorties multiples, tout en considérant des fonctions de perte lipschitziennes.

En outre, nous introduisons *Sketched Input Sketched Output Kernel Regression* (*SISOKR*), une méthode basée sur *IOKR* et tirant profit des projections aléatoires sur les noyaux d'entrée et de sortie pour obtenir un estimateur structuré de rang faible. Étant donné la dimension potentiellement infinie de l'espace de représentation de sortie et la fonction de perte quadratique, les projections aléatoires nous permettent ici de

construire des opérateurs de projection orthogonale vers des sous-espaces de dimensions réduites des espaces de représentation d'entrée et de sortie. Nous prouvons une borne d'excès de risque de cet estimateur utilisant des projections aléatoires entrée/sortie sous-gaussiennes ou de sous-échantillonnage et montrons qu'il atteint une vitesse d'apprentissage proche de l'optimal. En particulier, la conclusion de cette étude théorique est cohérente : si nous faisons face à un problème sous-jacent de rang faible (forte décroissance des valeurs propres des opérateurs de covariance entrée/sortie), nous pouvons utiliser un estimateur de rang faible grâce aux projections aléatoires et avoir de très bonnes performances statistiques. En outre, nous démontrons de solides performances empiriques de *SISOKR* sur des ensembles de données où les calculs requis par *IOKR* excèdent les capacités de la plupart des ordinateurs.

Enfin, nous proposons une architecture neuronale profonde capable de prédire dans l'espace caractéristique potentiellement de dimension infinie du noyau de sortie grâce à l'utilisation de projections aléatoires sur ce dernier. À cette fin, nous calculons la base formée par les fonctions propres de l'opérateur de covariance empirique de sortie projeté aléatoirement, et le réseau de neurones de *Deep Sketched Output Kernel Regression* (*DSOKR*) calcule par la suite une combinaison linéaire au sein de cette base et apprend ses coordonnées pendant l'entraînement. Ceci permet l'utilisation de méthodes d'optimisation à base de gradient pour n'importe quelle fonction de perte consistant en une composition de la perte quadratique et d'une fonction sous-différentiable, comme les fonctions de perte robustes standards par exemple. Ceci est également compatible avec toute sorte d'architecture neuronale, comme les transformeurs, ainsi que le confirment les expériences menées sur un problème de prédiction de molécules dont les données d'entrée sont des descriptions textuelles de ces dernières.

Notations

$:=$	Equal by definition
$\mathbb{N}^*$	Strictly positive integers
$\mathbb{R}$	Real numbers
$[\![n]\!]$	Set of integers from 1 to n ($\{1, \cdots, n\}$)
$a \lesssim b$	$\exists\, c > 0$ such that $a \leq cb$
$\mathcal{F}(\mathcal{X}, \mathcal{Y})$	Set of functions from a space $\mathcal{X}$ to a Hilbert space $\mathcal{Y}$
$\mathcal{L}(\mathcal{H}, \mathcal{K})$, $\mathcal{L}(\mathcal{H})$	Bounded linear operators between Hilbert spaces $\mathcal{H}$ and $\mathcal{K}$, shortened when $\mathcal{H} = \mathcal{K}$
$\langle \cdot, \cdot \rangle_{\mathcal{H}}$, $\| \cdot \|_{\mathcal{H}}$	Scalar product and norm in Hilbert space $\mathcal{H}$
$\| \cdot \|_{\mathrm{op}}$	Operator norm
$\| \cdot \|_{\mathrm{HS}}$	Hilbert-Schmidt norm
$A^{\#}$	Adjoint of operator A
$\mathrm{I}_{\mathcal{H}}$	Identity operator on space $\mathcal{H}$
$A^{\top}$	Transpose of matrix A
$A^{\dagger}$	Moore-Penrose inverse of matrix A
$A_{:i}$, $A_{j:}$	i-th column of matrix A, j-th row of matrix A
I_d	Identity matrix of dimension d
$\otimes$	Kronecker product of matrices, tensor product of Hilbert spaces or their elements
$\mathcal{X}$	Input space
$\mathcal{Y}$	Output space
$\mathcal{Z}$	Generic notation for at least a Polish space, denotes either $\mathcal{X}$ or $\mathcal{Y}$
$\mathrm{k}_{\mathcal{Z}} : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$	Positive definite kernel
$\mathcal{H}_{\mathcal{Z}}$	Reproducing Kernel Hilbert Space associated with $\mathrm{k}_{\mathcal{Z}}$
$\psi_{\mathcal{Z}} : z \in \mathcal{Z} \mapsto \mathrm{k}_{\mathcal{Z}}(\cdot, z) \in \mathcal{H}_{\mathcal{Z}}$	Canonical feature map

ρ	Joint probability distribution of the input/output pairs
n	Number of training samples
$\rho_{\mathcal{Z}}$	Marginal probability distribution of the i. i. d. training samples $\{z_1, \ldots, z_n\}$
$\mathrm{C}_{\mathcal{Z}} = \mathbb{E}_{z\sim\rho_{\mathcal{Z}}}[\psi_{\mathcal{Z}}(z) \otimes \psi_{\mathcal{Z}}]$	Covariance operator
$\mathrm{S}_{\mathrm{Z}} : f \in \mathcal{H}_{\mathcal{Z}} \mapsto \frac{1}{\sqrt{n}}(f(z_1), \ldots, f(z_n))^\top \in \mathbb{R}^n$	Sampling operator
$\mathrm{S}_{\mathrm{Z}}{}^{\#} : \alpha \in \mathbb{R}^n \mapsto \frac{1}{\sqrt{n}} \sum_{i=1}^n \alpha_i \, \psi_{\mathcal{Z}}(z_i) \in \mathcal{H}_{\mathcal{Z}}$	Adjoint of the sampling operator
$\widehat{\mathrm{C}}_{\mathrm{Z}} = \frac{1}{n} \sum_{i=1}^n \psi_{\mathcal{Z}}(z_i) \otimes \psi_{\mathcal{Z}}(z_i) = \mathrm{S}_{\mathrm{Z}}{}^{\#} \mathrm{S}_{\mathrm{Z}}$	Empirical covariance operator
$\mathrm{K}_{\mathrm{Z}} = (\mathrm{k}_{\mathcal{Z}}(z_i, z_j))_{1\leq I, j\leq n} = n \mathrm{S}_{\mathrm{Z}} \mathrm{S}_{\mathrm{Z}}{}^{\#}$	Kernel Gram matrix
$\mathrm{k}_{\mathrm{Z}}^{\mathrm{z}} = (\mathrm{k}_{\mathcal{Z}}(z, z_1), \ldots, \mathrm{k}_{\mathcal{Z}}(z, z_n))^\top \in \mathbb{R}^n$	Vector of kernel evaluations over an entry z and the training entries
$\mathrm{m}_{\mathcal{Z}}$	Sketching size
$\mathrm{R}_{\mathcal{Z}} \in \mathbb{R}^{\mathrm{m}_{\mathcal{Z}} \times n}$	Sketching matrix
$\widetilde{\mathrm{K}}_{\mathrm{Z}} = \mathrm{R}_{\mathcal{Z}} \mathrm{K}_{\mathrm{Z}} \mathrm{R}_{\mathcal{Z}}{}^\top$	Sketched kernel Gram matrix
$\widetilde{\mathrm{C}}_{\mathrm{Z}} = \mathrm{S}_{\mathrm{Z}}{}^{\#} \mathrm{R}_{\mathcal{Z}}{}^\top \mathrm{R}_{\mathcal{Z}} \mathrm{S}_{\mathrm{Z}}$	Sketched empirical covariance operator
$\widetilde{\mathrm{P}}_{\mathrm{Z}}$	Orthogonal projection operator induced by $\mathrm{R}_{\mathcal{Z}}$
$\sigma_i(A)$	i-th eigenvalue of the operator/matrix A
$\mathrm{diag}(v)$	Diagonal matrix induced by the vector v
Tr	Trace of operator or matrix
Im	Range of operator or matrix
Ker	Null space of operator or matrix
rank	Rank of operator or matrix
dim	Dimension of space
span	Linear space spanned by a set of vectors
$\lvert \cdot \rvert_+$	Positive part: $\lvert a \rvert_+ = \max(a, 0)$
$\lfloor \cdot \rfloor$	Floor integer part: $\lfloor a \rfloor = m \Leftrightarrow m \leq a < m+1$, $m \in \mathbb{N}$

Abbreviation

RKHS	Reproducing kernel Hilbert space
OVK	Operator-valued kernel
vv-RKHS	Vector-valued RKHS
ERM	Empirical Risk Minimisation
RFF	Random Fourier feature
KRR	Kernel ridge regression
KDE	Kernel Dependency Estimation
OKR	Output Kernel Regression
(S)I(S)OKR	(Sketched) Input (Sketched) Output Kernel Regression
DSOKR	Deep Sketched Output Kernel Regression
KPCA	Kernel Principal Component Analysis
SVD	Singular Value Decomposition
p. d.	positive definite
r. k.	reproducing kernel
i. i. d.	independent identically distributed
r. v.	random variable
w. r. t.	with respect to

1 Introduction

Contents

Structured prediction enlarges the scope of classical regression or classification problems to the prediction of complex outputs. Namely, it encompasses the prediction of objects of various natures, such as molecular identification (Brouard et al., 2016a), multi-label classification (Belanger and McCallum, 2016), label ranking (Korba et al., 2018), handwriting recognition (Cortes et al., 2007), sequence labeling (Tu and Gimpel, 2018), image reconstruction (Weston et al., 2003), image denoising (Belanger et al., 2017), semantic segmentation (Kirillov et al., 2023), protein 3D structure prediction (Jumper et al., 2021). Moreover, structured prediction approaches can be extended to tackle various kinds of problems, such as manifold valued regression (Rudi et al., 2018b) or conditional meta-learning (Wang et al., 2020).

Due to the wide variety of output types, it is challenging to design a unified theoretically grounded framework able to tackle indifferently graph prediction and label ranking for instance. Moreover, the usual non-linearity and high dimensionality of structured spaces induce difficulties in *efficiently* tackling such problems, both from *computational* and *statistical* perspectives. Finally, another challenge stems from learning relevant features from complex inputs as well, such as texts or images, while solving the prediction task.

In this thesis, we try to address these challenges thanks to surrogate methods. We stick within the supervised learning setting and first propose to focus on the scalability to large datasets of kernel methods. We then propose a scalable surrogate kernel method for structured prediction, together with an excess risk bound of the obtained estimator. We finally propose to extend this framework to neural networks. All these contributions are supported by synthetic and real-world experiments. We detail all the work conducted during this thesis in the following section.

1.1 Motivations and Contributions

In this section, we present the research questions that motivated this thesis and an overview of the contributions.

The metabolite identification problem. The metabolite identification problem (Cheng and Schenkman, 1983; Gentile et al., 1996; Zhang et al., 2000) is an emblematic example of structured prediction, and in particular of the surrogate kernel methods.

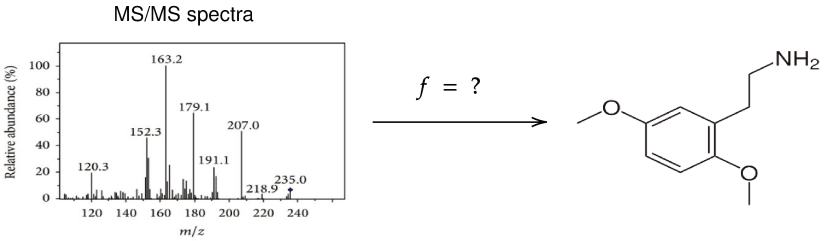


Figure 1.1: Illustration of the metabolite identification problem.

When the human body metabolizes a foreign molecule, the reaction produces molecules called metabolites. Hence, to design a new medicine, it is crucial to ensure that none of these metabolites is toxic. A widespread method to extract distinctive features from a biological sample is mass spectrometry, resulting in a tandem mass spectrum (MS/MS). From this spectrum, one can identify the molecular graph structure of the molecules in the sample. Being able to accurately predict the metabolites present in a biological sample based on its tandem mass spectrum is then of high interest. This problem is illustrated in Figure 1.1. We tackle this problem in this thesis, as well as other molecular identification problems, as it is an insightful example of structured prediction, where the output space is non-linear and high-dimensional, namely the space of connected node-labeled and edge-labeled graphs.

Structured space. Throughout this thesis, we consider structured output spaces $\mathcal{Y}$ that admit a non-linear embedding $\psi_{\mathcal{Y}}$ onto a Hilbert space $\mathcal{H}_{\mathcal{Y}}$. This embedding is either explicit, such as the molecular fingerprints (Ralaivola et al., 2005; Willett, 2006; Bajusz et al., 2017), i.e. vectors that encode the presence or absence of substructures in the molecule based on a given dictionary, or implicit. In that case, it can be induced by a similarity measure, such as the canonical feature map $y \mapsto \mathrm{k}_{\mathcal{Y}}(\cdot, y)$ of a positive definite output kernel $\mathrm{k}_{\mathcal{Y}} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ (Brouard et al., 2016b), or by a loss function $\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ (Ciliberto et al., 2020). This definition is consequently general, and we will focus on embeddings induced by a p. d. kernel in the following, i.e. the *Output Kernel Regression* framework.

Output Kernel Regression. Output Kernel Regression is a particular instance of surrogate kernel methods where the chosen output embedding is the canonical feature map of an output kernel $\psi_{\mathcal{Y}} : y \mapsto \mathrm{k}_{\mathcal{Y}}(\cdot, y)$. Surrogate methods offer a versatile and intuitive way to tackle structured prediction benefiting from the output embedding $\psi_{\mathcal{Y}}$. As illustrated in Figure 1.2, it consists in first solving the *surrogate* learning problem to obtain a mapping $h : \mathcal{X} \to \mathcal{H}_{\mathcal{Y}}$, and then retrieving the solution in the original output space $\mathcal{Y}$ by solving a *pre-image* problem. We then obtain a unified framework to tackle different structured prediction problems, e.g. graph prediction using an embedding induced by a graph kernel such as the shortest-path kernel, or handwriting recognition using an n-gram kernel. A question then naturally arises: how to learn this surrogate estimator h? In particular, which loss function can we use and which hypothesis space $\mathcal{H}$ can we consider?

Loss functions. Concerning the loss function, the very first idea is the squared distance within the feature space $\mathcal{H}_{\mathcal{Y}}$, i.e. $\ell(h(x), \psi_{\mathcal{Y}}(y)) = \|h(x) - \psi_{\mathcal{Y}}(y)\|^2_{\mathcal{H}_{\mathcal{Y}}}$ for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. Two main reasons justify this choice. First, since we use an embedding of

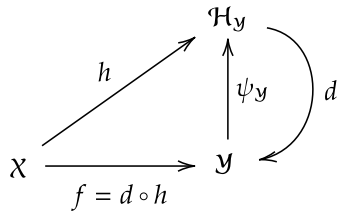


Figure 1.2: Illustration of the surrogate methods.

the outputs within a linear feature space, a relevant loss function is a function that takes into account the geometry of this feature space, i.e. that is a function of the inner product within this space, and the squared distance then appears as an intuitive choice. Second, from a practical viewpoint, since we focus on OKR, in the case where the embedding $\psi_\mathcal{Y}$ is implicit, we cannot directly compute $\psi_\mathcal{Y}(y)$ but the inner product between evaluations of this embedding, i.e. evaluations of the output kernel.

Hypothesis space for the surrogate problem. To handle the high or even infinite dimension of the output features, a classical choice in the literature is to use the kernel trick in the input space as well, and in particular, the generalisation of scalar Reproducing Kernel Hilbert Spaces to the Hilbertian case, namely the vector-valued RKHSs. This yields the so-called Input Output Kernel Regression model (Brouard et al., 2016b). Furthermore, combined with the squared loss, this choice benefits from the closed-form solution provided by the Kernel Ridge Regression.

Now, we are ready to present the four challenges we try to address in this thesis.

Scalability to large datasets. It is well-known that kernel methods do not scale up easily to very large datasets. In particular, kernels are here used both in the input and output spaces, which induces a heavy computational cost during both the learning and the inference phases.

Challenge 1: *Can we scale surrogate kernel methods up at both the training and inference phases, especially since they employ not only an input but also an output kernel?*

Excess risk bounds. Beyond their versatility, the other main advantage of surrogate kernel methods is their strong theoretical grounding. Building upon the theory of Operator-Valued Kernels and the Fisher consistency provided by kernel-induced losses and surrogate methods, it is possible to prove learning rates for the final structured estimator $f = d \circ h$. However, it is challenging to keep such guarantees while going beyond the scope of non-approximated KRR, in particular in the context of structured prediction.

Challenge 2: *Can we derive an excess risk bound for a scalable Output-Kernel-Regression-based estimator?*

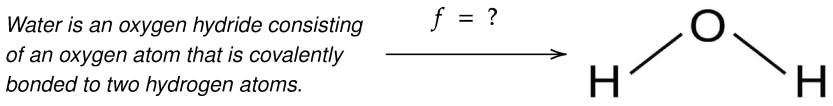


Figure 1.3: Illustration of the Text-to-Molecule problem.

Various losses. Although kernel-induced losses are not only defined by the choice of the squared distance within the feature space but also by the choice of the output kernel, which then defines a large panel of losses, it is technically possible to consider any loss that writes as $c(\|\psi_Y(\cdot) - \psi_Y(\cdot)\|^2_{\mathcal{H}_Y})$, where $c : \mathbb{R} \to \mathbb{R}$ is a non-decreasing function. Indeed, robust losses such as the ϵ-insensitive or Huber losses writes as previously mentioned and would add robustness to output outliers. Nonetheless, doing so is demanding during the learning phase, since we lose the KRR closed-form solution and obtain very high-dimensional parameters to optimize.

Challenge 3: *Can we efficiently employ a wider variety of losses within the Output Kernel Regression framework?*

Handle complex input data. Finally, a fourth challenge regards the design of the input kernel. Indeed, the latter takes operator values to cope with functions devoted to infinite outputs and requires a careful design. For instance, assume that we replace the tandem mass spectrum by natural language descriptions of the molecules as in the *Text2Mol* problem described in Figure 1.3, while there exist relevant kernels to handle tandem mass spectrum such as the probability product kernel, it is not the case for more complex data such as text. In such cases, more expressive models as deep neural networks are needed to tackle input representation learning.

Challenge 4: *Can we use kernel-induced losses with deep neural networks in spite of the possibly infinite dimension of the output feature space?*

We are now ready to introduce the three contributions present in this manuscript and summarized in Table 1.1.

Chapter 3: p-sparsified sketches for fast kernel methods with Lipschitz losses. We first take a step aside from structured prediction and focus on the input kernel, in particular the scalability of scalar and matrix-valued kernels while using generic Lipschitz losses. We extend the empirical excess risk bound obtained by Yang et al. (2017) on the scalar KRR problem based on the K_X-satisfiability concept to the expected risk with any Lipschitz loss and the multiple outputs setting. Moreover, we propose the p-sparsified sketches, a new sketching distribution, and prove their K_X-satisfiability. Thanks to the *decomposition trick*, the p-sparsified sketches are well-adapted to kernel methods and aim at the best possible tradeoff between computational and statistical performance, in terms of both time and space complexities, which is crucial for kernel methods. The extension to Lipschitz losses allows us to consider various problems such as the robust regression and the quantile regression. The implementations are provided on GitHub. This work, presented in Chapter 3, tackles challenges 1, 2, and 3 in the context of matrix-valued kernels, and constitutes

Table 1.1: Summary of the contributions.

Challenge	Chapter 3	Chapter 4	Chapter 5
1. Scalability	✓	✓	✓
2. Excess risk bound	✓	✓	
3. Various losses	✓		✓
4. Complex inputs			✓

a warm-up to the use of sketching on both the input and output kernels to scale IOKR up.

Chapter 4: Sketched Input Sketched Output Kernel Regression. While sketching has been previously leveraged as a way to reduce the number of parameters to learn during optimisation stage, we now exploit another interpretation, namely the ability to obtain random orthogonal projectors within a subspace of the feature space. Such a projector is obtained by computing the Singular Value Decomposition of the sketched empirical covariance operator $\widetilde{C}_Z$, it is basically the linear orthogonal projector onto its eigenbasis. In doing so, we introduce both input and output random projectors to scale surrogate kernel methods up at both the training and inference phase and note that input projectors mainly accelerate the training phase while output projectors mainly accelerate the inference phase. Besides, we extend the theoretical results from Rudi et al. (2015) on scalar Nyström KRR and Ciliberto et al. (2020) on non-approximated surrogate methods to the scalable version with sub-Gaussian sketches, including p-sparsified sketches. Our results prove close-to-optimal learning rates of the structured estimator based on the eigendecays of the input and output covariance operators: the faster the eigenvalues decrease, the better the rates, and we obtain optimal rates for finite-rank input/output covariances. The implementations are provided on GitHub. This work, presented in Chapter 4, tackles challenges 1 and 2 and offers the main tool to deploy kernel-induced losses to neural networks, i.e. the eigendecomposition of $\widetilde{C}_Y$.

Chapter 5: Deep Sketched Output Kernel Regression. Equipped with the eigenbasis of $\widetilde{C}_Y$, we introduce a new deep neural architecture that predicts in any generic Hilbert output feature space. More precisely, the last layer of such a neural network is a linear combination of the eigenfunctions of $\widetilde{C}_Y$. Then, benefiting from the finite dimension of such a basis and the kernel trick, we cope with the possible infinite dimension or implicitness of the output embedding $\psi_\mathcal{Y}$ and unlock the use of gradient-based methods and back-propagation to learn the previous neural network's layers, regardless of the neural architecture at hand, and consequently the input data at hand. Moreover, thanks to this approach, one can consider any loss $c(\|\psi_\mathcal{Y}(\cdot) - \psi_\mathcal{Y}(\cdot)\|^2_{\mathcal{H}_\mathcal{Y}})$ as previously stated, if c is differentiable or at least sub-differentiable. The implementations are provided on GitHub. This work, presented in Chapter 5, tackles challenges 1, 3, and 4.

1.2 Publications

These contributions have resulted in the following peer-reviewed publications and preprints ($\star$ indicates equal contribution):

- (El Ahmad et al., 2023) Tamim El Ahmad, Pierre Laforgue, and Florence d'Alché-Buc. Fast Kernel Methods for Generic Lipschitz Losses via p-Sparsified Sketches. In *Transactions on Machine Learning Research*, 2023. Reproduced in Chapter 3.

- (El Ahmad et al., 2024) Tamim El Ahmad, Luc Brogat-Motte, Pierre Laforgue, and Florence d'Alché-Buc. Sketch In, Sketch Out: Accelerating both Learning and Inference for Structured Prediction with Kernels. In *International Conference on Artificial Intelligence and Statistics*, 2024. Reproduced in Chapter 4.

- (El Ahmad et al., 2024) Tamim El Ahmad*, Junjie Yang*, Pierre Laforgue, and Florence d'Alché-Buc. Deep Sketched Output Kernel Regression for Structured Prediction. To appear in *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2024. Reproduced in Chapter 5.

2 Background

Contents

In this chapter, we present the notions upon which we build in this thesis. We first provide some reminders about kernel methods and structured prediction. Then, we present some previous works that deal with the above mentioned challenges.

2.1 Kernel Methods

In this section, we give some reminders about scalar-valued and operator-valued kernels. We refer the reader to Shawe-Taylor and Cristianini (2000, 2004); Steinwart and Christmann (2008b); Scholkopf and Smola (2018) for more details.

2.1.1 Kernel Methods for scalar-valued outputs

In machine learning, kernels are used to define a hypothesis space for a learning problem, namely the Reproducing Kernel Hilbert Space, which is a linear Hilbert space, based on a positive definite kernel. We first give the definition of a p. d. kernel.

Definition 2.1 (positive-definite kernel)**.** *A positive definite kernel* $\mathrm{k}_{\mathcal{X}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ *is an application such that*

- $\forall x, x' \in \mathcal{X}$, $\mathrm{k}_{\mathcal{X}}(x, x') = \mathrm{k}_{\mathcal{X}}(x', x)$ *(symmetry);*
- $\forall (\alpha_i)_{i=1}^n \in \mathbb{R}^n, (x_i)_{i=1}^n \in \mathcal{X}^n$, $\sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j \mathrm{k}_{\mathcal{X}}(x_i, x_j) \geq 0$ *(positive-definiteness).*

Positive-definite kernels can be seen as a way to define similarity measures between objects. Their advantage is that one can define such a similarity measure on various types of objects, and then on various spaces $\mathcal{X}$. We give classical kernel examples over different object types.

Example 2.2 (Vector kernels)**.** *Let* $\mathcal{X} = \mathbb{R}^d$*, for* $d \in \mathbb{N}^*$*. For all* $x, x' \in \mathbb{R}^d$*,*

- *Linear kernel:* $\mathrm{k}_{\mathcal{X}}(x, x') = x^\top x'$*;*
- *Polynomial kernel:* $\mathrm{k}_{\mathcal{X}}(x, x') = (1 + x^\top x')^p$ *with* $p \in \mathbb{N}^*$*;*
- *Gaussian kernel:* $\mathrm{k}_{\mathcal{X}}(x, x') = \exp(-\gamma \|x - x'\|_2^2)$ *with* $\gamma > 0$*.*

Example 2.3 (String kernel)**.** *Let* $\mathcal{X}$ *be the set of strings, i.e. the set of sequences of characters from an alphabet* Σ*. For strings* u *and* x*,* $|u|$ *denotes the number of characters in* u *and* $|x|_u$ *denotes the number of occurrences of* u *in* x*. Then, for* $x, x' \in \mathcal{X}$*, the n-gram kernel (Lodhi et al., 2002) is defined by*

$$\mathrm{k}_{\mathcal{X}}(x, x') = \sum_{|u|=n} |x|_u |x'|_u \,, \tag{2.1}$$

with $n \in \mathbb{N}^*$*.*

Example 2.4 (Graph kernel)**.** *Let* $\mathcal{X}$ *be the set of node-labeled graphs, i.e. the set of graphs* $G = (V, E)$ *where* V *denotes its set of vertices and* E *its set of edges. Let* $\mathcal{L} = \{1, \ldots, d\}$ *be the set of labels, and* $\ell : v \in V \mapsto \ell(v) \in \mathcal{L}$ *be the function that assigns a label for each vertex. Then, the vertex label histogram of* G *is a vector* $f = (f_1, \ldots, f_d)^\top$*, such that* $f_i = |\{v \in V : \ell(v) = i\}|$ *for each* $i \in \mathcal{L}$*. Let* f, f' *be the vertex label histograms of* G, G'*, respectively. Then, the vertex histogram kernel (Sugiyama and Borgwardt, 2015) is then defined as the linear kernel between* f *and* f'*, that is*

$$\mathrm{k}_{\mathcal{X}}(G, G') = f^\top f' . \tag{2.2}$$

You can find more graph kernel examples in Appendix C.2.

More generally, for any space Hilbert $\mathcal{H}_{\mathcal{X}}$ endowed with an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}_{\mathcal{X}}}$ and embedding $\psi_{\mathcal{X}} : \mathcal{X} \to \mathcal{H}_{\mathcal{X}}$, the kernel $\mathrm{k}_{\mathcal{X}}(\cdot, \cdot) = \langle \psi_{\mathcal{X}}(\cdot), \psi_{\mathcal{X}}(\cdot) \rangle_{\mathcal{H}_{\mathcal{X}}}$ is a p. d. one. Interestingly, it is shown that any p. d. kernel is uniquely associated with such a space $\mathcal{H}_{\mathcal{X}}$ called the Reproducing Kernel Hilbert Space (Aronszajn, 1950).

Reproducing Kernel Hilbert Space. We first describe how the RKHSs are constructed. Given a p. d. kernel $\mathrm{k}_{\mathcal{X}}$, we first consider the set of linear combinations of the functions $\mathrm{k}_{\mathcal{X}}(\cdot, x)$, i.e.

$$\mathcal{H}_{\mathcal{X}}^0 := \mathrm{span}(\mathrm{k}_{\mathcal{X}}(\cdot, x) \mid x \in \mathcal{X}). \tag{2.3}$$

Besides, for all $n, n' \in \mathbb{N}^*$, $(\alpha_i)_{i=1}^n \in \mathbb{R}^n$, $(\alpha'_i)_{i=1}^{n'} \in \mathbb{R}^{n'}$, $(x_i)_{i=1}^n \in \mathcal{X}^n$ and $(x'_i)_{i=1}^{n'} \in \mathcal{X}^{n'}$, it can be shown that the following application is an inner product over $\mathcal{H}_\mathcal{X}^0$

$$\left\langle \sum_{i=1}^{n} \alpha_i \, \mathrm{k}_\mathcal{X}(\cdot, x_i), \sum_{j=1}^{n'} \alpha'_j \, \mathrm{k}_\mathcal{X}(\cdot, x'_j) \right\rangle_{\mathcal{H}_\mathcal{X}^0} = \sum_{i=1}^{n} \sum_{j=1}^{n'} \alpha_i \alpha'_j \, \mathrm{k}_\mathcal{X}(x_i, x'_j). \tag{2.4}$$

As a consequence, $\mathcal{H}_\mathcal{X}^0$ is an inner product space, and one can easily show thanks to the Cauchy-Schwarz inequality that any Cauchy sequence in $\mathcal{H}_\mathcal{X}^0$ has a limit according to the norm associated to $\langle \cdot, \cdot \rangle_{\mathcal{H}_\mathcal{X}^0}$. Then, taking the completion, we obtain the following Hilbert space

$$\mathcal{H}_\mathcal{X} := \overline{\mathrm{span}(\mathrm{k}_\mathcal{X}(\cdot, x) \mid x \in \mathcal{X})}, \tag{2.5}$$

called the Reproducing Kernel Hilbert Space, and $\mathrm{k}_\mathcal{X}$ is called its reproducing kernel.

We then give a proper definition of a Reproducing Kernel Hilbert Space and a reproducing kernel.

Definition 2.5 (Reproducing Kernel Hilbert Space)**.** *Let $\mathcal{H}_\mathcal{X} \subset \mathbb{R}^\mathcal{X}$ be a class of functions forming a Hilbert space with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}_\mathcal{X}}$. The function $\mathrm{k}_\mathcal{X} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a reproducing kernel of $\mathcal{H}_\mathcal{X}$ if*

- $\forall x \in \mathcal{X}$, $\mathrm{k}_\mathcal{X}(\cdot, x) \in \mathcal{H}_\mathcal{X}$;
- $\forall x \in \mathcal{X}, h \in \mathcal{H}_\mathcal{X}$, $h(x) = \langle h, \mathrm{k}_\mathcal{X}(\cdot, x) \rangle_{\mathcal{H}_\mathcal{X}}$ *(reproducing property).*

If a r. k. exists, then $\mathcal{H}_\mathcal{X}$ is called a Reproducing Kernel Hilbert Space.

Equipped with Definitions 2.1 and 2.5, we state the following interesting properties of kernels.

Proposition 2.6. *A r. k. is uniquely associated to a RKHS $\mathcal{H}_\mathcal{X}$, we can then talk of "the" kernel of a RKHS, or "the" RKHS of a kernel. Moreover, a function $\mathrm{k}_\mathcal{X} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is p. d. if and only if it is a r. k.*

The latter can be easily shown thanks to the construction of RKHSs described above from one hand, and by taking the p. d. kernel $\langle \mathrm{k}_\mathcal{X}(\cdot, x), \mathrm{k}_\mathcal{X}(\cdot, x') \rangle_{\mathcal{H}_\mathcal{X}}$ from the other hand. To sum up, a p. d. is then associated to the Hilbert space $\mathcal{H}_\mathcal{X}$ called the RKHS, and the embedding $\psi_\mathcal{X} := \mathrm{k}_\mathcal{X}(\cdot, x)$ called the canonical feature map. Besides, the reproducing property assesses that any function in $\mathcal{H}_\mathcal{X}$ is a linear function.

Learning with scalar-valued kernels. Let us consider the supervised settings where $\mathcal{Y} \subseteq \mathbb{R}$, i.e. scalar regression or binary classification. We then have access to the i. i. d. training pairs $(x_i, y_i)_{i=1}^n \in (\mathcal{X} \times \mathbb{R})^n$ drawn from the unknown joint distribution ρ. As previously stated, p. d. kernels provide hypothesis spaces which are their RKHSs. Hence, equipped with a p. d. kernel $\mathrm{k}_\mathcal{X}$ together with its RKHS $\mathcal{H}_\mathcal{X}$ and a loss function $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, the goal is to estimate

$$f^* = \operatorname*{arg\,inf}_{f \in \mathcal{H}_\mathcal{X}} \mathbb{E}_{(x,y)\sim\rho}[\ell(f(x), y)], \tag{2.6}$$

which we call the minimisation of the expected risk. The choice of $\mathcal{H}_\mathcal{X}$ as a hypothesis space can be in many cases relevant. Indeed, this choice mainly relies on prior knowledge of the problem at hand and the underlying true target that maps x to y for any

$(x, y) \sim \rho$. As stated earlier, p. d. kernels can be considered as similarity measures over various data types or inferred from similarity measures, hence if a characteristic of the input data appears discriminative according to the problem at hand, a p. d. kernel that measures similarity based on this characteristic seems relevant. Since ρ is unknown and the expected risk is then intractable, we rather consider its empirical estimator, i.e. the mean over the evaluations of the loss over the training samples. Moreover, to avoid overfitting the training data, we add a regularisation term that controls the norm of the obtained estimator. Thus, let $\lambda > 0$ (adding a hyper-parameter), we solve the Empirical Risk Minimisation problem, i.e.

$$\hat{f} = \operatorname*{arg\,min}_{f \in \mathcal{H}_\mathcal{X}} \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i) + \lambda \|f\|^2_{\mathcal{H}_\mathcal{X}}. \tag{2.7}$$

Such a problem has been widely considered for various convex losses. The most famous example is the square loss, yielding the Kernel Ridge Regression, which has the specificity to have a closed-form solution. Moreover, the hinge loss yields the well-known Support Vector Machines (Cortes and Vapnik, 1995) for the binary classification and the ϵ-insensitive ℓ_1 loss yields the Support Vector Regression (Drucker et al., 1997).

Nonetheless, it does not appear straightforward how to solve eq. (2.7). Here comes another powerful tool inherent to kernel methods that relies on the reproducing property and simple orthogonality argument: the *representer theorem* (Kimeldorf and Wahba, 1971).

Theorem 2.7 (Representer theorem)**.** *Let $\mathrm{k}_\mathcal{X}$ be a kernel on $\mathcal{X}$ and let $\mathcal{H}_\mathcal{X}$ be its associated RKHS. Consider a set of points $(x_i)_{i=1}^n \in \mathcal{X}^n$. Let $V : \mathbb{R}^{n+1} \to \mathbb{R}$ be a function which is strictly increasing with respect to its last argument. Then any solution $\hat{f}$ to the problem*

$$\min_{f \in \mathcal{H}_\mathcal{X}} V\Big(f(x_1), \dots, f(x_n), \|f\|_{\mathcal{H}_\mathcal{X}}\Big), \tag{2.8}$$

can be written in the form

$$\hat{f} = \sum_{i=1}^{n} \alpha_i \, \mathrm{k}_\mathcal{X}(., x_i) \tag{2.9}$$

for some $(\alpha_i)_{i=1}^n \in \mathbb{R}^n$.

Noting that eq. (2.7) is a particular case of eq. (2.8), we have that $\hat{f} = \sum_{i=1}^n \hat{\alpha}_i \, \mathrm{k}_\mathcal{X}(\cdot, x_i)$ where $\hat{\alpha} = (\hat{\alpha}_1, \dots, \hat{\alpha}_n)^\top \in \mathbb{R}^n$ is the solution to

$$\min_{\alpha \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} \ell([\mathrm{K}_\mathrm{X} \alpha]_i, y_i) + \lambda \alpha^\top \mathrm{K}_\mathrm{X} \alpha, \tag{2.10}$$

where $\mathrm{K}_\mathrm{X} = (\mathrm{k}_\mathcal{X}(x_i, x_j)_{1 \le I, j \le n}$. As a consequence, we obtain an optimisation problem whose goal is to learn the n parameters α_is and can be solved thanks to classical gradient-based methods. A particular instance of such a problem is, as stated previously, KRR.

Example 2.8 (Kernel Ridge Regression)**.** *Setting $\ell : (y, y') \mapsto (y - y')^2$, we obtain as an optimisation problem on α*

$$\min_{\alpha \in \mathbb{R}^n} \frac{1}{n} \|\mathrm{K}_\mathrm{X} \alpha - Y\|_2^2 + \lambda \alpha^\top \mathrm{K}_\mathrm{X} \alpha. \tag{2.11}$$

Table 2.1: Comparison between scalar ($k_{\mathcal{X}}$) and operator-valued kernels (K).

	scalar-valued kernel	operator-valued kernel
kernel	$k_{\mathcal{X}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$	$K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$
symmetry	$k_{\mathcal{X}}(x, x') = k_{\mathcal{X}}(x', x)$	$K(x, x') = K(x', x)^{\#}$
positive-definiteness	$\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j k_{\mathcal{X}}(x_i, x_j) \geq 0$	$\sum_{i=1}^{n}\sum_{j=1}^{n} \langle y_i, K(x_i, x_j) y_j \rangle_{\mathcal{Y}} \geq 0$
reproducing property	$h(x) = \langle h, k_{\mathcal{X}}(\cdot, x) \rangle_{\mathcal{H}_{\mathcal{X}}}$	$h(x) = K(\cdot, x)^{\#} h$

where $Y = (y_1, \ldots, y_n)^\top \in \mathbb{R}^n$. *By setting the gradient of the above quantity to zero, we obtain the solution*

$$\hat{\alpha} = (K_X + n\lambda I_n)^{-1} Y \,. \tag{2.12}$$

Although admitting a closed-form solution is an advantage of KRR, it is obvious to note that the inversion of the n^2*-matrix* $K_X + n\lambda I_n$ *implies a* $\mathcal{O}(n^3)$ *time complexity, as well as a* $\mathcal{O}(n^2)$ *space complexity to store it in memory.*

Scalar-valued p. d. kernels offer then a principled to solve scalar regression tasks. We now introduce Operator-Valued Kernels that extend scalar-valued kernels to the regression onto a generic Hilbert space.

2.1.2 Kernel Methods for vector-valued outputs

The extension of RKHSs to the general case of outputs in a Hilbert space $\mathcal{Y}$ is the vector-valued RKHSs. As for scalar-valued RKHSs, we will first give a definition of Operator-Valued Kernels and then how to build vv-RKHSs. An overview of the main differences between scalar-valued kernels and OVKs is given in table 2.1. For more thorough details about OVKs, we refer the reader to Senkene and Tempel'man (1973); Micchelli and Pontil (2005); Caponnetto and De Vito (2007); Carmeli et al. (2010); Álvarez et al. (2012).

Definition 2.9 (Operator-Valued Kernel)**.** *An Opereator-Valued Kernel* $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ *is an application such that*

- $\forall x, x' \in \mathcal{X}$, $K(x, x') = K(x', x)^{\#}$ *(symmetry);*
- $\forall (y_i)_{i=1}^n \in \mathcal{Y}^n, (x_i)_{i=1}^n \in \mathcal{X}^n$, $\sum_{i=1}^{n}\sum_{j=1}^{n} \langle y_i, K(y_i, y_j) y_j \rangle_{\mathcal{Y}} \geq 0$ *(positive-definiteness).*

First, note that if $\mathcal{Y} = \mathbb{R}^d$ for $d \in \mathbb{N}^*$, $\mathcal{L}(\mathcal{Y}) = \mathbb{R}^{d \times d}$. In this specific case, we call the OVKs the matrix-valued kernels (Álvarez et al., 2012). We give the definition of the decomposable kernel, a particular and widely studied instance of OVKs.

Definition 2.10 (Decomposable Kernel)**.** *Let* $k_{\mathcal{X}}$ *be a scalar-valued p. d. kernel and* $M \in \mathcal{L}(\mathcal{Y})$*, for all* $x, x' \in \mathcal{X}$*, a decomposable kernel* K *is then defined such that*

$$K(x, x') = k_{\mathcal{X}}(x, x') M \,. \tag{2.13}$$

Such a kernel is popular because it separates its effects on the input and output data. One can then encode the prior knowledge about the inputs in $k_{\mathcal{X}}$ and about the outputs in M. We give some examples about M.

Example 2.11 (Identity decomposable kernel)**.** *The simplest OVK is obtained by setting $M = I_{\mathcal{Y}}$. This means that we encode no prior knowledge in M and then no restriction in the output space since* $\mathrm{Im}(I_{\mathcal{Y}}) = \mathcal{Y}$.

Matrix-valued decomposable kernels have also been used to tackle multi-task regression (Evgeniou et al., 2005; Sheldon, 2008) or joint quantile regression (Sangnier et al., 2016), as in chapter 3 of this thesis. We give two examples of such kernels.

Example 2.12 (Decomposable kernel for multi-task regression)**.** *It is possible to encode prior knowledge about the task relationships in M. Such relationships can be represented by a graph where each task is a node, and two assumed related tasks and connected by an edge. Then, if L denotes the Laplacian of this graph, Evgeniou et al. (2005) and Sheldon (2008) considered $M = (\mu L + (1-\mu) I_d)^{-1}$, with $\mu \in [0,1]$. As limiting cases, if $\mu = 0$, $M = I_d$ and all tasks are assumed independent, and if $\mu = 1$, we rely on the prior knowledge encoded by L.*

Example 2.13 (Decomposable kernel for joint quantile regression)**.** *The goal is to predict d quantile levels $(\tau_i)_{i \leq d} \in (0,1)$ of an output y given the input x. Sangnier et al. (2016) proposed $M_{ij} = \exp(-\gamma(\tau_i - \tau_j)^2)$, as it enforces the proximity of predictions between close quantiles levels and also limits the crossing phenomenon for the predicted quantiles.*

OVKs then generalise scalar-valued kernels to the generic Hilbert output space case and decomposable kernels give an intuitive instance of such kernels. We now show how to build vv-RKHSs, similarly to the scalar case.

Vector-valued Reproducing Kernel Hilbert Space. As for the scalar case, a vv-RKHS is obtained via the completion of a linear space, however not only the inputs but also the outputs are used to construct this set. Let $\mathcal{K}$ be an OVK, its vv-RKHS $\mathcal{H}$ is

$$\mathcal{H} := \overline{\mathrm{span}(\mathcal{K}(\cdot, x)y \mid x \in \mathcal{X}, y \in \mathcal{Y})} = \overline{\mathcal{H}^0}, \tag{2.14}$$

where the completion is based on the norm induced by the following inner product

$$\left\langle \sum_{i=1}^{n} \alpha_i \mathcal{K}(\cdot, x_i) y_i, \sum_{j=1}^{n'} \alpha'_j \mathcal{K}(\cdot, x'_j) y'_j \right\rangle_{\mathcal{H}^0} = \sum_{i=1}^{n} \sum_{j=1}^{n'} \alpha_i \alpha'_j \left\langle y_i, \mathcal{K}(x_i, x'_j) y'_j \right\rangle_{\mathcal{Y}}. \tag{2.15}$$

Similarly to the scalar case, an OVK is uniquely associated with a vv-RKHS $\mathcal{H}$.

Theorem 2.14 (vector-valued RKHS)**.** *Let $\mathcal{K}$ be an OVK. There is a unique Hilbert space $\mathcal{H}$ of functions from $\mathcal{X}$ to $\mathcal{Y}$, the vv-RKHS of $\mathcal{K}$, such that for all $x \in \mathcal{X}$, $y \in \mathcal{Y}$ and $h \in \mathcal{H}$*

- $(x' \mapsto \mathcal{K}\left(x, x'\right) y) \in \mathcal{H}$;
- $\langle h, \mathcal{K}\left(\cdot, x\right) y \rangle_{\mathcal{H}} = \left\langle h(x), y \right\rangle_{\mathcal{Y}}$ *(reproducing property).*

Learning with operator-valued kernels. Equipped with n i. i. d. training pairs $(x_i, y_i)_{i=1}^n \in (\mathcal{X} \times \mathcal{Y})^n$, an OVK $\mathcal{K}$ together with its vv-RKHS $\mathcal{H}$, a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ and a regularisation coefficient $\lambda > 0$, we aim at solving the following ERM problem

$$\hat{f} = \operatorname*{arg\,min}_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i) + \lambda \|f\|_{\mathcal{H}}^2. \tag{2.16}$$

Many machine learning tasks fall into the scope of the previous problem. First, as in chapter 3, multi-output regression corresponds to $\mathcal{Y} = \mathbb{R}^d$ for some $d \in \mathbb{N}^*$, and matrix-valued kernels proved to be well-suited to deal with such a problem (Micchelli and Pontil, 2005; Álvarez et al., 2012; Baldassarre et al., 2012; Brouard et al., 2016b; Sangnier et al., 2016, 2017). Functional Output Regression aims at predicting functions, the output space is then a space of functions, usually of square-integrable real-valued functions. Vv-RKHS have successfully been leveraged to tackle such problems, see e.g. Lian (2007); Kadri et al. (2010, 2016); Bouche et al. (2021); Lambert et al. (2022). Finally, and it is of particular interest in this thesis, surrogate kernel methods (Brouard et al., 2011; Kadri et al., 2013b; Brouard et al., 2016a,b; Ciliberto et al., 2016, 2020; Laforgue et al., 2020; Brogat-Motte et al., 2022b) are a particular instance of such a problem since the output space of the surrogate problem is the RKHS of a scalar-valued output kernel as illustrated in Figure 1.2.

Similarly to the scalar case, a *representer theorem* (Micchelli and Pontil, 2005) shows that $\hat{f}$ is a linear combination of the input features.

Theorem 2.15 (Micchelli and Pontil 2005, Theorem 4.2)**.** *Let $V : \mathcal{Y}^n \times \mathbb{R} \to \mathbb{R}$ be a function such that for any $\mathbf{y} \in \mathcal{Y}^n$, the partial function $t \mapsto V(\mathbf{y}, t)$ is strictly increasing. Then any solution $\hat{f}$ to the problem*

$$\min_{f \in \mathcal{H}} V\Big((f(x_1), \dots, f(x_n)), \|f\|_{\mathcal{H}}\Big), \tag{2.17}$$

can be written in the form

$$\hat{f} = \sum_{i=1}^{n} \mathcal{K}(\cdot, x_i)\alpha_i, \tag{2.18}$$

for some $(\alpha_i)_{i=1}^n \in \mathcal{Y}^n$.

As a consequence, $\hat{f} = \sum_{i=1}^n \mathcal{K}(\cdot, x_i)\hat{\alpha}_i$ where $\hat{\alpha} = (\hat{\alpha}_1, \dots, \hat{\alpha}_n)^\top \in \mathcal{Y}^n$ is the solution to

$$\min_{\alpha \in \mathcal{Y}^n} \frac{1}{n} \sum_{i=1}^{n} \ell\Big(\sum_{i=1}^{n} \mathcal{K}(x, x_i)\alpha_i, y_i\Big) + \lambda \sum_{i=1}^{n}\sum_{j=1}^{n} \Big\langle \alpha_i, \mathcal{K}(x_i, x_j)\alpha_j \Big\rangle_{\mathcal{Y}}. \tag{2.19}$$

Hence, assuming that $\mathcal{Y} = \mathbb{R}^d$ for $d \in \mathbb{N}^*$, $\hat{\alpha} \in \mathbb{R}^{nd}$ and we then obtain an optimisation problem on $n \times d$ parameters, which is usually a very large number but does not technically prevent from using classical gradient-based methods. However, in the case where $\mathcal{Y}$ is infinite-dimensional, such as Functional Output Regression or surrogate kernel methods, solving eq. (2.19) appears very difficult. When using the square loss $\ell(y, y') = \|y - y'\|_{\mathcal{Y}}^2$, Micchelli and Pontil (2005) shows that the $\hat{\alpha}_i$s satisfy the equations

$$\sum_{i=1}^{n} (\mathcal{K}(x_j, x_i) + n\lambda\delta_{ij})\hat{\alpha}_i = y_j, \tag{2.20}$$

where δ is the Kronecker symbol, i.e. $\delta_{ii} = 1$ and $\forall i \neq j$, $\delta_{ij} = 0$. In the case of an identity decomposable kernel, we recover a familiar closed-form solution

Example 2.16 (Vector-valued Kernel Ridge Regression)**.** *Setting $\ell : (y, y') \mapsto \|y - y'\|_{\mathcal{Y}}^2$ and $\mathcal{K} = \mathrm{k}_{\mathcal{X}} I_{\mathcal{Y}}$ for a scalar-valued p. d. kernel $\mathrm{k}_{\mathcal{X}}$, we obtain as a solution to eq. (2.19), $\hat{f}(\cdot) = \sum_{i=1}^n \hat{\alpha}_i(\cdot)y_i$ with, for all $x \in \mathcal{X}$,*

$$\hat{\alpha}(x) = (\mathrm{K}_X + n\lambda I_n)^{-1} \mathrm{k}_X^x, \tag{2.21}$$

where $\mathrm{k}_X^x = (\mathrm{k}_{\mathcal{X}}(x, x_1), \dots, \mathrm{k}_{\mathcal{X}}(x, x_n))^\top \in \mathbb{R}^n$.

Moreover, exploiting duality properties, Brouard et al. (2016b) shows that it is possible to obtain a parametrized dual problem of eq. (2.19) in the maximum margin regression case, i.e. $\ell(y, y') = \max(0, 1 - \langle y, y' \rangle_{\mathcal{Y}})$, which constitutes a generalisation of the scalar Support Vector Machines. Finally, Laforgue et al. (2020) shows that under some assumptions on the Fenchel-Legendre transform of the loss function and the stability of the kernel, i.e. for all $(x, x') \in \mathcal{X}$ and for all $y \in \text{span}((y_i)_{i=1}^n)$, $K(x, x')y \in \text{span}((y_i)_{i=1}^n)$, the solution to eq. (2.19) writes as $\hat{f}(\cdot) = \sum_{i=1}^n K(\cdot, x_i)\hat{\omega}_{ij}y_j$ where $\hat{\Omega} = (\omega_{ij})_{1 \leq i,j \leq n} \in \mathbb{R}^{n \times n}$ is a solution to the parametrized dual problem of eq. (2.19).

To sum up, kernels constitute a principled way to solve many machine learning problems with scalar outputs, by defining the RKHS, a hypothesis space enjoying many interesting properties, and in particular the representer theorem in order to solve ERM problems. Furthermore, we will see in section 2.4 that kernels offer solid theoretical foundations. Nevertheless, as pointed out by the KRR example, they suffer from a heavy dependency on the number of training samples n in terms of both space and time complexity. In the scalar case, looking at *eq.* (2.10), computing K_X implies storing n^2 values in memory, and learning n parameters implies heavy gradient or proximal operator computations. In the generic Hilbertian case, the resulting optimisation problems usually contain even more parameters to learn. Decreasing this complexity while keeping good statistical accuracy is one of the main challenges of this thesis, and we present some of the classical approximation techniques in section 2.3.

2.2 Structured Prediction

We here present the structured prediction problem in the supervised settings. We first give some examples of such a task and the main challenges that arise from it. Then, we introduce Output Kernel Regression, a surrogate method based on p. d. kernels defined on the output space. Finally, we give an overview of other different families of methods, namely the Conditional Random Fields (CRF) (Lafferty et al., 2001), the Structured Support Vector Machines (SSVM) (Tsochantaridis et al., 2004, 2005; Taskar et al., 2005), Max-margin Markov (M^3) networks (Taskar et al., 2003), and Structured Prediction Energy Networks (SPEN) (Belanger and McCallum, 2016).

Settings and challenges of structured prediction. The most studied settings in supervised regression are *regression* and *classification*, where the output space's dimension is *low*, e.g. space of real-valued vectors for regression or zeros/ones vectors for classification. Regarding the input data, many problems deal with *high-dimensional* objects such as images or molecules, e.g. molecular property prediction. In structured prediction, the challenge is to deal with *high-dimensional outputs*. Many examples of such outputs exist, e.g. graphs, binary vectors, permutations, or sequences of characters, and define many problems in various fields:

- computational biology: 2D molecular prediction (Brouard et al., 2016a), 3D molecular prediction (Jumper et al., 2021);
- natural language processing: handwriting recognition (Cortes et al., 2007), language translation (Bahdanau et al., 2015);
- computer vision: image reconstruction (Weston et al., 2003), image denoising (Belanger et al., 2017), facial landmark detection (Belharbi et al., 2017), semantic segmentation (Kirillov et al., 2023)

- recommendation systems: label ranking (Korba et al., 2018), information retrieval (Lindgren et al., 2021).

Due to the high dimension of the output space, solving structured prediction in a multi-task fashion without taking account of the structure of the output space, i.e. by predicting each component of the outputs independently, is difficult and can lead to poor statistical performance. In addition to their high dimension, structured spaces are usually very *large*. Let us consider rather simple structured tasks, namely multi-label prediction and label ranking, their corresponding output space sizes are $|\mathcal{Y}| = 2^d$ and $|\mathcal{Y}| = d!$ respectively, for some $d \in \mathbb{N}^*$, which grows exponentially in d. Furthermore, another challenge is the *lack of linear structure* in such spaces, making linear interpolation obsolete for instance. From an algorithmic viewpoint, many structured prediction tasks imply *discrete* spaces, such as graph prediction, making the use of gradient-based algorithms not straightforward. Finally, the *complexity* of such objects make it difficult to design a *general* structured prediction algorithm, that will be able to tackle graph prediction, label ranking and sequence prediction within the same framework for example. Hence, many works rather focus on a specific task, whether it is semantic segmentation (Kirillov et al., 2023) or 3D protein structure prediction (Jumper et al., 2021).

Energy function. In order to account for the relationships between the components of the output objects, as well as between the inputs and outputs, one can encode them into an *energy function* $(x, y) \mapsto E(x, y)$. The *estimator* $f : \mathcal{X} \to \mathcal{Y}$ is then obtained by maximizing the energy function over $\mathcal{Y}$, i.e. for all $x \in \mathcal{X}$

$$f(x) \coloneqq \operatorname*{arg\,max}_{y \in \mathcal{Y}} E(x, y). \tag{2.22}$$

The goal is then to properly define a relevant energy function E and to learn it based on the training data at hand. However, such a task seems very challenging. First, as previously mentioned, there is a wide variety of structured outputs, then it is not straightforward to design a *versatile* energy function that applies to any structured spaces according to an appropriate hyper-parameter choice. Concerning the learning part, let us consider the most standard paradigm in supervised learning, namely the minimization of the risk given a loss function $\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, we then obtain the following problem,

$$\min_{E : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}} \mathbb{E}_{(x,y)\sim\rho}[\Delta(\operatorname*{arg\,max}_{y' \in \mathcal{Y}} E(x, y'), y)]. \tag{2.23}$$

Hence, learning E through the loss function and the pre-image problem seems rather difficult, and would need differentiable approximations (Long et al., 2015; Niculae et al., 2018; Berthet et al., 2020; Niculae and Martins, 2020).

In the following, we present different families of structured prediction models that fit into this generic *energy-based* framework but with their own specifications. We first start with *Output Kernel Regression*, which is the framework upon which we build in chapters 4 and 5.

2.2.1 Output Kernel Regression

Kernel-induced loss. To cope with the lack of linear structure of the output space, Output Kernel Regression (Weston et al., 2003; Geurts et al., 2006; Cortes et al., 2005, 2007; Kadri et al., 2013a; Brouard et al., 2011, 2016a,b; Ciliberto et al., 2016, 2020)

endowes $\mathcal{Y}$ with an embedding $\psi_\mathcal{Y}$ onto a linear feature space $\mathcal{H}_\mathcal{Y}$ which is the canonical feature map of a p. d. output kernel $\mathrm{k}_\mathcal{Y} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$. As explained in section 2.1, two main advantages of p. d. kernels are their versatility to various object types, i.e. it is possible to define a p. d. kernel on many different spaces, and their ability to define similarity measures between objects, see e.g. Brouard et al. (2016a) for various choices of kernels for tandem mass spectra based on various criteria. As such properties have been mainly used on the input data, OKR leverages them on the output data. It is then possible to define a loss that takes into account the structure of the output objects based on this relevant similarity measure, and to compute it thanks to the kernel trick,

$$\Delta : (y, y') \in \mathcal{Y}^2 \mapsto \|\psi_\mathcal{Y}(y) - \psi_\mathcal{Y}(y')\|^2_{\mathcal{H}_\mathcal{Y}} = \mathrm{k}_\mathcal{Y}(y, y) + \mathrm{k}_\mathcal{Y}(y', y') - 2\mathrm{k}_\mathcal{Y}(y, y'), \tag{2.24}$$

and if $\mathrm{k}_\mathcal{Y}$ is normalized, i.e. such that for all $y \in \mathcal{Y}$, $\mathrm{k}_\mathcal{Y}(y, y) = 1$, then $\Delta(y, y') = 2 - 2\mathrm{k}_\mathcal{Y}(y, y')$. In the following, we assume without loss of generality that $\mathrm{k}_\mathcal{Y}$ is normalized. Note that, based on the choice of $\mathrm{k}_\mathcal{Y}$, such a loss defines a large panel of losses, not only across various output types but also among a given one. For example, in multi-label classification, choosing the linear kernel defines the Hamming loss, when the Tanimoto kernel (Tanimoto, 1958) defines the F1-loss. In label ranking, the Kemeny and Hamming embeddings define respectively Kendall's τ distance and the Hamming loss, see Korba et al. (2018) for more details. The goal is then to estimate $f^* : \mathcal{X} \to \mathcal{Y}$ such that

$$f^* = \operatorname*{arg\,inf}_{f:\mathcal{X}\to\mathcal{Y}} \mathcal{R}(f) = \operatorname*{arg\,inf}_{f:\mathcal{X}\to\mathcal{Y}} \mathbb{E}_{(x,y)\sim\rho}[\|\psi_\mathcal{Y}(f(x)) - \psi_\mathcal{Y}(y)\|^2_{\mathcal{H}_\mathcal{Y}}], \tag{2.25}$$

based on i. i. d. input/output training pairs $\{(x_1, y_n), \ldots, (x_n, y_n)\}$. It is not straightforward how to solve problem (2.25), since according to $\mathrm{k}_\mathcal{Y}$, Δ is not necessarily lower semi-continuous, convex, or differentiable, see Blondel et al. (2020) for an in-depth study of a generic way to construct a convex loss function for structured prediction problems.

Surrogate method. Hence, one rather solves such a problem by following a two-step approach as illustrated in Figure 1.2:

1. training step: one solves the surrogate regression problem obtained by replacing the outputs y_is by their embedded counterparts $\psi_\mathcal{Y}(y_i)$s, and then find $\hat{h} : \mathcal{X} \to \mathcal{H}_\mathcal{Y}$ by ERM which estimates

$$h^* = \operatorname*{arg\,min}_{h:\mathcal{X}\to\mathcal{Y}} \mathbb{E}_{(x,y)\sim\rho}[\|h(x) - \psi_\mathcal{Y}(y)\|^2_{\mathcal{H}_\mathcal{Y}}]; \tag{2.26}$$

2. inference step: retrieve the solution by solving a pre-image problem, then $\hat{f} = d \circ \hat{h}$ such that for all $x \in \mathcal{X}$,

$$\hat{f}(x) = d(\hat{h}(x)) = \operatorname*{arg\,min}_{y\in\mathcal{Y}} \|\hat{h}(x) - \psi_\mathcal{Y}(y)\|^2_{\mathcal{H}_\mathcal{Y}} = \operatorname*{arg\,max}_{y\in\mathcal{Y}} \langle \hat{h}(x), \psi_\mathcal{Y}(y)\rangle_{\mathcal{H}_\mathcal{Y}}. \tag{2.27}$$

We then relegate the difficult handling of structured objects through $\psi_\mathcal{Y}$ to the inference step. Moreover, from a theoretical perspective, Ciliberto et al. (2016, 2020) prove the *Fisher consistency* of such a model.

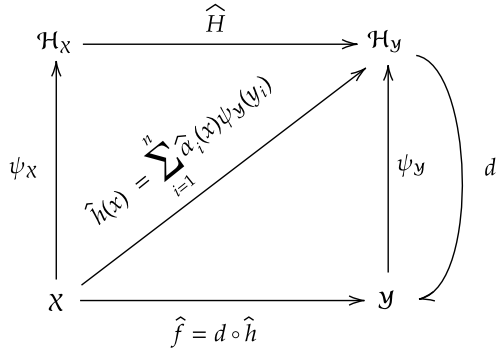


Figure 2.1: Illustration of Input Output Kernel Regression (Brouard et al., 2016b) with an identity decomposable input kernel, also corresponding to Kernel Dependency Estimation (Cortes et al., 2005).

Lemma 2.17 (Ciliberto et al. 2020, Lemma 1)**.** *Let $\mathcal{Y}$ be compact, $\mathrm{k}_\mathcal{Y} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be a normalized p. d. kernel and $\psi_\mathcal{Y} : y \mapsto \mathrm{k}_\mathcal{Y}(\cdot, y)$ its canonical feature map. Then, for f^* defined as in eq. (2.25) and h^* defined as in eq. (2.26), then*

$$h^*(x) = \mathbb{E}_y[\psi_\mathcal{Y}(y)|x], \tag{2.28}$$

and

$$f^*(x) = \operatorname*{arg\,min}_{y \in \mathcal{Y}} \|h^*(x) - \psi_\mathcal{Y}(y)\|^2_{\mathcal{H}_\mathcal{Y}}, \tag{2.29}$$

almost surely with respect to $\rho_\mathcal{X}$.

Moreover, they prove the *comparison inequality*, namely the fact that the excess risk of $\hat{f}$ is bounded by the excess risk of $\hat{h}$, we will give more details later.

Remark 2.18 (Implicit Loss Embeddings)**.** *Actually, Ciliberto et al. (2020) goes beyond the scope of OKR and gives such results for any loss Δ that admits what they call an* Implicit Loss Embedding*. Let $\mathcal{Y}$ be the structured output space and $\mathcal{Y}'$ the label space (such distinction comes from tasks such as label ranking, where $\mathcal{Y}$ denotes the set of permutations and $\mathcal{Y}'$ the set of scalar scores representing the relevance of the elements to rank based on an input $x \in \mathcal{X}$), a continuous map $\Delta : \mathcal{Y} \times \mathcal{Y}' \to \mathbb{R}$ admits an ILE if there exists a separable Hilbert space $\mathcal{H}$ and two measurable bounded maps $\psi_\mathcal{Y} : \mathcal{Y} \to \mathcal{H}$ and $\varphi : \mathcal{Y}' \to \mathcal{H}$, such that for any $y \in \mathcal{Y}$ and $y' \in \mathcal{Y}'$ we have*

$$\Delta(y, y') = \langle \psi_\mathcal{Y}(y), \varphi(y') \rangle_\mathcal{H}, \tag{2.30}$$

and $\|\varphi(y')\|_\mathcal{H} \leq 1$. See Ciliberto et al. (2020, Theorem 8) for the proof that any kernel-induced loss defined as in eq. (2.24) with a normalized kernel $\mathrm{k}_\mathcal{Y}$ admits admits an ILE, where $\mathcal{H} = \mathcal{H}_\mathcal{Y}$ and $c_\Delta = sup_{y \in \mathcal{Y}} \|\psi_\mathcal{Y}(y)\|_{\mathcal{H}_\mathcal{Y}} = 6$.

Then, this surrogate model is well-conditioned and offers strong theoretical groundings. Finally, we see that we fall into the scope of energy-based models based on the inference step, where the energy function E is given by $E : (x, y) \mapsto \langle \hat{h}(x), \psi_\mathcal{Y}(y) \rangle_{\mathcal{H}_\mathcal{Y}}$. The question that now raises is: how can we learn $\hat{h}$? In particular, how can we handle the fact that $\mathcal{H}_\mathcal{Y}$ may be infinite-dimensional and $\psi_\mathcal{Y}$ implicit?

Training step. While Weston et al. (2003) first proposed to compute the output Kernel Principal Component Analysis and to predict each component of the output features in a multi-task fashion via scalar KRR, we rather focus on leveraging Operator-Valued Kernels to manage the high dimension of $\mathcal{H}_\mathcal{Y}$. In particular, we present the most simple instance of Input Output Kernel Regression (Brouard et al., 2016b) that uses an identity decomposable input kernel, and that corresponds to the Kernel Dependency Estimation in Cortes et al. (2005), illustrated in Figure 2.1. Let $\mathrm{k}_\mathcal{X} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a p. d. kernel associated to the RKHS $\mathcal{H}_\mathcal{X}$ and $\mathcal{K} = \mathrm{k}_\mathcal{X} I_{\mathcal{H}_\mathcal{Y}}$ be the identity decomposable OVK associated to the vv-RKHS $\mathcal{H}$, and let $\lambda > 0$ be a regularisation parameter, we estimate $\hat{h}$ by solving the following regularized ERM problem,

$$\min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \|h(x_i) - \psi_\mathcal{Y}(y_i)\|^2_{\mathcal{H}_\mathcal{Y}} + \lambda \|h\|^2_\mathcal{H}. \tag{2.31}$$

Then, the represented theorem (Micchelli and Pontil, 2005) together with the square loss give a closed-form solution of the above problem, for all $x \in \mathcal{X}$,

$$\hat{h}(x) = \sum_{i=1} \hat{\alpha}_i(x)\, \psi_\mathcal{Y}(y_i) \quad \text{with} \quad \hat{\alpha}(x) = (\mathrm{K}_\mathrm{X} + n\lambda I_n)^{-1}\, \mathrm{k}_\mathrm{X}^\mathrm{x}, \tag{2.32}$$

where $\mathrm{k}_\mathrm{X}^\mathrm{x} = (\mathrm{k}_\mathcal{X}(x, x_1), \ldots, \mathrm{k}_\mathcal{X}(x, x_n))^\top \in \mathbb{R}^n$. We recover the classical matrix operation $(\mathrm{K}_\mathrm{X} + n\lambda I_n)^{-1}$ induced by KRR. Moreover, $\hat{h}$ admits an operator expression $\hat{h} : x \mapsto \widehat{H}\, \psi_\mathcal{X}(x)$ for $\widehat{H} : \mathcal{H}_\mathcal{X} \to \mathcal{H}_\mathcal{Y}$, which is very useful to derive theoretical guarantees, see section 2.4 for more details.

Inference step. Equipped with the above $\hat{h}$, we then obtain as structured estimator

$$\hat{f} : x \mapsto \operatorname*{arg\,max}_{y \in \mathcal{Y}} \ \mathrm{k}_\mathrm{X}^{\mathrm{x}\,\top} (\mathrm{K}_\mathrm{X} + n\lambda I_n)^{-1}\, \mathrm{k}_\mathrm{Y}^\mathrm{y}, \tag{2.33}$$

where $\mathrm{k}_\mathrm{Y}^\mathrm{y} = (\mathrm{k}_\mathcal{Y}(y, y_1), \ldots, \mathrm{k}_\mathcal{Y}(y, y_n))^\top \in \mathbb{R}^n$. In practice, predictions are performed in a discrete fashion by searching in a candidate set $\mathcal{Y}_c \subseteq \mathcal{Y}$ of size n_c. Hence, performing predictions on a test set X_te of size n_te mainly implies computing

$$S := \underbrace{\mathrm{K}_\mathrm{X}^{\mathrm{te,tr}}}_{n_\mathrm{te} \times n} \underbrace{(\mathrm{K}_\mathrm{X} + n\lambda I_n)^{-1}}_{n \times n} \underbrace{\mathrm{K}_\mathrm{Y}^{\mathrm{tr},c}}_{n \times n_c}, \tag{2.34}$$

where $\mathrm{K}_\mathrm{X}^{\mathrm{te,tr}} = \Big(\mathrm{k}_\mathcal{X}(x_i^\mathrm{te}, x_j)\Big)_{1 \le i \le n_\mathrm{te}, 1 \le j \le n} \in \mathbb{R}^{n_\mathrm{te} \times n}$, and $\mathrm{K}_\mathrm{Y}^{\mathrm{tr},c} = \Big(\mathrm{k}_\mathcal{Y}(y_i, y_j^c)\Big)_{1 \le i \le n, 1 \le j \le n_c} \in \mathbb{R}^{n \times n_c}$. Then, for each $1 \le i \le n_\mathrm{te}$,

$$\hat{f}(x_i^\mathrm{te}) = y_j^c \quad \text{where} \quad j = \operatorname*{arg\,max}_{1 \le j \le n_c} S_{ij}. \tag{2.35}$$

We now discuss the four challenges presented in chapter 1.

1. Scalability to large datasets. At the training phase, $(\mathrm{K}_\mathrm{X} + n\lambda I_n)^{-1}$ induces $\mathcal{O}(n^3)$ time and $\mathcal{O}(n^2)$ space complexities, which is standard for KRR. Moreover, at the inference phase, computing S induces $\mathcal{O}(n n_\mathrm{te} n_c + n^2 \min(n_\mathrm{te}, n_c))$ time and $\mathcal{O}(n^2 + n n_\mathrm{te} + n n_c)$ space complexities. IOKR then scales very poorly to large n, i.e. large datasets, both at the training and inference phases. We give more details about existing methods to

scale kernel methods up in section 2.3, but most of them focus on scalar regression, i.e. $\mathcal{Y} = \mathbb{R}$. In chapter 4, we provide Sketched Input Sketched Output Kernel Regression: a scalable IOKR-based method that leverages sketching in both the input and output feature spaces to reduce both training and inference complexities while still maintaining good statistical accuracy.

2. Excess risk bounds. In addition to its versatility, another main advantage of IOKR is its theoretical foundations. In particular, Ciliberto et al. (2016, 2020) show the *comparison inequality*, stating than the excess risk of $\hat{f}$ is bounded by the excess risk of $\hat{h}$.

Theorem 2.19 (Ciliberto et al. 2020, Theorem 3)**.** *Let $\mathcal{Y}$ be a compact set, $\mathrm{k}_\mathcal{Y} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be a normalized p. d. kernel and $\psi_\mathcal{Y} : y \mapsto \mathrm{k}_\mathcal{Y}(\cdot, y)$ its canonical feature map. Let $h : \mathcal{X} \to \mathcal{H}_\mathcal{Y}$ be measurable and $f : \mathcal{X} \to \mathcal{Y}$ such that, for any $x \in \mathcal{X}$,*

$$f(x) = \operatorname*{arg\,min}_{y \in \mathcal{Y}} \|h(x) - \psi_\mathcal{Y}(y)\|^2_{\mathcal{H}_\mathcal{Y}}. \tag{2.36}$$

Then,

$$\mathcal{R}(f) - \mathcal{R}(f^*) \leq 12\sqrt{\mathcal{E}(h) - \mathcal{E}(h^*)}, \tag{2.37}$$

where $\mathcal{E}(h) = \mathbb{E}_{(x,y)\sim\rho}[\|h(x) - \psi_\mathcal{Y}(y)\|^2_{\mathcal{H}_\mathcal{Y}}]$.

In section 2.4.2, we give the sketch of proof to derive the excess risk bound of $\hat{h}$, and then of $\hat{f}$ tanks to the above comparison inequality. However, incorporating sketching approximation makes it obsolete. We derive an excess risk for the scalable SISOKR estimator in chapter 4.

3. Various losses. The closed-form formula for $\hat{h}$ is obtained thanks to the input identity decomposable kernel and the square loss. However, in the case where one would use other losses, such as robust losses in the case of output outliers, it seems very difficult to learn $\hat{h}$ in general since $\mathcal{H}_\mathcal{Y}$ may be infinite-dimensional. Laforgue et al. (2020) shows that under some assumptions on the loss and the input kernel, $\hat{h}$ admits a parameterized expression whose parameters can be learned through a dual problem which is solvable by a projected gradient descent problem. In particular, they derive such dual problems for the ϵ-insensitive ℓ_1 and ℓ_2 losses, as well as for the Huber loss, see section 2.5 for more details. In chapter 4, we show that we can obtain a small orthonormal basis $\mathrm{span}((e_i)_{i=1}^p)$ of a subspace of $\mathcal{H}_\mathcal{Y}$ for some $p \in \mathbb{N}^*$, which is formed by the eigenfunctions of the sketched output empirical covariance operator. Then, in chapter 5, and building upon this result, we show that we can learn $\hat{h}$ for any loss $\Delta : (y, y') = c(\|\psi_\mathcal{Y}(y) - \psi_\mathcal{Y}(y')\|^2_{\mathcal{H}_\mathcal{Y}})$ with a differentiable or sub-differentiable $c : \mathbb{R} \to \mathbb{R}$ $\hat{h}$, by setting $\hat{h} : x \mapsto \sum_{i=1}^p \hat{g}(x)_i e_i$ where $\hat{g} : \mathcal{X} \to \mathbb{R}^p$ is either a linear function, a function induced by a matrix-valued kernel or a neural network. In fact, for such choices, it is possible to learn $\hat{g}$ by solving the surrogate ERM problem by gradient-based algorithms. Note that the ϵ-insensitive ℓ_2 and Huber loss write as $c(\|\psi_\mathcal{Y}(y) - \psi_\mathcal{Y}(y')\|^2_{\mathcal{H}_\mathcal{Y}})$ with differentiable functions c, while for the ϵ-insensitive ℓ_1, c is sub-differentiable.

4. Handle complex input data. IOKR strongly relies on an input kernel, however kernels show poor results on complex input data such as images or texts. Such inputs need more expressive models that have the ability to learn representations from them without prior knowledge, see section 2.6 for more details. As explained above, we introduce in chapter 5 Deep Sketched Output Kernel Regression, a deep architecture that exploits the basis obtained by the SVD of the sketched output empirical covariance operator.

2.2.2 Overview of Other Methods

We here give a brief background on some existing methods for structured prediction. We refer the reader to Bakir et al. (2007), Nowozin and Lampert (2011) and Deshwal et al. (2019) for deeper details, and to Nowak-Vila et al. (2019) and Nowak et al. (2020) for the theoretical analysis of CRF, SSVM, and M^3 networks.

Conditional Random Fields

CRF (Lafferty et al., 2001) generalizes logistic regression classifiers to structured prediction.

The idea is to compute the conditional probability $p(y \mid x)$ to obtain the prediction via Maximum A Posteriori (MAP), i.e. by maximizing the conditional probability over $\mathcal{Y}$ for a given input $x \in \mathcal{X}$, hence CRF falls into the scope of energy-based models with $E : (x, y) \mapsto p(y \mid x)$. This conditional probability is modeled by a parameterized graphical model, i.e. such that, for all $x \in \mathcal{X}$, $y \in \mathcal{Y}$,

$$p_\omega(y \mid x) = \frac{1}{Z(x, \omega)} \exp(-g_\omega(x, y)), \tag{2.38}$$

where $\omega \in \mathbb{R}^d$ for $d \in \mathbb{N}^*$ is the parameter to learn, $Z(x, \omega)$ is the normalisation factor ensuring that $\sum_{y \in \mathcal{Y}} p_\omega(y \mid x) = 1$, and g_ω encodes the input/output dependency through ω. A standard choice is $g_\omega : (x, y) = \omega^\top \Psi(x, y)$ with a joint feature map.

To learn ω, the goal is to make $p_\omega(y \mid x)$ close to the true conditional distribution by maximizing the regularized conditional log-likelihood, then by solving, for $\lambda > 0$,

$$\min_{\omega \in \mathbb{R}^d} \sum_{i)1}^{n} \omega^\top \Psi(x_i, y_i) + \sum_{i=1}^{n} Z(x_i, \omega) + \lambda \|\omega\|_2^2. \tag{2.39}$$

This convex optimisation problem can then be easily solved through a gradient descent algorithm.

The main limitation comes from the normalisation term Z obtained via a sum over $\mathcal{Y}$ whose size is usually very large for structured outputs. However, one can consider computationally efficient approximations based on the graphical model's structure, e.g. by the belief propagation algorithm (Nowozin and Lampert, 2011).

Structured support vector machines

SSVM (Tsochantaridis et al., 2004, 2005; Taskar et al., 2005) generalizes SVM to structured prediction, see section 2.5 for more details.

It uses joint feature maps $\Psi : \mathcal{X} \times \mathcal{Y} \to \mathcal{H}$ as well and computes an inference for an input $x \in \mathcal{X}$ by maximizing the inner product between a vector $\omega \in \mathcal{H}$ to learn and the joint feature map evaluation over the output space,

$$f(x) = \underset{y \in \mathcal{Y}}{\arg\max}\ E(x,y) = \underset{y \in \mathcal{Y}}{\arg\max}\ \langle \omega, \Psi(x,y) \rangle_{\mathcal{H}}. \tag{2.40}$$

Then, similarly to SVM, the idea is to learn ω via the following soft-margin problem,

$$\begin{aligned} &\min_{\omega \in \mathcal{H}, \xi \in \mathbb{R}^n}\ \frac{1}{2}\|\omega\|_2^2 + C\sum_{i=1}^{n} \xi_i \\ &\text{such that} \quad \xi_i \geq 0, \langle \omega, \Psi(x_i, y_i) - \Psi(x_i, y) \rangle_{\mathcal{H}} \geq 1 - \xi_i, \forall i \in [\![n]\!], \forall y \in \mathcal{Y} \setminus \{y_i\}. \end{aligned}$$

As for standard SVM, such a problem benefits from a parameterized dual problem and KKT condition to learn ω. Note also that such a problem corresponds to solving the primal classical ERM problem on f defined as in eq. (2.40) with the following loss evaluations, for $i \in [\![n]\!]$,

$$\Delta(f(x_i), y_i) = \max(1 + f(x_i) - \langle \omega, \Psi(x_i, y_i) \rangle_{\mathcal{H}}, 0), \tag{2.41}$$

and we will see that other methods exploit such losses (Belanger and McCallum, 2016).

Akin to CRF, the inference problem over the whole output space $\mathcal{Y}$ creates a computational burden, but various algorithms have been proposed to cope with this problem, e.g. the cutting plane algorithm (Nowozin and Lampert, 2011).

Max-margin Markov networks

Max-Margin Markov networks can be considered as a combination of CRF and SSVM. In fact, to have more efficient learning algorithms for the energy function E, it is then defined as a graphical model but trained via a max margin-based optimisation problem.

Structured Prediction Energy Networks

We here present SPEN and some deep-learning-based methods inspired or derived from it. For some reminders about deep learning, we refer the reader to section 2.6.2.

SPEN focuses on the case where the output space can be encoded via L-sized binary vectors, for $L \in \mathbb{N}^*$, and then uses the continuous relaxation $\bar{\mathcal{Y}} = [0,1]^L$ of $\mathcal{Y} = \{0,1\}^L$. Moreover, it uses deep neural networks to encode an energy function E. This energy function E is the sum of two energy functions. First, the local energy network E^{local} encodes the relationships between the inputs and outputs, i.e.

$$E^{\text{local}} : (x, \bar{y}) \in \mathcal{X} \times \bar{\mathcal{Y}} \mapsto \sum_{i=1}^{L} \bar{y}_i b_i^\top F(x), \tag{2.42}$$

where $F : x \mapsto g(A_2 g(A_1 x)) \in \mathbb{R}^f$, for $f \in \mathbb{N}^*$, is a 2-layer feature networks, with g being activation functions and A_1, A_2 and $(b_i)_{i=1}^L$ being the neural nets' weights to learn. Second, the global energy network E^{label} encodes the relationships between the outputs' components, and is then independent from the inputs, i.e.

$$E^{\text{label}} : \bar{y} \in \bar{\mathcal{Y}} \mapsto c_2^\top g(C_1 \bar{y}), \tag{2.43}$$

where g is an activation function as well, and C_1 and c_2 weights to learn. Then, given $\ell : \mathcal{Y}^2 \mathbb{R}\mathbb{R}$ an error function such as the Binary Cross Entropy and $E = E^{\text{local}} + E^{\text{label}}$, SPEN learns its eights by minimizing an SSVM loss function type,

$$\sum_{i=1}^{n} \max_{y} \max(\ell(y_i, y) - E(x_i, y) + E(x_i, y_i), 0), \tag{2.44}$$

thanks to mini-batch stochastic gradient descent and back-propagation. The inference is obtained via projected gradient descent thanks to the continuous relaxation of $\mathcal{Y}$ by solving

$$f : x \in \mathcal{X} \mapsto \arg\min_{y \in \mathcal{Y}} E(x, y). \tag{2.45}$$

One of the limitations of SPEN is its two-step process: one has to first learn the energy network via gradient descent to then performs an inference via another gradient descent. To build an end-to-end model, Belanger et al. (2017) uses a direct risk minimisation technique, stating that a prediction $\hat{y}$ induced by a gradient descent algorithm with an initialisation y_0, T epochs and learning rates $(\eta_t)_{t=1}^{T}$ is given by

$$\hat{y_T} = y_0 - \sum_{t=1}^{T} \eta_t \frac{d}{dy} E(x, y_t). \tag{2.46}$$

However, while the authors propose to choose y_0 by pre-training the feature network, it does not appear straightforward how to a priori choose T and $(\eta_t)_{t=1}^{T}$. Another solution proposed by Tu and Gimpel (2018) is to learn an inference network f via an approximated structured arg max inference. The energy and inference networks are then jointly trained in a GAN (Goodfellow et al., 2014) inspired fashion, which however can lead to heavy computations.

Deep Value Networks (Gygli et al., 2017) is also another gradient-based approach using the same architecture and relaxation as SPEN. The main difference comes from the training phase, when DVN explores other possibilities than generating outputs via gradient-based inference, such as adversarial tuples or random samples.

Finally, the other main limitation of SPENs relies on the considered continuous relaxation, narrowing SPENs' scope to structured tasks that can be formulated as multi-label classification. Graber et al. (2018) then proposes to formulate inference via a Lagrangian, instead of using a continuous relaxation of the output space.

In chapter 5, we show how to introduce kernel-induced losses to neural networks, or equivalently, neural networks to OKR, and then provide a versatile deep-learning-based structured prediction model in terms of the output types.

2.3 Scalability of Kernel Methods

We here present some approximation methods to tackle large-scale learning with kernels. We first introduce Random Fourier Features. Then, we present Sketching, starting with its particular most famous instance Nyström approximation

2.3.1 Random Fourier Features

The idea behind random features is to approximate the canonical feature map $\psi_{\mathcal{X}}$ of a scalar-valued p. d. kernel $\mathrm{k}_{\mathcal{X}}$ by a randomly generated feature map $\tilde{\psi}_{\mathcal{X}} : \mathcal{X} \to \mathbb{R}^{\mathrm{m}_{\mathcal{X}}}$ where $\mathrm{m}_{\mathcal{X}} \ll n$, i.e. for all $x, x' \in \mathcal{X}$,

$$\mathrm{k}_{\mathcal{X}}(x, x') = \langle \psi_{\mathcal{X}}(x), \psi_{\mathcal{X}}(x') \rangle_{\mathcal{H}_{\mathcal{X}}} \approx \tilde{\psi}_{\mathcal{X}}(x)^\top \tilde{\psi}_{\mathcal{X}}(x'). \tag{2.47}$$

Equipped with this approximation, we can then set as the hypothesis space to solve an ERM problem, the set of linear functions $f(\cdot) = \gamma^\top \tilde{\psi}_{\mathcal{X}}(\cdot)$ where $\gamma \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}}}$ is the solution to

$$\min_{\gamma \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}}}} \frac{1}{n} \sum_{i=1}^{n} \ell(\gamma^\top \tilde{\psi}_{\mathcal{X}}(x_i), y_i) + \lambda \|\gamma\|_2^2. \tag{2.48}$$

We then obtain an optimisation problem over $\mathrm{m}_{\mathcal{X}}$ parameters rather than the initial n parameters of eq. (2.10). A good example of the complexity reduction induced by such an approximation is the KRR.

Example 2.20 (Random features KRR). *Let $\ell : (y, y') \mapsto (y - y')^2$, eq. (2.48) rewrites*

$$\min_{\gamma \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}}}} \frac{1}{n} \|\tilde{\psi}_{\mathcal{X}}(X)\gamma - Y\|_2^2 + \lambda \|\gamma\|_2^2, \tag{2.49}$$

where $\tilde{\psi}_{\mathcal{X}}(X) = (\tilde{\psi}_{\mathcal{X}}(x_1), \ldots, \tilde{\psi}_{\mathcal{X}}(x_n))^\top \in \mathbb{R}^{n \times \mathrm{m}_{\mathcal{X}}}$. Thus, setting the gradient to zero, we obtain as a solution to the above problem

$$\gamma = \underbrace{(\tilde{\psi}_{\mathcal{X}}(X)^\top \tilde{\psi}_{\mathcal{X}}(X) + n\lambda I_{\mathrm{m}_{\mathcal{X}}})^{-1}}_{\mathrm{m}_{\mathcal{X}} \times \mathrm{m}_{\mathcal{X}}} \tilde{\psi}_{\mathcal{X}}(X)^\top Y. \tag{2.50}$$

It is clear that inverting an $\mathrm{m}_{\mathcal{X}}{}^2$-matrix rather than the initial n^2-matrix induces a huge complexity reduction for $\mathrm{m}_{\mathcal{X}} \ll n$.

The question now is: how can we build such random features such that they ensure a good approximation of the initial kernel?

The most popular approach is Random Fourier Features (Rahimi and Recht, 2007). RFF takes its roots into Bochner's theorem, stating that a continuous function defined on $\mathbb{R}^d$ for $d \in \mathbb{N}^*$ is p. d. if and only if it is the Fourier transform of a non-negative measure. As a consequence, one can approximate any continuous *shift-invariant* kernel thanks to Monte-Carlo sampling.

We detail this approach. First, we restrict the input space space $\mathcal{X} \subseteq \mathbb{R}^d$ for $d \in \mathbb{N}^*$. Let us define *shift-invariant* kernels.

Definition 2.21 (shift-invariant kernel). *Let $\mathrm{k}_{\mathcal{X}} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a p. d. kernel. We say that $\mathrm{k}_{\mathcal{X}}$ is shift-invariant if there exists a function $\mathrm{k}_{\mathcal{X}}{}^0$ on $\mathcal{X}$ such that for all $x, x' \in \mathcal{X}$, $\mathrm{k}_{\mathcal{X}}(x, x') = \mathrm{k}_{\mathcal{X}}{}^0(x - x')$.*

We now state the Bochner's theorem.

Theorem 2.22 (Bochner). *A continuous function $\mathrm{k}_{\mathcal{X}}{}^0 : \mathbb{R}^d \to \mathbb{R}$ is positive definite if and only if it is the Fourier transform of a finite non-negative Borel measure μ on $\mathbb{R}^d$.*

$$\mathrm{k}_{\mathcal{X}}{}^0(x) = \int_{\mathbb{R}^d} e^{i\omega^\top x} \mathrm{d}\mu(\omega) = \int_{\mathbb{R}^d} \cos(\omega^\top x) \mathrm{d}\mu(\omega). \tag{2.51}$$

The second equality is true since ${\mathrm{k}_{\mathcal{X}}}^0$ is real-valued and μ is defined on $\mathbb{R}^d$, therefore we can ignore the imaginary part. By the Bochner's theorem, we then obtain that, for all $x, x' \in \mathbb{R}^d$,

$$\mathrm{k}_{\mathcal{X}}(x,x') = \int_{\mathbb{R}^d} \cos(\omega^\top (x-x'))\mathrm{d}\mu(\omega) \tag{2.52}$$

$$= \int_{\mathbb{R}^d} (\cos(\omega^\top x)\cos(\omega^\top x') + \sin(\omega^\top x)\sin(\omega^\top x'))\mathrm{d}\mu(\omega). \tag{2.53}$$

As a consequence, this integral can be approximated by Monte Carlo. Let $(\omega_i)_{i=1}^{\mathrm{m}_{\mathcal{X}}/2}$ be i. i. d. draws from the probability measure μ, the approximate kernel is given by

$$\tilde{k}_{\mathcal{X}}(x,x') = \frac{2}{\mathrm{m}_{\mathcal{X}}} \sum_{i=1}^{\mathrm{m}_{\mathcal{X}}/2} \cos(\omega_i^\top x)\cos(\omega_i^\top x') + \sin(\omega_i^\top x)\sin(\omega_i^\top x') = \tilde{\psi}_{\mathcal{X}}(x)^\top \tilde{\psi}_{\mathcal{X}}(x'), \tag{2.54}$$

where

$$\tilde{\psi}_{\mathcal{X}} : x \mapsto \sqrt{\frac{2}{\mathrm{m}_{\mathcal{X}}}}(\cos(\omega_1^\top x), \ldots, \cos(\omega_{\mathrm{m}_{\mathcal{X}}/2}^\top x), \sin(\omega_1^\top x), \ldots, \sin(\omega_{\mathrm{m}_{\mathcal{X}}/2}^\top x))^\top. \tag{2.55}$$

By noting that, for all $b \in \mathbb{R}$, $2\cos(\omega^\top x + b)\cos(\omega^\top x' + b) = \cos(\omega^\top (x+x') + 2b) + \cos(\omega^\top (x - x'))$ and $\int_0^{2\pi} \cos(\omega^\top (x+x') + 2b) = 0$, we recover the expression provided in Rahimi and Recht (2007), i.e.

$$\mathrm{k}_{\mathcal{X}}(x,x') = \int_{\mathbb{R}^d} \int_0^{2\pi} \frac{1}{\pi} \cos(\omega^\top x + b)\cos(\omega^\top x' + b)\mathrm{d}\mu(\omega)\mathrm{d}b, \tag{2.56}$$

and then, by Monte Carlo as well, we obtain by sampling $\mathrm{m}_{\mathcal{X}}$ i. i. d. realisations ω_is from μ and b_is uniformly from $[0, 2\pi]$

$$\tilde{\psi}_{\mathcal{X}} : x \mapsto \frac{1}{\sqrt{\pi\, \mathrm{m}_{\mathcal{X}}}}(\cos(\omega_1^\top x + b_1), \ldots, \cos(\omega_{\mathrm{m}_{\mathcal{X}}}^\top x + b_{\mathrm{m}_{\mathcal{X}}}))^\top. \tag{2.57}$$

The most famous example of shift-invariant kernel is undoubtedly the Gaussian kernel.

Example 2.23 (Gaussian kernel). *Let* $\mathrm{k}_{\mathcal{X}} : x \times x' \mapsto \exp(-\frac{1}{2\sigma^2}\|x - x'\|_2^2)$ *be a Gaussian kernel for* $\sigma > 0$. *Its inverse Fourier transform is a Gaussian as well, more precisely, it admits an RFF where the* ω_i*s are randomly drawn from* $\mathcal{N}(0, 1/\sigma^2)$.

More generally, as pointed out by Rudi and Rosasco (2017) that gives some examples, it is possible to use RFF as long as a kernel admits an integral representation,

$$\mathrm{k}_{\mathcal{X}}(x,x') = \int_{\Omega} \tilde{\psi}_{\mathcal{X}}(x,\omega)\tilde{\psi}_{\mathcal{X}}(x',\omega)\mathrm{d}\mu(\omega), \tag{2.58}$$

for all $x, x' \in \mathcal{X}$ and where (Ω, μ) is a probability space and $\tilde{\psi}_{\mathcal{X}} : \mathcal{X} \times \Omega \to \mathbb{R}$. Sriperumbudur and Szabó (2015) provides a thorough analysis of approximation's quality inferred by RFF in terms of the supremum norm of the difference between $\tilde{k}_{\mathcal{X}}$ and $\mathrm{k}_{\mathcal{X}}$ for any compact set included in $\mathbb{R}^d$. Brault et al. (2016) generalizes RFF to shift-invariant OVKs, unlocking large-scale learning with RFF in the context of regression within a generic Hilbert space.

RFF is then a kernel approximation technique relying on random features. Note that these random features are independent of the training data and their expression only depends on the kernel and its integral representation. We will see in what follows that another popular technique provides random features, but these random features now depend on the training samples.

2.3.2 Nyström Approximation

The Nyström approximation is originally a low-rank approximation technique, where a positive semi-definite matrix is approximated by a low-rank matrix based on a sub-matrix of the original one. It then appears intuitive that most of the works about the scalability of kernels focus on Nyström, as sub-sampling from the training data reduces not only the time but also the space complexity induced by the n^2 Gram matrix. Many interpretations of scalar Nyström kernels come from its rich literature, namely the low-rank approximation of the Gram matrix (Drineas et al., 2005; Bach, 2013), the data-dependent random features (Williams and Seeger, 2001; Yang et al., 2012), the reduction of the hypothesis space (Rudi et al., 2015), or the orthogonal projection operator in the feature space (Rudi et al., 2015). We will present all of them and the links between them.

Low-rank approximation and data-dependent random features. Let $\mathrm{k}_{\mathcal{X}}$ be scalar p. d. kernel and $\{(x_i)_{i=1}^n\}$ be a training sample. Let $\mathrm{m}_{\mathcal{X}} \ll n$ and $\{(\tilde{x}_i)_{i=1}^{\mathrm{m}_{\mathcal{X}}}\}$ sampled from the training outputs, $\widetilde{\mathrm{K}}_{\mathrm{X}} = (\mathrm{k}_{\mathcal{X}}(\tilde{x}_i, \tilde{x}_j))_{1\leq i,j\leq \mathrm{m}_{\mathcal{X}}}$ and $\mathrm{K}_{\mathrm{X}n\mathrm{m}_{\mathcal{X}}} = (\mathrm{k}_{\mathcal{X}}(x_i, \tilde{x}_j))_{1\leq i\leq n, 1\leq j\leq \mathrm{m}_{\mathcal{X}}}$ be the ${\mathrm{m}_{\mathcal{X}}}^2$ and $n\times \mathrm{m}_{\mathcal{X}}$ sub-matrices, respectively, the Nyström low-rank approximation of the kernel Gram matrix K_{X} is then

$$\mathrm{K}_{\mathrm{X}n\mathrm{m}_{\mathcal{X}}} \widetilde{\mathrm{K}}_{\mathrm{X}}^{\dagger} \mathrm{K}_{\mathrm{X}n\mathrm{m}_{\mathcal{X}}}^{\top}. \tag{2.59}$$

Drineas et al. (2005) shows that the randomized approximation in eq. (2.59) gets very close with high probability and in expectation to the best rank $\mathrm{m}_{\mathcal{X}}$ approximation to K_{X}. Interestingly, Williams and Seeger (2001) first introduced such an approximation to obtain data-dependent random features. Indeed, let $\{(\sigma_i(\widetilde{\mathrm{K}}_{\mathrm{X}}), \tilde{\mathbf{u}}_i), i \in [\mathrm{m}_{\mathcal{X}}]\}$ be the eigenpairs of $\widetilde{\mathrm{K}}_{\mathrm{X}}$ in descending order, $\mathrm{p}_{\mathrm{X}} = \mathrm{rank}(\widetilde{\mathrm{K}}_{\mathrm{X}}) \leq \mathrm{m}_{\mathcal{X}}$, $\tilde{D}_{\mathrm{p}_{\mathrm{X}}} = \mathrm{diag}(\sigma_1(\widetilde{\mathrm{K}}_{\mathrm{X}}), \ldots, \sigma_{\mathrm{p}_{\mathrm{X}}}(\widetilde{\mathrm{K}}_{\mathrm{X}})) \in \mathbb{R}^{\mathrm{p}_{\mathrm{X}}\times \mathrm{p}_{\mathrm{X}}}$, and $\tilde{U}_{\mathrm{p}_{\mathrm{X}}} = (\tilde{\mathbf{u}}_1, \ldots, \tilde{\mathbf{u}}_{\mathrm{p}_{\mathrm{X}}}) \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}}\times \mathrm{p}_{\mathrm{X}}}$, we have that

$$\mathrm{K}_{\mathrm{X}n\mathrm{m}_{\mathcal{X}}} \widetilde{\mathrm{K}}_{\mathrm{X}}^{\dagger} \mathrm{K}_{\mathrm{X}n\mathrm{m}_{\mathcal{X}}}^{\top} = \mathrm{K}_{\mathrm{X}n\mathrm{m}_{\mathcal{X}}} \tilde{U}_{\mathrm{p}_{\mathrm{X}}} \tilde{D}_{\mathrm{p}_{\mathrm{X}}}^{-1/2} \tilde{D}_{\mathrm{p}_{\mathrm{X}}}^{-1/2} \tilde{U}_{\mathrm{p}_{\mathrm{X}}}^{\top} \mathrm{K}_{\mathrm{X}n\mathrm{m}_{\mathcal{X}}}^{\top} = \tilde{\psi}_{\mathcal{X}}(X)^{\top} \tilde{\psi}_{\mathcal{X}}(X), \tag{2.60}$$

where $\tilde{\psi}_{\mathcal{X}}(X) = (\tilde{\psi}_{\mathcal{X}}(x_1), \ldots, \tilde{\psi}_{\mathcal{X}}(x_n))^{\top} \in \mathbb{R}^{n\times \mathrm{p}_{\mathrm{X}}}$ with

$$\tilde{\psi}_{\mathcal{X}} : x \mapsto \tilde{D}_{\mathrm{p}_{\mathrm{X}}}^{-1/2} \tilde{U}_{\mathrm{p}_{\mathrm{X}}}^{\top} k_{\tilde{X}}^{x}, \tag{2.61}$$

where $k_{\tilde{X}}^{x} = (\mathrm{k}_{\mathcal{X}}(x, \tilde{x}_1), \ldots, \mathrm{k}_{\mathcal{X}}(x, \tilde{x}_{\mathrm{m}_{\mathcal{X}}}))^{\top} \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}}}$.

Remark 2.24 (Rank of $\widetilde{\mathrm{K}}_{\mathrm{X}}$)**.** *In general, a Gram matrix is invertible if all the entries used to compute it are unique. Hence, if all the training inputs are unique and the $\tilde{x}_i$s are sampled without replacement,* $\mathrm{p}_{\mathrm{X}} = \mathrm{m}_{\mathcal{X}}$.

As a consequence, Nyström can be used to compute random features whose dimension is smaller than n as RFF, however, their computations depend on the training data sampled. Moreover, another notable difference is that such an operation can be done for all kernels, even the ones not admitting an integral representation. Finally, Yang et al. (2012) compared Nyström random features with RFF, and their excess risk bounds highlighted that, when there is a large gap in the eigenspectrum of the kernel matrix, approaches based on the Nyström method can achieve better results than Random Features based approach.

Reduction of the hypothesis space. We saw that, thanks to the represented theorem, any solution of the penalized ERM problem in eq. (2.7) lies in $\mathrm{span}((\mathrm{k}_{\mathcal{X}}(\cdot, x_i)_{i=1}^n)$. A natural idea is then to restrict the hypothesis space to $\mathrm{span}((\mathrm{k}_{\mathcal{X}}(\cdot, \tilde{x}_i)_{i=1}^{\mathrm{m}_{\mathcal{X}}})$. Then, the Nyström estimator $\tilde{f} = \sum_{i=1}^{\mathrm{m}_{\mathcal{X}}} \tilde{\gamma}_i \mathrm{k}_{\mathcal{X}}(\cdot, x_i)$ where $\tilde{\gamma} = (\tilde{\gamma}_1, \dots, \tilde{\gamma}_{\mathrm{m}_{\mathcal{X}}})^\top \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}}}$ is the solution to

$$\min_{\gamma \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}}}} \frac{1}{n} \sum_{i=1}^{n} \ell([\mathrm{K}_{X n \mathrm{m}_{\mathcal{X}}} \gamma]_i, y_i) + \lambda \gamma^\top \widetilde{\mathrm{K}}_{\mathrm{X}} \gamma \,. \tag{2.62}$$

We then obtain an optimisation problem with $\mathrm{m}_{\mathcal{X}}$ parameters to learn rather than n. The connection between this interpretation and the Nyström features is not straightforward. Let us define the linear problem induced by the random features,

$$\min_{\gamma \in \mathbb{R}^{\mathrm{pX}}} \frac{1}{n} \sum_{i=1}^{n} \ell(\gamma^\top \tilde{\psi}_{\mathcal{X}}(x_i), y_i) + \lambda \|\gamma\|_2^2 \,. \tag{2.63}$$

As we show in the proof of Proposition 3.17 in chapter 3 for any sketching distribution, including Nyström approximation, problems (2.62) and (2.63) admit the same following dual problem

$$\min_{\zeta \in \mathbb{R}^n} \sum_{i=1}^{n} \ell_i^\star\left(-\zeta_i\right) + \frac{1}{\lambda n} \zeta^\top \mathrm{K}_{X n \mathrm{m}_{\mathcal{X}}} \widetilde{\mathrm{K}}_{\mathrm{X}}^{\dagger} \mathrm{K}_{X n \mathrm{m}_{\mathcal{X}}}^\top \zeta \,. \tag{2.64}$$

where $\ell_i^\star$ denotes the Fenchel-Legendre transform of $\ell_i : y \in \mathbb{R} \mapsto \ell(y, y_)$ for any $i \leq n$. Hence, in the case where strong duality holds, problems (2.62) and (2.63) admit the same minimal values, and their solutions can be linked thanks to Karush-Kuhn-Tucker conditions. Moreover, note that we recover the low-rank Nyström approximation matrix in the regularisation of the dual problem.

Orthogonal projection operator. Last but not least, Rudi et al. (2015) provides an excess risk bound of the Nyström KRR estimator obtained by solving eq. (2.62) thanks to an analysis based on the induced orthogonal projector onto $\mathrm{span}((\mathrm{k}_{\mathcal{X}}(\cdot, \tilde{x}_i)_{i=1}^{\mathrm{m}_{\mathcal{X}}})$. First, let $S_{\tilde{X}} : f \in \mathcal{H}_{\mathcal{X}} \mapsto (1/\sqrt{\mathrm{m}_{\mathcal{X}}})(f(\tilde{x}_1), \dots, f(\tilde{x}_{\mathrm{m}_{\mathcal{X}}}))^\top \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}}}$ be the sub-sampling operator and $\widetilde{\mathrm{C}}_{\mathrm{X}} = (1/\mathrm{m}_{\mathcal{X}}) \sum_{i=1}^{\mathrm{m}_{\mathcal{X}}} \psi_{\mathcal{X}}(\tilde{x}_i) \otimes \psi_{\mathcal{X}}(\tilde{x}_i)$ be the Nyström approximation of the empirical covariance operator $\widehat{\mathrm{C}}_{\mathrm{X}}$. Yang et al. (2012) shows that the non-zero eigenfunctions of $\widetilde{\mathrm{C}}_{\mathrm{X}}$ are

$$\tilde{e}_i = \sqrt{\frac{n}{\sigma_i(\widetilde{\mathrm{K}}_{\mathrm{X}})}} S_{\tilde{X}}^{\#} \tilde{\mathbf{u}}_i \,, \tag{2.65}$$

associated to the eigenvalues $\sigma_i(\widetilde{\mathrm{K}}_{\mathrm{X}})/n$, for $1 \leq i \leq \mathrm{p}_{\mathrm{X}}$.

Remark 2.25 (Link with random features)**.** *As pointed out by Yang et al. (2012), note that the random features $\tilde{\psi}_{\mathcal{X}}$ are simply the vector formed by the evaluations of the $\tilde{e}_i$s, i.e.*

$$\tilde{\psi}_{\mathcal{X}} : x \mapsto (\tilde{e}_1(x), \dots, \tilde{e}_{\mathrm{p}_{\mathrm{X}}}(x))^\top \,. \tag{2.66}$$

Since $\mathrm{Im}(\widetilde{\mathrm{C}}_{\mathrm{X}}) = \mathrm{span}((\mathrm{k}_{\mathcal{X}}(\cdot, \tilde{x}_i)_{i=1}^{\mathrm{m}_{\mathcal{X}}})$, $\mathrm{span}((\tilde{e}_i)_{i=1}^{\mathrm{p}_{\mathrm{X}}})$ is an orthonormal basis of $\mathrm{span}((\mathrm{k}_{\mathcal{X}}(\cdot, \tilde{x}_i)_{i=1}^{\mathrm{m}_{\mathcal{X}}})$, we can then compute the expression of the orthogonal projector $\widetilde{\mathrm{P}}_{\mathrm{X}}$ onto the latter

$$\widetilde{\mathrm{P}}_{\mathrm{X}} = \sum_{i=1}^{\mathrm{p}_{\mathrm{X}}} \langle \cdot, \tilde{e}_i \rangle_{\mathcal{H}_{\mathcal{X}}} \tilde{e}_i = S_{\tilde{X}}^{\#} \left(S_{\tilde{X}} S_{\tilde{X}}^{\#} \right)^{\dagger} S_{\tilde{X}} \,. \tag{2.67}$$

We prove the above equation in chapter 4 for any sketching distribution. This is a powerful tool since it gives the understanding of the effects of Nyström approximation in the RKHS. It allowed us, generalized to any sketching distributions, to use sketching in the case of surrogate kernel methods where the output space is an RKHS, i.e. a possibly infinite-dimensional Hilbert space, unlike most of the works on sketched kernels that focus on scalar regression. We did it by using such projectors on the output kernel as well, see chapters 4 and 5, which is very convenient. Recently, Meanti et al. (2023) used such projectors on both the input and output feature spaces as well to learn large-scale dynamical systems.

Computationally, Nyström is very efficient since it prevents computing the whole n^2 Gram matrix and then having matrix multiplications whose time complexity is $\mathcal{O}(n^2)$, implying rather a $\mathcal{O}(n\,\mathrm{m}_{\mathcal{X}})$ space complexity because of $\mathrm{K}_{\mathrm{X}n\mathrm{m}_{\mathcal{X}}}$ and a time complexity linear in n.

Another crucial point about Nyström approximation is the sub-sampling strategy. The very first idea is obviously to uniformly sample $\{(\tilde{x}_i)_{i=1}^{\mathrm{m}_{\mathcal{X}}}\}$ from the initial $\{(x_i)_{i=1}^{n}\}$. This is very efficient but it can lead to poor results. For instance, assume that the inputs are sampled such that $x = bx_1 + (1-b)x_2$, where b is a Bernoulli random variable with parameter $p \in (0,1)$ and $x_1 \sim \rho_{\mathcal{X}_1}$, $x_2 \sim \rho_{\mathcal{X}_2}$ with $\rho_{\mathcal{X}_1}, \rho_{\mathcal{X}_2}$ two probability distributions over $\mathcal{X}$. If the sub-sampling is unbalanced between entries following $\rho_{\mathcal{X}_1}$ and $\rho_{\mathcal{X}_2}$, then the resulting estimator will be biased. Moreover, assume that p is close to 1 and $\rho_{\mathcal{X}_2}$ is an outlier distribution, then sampling too many entries $x \sim \rho_{\mathcal{X}_2}$ is not desirable. To cope with such limitations, many works have explored more elaborated and data-dependent sub-sampling strategies. First, giving more importance to entries based on their impacts on the eigendecomposition of K_{X}, namely the leverage scores, has attracted a lot of attention, but computing the SVD of K_{X} is computationally expensive. Hence, many works explored ways to compute relevant approximate leverage scores (Alaoui and Mahoney, 2015; Rudi et al., 2015; Musco and Musco, 2017; Rudi et al., 2018a; Cherfaoui et al., 2022). Kumar et al. (2012) also explores adaptive strategies. Finally, combined with other techniques, Nyström attains impressive performance. Rudi et al. (2017) first combines it with an efficient preconditioning and a stochastic gradient solver before Meanti et al. (2020) that adds a GPU-adapted implementation, enabling kernel methods to datasets with billions of samples.

2.3.3 Sketching

In addition to data-dependent sampling, and since Nyström approximation corresponds to a specific sketching distribution, i.e. the sub-sampling sketching, another solution is to consider other more statistically accurate sketching distributions. Before diving into deeper details about Nyström and sub-sampling sketching, let us introduce *sketching*.

Sketching (Mahoney et al., 2011; Woodruff, 2014) is a dimension reduction technique based on linear random projections. It has been leveraged in many machine learning fields, such as low-rank approximation (Gittens and Mahoney, 2016; Tropp et al., 2017) or optimisation (Pilanci and Wainwright, 2016; Gower et al., 2021). These linear random projections can then be encoded in a random matrix, and many matrix distributions exist with various computational and statistical properties. We will start with the sub-sampling sketching in order to show how to generalize Nyström approximation to any sketching distribution, and then we will present the Johnson-

Lindenstrauss lemma (Johnson and Lindenstrauss, 1984), a key foundation of sketching, and present some distributions that fit into this framework.

Sub-sampling sketching is Nyström approximation. In the following, for $\mathrm{m}_{\mathcal{X}} \ll n$, $\mathrm{R}_{\mathcal{X}} \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}} \times n}$ denotes the sketching matrix. We give the definition of a sub-sampling sketching matrix.

Definition 2.26 (Sub-sampling sketching)**.** *Let $(p_i)_{i=1}^n \in [0,1]^n$ such that $\sum_{i=1}^n p_i = 1$. Let $\{i_1, \ldots, i_{\mathrm{m}_{\mathcal{X}}}\}$ be $\mathrm{m}_{\mathcal{X}}$ indices drawn from $\{1, \ldots, n\}$ according to the probabilities p_is, with or without replacement. Then a sub-sampling sketching matrix $\mathrm{R}_{\mathcal{X}}$ is composed with rows $\mathrm{R}_{\mathcal{X}\,j:}$s such that, for all $1 \leq j \leq \mathrm{m}_{\mathcal{X}}$,*

$$\mathrm{R}_{\mathcal{X}\,j:} = \frac{1}{\sqrt{\mathrm{m}_{\mathcal{X}}\, p_{i_j}}} I_{i_j:}, \tag{2.68}$$

where $I_{i_j:}$ is the i_j-th row of the n^2 identity matrix I_n.

Remark 2.27 (Rows coefficients)**.** *Note that the rows coefficients $1/\sqrt{\mathrm{m}_{\mathcal{X}}\, p_{i_j}}$ are important to ensure that $\mathbb{E}_{\mathrm{R}_{\mathcal{X}}}[\mathrm{R}_{\mathcal{X}}{}^{\top} \mathrm{R}_{\mathcal{X}}] = I_n$, which is a desirable property of sketching matrices as we will see with the Johnson-Lindenstrauss lemma. However, for the sake of simplicity and to facilitate the comparison with Nyström approximation, we will not consider these coefficients in the following, i.e. for all $1 \leq j \leq \mathrm{m}_{\mathcal{X}}$,*

$$\mathrm{R}_{\mathcal{X}\,j:} = I_{i_j:}, \tag{2.69}$$

unless for the orthogonal projection part where such coefficients are needed.

It is called sub-sampling because the multiplication $\mathrm{R}_{\mathcal{X}} \cdot A$ of such a matrix with another matrix A results in a sub-matrix of A, where its rows at indices $\{i_1, \ldots, i_{\mathrm{m}_{\mathcal{X}}}\}$ are sampled. This means that for a Gram matrix K_{X}, $\mathrm{R}_{\mathcal{X}} \cdot \mathrm{K}_{\mathrm{X}} = \mathrm{K}_{\mathrm{X} n \mathrm{m}_{\mathcal{X}}}^{\top}$ according to the notations introduced in the Nyström's section. Let us give a quick example.

Example 2.28. *Let $n = 5$, $\mathrm{m}_{\mathcal{X}} = 2$, $p_i = 1/5$ $\forall 1 \leq i \leq 5$ and $i_1 = 1, i_2 = 4$. Then* $\mathrm{R}_{\mathcal{X}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$,

$$\mathrm{K}_{\mathrm{X} n \mathrm{m}_{\mathcal{X}}}^{\top} = \begin{pmatrix} \mathrm{k}_{\mathcal{X}}(x_1, x_1) & \mathrm{k}_{\mathcal{X}}(x_1, x_2) & \mathrm{k}_{\mathcal{X}}(x_1, x_3) & \mathrm{k}_{\mathcal{X}}(x_1, x_4) & \mathrm{k}_{\mathcal{X}}(x_1, x_5) \\ \mathrm{k}_{\mathcal{X}}(x_4, x_1) & \mathrm{k}_{\mathcal{X}}(x_4, x_2) & \mathrm{k}_{\mathcal{X}}(x_4, x_3) & \mathrm{k}_{\mathcal{X}}(x_4, x_4) & \mathrm{k}_{\mathcal{X}}(x_4, x_5) \end{pmatrix} = \mathrm{R}_{\mathcal{X}} \mathrm{K}_{\mathrm{X}}, \tag{2.70}$$

and

$$\widetilde{\mathrm{K}}_{\mathrm{X}} = \begin{pmatrix} \mathrm{k}_{\mathcal{X}}(x_1, x_1) & \mathrm{k}_{\mathcal{X}}(x_1, x_4) \\ \mathrm{k}_{\mathcal{X}}(x_4, x_1) & \mathrm{k}_{\mathcal{X}}(x_4, x_4) \end{pmatrix} = \mathrm{R}_{\mathcal{X}} \mathrm{K}_{\mathrm{X}} \mathrm{R}_{\mathcal{X}}{}^{\top}. \tag{2.71}$$

This means that we can reformulate all the above interpretations of Nyström in terms of the sketching matrix $\mathrm{R}_{\mathcal{X}}$. First, the low-rank approximated Gram matrix is

$$\mathrm{K}_{\mathrm{X}} \mathrm{R}_{\mathcal{X}}{}^{\top} (\mathrm{R}_{\mathcal{X}} \mathrm{K}_{\mathrm{X}} \mathrm{R}_{\mathcal{X}}{}^{\top})^{\dagger} \mathrm{R}_{\mathcal{X}} \mathrm{K}_{\mathrm{X}}. \tag{2.72}$$

Then, by defining $\widetilde{\mathrm{K}}_{\mathrm{X}} = \mathrm{R}_{\mathcal{X}} \mathrm{K}_{\mathrm{X}} \mathrm{R}_{\mathcal{X}}{}^{\top}$, we obtain the following random features expression

$$\tilde{\psi}_{\mathcal{X}} : x \mapsto \tilde{D}_{\mathrm{p_X}}^{-1/2} \tilde{U}_{\mathrm{p_X}}^{\top} \mathrm{R}_{\mathcal{X}}\, k_X^x. \tag{2.73}$$

Table 2.2: Summary of the different interpretations of sketching applied to kernel methods. Recall that Nyström approximation corresponds to using a sub-sampling sketching matrix $\mathrm{R}_{\mathcal{X}}$.

low-rank matrix	$\mathrm{K}_{\mathrm{X}}\mathrm{R}_{\mathcal{X}}{}^{\top}(\mathrm{R}_{\mathcal{X}}\mathrm{K}_{\mathrm{X}}\mathrm{R}_{\mathcal{X}}{}^{\top})^{\dagger}\mathrm{R}_{\mathcal{X}}\mathrm{K}_{\mathrm{X}}$
random feature	$\tilde{\psi}_{\mathcal{X}} : x \mapsto \tilde{D}_{\mathrm{p_X}}^{-1/2}\tilde{U}_{\mathrm{p_X}}^{\top}\mathrm{R}_{\mathcal{X}}k_X^x$
hypothesis space	$\mathrm{span}((\sum_{j=1}^{n}\mathrm{R}_{\mathcal{X}_{\mathrm{ij}}}\mathrm{k}_{\mathcal{X}}(\cdot,x_j))_{i=1}^{\mathrm{m}_{\mathcal{X}}})$
orthogonal projector	$\widetilde{\mathrm{P}}_{\mathrm{X}} = (\mathrm{R}_{\mathcal{X}}\mathrm{S}_{\mathrm{X}})^{\#}\left(\mathrm{R}_{\mathcal{X}}\mathrm{S}_{\mathrm{X}}(\mathrm{R}_{\mathcal{X}}\mathrm{S}_{\mathrm{X}})^{\#}\right)^{\dagger}\mathrm{R}_{\mathcal{X}}\mathrm{S}_{\mathrm{X}}$

Furthermore, by noting that for all $1 \leq j \leq \mathrm{m}_{\mathcal{X}}$, $\mathrm{k}_{\mathcal{X}}(\cdot,\tilde{x_j}) = \mathrm{k}_{\mathcal{X}}(\cdot,x_{i_j}) = \sum_{i=1}^{n}\mathrm{R}_{\mathcal{X}_{\mathrm{ji}}}\mathrm{k}_{\mathcal{X}}(\cdot,x_i)$, we obtain that $\mathrm{span}((\mathrm{k}_{\mathcal{X}}(\cdot,\tilde{x}_i)_{i=1}^{\mathrm{m}_{\mathcal{X}}}) = \mathrm{span}((\sum_{j=1}^{n}\mathrm{R}_{\mathcal{X}_{\mathrm{ij}}}\mathrm{k}_{\mathcal{X}}(\cdot,x_j))_{i=1}^{\mathrm{m}_{\mathcal{X}}})$. Problem (2.62) can then admits a sketching reformulation for the sketched estimator $\tilde{f} = \sum_{i=1}^{n}[\mathrm{R}_{\mathcal{X}}{}^{\top}\tilde{\gamma}]_i \cdot \mathrm{k}_{\mathcal{X}}(\cdot,x_i)$ where $\tilde{\gamma} = (\tilde{\gamma}_1,\ldots,\tilde{\gamma}_{\mathrm{m}_{\mathcal{X}}})^{\top} \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}}}$ is the solution to

$$\min_{\gamma\in\mathbb{R}^{\mathrm{m}_{\mathcal{X}}}} \frac{1}{n}\sum_{i=1}^{n}\ell([\mathrm{K}_{\mathrm{X}}\mathrm{R}_{\mathcal{X}}{}^{\top}\gamma]_i, y_i) + \lambda\gamma^{\top}\widetilde{\mathrm{K}}_{\mathrm{X}}\gamma\,. \tag{2.74}$$

Finally, note that for all $f \in \mathcal{H}_{\mathcal{X}}$, $S_{\tilde{X}}f = \frac{1}{\sqrt{\mathrm{m}_{\mathcal{X}}}}(f(\tilde{x}_1),\ldots,f(\tilde{x}_{\mathrm{m}_{\mathcal{X}}}))^{\top} = \frac{1}{\sqrt{\mathrm{m}_{\mathcal{X}}}}\mathrm{R}_{\mathcal{X}}(f(x_1),\ldots,f(x_n))^{\top} = \mathrm{R}_{\mathcal{X}}S_X$ and $\widetilde{\mathrm{C}}_{\mathrm{X}} = \frac{1}{\mathrm{m}_{\mathcal{X}}}\sum_{i=1}^{\mathrm{m}_{\mathcal{X}}}\psi_{\mathcal{X}}(\tilde{x}_i)\otimes\psi_{\mathcal{X}}(\tilde{x}_i) = S_{\tilde{X}}^{\#}S_{\tilde{X}}$, we then have that the sketched sampling operator is $\mathrm{R}_{\mathcal{X}}S_X$ and the corresponding sketched empirical covariance is given by $\widetilde{\mathrm{C}}_{\mathrm{X}} = (\mathrm{R}_{\mathcal{X}}S_X)^{\#}\mathrm{R}_{\mathcal{X}}S_X$. As a consequence, its eigenfunctions are $\tilde{e}_i = \sqrt{\frac{n}{\sigma_i(\widetilde{\mathrm{K}}_{\mathrm{X}})}}S_X^{\#}\mathrm{R}_{\mathcal{X}}{}^{\top}\tilde{\mathbf{u}}_i$ for all $i \in [\![\mathrm{m}_{\mathcal{X}}]\!]$, and then the sketched orthogonal projection operator is

$$\widetilde{\mathrm{P}}_{\mathrm{X}} = (\mathrm{R}_{\mathcal{X}}\mathrm{S}_{\mathrm{X}})^{\#}\left(\mathrm{R}_{\mathcal{X}}\mathrm{S}_{\mathrm{X}}(\mathrm{R}_{\mathcal{X}}\mathrm{S}_{\mathrm{X}})^{\#}\right)^{\dagger}\mathrm{R}_{\mathcal{X}}\mathrm{S}_{\mathrm{X}}\,. \tag{2.75}$$

Thanks to the Nyström approximation and its equivalence with sub-sampling sketching, many ways to leverage any sketching distribution for kernel methods are now unlocked, summarized in table 2.2. We now present the Johnson-Lindenstrauss to show the interest of considering other sketching distributions than sub-sampling.

Johnson-Lindenstrauss lemma. The J-L lemma is a well-known result of dimension reduction or compression. It states that a set of points in a high-dimensional space can be embedded into a space of much lower dimension in such a way that distances between the points are nearly preserved.

Lemma 2.29 (Johnson and Lindenstrauss (1984))**.** *Given $0 < \varepsilon < 1$, a set $\mathcal{S}$ of n points in $\mathbb{R}^D$, and an integer $d > 8(\log n)/\varepsilon^2$, there is a linear map $h : \mathbb{R}^D \to \mathbb{R}^d$ such that*

$$(1-\varepsilon)\|u-v\|^2 \leq \left\|h(u)-h(v)\right\|^2 \leq (1+\varepsilon)\|u-v\|^2\,, \tag{2.76}$$

for all $u, v \in \mathcal{S}$.

Proof [Sketch of proof.] The classical proof relies on random projection. Let $R = (1/\sqrt{d})(R_{ij})_{1\leq i\leq d, 1\leq j\leq D}$ such that the R_{ij}s are i. i. d. standard normal r. v., i.e. $R_{ij} \sim \mathcal{N}(0,1)$. Using the fact that, for all $u \in \mathbb{R}^D$,

$$\mathbb{E}_R[\|Ru\|_2^2] = u^{\top}\mathbb{E}_R[R^{\top}R]u = u^{\top}I_D u = \|u\|_2^2, \tag{2.77}$$

and the Markov inequality, one can show that eq. (2.76) is true with high probability. Boucheron et al. (2013) even shows that it is sufficient to consider sub-Gaussian R_{ij} to show such a result thanks to the Bernstein inequality, but with a higher lower bound on d. ■

Sketching distributions able to prove the J-L lemma are then of particular interest since it means that they ensure good statistical properties. We give two classical examples: the sub-Gaussian sketches, as considered in Boucheron et al. (2013), and the CountSketch (Clarkson and Woodruff, 2017), a well-known sketching distribution that defines sparse matrices.

Definition 2.30 (Sub-Gaussian sketching)**.** *A sub-Gaussian sketch* $\mathrm{R}_{\mathcal{X}} \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}} \times n}$ *is composed of i.i.d. entries such that* $\mathbb{E}\left[\mathrm{R}_{\mathcal{X}_{\mathrm{ij}}}\right] = 0$, $\mathbb{E}\left[{\mathrm{R}_{\mathcal{X}_{\mathrm{ij}}}}^2\right] = 1/\mathrm{m}_{\mathcal{X}}$ *and* $\mathrm{R}_{\mathcal{X}_{\mathrm{ij}}}$ *is* $\frac{{\nu_{\mathcal{X}}}^2}{\mathrm{m}_{\mathcal{X}}}$*-sub-Gaussian, for all* $1 \leq i \leq \mathrm{m}_{\mathcal{X}}$ *and* $1 \leq j \leq n$*, where* $\nu_{\mathcal{X}} \geq 1$.

Example 2.31 (Sub-Gaussian sketching distributions)**.**

- *A matrix composed with i. i. d. Gaussian r. v. such that* $\mathrm{R}_{\mathcal{X}_{\mathrm{ij}}} \sim \mathcal{N}(0, 1/\mathrm{m}_{\mathcal{X}})$ *is sub-Gaussian with* $\nu_{\mathcal{X}} = 1$.
- *By Hoeffding's lemma, any matrix composed with i. i. d. r. v. taking values in a bounded interval* $[a, b]$ *is* $(b-a)^2/4$*-sub-Gaussian.*
- *The p-sparsified sketches introduced in chapter 3 are also sub-Gaussian with* ${\nu_{\mathcal{X}}}^2 = 1/p$.

Definition 2.32 (CountSketch)**.** *Let* $\{i_1, \ldots, i_n\}$ *be n indices uniformly drawn from* $\{1, \ldots, \mathrm{m}_{\mathcal{X}}\}$*. Then a CountSketch matrix* $\mathrm{R}_{\mathcal{X}}$ *is composed with columns* $\mathrm{R}_{\mathcal{X}:j}$*s such that, for all* $1 \leq j \leq n$*,*

$$\mathrm{R}_{\mathcal{X}:j} = r_j I_{:i_j}, \tag{2.78}$$

where the r_j*s are i. i. d. Rademacher variables, i.e. such that* $\mathbb{P}(r_j = 1) = \mathbb{P}(r_j = -1) = 1/2$*, and* $I_{:i_j}$ *is the* i_j*-th column of the* ${\mathrm{m}_{\mathcal{X}}}^2$ *identity matrix* $I_{\mathrm{m}_{\mathcal{X}}}$.

Hence, to cope with the statistical limitations of uniform sub-sampling, one can consider data-independent sketching distributions compatible with the J-L lemma, which constitutes another solution than the data-dependent sub-sampling strategies. However, it comes with higher computational costs. For instance, with Gaussian, it is mandatory to compute the whole Gram matrix K_{X}, causing $\mathcal{O}(n^2)$ space complexity, and computing $\mathrm{R}_{\mathcal{X}} \cdot \mathrm{K}_{\mathrm{X}}$ causes $\mathcal{O}(n^2\, \mathrm{m}_{\mathcal{X}})$ time complexity. Concerning CountSketch, its advantage is that for any matrix A, $\mathrm{R}_{\mathcal{X}} \cdot A$'s time complexity is $\mathcal{O}(\mathrm{nnz}(A))$, $\mathrm{nnz}(A)$ denoting the number of non-zero elements. It is then of particular interest to use CountSketch on sparse matrices, but it is not the case for usually dense Gram matrices, both time and space complexities then remain linear in n^2. Chen and Yang (2021a) introduce Accumulation sketching, a distribution adapted to kernel methods. It is the sum of m sub-sampling sketchings where each row is multiplied with i. i. d. Rademacher variables and the idea is that when $m \to \infty$, Accumulation sketching tends to sub-Gaussian sketching, there exists then an interesting tradeoff between computational and statistical performance controlled by m.

Definition 2.33 (Accumulation sketching). *Let $m \in \mathbb{N}^*$ and $(p_i)_{i=1}^n \in [0,1]^n$ such that $\sum_{i=1}^n p_i = 1$. An Accumulation sketching matrix $\mathrm{R}_\mathcal{X}$ is such that*

$$\mathrm{R}_\mathcal{X} = \sum_{i=1}^{m} \mathrm{R}_\mathcal{X}{}^i, \tag{2.79}$$

where the $\mathrm{R}_\mathcal{X}{}^i$s are composed with rows $\mathrm{R}_\mathcal{X}{}^i_{j:}$s such that, for all $1 \leq j \leq \mathrm{m}_\mathcal{X}$,

$$\mathrm{R}_\mathcal{X}{}^i_{j:} = \frac{r_j^i}{\sqrt{m\,\mathrm{m}_\mathcal{X}\, p_{l_j^i}}} I_{l_j:}, \tag{2.80}$$

where the r_j^is are i. i. d. Rademacher variables, $\{l_1^i, \ldots, l_{\mathrm{m}_\mathcal{X}}^i\}$ are $\mathrm{m}_\mathcal{X}$ indices drawn from $\{1, \ldots, n\}$ according to the probabilities p_is with replacement and $I_{l_j:}$ is the l_j-th row of the n^2 identity matrix I_n.

We also introduce p-sparsified sketches in chapter 3, a sketching distribution adapted to kernels and aiming at the best possible tradeoff between computational and statistical performance as well, and compare it with Accumulation sketching.

Finally, it is worth noticing the contribution of Kpotufe and Sriperumbudur (2020) which aims at building an embedding $\tilde{\Phi} : \mathcal{H}_\mathcal{X} \to \mathbb{R}^{\mathrm{m}_\mathcal{X}}$ based on Gaussian sketching and Monte Carlo sampling that satisfies a property close to the Jonhson-Lindenstrauss lemma up to a bias caused by the covariance operator. More precisely, by noting $\mathrm{C}_\mathcal{X} = \mathbb{E}_{x\sim\rho_\mathcal{X}}[\psi_\mathcal{X}(x) \otimes \psi_\mathcal{X}(x)]$ the covariance operator, the authors show with a high probability that, for any $f, g \in \mathcal{H}_\mathcal{X}$,

$$\frac{|\tilde{\Phi}(f)^\top \tilde{\Phi}(g) - \langle g, \mathrm{C}_\mathcal{X}{}^3 f\rangle_{\mathcal{H}_\mathcal{X}}|}{\|f\|_{\mathcal{H}_\mathcal{X}} \|g\|_{\mathcal{H}_\mathcal{X}}} \leq \varepsilon, \tag{2.81}$$

for some ε non-increasing with respect to n and $\mathrm{m}_\mathcal{X}$.

To sum up, in order to alleviate the computational burden of kernel methods, two main methods exist, namely RFF and sketching, with Nyström its particular instance. Concerning theory, many works prove the excess risk bounds of the estimators induced by such techniques, such as Rudi and Rosasco (2017); Li et al. (2021) for RFF, Yang et al. (2012); Rudi et al. (2015) for Nyström and Yang et al. (2017); Chen and Yang (2021a); Lacotte and Pilanci (2022) for sketching. We give more details in the next section

In this thesis, we focus on sketching to first scale scalar-valued and matrix-valued kernel machines in chapter 3, mainly considering the hypothesis space interpretation. We also provide the ERM problem to solve for scalar regression based on sketched random features. Concerning surrogate kernel methods, we focus on the orthogonal projection perspective for both input and output kernels to design the SISOKR algorithm in chapter 4. Finally, applied to the output kernel, this angle also allows us to consider kernel-induced losses with neural networks in the DSOKR model, see chapter 5.

2.4 Theoretical Guarantees of Kernel Methods

In this section, we present two main sketches of proof to obtain statistical learning guarantees for kernel methods. We first define the meaning of statistical learning guarantees in this thesis.

In the supervised learning settings, we have training pairs $(x_i, y_i) \in (\mathcal{X} \times \mathcal{Y})^n$ assumed to be i. i. d. realisations of an unknown joint distribution ρ. Our goal is then to estimate the mapping $f : \mathcal{X} \to \mathcal{Y}$ such that for any $(x, y) \sim \rho$, $f(x) = y$. To do so, we first equip ourselves with a discrepancy measure over the output objects $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ known to be relevant for the task at hand, and aimed at estimating the mapping f^* minimizing the associated expected risk, i.e.

$$f^* = \operatorname*{arg\,inf}_{f:\mathcal{X}\to\mathcal{Y}} \mathbb{E}[\ell_f] = \operatorname*{arg\,inf}_{f:\mathcal{X}\to\mathcal{Y}} \mathbb{E}[\ell(f(x), y)]. \tag{2.82}$$

As stated in the previous section, two main problems then arise: (i) solving such a problem within the whole space of functions $\mathcal{Y}^{\mathcal{X}}$ is intractable and (ii) it is impossible to compute the expected risk since ρ is unknown. Then, we consider a hypothesis space $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ which is the RKHS or vv-RKHS of an input kernel $\mathrm{k}_{\mathcal{X}}$ or $\mathcal{K}$ respectively in this section, and we rather minimize the empirical risk, i.e.

$$\hat{f} = \operatorname*{arg\,inf}_{f\in\mathcal{H}} \mathbb{E}_n[\ell_f] = \operatorname*{arg\,inf}_{f\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i), y_i), \tag{2.83}$$

and in the case of kernel methods, we usually add a regularisation penalty to avoid overfitting the training data.

We are now ready to introduce the concept of *excess risk*. Solving the ERM problem ensures having an estimator $\hat{f}$ inducing small errors on the training set, however, the initial goal is to build an estimator able to predict the correct output associated with any input drawn from its marginal distribution $\rho_{\mathcal{X}}$. The excess risk $\mathbb{E}[\ell_{\hat{f}}] - \mathbb{E}[\ell_{f^*}]$ of $\hat{f}$ characterizes such a property. In particular, controlling such a term indicates if $\hat{f}$ converges to f^*, and at each rate in terms of the number of training data n. The objective is then to choose the hypothesis space $\mathcal{H}$ such that we are able to derive an excess risk bound, i.e. for $\delta \in (0, 1)$, with probability at least $1 - \delta$

$$\mathbb{E}[\ell_{\hat{f}}] - \mathbb{E}[\ell_{f^*}] \leq \mathcal{S}(n, \delta), \tag{2.84}$$

where $\mathcal{S}(n, \delta) : \mathbb{R}^2 \to \mathbb{R}$ is a non-increasing function with respect to δ and a non-increasing function with respect to n such that $\mathcal{S}(n) \underset{n\to\infty}{\to} 0$. The faster $\mathcal{S}$ tends to 0, the better $\hat{f}$.

Remark 2.34. *In the sketching literature, some works rather focus on the error between the approximated estimator $\tilde{f}$ obtained via sketching and the standard one $\hat{f}$ based on the RKHS norm, i.e. $\|\tilde{f} - \hat{f}\|_{\mathcal{H}}$, see e.g. Lacotte et al. (2019); Lacotte and Pilanci (2022) for $\mathcal{Y} = \mathbb{R}$. In this thesis, we choose to focus on the excess risk induced by the expected risk.*

Nevertheless, computing such a bound against f^* without further knowledge is impossible. We then state the *attainability* assumption, a very standard assumption in the kernel literature (Caponnetto and De Vito, 2007; Rudi et al., 2015; Rudi and Rosasco, 2017; Li et al., 2021), assessing that there exists a minimizer of the expected risk over the hypothesis space $\mathcal{H}$.

Assumption 2.35 (Attainability). *There exists an $f_{\mathcal{H}} \in \mathcal{H}$ such that*

$$\mathbb{E}[\ell_{f_{\mathcal{H}}}] = \min_{f \in \mathcal{H}} \mathbb{E}[\ell_f]. \tag{2.85}$$

It is more now realistic to derive a bound on the excess risk against $f_{\mathcal{H}}$ rather than f^*. Actually, by looking at the excess risk against f^*, we have the following decomposition

$$\mathbb{E}[\ell_{\hat{f}}] - \mathbb{E}[\ell_{f^*}] = \mathbb{E}[\ell_{\hat{f}}] - \mathbb{E}[\ell_{f_{\mathcal{H}}}] + \mathbb{E}[\ell_{f_{\mathcal{H}}}] - \mathbb{E}[\ell_{f^*}], \tag{2.86}$$

which implies that, if we have a bound on $\mathbb{E}[\ell_{f_{\mathcal{H}}}] - \mathbb{E}[\ell_{f^*}]$, controlling $\mathbb{E}[\ell_{\hat{f}}] - \mathbb{E}[\ell_{f_{\mathcal{H}}}]$ allows to control $\mathbb{E}[\ell_{\hat{f}}] - \mathbb{E}[\ell_{f^*}]$. The bias term $\mathbb{E}[\ell_{f_{\mathcal{H}}}] - \mathbb{E}[\ell_{f^*}]$ comes from the hypothesis space $\mathcal{H}$ considered and the fact that it does not necessarily include the true underlying target f^*. This is why it is crucial to use the maximum of prior knowledge about the problem to choose the best possible $\mathcal{H}$. In particular, even if $f^* \notin \mathcal{H}$, it is impossible to control the bias term without any information on it. One of the reasons why kernel methods are a popular family of machine learning algorithms is because they define hypothesis spaces with strong theoretical properties for a wide variety of input data. Furthermore, for a given supervised learning task, if an expert finds features of the input data that are discriminative in terms of their associated output, it is possible to design a p. d. kernel from a similarity measure based on such features and then obtain a hypothesis space that would contain the target f^*. See Brouard et al. (2016a) for the example of metabolite identification, where various input kernels defining various similarity measures on tandem mass spectra were used.

We now present two main ways to derive excess risk bounds for kernel methods that we use in this thesis. The first one, used in 3 and based on Rademacher complexities, can derive excess risk bounds for scalar-valued and matrix-valued kernel machines. The second one, specific to KRR and benefitting from the closed-form solution of $\hat{f}$ and its operator expression, can derive excess risk bounds to operator-valued kernels and is leveraged in chapter 4.

2.4.1 Rademacher Complexity

In this section, we focus on $\mathcal{Y} = \mathbb{R}^d$ for $d \in \mathbb{N}^*$ and the excess risk against $f_{\mathcal{H}}$. We first introduce what we call the generalisation/approximation decomposition of the excess risk

$$\mathbb{E}[\ell_{\hat{f}}] - \mathbb{E}[\ell_{f_{\mathcal{H}}}] = \underbrace{\mathbb{E}[\ell_{\hat{f}}] - \mathbb{E}_n[\ell_{\hat{f}}]}_{\text{generalisation error}} + \underbrace{\mathbb{E}_n[\ell_{\hat{f}}] - \mathbb{E}_n[\ell_{f_{\mathcal{H}}}]}_{\text{approximation error}} + \underbrace{\mathbb{E}_n[\ell_{f_{\mathcal{H}}}] - \mathbb{E}[\ell_{f_{\mathcal{H}}}]}_{\text{generalisation error}}. \tag{2.87}$$

By noting that $\hat{f} = \arg\min_{f \in \mathcal{H}} \mathbb{E}_n[\ell_f]$, we have that $\mathbb{E}_n[\ell_{\hat{f}}] - \mathbb{E}_n[\ell_{f_{\mathcal{H}}}] \leq 0$, and then

$$\mathbb{E}[\ell_{\hat{f}}] - \mathbb{E}[\ell_{f_{\mathcal{H}}}] \leq 2 \sup_{f \in \mathcal{H}} \left| \mathbb{E}[\ell_f] - \mathbb{E}_n[\ell_f] \right|. \tag{2.88}$$

A powerful tool to control generalisation errors and very well adapted to kernels is the *Rademacher complexities*, introduced by Bartlett and Mendelson (2003). For a class of functions $\mathcal{H} \subseteq \mathbb{R}^{\mathcal{X}}$, the empirical Rademacher complexity is defined as

$$\hat{R}_n(\mathcal{H}) = \mathbb{E}_\epsilon \left[\sup_{f \in \mathcal{H}} \left| \frac{2}{n} \sum_{i=1}^{n} \epsilon_i f(x_i) \right| \mid x_1, \ldots, x_n \right], \tag{2.89}$$

where $\epsilon = (\epsilon_i)_{i=1}^n$ are independent Rademacher random variables such that $\mathbb{P}\left\{\epsilon_i = 1\right\} = \mathbb{P}\left\{\epsilon_i = -1\right\} = 1/2$. The generalisation to the finite-dimensional vector-valued case, i.e. $\mathcal{Y} \subseteq \mathbb{R}^d$ for $d \in \mathbb{N}^*$, is given by

$$\mathcal{E}_n(\mathcal{H}) = \mathbb{E}_\epsilon \left[\sup_{f \in \mathcal{H}} \left| \frac{2}{n} \sum_{i=1}^n \sum_{j=1}^d \epsilon_{ij} f(x_i)_j \right| \mid x_1, \dots, x_n \right] \tag{2.90}$$

$$= \mathbb{E}_{(\boldsymbol{\epsilon}_i)_{i=1}^n} \left[\sup_{f \in \mathcal{H}} \left| \frac{2}{n} \sum_{i=1}^n \left\langle \boldsymbol{\epsilon}_i, f(x_i) \right\rangle_{\mathbb{R}^d} \right| \mid x_1, \dots, x_n \right], \tag{2.91}$$

where $\epsilon = (\epsilon_{11}, \dots, \epsilon_{np})$ are nd independent Rademacher variables, and for all $1 \leq i \leq n$, $\boldsymbol{\epsilon}_i = \left(\epsilon_{i1}, \dots, \epsilon_{id}\right)^\top$. The corresponding Rademacher complexity is then defined as the expectation of the empirical Rademacher complexity

$$R_n(\mathcal{H}) = \mathbb{E}_{(x_i)_{i=1}^n} \left[\hat{R}_n(\mathcal{H}) \right]. \tag{2.92}$$

We now state the theorem that uses the Rademacher complexity of $\mathcal{H}$ to derive a bound of the generalisation error.

Theorem 2.36. *(Bartlett and Mendelson, 2003, Theorem 8) Let $\left\{x_i, y_i\right\}_{i=1}^n$ be i.i.d samples from ρ and let $\mathcal{H}$ be the space of functions mapping from $\mathcal{X}$ to $\mathbb{R}$. Denote a loss function with $\ell : \mathcal{Y} \times \mathcal{Y} \to [0,1]$ and recall the learning risk function for all $f \in \mathcal{H}$ is $\mathbb{E}\left[\ell_f\right]$, together with the corresponding empirical risk function $\mathbb{E}_n\left[\ell_f\right] = (1/n)\sum_{i=1}^n \ell\left(y_i, f\left(x_i\right)\right)$. Then, for a sample of size n, for all $f \in \mathcal{H}$ and $\delta \in (0,1)$, with probability $1 - \delta/2$, we have that*

$$\mathbb{E}\left[\ell_f\right] \leq \mathbb{E}_n\left[\ell_f\right] + R_n(\ell \circ \mathcal{H}) + \sqrt{\frac{8\log(4/\delta)}{n}} \tag{2.93}$$

where $\ell \circ \mathcal{H} = \{(x,y) \to l(y, f(x)) - \ell(y,0) \mid f \in \mathcal{H}\}$.

The proof mainly relies on McDiarmid's inequality (McDiarmid et al., 1989). Thanks to the above theorem, if we have a bound of the Rademacher complexity $R_n(\ell \circ \mathcal{H})$, we are then able to derive a bound of the generalisation error.

First, a very classical assumption on the loss function is the Lipschitz-continuity.

Assumption 2.37 (Lipschitz loss)**.** *For all $y \in \mathcal{Y}$, $z \mapsto \ell(z,y)$ is L-Lipschitz, for $L > 0$.*

This is a standard assumption satisfied by many loss functions such as the maximum-margin, robust or pinball losses considered in section 2.5. Moreover, by assuming bounded outputs, the square loss also satisfies the above assumption. Then, using Corollary 1 from Maurer (2016), we have that:

$$R_n(\ell \circ \mathcal{H}) \leq \sqrt{2} L \mathcal{E}_n(\mathcal{H}). \tag{2.94}$$

The goal is now to derive a bound of $\mathcal{E}_n(\mathcal{H})$. Following Maurer (2016), there are two main steps to do so. First, we use the reproducing property of the RKHS $\mathcal{H}$ and the Cauchy-Schwarz inequality to obtain

$$\sup_{f\in\mathcal{H}}\left|\sum_{i=1}^{n}\left\langle \boldsymbol{\epsilon}_i, f(x_i)\right\rangle_{\mathbb{R}^d}\right| \leq \sup_{f\in\mathcal{H}}\left\|f\right\|_{\mathcal{H}}\left\|\sum_{i=1}^{n}\mathcal{K}_{x_i}\boldsymbol{\epsilon}_i\right\|_{\mathcal{H}}. \tag{2.95}$$

However, considering the whole space $\mathcal{H}$, $\sup_{f\in\mathcal{H}}\left\|f\right\|_{\mathcal{H}} = \infty$. Then, we restrict the hypothesis space to the unit ball of $\mathcal{H}$, i.e. $\mathcal{B}(\mathcal{H}) \coloneqq \{f \in \mathcal{H} : \|f\|_{\mathcal{H}} \leq 1\}$.

Assumption 2.38 (Unit ball)**.** *The hypothesis set considered is* $\mathcal{B}\left(\mathcal{H}_k\right)$.

As stated in Rudi and Rosasco (2017, Remark 2), Assumption 2.35 implies that $f_{\mathcal{H}}$ has a bounded norm. Moreover, we can further assume as in Li et al. (2021) that any estimator obtained via ERM has a bounded norm, especially since a regularisation penalty controlling the norm is typically considered in kernel methods, and the regularisation parameter $\lambda > 0$ is usually validated on the training data. Hence, by considering $R > 0$ such that all these estimators lie in the ball of radius R, we then obtain that $\sup_{\|f\|_{\mathcal{H}}\leq R}\left\|f\right\|_{\mathcal{H}} = R$, and this result holds uniformly for any radius R. Hence, without loss of generality and up to a normalisation of $\mathcal{H}$, we can restrict the hypothesis space to the unit ball. Finally, we use the Jensen inequality and this final assumption on the input kernel $\mathcal{K}$ to conclude the proof.

Assumption 2.39 (Trace-class kernel)**.** *There exists* $\kappa > 0$ *such that, for all* $x \in \mathcal{X}$,

$$\operatorname{Tr}(\mathcal{K}(x,x)) \leq \kappa. \tag{2.96}$$

This is a very standard assumption in the kernel literature. For scalar-valued kernels, it simply means that for all $x \in \mathcal{X}$, $\mathrm{k}_{\mathcal{X}}(x,x)$ is bounded, which is the case for the radial kernels such as the Gaussian one for instance. In the case of OVK, Caponnetto and De Vito (2007) considers this assumption.

Remark 2.40 (Decomposable kernels)**.** *For decomposable kernels* $\mathcal{K} = \mathrm{k}_{\mathcal{X}} M$*, one can simply assume that there exists* $\kappa_{\mathcal{X}} > 0$ *such that for all* $x \in \mathcal{X}$, $\mathrm{k}_{\mathcal{X}}(x,x) \leq \kappa_{\mathcal{X}}$*. In fact,* $\operatorname{Tr}(\mathcal{K}(x,x)) = \mathrm{k}_{\mathcal{X}}(x,x)\operatorname{Tr}(M) \leq \kappa_{\mathcal{X}}\operatorname{Tr}(M)$ *in this case, and for* $M = I_d$, $\operatorname{Tr}(\mathcal{K}(x,x)) \leq d\,\kappa_{\mathcal{X}}$.

We can now give the bound of the Rademacher complexity of $\mathcal{H}$'s unit ball,

$$R_n(\mathcal{B}(\mathcal{H})) = 2\sqrt{\frac{\kappa}{n}}, \tag{2.97}$$

and then for $\delta \in (0,1)$, with probability $1-\delta/2$,

$$\sup_{f\in\mathcal{B}(\mathcal{H})}\left|E[\ell_f] - \mathbb{E}_n[\ell_f]\right| \lesssim \sqrt{\log(4/\delta)/n}. \tag{2.98}$$

However, note that in kernel methods, $\hat{f}$ is usually the minimizer of the penalized ERm problem for a regularisation parameter $\lambda > 0$, then $\hat{f} \neq \arg\min_{f\in\mathcal{H}} \mathbb{E}_n[\ell_f]$. To cope with this, Yang et al. (2012) includes the regularisation term in its excess risk and then studies $\mathbb{E}[\ell_{\hat{f}}] + \frac{\lambda}{2}\|\hat{f}\|_{\mathcal{H}}^2 - \mathbb{E}[\ell_{f_{\mathcal{H}}}] - \frac{\lambda}{2}\|f_{\mathcal{H}}\|_{\mathcal{H}}^2$. Other works aim at controlling the

approximation error term as well. However, while doing so for the square loss appears feasible thanks to the closed-form solution, it is challenging for a generic loss ℓ. Li et al. (2021) first focuses on the RFF KRR case, and then uses this result to generalize it to any Lipschitz loss thanks to the Jensen inequality, but obtaining then a degraded rate compared with the KRR with a square root function applied to the KRR learning rate. We adopt the same strategy for the sketched scalar-valued and matrix-valued kernel machines, using the KRR approximation error's bound obtained by Yang et al. (2017), and generalizing it to the decomposable matrix-valued case.

Finally, note that Bartlett et al. (2005) introduce *local Rademacher complexities* that focus on the subspace of the low-variance estimators f, i.e. such that

$$\mathbb{E}_x\Big[f(x) - f_{\mathcal{H}}(x)\Big]^2 \leq B\,\mathbb{E}\Big[\ell_f - \ell_{f_{\mathcal{H}}}\Big], \tag{2.99}$$

for $B > 0$. The authors compute their Rademacher complexity called the local Rademacher complexity for any RKHS thanks to a sub-root function. This allows to derive refined faster learning rates compared with the previous ones, and it is worth mentioning that Li et al. (2021) uses such results to derive refined rates for RFF as well.

2.4.2 Statistical analysis of Kernel Ridge Regression

In this section, we consider the general case where $\mathcal{Y}$ is a possibly infinite-dimensional Hilbert space and focus on the KRR, i.e. $\ell : (y, y') \mapsto \|y - y'\|_{\mathcal{Y}}^2$, with the input identity decomposable kernel $\mathcal{K} = \mathrm{k}_{\mathcal{X}} I_{\mathcal{Y}}$ as in chapter 4 (where the output space is the RKHS of an output kernel). We will summarize results from Caponnetto and De Vito (2007), Ciliberto et al. (2016) and Ciliberto et al. (2020) to obtain an excess risk bound for $\hat{f}$. Note that Caponnetto and De Vito (2007) assumes the input kernel to be trace-class, which is rather restrictive in the case where $\dim(\mathcal{Y}) = \infty$ since the identity decomposable kernel is consequently not trace-class. Ciliberto et al. (2016) extended the results from Caponnetto and De Vito (2007) to this case.

We remind that the KRR estimator for $\mathcal{K} = \mathrm{k}_{\mathcal{X}} I_{\mathcal{Y}}$ is $\hat{f}(\cdot) = \sum_{i=1} \hat{\alpha}_i(\cdot) y_i$ where

$$\hat{\alpha} : x \in \mathcal{X} \mapsto (\mathrm{K}_{\mathrm{X}} + n\lambda I_n)^{-1}\, \mathrm{k}_{\mathrm{X}}^{x}, \tag{2.100}$$

and $\mathrm{k}_{\mathrm{X}}^{x} = (\mathrm{k}_{\mathcal{X}}(x, x_i))_{i=1}^n \in \mathbb{R}^n$. Hence, we aim at deriving an upper bound for its excess risk

$$\mathbb{E}[\ell_{\hat{f}}] - \mathbb{E}[\ell_{f^*}] = \mathbb{E}_{(x,y)\sim\rho}[\|\hat{f}(x) - y\|_{\mathcal{Y}}^2] - \mathbb{E}_{(x,y)\sim\rho}[\|f^*(x) - y\|_{\mathcal{Y}}^2] = \mathcal{E}(\hat{f}) - \mathcal{E}(f^*), \tag{2.101}$$

where $f^* = \arg\min_{f:\mathcal{X}\to\mathcal{Y}} \mathcal{E}(f)$. One can show that f^* is given by

$$f^* : x \mapsto \mathbb{E}_y[y \mid x], \tag{2.102}$$

and that

$$\mathcal{E}(\hat{f}) - \mathcal{E}(f^*) = \mathbb{E}_x[\|\hat{f}(x) - f^*(x)\|_{\mathcal{Y}}^2], \tag{2.103}$$

see e.g. Ciliberto et al. (2020, Lemma A.2) for a proof.

Operator expression. We first derive the operator expression of $\hat{f}$, i.e. the expression of $\widehat{F} : \mathcal{H}_{\mathcal{X}} \to \mathcal{Y}$ such that $\|\widehat{F}\|_{\mathrm{HS}} < \infty$ and

$$\hat{f}(\cdot) = \widehat{F}\,\psi_{\mathcal{X}}(\cdot). \tag{2.104}$$

Besides, we define the input/output sampling operators (Smale and Zhou, 2007),

$$\mathrm{S_X} : h \in \mathcal{H}_\mathcal{X} \mapsto \frac{1}{\sqrt{n}}(\langle h, \psi_\mathcal{X}(x_i)\rangle_{\mathcal{H}_\mathcal{X}})_{i=1}^n \in \mathbb{R}^n, \quad \mathrm{S_Y} : y \in \mathcal{Y} \mapsto \frac{1}{\sqrt{n}}(\langle y, y_i\rangle_\mathcal{Y})_{i=1}^n \in \mathbb{R}^n, \tag{2.105}$$

as well as the empirical covariances

$$\widehat{V} = \frac{1}{n}\sum_{i=1}^n y_i \otimes \psi_\mathcal{X}(x_i), \quad \widehat{\mathrm{C}}_\mathrm{X} = \frac{1}{n}\sum_{i=1}^n \psi_\mathcal{X}(x_i) \otimes \psi_\mathcal{X}(x_i), \tag{2.106}$$

and finally note that

$$\widehat{V} = \mathrm{S_Y}^\# \mathrm{S_X}, \quad \widehat{\mathrm{C}}_\mathrm{X} = \mathrm{S_X}^\# \mathrm{S_X}, \quad \mathrm{K_X} = n\,\mathrm{S_X}\,\mathrm{S_X}^\#. \tag{2.107}$$

Hence, going from the above expression of $\hat{f}$, for all $x \in \mathcal{X}$,

$$\begin{aligned}
\hat{f}(x) &= \sqrt{n}\,\mathrm{S_Y}^\#\,\hat{\alpha}(x) && (2.108)\\
&= \sqrt{n}\,\mathrm{S_Y}^\#(n\,\mathrm{S_X}\,\mathrm{S_X}^\# + n\lambda I_n)^{-1}\sqrt{n}\,\mathrm{S_X}\,\psi_\mathcal{X}(x) && (2.109)\\
&= \underbrace{\mathrm{S_Y}^\#\,\mathrm{S_X}}_{=\widehat{V}}\underbrace{(\mathrm{S_X}^\#\,\mathrm{S_X} + \lambda I_{\mathcal{H}_\mathcal{X}})^{-1}}_{=\widehat{\mathrm{C}}_\mathrm{X}}\,\psi_\mathcal{X}(x), && (2.110)
\end{aligned}$$

where the last equality comes from a standard Woodbury formula. We are now ready for the analysis of the excess risk.

Attainaility assumption. It is possible to either consider the weaker above assumption stating that there exists $f_\mathcal{H} \in \mathcal{H}$ that minimizes the expected risk and use it as a target, which would leave the analysis of the induced bias term $\mathcal{E}(f_\mathcal{H}) - \mathcal{E}(f^*)$ out, or the stronger assumption that $f^* \in \mathcal{H}$. We choose the latter to be consistent with chapter 4.

Assumption 2.41 (Attainability). *$f^* \in \mathcal{H}$, then there exists $F : \mathcal{H}_\mathcal{X} \to \mathcal{Y}$ such that $\{F\|_\mathrm{HS} < \infty$ and*

$$f^*(\cdot) = F\,\psi_\mathcal{X}(\cdot). \tag{2.111}$$

Then, by Ciliberto et al. (2020, Lemma B.9), and with

$$V = \mathbb{E}_{x,y}[y \otimes \psi_\mathcal{X}(x)], \quad \mathrm{C}_\mathcal{X} = \mathbb{E}_x[\psi_\mathcal{X}(x) \otimes \psi_\mathcal{X}(x)], \tag{2.112}$$

we have that

$$F = V\,\mathrm{C}_\mathcal{X}{}^\dagger, \tag{2.113}$$

where $\mathrm{C}_\mathcal{X}{}^\dagger$ denotes the Moore-Penrose inverse of $\mathrm{C}_\mathcal{X}{}^\dagger$. As a consequence, the excess risk rewrites as the Hilbert-Schmidt norm of the difference between $\widehat{F}$ and F against the covariance $\mathrm{C}_\mathcal{X}$,

$$\mathbb{E}_x[\|\hat{f}(x) - f^*(x)\|_\mathcal{Y}^2] = \|(\widehat{F} - F)\,\mathrm{C}_\mathcal{X}{}^{1/2}\|_\mathrm{HS}^2. \tag{2.114}$$

The analysis of the excess risk now boils down to liner algebra and concentration inequalities.

Bias-variance decomposition. Let f_λ^* be the following regularized optimal estimator,

$$f_\lambda^*(\cdot) = F_\lambda \psi_{\mathcal{X}}(\cdot) \quad \text{with} \quad F_\lambda = F\, \mathrm{C}_{\mathcal{X}}(\mathrm{C}_{\mathcal{X}} + \lambda I_{Hx})^{-1}, \tag{2.115}$$

the excess risk then admits the following bias-variance decomposition

$$\mathcal{E}(\hat{f}) - \mathcal{E}(f^*) \leq \mathbb{E}_x[\|\hat{f}(x) - f_\lambda^*(x)\|_{\mathcal{Y}}^2] + \mathbb{E}_x[\|f_\lambda^*(x) - f^*(x)\|_{\mathcal{Y}}^2] \tag{2.116}$$

$$= \underbrace{\|(\widehat{F} - F_\lambda)\mathrm{C}_{\mathcal{X}}{}^{1/2}\|_{\mathrm{HS}}^2}_{\text{variance}} + \underbrace{\|(F_\lambda - F)\mathrm{C}_{\mathcal{X}}{}^{1/2}\|_{\mathrm{HS}}^2}_{\text{bias}}, \tag{2.117}$$

where the last equality is obtained via similar derivations than in eq. (2.114).

Variance bound. From the proof of Ciliberto et al. (2016, Lemma 18), one can prove that

$$\|(\widehat{F} - F_\lambda)\mathrm{C}_{\mathcal{X}}{}^{1/2}\|_{\mathrm{HS}}^2 \lesssim \|(\mathrm{C}_{\mathcal{X}} + \lambda I_{\mathcal{H}_{\mathcal{X}}})^{-1/2}\mathrm{C}_{\mathcal{X}}{}^{1/2}\|_{\mathrm{HS}}^2 n^{-1}. \tag{2.118}$$

Controlling $\|(\mathrm{C}_{\mathcal{X}} + \lambda I_{\mathcal{H}_{\mathcal{X}}})^{-1/2}\mathrm{C}_{\mathcal{X}}{}^{1/2}\|_{\mathrm{HS}}^2$ then gives rise of the *capacity condition*, a classical assumption in KRR literature (Caponnetto and De Vito, 2007; Ciliberto et al., 2020).

Assumption 2.42 (Capacity condition)**.** *For all $\lambda > 0$, there exists $\gamma \in [0,1]$ such that*

$$\|(\mathrm{C}_{\mathcal{X}} + \lambda I_{\mathcal{H}_{\mathcal{X}}})^{-1/2}\mathrm{C}_{\mathcal{X}}{}^{1/2}\|_{\mathrm{HS}}^2 \lesssim \lambda^{-\gamma}. \tag{2.119}$$

First, as in Ciliberto et al. (2020), note that if there exists $\kappa_{\mathcal{X}} > 0$ such that for all $x \in \mathcal{X}$, $\mathrm{k}_{\mathcal{X}}(x,x) \leq \kappa_{\mathcal{X}}$, then the above assumption is true for $\gamma = 1$. Moreover, note that this assumption is actually an assumption over the eigendecay of $\mathrm{C}_{\mathcal{X}}$, the faster it is, the lower γ. As a limiting case, if $\mathrm{C}_{\mathcal{X}}$ is finite-rank, then $\gamma = 0$. If for all $j \in \mathbb{N}^*$, $\sigma_j(\mathrm{C}_{\mathcal{X}}) \lesssim j^{-\beta}$ for $\beta > 1$, then $\gamma = 1/\beta$. Finally, the eigendecay is characterized by the kernel $\mathrm{k}_{\mathcal{X}}$ and the marginal distribution $\rho_{\mathcal{X}}$, see Ciliberto et al. (2020) for further details and an example of kernel and marginal distribution satisfying the capacity condition.

Remark 2.43 (Effective dimension)**.** *In some works, as in Ciliberto et al. (2020) for instance, the above quantity $\|(\mathrm{C}_{\mathcal{X}} + \lambda I_{\mathcal{H}_{\mathcal{X}}})^{-1/2}\mathrm{C}_{\mathcal{X}}{}^{1/2}\|_{\mathrm{HS}}^2$ is called the* effective dimension,

$$\|(\mathrm{C}_{\mathcal{X}} + \lambda I_{\mathcal{H}_{\mathcal{X}}})^{-1/2}\mathrm{C}_{\mathcal{X}}{}^{1/2}\|_{\mathrm{HS}}^2 = \mathrm{Tr}(\mathrm{C}_{\mathcal{X}}(\mathrm{C}_{\mathcal{X}} + \lambda I_{\mathcal{H}_{\mathcal{X}}})^{-1}) = \mathrm{d}_{\mathrm{eff}}^{\mathcal{X}}(\lambda). \tag{2.120}$$

Bias bound. Some quick derivations and using the fact that $I_{\mathcal{H}_{\mathcal{X}}} = (\mathrm{C}_{\mathcal{X}} + \lambda I_{\mathcal{H}_{\mathcal{X}}})(\mathrm{C}_{\mathcal{X}} + \lambda I_{\mathcal{H}_{\mathcal{X}}})^{-1}$, we obtain that

$$\|(F_\lambda - F)\mathrm{C}_{\mathcal{X}}{}^{1/2}\|_{\mathrm{HS}}^2 = \|(F\,\mathrm{C}_{\mathcal{X}}(\mathrm{C}_{\mathcal{X}} + \lambda I_{Hx})^{-1} - F)\mathrm{C}_{\mathcal{X}}{}^{1/2}\|_{\mathrm{HS}}^2 \tag{2.121}$$

$$= \lambda^2 \|F(\mathrm{C}_{\mathcal{X}} + \lambda I_{\mathcal{H}_{\mathcal{X}}})^{-1}\mathrm{C}_{\mathcal{X}}{}^{1/2}\|_{\mathrm{HS}}^2 \tag{2.122}$$

Controlling $\|F(\mathrm{C}_{\mathcal{X}} + \lambda I_{\mathcal{H}_{\mathcal{X}}})^{-1}\mathrm{C}_{\mathcal{X}}{}^{1/2}\|_{\mathrm{HS}}^2$ then gives rise of the *source condition*, another standard assumption in KRR literature (Caponnetto and De Vito, 2007; Ciliberto et al., 2020).

Assumption 2.44 (Source condition)**.** *There exists $\mu \in [0,1]$ such that, for all $\lambda > 0$,*

$$\|F(\mathrm{C}_{\mathcal{X}} + \lambda I_{\mathcal{H}_{\mathcal{X}}})^{-1}\mathrm{C}_{\mathcal{X}}{}^{1/2}\|_{\mathrm{HS}}^2 \lesssim \lambda^{-\mu}. \tag{2.123}$$

This assumption is always verified for $\mu = 1$ because $\|F\|_{\mathrm{HS}} < \infty$. The more the right eigenvectors of F are aligned with the eigenvectors of $\mathrm{C}_{\mathcal{X}}$ the smaller μ, and the less increasing λ will cause a significant bias.

Learning rate. Then equipped with the above results, we are able to derive the learning rate of the KKR estimator $\hat{f}$. With $\lambda = n^{-\frac{1}{2(1-\mu+\gamma}}$, corresponding to the best bias-variance trade-off, and for $\delta \in (0,1)$, then with probability $1-\delta$,

$$\mathcal{E}(\hat{f}) - \mathcal{E}(f^*) \lesssim \log(4/\delta) n^{-\frac{1-\mu}{2(1-\mu+\gamma}}. \tag{2.124}$$

The stronger the capacity and the source condition, the faster the learning rate, from $n^{-1/4}$ to $n^{-1/2}$. Concerning the tightness of this bound, Caponnetto and De Vito (2007) provides minimax rates for many settings of KRR estimators.

Rudi and Rosasco (2017) and Rudi et al. (2015) use such techniques of proof for large-scale learning with RFF and Nyström approximation respectively, but in the case of scalar regression, i.e. $\mathcal{Y} = \mathbb{R}$. In particular, in Rudi et al. (2015), the authors show that the effect of the Nyström approximation boils down to controlling the following term

$$\|(\widetilde{\mathrm{P}}_{\mathrm{X}} - I_{\mathcal{H}_\mathcal{X}}) \mathrm{C}_\mathcal{X}^{1/2}\|_{\mathrm{op}}, \tag{2.125}$$

which is the operator norm of the projected covariance operator. They use a Bernstein's inequality for the sum of random operators to derive a bound of this quantity. In chapter 4 we build upon all these works to derive an error decomposition where we recover the classical KRR error and the term in eq. (2.125) for both the input and output kernels, called the *sketching reconstruction error*. Finally, we use a concentration inequality of Koltchinskii and Lounici (2017) for sum of sub-Gaussian random variables in a separable Hilbert space to derive a bound of the sketching reconstruction error and conclude the proof.

2.5 Beyond the Square Loss for Kernel Methods

We here present standard kernel-based models relying on losses different than the square one. We start with the Support Vector Machines (Cortes and Vapnik, 1995) and then present robust losses such as the ϵ-insensitive losses (Steinwart and Christmann, 2008a) and the Huber loss (Huber, 1964). Finally, we present quantile regression that uses the pinball loss (Koenker, 2005), which is one of the problems we consider in chapter 3. For each task, we first focus on the scalar case and then present some works extending them to the vector-valued case.

2.5.1 Support Vector Machines

Originally, SVM are designed to solve binary classification, i.e. $\mathcal{Y} = \{-1, 1\}$. We first consider $\mathcal{X} = \mathbb{R}^d$ for $d \in \mathbb{N}^*$ and we will later explain why kernels are well-suited to SVM. The idea behind SVM is that the data are linearly separable: we can find a hyperplane that separates the -1 and $+1$ data, as illustrated in 2.2. A hyperplane H is characterized by a normal vector $\omega \in \mathbb{R}^d$ and its offset $b \in \mathbb{R}$. Any point $x \in H$ is such that $h(x) = \omega^\top x + b = 0$ and the sign $\mathrm{sgn}(h(x))$ of $h(x)$ determines from each side of the hyperplane x is. The estimator is then $f : x \in \mathbb{R}^d \mapsto \mathrm{sgn}(\omega^\top x + b) \in \{-1, 1\}$. The goal is to find the best possible $(\omega, b) \in \mathbb{R}^{d+1}$ such that

1. for all training pairs (x_i, y_i), x_i is well-classified, i.e. $y_i(\omega^\top x_i + b) \geq 0$;
2. the distance between the hyperplane and its closest point is maximized, i.e. $\max_{(\omega,b)\in\mathbb{R}^{d+1}} \min_{1\leq i\leq n} \frac{y_i(\omega^\top x_i + b)}{\|w\|_2}$ since the distance of a point x to H is given by $|\omega^\top x + b|/\|w\|_2$.

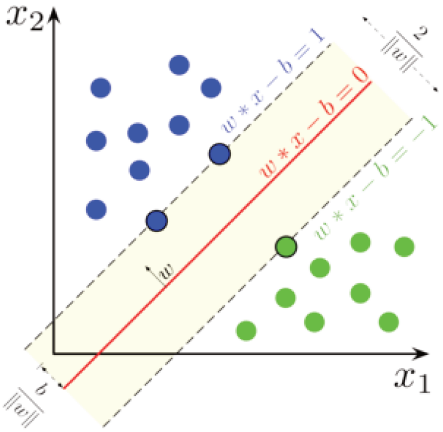


Figure 2.2: Illustration of Support Vector Machines.

Up to a normalisation of ω and b, we can set $\min_{1\leq i\leq n} y_i(\omega^\top x_i + b) = 1$, and then the primal SVM problem is

$$\min_{(\omega,b)\in\mathbb{R}^{d+1}} \frac{1}{2}\|\omega\|_2^2 \quad \text{such that} \quad y_i(\omega^\top x_i + b) \geq 1\,. \tag{2.126}$$

This is called *hard-margin* SVM since we consider the restrictive constraint of all training points being well-separated by the hyperplane, i.e. $y_i(\omega^\top x_i + b) \geq 0$ for all $i \in [\![n]\!]$. It is possible to introduce *slack variables* ξ_i that tolerate wrongly classified points and are controlled by a regularisation parameter $C > 0$, i.e.

$$\begin{aligned}&\min_{(\omega,b)\in\mathbb{R}^{d+1},\xi\in\mathbb{R}^n} \frac{1}{2}\|\omega\|_2^2 + C\sum_{i=1}^n \xi_i \\ &\text{such that} \quad \xi_i \geq 0, y_i(\omega^\top x_i + b) \geq 1 - \xi_i, \forall i \in [\![n]\!].\end{aligned}$$

Setting a high C induces small ξs and then harder margins, whereas a small C allows higher ξs and then softer margins. Finally, by lagrangian duality and KKT conditions, we obtain that the SVM estimator $\hat{f} = \mathrm{sgn}(\hat{\omega}^\top x + \hat{b})$ is obtained with $\hat{\omega} = \sum_{i=1}^n \hat{\alpha}_i y_i x_i$ where $\hat{\alpha}$ is the solution to the dual problem

$$\begin{aligned}&\min_{\alpha\in\mathbb{R}^n} -\frac{1}{2}\sum_{i,j=1}^n \alpha_i\alpha_j y_i y_j x_i^\top x_j + \sum_{i=1}^n \alpha_i \\ &\text{such that} \quad \sum_{i=1}^n \alpha_i y_i = 0, 0 \leq \alpha_i \leq C, \forall i \in [\![n]\!].\end{aligned}$$

We then saw that SVM relies on the assumption that the data are linearly separable, but this is not always the case. Moreover, as we saw in 2.1, p. d. kernels define an embedding onto the linear RKHS, which is called the *kernel trick*, as illustrated with a small example in 2.3. Since the SVM dual problem and the final estimator depend on the inputs only through their pair-wise inner products, employing an input kernel and replacing every x by their embedded counterpart $\psi_{\mathcal{X}}(x)$ is very easy. Moreover, one can then consider any input $\mathcal{X}$ on which it is possible to define a p. d. kernel $\mathrm{k}_{\mathcal{X}}$.

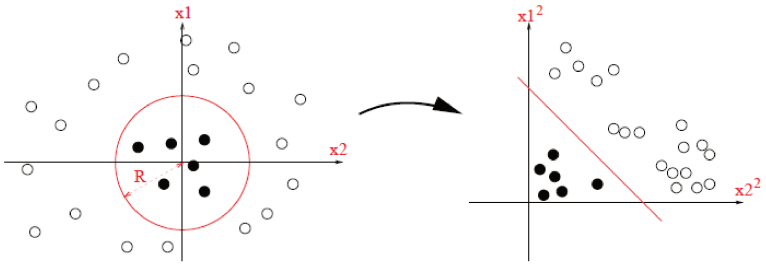


Figure 2.3: Illustration of the kernel trick with $\mathrm{k}_{\mathcal{X}}(x,x') = (x^\top x')^2 = \psi_{\mathcal{X}}(x)^\top \psi_{\mathcal{X}}(x')$ where $\psi_{\mathcal{X}} : (x_1,x_2)^\top \mapsto (x_1^2, \sqrt{2}x_1x_2, x_2^2)^\top$.

Finally, we obtain $\hat{f} = \mathrm{sgn}(\sum_{i=1}^n \hat{\alpha}_i y_i \mathrm{k}_{\mathcal{X}}(x_i,x) + \hat{b})$ where $\hat{\alpha}$ is the solution to

$$\min_{\alpha\in\mathbb{R}^n} -\frac{1}{2}\sum_{i,j=1}^n \alpha_i\alpha_j y_i y_j \mathrm{k}_{\mathcal{X}}(x_i,x_j) + \sum_{i=1}^n \alpha_i$$

$$\text{such that} \quad \sum_{i=1}^n \alpha_i y_i = 0, 0 \le \alpha_i \le C, \forall i \in [\![n]\!].$$

We now introduce the link with the Hinge loss (Boser et al., 1992). Indeed, it can be shown that the above estimator $\hat{f} = \mathrm{sgn}(\hat{h})$ can be obtained from the following ERM primal problem

$$\hat{h} = \operatorname*{arg\,min}_{h\in\mathcal{H}_{\mathcal{X}}} \frac{1}{n}\sum_{i=1}^n \ell(h(x_i),y_i) + \lambda\|h\|^2_{\mathcal{H}_{\mathcal{X}}}, \tag{2.7}$$

where $\ell : (y,y') \mapsto \max(1-yy',0)$ is the Hinge loss. The proof simply consists in invoking the represented theorem to obtain that $\hat{h} = \sum_{i=1}^n \hat{\alpha}_i \mathrm{k}_{\mathcal{X}}(\cdot,x_i)$, introducing the slack variables $\xi_i \ge \max(1-y_i[\mathrm{K}_X\alpha]_i,0)$ and computing its dual problem. Note that $C = 1/2n\lambda$.

SVM is then a very famous and insightful example of the relevance of kernel methods with a different loss function than the square one. Tsochantaridis et al. (2004) even generalizes SVM to the case of the structured outputs, considering instead joint feature maps $\Psi(x,y)$ taking into account input/output and output/output dependencies as well. They derive a dual problem and an algorithm to solve it. Given input/output kernels $\mathrm{k}_{\mathcal{X}}$ and $\mathrm{k}_{\mathcal{Y}}$, considering the product joint feature map $\Psi(x,y) = (\psi_{\mathcal{X}}(x), \psi_{\mathcal{Y}}(y))$ is then a particular case of their framework, as in Szedmak et al. (2006) where $\mathrm{k}_{\mathcal{Y}}$ is the linear kernel. Finally, Brouard et al. (2016b) considers OVK to solve problem (2.7) in the case where $\mathcal{Y}$ is a generic Hilbert space, corresponding to the generalisation of the Hinge loss $\ell : (y,y') \mapsto \max(1-\langle y,y'\rangle_{\mathcal{Y}},0)$. The authors derive a parameterised dual problem to obtain the corresponding estimator.

2.5.2 Robust Losses

Another standard example is the robust losses. Assuming that the training output data are contaminated by outliers having extremely high or low values, the idea behind robust losses is to avoid overfitting to such values and then to lower the prediction error induced by such outputs.

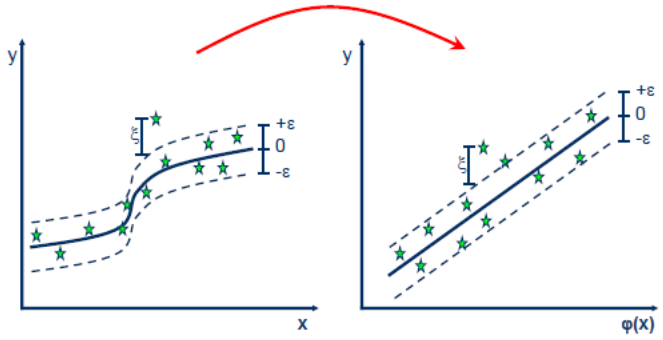


Figure 2.4: Illustration of Support Vector Regression.

Support Vector Regression (Drucker et al., 1997) adapts the SVM to the regression case and is a very good example of a robust regression model. As illustrated in Figure 2.4, the idea is now to find the optimal hyperplane H induced by (ω, b) such that the data points lie in an ε-tube around it, for $\varepsilon > 0$, i.e. $|y_i - \langle \omega, \psi_{\mathcal{X}}(x_i)\rangle_{\mathcal{H}_{\mathcal{X}}} - b| \leq \varepsilon$, yielding the following constraints, for all $i \in [\![n]\!]$,

$$\begin{aligned} 0 \leq y_i - \langle \omega, \psi_{\mathcal{X}}(x_i)\rangle_{\mathcal{H}_{\mathcal{X}}} - b \leq \varepsilon\,, \\ 0 \leq \langle \omega, \psi_{\mathcal{X}}(x_i)\rangle_{\mathcal{H}_{\mathcal{X}}} + b - y_i \leq \varepsilon\,. \end{aligned}$$

Then similarly to SVM, we can introduce the slack variables ξ_i, ξ_i' and the primal problem is then

$$\begin{aligned} \min_{\omega, b, \xi, \xi'} \ & \frac{1}{2}\|\omega\|^2_{\mathcal{H}_{\mathcal{X}}} + C\sum_{i=1}^{n}(\xi_i \xi_i') \\ \text{such that} \quad & y_i - \langle \omega, \psi_{\mathcal{X}}(x_i)\rangle_{\mathcal{H}_{\mathcal{X}}} - b \leq \varepsilon + \xi_i\,, \\ & \langle \omega, \psi_{\mathcal{X}}(x_i)\rangle_{\mathcal{H}_{\mathcal{X}}} + b - y_i \leq \varepsilon + \xi_i'\,, \\ & \xi_i, \xi_i' \geq 0, i \in [\![n]\!]. \end{aligned}$$

Finally, by duality and KKT conditions, the final estimator $\hat{f} = \sum_{i=1}^{n}(\hat{\alpha}_i - \hat{\alpha}_i')\mathrm{k}_{\mathcal{X}}(\cdot, x_i) + \hat{b}$ where $\hat{\alpha}$ and $\hat{\alpha}'$ are solutions to the dual problem

$$\begin{aligned} \min_{\alpha, \alpha'} \ & \frac{1}{2}\sum_{i,j=1}^{n}(\alpha_i - \alpha_i')(\alpha_j - \alpha_j')\mathrm{k}_{\mathcal{X}}(x_i, x_j) + \varepsilon\sum_{i=1}^{n}(\alpha_i + \alpha_i') - \sum_{i=1}^{n} y_i(\alpha_i - \alpha_i') \\ \text{such that} \quad & \sum_{i=1}^{n} \alpha_i - \alpha_i' = 0 \\ & 0 \leq \alpha_i, \alpha_i' \leq C, i \in [\![n]\!]. \end{aligned}$$

As for SVM, this corresponds to solving the ERM problem with the ε-insensitive ℓ_1 loss, i.e. $\ell : (y, y') \mapsto \max(|y - y'| - \varepsilon, 0)$.

The ε-insensitive ℓ_1 loss can be generalized for $\mathcal{Y}$ being a generic Hilbert case and in this case, $\ell : (y, y') \mapsto \max(\|y - y'\|_{\mathcal{Y}} - \varepsilon, 0)$. The same applies for the ε-insensitive ℓ_2 loss, $\ell : (y, y') \mapsto \max(\|y - y'\|_{\mathcal{Y}} - \varepsilon, 0)^2$, as well as the Huber loss (Huber, 1964), defined,

for $\kappa > 0$, by

$$\forall y, y' \in \mathcal{Y}, \quad \ell(y, y') = \begin{cases} \frac{1}{2}\|y - y'\|_{\mathcal{Y}}^2 & \text{if } \|y - y'\|_{\mathcal{Y}} \leq \kappa \\ \kappa\left(\|y - y'\|_{\mathcal{Y}} - \frac{\kappa}{2}\right) & \text{otherwise} \end{cases}. \tag{2.127}$$

Sangnier et al. (2017) considers $\mathcal{Y} = \mathbb{R}^d$ for $d \in \mathbb{N}^*$ and designs a primal-dual coordinate descent algorithm to obtain the estimator induced by solving the ERM problem with such losses. Laforgue et al. (2020) goes further and consider $\mathcal{Y}$ to be a generic Hilbert space. Thanks to some assumptions on the loss function and the input OVK $\mathcal{K}$, in particular

1. $\forall i \leqslant n, \forall \left(\alpha^{\mathbf{Y}}, \alpha^{\perp}\right) \in \text{span}((y_i)_{i=1}^n) \times \text{span}((y_i)_{i=1}^n)^{\perp}$, it holds $\ell_i^{\star}\left(\alpha^{\mathbf{Y}}\right) \leqslant \ell_i^{\star}\left(\alpha^{\mathbf{Y}} + \alpha^{\perp}\right)$, where where $\ell_i^{\star}$ denotes the Fenchel-Legendre transform of $\ell_i : y \in \mathbb{R} \mapsto \ell(y, y_)$ for any $i \leq n$;

2. $\forall i, j \leqslant n$, $\text{span}((y_i)_{i=1}^n)$ is invariant by $\mathcal{K}\left(x_i, x_j\right)$, i.e. if $y \in \text{span}((y_i)_{i=1}^n)$, then $\mathcal{K}(x_i, x_j)y \in \text{span}((y_i)_{i=1}^n)$;

the authors show that the solution $\hat{f}$ to the standard ERM problem over the vv-RKHS of $\mathcal{K}$ writes as $\hat{f} = \sum_{i,j=1}^n \mathcal{K}(\cdot, x_i)\hat{\omega}_{ij}y_j$ where $\hat{\Omega} = (\hat{\omega})_{1 \leq i,j \leq n}$ is the solution to a parameterized dual problem, see Laforgue et al. (2020)[Theorem 4]. In particular, they show the following result for the above-mentioned robust losses.

Theorem 2.45 (Laforgue et al. 2020, Theorem 6)**.** *If $\mathcal{K} = \mathrm{k}_{\mathcal{X}} I_{\mathcal{Y}}$, $\hat{\Omega} = \hat{W} V^{-1}$ where $\hat{W}$ is the solution to the ε-Ridge regression, κ-Huber regression, and ε-SVR dual problems*

$$\min_{W \in \mathbb{R}^{n \times n}} \frac{1}{2}\|AW - B\|_{Fro}^2 + \varepsilon\|W\|_{2,1} \tag{D1}$$

$$\begin{aligned} &\min_{W \in \mathbb{R}^{n \times n}} \frac{1}{2}\|AW - B\|_{Fro}^2 \\ &s.t.\ \|W\|_{2,\infty} \leqslant \kappa \end{aligned} \tag{D2}$$

$$\begin{aligned} &\min_{W \in \mathbb{R}^{n \times n}} \frac{1}{2}\|AW - B\|_{Fro}^2 + \varepsilon\|W\|_{2,1}, \\ &s.t.\ \|W\|_{2,\infty} \leqslant 1 \end{aligned} \tag{D3}$$

with V, A, B such that: $VV^{\top} = (\langle y_i, y_j \rangle_{\mathcal{Y}})_{1 \leq i,j \leq n}$, $A^{\top}A = \frac{\mathrm{K}_{\mathrm{X}}}{\Lambda n} + I_n$ (or $A^{\top}A = \mathrm{K}_{\mathrm{X}}/(\Lambda n)$ for the ε-SVR), and $A^{\top}B = V$.

In chapter 3, we consider the κ-Huber and ε-SVR with $\mathcal{Y} = \mathbb{R}^d$ for some $d \in \mathbb{N}^*$ and show how to leverage sketching on matrix-valued kernels to reduce time and space complexities. However, we choose to solve the primal problem rather than the dual one as in Sangnier et al. (2017) or Laforgue et al. (2020) as it is more adapted to sketching, as discussed in section 3.2.3.

2.5.3 Pinball Loss

Conditional quantile regression (Koenker and Bassett Jr, 1978; Koenker, 2005; Takeuchi and Furuhashi, 2004) aims at estimating quantile levels of the output conditional distribution $\rho_{\mathcal{Y}|\mathcal{X}}$, which is very helpful in many applications such as medicine, economics, social sciences or ecology. To this end, Koenker and Bassett Jr (1978) introduced the pinball loss. Let $d \in \mathbb{N}^*$, $(\tau_i)_{i \leq d} \in (0,1)$ the quantile levels to predict, $\mathbb{1}_d = (1,\dots,1)^\top \in \mathbb{R}^d$, for an input/output pair $(x,y) \in \mathcal{X} \times \mathbb{R}$ and an estimator $f : \mathcal{X} \to \mathbb{R}^d$, the pinball loss is given by

$$\ell(f(x), y\mathbb{1}_d) = \sum_{i=1}^{d} \ell_{\tau_i}(f(x)_i, y) = \sum_{i=1}^{d} \tau_i |f(x)_i - y_i| \mathbb{1}_{\{f(x)_i - y_i \geq 0\}} + (1-\tau_i)|f(x)_i - y_i| \mathbb{1}_{\{f(x)_i - y_i \leq 0\}}, \tag{2.128}$$

where $\mathbb{1}_{\{\cdot\}}$ denotes the indicator function. They prove that, for $z \sim \rho_{\mathcal{Z}}$ a real-valued r. v., F its cumulative distribution function and $\tau \in (0,1)$, if

$$\hat{\mu} \in \operatorname*{arg\,min}_{\mu \in \mathbb{R}} \mathbb{E}_{z \sim \rho_{\mathcal{Z}}}[\ell_\tau(z, \mu)], \tag{2.129}$$

then

$$\hat{\mu} = F^{-1}(\tau) = \inf\{z \in \mathbb{R} : F(z) \geq \tau\} \tag{2.130}$$

is the τ-quantile of the distribution F.

Hence, joint quantile regression boils down to solving the classical ERM problem with an input matrix-valued kernel $\mathcal{K}$ and the pinball loss. As proposed by Sangnier et al. (2016) and explained in example 2.13, the decomposable kernel $\mathcal{K} = \mathrm{k}_{\mathcal{X}} M$ with $M_{ij} = \exp(-\gamma(\tau_i - \tau_j)^2)$ is well-suited for such a task since it enforces the proximity of predictions between close quantiles levels and also limits the crossing phenomenon for the predicted quantiles. Sangnier et al. (2016) and Sangnier et al. (2017) propose primal-dual coordinate descent algorithm to solve this task. In Brault et al. (2019), the authors propose to predict an infinite number of quantile levels by leveraging OVK and a parameterized expression of the solution.

In chapter 3, we conduct experiments on joint quantile regression, using the above matrix-valued kernel and our proposed sketched estimator.

2.6 Representation Learning from Complex Data

In this section, we present some techniques to learn representations from complex data. First, we present kernel learning as a way to add expressiveness to kernel methods, since we build upon IOKR in this thesis. Then, we introduce deep learning.

2.6.1 Kernel Learning

There exist different kernel learning techniques. In this thesis, we briefly present Multiple Kernel Learning (MKL), Decomposable Kernel Learning (DKL) and Deep Kernel Learning (DKL).

Multiple Kernel Learning. The idea behind MKL is to consider as a hypothesis space, the feature space induced by a weighted sum of M p. d. kernels. In fact, the sum of p. d. kernels is a p. d. kernel, and the product of a non-negative real

number with a p. d. kernel is also a p.d. kernel. Then, let $M \in \mathbb{N}^*$, $\Sigma_M = \{\eta = (\eta_1, \ldots, \eta_M)^\top \in \mathbb{R}^M : \eta_i \geq 0 \forall i \in [\![M]\!], \sum_{i=1}^M \eta_i = 1\}$, $\eta \in \Sigma_M$ and $\mathrm{k}_{\mathcal{X}1}, \ldots, \mathrm{k}_{\mathcal{X}M}$ be M p. d. kernels associated to the RKHSs $\mathcal{H}_{\mathcal{X}1}, \ldots, \mathcal{H}_{\mathcal{X}M}$. The function $\mathrm{k}_{\mathcal{X}} = \sum_{i=1}^M \eta_i \mathrm{k}_{\mathcal{X}i}$ is a p. d. kernel, its canonical feature map is the concatenation of the weighted canonical feature maps of the $\mathrm{k}_{\mathcal{X}i}$s, and its induced feature space $\mathcal{H}_{\mathcal{X}}$ is the direct sum of RKHSs, i.e. $\mathcal{H}_{\mathcal{X}} = \mathcal{H}_{\mathcal{X}1} \bigoplus \ldots \bigoplus \mathcal{H}_{\mathcal{X}M}$. The goal is then to learn the estimator induced by such a hypothesis, which implies finding the best function within each RKHS and the best weights (Lanckriet et al., 2004; Bach et al., 2004; Rakotomamonjy et al., 2008; Koltchinskii and Yuan, 2010; Gönen and Alpaydin, 2011). In particular, for a loss $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, it is possible to show that the solution $\hat{f}$ to

$$\min_{\eta \in \Sigma_M} \min_{f \in \mathcal{H}_{\mathcal{X}}} \frac{1}{n} \sum_{i=1}^n \ell(f(x_i), y_i) + \lambda \|f\|^2_{\mathcal{H}_{\mathcal{X}}}, \tag{2.131}$$

is $\hat{f} = \sum_{i=1}^M \hat{f}_i$, where $(\hat{f}_1, \ldots, \hat{f}_M) \in \mathcal{H}_{\mathcal{X}1} \times \ldots \times \mathcal{H}_{\mathcal{X}M}$ is the solution to

$$\min_{f_1 \in \mathcal{H}_{\mathcal{X}1}, \ldots, f_M \in \mathcal{H}_{\mathcal{X}M}} \frac{1}{n} \sum_{i=1}^n \ell\Big(\sum_{j=1}^M f_j(x_i), y_i\Big) + \lambda \Big(\sum_{j=1}^M \|f_j\|_{\mathcal{H}_{\mathcal{X}j}}\Big)^2. \tag{2.132}$$

Several works have used such a result to tackle SVM via MKL (Lanckriet et al., 2004; Bach et al., 2004; Rakotomamonjy et al., 2008; Gönen and Alpaydin, 2011), and Rakotomamonjy et al. (2008) proposed *SimpleMKL*, an algorithm to solve problem (2.132) using a reduced gradient algorithm and duality gap stopping criterion. Moreover, such an approach can be extended to OVK (Kadri et al., 2012). In particular, Brouard et al. (2016a) leverages MKL in the IOKR framework for metabolite identification to learn from various input kernels, thanks to the ALIGNF approach (Cortes et al., 2012). Such an approach learns the weights η_is by maximizing the centered kernel alignment between the combined kernel matrix and an ideal target kernel matrix.

Decomposable Kernel Learning. Another approach consists in learning decomposable kernels (Dinuzzo et al., 2011; Lim et al., 2015). We remind that a decomposable OVK $\mathcal{K}$ is such that $\mathcal{K} = \mathrm{k}_{\mathcal{X}} M$, where $\mathrm{k}_{\mathcal{X}}$ is a p. d. scalar-valued kernel and $M : \mathcal{Y} \to \mathcal{Y}$ is a self-adjoint positive semidefinite operator. While M is usually chosen and fixed a priori, the idea of DKL is to learn it during training. Then, let $d \in \mathbb{N}^*$ and $\mathcal{Y} = \mathbb{R}^d$, let $\mathcal{K}$ as above and $\mathcal{H}$ its vv-RKHS, and let $p \in \mathbb{N}^*$ and $\mathcal{S}_+^{d,p}$ bet the set of positive semidefinite matrices in $\mathbb{R}^{d \times d}$ whose rank is less than or equal to p, Dinuzzo et al. (2011) focus on solving

$$\min_{M \in \mathcal{S}_+^{d,p}} \min_{f \in \mathcal{H}} \sum_{i=1}^n \frac{\|f(x_i) - y_i\|_2^2}{2\lambda} + \frac{\|f\|^2_{\mathcal{H}}}{2} + \frac{\mathrm{Tr}(M)}{2}, \tag{2.133}$$

with $\lambda > 0$. The authors show that problem (2.133) can be rewritten as

$$\min_{M \in \mathcal{S}_+^{d,p}} \min_{A \in \mathbb{R}^{n \times d}} \frac{\|Y - \mathrm{K}_X A M\|_F^2}{2\lambda} + \frac{\mathrm{Tr}(A^\top \mathrm{K}_X A M)}{2} + \frac{\mathrm{Tr}(M)}{2}, \tag{2.134}$$

and they consequently propose a block coordinate descent strategy to solve problem (2.134). Such an approach is also relevant in the context of autoregressive models (Lim et al., 2015), i.e. problems where $\mathcal{X} = \mathcal{Y} = \mathbb{R}^d$ which appears in geostatistics problems, such as meteorology, for example.

However, even if MKL or DKL bring more expressiveness to kernel methods, they still fail at dealing with very complex input data such as texts. Indeed, string kernels provide good representations when the sequences of characters are not very long, and in particular when the alphabet contains a few characters, as in DNA or RNA sequences where it contains only four characters. To learn representations from sentences, it is crucial to turn towards more expressive models, typically neural networks.

Deep Kernel Learning. We first briefly present a kernel learning technique using neural networks before properly introducing deep learning. DKL (Wilson et al., 2016) consists in using neural networks to obtain parameterized deep kernels. Let $\mathcal{X}'$ be a latent space, $k_{\mathcal{X}'} : \mathcal{X}' \times \mathcal{X}' \to \mathbb{R}$ be a p. d. kernel associated to the RKHS $\mathcal{H}_{\mathcal{K}'}$ and $\phi_\theta : \mathcal{X} \to \mathcal{X}'$ be a neural network with weights $\theta \in \Theta$, then one obtains the deep kernel $k_{\mathcal{X}\theta} = k_{\mathcal{X}'}(\phi_\theta(\cdot), \phi_\theta(\cdot))$. As a consequence, in the case of supervised learning for instance, and given a loss function ℓ and a regularisation parameter $\lambda > 0$, one solves

$$\min_{\theta\in\Theta} \min_{f\in\mathcal{H}_{\mathcal{K}'}} \frac{1}{n}\sum_{i=1}^{n} \ell(f(\phi_\theta(x_i)), y_i) + \lambda\|f\|^2_{\mathcal{H}_{\mathcal{K}'}}. \tag{2.135}$$

The represented theorem then gives that the solution of the inner problem writes as $x' \in \mathcal{X}' \mapsto \sum_{i=1}^n \alpha_i k_{\mathcal{X}'}(x', \phi_\theta(x_i))$ for some α_is, and then one can learn $\hat{f} : x \in \mathcal{X} \mapsto \sum_{i=1}^n \hat{\alpha}_i k_{\mathcal{X}'}(x', \phi_\theta(x_i))$ by solving the optimisation problem over α and θ. Such an approach has been used to learn Deep Gaussian Processes (Damianou and Lawrence, 2013), estimate Exponential Family Densities (Wenliang et al., 2019), Two-Sample Tests (Liu et al., 2020) or learn a kernel family for a variety of tasks in few-shot regression settings (Tossou et al., 2019). However, it induces difficulties in learning the neural network's weights by back-propagation since the objective function is then a function of compositions between the kernel function and the neural network, in addition to causing heavy computations because of the kernel. Note that here, the kernel is applied to the neural net which is first applied to the inputs, Dührkop (2022) explores the other way around, by first applying a kernel on tandem mass spectra, and then applying a neural net to the random features obtained via Nyström approximation, as explained in section 2.3.2.

2.6.2 Deep Learning

We here introduce deep learning. We refer the reader to Goodfellow et al. (2016) for more thorough details.

Neural Networks architectures. Deep learning provides a principled way to learn from non-linear parameterized functions, namely Neural Networks. The very first and most simple example of NN architecture is the Feedforward NN.

Definition 2.46 (Feedforward Neural Network)**.** *Let $d_0 \in \mathbb{N}^*$, $\mathcal{X} = \mathbb{R}^{d_0}$ and $L \in \mathbb{N}^*$, a L-layer neural networks f is the composition of L layers $f_l : \mathbb{R}^{d_{l-1}} \to \mathbb{R}^{d_l}$ of sizes $d_1, \dots, d_L \in \mathbb{N}^*$,*

$$f_{W,b}(x) = f_L \circ \dots \circ f_1(x), \tag{2.136}$$

where each layer f_l is the composition of an affine map and a non-linear activate function $\phi : \mathbb{R} \to \mathbb{R}$ applied element-wise on the output of each layer, i.e. for all $l \in [\![L]\!]$, $i \in [\![d]\!]_l$ and $x \in \mathbb{R}^{d_0}$,

$$[f_l(x)]_i = \phi([W_l f_{l-1}(x) + b_l]_i), \tag{2.137}$$

with $W_l \in \mathbb{R}^{d_l \times d_{l-1}}$ *and* $b_l \in \mathbb{R}^{d_l}$.

The depth of a NN refers to its number of layers, and a L-layer NN is said to have $L-1$ hidden layers. FNNs deal with input vectors, however, such an architecture based on the composition of parameterized functions can be extended to other data types, based on the W_is used. For instance, Convolutional NNs (LeCun et al., 1995) extract features from images, i.e. 3D tensors, by applying filters on them. In this case, the W_is encode convolutional operations, and if the task at hand is classification with C classes, then the last layers will be fully connected outputting a vector, which corresponds to the FNN architecture. We give some examples of deep architectures:

- Feedforward NNS;
- Convolution NNs (LeCun et al., 1995) deal with pattern recognition (e.g. from images, speeches, time-series);
- Recurrent NNs (Sak et al., 2014) or Transformers (Vaswani et al., 2017) deal with sequential data (e.g. texts, biological sequences, videos);
- Graph NNs (Scarselli et al., 2009) deal with graphs (e.g. molecules, social networks);
- Generative NNs such as GANs (Goodfellow et al., 2014) or VAEs (Kingma and Welling, 2013), deal with sample generation from an unknown distribution.

In chapter 5, we conduct experiments on two molecular identification datasets where the input data are either the Simplified Molecular Input Line-Entry Systems - strings describing the chemical structure - or texts (more than 20 words) from the Chemical Entities of Biological Interest database describing the corresponding output molecules. Hence, we give some more details about the considered transformers to deal with such tasks, namely Bidirectional Encoder Representations from Transformers (Devlin et al., 2019) and SciBERT (Beltagy et al., 2019). BERT is a language representation model parameterized by a multi-layer bidirectional Transformer encoder (Vaswani et al., 2017), which is pre-trained from unlabeled text on two unsupervised tasks, called Masked Language Model and Next Sentence Prediction. The obtained representations can be then used for a supervised downstream task with the parameters of BERT fine-tuned at the same time. SCIBERT is a variant of BERT on a random sample of 1.14M papers pre-trained from Semantic Scholar (Ammar et al., 2018). See (Qiu et al., 2020) for more details about language representations with pre-trained models.

Training NNs. Given a NN $f_{W,b}$ defined as in Definition 2.46 and a loss function ℓ, the goal is then to estimate the weights $(\hat{W}, \hat{b})$ solution to the ERM problem

$$\min_{W,b} \frac{1}{n} \sum_{i=1}^{n} \ell((f_{W,b}(x_i)), y_i). \tag{2.138}$$

Doing so is challenging for several reasons. First, the above objective function is non-convex w. r. t. to the weights (W, b), hence it contains local minima, saddle points, and wide flat regions. Moreover, from a computational viewpoint, computing its gradient is not straightforward and costly. As a NN is a composition of several layers, its

gradient computation is very well-adapted to the use of the so-called *chain rule*, and then the *back-propagation* of each layer's gradient from the last to the first layer is always used to do so (Rumelhart et al., 2013). Finally, to reduce the cost of computing the gradient of $\frac{1}{n}\sum_{i=1}^{n}\ell((f_{W,b}(x_i)),y_i)$, *Stochastic Gradient Descent* (Robbins and Monro, 1951) proposes to rather use random estimations of it based on randomly drawn training samples at each gradient descent step. Moreover, SGD combines very well with techniques relying on moments of the stochastic gradient (Kingma and Ba, 2015) or second-order methods (Liu and Nocedal, 1989) to fine-tune the learning rates of the gradient descent. Finally, the rise of Graphics processing units democratizes NNs and makes them very well-suited to large-scale learning.

Deep learning theory. Unlike kernel methods, one of the biggest challenges for deep learning is to understand it in theory. From an optimisation viewpoint first, as said earlier, the non-convexity of the objective function inherent to the NNs' architectures makes it difficult to derive convergence guarantees. However, some works investigate to what extent the SGD provides very good generalisation properties to NNs either through learning rate decays (Li et al., 2019) or wide and flat local minima (He et al., 2019). Furthermore, other lines of research emerge in deep learning theory. An initial focus is on examining how over-parameterized NN behaves in terms of generalization. This is achieved by analyzing infinite-width neural networks (i.e. shallow NNs with a large number of neurons), revealing that at this extreme, a NN model can be likened to a RKHS model with a specific kernel known as the Neural Tangent Kernel (Jacot et al., 2018). This insight enables the application of generalization principles from kernel methods. Moreover, some works tackle the stability (Bousquet and Elisseeff, 2002) and generalisation error of NNs through the stability and generalisation error of GD and SGD algorithms (Charles and Papailiopoulos, 2018; Richards and Kuzborskij, 2021). Finally, Schmidt-Hieber (2017) provides excess risk bounds of NNs using ReLU activation functions in the non-parametric regression settings and exhibits the dependency of such bounds w. r. t. to many NN's parameters, such as its number of layers.

Connections with kernel methods. In addition to Neural Tangent Kernels, which are obtained via the Taylor expansion of the NN and are the inner products of two evaluations of the NN's gradient descent w. r. t. its weights, many works explore the existing links between deep learning and kernel methods. Mairal et al. (2014) introduces *Convolutional Kernel Networks*, a CNN architecture that does not learn either to represent data or to solve a classification task but learns to approximate the kernel feature map on training data to learn invariant image representations. Many works build upon CKN: Chen et al. (2019) extends it to *Recurrent Kernel Networks*, dealing with sequential data and Chen et al. (2020) extends it to *Graph Convolutional Kernel Networks*, dealing with graph-structured data. Giffon et al. (2019) uses an adaptive variant of the Nyström method for kernel approximation as a drop-in replacement for dense layers in CNNs. In order to extend the Autoencoder scheme to learn representations of structured input x lying in a Hilbert space $\mathcal{X}$, Laforgue et al. (2019) introduces *Kernel AutoEncoders*, an autoencoder whose each layer is a function defined within a vv-RKHS. Inspired by KAEs and building upon the Reproducing Kernel Hilbert C^*-Modules, a generalisation of RKHSs by means of C^*-algebra, Hashimoto et al. (2024) introduces deep RKHM, a deep architecture which is the composition of functions within RKHMs thanks to the Perron–Frobenius operator.

In this thesis, deep learning is used in chapter 5 to solve structured prediction problems with complex inputs such as images or texts. In particular, we show how to leverage kernel-induced losses with neural networks, or equivalently how to leverage neural networks within Output Kernel Regression, and conduct experiments on molecular identification with input text data.

3

Fast Kernel Methods for Generic Lipschitz Losses via p-Sparsified Sketches

Contents

3.1 Introduction

In this chapter, we first focus on the input kernel and study how to use sketching to scale scalar-valued and matrix-valued kernel machines up. Sketching, which consists of looking for solutions among a subspace of reduced dimension, is a well-studied approach to alleviate these computational burdens. However, statistically accurate sketches, such as the Gaussian one, usually contain few null entries, such that their application to kernel methods and their non-sparse Gram matrices remains slow in practice. Here, we show that sparsified Gaussian (and Rademacher) sketches still produce theoretically valid approximations while allowing for important time and space savings thanks to an efficient *decomposition trick*. To support our method, we derive excess risk bounds for both single and multiple output kernel problems, with generic Lipschitz losses, hereby providing new guarantees for a wide range of applications, from robust regression to multiple quantile regression. Our theoretical results are complemented with experiments showing the empirical superiority of our approach over state-of-the-art sketching methods.

Contributions. Our goal is to provide a framework to speed up both scalar and matrix-valued kernel methods which is as general as possible while maintaining good theoretical guarantees. For that purpose, we present three contributions, which may be of independent interest.

- We derive excess risk bounds for sketched kernel machines with generic Lipschitz-continuous losses, both in the scalar and multiple output cases. We hereby solve an open problem from Yang et al. (2017), and provide a first analysis to the sketching of vector-valued kernel methods.

- We show that sparsified Gaussian and Rademacher sketches provide valid approximations when applied to kernel methods. They maintain theoretical guarantees while inducing important space and computation savings, as opposed to plain sketches.

- We discuss how to learn these new sketched kernel machines, through an approximated feature map. We finally present experiments using Lipschitz losses, such as robust and quantile regression, on both synthetic and real-world datasets, supporting the relevance of our approach.

- A Python implementation of our approach is publicly available on GitHub.

3.2 Sketching Kernels Machines with Lipschitz-Continuous Losses

In this section, we derive excess risk bounds for sketched kernel machines with generic Lipschitz losses, for both scalar and multiple output regression.

3.2.1 Scalar Kernel Machines

We consider a general regression framework, from an input space $\mathcal{X}$ to some scalar output space $\mathcal{Y} \subseteq \mathbb{R}$. Given a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ such that $z \mapsto \ell(z, y)$ is proper, lower semi-continuous and convex for every y, our goal is to estimate $f^* = \arg\inf_{f \in \mathcal{H}} \mathbb{E}_{(X,Y)\sim\rho}\left[\ell\left(f(X), Y\right)\right]$, where $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ is a hypothesis set, and ρ is a joint distribution over $\mathcal{X} \times \mathcal{Y}$. Since ρ is usually unknown, we assume that we have access to a training dataset $\{(x_i, y_i)\}_{i=1}^n$ composed of i.i.d. realisations drawn from ρ. We recall the definitions of a scalar-valued kernel and its RKHS (Aronszajn, 1950).

Definition 3.1 (Scalar-valued kernel)**.** *A scalar-valued kernel is a symmetric function* $\mathrm{k}_{\mathcal{X}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ *such that for all* $n \in \mathbb{N}$*, and any* $\left(x_i\right)_{i=1}^n \in \mathcal{X}^n$*,* $\left(\alpha_i\right)_{i=1}^n \in \mathbb{R}^n$*, we have* $\sum_{i,j=1}^n \alpha_i \, \mathrm{k}_{\mathcal{X}}\left(x_i, x_j\right)\alpha_j \geq 0$*.*

Theorem 3.2 (RKHS)**.** *Let* $\mathrm{k}_{\mathcal{X}}$ *be a kernel on* $\mathcal{X}$*. Then, there exists a unique Hilbert space of functions* $\mathcal{H}_{\mathcal{X}} \subset \mathbb{R}^{\mathcal{X}}$ *such that* $\mathrm{k}_{\mathcal{X}}\left(\cdot, x\right) \in \mathcal{H}_{\mathcal{X}}$ *for all* $x \in \mathcal{X}$*, and such that we have* $h(x) = \langle h, \mathrm{k}_{\mathcal{X}}\left(\cdot, x\right)\rangle_{\mathcal{H}_{\mathcal{X}}}$ *for any* $\left(h, x\right) \in \mathcal{H}_{\mathcal{X}} \times \mathcal{X}$*.*

A kernel machine computes a proxy for f^* by solving

$$\min_{f \in \mathcal{H}_{\mathcal{X}}} \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i) + \frac{\lambda_n}{2} \|f\|_{\mathcal{H}_{\mathcal{X}}}^2, \tag{3.1}$$

where $\lambda_n > 0$ is a regularization parameter. By the representer theorem (Kimeldorf and Wahba, 1971; Schölkopf et al., 2001), the solution to Problem (3.1) is given by $\hat{f} = \sum_{i=1}^n \hat{\alpha}_i \, \mathrm{k}_{\mathcal{X}}(\cdot, x_i)$, with $\hat{\alpha} \in \mathbb{R}^n$ the solution to

$$\min_{\alpha \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^n \ell([\mathrm{K}_{\mathrm{X}} \alpha]_i, y_i) + \frac{\lambda_n}{2} \alpha^\top \mathrm{K}_{\mathrm{X}} \alpha \,, \tag{3.2}$$

where $\mathrm{K}_{\mathrm{X}} \in \mathbb{R}^{n \times n}$ is the kernel Gram matrix such that $\mathrm{K}_{\mathrm{X}ij} = \mathrm{k}_{\mathcal{X}}(x_i, x_j)$.

Definition 3.3 (Regularized Kernel-based Sketched Estimator)**.** *Given a matrix $\mathrm{R}_{\mathcal{X}} \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}} \times n}$, with $\mathrm{m}_{\mathcal{X}} \ll n$, sketching consists in imposing the substitution $\alpha = \mathrm{R}_{\mathcal{X}}{}^\top \gamma$ in the empirical risk minimization problem stated in eq. (3.2). We then obtain an optimisation problem of reduced size on γ, that yields the sketched estimator $\tilde{f} = \sum_{i=1}^n [\mathrm{R}_{\mathcal{X}}{}^\top \tilde{\gamma}]_i \, \mathrm{k}_{\mathcal{X}}(\cdot, x_i)$, where $\tilde{\gamma} \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}}}$ is a solution to*

$$\min_{\gamma \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}}}} \frac{1}{n} \sum_{i=1}^n \ell([\mathrm{K}_{\mathrm{X}} \mathrm{R}_{\mathcal{X}}{}^\top \gamma]_i, y_i) + \frac{\lambda_n}{2} \gamma^\top \mathrm{R}_{\mathcal{X}} \mathrm{K}_{\mathrm{X}} \mathrm{R}_{\mathcal{X}}{}^\top \gamma \,. \tag{3.3}$$

In practice, one usually obtains the matrix $\mathrm{R}_{\mathcal{X}}$ by sampling it from a random distribution. The literature is rich in examples of distributions that can be used to generate the sketching matrix $\mathrm{R}_{\mathcal{X}}$. For instance, the sub-sampling matrices, where each line of $\mathrm{R}_{\mathcal{X}}$ is sampled from I_n, have been widely studied in the context of kernel methods. They are computationally efficient from both time and space perspectives and yield the so-called Nyström approach (Williams and Seeger, 2001; Rudi et al., 2015). More complex distributions, such as Randomized Orthogonal System (ROS) sketching or Gaussian sketch matrices, have also been considered (Yang et al., 2017). In this work, we first give a general theoretical analysis of regularized kernel-based sketched estimators for any K_{X}-satisfiable sketch matrix (Definition 3.4). Then, we introduce the p-sparsified sketches and prove their K_{X}-satisfiablity, as well as their relevance for kernel methods in terms of statistical and computational trade-off.

Works about sketched kernel machines usually assess the performance of $\tilde{f}$ by upper bounding its squared $L^2(\mathbb{P}_N)$ error, i.e., $(1/n)\sum_{i=1}^n (\tilde{f}(x_i) - f_{\mathcal{H}_{\mathcal{X}}}(x_i))^2$, where $f_{\mathcal{H}_{\mathcal{X}}}$ is the minimizer of the true risk over $\mathcal{H}_{\mathcal{X}}$, supposed to be attained (Yang et al., 2017, Equation 2), or through its (relative) recovery error $\|\tilde{f} - \hat{f}\|_{\mathcal{H}_{\mathcal{X}}} / \|\hat{f}\|_{\mathcal{H}_{\mathcal{X}}}$, see Theorem 3 in Lacotte and Pilanci (2022). In contrast, we focus on the excess risk of $\tilde{f}$, the original quantity of interest. As revealed by the proof of Theorem 3.10, the approximation error of the excess risk can be controlled in terms of the $L^2(\mathbb{P}_N)$ error, and we actually recover the results from Yang et al. (2017) when we particularize to the square loss with bounded outputs (second bound in Theorem 3.10). Furthermore, studying the excess risk allows to better position the performances of $\tilde{f}$ among the known off-the-shelf kernel-based estimators available for the targeted problem. To achieve this study, we rely on the key notion of K_{X}-satisfiability for a sketch matrix (Yang et al., 2017; Liu et al., 2019; Chen and Yang, 2021a).

Let $\mathrm{K}_{\mathrm{X}}/n = UDU^\top$ be the eigendecomposition of the Gram matrix, where $D = (1/n) \cdot \mathrm{diag}(\sigma_1(\mathrm{K}_{\mathrm{X}}), \ldots, \sigma_n(\mathrm{K}_{\mathrm{X}}))$ stores the eigenvalues of $\mathrm{K}_{\mathrm{X}}/n$ in decreasing order. Let δ_n^2 be the critical radius of $\mathrm{K}_{\mathrm{X}}/n$, i.e., the lowest value such that $\psi(\delta_n) = (\frac{1}{n}\sum_{i=1}^n \min(\delta_n^2, \sigma_i(\mathrm{K}_{\mathrm{X}})/n))^{1/2} \leq \delta_n^2$. The existence and uniqueness of δ_n^2 is guaranteed for any RKHS associated with a positive definite kernel (Bartlett et al., 2006; Yang et al., 2017). Note that δ_n^2 is similar to the parameter $\tilde{\varepsilon}^2$ used in Yang et al. (2012) to analyze Nyström

approximation for kernel methods. We define the statistical dimension of K_X as $d_n = \min\left\{j \in \{1,\dots,n\} \colon \sigma_j(\mathrm{K}_X)/n \leq \delta_n^2\right\}$, with $d_n = n$ if no such index j exists.

Definition 3.4 (K_X-satisfiability, Yang et al. 2017)**.** *Let $c > 0$ be independent of n, $U_1 \in \mathbb{R}^{n \times d_n}$ and $U_2 \in \mathbb{R}^{n \times (n-d_n)}$ be the left and right blocks of the matrix U previously defined, and $D_2 = (1/n)\operatorname{diag}\left(\sigma_{d_n+1}(\mathrm{K}_X), \dots, \sigma_n(\mathrm{K}_X)\right)$. A matrix $\mathrm{R}_{\mathcal{X}}$ is said to be K_X-satisfiable for c if we have*

$$\left\|\left(\mathrm{R}_{\mathcal{X}} U_1\right)^\top \mathrm{R}_{\mathcal{X}} U_1 - I_{d_n}\right\|_{\mathrm{op}} \leq 1/2, \quad \text{and} \quad \left\|\mathrm{R}_{\mathcal{X}} U_2 D_2^{1/2}\right\|_{\mathrm{op}} \leq c\delta_n. \tag{3.4}$$

Roughly speaking, a matrix is K_X-satisfiable if it defines an isometry on the largest eigenvectors of K_X, and has a small operator norm on the smallest eigenvectors. For random sketching matrices, it is common to show K_X-satisfiability with high probability under some condition on the sketch size $\mathrm{m}_{\mathcal{X}}$, see e.g., Yang et al. (2017, Lemma 5) for Gaussian sketches, Chen and Yang (2021a, Theorem 8) for Accumulation sketches. In Section 3.3, we show similar results for p-sparsified sketches.

To derive our excess risk bounds, we place ourselves in the framework of Li et al. (2021), see Sections 2.1 and 3 therein. Namely, we assume that the true risk is minimized over $\mathcal{H}_{\mathcal{X}}$ at $f_{\mathcal{H}_{\mathcal{X}}} := \arg\min_{f \in \mathcal{H}_{\mathcal{X}}} \mathbb{E}\left[\ell\left(f\left(X\right), Y\right)\right]$. The existence of $f_{\mathcal{H}_{\mathcal{X}}}$ is standard in the literature (Caponnetto and De Vito, 2007; Rudi and Rosasco, 2017; Yang et al., 2017), and implies that $f_{\mathcal{H}_{\mathcal{X}}}$ has bounded norm, see e.g., Rudi and Rosasco (2017, Remark 2). Similarly to Li et al. (2021), we also assume that estimators returned by Empirical Risk Minimization have bounded norms. Hence, all estimators considered in the present paper belong to some ball of finite radius R. However, we highlight that our results do not require prior knowledge on R, and hold uniformly for all finite R. As a consequence, we consider without loss of generality as hypothesis set the unit ball $\mathcal{B}\left(\mathcal{H}_{\mathcal{X}}\right)$ in $\mathcal{H}_{\mathcal{X}}$, up to an *a posteriori* rescaling of the bounds by R to recover the general case.

Assumption 3.5. *The true risk is minimized at $f_{\mathcal{H}_{\mathcal{X}}}$.*

Assumption 3.6. *The hypothesis set considered is $\mathcal{B}\left(\mathcal{H}_{\mathcal{X}}\right)$.*

Assumption 3.7. *For all $y \in \mathcal{Y}$, $z \mapsto \ell(z, y)$ is L-Lipschitz, for $L > 0$.*

Assumption 3.8. *For all $x \in \mathcal{X}$, we have $k(x, x) \leq \kappa_{\mathcal{X}}$.*

Assumption 3.9. *The sketch $\mathrm{R}_{\mathcal{X}}$ is K_X-satisfiable with constant $c > 0$.*

Note that we discuss some directions to relax Assumption 3.6 in Appendix A.2. Many loss functions satisfy Assumption 3.7, such as the hinge loss ($L = 1$), used in SVMs (Cortes and Vapnik, 1995), the ϵ-insensitive ℓ_1 (Drucker et al., 1997), the κ-Huber loss, known for robust regression (Huber, 1964), the pinball loss, used in quantile regression (Steinwart and Christmann, 2011), or the square loss with bounded outputs. Assumption 3.8 is standard (e.g., $\kappa_{\mathcal{X}} = 1$ for the Gaussian kernel). Under Assumptions 3.5 to 3.9 we have the following result.

Theorem 3.10. *Let $\tilde{f}$ as in Definition 3.3, suppose that Assumptions 3.5 to 3.9 hold, and let $C = 1 + \sqrt{6}c$, with c the constant from Assumption 3.9. Then, for any $\delta \in (0,1)$ with probability at least $1-\delta$ we have*

$$\mathbb{E}\left[\ell_{\tilde{f}}\right] \leq \mathbb{E}\left[\ell_{f_{\mathcal{H}_\mathcal{X}}}\right] + LC\sqrt{\lambda_n + \delta_n^2} + \frac{\lambda_n}{2} + 8L\sqrt{\frac{\kappa_\mathcal{X}}{n}} + 2\sqrt{\frac{8\log\left(4/\delta\right)}{n}}, \tag{3.5}$$

where $\mathbb{E}\left[\ell_f\right] = \mathbb{E}_{(X,Y)\sim\rho}\left[\ell(f(X),Y)\right]$. Furthermore, if $\ell\left(z,y\right) = \left(z-y\right)^2/2$ and $\mathcal{Y} \subset [0,1]$, with probability at least $1-\delta$ we have

$$\mathbb{E}\left[\ell_{\tilde{f}}\right] \leq \mathbb{E}\left[\ell_{f_{\mathcal{H}_\mathcal{X}}}\right] + \left(C^2 + \frac{1}{2}\right)\lambda_n + C^2\delta_n^2 + 8\frac{\kappa_\mathcal{X} + \sqrt{\kappa_\mathcal{X}}}{\sqrt{n}} + 2\sqrt{\frac{8\log\left(4/\delta\right)}{n}}. \tag{3.6}$$

Proof [Proof sketch] The proof relies on the decomposition of the excess risk into two generalization error terms and an approximation error term, i.e.,

$$\mathbb{E}[\ell_{\tilde{f}}] - \mathbb{E}[\ell_{f_{\mathcal{H}_\mathcal{X}}}] = \mathbb{E}[\ell_{\tilde{f}}] - \mathbb{E}_n[\ell_{\tilde{f}}] + \mathbb{E}_n[\ell_{\tilde{f}}] - \mathbb{E}_n[\ell_{f_{\mathcal{H}_\mathcal{X}}}] + \mathbb{E}_n[\ell_{f_{\mathcal{H}_\mathcal{X}}}] - \mathbb{E}[\ell_{f_{\mathcal{H}_\mathcal{X}}}], \tag{3.7}$$

where $\mathbb{E}_n[\ell_f] = (1/n)\sum_{i=1}^n \ell(f(x_i), y_i)$. The two generalization errors (of $\tilde{f}$ and $f_{\mathcal{H}_\mathcal{X}}$) can be bounded using Bartlett and Mendelson (2003, Theorem 8) together with Assumptions 3.5 to 3.8. For the last term, we can use Jensen's inequality and the Lipschitz continuity of the loss to upper bound this approximation error by the square root of the sum of the square residuals of the Kernel Ridge Regression with targets the $f_{\mathcal{H}_\mathcal{X}}(x_i)$. The latter can in turn be upper bounded using Assumptions 3.5 and 3.9 and Lemma 2 from Yang et al. (2017). When considering the square loss, Jensen's inequality is not necessary anymore, leading to the improved second term in the right-hand side of the last inequality in theorem 3.10. ■

Recall that the rates in Theorem 3.10 are incomparable as is to that of Yang et al. (2017, Theorem 2), since we focus on the excess risk while the authors study the squared $L^2(\mathbb{P}_N)$ error. Precisely, we recover their results as a particular case with the square loss and bounded outputs, up to the generalization errors. Instead, note that we do recover the rates of Li et al. (2021, Theorem 1), based on a similar framework. Our bounds feature two different terms: a quantity related to the generalization errors, and a quantity governed by δ_n, deriving from the K_X-satisfiability analysis. The behaviour of the critical radius δ_n crucially depends on the choice of the kernel. In Yang et al. (2017), the authors compute its decay rate for different kernels. For instance, we have $\delta_n^2 = \mathcal{O}(\sqrt{\log(n)/n})$ for the Gaussian kernel, $\delta_n^2 = \mathcal{O}\left(1/n\right)$ for polynomial kernels, or $\delta_n^2 = \mathcal{O}(n^{-2/3})$ for first-order Sobolev kernels. Note finally that by setting $\lambda_n \propto 1/\sqrt{n}$ we attain a rate of $\mathcal{O}\left(1/\sqrt{n}\right)$, that is minimax for the kernel ridge regression, see Caponnetto and De Vito (2007).

Remark 3.11. *Note that a standard additional assumption on the second order moments of the functions in $\mathcal{H}_\mathcal{X}$ (Bartlett et al., 2005) allows to derive refined learning rates for the generalization errors. These refined rates are expressed in terms of $\hat{r}^\star_{\mathcal{H}_\mathcal{X}}$, the fixed point of a new sub-root function $\hat{\psi}_n$. In order to make the approximation error of the same order, it is then necessary to prove the K_X-satisfiability of $\mathrm{R}_\mathcal{X}$ with respect to $\hat{r}^{\star 2}_{\mathcal{H}_\mathcal{X}}$ instead of δ_n^2. Whether it is possible to prove such a K_X-satisfiability for standard sketches is however a nontrivial question, left as future work.*

3.2.2 Matrix-valued Kernel Machines

In this section, we extend our results to multiple output regression, tackled in vector-valued RKHSs. Note that the output space $\mathcal{Y}$ is now a subset of $\mathbb{R}^d$, with $d \geq 2$. We start by recalling important notions about Matrix-Valued Kernels (MVKs) and vector-valued RKHSs (vv-RKHSs).

Definition 3.12 (Matrix-valued kernel)**.** *A MVK is an application $\mathcal{K} : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathbb{R}^d)$, where $\mathcal{L}(\mathbb{R}^d)$ is the set of bounded linear operators on $\mathbb{R}^d$, such that $\mathcal{K}\left(x,x'\right) = \mathcal{K}\left(x',x\right)^\top$ for all $(x,x') \in \mathcal{X}^2$, and such that for all $n \in \mathbb{N}$ and any $\left(x_i,y_i\right)_{i=1}^n \in (\mathcal{X}\times\mathcal{Y})^n$ we have $\sum_{i,j=1}^n y_i^\top \mathcal{K}\left(x_i,x_j\right) y_j \geqslant 0$.*

Theorem 3.13 (Vector-valued RKHS)**.** *Let $\mathcal{K}$ be a MVK. There is a unique Hilbert space $\mathcal{H} \subset \mathcal{F}(\mathcal{X},\mathbb{R}^d)$, the vv-RKHS of $\mathcal{K}$, such that for all $x \in \mathcal{X}$, $y \in \mathbb{R}^d$ and $f \in \mathcal{H}$ we have $x' \mapsto \mathcal{K}\left(x,x'\right)y \in \mathcal{H}$, and $\langle f, \mathcal{K}\left(\cdot,x\right)y\rangle_{\mathcal{H}} = f(x)^\top y$.*

Note that we focus in this paper on the finite-dimensional case, i.e., $\mathcal{Y} \subset \mathbb{R}^d$, such that for all $x,x' \in \mathcal{X}$, we have $\mathcal{K}(x,x') \in \mathbb{R}^{d\times d}$. For a training sample $\{x_1,\dots,x_n\}$, we define the Gram matrix as $\mathbf{K} = \left(\mathcal{K}(x_i,x_j)\right)_{1\leq i,j\leq n} \in \mathbb{R}^{nd\times nd}$. A common assumption consists in considering decomposable kernels: we assume that there exists a scalar kernel $\mathrm{k}_{\mathcal{X}}$ and a positive semi-definite matrix $M \in \mathbb{R}^{d\times d}$ such that for all $x,x' \in \mathcal{X}$ we have $\mathcal{K}(x,x') = \mathrm{k}_{\mathcal{X}}(x,x')M$. The Gram matrix can then be written $\mathbf{K} = \mathrm{K}_{\mathrm{X}} \otimes M$, where $\mathrm{K}_{\mathrm{X}} \in \mathbb{R}^{n\times n}$ is the scalar Gram matrix, and $\otimes$ denotes the Kronecker product. Decomposable kernels are widely spread in the literature as they provide a good compromise between computational simplicity and expressivity —note that in particular, they encapsulate independent learning, achieved with $M = I_d$. We now discuss two examples of relevant output matrices.

Example 3.14. *In joint quantile regression, one is interested in predicting d different conditional quantiles of an output y given the input x. If $(\tau_i)_{i\leq d} \in (0,1)$ denote the d different quantile levels, it has been shown in Sangnier et al. (2016) that choosing $M_{ij} = \exp(-\gamma(\tau_i - \tau_j)^2)$ favors close predictions for close quantile levels, while limiting crossing effects.*

Example 3.15. *In multiple output regression, it is possible to leverage prior knowledge of the task relationships to design a relevant output matrix M. For instance, let P be the $d \times d$ adjacency matrix of a graph in which the vertices are the tasks and an edge exists between two tasks if and only if they are (thought to be) related. Denoting by L_P the graph Laplacian associated to P, Evgeniou et al. (2005) and Sheldon (2008) have proposed to use $M = \left(\mu L_P + (1-\mu)I_d\right)^{-1}$, with $\mu \in [0,1]$. When $\mu = 0$, we have $M = I_d$ and all tasks are considered independent. When $\mu = 1$, we only rely on the prior knowledge encoded in P.*

Given a sample $\left(x_i,y_i\right)_{i=1}^n \in \left(\mathcal{X},\mathbb{R}^d\right)^n$ and a decomposable kernel $\mathcal{K} = \mathrm{k}_{\mathcal{X}} M$ (its associated vv-RKHS is $\mathcal{H}$), the penalized empirical risk minimisation problem is

$$\min_{f\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^n \ell(f(x_i),y_i) + \frac{\lambda_n}{2}\|f\|_{\mathcal{H}}^2, \tag{3.8}$$

where $\ell : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a loss such that $z \mapsto \ell(z, y)$ is proper, lower semi-continuous and convex for all $y \in \mathbb{R}^d$. By the vector-valued representer theorem (Micchelli and Pontil, 2005), we have that the solution to Problem (3.8) writes $\hat{f} = \sum_{j=1}^n \mathcal{K}(\cdot, x_j)\hat{\alpha}_j = \sum_{j=1}^n \mathrm{k}_{\mathcal{X}}(\cdot, x_j) M \hat{\alpha}_j$, where $\hat{A} = \left(\hat{\alpha}_1, \dots, \hat{\alpha}_n\right)^\top \in \mathbb{R}^{n \times d}$ is the solution to the problem

$$\min_{A \in \mathbb{R}^{n \times d}} \frac{1}{n} \sum_{i=1}^n \ell\left(\left[\mathrm{K}_{\mathrm{X}} A M\right]_{i:}^\top, y_i\right) + \frac{\lambda_n}{2} \operatorname{Tr}\left(\mathrm{K}_{\mathrm{X}} A M A^\top\right).$$

In this context, sketching consists in making the substitution $A = \mathrm{R}_{\mathcal{X}}{}^\top \Gamma$, where $\mathrm{R}_{\mathcal{X}} \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}} \times n}$ is a sketch matrix and $\Gamma \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}} \times d}$ is the parameter of reduced dimension to be learned. The solution to the sketched problem is then $\tilde{f} = \sum_{j=1}^n \mathrm{k}_{\mathcal{X}}(\cdot, x_j) M \left[\mathrm{R}_{\mathcal{X}}{}^\top \tilde{\Gamma}\right]_{j:}$, with $\tilde{\Gamma} \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}} \times d}$ minimizing

$$\frac{1}{n} \sum_{i=1}^n \ell\left(\left[\mathrm{K}_{\mathrm{X}} \mathrm{R}_{\mathcal{X}}{}^\top \Gamma M\right]_{i:}, y_i\right) + \frac{\lambda_n}{2} \operatorname{Tr}\left(\mathrm{R}_{\mathcal{X}} \mathrm{K}_{\mathrm{X}} \mathrm{R}_{\mathcal{X}}{}^\top \Gamma M \Gamma^\top\right).$$

Theorem 3.16. *Suppose that Assumptions 3.5 to 3.9 hold, that $\mathcal{K} = \mathrm{k}_{\mathcal{X}} M$ is a decomposable kernel with M invertible, and let C as in Theorem 3.10. Then for any $\delta \in (0, 1)$ with probability at least $1 - \delta$ we have*

$$\mathbb{E}\left[\ell_{\tilde{f}}\right] \leq \mathbb{E}\left[\ell_{f_{\mathcal{H}}}\right] + LC\sqrt{\lambda_n + \|M\|_{\mathrm{op}}\, \delta_n^2} + \frac{\lambda_n}{2} + 8L\sqrt{\frac{\kappa_{\mathcal{X}} \operatorname{Tr}\left(M\right)}{n}} + 2\sqrt{\frac{8 \log\left(4/\delta\right)}{n}}. \tag{3.9}$$

Furthermore, if $\ell\left(z, y\right) = \left\|z - y\right\|_2^2 / 2$ and $\mathcal{Y} \subset \mathcal{B}\left(\mathbb{R}^d\right)$, with probability at least $1 - \delta$ we have that

$$\begin{aligned}\mathbb{E}\left[\ell_{\tilde{f}}\right] \leq \mathbb{E}\left[\ell_{f_{\mathcal{H}}}\right] + \left(C^2 + \frac{1}{2}\right)\lambda_n + C^2 \|M\|_{\mathrm{op}}\, \delta_n^2 \\ + 8 \operatorname{Tr}\left(M\right)^{1/2} \frac{\kappa_{\mathcal{X}} \left\|M\right\|_{\mathrm{op}}^{1/2} + \kappa_{\mathcal{X}}{}^{1/2}}{\sqrt{n}} + 2\sqrt{\frac{8 \log\left(4/\delta\right)}{n}}.\end{aligned} \tag{3.10}$$

Proof [Proof sketch.] The proof follows that of Theorem 3.10. The main challenge is to adapt Yang et al. (2017, Lemma 2) to the multiple output setting. To do so, we leverage that $\mathcal{K}$ is decomposable, such that the K_{X}-satisfiability of $\mathrm{R}_{\mathcal{X}}$ is sufficient, where K_{X} the scalar Gram matrix. ■

Note that for $M = I_d$ (independent prior), the third term of the right-hand side of both inequalities becomes of order $\sqrt{d/n}$, that is typical of multiple output problems. If moreover we instantiate the bound for $d = 1$, we recover exactly Theorem 3.10. Finally, similarly to the scalar case in theorem 3.10, looking at the least square case (eq. (3.10)), by setting $\lambda_n \propto 1/\sqrt{n}$, we attain the minimax rate of $\mathcal{O}(1/\sqrt{n})$, as stated in Caponnetto and De Vito (2007) and Ciliberto et al. (2020, Theorem 5). To the best of our knowledge, Theorem 3.16 is the first theoretical result about sketched vector-valued kernel machines. We highlight that it applies to generic Lipschitz losses and provides a bound directly on the excess risk.

3.2.3 Algorithmic details

We now discuss how to solve single and multiple output optimization problems. Let $\{(\sigma_i(\widetilde{\mathrm{K}}_\mathrm{X}), \tilde{\mathbf{u}}_i), i \in [\mathrm{m}_\mathcal{X}]\}$ be the eigenpairs of $\widetilde{\mathrm{K}}_\mathrm{X} = \mathrm{R}_\mathcal{X}\mathrm{K}_\mathrm{X}\mathrm{R}_\mathcal{X}^\top$ in descending order, $\tilde{U} = [\tilde{U}_{ij}]_{\mathrm{m}_\mathcal{X}\times\mathrm{m}_\mathcal{X}} = \left(\tilde{\mathbf{u}}_1, \ldots, \tilde{\mathbf{u}}_{\mathrm{m}_\mathcal{X}}\right)$, $\mathrm{p}_\mathrm{X} = \mathrm{rank}(\widetilde{\mathrm{K}}_\mathrm{X})$, $\tilde{D}_{\mathrm{p}_\mathrm{X}} = \mathrm{diag}(\sigma_1(\widetilde{\mathrm{K}}_\mathrm{X}), \ldots, \sigma_{\mathrm{p}_\mathrm{X}}(\widetilde{\mathrm{K}}_\mathrm{X}))$, and $\tilde{U}_{\mathrm{p}_\mathrm{X}} = (\tilde{\mathbf{u}}_1, \ldots, \tilde{\mathbf{u}}_{\mathrm{p}_\mathrm{X}})$.

Proposition 3.17. *Solving Problem* (3.3) *is equivalent to solving*

$$\min_{\omega\in\mathbb{R}^{\mathrm{p}_\mathrm{X}}} \frac{1}{n}\sum_{i=1}^{n} \ell\left(\omega^\top \tilde{\psi}_\mathcal{X}\left(x_i\right), y_i\right) + \frac{\lambda_n}{2}\|\omega\|_2^2, \tag{3.11}$$

where $\tilde{\psi}_\mathcal{X}(x) = \tilde{D}_{\mathrm{p}_\mathrm{X}}^{-1/2}\tilde{U}_{\mathrm{p}_\mathrm{X}}^\top \mathrm{R}_\mathcal{X}\left(\mathrm{k}_\mathcal{X}\left(x, x_1\right), \ldots, \mathrm{k}_\mathcal{X}\left(x, x_n\right)\right)^\top \in \mathbb{R}^{\mathrm{p}_\mathrm{X}}$.

The proof is available in Appendix A.1. Problem (3.11) thus writes as a linear problem with respect to the feature maps induced by the sketch, generalizing the results established in Yang et al. (2012) for sub-sampling sketches. When considering multiple outputs, it is also possible to derive a linear feature map version when the kernel is decomposable. These feature maps are of the form $\tilde{\psi}_\mathcal{X} \otimes M^{1/2}$, yielding matrices of size $nd \times \mathrm{p}_\mathrm{X} d$ that are prohibitive in terms of space. Note that an alternative way is to see sketching as a projection of the $\mathrm{k}_\mathcal{X}(\cdot, x_i)$ into $\mathbb{R}^{\mathrm{p}_\mathrm{X}}$ (Chatalic et al., 2022). Instead, we directly learn Γ. For both single and multiple output problems, we consider losses not differentiable everywhere in section 3.4 and apply ADAM Stochastic Subgradient Descent (Kingma and Ba, 2015) for its ability to handle large datasets.

Discussion with dual implementation. In the previous sections, sketching is always leveraged in primal problems. However, for some of the loss functions we consider, dual problems are usually more attractive (Cortes and Vapnik, 1995; Laforgue et al., 2020). This naturally raises the question of investigating the interplay between sketching and duality on the algorithmic level.

The first idea consists in computing the dual problem to the sketched problem (3.3). It writes

$$\min_{\zeta\in\mathbb{R}^n} \sum_{i=1}^{n} \ell_i^\star\left(-\zeta_i\right) + \frac{1}{2\lambda_n n}\zeta^\top \mathrm{K}_\mathrm{X}\mathrm{R}_\mathcal{X}^\top(\mathrm{R}_\mathcal{X}\mathrm{K}_\mathrm{X}\mathrm{R}_\mathcal{X}^\top)^\dagger \mathrm{R}_\mathcal{X}\mathrm{K}_\mathrm{X}\zeta, \tag{3.12}$$

where $\ell_i = \ell(\cdot, y_i)$, and $f^\star$ denotes the Fenchel-Legendre transform of f, such that $f^\star(\theta) = \sup_x \langle\theta, x\rangle - f(x)$, see Appendix A.1 for the proof. First note that sketching with a subsampling matrix in the primal is thus equivalent to using a Nyström approximation in the dual. This remark generalizes for any loss function the observation made in Yang et al. (2017) for the kernel Ridge regression. However, although the $\ell_i^\star$ might be easier to optimize, solving (3.12) seems not a meaningful option, as duality brought us back to optimizing over $\mathbb{R}^n$, what we initially intended to avoid. The natural alternative thus appears to use duality first and then sketching. The resulting problem writes

$$\min_{\theta\in\mathbb{R}^{\mathrm{m}_\mathcal{X}}} \sum_{i=1}^{n} \ell_i^\star\left(-[\mathrm{R}_\mathcal{X}^\top\theta]_i\right) + \frac{1}{2\lambda_n n}\theta^\top \mathrm{R}_\mathcal{X}\mathrm{K}_\mathrm{X}\mathrm{R}_\mathcal{X}^\top\theta. \tag{3.13}$$

It is interesting to note that (3.13) is also the sketched version of Problem (3.12), which we recall is itself the dual to the sketched primal problem. Hence, sketching in the dual can be seen as a double approximation. As a consequence, the objective value

reached by minimizing (3.13) is always larger than that achieved by minimizing (3.3), and theoretical guarantees for such an approach are likely to be harder to obtain. Another limitation of (3.13) regards the $\ell_i^\star(-[\mathrm{R}_\mathcal{X}{}^\top \theta]_i)$. Indeed, these terms generally contain the non-differentiable part of the objective function (for the ϵ-insensitive Ridge regression we have $\sum_i \ell_i^\star(\theta_i) = \frac{1}{2}\|\theta\|_2^2 + \langle \theta, y\rangle + \epsilon\|\theta\|_1$ for instance), and are usually minimized by proximal gradient descent. However, using a similar approach for (3.13) is impossible since the proximal operator of $\ell_i^\star(\mathrm{R}_\mathcal{X}{}^\top \cdot)$ is only computable if $\mathrm{R}_\mathcal{X}{}^\top \mathrm{R}_\mathcal{X} = I_n$, which is never the case. Instead, one may use a primal-dual algorithm (Chambolle et al., 2018; Vu, 2011; Condat, 2013), which solves the saddle-point optimization problem of the Lagrangian, but maintains a dual variable in $\mathbb{R}^n$. Coordinate descent versions of such algorithms (Fercoq and Bianchi, 2019; Alacaoglu et al., 2020) may also be considered, as they leverage the possible sparsity of S to reduce the per-iteration cost. In order to converge, these algorithms however require a number of iterations that are of the order of n, making them hardly relevant in the large-scale setting we consider.

For all the reasons listed above, we thus believe that minimizing (3.12) or (3.13) is not theoretically relevant nor computationally attractive and that running stochastic (sub-)gradient descent on the primal problem, as detailed at the beginning of the section, is the best way to proceed algorithmically despite the possibly more elegant dual formulations. Finally, we highlight that although the condition $\mathrm{R}_\mathcal{X}{}^\top \mathrm{R}_\mathcal{X} = I_n$ is almost surely not verified (we have $\mathrm{R}_\mathcal{X} \in \mathbb{R}^{\mathrm{m}_\mathcal{X} \times n}$ with $\mathrm{m}_\mathcal{X} < n$), we still have $\mathbb{E}[\mathrm{R}_\mathcal{X}{}^\top \mathrm{R}_\mathcal{X}] = I_n$ for most sketching matrices. An interesting research direction could thus consist of running a proximal gradient descent assuming that $\mathrm{R}_\mathcal{X}{}^\top \mathrm{R}_\mathcal{X} = I_n$, and controlling the error incurred by such an approximation.

3.3 p-Sparsified Sketches

We now introduce the p-sparsified sketches, and establish their $\mathrm{K}_\mathcal{X}$-satisfiability. The p-sparsified sketching matrices are composed of i.i.d. Rademacher or centered Gaussian entries, multiplied by independent Bernoulli variables of parameter p (the non-zero entries are scaled to ensure that $\mathrm{R}_\mathcal{X}$ defines an isometry in expectation). The sketch sparsity is controlled by p, and when the latter becomes small enough, $\mathrm{R}_\mathcal{X}$ contains many columns full of zeros. It is then possible to rewrite $\mathrm{R}_\mathcal{X}$ as the product of a sub-Gaussian and a sub-sampling sketch of reduced size, which greatly accelerates the computations.

Definition 3.18. *Let $\mathrm{m}_\mathcal{X} < n$, and $p \in (0,1]$. A p-Sparsified Rademacher (p-SR) sketching matrix is a random matrix $\mathrm{R}_\mathcal{X} \in \mathbb{R}^{\mathrm{m}_\mathcal{X} \times n}$ whose entries $\mathrm{R}_{\mathcal{X}ij}$ are independent and identically distributed (i.i.d.) as follows*

$$\mathrm{R}_{\mathcal{X}ij} = \begin{cases} \frac{1}{\sqrt{\mathrm{m}_\mathcal{X} p}} & \text{with probability} \quad \frac{p}{2} \\ 0 & \text{with probability} \quad 1-p \\ \frac{-1}{\sqrt{\mathrm{m}_\mathcal{X} p}} & \text{with probability} \quad \frac{p}{2} \end{cases} \tag{3.14}$$

A p-Sparsified Gaussian (p-SG) sketching matrix is a random matrix $\mathrm{R}_\mathcal{X} \in \mathbb{R}^{\mathrm{m}_\mathcal{X} \times n}$ whose entries $\mathrm{R}_{\mathcal{X}ij}$ are i.i.d. as follows

$$\mathrm{R}_{\mathcal{X}ij} = \begin{cases} \frac{1}{\sqrt{\mathrm{m}_\mathcal{X} p}} G_{ij} & \text{with probability} \quad p \\ 0 & \text{with probability} \quad 1-p \end{cases} \tag{3.15}$$

where the G_{ij} are i.i.d. standard normal random variables. Note that standard Gaussian sketches are a special case of p-SG sketches, corresponding to $p = 1$.

Several works partially addressed p-SR sketches in the past literature. For instance, Baraniuk et al. (2008) establish that p-SR sketches satisfy the Restricted Isometry Property (based on concentration results from Achlioptas (2001)), but only for $p = 1$ and $p = 1/3$. In Li et al. (2006), the authors consider generic p-SR sketches but do not provide any theoretical result outside of a moment analysis. The *i.i.d. sparse embedding matrices* from Cohen (2016) are basically $m/\mathrm{m}_{\mathcal{X}}$-SR sketches, where $m \geq 1$, leading each column to have exactly m nonzero elements in expectation. However, we were not able to reproduce the proof of the Johnson-Linderstrauss property proposed by the author for his sketch (Theorem 4.2 in the paper, equivalent to the first claim of $\mathrm{K}_{\mathcal{X}}$-satisfiability, left-hand side of (3.4)). More precisely, we think that the assumptions considering "each entry is independently nonzero with probability $m/\mathrm{m}_{\mathcal{X}}$" and "each column has a fixed number of nonzero entries" (m here) are conflicting. As far as we know, this is the first time p-SG sketches are introduced in the literature. Note that both (3.14) and (3.15) can be rewritten as $\mathrm{R}_{\mathcal{X}ij} = (1/\sqrt{\mathrm{m}_{\mathcal{X}}\, p})B_{ij}R_{ij}$, where the B_{ij} are i.i.d. Bernouilli random variables of parameter p, and the R_{ij} are i.i.d. random variables, independent from the B_{ij}, such that $\mathbb{E}[R_{ij}] = 0$ and $\mathbb{E}[R_{ij}R_{i'j'}] = 1$ if $i = i'$ and $j = j'$, and 0 otherwise. Namely, for p-SG sketches $R_{ij} = G_{ij}$ is a standard Gaussian variable while for p-SR sketches it is a Rademacher random variable. It is then easy to check that p-SR and p-SG sketches define isometries in expectation. In the next theorem, we show that p-sparsified sketches are $\mathrm{K}_{\mathcal{X}}$-satisfiable with high probability.

Theorem 3.19. *Let $\mathrm{R}_{\mathcal{X}}$ be a p-sparsified sketching matrix. Then, there are some universal constants $C_0, C_1 > 0$ and a constant $c(p)$, increasing with p, such that for $\mathrm{m}_{\mathcal{X}} \geq \max\left(C_0 d_n/p^2, \delta_n^2 n\right)$ and with a probability at least $1 - C_1 e^{-\mathrm{m}_{\mathcal{X}} c(p)}$, the sketch $\mathrm{R}_{\mathcal{X}}$ is $\mathrm{K}_{\mathcal{X}}$-satisfiable for $c = \frac{2}{\sqrt{p}}\left(1 + \sqrt{\log\left(5\right)}\right) + 1$.*

Proof [Proof sketch.] To prove the left-hand side of (3.4), we use Boucheron et al. (2013, Theorem 2.13), which shows that any i.i.d. sub-Gaussian sketch matrix satisfies the Johnson-Lindenstrauss lemma with high probability. To prove the right-hand side of (3.4), we work conditionally on a realization of the B_{ij}, and use concentration results of Lipschitz functions of Rademacher or Gaussian random variables (Tao, 2012). We highlight that such concentration results do not hold for sub-Gaussian random variables in general, preventing from showing $\mathrm{K}_{\mathcal{X}}$-satisfiability of generic sparsified sub-Gaussian sketches. Note that having $\mathrm{R}_{\mathcal{X}ij} \propto B_{ij}R_{ij}$ is key, and that sub-sampling uniformly at random non-zero entries instead of using i.i.d. Bernoulli variables would make the proof significantly more complex. We highlight that Theorem 3.19 strictly generalizes Yang et al. (2017, Lemma 5), recovered for $p = 1$, and extends the results to Rademacher sketches. ■

Hence, by combining theorem 3.19 with either theorem 3.10 or theorem 3.16, we are able to provide the learning rate of the sketched estimator $\tilde{f}$ for classical kernel examples. We summarize in Table 3.1 the different behaviours of δ_n^2 and d_n in the different spectrum regimes considered, in order to explicit the exact condition on s in each case. More specifically, for a Dth-order polynomial kernel for instance, d_n, for any n is at most $D + 1$, leading to $\mathrm{m}_{\mathcal{X}}$ of order $D + 1$ to be sufficient. Finally, we can derive

Table 3.1: Statistical dimension, lower bound obtained on s, and learning rate obtained for excess risk with p-sparsified sketches for different kernels.

Kernel	δ_n^2	d_n	$\mathrm{m}_{\mathcal{X}}$	Learning rate
Gaussian	$\mathcal{O}\left(\frac{\sqrt{\log(n)}}{n}\right)$	$\propto \sqrt{\log(n)}$	$\Omega\left(\sqrt{\log(n)}/p^2\right)$	$\mathcal{O}\left(\frac{\left(\log(n)\right)^{1/4}}{n^{1/2}}\right)$
Polynomial	$\mathcal{O}\left(\frac{1}{n}\right)$	$\propto 1$	$\Omega\left(\frac{1}{p^2}\right)$	$\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$
Sobolev	$\mathcal{O}\left(\frac{1}{n^{2/3}}\right)$	$\propto n^{1/3}$	$\Omega\left(n^{1/3}/p^2\right)$	$\mathcal{O}\left(\frac{1}{n^{1/3}}\right)$

the learning rate obtained as well as the exact condition on $\mathrm{m}_{\mathcal{X}}$ for each scenario, see Table 3.1. Compared with Random Fourier Features (Li et al., 2021), we see that we obtain slightly degraded learning rates for Gaussian and first-order Sobolev kernels, in comparison with the $\mathcal{O}\left(1/\sqrt{n}\right)$ rate the authors obtain. Our rates remain however very close.

Computational property of p-sparsified sketches. In addition to be statistically accurate, p-sparsified sketches are computationally efficient. Indeed, recall that the main quantity one has to compute when sketching a kernel machine is the matrix $\widetilde{\mathrm{K}}_{\mathrm{X}} = \mathrm{R}_{\mathcal{X}} \mathrm{K}_{\mathrm{X}} \mathrm{R}_{\mathcal{X}}^{\top}$. With standard Gaussian sketches, that are known to be theoretically accurate, this computation takes $\mathcal{O}(\mathrm{m}_{\mathcal{X}} n^2)$ operations. Sub-sampling sketches are notoriously less precise, but since they act as masks over the Gram matrix K_{X}, computing $\widetilde{\mathrm{K}}_{\mathrm{X}}$ can be done in $\mathcal{O}(\mathrm{m}_{\mathcal{X}}{}^2)$ operations only, without having to store the entire Gram matrix upfront. Now, let $\mathrm{R}_{\mathcal{X}} \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}} \times n}$ be a p-sparsified sketch, and $\mathrm{m}_{\mathcal{X}}{}' = \sum_{j=1}^{n} \mathbb{I}\{\mathrm{R}_{\mathcal{X}:j} \neq 0_{\mathrm{m}_{\mathcal{X}}}\}$ be the number of columns of $\mathrm{R}_{\mathcal{X}}$ with at least one nonzero element. The crucial observation that makes $\mathrm{R}_{\mathcal{X}}$ computationally efficient is that we have

$$\mathrm{R}_{\mathcal{X}} = \mathrm{R}_{\mathcal{X}\mathrm{SG}} \, \mathrm{R}_{\mathcal{X}\mathrm{SS}}, \tag{3.16}$$

where $\mathrm{R}_{\mathcal{X}\mathrm{SG}} \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}} \times \mathrm{m}_{\mathcal{X}}{}'}$ is obtained by deleting the null columns from $\mathrm{R}_{\mathcal{X}}$, and $\mathrm{R}_{\mathcal{X}\mathrm{SS}} \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}}{}' \times n}$ is a sub-Sampling sketch whose sampling indices correspond to the indices of the columns in $\mathrm{R}_{\mathcal{X}}$ with at least one non-zero entry[1]. We refer to (3.16) as the *decomposition trick*. This decomposition is key, as we can apply first a fast sub-sampling sketch, and then a sub-Gaussian sketch on the sub-sampled Gram matrix of reduced size. Note that $\mathrm{m}_{\mathcal{X}}{}'$ is a random variable. By independence of the entries, each column is null with probability $\left(1-p\right)^{\mathrm{m}_{\mathcal{X}}}$. Then, by the independence of the columns we have that $\mathrm{m}_{\mathcal{X}}{}'$ follows a Binomial distribution with parameters n and $1-\left(1-p\right)^{\mathrm{m}_{\mathcal{X}}}$, such that $\mathbb{E}\left[\mathrm{m}_{\mathcal{X}}{}'\right] = n(1-\left(1-p\right)^{\mathrm{m}_{\mathcal{X}}})$. See Algorithm 3.1 for the detailed process of generating a p-sparsified sketch and decomposing it as a product of a sub-Gaussian sketch $\mathrm{R}_{\mathcal{X}\mathrm{SG}}$ and a sub-Sampling sketch $\mathrm{R}_{\mathcal{X}\mathrm{SS}}$.

Hence, the sparsity of the p-sparsified sketches, controlled by parameter p, is an interesting degree of freedom to add: it preserves statistical guarantees (Theorem 3.19) while speeding-up calculations (3.16). Of course, there is no free lunch and one loses

[1] Precisely, $\mathrm{R}_{\mathcal{X}\mathrm{SS}}$ is the identity matrix $I_{\mathrm{m}_{\mathcal{X}}{}'}$, augmented with $n - \mathrm{m}_{\mathcal{X}}{}'$ null columns inserted at the indices of the null columns of $\mathrm{R}_{\mathcal{X}}$.

Algorithm 3.1 Generation of a p-sparsified sketch

input: $\mathrm{m}_{\mathcal{X}}$, n and p

Generate a $\mathrm{m}_{\mathcal{X}} \times n$ matrix B whose entries are i.i.d. Bernouilli random variables of parameter p.

indices $\longleftarrow$ indices of non-null columns of B.

$B' \longleftarrow B$ where all null columns have been deleted.

Generate a matrix M_{SG} of the same size as B' whose entries are either i.i.d. Gaussian or Rademacher random variables.

$\mathrm{R}_{\mathcal{X}\mathrm{SG}} \longleftarrow M_{\mathrm{SG}} \circ B'$, where $\circ$ denotes the component-wise Hadamard matrix product.

return $\mathrm{R}_{\mathcal{X}\mathrm{SG}}$ *and* indices

on one side what is gained on the other: when p decreases (sparser sketches), the lower bound to get guarantees $\mathrm{m}_{\mathcal{X}} \gtrsim d_n/p^2$ increases, but the expected number of non-null columns $\mathrm{m}_{\mathcal{X}}{}'$ decreases, thus accelerating computations (note that for $p = 1$ we exactly recover the lower bound and number of non-null columns for Gaussian sketches).

Corollary 3.20. *Let* $\mathrm{R}_{\mathcal{X}} \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}} \times n}$ *be a* p*-sparsified sketching matrix, and* C_0, C_1 *and* $c(p)$ *as in theorem 3.19. Then, setting* $p \approx 0.7$ *and* $s = C_0 d_n/(0.7^2)$, $\mathrm{R}_{\mathcal{X}}$ *is* K_{X}*-satisfiable for* $c = 9$, *with a probability at least* $1 - C_1 e^{-\mathrm{m}_{\mathcal{X}} c(0.7)}$. *These values of* p *and* $\mathrm{m}_{\mathcal{X}}$ *minimize computations while maintaining the guarantees.*

Proof By substituting $\mathrm{m}_{\mathcal{X}} = C_0 d_n/p^2$ into $\mathbb{E}[\mathrm{m}_{\mathcal{X}}{}']$, one can show that it is optimal to set $p \approx 0.7$, independently from C_0 and d_n. ■

corollary 3.20 gives the optimal values of p and $\mathrm{m}_{\mathcal{X}}$ that ensure K_{X}-satisfiability of a p-sparsified sketching matrix while having some complexity reduction. However, the lower bound in Theorem 3.19 is a sufficient condition, that might be conservative. Looking at the problem of setting $\mathrm{m}_{\mathcal{X}}$ and p from the practitioner's point of view, we also provide more aggressive empirical guidelines. Indeed, although this regime is not covered by Theorem 3.19, experiments show that setting $\mathrm{m}_{\mathcal{X}}$ as for the Gaussian sketch and p smaller than $1/\mathrm{m}_{\mathcal{X}}$ yield very interesting results, see Figure 3.1c. Overall, p-sparsified sketches (i) generalize Gaussian sketches by introducing sparsity as a new degree of freedom, (ii) enjoy a regime in which theoretical guarantees are preserved and computations (slightly) accelerated, (iii) empirically yield competitive results also in aggressive regimes not covered by theory, thus achieving a wide range of interesting accuracy/computations tradeoffs.

Complexity Comparison. We first recall the time and space complexities for elementary matrix products. The main advantage of using Sub-Sampling matrices is that computing $\mathrm{R}_{\mathcal{X}}\mathrm{K}_{\mathrm{X}}$ is equivalent to sampling $\mathrm{m}_{\mathcal{X}}$ training inputs and construct a $\mathrm{m}_{\mathcal{X}} \times n$ Gram matrix, hence we gain huge time complexity since we do not compute a matrix multiplication, as well as space complexity since we do not compute a $n \times n$ Gram matrix. As a consequence, the main advantage of our p-sparsified sketches is their ability to be decomposed as $\mathrm{R}_{\mathcal{X}} = \mathrm{R}_{\mathcal{X}\mathrm{SG}}\mathrm{R}_{\mathcal{X}\mathrm{SS}}$, where $\mathrm{R}_{\mathcal{X}\mathrm{SG}} \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}} \times \mathrm{m}_{\mathcal{X}'}}$ is a sparse sub-Gaussian sketch and $\mathrm{R}_{\mathcal{X}\mathrm{SS}} \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}'} \times n}$ is a sub-Sampling sketch. This *decomposition trick* is particularly interesting when p is small, and since $\mathrm{m}_{\mathcal{X}}{}'$ follows a Binomial distribution of parameters n and $1 - \left(1-p\right)^{\mathrm{m}_{\mathcal{X}}}$ and we assume in our settings that n

is large, hence we have that $\mathrm{m}_{\mathcal{X}'} \approx \mathbb{E}\left[\mathrm{m}_{\mathcal{X}'}\right] \underset{p\to 0}{\sim} n\,\mathrm{m}_{\mathcal{X}}\,p$. In the following, we take $\mathrm{m}_{\mathcal{X}'} = n\,\mathrm{m}_{\mathcal{X}}\,p$. We recall that Accumulation matrices from Chen and Yang (2021a) write as $\mathrm{R}_{\mathcal{X}} = \sum_{i=1}^{m} \mathrm{R}_{\mathcal{X}(i)}$, where the $\mathrm{R}_{\mathcal{X}(i)}$s are sub-sampling matrices whose each row is multiplied by independent Rademacher variables. Hence, both p-sparsified and Accumulation sketches are interesting since they completely benefit from the computational efficiency of sub-sampling matrices. See table 3.2 for complexity analysis of matrix multiplications. Going into the complexity of the learning algorithms, the main difference between single and multiple output settings is the computation of feature maps, relying on the construction of $\mathrm{R}_{\mathcal{X}}\mathrm{K}_{\mathrm{X}}\mathrm{R}_{\mathcal{X}}{}^{\top}$ and the computation of the square root of its pseudo-inverse for the single output setting which is not present in the multiple output settings. We assume in our framework that d and even d^2 are typically very small in comparison with n. Hence, we have that the complexity in the single output case is dominated by the complexity of the operation $\mathrm{R}_{\mathcal{X}}\mathrm{K}_{\mathrm{X}}\mathrm{R}_{\mathcal{X}}{}^{\top}$, whereas in the multiple output case, it is dominated by the complexity of the operation $\mathrm{R}_{\mathcal{X}}\mathrm{K}_{\mathrm{X}}$. We see that from a time complexity perspective, p-sparsified sketches outperform Accumulation sketches in single output settings as long as $p \leq m/n\sqrt{\mathrm{m}_{\mathcal{X}}}$, and in multiple output settings as long as $p \leq m/n\,\mathrm{m}_{\mathcal{X}}$. From a space complexity perspective, Accumulation is always better as $n\,\mathrm{m}_{\mathcal{X}}\,p$ is typically greater than $\mathrm{m}_{\mathcal{X}}$, otherwise it shows poor performance. However, p is usually chosen such that $n\,\mathrm{m}_{\mathcal{X}}\,p$ is not very large compared with $\mathrm{m}_{\mathcal{X}}$.

Table 3.2: Complexity of matrix operations for each sketching type.

Sketching type	Complexity type	$\mathrm{R}_{\mathcal{X}}\mathrm{K}_{\mathrm{X}}$	$\mathrm{R}_{\mathcal{X}}\mathrm{K}_{\mathrm{X}}\mathrm{R}_{\mathcal{X}}{}^{\top}$
Gaussian	time	$\mathcal{O}\left(n^2\,\mathrm{m}_{\mathcal{X}}\right)$	$\mathcal{O}\left(n^2\,\mathrm{m}_{\mathcal{X}}\right)$
	space	$\mathcal{O}\left(n^2\right)$	$\mathcal{O}\left(n^2\right)$
p-sparsified	time	$\mathcal{O}\left(n^2\,\mathrm{m}_{\mathcal{X}}{}^2\,p\right)$	$\mathcal{O}\left(n^2\,\mathrm{m}_{\mathcal{X}}{}^3\,p^2\right)$
	space	$\mathcal{O}\left(n^2\,\mathrm{m}_{\mathcal{X}}\,p\right)$	$\mathcal{O}\left(n^2\,\mathrm{m}_{\mathcal{X}}{}^2\,p^2\right)$
Accumulation	time	$\mathcal{O}\left(n\,\mathrm{m}_{\mathcal{X}}\,m\right)$	$\mathcal{O}\left(\mathrm{m}_{\mathcal{X}}{}^2\,m^2\right)$
	space	$\mathcal{O}\left(n\,\mathrm{m}_{\mathcal{X}}\right)$	$\mathcal{O}\left(\mathrm{m}_{\mathcal{X}}{}^2\right)$
CountSketh	time	$\mathcal{O}\left(n^2\right)$	$\mathcal{O}\left(n^2\right)$
	space	$\mathcal{O}\left(n^2\right)$	$\mathcal{O}\left(n^2\right)$
Sub-Sampling	time	$\mathcal{O}\left(n\,\mathrm{m}_{\mathcal{X}}\right)$	$\mathcal{O}\left(\mathrm{m}_{\mathcal{X}}{}^2\right)$
	space	$\mathcal{O}\left(n\,\mathrm{m}_{\mathcal{X}}\right)$	$\mathcal{O}\left(\mathrm{m}_{\mathcal{X}}{}^2\right)$

Related works. Sparse sketches have been widely studied in the literature, see Clarkson and Woodruff (2017); Nelson and Nguyên (2013); Derezinski et al. (2021). However, these sketches are well-suited when applied to sparse matrices (e.g., matrices induced by graphs). In fact, given a matrix A, computing $\mathrm{R}_{\mathcal{X}}A$ with these types of sketching has a time complexity of the order of $\mathrm{nnz}\left(A\right)$, the number of nonzero elements of A. Besides, these sketches usually are constructed such that each column has at least one nonzero element (e.g. CountSketch, OSNAP), hence no *decomposition trick* is possible. Regarding kernel methods, since a Gram matrix is typically dense (e.g., with the Gaussian kernel, $\mathrm{nnz}\left(\mathrm{K}_{\mathrm{X}}\right) = n^2$), and since no decomposition trick can be applied, one has to compute the whole matrix K_{X} and store it, such that time and

space complexity implied by such sketches are of the order of n^2. In practice, we show that we can set p small enough to computationally outperform classical sparse sketches and still obtain similar statistical performance. Note that an important line of research is devoted to improving the statistical performance of Nyström's approximation, either by adaptive sampling (Kumar et al., 2012; Wang and Zhang, 2013; Gittens and Mahoney, 2013), or leverage scores (Alaoui and Mahoney, 2015; Musco and Musco, 2017; Rudi et al., 2018a; Chen and Yang, 2021b). We took the opposite route, as p-SG sketches are accelerated but statistically degraded versions of the Gaussian sketch.

3.4 Experiments

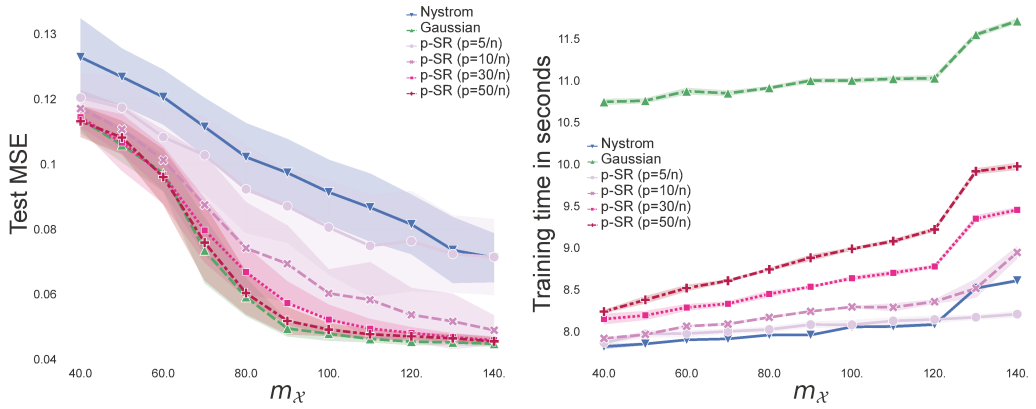
We now empirically compare the performance of p-sparsified sketches against state-of-he-art approaches, namely Nyström approximation (Williams and Seeger, 2001), Gaussian sketch (Yang et al., 2017), Accumulation sketch (Chen and Yang, 2021a), CountSketch (Clarkson and Woodruff, 2017) and Random Fourier Features (Rahimi and Recht, 2007). We chose not to benchmark ROS sketches as CountSketch has equivalent statistical accuracy while being faster to compute. Results reported are averaged over 30 replicates.

3.4.1 Scalar regression

Robust regression. We generate a dataset composed of $n = 10,000$ training datapoints: $9,900$ input points drawn i.i.d. from $\mathcal{U}\left(\left[0_{10}, \mathbb{1}_{10}\right]\right)$ and 100 other drawn i.i.d. from $\mathcal{N}\left(1.5\mathbb{1}_{10}, 0.25 I_{10}\right)$. The outputs are generated as $y = f^\star(x) + \epsilon$, where $\epsilon \sim \mathcal{N}\left(0,1\right)$ and

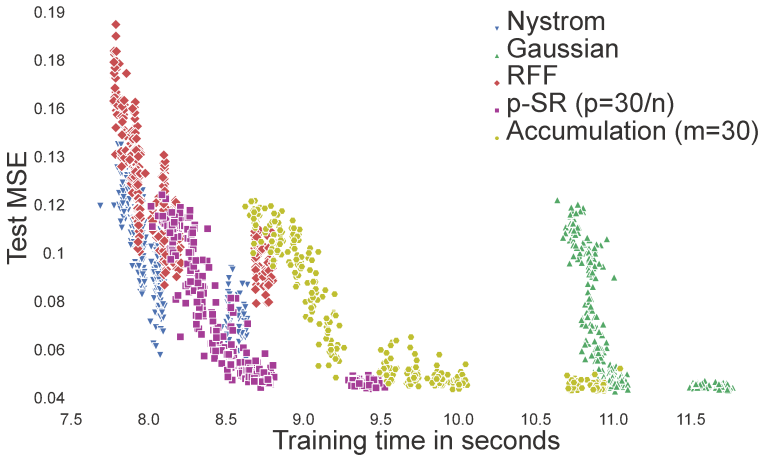
$$f^\star(x) = 0.1 e^{4x_1} + \frac{4}{1 + e^{-20\left(x_2 - 0.5\right)}} + 3x_3 + 2x_4 + x_5\,,$$

as introduced in Friedman (1991). We generate a test set of $n_{te} = 10,000$ points in the same way. We use the Gaussian kernel and select its bandwidth —as well as parameters λ_n and κ (and ϵ for ϵ-SVR)— via 5-folds cross-validation. We solve this 1D regression problem using the κ-Huber loss, described in Section 2.5.2. We learn the sketched kernel machines for different values of $\mathrm{m}_{\mathcal{X}}$ (from 40 to 140) and several values of p, the probability of being non-null in a p-SR sketch. Figure 3.1a presents the test error as a function of the sketch size $\mathrm{m}_{\mathcal{X}}$. Figure 3.1b shows the corresponding computational training time. All methods reduce their test error, measured in terms of the relative Mean Squared Error (MSE) when $\mathrm{m}_{\mathcal{X}}$ increases. Note that increasing p increases both the precision and the training time, as expected. This behaviour recalls the Accumulation sketches, since we observe a form of interpolation between the Nyström and Gaussian approximations. The behaviour of all the different sketched kernel machines is shown in Figure 3.1c, where each of them appears as a point (training time, test MSE). We observe that p-SR sketches attain the smallest possible error ($MSE \leq 0.05$) at the lowest training time budget (mostly around $5.6 < time < 6.6$). Moreover, p-SR sketches obtain a similar precision range as the Accumulation sketches, but for smaller training times (both approaches improve upon Gaussian sketch in that respect). Nyström sketching, which similarly to our approach does not need computing the entire Gram matrix, is fast to compute. The method is however known to be sensitive to the non-homogeneity of the marginal distribution



(a) Test relative MSE w.r.t. $m_{\mathcal{X}}$ with κ-Huber.

(b) Training time (seconds) w.r.t. $m_{\mathcal{X}}$ with κ-Huber.



(c) Test relative MSE w.r.t. training times with κ-Huber.

Figure 3.1: Trade-off between Accuracy and Efficiency for p-SR sketches with κ-Huber loss on synthetic dataset.

of the input data (Yang et al., 2017, Section 3.3). In contrast, the sub-Gaussian mixing matrix $R_{\mathcal{X}SG}$ in (3.16) makes p-sparsified sketches more robust, as empirically shown in Figure 3.1c. See Appendix A.3 for results on p-SG sketches.

3.4.2 Vector-valued regression

Table 3.3: Test pinball and crossing loss and training times (in seconds) with and without sketching ($m_{\mathcal{X}} = 50$).

Dataset	Metrics	w/o Sketch	$20/n_{tr}$-SR	$20/n_{tr}$-SG	Acc. $m = 20$	CountSketch
Boston	Pinball loss ↓	**51.28 ± 0.67**	54.75 ± 0.74	54.78 ± 0.72	54.73 ± 0.75	54.60 ± 0.72
	Crossing loss ↓	0.34 ± 0.13	0.26 ± 0.08	0.11 ± 0.07	0.15 ± 0.07	**0.10 ± 0.05**
	Training time ↓	6.97 ± 0.25	1.43 ± 0.07	1.38 ± 0.08	1.48 ± 0.05	**1.23 ± 0.07**
otoliths	Pinball loss ↓	2.78	2.66 ± 0.02	**2.64 ± 0.02**	2.67 ± 0.03	2.65 ± 0.02
	Crossing loss ↓	**5.18**	5.46 ± 0.06	5.43 ± 0.05	5.46 ± 0.06	5.44 ± 0.05
	Training time ↓	606.8	20.4 ± 0.5	**20.0 ± 0.3**	22.1 ± 0.4	20.9 ± 0.3

Joint quantile regression. We choose the quantile levels as follows $\tau = (0.1, 0.3, 0.5, 0.7, 0.9)$. We apply a subgradient algorithm to minimize the pinball loss described in Section 2.5.3 with ridge regularization and a kernel $\mathcal{K} = \mathrm{k}_{\mathcal{X}} M$ with M discussed in Example 3.14, and $\mathrm{k}_{\mathcal{X}}$ a Gaussian kernel. We select regularisation parameter λ_n and bandwidth of kernel σ^2 via a 5-fold cross-validation. We showcase the behaviour of the proposed algorithm for Joint Sketched Quantile Regression on two datasets: the Boston Housing dataset (Harrison Jr and Rubinfeld, 1978), composed of 506 data points devoted to house price prediction, and the Fish Otoliths dataset (Moen et al., 2018; Ordoñez et al., 2020), dedicated to fish age prediction from images of otoliths (calcium carbonate structures), composed of a train and test sets of size 3780 and 165 respectively. The results are averages over 10 random 70% – 30% train-test splits for Boston dataset. For the Otoliths dataset we kept the initial given train-test split. The results are reported in Table 3.3. Sketching allows for a massive reduction of the training times while preserving the statistical performances. As a comparison, according to the results of Sangnier et al. (2016), the best benchmark result for the Boston dataset in terms of test pinball loss is 47.4, while best test crossing loss is 0.48, which shows that our implementation does not compete in terms of quantile prediction but preserves the non-crossing property.

Table 3.4: ARRMSE and training times (in sec) with square loss and $\mathrm{m}_{\mathcal{X}} = 100$ when using Sketching.

Dataset	Metrics	w/o Sketch	$20/n_{tr}$-SR	$20/n_{tr}$-SG	Acc. $m = 20$	CountSketch
rf1	ARRMSE ↓	**0.575**	0.584 ± 0.003	0.583 ± 0.003	0.592 ± 0.001	$\mathbf{0.575 \pm 0.0005}$
	Training time ↓	1.73	$\mathbf{0.22 \pm 0.025}$	0.25 ± 0.005	0.60 ± 0.0004	0.66 ± 0.013
rf2	ARRMSE ↓	**0.578**	0.671 ± 0.009	0.656 ± 0.006	0.796 ± 0.006	0.715 ± 0.011
	Training time ↓	1.77	0.28 ± 0.003	$\mathbf{0.27 \pm 0.003}$	0.82 ± 0.003	0.62 ± 0.001
scm1d	ARRMSE ↓	**0.418**	0.422 ± 0.002	0.423 ± 0.001	0.423 ± 0.001	0.420 ± 0.001
	Training time ↓	9.36	$\mathbf{0.45 \pm 0.022}$	$\mathbf{0.45 \pm 0.019}$	0.86 ± 0.006	2.49 ± 0.035
scm20d	ARRMSE ↓	0.755	0.754 ± 0.003	0.754 ± 0.003	$\mathbf{0.753 \pm 0.001}$	0.754 ± 0.002
	Training time ↓	6.16	$\mathbf{0.38 \pm 0.016}$	$\mathbf{0.38 \pm 0.017}$	0.70 ± 0.032	1.91 ± 0.047

Multi-output regression. We finally conducted experiments on multi-output kernel ridge regression. We used decomposable kernels and took the largest datasets[2] introduced in Spyromitros-Xioufis et al. (2016). They consist of four datasets, divided into two groups: River Flow (rf1 and rf2) both composed of 4108 training data, and Supply Chain Management (scm1d and scm20d) composed of 8145 and 7463 training data respectively (more details and additional results can be found in Appendix A.3). We compare our non-sketched decomposable matrix-valued kernel machine with the sketched version. For the sake of conciseness, we only report here the Average Relative Root Mean Squared Error (ARRMSE), see Table 3.4 and Appendix A.3. For all datasets, sketching shows strong computational improvements while maintaining the accuracy of non-sketched approaches.

Note that for both joint quantile regression and multi-output regression the results obtained after sketching (no matter the sketch chosen) are almost the same as those attained without sketching. It might be explained by two factors. First, the datasets studied have relatively small training sizes (from 354 training data for Boston to 8145 for scm1d). Second, predicting jointly multiple outputs is a complex task, so it appears

[2] available at http://mulan.sourceforge.net/datasets-mtr.html.

more natural to obtain fewer differences and variances using various types of sketches (or no sketch). However, in all cases sketching induces a huge time saver.

3.5 Conclusion

We proposed excess-risk bounds for sketched kernel machines in the context of Lipschitz losses, with results valid for both scalar and matrix-valued kernels. We introduced a novel sketching scheme that leverages the good empirical statistical guarantees of Gaussian Sketching while combining them with the low cost of Nyström sketching. Numerical experiments show that this novel scheme opens the door to many applications beyond the squared loss. Improvements in multi-output regression can certainly be obtained by applying low-rank considerations in the output space as well.

4

Sketch In, Sketch Out: Accelerating both Learning and Inference for Structured Prediction with Kernels

Contents

4.1 Introduction

Equipped with p-sparsified sketches and the knowledge on sketching the input kernel, we tackle in this chapter the scalability of surrogate kernel methods, and in particular Input Output Kernel Regression. Leveraging the kernel trick in both the input and output spaces, surrogate kernel methods are a flexible and theoretically grounded solution to structured output prediction. If they provide state-of-the-art performance on complex data sets of moderate size (e.g., in chemoinformatics), these approaches however fail to scale. We propose to equip surrogate kernel methods with sketching-based approximations, applied to both the input and output feature maps. We prove excess risk bounds on the original structured prediction problem, showing how to attain close-to-optimal rates with a reduced sketch size that depends on the eigendecay of the input/output covariance operators. From a computational perspective, we show that the two approximations have distinct but complementary impacts: sketching the input kernel mostly reduces training time, while sketching the output kernel decreases the inference time. Empirically, our approach is shown to scale, achieving state-of-the-art performance on benchmark data sets where non-sketched methods are intractable. Motivated by surrogate structured prediction, we make the following contributions:

- We apply sketching to the vector-valued kernel regression problem solved in structured prediction, both on inputs and outputs, which accelerates respectively learning and inference.
- We derive excess risk bounds controlled by the properties of the sketched projection operators.
- We prove that sub-Gaussian sketches provide close-to-optimal rates with small sketch sizes.

- We empirically show that our algorithms maintain good accuracy on moderate-size data sets while enabling kernel surrogate methods on large datasets where the standard approach is simply intractable.
- A Python implementation of our approach is publicly available on GitHub.

Notations. We introduce now generic notations for the input (output) space and kernel. If $\mathcal{Z}$ denotes a generic Polish space, $k_{\mathcal{Z}}$ is a positive definite kernel over $\mathcal{Z}$ and $\psi_{\mathcal{Z}}(z) := k_{\mathcal{Z}}(\cdot, z)$ is the canonical feature map of $k_{\mathcal{Z}}$. $\mathcal{H}_{\mathcal{Z}}$ denotes the Reproducing Kernel Hilbert Space (RKHS) associated to $k_{\mathcal{Z}}$. $S_Z : f \in \mathcal{H}_{\mathcal{Z}} \mapsto (1/\sqrt{n})(f(z_1), \ldots, f(z_n))^\top$ is the sampling operator over $\mathcal{H}_{\mathcal{Z}}$ (Smale and Zhou, 2007). For any operator A, we denote $A^\#$ its adjoint. The adjoint of S_Z is defined as ${S_Z}^\# : \alpha \in \mathbb{R}^n \mapsto (1/\sqrt{n}) \sum_{i=1}^n \alpha_i \psi_{\mathcal{Z}}(z_i)$. If z is a r.v. distributed according to $\rho_{\mathcal{Z}}$, its covariance operator over $\mathcal{H}_{\mathcal{Z}}$ is $C_{\mathcal{Z}} = \mathbb{E}_z[\psi_{\mathcal{Z}}(z) \otimes \psi_{\mathcal{Z}}(z)]$, and its empirical counterpart $\widehat{C}_Z = (1/n) \sum_{i=1}^n \psi_{\mathcal{Z}}(z_i) \otimes \psi_{\mathcal{Z}}(z_i) = {S_Z}^\# S_Z$, where $\{(z_i)_{i=1}^n\}$ is i.i.d. drawn from $\rho_{\mathcal{Z}}$. The Moore-Penrose inverse of M is denoted $M^\dagger$.

4.2 Background

We now recall the structured prediction setting based on a kernel-induced loss, and a state-of-the-art surrogate approach to solve it. We also provide reminders about sketching as a way to scale kernel methods up.

Structured prediction with surrogate kernel methods. Let $\mathcal{X}$ be the input space and $\mathcal{Y}$ a structured output space. In general, $\mathcal{Y}$ is finite and extremely large. Define a positive definite kernel $k_{\mathcal{Y}} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, that measures how close two objects from $\mathcal{Y}$ are. We consider the loss function induced by $k_{\mathcal{Y}}$, defined as $\ell : (y, y') \to \|\psi_{\mathcal{Y}}(y) - \psi_{\mathcal{Y}}(y')\|^2_{\mathcal{H}_{\mathcal{Y}}}$. Note that it can be computed using the kernel trick. Given an unknown joint probability distribution ρ defined on $\mathcal{X} \times \mathcal{Y}$, the goal of structured prediction is to approximate

$$f^* = \operatorname*{arg\,min}_{f : \mathcal{X} \to \mathcal{Y}} \mathcal{R}(f), \tag{4.1}$$

where $\mathcal{R}(f) = \mathbb{E}_{(x,y)\sim\rho}\left[\|\psi_{\mathcal{Y}}(y) - \psi_{\mathcal{Y}}(f(x))\|^2_{\mathcal{H}_{\mathcal{Y}}}\right]$, using only an i.i.d. sample $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ drawn from ρ. Estimating directly f^* is not tractable, such that many works (Cortes et al., 2005; Geurts et al., 2006; Brouard et al., 2011; Ciliberto et al., 2016) have proposed instead the following two-step approach:

1. Surrogate Regression: Find an estimator $\hat{h}$ of the surrogate target $h^* : x \mapsto \mathbb{E}_y[\psi_{\mathcal{Y}}(y) | x]$ such that

$$h^* = \operatorname*{arg\,min}_{h} \; \mathbb{E}_{(x,y)} \left[\left\| h(x) - \psi_{\mathcal{Y}}\left(y\right) \right\|^2_{\mathcal{H}_{\mathcal{Y}}} \right].$$

2. Pre-image: Define $\hat{f}$ by decoding $\hat{h}$, i.e.,

$$\hat{f}(x) = d(\hat{h}(x)) := \operatorname*{arg\,min}_{y \in \mathcal{Y}} \left\| \hat{h}(x) - \psi_{\mathcal{Y}}(y) \right\|^2_{\mathcal{H}_{\mathcal{Y}}}.$$

The surrogate regression in Step 1 is much easier to handle than the initial structured prediction problem: it avoids learning f through the composition with the implicit feature map $\psi_{\mathcal{Y}}$, and relegates the difficulty of handling structured objects to Step 2,

i.e. at inference. In addition, vector-valued regression into infinite-dimensional spaces is a well-studied problem, that can be solved by using the kernel trick in the output space. This two-step approach belongs to the general framework of SELF (Ciliberto et al., 2016) and ILE (Ciliberto et al., 2020) and enjoys valuable theoretical guarantees. It is Fisher consistent, i.e., h^* yields f^* after decoding, and the excess risk of $\hat{f}$ is controlled by that of $\hat{h}$.

Input Output ridge Kernel Regression. A common choice to tackle in practice the surrogate regression problem consists in solving a *kernel ridge regression problem*, leveraging kernels in both input and output spaces. The hypothesis space is chosen as a vector-valued Reproducing Kernel Hilbert Space (vv-RKHS) (Senkene and Tempel'man, 1973; Micchelli and Pontil, 2005; Carmeli et al., 2006, 2010). In a nutshell, if $\mathcal{F}$ denotes a Hilbert space, a mapping $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{F})$, where $\mathcal{L}(\mathcal{F})$ is the set of bounded linear operators on $\mathcal{F}$, is an operator-valued kernel (OVK) if it satisfies the following properties: $K\left(x,x'\right) = K\left(x',x\right)^{\#}$ for all $(x,x') \in \mathcal{X}^2$ (symmetry), and $\sum_{i,j=1}^{n} \left\langle \varphi_i, K\left(x_i,x_j\right)\varphi_j)\right\rangle_{\mathcal{F}} \geqslant 0$ for all $n \in \mathbb{N}$ and $\left(x_i,\varphi_i\right)_{i=1}^{n} \in (\mathcal{X}\times\mathcal{F})^n$ (positive-definiteness). Similarly to the scalar case, given an OVK K, one can define a unique Hilbert space $\mathcal{H}$ of functions from $\mathcal{X}$ to $\mathcal{F}$ that enjoys the reproducing kernel property, i.e., such that for all $x \in \mathcal{X}$, $\varphi \in \mathcal{F}$ and $f \in \mathcal{H}$ we have $x' \mapsto K\left(x,x'\right)\varphi \in \mathcal{F}$, and $\langle f, K\left(\cdot,x\right)\varphi\rangle_{\mathcal{H}} = \left\langle f(x),\varphi\right\rangle_{\mathcal{F}}$.

In what follows, we opt for the identity decomposable OVK $K : \mathcal{X}\times\mathcal{X} \to \mathcal{L}(\mathcal{H}_y)$, defined as: $K\left(x,x'\right) = \mathrm{k}_{\mathcal{X}}\left(x,x'\right) I_{\mathcal{H}_y}$, where $\mathrm{k}_{\mathcal{X}} : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ is a p.d. scalar-valued kernel on $\mathcal{X}$. In *Input Output Kernel Ridge Regression* (IOKR for short, Brouard et al. 2011; Kadri et al. 2013b; Brouard et al. 2016b; Ciliberto et al. 2020, also introduced as Kernel Dependency Estimation by Weston et al. (2003)), the estimator of the surrogate regression is obtained by solving the following Ridge regression problem within $\mathcal{H}$, given a regularisation penalty $\lambda > 0$,

$$\hat{h} = \operatorname*{arg\,min}_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \left\| \psi_{\mathcal{Y}}(y_i) - h(x_i)\right\|_{\mathcal{H}_y}^2 + \lambda\|h\|_{\mathcal{H}}^2. \tag{4.2}$$

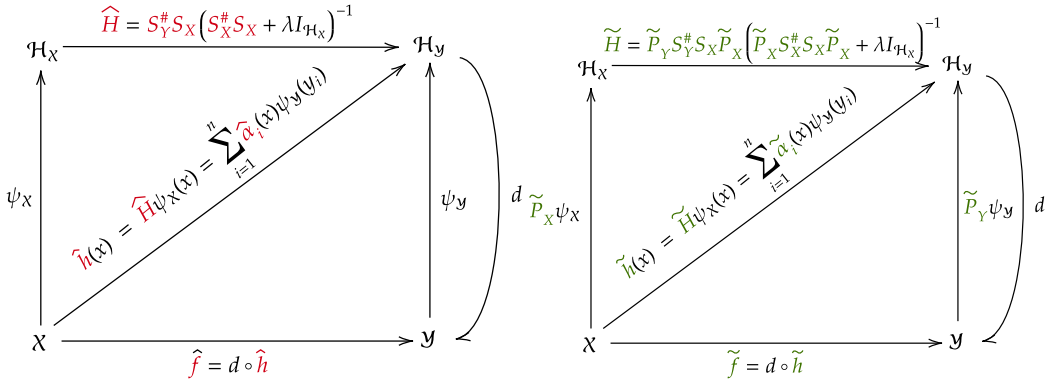
Interestingly, the unique solution to the above problem can be expressed in different ways. On one hand, we can derive from the representer theorem in vv-RKHSs (Micchelli and Pontil, 2005) the following expression:

$$\hat{h}(x) = \sum_{i=1}^{n} \hat{\alpha}_i(x)\,\psi_{\mathcal{Y}}(y_i), \tag{4.3}$$

with $\hat{\alpha}(x) = (\mathrm{K}_{\mathrm{X}} + n\lambda I_n)^{-1}\,\mathrm{k}_{\mathrm{X}}^{\mathrm{x}} := \widehat{\Omega}\,\mathrm{k}_{\mathrm{X}}^{\mathrm{x}}$, where $\mathrm{K}_{\mathrm{X}} = \left(\mathrm{k}_{\mathcal{X}}(x_i,x_j)\right)_{i,j=1}^{n}$ and $\mathrm{k}_{\mathrm{X}}^{\mathrm{x}} = \Big(\mathrm{k}_{\mathcal{X}}(x,x_1),\ldots,$ $\mathrm{k}_{\mathcal{X}}(x,x_n)\Big)$. On the other hand, using an operator view one obtains

$$\hat{h}(x) = \widehat{H}\,\psi_{\mathcal{X}}(x), \tag{4.4}$$

where $\widehat{H} = \mathrm{S}_{\mathrm{Y}}{}^{\#}\,\mathrm{S}_{\mathrm{X}}\left(\widehat{\mathrm{C}}_{\mathrm{X}} + \lambda I\right)^{-1}$. The latter expression can be seen as a re-writing of the first (Ciliberto et al., 2016), echoing the KDE equations with finite-dimensional feature maps (Cortes et al., 2005). It can also be related to the conditional kernel empirical mean embedding (Grünewälder et al., 2012).



(a) IOKR. (b) SISOKR.

Figure 4.1: IOKR (left) and SISOKR (right) in the KDE setting. Note that SISOKR consists in IOKR when kernels $k_{\mathcal{Z}}$ are replaced with their projected versions $\tilde{k}_{\mathcal{Z}}(\cdot,\cdot) = \langle \psi_{\mathcal{Z}}(\cdot), \widetilde{P}_Z \psi_{\mathcal{Z}}(\cdot) \rangle_{\mathcal{H}_{\mathcal{Z}}}$. However, this new output kernel changes the pre-image problem, and consequently the estimator $\tilde{f}$. In the paper, we modify $\widetilde{H}$ (and not the kernels) in order to use the comparison inequality from Ciliberto et al. (2020), see the proof of corollary 4.12.

The final estimator $\hat{f}$ is computed using the expression in (4.3), in order to benefit from the kernel trick:

$$\hat{f}(x) = \operatorname*{arg\,min}_{y \in \mathcal{Y}} \; k_{\mathcal{Y}}(y,y) - 2 k_X^{x\,T} \widehat{\Omega} k_Y^y, \tag{4.5}$$

where $k_Y^y = \left(k_{\mathcal{Y}}\left(y, y_1\right), \ldots, k_{\mathcal{Y}}\left(y, y_n\right)\right)^\top$. The training phase thus involves the inversion of a $n \times n$ matrix, whose cost without any approximation is $\mathcal{O}(n^3)$. Besides, it implies storing n^2 values in memory, which induces a heavy space complexity as well. In practice, decoding is performed by searching in a candidate set $\mathcal{Y}_c \subseteq \mathcal{Y}$ of size n_c. Hence, performing predictions on a test set X_{te} of size n_{te} mainly implies computing

$$\underbrace{K_X^{\text{te,tr}}}_{n_{\text{te}} \times n} \underbrace{\widehat{\Omega}}_{n \times n} \underbrace{K_Y^{\text{tr},c}}_{n \times n_c}, \tag{4.6}$$

where $K_X^{\text{te,tr}} = \left(k_{\mathcal{X}}(x_i^{\text{te}}, x_j)\right)_{1 \le i \le n_{\text{te}}, 1 \le j \le n} \in \mathbb{R}^{n_{\text{te}} \times n}$, and $K_Y^{\text{tr},c} = \left(k_{\mathcal{Y}}(y_i, y_j^c)\right)_{1 \le i \le n, 1 \le j \le n_c} \in \mathbb{R}^{n \times n_c}$. The complexity of the decoding part is $\mathcal{O}\left(n_{\text{te}} n n_c\right)$, considering $n_{\text{te}} < n \le n_c$. IOKR thus suffers from both heavy time and space computational costs. To cope with this limitation, we develop a general sketching approach that applies to both input and output feature spaces, accelerating both training and decoding.

Sketching for kernel methods. Applied to kernel methods to reduce their dependency in n, sketching can be seen as linear projections induced by a random matrix R (the sketching matrix) drawn from a probability distribution over $\mathbb{R}^{m \times n}$, where $m \ll n$. Classic examples include Nyström's approximation, where each row of R is randomly drawn from the rows of the identity matrix I_n, and Gaussian sketches, where all entries of R are i.i.d. Gaussian random variables. Nyström's approximation acts as a random training data sub-sampler, but it can be interpreted in many ways. In Drineas

et al. (2005); Bach (2013), it is shown to generate a low-rank approximation of the Gram matrix, while in Williams and Seeger (2001); Yang et al. (2012), it is seen as a way to construct data-dependent finite-dimensional random features. In Rudi et al. (2015), instead, it is presented as a projection onto a small subspace of the RKHS. For other sketching schemes such as Gaussian or Randomized Orthogonal Systems, most of the works adopt an optimization viewpoint, where a variable substitution is operated after the application of a Representer theorem (Yang et al., 2017; Lacotte and Pilanci, 2022). An interesting view provided in Kpotufe and Sriperumbudur (2020) explores the construction of random features based on Gaussian sketching. All these works are however limited to sketching the *input* kernel, in scalar regression problems. In this work: (1) we generalize input sketching to vector-valued problems, (2) we sketch the outputs, which is critical to scale-up surrogate methods with kernelized outputs.

4.3 Sketched Input Sketched Output Kernel Regression

The goal of this section is to construct a low-rank estimator of $\hat{h}$ by using sketching on both the input and output kernels. Note that sketching the feature maps is not desirable here: if we replace the output features $\psi_{\mathcal{Y}}(y_i) \in \mathcal{H}_{\mathcal{Y}}$ with some sketch-dependent approximations $\tilde{\psi}_{\mathcal{Y}}(y_i) \in \mathbb{R}^m$ we become unable to compare the resulting $\tilde{h}$ to the target h^*. Indeed, $\tilde{h}$ is an approximation of $x \mapsto \mathbb{E}_y[\tilde{\psi}_{\mathcal{Y}}(y)|x]$, which is a biased version of h^* due to the sketch realization. Instead, as we show below, seeing sketching as orthogonal projections provides a natural way to solve our problem. Ultimately, this gives rise to an estimator $\tilde{f}$ for structured prediction which is versatile, easy-to-implement, theoretically-based and scalable to large data sets.

Low-rank estimator. Given two orthogonal projection operators $\widetilde{\mathrm{P}}_{\mathrm{X}}$ and $\widetilde{\mathrm{P}}_{\mathrm{Y}}$, we start from (4.4) and replace the sampling operators on both sides, S_{X} and S_{Y}, by their projected counterparts, $\mathrm{S}_{\mathrm{X}}\widetilde{\mathrm{P}}_{\mathrm{X}}$ and $\mathrm{S}_{\mathrm{Y}}\widetilde{\mathrm{P}}_{\mathrm{Y}}$, so as to encode dimension reduction. The proposed low-rank estimator is expressed as follows:

$$\tilde{h}(x) = \widetilde{\mathrm{P}}_{\mathrm{Y}}\,\mathrm{S}_{\mathrm{Y}}{}^{\#}\,\mathrm{S}_{\mathrm{X}}\widetilde{\mathrm{P}}_{\mathrm{X}}\left(\widetilde{\mathrm{P}}_{\mathrm{X}}\widehat{\mathrm{C}}_{\mathrm{X}}\widetilde{\mathrm{P}}_{\mathrm{X}} + \lambda I_{\mathcal{H}_{\mathcal{X}}}\right)^{-1}\psi_{\mathcal{X}}(x). \tag{4.7}$$

We now show how to design the projection operators using sketching and then derive the novel expression of the low-rank estimator in terms of a weighted combination of the training outputs: $\tilde{h}(x) = \sum_{i=1}^{n} \tilde{\alpha}_i\, \psi_{\mathcal{Y}}(y_i)$, yielding a reduced computational cost. IOKR and SISOKR approaches are illustrated on Figure 4.1.

Sketching. In this work, we chose to leverage sketching to obtain random projectors within the input and output feature spaces. Indeed, sketching consists of approximating a feature map $\psi_{\mathcal{Z}} : \mathcal{Z} \to \mathcal{H}_{\mathcal{Z}}$ by projecting it thanks to a random projection operator $\widetilde{\mathrm{P}}_{\mathrm{Z}}$ defined as follows. Given a random matrix $\mathrm{R}_{\mathcal{Z}} \in \mathbb{R}^{\mathrm{m}_{\mathcal{Z}} \times n}$, n data $(z_i)_{i=1}^n \in \mathcal{Z}$ and $\mathrm{m}_{\mathcal{Z}} \ll n$, the linear subspace defining $\widetilde{\mathrm{P}}_{\mathrm{Z}}$ is constructed as the linear subspace generated by the span of the following $\mathrm{m}_{\mathcal{Z}}$ random vectors

$$\sum_{j=1}^{n} (\mathrm{R}_{\mathcal{Z}})_{ij}\, \psi_{\mathcal{Z}}(z_j) \in \mathcal{H}_{\mathcal{Z}}, \quad i = 1, \ldots, \mathrm{m}_{\mathcal{Z}}\,.$$

Let us dive into deeper details and give the expression of $\widetilde{\mathrm{P}}_{\mathrm{Z}}$. Let $\left\{\left(\sigma_i(\widetilde{\mathrm{K}}_{\mathrm{Z}}), \tilde{\mathbf{u}}_i^Z\right), i \in [\mathrm{m}_{\mathcal{Z}}]\right\}$ be the eigenpairs of $\widetilde{\mathrm{K}}_{\mathrm{Z}}$ ranked in descending order of eigenvalues, $\mathrm{p}_{\mathrm{Z}} = \mathrm{rank}\left(\widetilde{\mathrm{K}}_{\mathrm{Z}}\right)$, and for all $1 \leq i \leq \mathrm{p}_{\mathrm{Z}}$, $\tilde{e}_i^Z = \sqrt{\frac{n}{\sigma_i(\widetilde{\mathrm{K}}_{\mathrm{Z}})}} \mathrm{S}_{\mathrm{Z}}{}^{\#} \mathrm{R}_{\mathcal{Z}}{}^{\top} \tilde{\mathbf{u}}_i^Z$. The following result then holds, see Appendix B.2 for the proof.

Proposition 4.1 (Expression of the orthogonal projector)**.** *The $\tilde{e}_i^Z$s are the eigenfunctions, associated to the eigenvalues $\sigma_i(\widetilde{\mathrm{K}}_{\mathrm{Z}})/n$ of $\widetilde{\mathrm{C}}_{\mathrm{Z}}$. Furthermore, let $\widetilde{\mathcal{H}}_{\mathcal{Z}} = \mathrm{span}\left(\tilde{e}_1^z, \ldots, \tilde{e}_{\mathrm{p}_{\mathrm{Z}}}^z\right)$, the orthogonal projector $\widetilde{\mathrm{P}}_{\mathrm{Z}}$ onto $\widetilde{\mathcal{H}}_{\mathcal{Z}}$ writes as*

$$\widetilde{\mathrm{P}}_{\mathrm{Z}} = (\mathrm{R}_{\mathcal{Z}} \mathrm{S}_{\mathrm{Z}})^{\#} \left(\mathrm{R}_{\mathcal{Z}} \mathrm{S}_{\mathrm{Z}} (\mathrm{R}_{\mathcal{Z}} \mathrm{S}_{\mathrm{Z}})^{\#}\right)^{\dagger} \mathrm{R}_{\mathcal{Z}} \mathrm{S}_{\mathrm{Z}} \,. \tag{4.8}$$

Table 4.1: Time and space complexities at training and inference for the IOKR and SISOKR algorithms with sub-sampling, p-sparsified ($p \in (0,1]$) or Gaussian sketching, for a test set of size n_{te} and a candidate set of size n_c, such that $n_{te} \leq \mathrm{m}_{\mathcal{X}}, \mathrm{m}_{\mathcal{Y}} < n \leq n_c$. For the sake of simplicity, we omit the $\mathcal{O}(\cdot)$ in the following.

Method	Training		Inference	
	Time	Space	Time	Space
IOKR	n^3	n^2	$n_{te} n n_c$	$n n_c$
SISOKR (sub-sampling)	$\max(\mathrm{m}_{\mathcal{X}}, \mathrm{m}_{\mathcal{Y}}) n$	$\max(\mathrm{m}_{\mathcal{X}}, \mathrm{m}_{\mathcal{Y}}) n$	$n_{te} \mathrm{m}_{\mathcal{Y}} n_c$	$\mathrm{m}_{\mathcal{Y}} n_c$
SISOKR (p-sparsified)	$\max(\mathrm{m}_{\mathcal{X}}, \mathrm{m}_{\mathcal{Y}})^2 p n$	$\max(\mathrm{m}_{\mathcal{X}}, \mathrm{m}_{\mathcal{Y}}) p n$	$\max(n_{te}, n \mathrm{m}_{\mathcal{Y}} p) \mathrm{m}_{\mathcal{Y}} n_c$	$n p \mathrm{m}_{\mathcal{Y}} n_c$
SISOKR (Gaussian)	$\max(\mathrm{m}_{\mathcal{X}}, \mathrm{m}_{\mathcal{Y}}) n^2$	n^2	$n \mathrm{m}_{\mathcal{Y}} n_c$	$n n_c$

Sketched Input Sketched Output Kernel Regression (SISOKR). The SISOKR estimator is the low-rank estimator $\tilde{h}$, where both $\widetilde{\mathrm{P}}_{\mathrm{X}}$ and $\widetilde{\mathrm{P}}_{\mathrm{Y}}$ have been chosen as (4.8), for some random sketches $\mathrm{R}_{\mathcal{X}}$ and $\mathrm{R}_{\mathcal{Y}}$. It also admits the following expression based on a linear combination of the $\psi_{\mathcal{Y}}(y_i)$. The proof of the following proposition is given in Appendix B.2.

Proposition 4.2 (Expression of SISOKR)**.** $\forall\, x \in \mathcal{X}$,

$$\tilde{h}(x) = \sum_{i=1}^{n} \tilde{\alpha}_i(x)\, \psi_{\mathcal{Y}}\left(y_i\right), \tag{4.9}$$

where $\tilde{\alpha}(x) = \mathrm{R}_{\mathcal{Y}}{}^{\top} \widetilde{\Omega}\, \mathrm{R}_{\mathcal{X}}\, \mathrm{k}_{\mathrm{X}}^{\mathrm{x}}$ and

$$\widetilde{\Omega} = \widetilde{\mathrm{K}}_{\mathrm{Y}}{}^{\dagger} \mathrm{R}_{\mathcal{Y}} \mathrm{K}_{\mathrm{Y}} \mathrm{K}_{\mathrm{X}} \mathrm{R}_{\mathcal{X}}{}^{\top} (\mathrm{R}_{\mathcal{X}} \mathrm{K}_{\mathrm{X}}{}^2 \mathrm{R}_{\mathcal{X}}{}^{\top} + n\lambda \widetilde{\mathrm{K}}_{\mathrm{X}})^{\dagger}, \tag{4.10}$$

with $\widetilde{\mathrm{K}}_{\mathrm{X}} = \mathrm{R}_{\mathcal{X}} \mathrm{K}_{\mathrm{X}} \mathrm{R}_{\mathcal{X}}{}^{\top}$ and $\widetilde{\mathrm{K}}_{\mathrm{Y}} = \mathrm{R}_{\mathcal{Y}} \mathrm{K}_{\mathrm{Y}} \mathrm{R}_{\mathcal{Y}}{}^{\top}$.

Note that the matrix quantity that we recover above, $\mathrm{K}_{\mathrm{X}} \mathrm{R}_{\mathcal{X}}{}^{\top} \left(\mathrm{R}_{\mathcal{X}} \mathrm{K}_{\mathrm{X}}{}^2 \mathrm{R}_{\mathcal{X}}{}^{\top} + n\lambda \widetilde{\mathrm{K}}_{\mathrm{X}}\right)^{\dagger} \cdot \mathrm{R}_{\mathcal{X}}\, \mathrm{k}_{\mathrm{X}}^{\mathrm{x}}$, is typical to sketched kernel Ridge regression (Rudi et al., 2015; Yang et al., 2017). It allows the reduction of the size of the matrix to invert, which is now an $\mathrm{m}_{\mathcal{X}} \times \mathrm{m}_{\mathcal{X}}$ matrix. This is the main reason for the reduction of the learning step's complexity and is due to the input sketching. Nonetheless, we still need to perform matrix multiplication $\mathrm{R}_{\mathcal{X}} \mathrm{K}_{\mathrm{X}}$, whose efficiency depends on the sketch used). Note that output

sketching also requires additional operations, but the overall cost of computing $\tilde{\alpha}$ remains negligible compared to $\mathcal{O}(n^3)$, see "training time" column in table 4.1. As an example, with input/output Gaussian sketching which is the less efficient one, the time complexity is of order $\max(\mathrm{m}_\mathcal{X}, \mathrm{m}_\mathcal{Y})n^2$, where $\mathrm{m}_\mathcal{X}, \mathrm{m}_\mathcal{Y} \ll n$. We obtain the corresponding structured prediction estimator $\tilde{f}$ by decoding $\tilde{h}$, i.e., by replacing $\widehat{\Omega}$ by $\widetilde{\Omega}$ in (4.5). In fact, the main quantity we have to compute for prediction is now

$$\underbrace{\mathrm{K}_\mathrm{X}^{\mathrm{te,tr}} \mathrm{R}_\mathcal{X}^\top}_{n_\mathrm{te} \times \mathrm{m}_\mathcal{X}} \underbrace{\widetilde{\Omega}}_{\mathrm{m}_\mathcal{X} \times \mathrm{m}_\mathcal{Y}} \underbrace{\mathrm{R}_\mathcal{Y} \mathrm{K}_\mathrm{Y}^{\mathrm{tr},c}}_{\mathrm{m}_\mathcal{Y} \times n_c}. \tag{4.11}$$

The time complexity of this operation is $\mathcal{O}(n_\mathrm{te} m_\mathcal{Y} n_c)$ if $n_\mathrm{te} \leq \mathrm{m}_\mathcal{X}, \mathrm{m}_\mathcal{Y} < n \leq n_c$, which is a significant complexity reduction (the dependence in n vanishes), governed by the output sketch size $\mathrm{m}_\mathcal{Y}$, see table 4.1 for more details.

4.4 Theoretical Analysis

In this section, we present a statistical analysis of the proposed estimators $\tilde{h}$ and $\tilde{f}$. After introducing the assumptions on the learning task, we upper bound the excess risk of the sketched kernel ridge estimator, highlighting the approximation errors due to sketching. We then provide bounds for these approximation error terms. Finally, we study under which setting the proposed estimators $\tilde{h}$ and $\tilde{f}$ obtain substantial computational gains, while still benefiting from close-to-optimal learning rates. We consider the following set of common assumptions in the kernel literature (Bauer et al., 2007; Steinwart et al., 2009; Rudi et al., 2015; Pillaud-Vivien et al., 2018; Fischer and Steinwart, 2020; Ciliberto et al., 2020; Brogat-Motte et al., 2022b).

Assumption 4.3 (Attainability)**.** *We assume that $h^* \in \mathcal{H}$, i.e., that there is a linear operator $H : \mathcal{H}_\mathcal{X} \to \mathcal{H}_\mathcal{Y}$, with $\|H\|_\mathrm{HS} < +\infty$, s.t. $h^*(x) = H\,\psi_\mathcal{X}(x)$, $\forall\, x \in \mathcal{X}$.*

This is a standard assumption in the context of least-squares regression (Caponnetto and De Vito, 2007), making the target h^* belong to the hypothesis space. Note that relaxing this assumption is possible, although it would add a bias term that still requires some knowledge about h^* to be bounded. For instance, if h^* is supposed to be square-integrable, one usually chooses a RKHS associated with a universal operator-valued kernel, which is dense in the space of the square-integrable functions (Carmeli et al., 2010, Section 4). We now describe a set of generic assumptions that have to be satisfied by both input and output kernels $\mathrm{k}_\mathcal{X}$ and $\mathrm{k}_\mathcal{Y}$.

Assumption 4.4 (Bounded kernel)**.** *There exists $\kappa_\mathcal{Z} > 0$ such that $\mathrm{k}_\mathcal{Z}(z,z) \leq {\kappa_\mathcal{Z}}^2$, $\forall\, z \in \mathcal{Z}$. We note $\kappa_\mathcal{X}, \kappa_\mathcal{Y} > 0$ for the input and output kernels $\mathrm{k}_\mathcal{X}$ and $\mathrm{k}_\mathcal{Y}$ respectively.*

Assumption 4.5 (Capacity condition)**.** *There exists $\gamma_\mathcal{Z} \in [0,1]$ such that $\mathrm{Q}_\mathcal{Z} := \mathrm{Tr}({\mathrm{C}_\mathcal{Z}}^{\gamma_\mathcal{Z}}) < +\infty$.*

Note that Assumption 4.5 is always verified for $\gamma_\mathcal{Z} = 1$, as $\mathrm{Tr}(\mathrm{C}_\mathcal{Z}) = \mathbb{E}[\|\psi_\mathcal{Z}(z)\|^2_{\mathcal{H}_\mathcal{Z}}] < +\infty$ from Assumption 4.4, and that the smaller $\gamma_\mathcal{Z}$ the faster the eigendecay of $\mathrm{C}_\mathcal{Z}$, with $\gamma_\mathcal{Z} = 0$ when $\mathrm{C}_\mathcal{Z}$ is of finite rank. More generally, this assumption is for instance verified for a Sobolev kernel and a marginal distribution whose density is upper-bounded (Ciliberto et al., 2020, Assumption 2).

Assumption 4.6 (Embedding property)**.** *There exist* $\mathrm{b}_{\mathcal{Z}} > 0$ *and* $\mu_{\mathcal{Z}} \in [0,1]$ *such that* $\psi_{\mathcal{Z}}(z) \otimes \psi_{\mathcal{Z}}(z) \preceq \mathrm{b}_{\mathcal{Z}} \mathrm{C}_{\mathcal{Z}}{}^{1-\mu_{\mathcal{Z}}}$ *almost surely.*

Note that Assumption 4.6 is always verified for $\mu_{\mathcal{Z}} = 1$, as $\psi_{\mathcal{Z}}(z) \otimes \psi_{\mathcal{Z}}(z) \preceq \kappa_{\mathcal{Z}}{}^2 I_{\mathcal{H}_{\mathcal{Z}}}$ by Assumption 4.4, and that the smaller $\mu_{\mathcal{Z}}$, the stronger the assumption, with $\mu_{\mathcal{Z}} = 0$ when $\mathrm{C}_{\mathcal{Z}}$ is of finite. It allows to control the regularity of the functions in $\mathcal{H}_{\mathcal{Z}}$ with respect to the L^∞-norm, as it implies $\|h\|_{L_\infty} \leq \mathrm{b}_{\mathcal{Z}}{}^{1/2} \|h\|_{\mathcal{H}_{\mathcal{Z}}}^{\mu} \mathbb{E}[h(z)^2]^{(1-\mu)/2}$ (Pillaud-Vivien et al., 2018). For instance, an absolutely continuous distribution whose density is lower-bounded almost everywhere and a Matérn kernel verifies Assumption 4.6 (Pillaud-Vivien et al., 2018, Example 2).

SISOKR Excess-Risk. We can now provide a bound on the excess risk of SISOKR.

Theorem 4.7 (SISOKR excess risk bound)**.** *Let* $\delta \in (0,1]$, $n \in \mathbb{N}$ *such that* $\lambda = n^{-1/(1+\gamma_{\mathcal{X}})} \geq \frac{9\kappa_{\mathcal{X}}{}^2}{n} \log(\frac{n}{\delta})$. *Under Assumptions 4.3 to 4.6, with probability* $1-\delta$ *we have*

$$\mathbb{E}_x \left[\|\tilde{h}(x) - h^*(x)\|_{\mathcal{H}_{\mathcal{Y}}}^2 \right]^{\frac{1}{2}} \leq S(n,\delta) + c_2 A_{\rho_{\mathcal{X}}}^{\psi_{\mathcal{X}}}(\widetilde{P}_X) + A_{\rho_{\mathcal{Y}}}^{\psi_{\mathcal{Y}}}(\widetilde{P}_Y), \tag{4.12}$$

where $S(n,\delta) = c_1 \log(4/\delta) n^{-\frac{1}{2(1+\gamma_{\mathcal{X}})}}$ *and*

$$A_{\rho_{\mathcal{Z}}}^{\psi_{\mathcal{Z}}}(\widetilde{P}_Z) = \mathbb{E}_z \left[\|(\widetilde{\mathrm{P}}_Z - I_{\mathcal{H}_{\mathcal{Z}}}) \psi_{\mathcal{Z}}(z)\|_{\mathcal{H}_{\mathcal{Z}}}^2 \right]^{\frac{1}{2}}, \tag{4.13}$$

with $c_1, c_2 > 0$ *constants independent of* n *and* δ.

Proof [Proof sketch.] The proof relies on a decomposition of the operator $\widetilde{H}$ such that $\tilde{h}(x) = \widetilde{H}\psi_{\mathcal{X}}(x)$, see (271). The first term in (4.12) corresponds to the non-sketched kernel Ridge regression error, and the second term to the input sketching error. The latter extends both the results of Ciliberto et al. (2020) to sketched estimators and that of Rudi et al. (2015) to the vector vector-valued case. The third term, i.e., the output sketching error is specific to our framework and derives from the expression of h^* and Jensen's inequality. ■

The learning rate of the first term, i.e., the non-sketched kernel Ridge regression error, has been shown to be optimal under our set of assumptions in a minimax sense (Caponnetto and De Vito, 2007). The second and the third terms are approximation errors due to the sketching of the input and the output kernels, respectively. In particular, they write as *reconstruction errors* (Blanchard et al., 2007) associated to the random projection $\widetilde{\mathrm{P}}_{\mathrm{X}}$ and $\widetilde{\mathrm{P}}_{\mathrm{Y}}$ of the feature maps $\psi_{\mathcal{X}}$ and $\psi_{\mathcal{Y}}$ through the input and output marginal distributions.

Sketching Reconstruction Error. In theorem 4.9, we give bounds on the sketching reconstruction error for the family of sub-Gaussian sketches, enlarging the scope of sketching distributions whose reconstruction error's bound is known —it was previously limited to uniform and approximate leverage scores sub-sampling sketches

(Rudi et al., 2015). More generally, note that are admissible in our theoretical framework all sketching distributions for which concentration bounds on the induced empirical covariance operators can be derived since quantity $A_{\rho_{\mathcal{Z}}}^{\psi_{\mathcal{Z}}}(\widetilde{P}_Z)$ is then easily controlled. We now recall the definition of sub-Gaussian sketches and show how to bound their reconstruction error.

Definition 4.8. *A sub-Gaussian sketch* $\mathrm{R}_{\mathcal{Z}} \in \mathbb{R}^{\mathrm{m}_{\mathcal{Z}} \times n}$ *is composed of i.i.d. entries such that* $\mathbb{E}\left[\mathrm{R}_{\mathcal{Z}_{\mathrm{ij}}}\right] = 0$, $\mathbb{E}\left[{\mathrm{R}_{\mathcal{Z}_{\mathrm{ij}}}}^2\right] = 1/\mathrm{m}_{\mathcal{Z}}$ *and* $\mathrm{R}_{\mathcal{Z}_{\mathrm{ij}}}$ *is* $\frac{{\nu_{\mathcal{Z}}}^2}{\mathrm{m}_{\mathcal{Z}}}$*-sub-Gaussian, for all* $1 \leq i \leq \mathrm{m}_{\mathcal{Z}}$ *and* $1 \leq j \leq n$*, where* $\nu_{\mathcal{Z}} \geq 1$.

Recall that a standard normal r.v. is 1-sub-Gaussian. Moreover, by Hoeffding's lemma, any r.v. taking values in a bounded interval $[a, b]$ is $(b-a)^2/4$-sub-Gaussian. Hence, any sketch matrix composed of i.i.d. Gaussian or bounded r.v. is a sub-Gaussian sketch. Finally, note that p-sparsified sketches (El Ahmad et al., 2023) are sub-Gaussian with ${\nu_{\mathcal{Z}}}^2 = 1/p$, with $p \in]0, 1]$.

Theorem 4.9 (sub-Gaussian sketching reconstruction error)**.** *For* $\delta \in \left(0, 1/e\right]$, $n \in \mathbb{N}$ *sufficiently large such that* $\frac{9}{n}\log(n/\delta) \leq n^{-\frac{1}{1+\gamma_{\mathcal{Z}}}} \leq \|\mathrm{C}_{\mathcal{Z}}\|_{\mathrm{op}}/2$*, then if*

$$\mathrm{m}_{\mathcal{Z}} \geq c_4 \max\left(\nu_{\mathcal{Z}}^2 n^{\frac{\gamma_{\mathcal{Z}}+\mu_{\mathcal{Z}}}{1+\gamma_{\mathcal{Z}}}}, \nu_{\mathcal{Z}}^4 \log\left(1/\delta\right)\right), \tag{4.14}$$

with probability $1-\delta$ *we have*

$$\mathbb{E}_z\left[\|(\widetilde{\mathrm{P}}_Z - I_{\mathcal{H}_{\mathcal{Z}}})\,\psi_{\mathcal{Z}}(z)\|_{\mathcal{H}_{\mathcal{Z}}}^2\right] \leq c_3\, n^{-\frac{1-\gamma_{\mathcal{Z}}}{1+\gamma_{\mathcal{Z}}}}, \tag{4.15}$$

where $c_3, c_4 > 0$ *are constants independents of* $n, \mathrm{m}_{\mathcal{Z}}, \delta$.

Proof [Proof sketch] The proof essentially consists of bounding the difference between the empirical covariance operator and its sketched counterpart in operator norm, see (320). The latter rewrites as a sum of sub-Gaussian random variables in a separable Hilbert space, and we invoke Koltchinskii and Lounici (2017, Theorem 9). ■

Hence, depending on the regularity of the distribution (defined through our set of assumptions), one can obtain a small reconstruction error even with a small sketching size. For instance, if $\mu_{\mathcal{Z}} = \gamma_{\mathcal{Z}} = 1/3$, one obtains a reconstruction error of order $n^{-1/2}$ by using a sketching size of order $n^{1/2} \ll n$. As a limiting case, when $\mu_{\mathcal{Z}} = \gamma_{\mathcal{Z}} = 0$, one obtains a reconstruction error of order n^{-1} when using a constant sketching size.

Remark 4.10 (Comparison to Nyström's approximation)**.** *Note that the rate in theorem 4.9 is the same as that obtained with Nyström's approximation. However, our lower bound on the sketching size is slightly better. Recall that for uniform Nyström it is of order* $\max\left(n^{\frac{\gamma_{\mathcal{Z}}+\mu_{\mathcal{Z}}}{1+\gamma_{\mathcal{Z}}}}, 1\right)\left(\log(n) + \log\left(4\,{\kappa_{\mathcal{Z}}}^2/\delta\right)\right)$.

Remark 4.11 (Relaxation of Assumption 4.6)**.** *Assumption 4.6 allows to derive an upper bound of* $\mathcal{N}_{\mathcal{Z}}^{\infty}(t)$*, with* $t = n^{-\frac{1}{1+\gamma_{\mathcal{Z}}}}$*, that appears in the lower bound of the sketching size* $\mathrm{m}_{\mathcal{Z}}$*, see lemma .26 in Appendix B.6 and the proof of theorem 4.9 in Appendix B.4. However, we also have that* $\mathcal{N}_{\mathcal{Z}}^{\infty}(t) \leq t^{-1}$*, hence, if* $\mu_{\mathcal{Z}} + \gamma_{\mathcal{Z}} \geq 1 + \frac{\log(\mathrm{b}_{\mathcal{Z}}\mathrm{Q}_{\mathcal{Z}})(1+\gamma_{\mathcal{Z}})}{\log(n)}$*, we can relax*

Assumption 4.6 and rather obtain

$$\mathrm{m}_{\mathcal{Z}} \geq c_4 \max\left(\nu_{\mathcal{Z}}^2 n^{\frac{1}{1+\gamma_{\mathcal{Z}}}}, \nu_{\mathcal{Z}}^4 \log\left(1/\delta\right)\right), \tag{4.16}$$

as a lower bound.

Learning rates for SISOKR with sub-Gaussian sketches. For the sake of presentation, we use $\lesssim$ to keep only the dependencies in $n, \delta, \nu, \gamma, \mu$. We note $a \vee b \coloneqq \max(a, b)$.

Corollary 4.12 (SISOKR learning rates)**.** *Consider the Assumptions of Theorems 4.7 and 4.9, that* $\|\psi_{\mathcal{Y}}(y)\|_{\mathcal{H}_{\mathcal{Y}}} = \kappa_{\mathcal{Y}}$ *for all* $y \in \mathcal{Y}$*, and* $n \in \mathbb{N}$ *such that* $\frac{9}{n}\log(n/\delta) \leq n^{-\frac{1}{1+\gamma_{\mathcal{Z}}}} \leq \|\mathrm{C}_{\mathcal{Z}}\|_{\mathrm{op}}/2$ *for* $\mathcal{Z} \in \{\mathcal{X}, \mathcal{Y}\}$*. Set*

$$\mathrm{m}_{\mathcal{Z}} \gtrsim \max\left(\nu_{\mathcal{Z}}^2 n^{\frac{\gamma_{\mathcal{Z}}+\mu_{\mathcal{Z}}}{1+\gamma_{\mathcal{Z}}}}, \nu_{\mathcal{Z}}^4 \log\left(1/\delta\right)\right) \tag{4.17}$$

for $\mathcal{Z} \in \{\mathcal{X}, \mathcal{Y}\}$*. Then with probability* $1-\delta$

$$\mathcal{R}(\tilde{f}) - \mathcal{R}(f^*) \lesssim \log\left(4/\delta\right) n^{-\frac{1-\gamma_{\mathcal{X}} \vee \gamma_{\mathcal{Y}}}{2(1+\gamma_{\mathcal{X}} \vee \gamma_{\mathcal{Y}})}}. \tag{4.18}$$

Proof Using Theorems 4.7 and 4.9 to bound $A_{\rho_{\mathcal{X}}}^{\psi_{\mathcal{X}}}(\widetilde{P}_X)$ and $A_{\rho_{\mathcal{Y}}}^{\psi_{\mathcal{Y}}}(\widetilde{P}_Y)$ gives that with probability $1-\delta$ it holds

$$\mathbb{E}_x\left[\|\tilde{h}(x) - h^*(x)\|_{\mathcal{H}_{\mathcal{Y}}}^2\right]^{\frac{1}{2}} \lesssim \log\left(4/\delta\right) n^{-\frac{1-\gamma_{\mathcal{X}} \vee \gamma_{\mathcal{Y}}}{2(1+\gamma_{\mathcal{X}} \vee \gamma_{\mathcal{Y}})}}. \tag{4.19}$$

We then apply the comparison inequality (Ciliberto et al., 2020) to the loss $\Delta(y, y') = \|\psi_{\mathcal{Y}}(y) - \psi_{\mathcal{Y}}(y')\|_{\mathcal{H}_{\mathcal{Y}}}^2$. ∎

This corollary shows that under strong enough regularity assumptions, the proposed estimators benefit from a close-to-optimal learning rate, even with small input and output sketching sizes. For instance, if $\mu_{\mathcal{X}} = \mu_{\mathcal{Y}} = \gamma_{\mathcal{X}} = \gamma_{\mathcal{Y}} = 1/3$, one obtains a learning rate of $\mathcal{O}(n^{-1/4})$, instead of the optimal rate of $\mathcal{O}(n^{-3/8})$ under the same assumptions, but only requiring sketching sizes $\mathrm{m}_{\mathcal{X}}, \mathrm{m}_{\mathcal{Y}}$ of order $n^{1/2} \ll n$. As a limiting case, when $\mu_{\mathcal{X}} = \mu_{\mathcal{Y}} = \gamma_{\mathcal{X}} = \gamma_{\mathcal{Y}} = 0$, one attains the optimal $\mathcal{O}(n^{-1/2})$ learning rate using constant sketching sizes.

Remark 4.13 (Other Sketches)**.** *Although we focused on sub-Gaussian sketches, any sketching distribution admitting concentration bounds for operators on separable Hilbert spaces allows bounding the quantity* $A_{\rho_{\mathcal{Z}}}^{\psi_{\mathcal{Z}}}(\widetilde{P}_Z)$ *and is then admissible for our theoretical framework. For instance, as shown in Rudi et al. (2015), uniform and approximate leverage scores subsampling schemes fit into the presented theory.*

Remark 4.14 (Application to Least Squares Regression)**.** *This model and theoretical framework applies to any least squares regression problem with identity separable input kernel and separable Hilbert output space* $\mathcal{Y}$*. It corresponds to having the linear output kernel* $\mathrm{k}_{\mathcal{Y}}(\cdot,\cdot) = \langle \cdot, \cdot \rangle_{\mathcal{Y}}$*, and then* $\psi_{\mathcal{Y}} = I_{\mathcal{Y}}$*.*

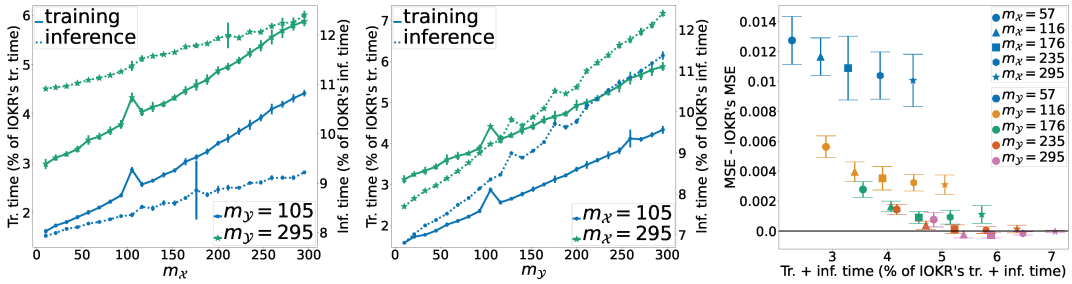


Figure 4.2: Variation of training and inference time w.r.t. $\mathrm{m}_{\mathcal{X}}$ and $\mathrm{m}_{\mathcal{Y}}$ (left and center), and trade-off performance against computational time (right) for SISOKR with $(2 \cdot 10^{-3})$-SR input/output sketches on synthetic data.

Remark 4.15 (Comparison to K_Z-satisfiability)**.** *Note that our framework significantly departs from that of* K_Z*-satisfiability (Yang et al., 2017; Chen and Yang, 2021a), a popular approach to analyzing sketching in kernel methods. First, we highlight that* K_Z*-satisfiability provides Gram matrix-specific bounds (through the critical radius), while ours are expressed in terms of quantities characteristics to the kernel. It then allows to upper bound the squared* $L^2(\mathbb{P}_n)$ *error between* $\tilde{h}$ *and* h^**, while our projection point of view provides a direct control on the excess risk. Finally, it is worth noting that our approach shows that sub-Gaussian sketches are admissible, which cannot be proven through* K_Z*-satisfiability.*

4.5 Experiments

In this section, we present experiments on synthetic and real-world data sets. SIOKR and ISOKR denote the models with sketching leveraged only on the inputs (resp. outputs). Results are averaged over 30 replicates, unless for the metabolite's experiments (5 replicates).

On the choice of the sketching types and its hyper-parameters. We focus on uniform sub-sampling (Rudi et al., 2015) and p-sparsified (p-SR/SG) (El Ahmad et al., 2023) sketches, which are covered by our theory. Sub-sampling is the most efficient approach computationally, but we empirically observe that p-SR/SG sketching is more accurate statistically. For SIOKR/ISOKR, we privilege accuracy and p-SR/SG sketching, as it is already providing substantial training/inference accelerations. Regarding SISOKR, we want the method to be the fastest both in training and inference. However, since output sketching adds training computations, we compensate and use input sub-sampling to remain faster in training than SIOKR. Regarding the input/output sketching sizes $\mathrm{m}_{\mathcal{X}}$ and $\mathrm{m}_{\mathcal{Y}}$, the first way consists of leveraging the theoretical lower bounds derived for $\mathrm{m}_{\mathcal{X}}$ and $\mathrm{m}_{\mathcal{Y}}$, see Equation (4.14). Indeed, by computing the Singular Value Decomposition of the input/output Gram matrix, one may determine their eigendecay (i.e., γ_Z, μ_Z, ν_Z) and set $\mathrm{m}_{\mathcal{X}}$ and $\mathrm{m}_{\mathcal{Y}}$ accordingly. However, computing the SVD is very expensive, hence one can rather compute the approximate leverage scores as in Alaoui and Mahoney (2015) for instance. In the following, we instead adopt an empirical routine. Given training and/or inference time budgets (corresponding e.g., to IOKR's training/inference times or the hardware limitations),

Table 4.2: F_1 scores on tag prediction from text data.

Method	Bibtex	Bookmarks	Mediamill
LR	37.2	30.7	NA
SPEN	42.2	34.4	NA
PRLR	44.2	34.9	NA
DVN	44.7	37.1	NA
SISOKR	44.1 ± 0.07	$\mathbf{39.3} \pm 0.61$	57.26 ± 0.04
ISOKR	44.8 ± 0.01	NA	58.02 ± 0.01
SIOKR	44.7 ± 0.09	39.1 ± 0.04	57.33 ± 0.04
IOKR	**44.9**	NA	**58.17**

we start from small $\mathrm{m}_{\mathcal{X}}$ and $\mathrm{m}_{\mathcal{Y}}$, which we progressively increase to maximize accuracy while respecting the budget. For the p-SR/SG sketches, we always set $p = 20/n$.

Synthetic Least Squares Regression. We generate a synthetic data set of least squares regression, with $n = 10\,000$ training data points, $\mathcal{X} = \mathcal{Y} = \mathbb{R}^d$, $d = 300$, and use input and output linear kernels, hence $\mathcal{H}_{\mathcal{X}} = \mathcal{H}_{\mathcal{Y}} = \mathbb{R}^d$. We construct covariance matrices $\mathrm{C}_{\mathcal{X}}$ and E by drawing randomly their eigenvectors such that their eigenvalues are $\sigma_i(\mathrm{C}_{\mathcal{X}}) = i^{-3/2}$ and $\sigma_i(E) = 0.2\, i^{-1/10}$. We draw $H_0 \in \mathbb{R}^{d \times d}$ with i.i.d. coefficients from the standard normal distribution and set $H = \mathrm{C}_{\mathcal{X}} H_0$. For $i \leq n$, we generate inputs $x_i \sim \mathcal{N}(0, \mathrm{C}_{\mathcal{X}})$, noise $\epsilon_i \sim \mathcal{N}(0, E)$ and outputs $y_i = Hx_i + \epsilon_i$. We generate validation and test sets of $n_{\mathrm{val}} = n_{\mathrm{te}} = 1000$ points in the same way. Such choices for $\mathrm{C}_{\mathcal{X}}$ (with a polynomial eigenvalue decay), E (with very low eigenvalues and eigenvalue decay), and $H = \mathrm{C}_{\mathcal{X}} H_0$ enforce a high eigenvalue decay for $\mathrm{C}_{\mathcal{Y}}$ (since it will have a similar eigendecay as $\mathrm{C}_{\mathcal{X}}$) while being a favorable setting to deploy sketching, as the true regression function H is low rank. We select the regularisation penalty λ via 1-fold cross-validation. We learn the SISOKR model for different values of $\mathrm{m}_{\mathcal{X}}$ and $\mathrm{m}_{\mathcal{Y}}$ (from 10 to 295) and $(2 \cdot 10^{-3})$-SR input and output sketches. Note that for such a problem where $\mathcal{Y} = \mathcal{H}_{\mathcal{Y}}$, no decoding step is needed for inference. We still perform an artificial pre-image problem to illustrate the computational benefit of sketching during this phase.

Figure 4.2 (left and center) presents computational training (solid lines) and inference (dotted lines) time (as a percentage of IOKR's training/inference time) w.r.t. $\mathrm{m}_{\mathcal{X}}$ (resp. $\mathrm{m}_{\mathcal{Y}}$) for two values of $\mathrm{m}_{\mathcal{Y}}$ (resp. $\mathrm{m}_{\mathcal{X}}$). First, since $\mathrm{m}_{\mathcal{X}}, \mathrm{m}_{\mathcal{Y}} \leq 295 \ll n = 10\,000$, note that SISOKR's training and inference times are significantly smaller than IOKR's (between 2 and 6% of IOKR's training time and 8 and 12% IOKR's inference time). On Figure 4.2 (left) the slopes of the training time's lines are higher than the inference time's ones, while the opposite happens on Figure 4.2 (center). This confirms that training complexity is more sensitive to $\mathrm{m}_{\mathcal{X}}$, while inference complexity is governed by $\mathrm{m}_{\mathcal{Y}}$. Figure 4.2 (right) presents the difference with IOKR's test errors, in terms of Mean Squared Error (MSE), for some choices of $\mathrm{m}_{\mathcal{X}}$ and $\mathrm{m}_{\mathcal{Y}}$, as a function of the sum of the training and inference times. The MSE decreases as the sketch sizes increase and at a faster rate with respect to $\mathrm{m}_{\mathcal{X}}$. This might be due to the fact that we directly control the eigendecay of $\mathrm{C}_{\mathcal{X}}$, whereas $\mathrm{C}_{\mathcal{Y}} = \mathrm{C}_{\mathcal{X}} H_0 \mathrm{C}_{\mathcal{X}} H_0^{\top} \mathrm{C}_{\mathcal{X}} + E$, such that its range is not totally controlled by $\mathrm{C}_{\mathcal{X}}$. SISOKR obtains better MSE performance than IOKR for $\mathrm{m}_{\mathcal{X}} \geq 116$ and $\mathrm{m}_{\mathcal{Y}} = 295$, which is consistent with the results obtained when applying sketching to the input (resp. output) kernel only, see Appendix B.7.

Table 4.3: Training/inference times (in seconds).

Method	Bibtex	Bookmarks	Mediamill
SISOKR	**1.41 ± 0.03 / 0.46 ± 0.01**	**118 ± 1.5 / 20 ± 0.2**	**66 ± 0.1 / 4 ± 0.01**
ISOKR	2.51 ± 0.06 / 0.58 ± 0.01	NA	636 ± 3.7 9 ± 0.2
SIOKR	1.99 ± 0.07 / 1.22 ± 0.03	354 ± 2.1 / 297 ± 2.1	199 ± 0.1 / 121 ± 0.02
IOKR	2.54 / 1.18	NA	621 / 204

Multi-Label Classification. We compare our models to state-of-the-art multi-label and structured prediction methods, namely IOKR (Brouard et al., 2016b), logistic regression (LR) trained independently for each label (Lin et al., 2014), the multi-label approach Posterior-Regularized Low-Rank (PRLR) (Lin et al., 2014), the energy-based model Structured Prediction Energy Networks (SPEN) (Belanger and McCallum, 2016) and Deep Value Networks (DVN) (Gygli et al., 2017). Results are taken from the cited articles. Data sets Bibtex and Bookmarks are tag recommendation problems, in which the objective is to propose a relevant set of tags (e.g., url, description, journal volume) to users when they add a new Bookmark or Bibtex entry to the social bookmarking system Bibsonomy. The MediaMill Challenge (Snoek et al., 2006) is a multi-label classification problem, where the goal is to detect the presence of semantic concepts in a video. They contain respectively $n = 4880$, $n = 60\,000$ and $n = 30\,993$ training points, see Appendix B.7 for details. We use the train-test splits available at https://mulan.sourceforge.net/datasets-mlc.html.

For all multi-label experiments, we use Gaussian input and output kernels with widths σ^2_{in} and σ^2_{out}. We use p-SG input (resp. output) sketches for SIOKR (resp. ISOKR), uniform sub-sampling input sketches, and p-SG output sketches for SISOKR. For Bibtex experiments, we choose $\mathrm{m}_{\mathcal{X}} = 2250$ and $\mathrm{m}_{\mathcal{Y}} = 200$, for Bookmarks experiments, $\mathrm{m}_{\mathcal{X}} = 13\,000$ and $\mathrm{m}_{\mathcal{Y}} = 750$, and for Mediamill experiments, $\mathrm{m}_{\mathcal{X}} = 8\,000$ and $\mathrm{m}_{\mathcal{Y}} = 500$. All the training data are used as candidate sets. The performance is measured by example-based F1 score, and hyper-parameters are selected on logarithmic grids by 5-fold cross-validation. The results in table 4.2 show that surrogate methods (last four columns) compete with SOTA methods, including deep-learning-based methods such as SPEN or DVN. On Bibtex, sketched models preserve good performance compared to IOKR (which performs best) while being faster to train (SIOKR and SISOKR) and significantly faster for inference (ISOKR and SISOKR), see table 4.3. Since the Bookmarks data set is too large, storing the whole n^2-Gram matrix K_{X} exceeds CPU's space limitations, which highlights the necessity of efficient sketching approximations such that sub-sampling or p-SR/SG sketches for kernel methods. Hence, we can only test SIOKR and SISOKR models on this data set, which outperforms other methods. SISOKR's inference phase is notably faster than SIOKR's (20 seconds vs. 5 minutes). Similarly, on the Mediamill problem, our approximated approaches are shown to be significantly faster to run while suffering a minimal reduction in F_1 score. Note that, with the same sketch matrix $\mathrm{R}_{\mathcal{X}}$, SIOKR's training is faster than SISOKR's as there is no additional computation on Gram matrix K_{Y}. In table 4.3, SISOKR is faster to train as it uses a more efficient input sketching (sub-sampling vs. p-SG).

Metabolite Identification. Metabolite identification consists of predicting small molecules, called metabolites, from their tandem mass spectrum. The metabolite structure is represented as a binary vector of length $d = 7593$, called a fingerprint. Each entry of the fingerprint encodes the presence or absence of a molecular property.

Table 4.4: Standard errors for the metabolite identification problem and computation times (in seconds).

Method	kernel loss ↓	Top-1 | 5 | 10 accuracies ↑	training ↓	inference ↓
SPEN	0.537 ± 0.008	25.9% | 54.1% | 64.3%	NA	NA
SISOKR	0.566 ± 0.007	25.1% | 54.2% | 64.7%	4.05 ± 0.05	$\mathbf{1112 \pm 29}$
ISOKR	0.509 ± 0.009	28.0% | 58.9% | 68.9%	6.25 ± 50.31	1133 ± 32
SIOKR	0.492 ± 0.008	29.5% | 61.3% | 70.9%	$\mathbf{1.25 \pm 0.02}$	1179 ± 37
IOKR	$\mathbf{0.486 \pm 0.008}$	**29.6**% | **61.6**% | **71.4**%	3.54 ± 0.15	1191 ± 38

IOKR is the SOTA method for this problem (Brouard et al., 2016a). The data set consists of $n = 6974$ training labeled mass spectra, the median size of the candidate sets is 292 and the largest candidate set contains 36 918 fingerprints. This metabolite identification problem thus involves high-dimensional complex outputs, for which the choice of the output kernel is crucial, and a large number of candidates, making the inference step long.

Our experimental protocol is similar to that of Brouard et al. (2016a) (5-CV Outer / 4-CV Inner loops). We use probability product input kernel for mass spectra and Gaussian-Tanimoto output kernel (Ralaivola et al., 2005) – with width σ^2 – for the molecular fingerprints. We select hyper-parameters λ and σ^2 in logarithmic grids based on MSE in $\mathcal{H}_\mathcal{Y}$ (hence no decoding is needed during selection). For the sketched models, we use p-SR input (resp. output) sketches for SIOKR (resp. ISOKR), and uniform sub-sampling input sketches and p-SR output sketches for SISOKR, with $\mathrm{m}_\mathcal{X} = 1500$, and $\mathrm{m}_\mathcal{Y} = 800$.

We compare our sketched models with IOKR and SPEN, see table 4.4. Results for SPEN are taken from Brogat-Motte et al. (2022b). SIOKR obtains results similar to IOKR while being slightly faster in both the training and inference phases. ISOKR is slightly less accurate, but outperforms (S)IOKR in terms of inference time, while SISOKR has the fastest inference phase and still competes with SPEN statistically. We observe here that it is difficult to reduce significantly the inference time while keeping a good accuracy and to reduce both the training and inference time. This is due to the particular setting of the metabolite data set. Indeed, each molecule is associated with a specific candidate set, so when performing predictions one has to run through each element one by one to pick its candidate set. When performing predictions, one has to compute the matrix multiplication (4.11), which has a smaller complexity than (4.6), given that $\mathrm{R}_\mathcal{Y} \mathrm{K}_\mathrm{Y}^{\mathrm{tr},c}$ is already known. However, in the case of metabolite identification, one has to perform it for each test data, which takes most of the inference for both ISOKR and SISOKR models. As an example, for the 1133 (resp. 1112) seconds-long ISOKR's (resp. SISOKR) inference phase, computing $\mathrm{R}_\mathcal{Y} \mathrm{K}_\mathrm{Y}^{\mathrm{tr},c}$ takes 940 (resp. 917) seconds. Since we have access to all candidate sets for each molecule, one could pre-process these data beforehand and perform these matrix multiplications during training, leading to a high training time, but a very small inference time, which could be of interest according to the practitioner's wish. When candidate sets are known and fixed (e.g., in multi-label prediction), sketching the output kernel is of particular interest as no additional operation is needed for each prediction.

4.6 Conclusion

In this chapter, we scale up surrogate methods for structured prediction based on kernel Ridge regression by using random projections for both inputs and outputs. An interesting avenue for future work is the study of non-parametric estimators with kernelized outputs that do not benefit from the Ridge regression closed-form. The approach proposed in the next chapter continues on this path and allows to handle the losses $(y, y') \mapsto c(\|\psi_{\mathcal{Y}}(y) - \psi_{\mathcal{Y}}(y')\|^2_{\mathcal{H}_y})$, where $c : \mathbb{R} \to \mathbb{R}$ is a differentiable or sub-differentiable function, such as the robust losses introduced in section 2.5 and tackled in Laforgue et al. (2020).

5

Deep Sketched Output Kernel Regression for Structured Prediction

Contents

5.1 Introduction

In this chapter, we show how to introduce kernel-induced losses to deep neural networks thanks to the sketching applied to the output kernel, as previously explored. By leveraging the kernel trick in the output space, kernel-induced losses provide a principled way to define structured output prediction tasks for a wide variety of output modalities. In particular, they have been successfully used in the context of surrogate non-parametric regression, where the kernel trick is typically exploited in the input space as well. However, when inputs are images or texts, more expressive models such as deep neural networks seem more suited than non-parametric methods. We here tackle the question of how to train neural networks to solve structured output prediction tasks, while still benefiting from the versatility and relevance of kernel-induced losses. We design a novel family of deep neural architectures, whose last layer predicts in a data-dependent finite-dimensional subspace of the infinite-dimensional output feature space deriving from the kernel-induced loss. This subspace is chosen as the span of the eigenfunctions of a randomly-approximated version of the empirical kernel covariance operator. Interestingly, this approach unlocks the use of gradient descent algorithms (and consequently of any neural architecture) for structured prediction. Experiments on synthetic tasks as well as real-world supervised graph prediction problems show the relevance of our method.

In our proposition to solve structured prediction from complex input data, we make the following contributions:

- We introduce Deep Sketched Output Kernel Regression, a family of deep neural architectures whose last layer predicts a data-dependent finite-dimensional representation of the outputs, that lies in the infinite-dimensional feature space deriving from the kernel-induced loss.
- This last layer is computed beforehand, and is the eigenbasis of the sketched empirical covariance operator, unlocking the use of gradient-based techniques to learn the weights of the previous layers for any neural architecture.
- We empirically show the relevance of our approach on a synthetic least squares regression problem, and provide a strategy to select the sketching size.
- We show that DSOKR performs well on two text-to-molecule datasets.
- A Python implementation of our approach is publicly available on GitHub.

We emphasize again that this chapter is based on the following article to appear in *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2024 (★ indicates equal contribution): Tamim El Ahmad★, Junjie Yang★, Pierre Laforgue, and Florence d'Alché-Buc. Deep Sketched Output Kernel Regression for Structured Prediction.

5.2 Deep Sketched Output Kernel Regression

In this section, we set up the problem of structured prediction. Specifically, we consider surrogate regression approaches for kernel-induced losses. By introducing a last layer able to make predictions in a Reproducing Kernel Hilbert Space (RKHS), we unlock the use of deep neural networks as hypothesis space.

Consider the general regression task from an input domain $\mathcal{X}$ to a structured output domain $\mathcal{Y}$ (e.g., the set of labeled graphs of arbitrary size). Learning a mapping from $\mathcal{X}$ to $\mathcal{Y}$ naturally requires taking into account the structure of the output space. One way to do so is the *Output Kernel Regression* (OKR) framework (Weston et al., 2003; Cortes et al., 2005; Geurts et al., 2006; Brouard et al., 2011, 2016b), also known as surrogate regression methods.

Output Kernel Regression. A positive definite (p.d.) kernel $\mathrm{k}_{\mathcal{Y}} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a symmetric function such that for all $n \geq 1$, and any $\left(y_i\right)_{i=1}^n \in \mathcal{Y}^n$, $\left(\alpha_i\right)_{i=1}^n \in \mathbb{R}^n$, we have $\sum_{i,j=1}^n \alpha_i \, \mathrm{k}_{\mathcal{Y}}\left(y_i, y_j\right) \alpha_j \geq 0$. Such a kernel is associated with a canonical feature map $\psi_{\mathcal{Y}} : y \in \mathcal{Y} \mapsto \mathrm{k}_{\mathcal{Y}}(\cdot, y)$, which is uniquely associated with a Hilbert space of functions $\mathcal{H} \subset \mathbb{R}^{\mathcal{Y}}$, the RKHS, such that $\psi_{\mathcal{Y}}(y) \in \mathcal{H}_{\mathcal{Y}}$ for all $y \in \mathcal{Y}$, and $h\left(y\right) = \langle h, \psi_{\mathcal{Y}}(y) \rangle_{\mathcal{H}_{\mathcal{Y}}}$ for any $\left(h, y\right) \in \mathcal{H}_{\mathcal{Y}} \times \mathcal{Y}$. Given a p.d. kernel $\mathrm{k}_{\mathcal{Y}}$, $\psi_{\mathcal{Y}}$ its canonical feature map and $\mathcal{H}_{\mathcal{Y}}$ its RKHS, the OKR approach that we consider in this work exploits the kernel-induced squared loss:

$$\Delta(y, y') := \| \psi_{\mathcal{Y}}(y) - \psi_{\mathcal{Y}}(y') \|^2_{\mathcal{H}_{\mathcal{Y}}} = \mathrm{k}_{\mathcal{Y}}(y, y) - 2\,\mathrm{k}_{\mathcal{Y}}(y, y') + \mathrm{k}_{\mathcal{Y}}(y', y'). \tag{5.1}$$

The versatility of loss (5.1) stems from the large variety of kernels that have been designed to compare structured objects (Gärtner, 2008; Korba et al., 2018; Borgwardt et al., 2020). In multi-label classification, for instance, choosing the linear kernel

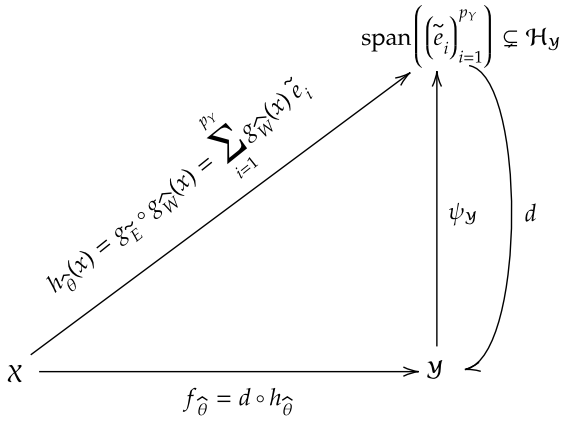


Figure 5.1: Illustration of DSOKR model.

or the Tanimoto kernel induces respectively the Hamming and the F1-loss (Tanimoto, 1958). In label ranking, Kemeny and Hamming embeddings define respectively Kendall's τ distance and the Hamming loss (Korba et al., 2018; Nowak et al., 2020). For sequence prediction tasks, n-gram kernels have been proven useful (Cortes et al., 2007; Kadri et al., 2013a; Nowak et al., 2020), while an abundant collection of kernels has been designed for graphs, based either on bags of structures or information propagation, see Appendix C.2 and Borgwardt et al. (2020) for examples.

If kernel-induced losses can be computed easily thanks to the kernel trick, note that most of them are however non-differentiable. In particular, this largely compromises their use within deep neural architectures, which are however key to achieving state-of-the-art performances in many applications. In this work, we close this gap and propose an approach that benefits from both the expressivity of neural networks for input image/textual data, as well as the relevance of kernel-induced losses for structured outputs. Formally, let ρ be a joint probability distribution on $\mathcal{X} \times \mathcal{Y}$. Our goal is to design a family $(f_\theta)_{\theta \in \Theta} \subset \mathcal{Y}^\mathcal{X}$ of neural networks with outputs in $\mathcal{Y}$ that can minimize the kernel-induced loss, i.e., that can solve

$$\min_{\theta \in \Theta} \mathbb{E}_{(x,y)\sim\rho} \left[\left\| \psi_\mathcal{Y}(y) - \psi_\mathcal{Y}\left(f_\theta(x)\right) \right\|_{\mathcal{H}_\mathcal{Y}}^2 \right]. \tag{5.2}$$

To do so, we assume that we can access a training sample $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ drawn i.i.d. from ρ. Since learning f_θ through $\psi_\mathcal{Y}$ is difficult, we employ a two-step method. First, we solve the surrogate empirical problem

$$\hat{\theta} \in \operatorname*{arg\,min}_{\theta \in \Theta} L(\theta) = \operatorname*{arg\,min}_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \| h_\theta(x_i) - \psi_\mathcal{Y}(y_i) \|_{\mathcal{H}_\mathcal{Y}}^2, \tag{5.3}$$

where $(h_\theta)_{\theta \in \Theta} \subset \mathcal{H}^\mathcal{X}$ is a family of neural networks with outputs in $\mathcal{H}$. We then retrieve the solution by solving for any prediction the pre-image problem

$$f_{\hat{\theta}}(x) = \operatorname*{arg\,min}_{y \in \mathcal{Y}} \| h_{\hat{\theta}}(x) - \psi_\mathcal{Y}(y) \|_{\mathcal{H}_\mathcal{Y}}^2. \tag{5.4}$$

This approach nonetheless raises a major challenge. Indeed, the dimension of the canonical feature space $\mathcal{H}_\mathcal{Y}$ may be infinite, making the training very difficult. The question we have to answer now is: *how can we design a neural architecture that is able to learn infinite-dimensional output kernel features?*

Neural networks with infinite-dimensional outputs. We propose a novel architecture of neural networks to compute the function h_θ with values in $\mathcal{H}_\mathcal{Y}$. Let $p \geq 1$, our architecture is the composition of two networks: an input neural network, denoted $g_W : \mathcal{X} \to \mathbb{R}^p$, with generic parameter W, and a last layer composed of a unique *functional* neuron, denoted $g_\mathrm{E} : \mathbb{R}^p \to \mathcal{H}_\mathcal{Y}$, that predicts in $\mathcal{H}_\mathcal{Y}$. The latter depends on the kernel $\mathrm{k}_\mathcal{Y}$ used in the loss definition, and on a finite basis $\mathrm{E} = ((e_j)_{j=1}^p) \in {\mathcal{H}_\mathcal{Y}}^p$ of elements in $\mathcal{H}_\mathcal{Y}$. We let $\theta = (W, \mathrm{E})$, and for any $x \in \mathcal{X}$, we have

$$h_\theta(x) \coloneqq g_\mathrm{E} \circ g_W(x), \tag{5.5}$$

where g_W typically implements a $L-1$ neural architecture encompassing, multilayered perceptrons, convolutional neural networks, or transformers. Instead, g_E computes a linear combination of some basis functions $\mathrm{E} = (e_j)_{j=1}^p \in {\mathcal{H}_\mathcal{Y}}^p$

$$g_\mathrm{E} : z \in \mathbb{R}^p \mapsto \sum_{j=1}^p z_j e_j \in \mathcal{H}_\mathcal{Y} \,. \tag{5.6}$$

With this architecture, computations remain finite, and the input neural network outputs the coefficients of the basis expansion, generating predictions in $\mathcal{H}_\mathcal{Y}$.

Remark 5.1 (Input Neural net's last layers)**.** *Since the neural network g_W learns the coordinates of the surrogate estimator in the basis, its last layers are always mere fully connected ones, regardless of the nature of the output data at hand.*

5.2.1 Learning neural networks with infinite-dimensional outputs

Learning the surrogate regression model h_θ now boils down to computing $\theta = (W, \mathrm{E})$. We propose to solve this problem in two steps. First, we learn a suitable E using only the output training data $(\psi_\mathcal{Y}(y_i))_{i=1}^n$ in an unsupervised fashion. Then, we use standard gradient-based algorithms to learn W through the frozen last layer, minimizing the loss on the whole supervised training sample $(x_i, \psi_\mathcal{Y}(y_i))_{i=1}^n$.

Estimating the functional last unit g_E. A very first idea is to choose E as the non-orthogonal dictionary $\psi_\mathcal{Y}(y_j)_{j=1}^n$. But this choice induces a very large output dimension (namely, $p = n$) for large training datasets.

An alternative consists in using Kernel Principal Component Analysis (Schölkopf et al., 1997). Given a marginal probability distribution over $\mathcal{Y}$, let $\mathrm{C}_\mathcal{Y} = \mathbb{E}_y[\psi_\mathcal{Y}(y) \otimes \psi_\mathcal{Y}(y)]$ be the covariance operator associated with $\mathrm{k}_\mathcal{Y}$, and $\widehat{\mathrm{C}}_\mathrm{Y} = (1/n)\sum_{i=1}^n \psi_\mathcal{Y}(y_i) \otimes \psi_\mathcal{Y}(y_i)$ its empirical counterpart. Let S_Y be the sampling operator that transforms a function $f \in \mathcal{H}_\mathcal{Y}$ into the vector $(1/\sqrt{n})(f(x_1), \ldots, f(x_n))^\top$ in $\mathbb{R}^n$, and denote by ${\mathrm{S}_\mathrm{Y}}^\#$ its adjoint. We have ${\mathrm{S}_\mathrm{Y}}^\# : \alpha \in \mathbb{R}^n \mapsto (1/\sqrt{n})\sum_{i=1}^n \alpha_i \psi_\mathcal{Y}(y_i) \in \mathcal{H}_\mathcal{Y}$, and $\widehat{\mathrm{C}}_\mathrm{Y} = {\mathrm{S}_\mathrm{Y}}^\# \mathrm{S}_\mathrm{Y}$. KPCA provides the eigenbasis of $\widehat{\mathrm{C}}_\mathrm{Y}$ by computing the SVD of the output Gram matrix, for a prohibitive computational cost of $\mathcal{O}(n^3)$. In practice, though, it is often the case that the so-called *capacity condition* holds (Ciliberto et al., 2020; El Ahmad et al., 2024), i.e., that the spectrum of the empirical covariance operator enjoys a large eigendecay. It is then possible to efficiently approximate the eigenbasis of $\widehat{\mathrm{C}}_\mathrm{Y}$ using random projections techniques (Mahoney et al., 2011; Woodruff, 2014), also known as sketching, solving this way the computational and memory issues.

Sketching for kernel methods. Sketching Mahoney et al. (2011); Woodruff (2014) is a dimension reduction technique based on random linear projections. Since the goal is to reduce the dependency on the number of training samples n in kernel methods, such linear projections can be encoded by a randomly drawn matrix $\mathrm{R}_{\mathcal{Y}} \in \mathbb{R}^{\mathrm{m}_{\mathcal{Y}} \times n}$, where $\mathrm{m}_{\mathcal{Y}} \ll n$. Standard examples include Nyström approximation (Meanti et al., 2020), where each row of $\mathrm{R}_{\mathcal{Y}}$ is randomly drawn from the rows of the identity matrix I_n, also called sub-sampling sketches, and Gaussian sketches (Yang et al., 2017), where all entries of $\mathrm{R}_{\mathcal{Y}}$ are i.i.d. Gaussian random variables. As they act as a random training data sub-sampler and then largely reduce both the time and space complexities induced by kernel methods, sub-sampling sketches are the most popular sketching type applied to kernels, while Gaussian sketches are less computationally efficient but offer better statistical properties. Hence, given a sketching matrix $\mathrm{R}_{\mathcal{Y}} \in \mathbb{R}^{\mathrm{m}_{\mathcal{Y}} \times n}$, one can defines $\tilde{\mathcal{H}}_{\mathcal{Y}} = \mathrm{span}((\sum_{j=1}^{n} \mathrm{R}_{\mathcal{Y}_{ij}} \psi_{\mathcal{Y}}(y_j))_{i=1}^{\mathrm{m}_{\mathcal{Y}}})$ which is a low-dimensional linear subspace of $\mathcal{H}_{\mathcal{Y}}$ of dimension at most $\mathrm{m}_{\mathcal{Y}}$. One can even compute the basis $\tilde{E}$ of $\tilde{\mathcal{H}}_{\mathcal{Y}}$, providing the last layer $g_{\tilde{E}}$.

Sketching to estimate $g_{\tilde{E}}$. We here show how to compute the basis $\tilde{E}$ of $\tilde{\mathcal{H}}_{\mathcal{Y}}$. Let $\mathrm{m}_{\mathcal{Y}} < n$, and $\mathrm{R}_{\mathcal{Y}} \in \mathbb{R}^{\mathrm{m}_{\mathcal{Y}} \times n}$ be a sketching matrix. Let $\widetilde{\mathrm{K}}_{\mathrm{Y}} = \mathrm{R}_{\mathcal{Y}} \mathrm{K}_{\mathrm{Y}} \mathrm{R}_{\mathcal{Y}}{}^{\top} \in \mathbb{R}^{\mathrm{m}_{\mathcal{Y}} \times \mathrm{m}_{\mathcal{Y}}}$ be the sketched Gram matrix, and $\left\{(\sigma_i(\widetilde{\mathrm{K}}_{\mathrm{Y}}), \tilde{\mathbf{u}}_i), i \in [\mathrm{m}_{\mathcal{Y}}]\right\}$ its eigenpairs, in descending order. We set $\mathrm{p}_{\mathrm{Y}} = \mathrm{rank}\left(\widetilde{\mathrm{K}}_{\mathrm{Y}}\right)$. Note that $p \leq \mathrm{m}_{\mathcal{Y}}$, and that $p = \mathrm{m}_{\mathcal{Y}}$ for classical examples, e.g. full-rank K_{Y} and sub-sample without replacement or Gaussian $\mathrm{R}_{\mathcal{Y}}$. We remind and rephrase Proposition 4.1 from chapter 4 that provides the eigenfunctions of the sketched empirical covariance operator.

Proposition 5.2. *(El Ahmad et al., 2024, Proposition 2) The eigenfunctions of the sketched empirical covariance operator* $\widetilde{\mathrm{C}}_{\mathrm{Y}} = \mathrm{S}_{\mathrm{Y}}{}^{\#} \mathrm{R}_{\mathcal{Y}}{}^{\top} \mathrm{R}_{\mathcal{Y}} \mathrm{S}_{\mathrm{Y}}$ *are the* $\tilde{e}_j = \sqrt{\frac{n}{\sigma_j(\widetilde{\mathrm{K}}_{\mathrm{Y}})}} \mathrm{S}_{\mathrm{Y}}{}^{\#} \mathrm{R}_{\mathcal{Y}}{}^{\top} \tilde{\mathbf{u}}_j \in \mathcal{H}_{\mathcal{Y}}$, *for* $j \leq \mathrm{p}_{\mathrm{Y}}$.

Hence, computing the eigenfunctions of $\widetilde{\mathrm{C}}_{\mathrm{Y}}$ provides a basis of $\mathcal{H}_{\mathcal{Y}}$ of dimension p_{Y}. Note that in sketched KPCA, which has been explored via Nyström approximation in (Sterge et al., 2020; Sterge and Sriperumbudur, 2022), one solves for $i = 1, \ldots, \mathrm{m}_{\mathcal{Y}}$

$$f_i = \underset{f \in \mathcal{H}_{\mathcal{Y}}}{\arg\max} \left\{ \langle f, \widehat{\mathrm{C}}_{\mathrm{Y}} f \rangle_{\mathcal{H}_{\mathcal{Y}}} : f \in \tilde{\mathcal{H}}_{\mathcal{Y}}, \left\| f \right\|_{\mathcal{H}_{\mathcal{Y}}} = 1, f \perp \{f_1, \ldots, f_{i-1}\} \right\} \tag{5.7}$$

where $\tilde{\mathcal{H}}_{\mathcal{Y}} = \mathrm{span}((\sum_{j=1}^{n} \mathrm{R}_{\mathcal{Y}_{ij}} \psi_{\mathcal{Y}}(y_j))_{i=1}^{\mathrm{m}_{\mathcal{Y}}})$. Let $\widetilde{\mathrm{P}}_{\mathrm{Y}}$ be the orthogonal projector onto the basis $(\tilde{e}_1, \ldots, \tilde{e}_{\mathrm{p}_{\mathrm{Y}}})$, solving eq. (5.7) is equivalent to compute the eigenfunctions of the projected empirical covariance operator $\widetilde{\mathrm{P}}_{\mathrm{Y}} \widehat{\mathrm{C}}_{\mathrm{Y}} \widetilde{\mathrm{P}}_{\mathrm{Y}}$, i.e., to compute the KPCA of the projected kernel $\langle \widetilde{\mathrm{P}}_{\mathrm{Y}} \psi_{\mathcal{Y}}(\cdot), \widetilde{\mathrm{P}}_{\mathrm{Y}} \psi_{\mathcal{Y}}(\cdot) \rangle_{\mathcal{H}_{\mathcal{Y}}}$. Besides, as for the SVD of $\widetilde{\mathrm{C}}_{\mathrm{Y}}$, sketched KPCA needs the SVD of $\widetilde{\mathrm{K}}_{\mathrm{Y}}$ to obtain its square root, but also requires the additional $\widetilde{K}_{\mathrm{Y}}^{1/2} \mathrm{R}_{\mathcal{Y}} \cdot \mathrm{K}_{\mathrm{Y}}{}^{2} \mathrm{R}_{\mathcal{Y}}{}^{\top} \widetilde{K}_{\mathrm{Y}}^{1/2}$ SVD computation.

Remark 5.3 (Special case of Nyström approximation)**.** *The Nyström approximation is a well-known example of the sketching framework. In that case, the approximation by Nyström subsampling of* $\widetilde{\mathrm{C}}_{\mathrm{Y}}$ *reads* $\widetilde{\mathrm{C}}_{\mathrm{Y}} = (1/\mathrm{m}_{\mathcal{Y}}) \sum_{i=1}^{\mathrm{m}_{\mathcal{Y}}} \psi_{\mathcal{Y}}(\tilde{y}_i) \otimes \psi_{\mathcal{Y}}(\tilde{y}_i)$, *whose eigenfunctions can be computed thanks to the SVD of the approximated Gram matrix* $\widetilde{\mathrm{K}}_{\mathrm{Y}} = (\mathrm{k}_{\mathcal{Y}}(\tilde{y}_i, \tilde{y}_j))_{1 \leq i,j \leq \mathrm{m}_{\mathcal{Y}}}$, *where* $\{(\tilde{y}_i)_{i=1}^{\mathrm{m}_{\mathcal{Y}}}\}$ *are sampled from the training outputs, see Yang et al. (2012); Rudi et al. (2015). This approximation has been leveraged to produce estimators of scalar-valued functions like kernel ridge regression for instance.*

Remark 5.4 (Random Fourier Features). *Another popular kernel approximation is the Random Fourier Features (Rahimi and Recht, 2007; Rudi and Rosasco, 2017; Li et al., 2021). They approximate a kernel function as the inner product of small random features using Monte-Carlo sampling when the kernel writes as the Fourier transform of a probability distribution. Such an approach, however, defines a new randomly approximated kernel, then a new randomly approximated loss, which can induce learning difficulties due to the bias and variance inherent to the approximation. Unlike RFF, sketching is not limited to kernels writing as the Fourier transform of a probability distribution and to defining an approximated loss, it allows the building of a low-dimensional basis within the original feature space of interest.*

Learning the input neural network g_W. Equipped with the basis $\tilde{E} = (\tilde{e}_j)_{j \leq \mathrm{p_Y}}$, we can compute a novel expression of the loss $L(\theta) = L(\tilde{E}, W)$.

Proposition 5.5. *Given the pre-trained basis $\tilde{E} = (\tilde{e}_j)_{j \leq \mathrm{p_Y}}$, $L(\tilde{E}, W)$ expresses as*

$$L(\tilde{E}, W) = \frac{1}{n} \sum_{i=1}^{n} \left\| g_W(x_i) - \tilde{\psi}_{\mathcal{Y}}(y_i) \right\|_2^2, \tag{5.8}$$

where $\tilde{\psi}_{\mathcal{Y}}(y) = (\tilde{e}_1(y), \ldots, \tilde{e}_{\mathrm{p_Y}}(y))^\top = \widetilde{D}_{\mathrm{p_Y}}^{-1/2} \widetilde{U}_{\mathrm{p_Y}}^\top \mathrm{R}_{\mathcal{Y}} \mathrm{k}_{\mathrm{Y}}^{y} \in \mathbb{R}^{\mathrm{p_Y}}$, $\widetilde{U}_{\mathrm{p_Y}} = (\tilde{\mathbf{u}}_1, \ldots, \tilde{\mathbf{u}}_{\mathrm{p_Y}})$, $\widetilde{D}_{\mathrm{p_Y}} = \mathrm{diag}(\sigma_1(\widetilde{\mathrm{K}}_{\mathrm{Y}}), \ldots, \sigma_{\mathrm{p_Y}}(\widetilde{\mathrm{K}}_{\mathrm{Y}}))$, and $\mathrm{k}_{\mathrm{Y}}^{y} = (\mathrm{k}_{\mathcal{Y}}(y, y_1), \ldots, \mathrm{k}_{\mathcal{Y}}(y, y_n))$.

Proof For any pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$, the loss function is given by

$$\left\| h_\theta(x) - \psi_{\mathcal{Y}}(y) \right\|_{\mathcal{H}_{\mathcal{Y}}}^2 = \left\| \sum_{i=1}^{\mathrm{p_Y}} g_W(x)_j \tilde{e}_j - \psi_{\mathcal{Y}}(y) \right\|_{\mathcal{H}_{\mathcal{Y}}}^2 \tag{5.9}$$

$$= \sum_{i,j=1}^{\mathrm{p_Y}} g_W(x)_i g_W(x)_j \langle \tilde{e}_i, \tilde{e}_j \rangle_{\mathcal{H}_{\mathcal{Y}}} - 2 \sum_{j=1}^{\mathrm{p_Y}} g_W(x)_j \langle \tilde{e}_j, \psi_{\mathcal{Y}}(y) \rangle_{\mathcal{H}_{\mathcal{Y}}} + \mathrm{k}_{\mathcal{Y}}(y, y) \tag{5.10}$$

$$= \left\| g_W(x) \right\|_2^2 - 2 g_W(x)^\top \tilde{\psi}_{\mathcal{Y}}(y) + \mathrm{k}_{\mathcal{Y}}(y, y), \tag{5.11}$$

since $\tilde{E}$ is an orthonormal basis, and $\langle \tilde{e}_j, \psi_{\mathcal{Y}}(y) \rangle_{\mathcal{H}_{\mathcal{Y}}} = \tilde{e}_j(y) = \tilde{\psi}_{\mathcal{Y}}(y)_j$ by the reproducing property. Noting that

$$\left\| g_W(x) - \tilde{\psi}_{\mathcal{Y}}(y) \right\|_2^2 = \left\| g_W(x) \right\|_2^2 - 2 g_W(x)^\top \tilde{\psi}_{\mathcal{Y}}(y) + \left\| \tilde{\psi}_{\mathcal{Y}}(y) \right\|_2^2, \tag{5.12}$$

and that both $\mathrm{k}_{\mathcal{Y}}(y, y)$ and $\left\| \tilde{\psi}_{\mathcal{Y}}(y) \right\|_2^2$ are independent of W concludes the proof. ■

Finally, given $\tilde{E}$ and Prop. 5.5, learning the full network h_θ boils down to learning the input neural network g_W and thus finding a solution $\hat{W}$ to

$$\min_{W \in \mathcal{W}} \frac{1}{n} \sum_{i=1}^{n} \left\| g_W(x_i) - \tilde{\psi}_{\mathcal{Y}}(y_i) \right\|_2^2. \tag{5.13}$$

A classical stochastic gradient descent algorithm can then be applied to learn W. Compared to the initial loss (5.3), the relevance of (5.13) is governed by the quality of the approximation of $\widehat{\mathrm{C}}_{\mathrm{Y}}$ by $\widetilde{\mathrm{C}}_{\mathrm{Y}}$. If our approach regularises the solution (the range of the surrogate estimator h_θ is restricted from $\mathcal{H}_{\mathcal{Y}}$ to E), this restriction may not be limiting if we set $\mathrm{m}_{\mathcal{Y}} \geq \mathrm{p_Y}$ high enough to capture all the information contained in $\widehat{\mathrm{C}}_{\mathrm{Y}}$. We discuss strategies to correctly set $\mathrm{m}_{\mathcal{Y}}$ at the beginning of section 5.3.

Beyond the square loss. Thanks to this basis approach and the output kernel trick, an evaluation of the loss is given by eq. (5.11), on which one can easily perform a back-propagation gradient descent to train the neural network's weights W. Moreover, one can easily consider any loss $c(\|z - z'\|^2_{\mathcal{H}_\mathcal{Y}})$, for $z, z' \in \mathcal{H}_\mathcal{Y}$, where $c : \mathbb{R} \to \mathbb{R}$ is a differentiable or sub-differentiable function. In fact, for all $u \in \mathbb{R}$, let $c'(u)$ denotes its derivative or one of its sub-derivative at u, and, for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$, $l(W; x, y) = \|g_E \circ g_W(x) - \psi_\mathcal{Y}(y)\|^2_{\mathcal{H}_\mathcal{Y}}$, then we have that

$$\frac{\partial}{\partial W}\, c(l(W;x,y)) = c'\Big(l(W;x,y)\Big)\left(\frac{\partial}{\partial W}\, \|g_W(x)\|_2^2 - 2\frac{\partial}{\partial W}\; \tilde{\psi}_\mathcal{Y}(y)^\top g_W(x)\right). \tag{5.14}$$

The robust losses considered in Laforgue et al. (2020) typically writes as above. Furthermore, going back to the shallow architecture with an input kernel, an alternative to the double represented theorem of Laforgue et al. (2020) presented in section 2.5.2 would be to use this basis approach and solve the primal ERM problem. Indeed, let $\mathrm{k}_\mathcal{X} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be an input p. d. kernel, $\mathcal{K} = \mathrm{k}_\mathcal{X} I_{\mathrm{p_Y}}$ be an input identity decomposable kernel associated to a vv-RKHS $\mathcal{H}$, and $\lambda > 0$, one could obtain an IOKR surrogate estimator $\hat{h} = g_E \circ \hat{g}$ by solving

$$\min_{g \in \mathcal{H}}\; \frac{1}{n}\sum_{i=1}^{n} c\Big(\|g_E \circ g(x_i) - \psi_\mathcal{Y}(y_i)\|^2_{\mathcal{H}_\mathcal{Y}}\Big) + \lambda \|g\|^2_\mathcal{H}. \tag{5.15}$$

Thanks to the representer theorem (Micchelli and Pontil, 2005), $g_{\hat{W}} : x \mapsto \hat{W}^\top \mathrm{k}^\mathrm{x}_\mathrm{X}$ where $\mathrm{k}^\mathrm{x}_\mathrm{X} = (\mathrm{k}_\mathcal{X}(x, x_i)_{i=1}^n)$ and $\hat{W} \in \mathbb{R}^{n \times \mathrm{p_Y}}$ is the solution to

$$\min_{W \in \mathbb{R}^{n \times \mathrm{p_Y}}}\; \frac{1}{n}\sum_{i=1}^{n} c\Big(k_X^{x_i\top} W W^\top k_X^{x_i} - 2k_X^{x_i\top} W\, \tilde{\psi}_\mathcal{Y}(y) + \mathrm{k}_\mathcal{Y}(y,y)\Big) + \lambda\, \mathrm{Tr}(\mathrm{K_X}\, W W^\top), \tag{5.16}$$

with $\mathrm{K_X} = (\mathrm{k}_\mathcal{X}(x_i, x_j))_{1 \le i,j \le n}$ the input Gram matrix. We refer the reader to Appendix C.1 for further details, in particular with the ε-insensitive ℓ_1, ℓ_2 and Huber losses.

5.2.2 The pre-image problem at inference time

We focus now on the decoding part, i.e., on computing

$$d \circ h_{\hat{\theta}}(x) = \operatorname*{arg\,min}_{y \in \mathcal{Y}}\; \mathrm{k}_\mathcal{Y}(y,y) - 2 g_{\hat{W}}(x)^\top \tilde{\psi}_\mathcal{Y}(y) = \operatorname*{arg\,max}_{y \in \mathcal{Y}}\; g_{\hat{W}}(x)^\top \tilde{\psi}_\mathcal{Y}(y) \tag{5.17}$$

if we assume $\mathrm{k}_\mathcal{Y}$ to be normalized, i.e. $\mathrm{k}_\mathcal{Y}(y, y') = 1, \forall y, y' \in \mathcal{Y}$. For a test set $X^{te} = (x_1^{te}, \ldots, x_{n_{te}}^{te}) \in \mathcal{X}^{n_{te}}$ and a candidate set $Y^c = (y_1^c, \ldots, y_{n_c}^c) \in \mathcal{Y}^{n_c}$, for all $1 \le i \le n_{te}$, the prediction is given by

$$f_{\hat{\theta}}(x_i^{te}) = y_j^c \quad \text{where} \quad j = \operatorname*{arg\,max}_{1 \le j \le n_c}\; g_{\hat{W}}(x_i^{te})^\top \tilde{\psi}(y_j^c). \tag{5.18}$$

Hence, the decoding is particularly suited to problems for which we have some knowledge of the possible outcomes, such as molecular identification problems (Brouard et al., 2016a). More generally, when the range of the outputs' marginal distribution lies within a small subspace of the output space, such an approach is relevant. In other settings, this induces a limitation of the method. To cope with it, some solutions can be explored. When the output kernel is differentiable, it may also be solved using standard gradient-based methods. Finally, some ad-hoc ways to solve the pre-image problem exist for specific kernels, see e.g., Cortes et al. (2007) for the sequence prediction via n-gram kernels, or Korba et al. (2018) for label ranking via Kemeny, Hamming or Lehmer embeddings. The DSOKR framework is summarized in Algorithm 5.1.

Algorithm 5.1 Deep Sketched Output Kernel Regression (DSOKR)

input: training $\{(x_i, y_i)\}_{i=1}^{n}$, validation $\{(x_i^{\text{val}}, y_i^{\text{val}})\}_{i=1}^{n_{\text{val}}}$ pairs, test inputs $\{x_i^{\text{te}}\}_{i=1}^{n_{\text{te}}}$, candidate outputs test inputs $\{y_i^{\text{c}}\}_{i=1}^{n_{\text{c}}}$, normalized output kernel $\text{k}_{\mathcal{Y}}$, sketching matrix $\text{R}_{\mathcal{Y}} \in \mathbb{R}^{\text{m}_{\mathcal{Y}} \times n}$, neural network g_W

init : $\widetilde{\text{K}}_{\text{Y}} = \text{R}_{\mathcal{Y}} \text{K}_{\text{Y}} \text{R}_{\mathcal{Y}}^{\top} \in \mathbb{R}^{\text{m}_{\mathcal{Y}} \times \text{m}_{\mathcal{Y}}}$ where $\text{K}_{\text{Y}} = (\text{k}_{\mathcal{Y}}(y_i, y_j))_{1 \leq i,j \leq n} \in \mathbb{R}^{n \times n}$

// 1. a. Training of g_E: computations for the basis $\widetilde{E}$

- Construct $\widetilde{D}_{\text{p}_\text{Y}} \in \mathbb{R}^{\text{p}_\text{Y} \times \text{p}_\text{Y}}$, $\widetilde{U}_{\text{p}_\text{Y}} \in \mathbb{R}^{\text{m}_{\mathcal{Y}} \times \text{p}_\text{Y}}$ such that $\widetilde{U}_{\text{p}_\text{Y}} \widetilde{D}_{\text{p}_\text{Y}} \widetilde{U}_{\text{p}_\text{Y}}^{\top} = \widetilde{\text{K}}_{\text{Y}}$ (SVD of $\widetilde{\text{K}}_{\text{Y}}$)
- $\widetilde{\Omega} = \widetilde{D}_{\text{p}_\text{Y}}^{-1/2} \widetilde{U}_{\text{p}_\text{Y}}^{\top} \in \mathbb{R}^{\text{p}_\text{Y} \times \text{m}_{\mathcal{Y}}}$

// 1. b. Training of g_W: solving the surrogate problem

- $\tilde{\psi}_{\mathcal{Y}}(y_i) = \widetilde{\Omega} \, \text{R}_{\mathcal{Y}} \, k_Y^{y_i} \in \mathbb{R}^{\text{p}_\text{Y}}, \forall \; 1 \leq i \leq n$, $\tilde{\psi}_{\mathcal{Y}}(y_i^{\text{val}}) = \widetilde{\Omega} \, \text{R}_{\mathcal{Y}} \, k_Y^{y_i^{\text{val}}} \in \mathbb{R}^{\text{p}_\text{Y}}, \forall \; 1 \leq i \leq n_{\text{val}}$
- $\hat{W} = \underset{W \in \mathcal{W}}{\arg\min} \; \frac{1}{n} \sum_{i=1}^{n} \left\| g_W(x_i) - \tilde{\psi}_{\mathcal{Y}}(y_i) \right\|_2^2$ (training of g_W with training $\{(x_i, \tilde{\psi}_{\mathcal{Y}}(y_i))\}_{i=1}^{n}$ and validation $\{(x_i^{\text{val}}, \tilde{\psi}_{\mathcal{Y}}(y_i^{\text{val}}))\}_{i=1}^{n_{\text{val}}}$ pairs and Mean Squared Error loss)

// 2. Inference

- $\tilde{\psi}_{\mathcal{Y}}(y_i^{\text{c}}) = \widetilde{\Omega} \, \text{R}_{\mathcal{Y}} \, k_Y^{y_i^{\text{c}}} \in \mathbb{R}^{\text{p}_\text{Y}}, \forall \; 1 \leq i \leq n_{\text{c}}$
- $f_{\hat{\theta}}(x_i^{\text{te}}) = y_j^{\text{c}}$ where $j = \underset{1 \leq j \leq n_{\text{c}}}{\arg\max} \; g_{\hat{W}}(x_i^{\text{te}})^{\top} \tilde{\psi}(y_j^{\text{c}}), \; \forall \; 1 \leq i \leq n_{\text{te}}$

return $f_{\hat{\theta}}(x_i^{\text{te}}), \forall \; 1 \leq i \leq n_{\text{te}}$

Ensemble strategy. Another interesting feature of DSOKR is the fact that the computation of DSOKR's last layer g_E depends on a draw of the sketching matrix $\text{R}_{\mathcal{Y}}$, which means that DSOKR is particularly well-suited to the aggregation via multiple draws of the sketching matrix and the training of the corresponding neural networks. For instance, we can easily consider two ways of aggregating multiple DSOKR models at the pre-image stage, either by averaging or maximizing these models' scores. Let $T \in \mathbb{N}^*$, and for $t \in [\![T]\!]$, $h_{\hat{\theta}_t} = g_{\tilde{E}_t} \circ g_{\hat{W}_t}$ denotes the trained DSOKR neural network based on the sketching matrix $\text{R}_{\mathcal{Y}_t}$, for any input x^{te} and candidate y^c, the score to maximize during the pre-image problem is given by

$$s^{\text{mean}}(x^{te}, y^c) = \sum_{t=1}^{T} \omega_t \; g_{\hat{W}_t}(x^{te})^{\top} \tilde{\psi}_{\mathcal{Y}_t}(y^c) \quad \text{or} \quad s^{\max}(x^{te}, y^c) = \underset{1 \leq t \leq T}{\arg\max} \; g_{\hat{W}_t}(x^{te})^{\top} \tilde{\psi}_{\mathcal{Y}_t}(y^c), \tag{5.19}$$

where $\omega_t \geq 0$ for all $t \in [\![T]\!]$ and $\sum_{t=1}^{T} \omega_t$. Such an approach reduces the bias induced by a single draw of the sketching matrix and leads to better results, as pointed out by the experiments led on the ChEBI-20 dataset, see section 5.3.3.

5.3 Experiments

In this section, we first present a range of strategies to select the sketching size and an analysis of our proposed DSOKR on a synthetic dataset. Besides, we show the effectiveness of DSOKR through its application to two real-world Supervised Graph Prediction (SGP) tasks: SMILES to Molecule and Text to Molecule. The code to reproduce our results is available at: https://github.com/tamim-el/dsokr.

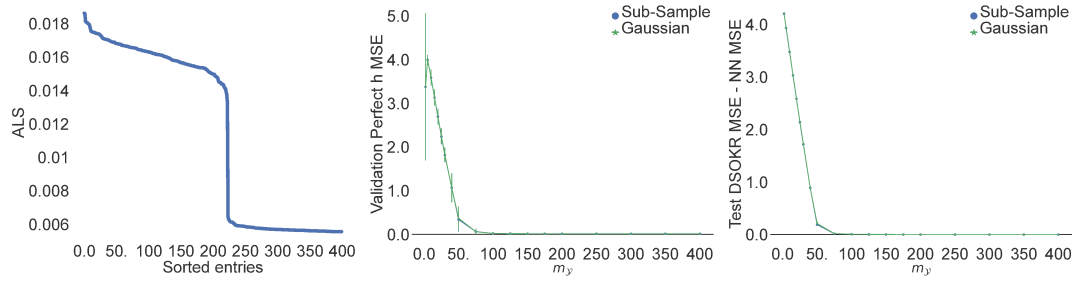


Figure 5.2: Sorted 400 highest ALS (left), validation MSE of *Perfect h* w.r.t. $\mathrm{m}_\mathcal{Y}$ (center) and the difference between test MSE of DSOKR and NN w.r.t. $\mathrm{m}_\mathcal{Y}$ (right).

Sketching size selection strategy. A critical DSOKR's hyper-parameter is the sketching size $\mathrm{m}_\mathcal{Y}$. Indeed, the optimal choice is the dimension of the subspace containing the output features. However, to estimate this dimension, one has to compute the eigenvalues of K_Y, which has the prohibitive complexity of $\mathcal{O}(n^3)$. Hence, a first solution is to compute the Approximate Leverage Scores (ALS) as described in Alaoui and Mahoney (2015). This is an approximation of the eigenvalues of K_Y that relies on sub-sampling $n_S < n$ entries within the whole training set. Moreover, we use another technique that we call *Perfect h*. Considering any pair (x, y) in a validation set, we replace $g_W(x)$ by the "perfect" coefficients of the expansion, i.e., for each $j = 1, \ldots, \mathrm{p}_\mathrm{Y}$, $\langle \tilde{e}_j, \psi_\mathcal{Y}(y) \rangle_{\mathcal{H}_\mathcal{Y}}$ and define "perfect" surrogate estimator $h_{\psi_\mathcal{Y}}$ as follows

$$h_{\psi_\mathcal{Y}}(x) = \sum_{j=1}^{\mathrm{p}_\mathrm{Y}} \langle \tilde{e}_j, \psi_\mathcal{Y}(y) \rangle_{\mathcal{H}_\mathcal{Y}} \, \tilde{e}_j = \sum_{j=1}^{\mathrm{p}_\mathrm{Y}} \tilde{\psi}_\mathcal{Y}(y)_j \, \tilde{e}_j \,. \tag{5.20}$$

Then, we evaluate the performance of this "perfect" surrogate estimator $h_{\psi_\mathcal{Y}}$ on a validation set to select $\mathrm{m}_\mathcal{Y}$. Hence, *Perfect h* allows to select the minimal $\mathrm{m}_\mathcal{Y}$ in the range given by ALS such that the performance of $h_{\psi_\mathcal{Y}}$ reaches an optimal value.

5.3.1 Analysis of DSOKR on Synthetic Least Squares Regression

Dataset. We generate a synthetic dataset of least-squares regression, using then a linear output kernel, with $n = 50,000$ training data points, $\mathcal{X} = \mathbb{R}^{2,000}$, $\mathcal{Y} = \mathbb{R}^{1,000}$, and $\mathcal{H}_\mathcal{Y} = \mathcal{Y} = \mathbb{R}^{1,000}$. The goal is to build this dataset such that the outputs lie in a subspace of $\mathcal{Y}$ of dimension $d = 50 < 1,000$. Hence, given d randomly drawn orthonormal vectors $(u_j)_{j=1}^d$, for all $1 \leq i \leq n$, the outputs are such that $y_i = \sum_{j=1}^d \alpha(x_i)_j u_j + \varepsilon_i$, where α is a function of the inputs and $\varepsilon_i \sim \mathcal{N}(0, \sigma^2 I_{1,000})$ are i.i.d. with $\sigma^2 = 0.01$. We generate i.i.d. normal distributed inputs $x_i \sim \mathcal{N}(0, C)$, where $(\sigma_j(C) = j^{-1/2})_{j=1}^{2,000}$ and its eigenvectors are randomly drawn. Finally, we draw $H \in \mathbb{R}^{d \times 2,000}$ with i.i.d. coefficients from the standard normal distribution, and the outputs are given for $1 \leq i \leq n$ by

$$y_i = UHx_i + \varepsilon_i \,, \tag{5.21}$$

where $U = (u_1, \ldots, u_d) \in \mathbb{R}^{1,000 \times d}$. We generate validation and test sets of $n_{\mathrm{val}} = 5,000$ and $n_{\mathrm{te}} = 10,000$ points in the same way.

Experimental settings. We first compute the ALS as described above. We take as regularisation penalty $\lambda = 10^{-4}$, sampling parameter $n_S = \sqrt{n}$ and probability vector $(p_i = 1/n)_{i=1}^n$ (uniform sampling). Then, we perform the sketching size selection strategy *Perfect h*. Note that using a linear output kernel, $\psi_\mathcal{Y} : y \in \mathbb{R}^{1,000} \mapsto y$, then $\tilde{e}_i = (1/\sqrt{\sigma_i(\widetilde{\mathrm{K}}_\mathrm{Y})})\tilde{\mathbf{u}}_i^\top \mathrm{R}_\mathcal{Y} Y$, where $Y = (y_1, \ldots, y_n)^\top \in \mathbb{R}^{n \times 1,000}$, and

$$h_{\hat{\theta}}(x) = Y^\top \mathrm{R}_\mathcal{Y}{}^\top \widetilde{U}_{\mathrm{p_Y}} \widetilde{D}_{\mathrm{p_Y}}^{-1/2} g_{\hat{W}}(x). \tag{5.22}$$

Finally, we perform our DSOKR model whose neural network g_W is a Single-Layer Perceptron, i.e. with no hidden layer, and compare it with an SLP whose output size is 1,000, and trained with a Mean Squared Error loss, that we call "NN". We select the optimal number of epochs thanks to the validation set and evaluate the performance via the MSE. We use the ADAM (Kingma and Ba, 2015) optimizer. For the *Perfect h* and DSOKR models and any sketching size $\mathrm{m}_\mathcal{Y} \in [2, 400]$, we average the results over five replicates of the models. We use uniform sub-sampling without replacement and Gaussian sketching distributions.

Experimental results. Figure 5.2 (left) presents the sorted 400 highest leverage scores. This gives a rough estimate of the optimal sketching size since the leverage scores converge to a minimal value starting from 200 approximately, which is an upper bound of the true basis dimension $d = 50$. Figure 5.2 (center) shows that *Perfect h* is a relevant strategy to fine-tune $\mathrm{m}_\mathcal{Y}$ since the obtained optimal value is $\mathrm{m}_\mathcal{Y} = 75$, which is very close to $d = 50$. This small difference comes from the added noise ε_i. Moreover, this value corresponds to the optimal value based on the DSOKR test MSE. In fact, Figure 5.2 (right) presents the performance DSOKR for many $\mathrm{m}_\mathcal{Y}$ values compared with NN. DSOKR performance converges to the NN's performance for $\mathrm{m}_\mathcal{Y} = 75$ as well. Hence, we show that DSOKR attains optimal performance if its sketching size is set as the dimension of the output marginal distribution's range, which can be estimated thanks to the ALS and the *Perfect h* strategies. There is no difference between sub-sample and Gaussian sketching since the dataset is rather simple. Moreover, note that the neural network of the DSOKR model for $\mathrm{m}_\mathcal{Y} = 75$ contains 150,075 parameters, whereas the NN model contains 2,001,000 parameters. Then, our sketched basis strategy, even in the context of multi-output regression, allows to reduce the size of the last layer, simplifying the regression problem and reducing the number of weights to learn.

5.3.2 SMILES to Molecule: SMI2Mol

Dataset. We use the QM9 molecule dataset (Ruddigkeit et al., 2012; Ramakrishnan et al., 2014), containing around 130,000 small organic molecules. These molecules have been processed using RDKit[1], with aromatic rings converted to their Kekule form and hydrogen atoms removed. We also remove molecules containing only one atom. Each molecule contains up to 9 atoms of Carbon, Nitrogen, Oxygen, or Fluorine, along with three types of bonds: single, double, and triple. As input features, we use the Simplified Molecular Input Line-Entry System (SMILES), which are strings describing their chemical structure. We refer to the resulting dataset as **SMI2Mol**.

[1] RDKit: Open-source cheminformatics. https://www.rdkit.org

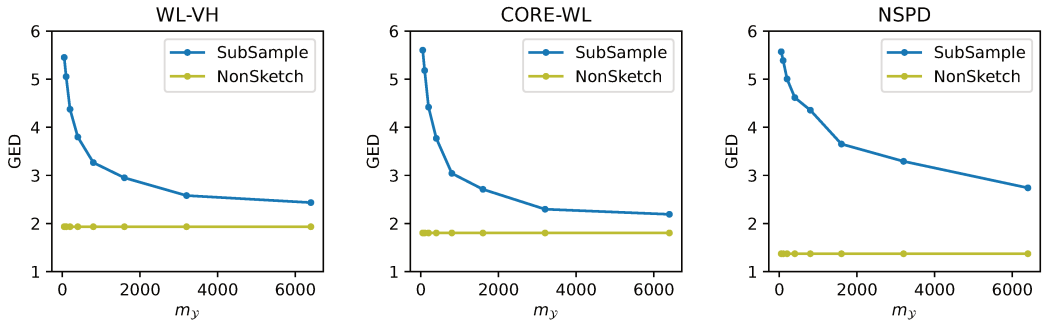


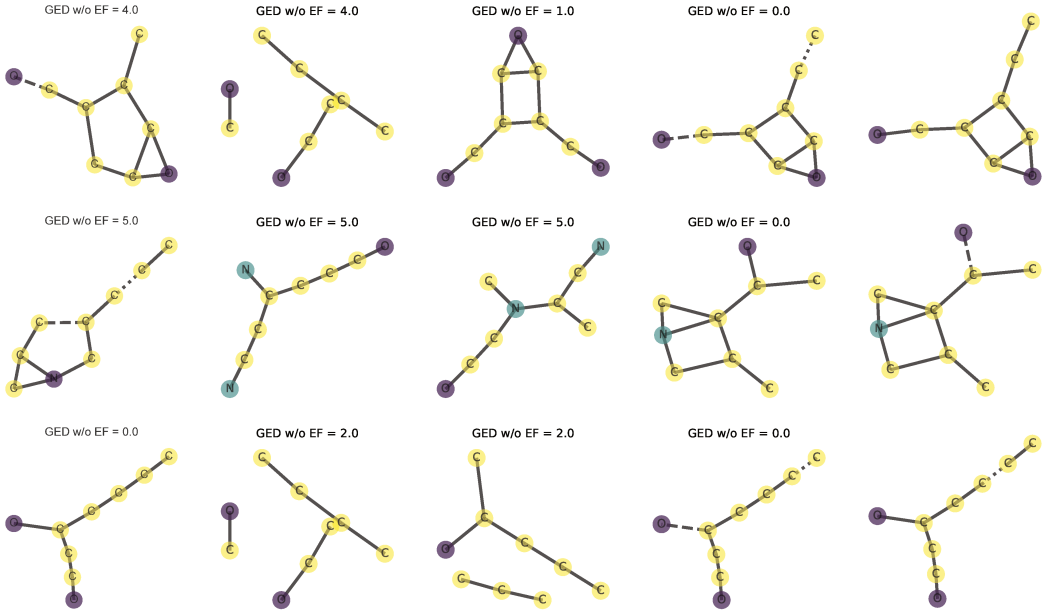
Figure 5.3: The GED w/ edge feature w.r.t. thes ketching size m_y for *Perfect h* for three graph kernels on SMI2Mol ($m_y > 6400$ is too costly computationally).

Table 5.1: Edit distance of different methods on SMI2Mol test set

	GED w/o edge feature ↓	GED w/ edge feature ↓
SISOKR	3.330 ± 0.080	4.192 ± 0.109
NNBary-FGW	5.115 ± 0.129	-
Sketched ILE-FGW	2.998 ± 0.253	-
DSOKR	$\mathbf{1.951 \pm 0.074}$	$\mathbf{2.960 \pm 0.079}$

Experimental set-up. Using all SMILES-Molecule pairs, we build five splits using different seeds. Each split has 131,382 training samples, 500 validation samples, and 2,000 test samples. In DSOKR, g_W is a Transformer (Vaswani et al., 2017). The SMILES strings are tokenized into character sequences as inputs for the Transformer encoder. To define the loss on output molecules, we cross-validate several graph kernels, including the Weisfeiler-Lehman subtree kernel (WL-VH) (Shervashidze et al., 2011), the neighborhood subgraph pairwise distance kernel (NSPD) (Costa and Grave, 2010), and the core Weisfeiler-Lehman subtree kernel (CORE-WL) (Nikolentzos et al., 2018). Note that NSPD accounts for edge labels, unlike the other two kernels, see Appendix C.2 for more details. We use the implementation of the graph kernels provided by the Python library GraKel (Siglidis et al., 2020). We employ sub-sample sketching for the output kernel. The sketching size m_y is fixed using our proposed *Perfect h* strategy. Our method is benchmarked against SISOKR (El Ahmad et al., 2024), NNBary-FGW (Brogat-Motte et al., 2022a), and ILE-FGW (Brogat-Motte et al., 2022a). For ILE-FGW and SISOKR, we additionally use SubSample sketching (Rudi et al., 2015) for input kernel approximation. To ensure a fair comparison, both SISOKR and ILE-FGW adopt the 3-gram kernel for the input strings, whereas NNBary-FGW and DSOKR use a Transformer encoder. The performance is evaluated using Graph Edit Distance (GED), implemented by the NetworkX package (Hagberg et al., 2008).

Experimental results. Figure 5.3 displays the GED obtained by *Perfect h* concerning various graph kernels. Based on this visualization, we have set the sketching sizes of WL-VH, CORE-WL, and NPSD to 3200, 3200, and 6400 respectively. Table 5.1



(a) SISOKR (b) NNBary (c) ILE (d) DSOKR (e) True target

Figure 5.4: Predicted molecules on the SMI2Mol dataset.

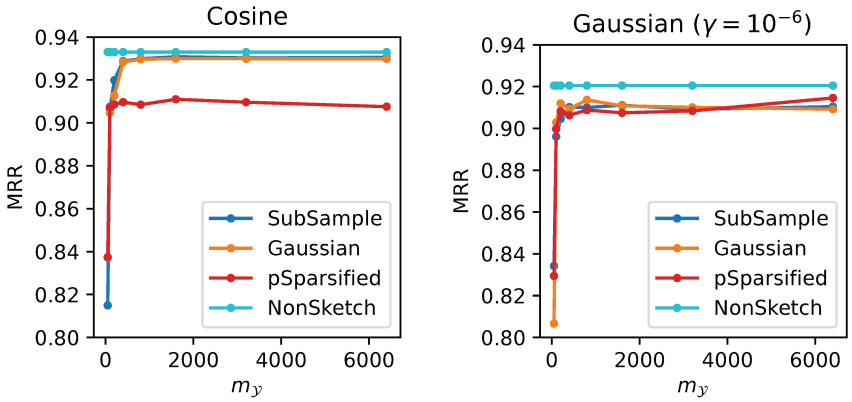


Figure 5.5: The MRR scores on ChEBI-20 validation set w.r.t. m_y for *Perfect h* when the output kernel is Cosine or Gaussian on the ChEBI-20 dataset.

showcases the performance of various methods of SGP. Notably, DSOKR outperforms all baseline methods. It is evident that while graph kernels and the fused Gromov-Wasserstein (FGW) distance induce a meaningful feature space, the capabilities of SISOKR and ILE-FGW are constrained by the input kernels, thus highlighting the relevance of our proposed method. For further insight, a comparison of some prediction examples is provided in Figure 5.4 and Appendix C.3.

Table 5.2: Performance of different methods on ChEBI-20 test set. All the methods based on NNs use SciBERT as input text encoder for fair comparison. The number in the ensemble setting indicates the number of single models used.

	Hits@1 ↑	Hits@10 ↑	MRR ↑
SISOKR	0.4%	2.8%	0.015
SciBERT Regression	16.8%	56.9%	0.298
CMAM - MLP	34.9%	84.2%	0.513
CMAM - GCN	33.2%	82.5%	0.495
CMAM - Ensemble (MLP×3)	39.8%	87.6%	0.562
CMAM - Ensemble (GCN×3)	39.0%	87.0%	0.551
CMAM - Ensemble (MLP×3 + GCN×3)	44.2%	**88.7**%	0.597
DSOKR - SubSample Sketch	48.2%	87.4%	0.624
DSOKR - Gaussian Sketch	49.0%	87.5%	0.630
DSOKR - Ensemble (SubSample×3)	**51.0**%	88.2%	**0.642**
DSOKR - Ensemble (Gaussian×3)	50.5%	87.9%	**0.642**
DSOKR - Ensemble (SubSample×3 + Gaussian×3)	50.0%	88.3%	0.640

5.3.3 Text to Molecule: ChEBI-20

Dataset. The ChEBI-20 (Edwards et al., 2021) dataset contains 33,010 pairs of compounds and descriptions. The compounds come from PubChem (Kim et al., 2016, 2019), and their descriptions (more than 20 words) from the Chemical Entities of Biological Interest (ChEBI) database (Hastings et al., 2016). The dataset is divided as follows: 80% for training, 10% for validation, and 10% for testing. The candidate set contains all compounds. The mean and median number of atoms per molecule is 32 and 25 respectively, and the mean and median number of words per description is 55 and 51 respectively.

Experimental set-up. For our method DSOKR, we use SciBERT (Beltagy et al., 2019) with an additional linear layer to parameterize g_W. The maximum length of the input tokens is set to 256. Mol2vec (Jaeger et al., 2018) is used as the output molecule representation, which is a vector of dimension 300. Based on the Mol2vec representation, we conduct cross-validation using the following kernels: Cosine kernel and Gaussian kernel with gamma chosen from $\{10^{-9}, 10^{-6}, 10^{-3}, 1\}$, along with the following three sketches: sub-sampling (Rudi et al., 2015), Gaussian (Yang et al., 2017), and p-sparsified (El Ahmad et al., 2023). The sketching size for all combinations of the output kernels and sketches is determined using the *Perfect h* strategy. As for the baselines, we consider SciBERT Regression, Cross-Modal Attention Model (CMAM) (Edwards et al., 2021), and SISOKR. In the case of SciBERT Regression, we address the regression problem using Mean Squared Error loss, where the output space is the embedding space of Mol2vec, within a function space parameterized by SciBERT. CMAM aims to enhance the cosine similarity between the text embedding and the corresponding molecule in true pairs by employing a contrastive loss function. Specifically, the former is derived from SciBERT, while the latter is generated using either a multi-layer perceptron (MLP) or a graph convolutional network (GCN) atop the Mol2vec repres-

entation. We reproduce the results of CMAM with the codes[2] released by Edwards et al. (2021). In SISOKR, we use SciBERT embeddings as input features, leveraging the cosine kernel atop them. We maintain the identical output kernel sketching setup as in DSOKR. For all methods, we train the model using the best hyper-parameters with three random seeds and report the one with the best validation performance The performance is evaluated with mean reciprocal rank (MRR), Hits@1 and Hits@10. We could not benchmark AMAN (Zhao et al., 2024), as no implementation is publicly available.

Ensemble. In Edwards et al. (2021), the authors propose an ensemble strategy to enhance the results by aggregating the ranks obtained by different training of their models. If for each $1 \leq t \leq T$, R_t denotes the ranking returned by the model t, the new score is computed as follows

$$s(y_i) = \sum_{t=1}^{T} w_t R_t(y_i) \quad s.t. \quad \sum_{i=1}^{T} \omega_t = 1 \tag{5.23}$$

for each y_i in the candidate set. As discussed in section 5.2.2, DSOKR is particularly amenable to ensemble strategies based on multiple draws of the sketching matrix, such as the averaging or maximizing strategies described in eq. (5.19). We explore the ensemble method proposed by Edwards et al. (2021) as well as the two methods we propose for DSOKR models and subsequently select the optimal one based on its validation performance.

Experimental results. Figure 5.5 illustrates the validation MRR scores with *Perfect h*, for many m values, and either Cosine or Gaussian output kernels. It is evident that for both the Cosine kernel and Gaussian kernel (with $\gamma = 10^{-6}$) employing various sketching methods, the MRR score stabilizes as the sketching size exceeds 100, and that Cosine outperforms Gaussian. This observation allows us to choose $m_{\mathcal{Y}} = 100$, smaller than the original Mol2vec dimension, which is 300. Table 5.2 presents a comprehensive comparison of DSOKR with various baseline models. Firstly, comparing DSOKR with SISOKR reveals the critical importance of employing deep neural networks when dealing with complex structured inputs and DSOKR makes it possible in the case of functional output space. Secondly, the notable improvement over SciBERT Regression underscores the value of employing kernel sketching to derive more compact and better output features, thereby facilitating regression problem-solving. Lastly, DSOKR outperforms the sota CMAP for both single and ensemble models. See Appendix C.3 for more details.

5.4 Conclusion

We designed a new architecture of neural networks able to minimize kernel-induced losses for structured prediction and achieve state-of-the-art performance on molecular identification. Moreover, we exploited the amenability of sketching to ensemble strategies by proposing two methods of aggregating multiple DSOKR models corresponding to multiple draws of the sketching matrix at the inference stage. This reduces the bias induced by a single draw of the sketching matrix and enhances the empirical

[2] https://github.com/cnedwards/text2mol

performance of the model, as highlighted in the experiments on the ChEBI-20 dataset. An interesting avenue for future work is to derive excess risk for this estimator by combining deep learning theory and surrogate regression bounds.

6 Conclusion

Contents

6.1 Summary of the Contributions

In this thesis, we addressed structured prediction in the supervised learning setting, going beyond the standard problems of regression or classification where the output data to predict are usually low-dimensional vectors. In structured prediction, outputs are instead complex objects (e.g. graphs, permutations, or sequences), and we had to face new challenges due to the high-dimension, the lack of linear structure, and the large size of such discrete structured spaces.

We chose to build upon surrogate kernel methods, and in particular Input Output Kernel Regression, due to their versatility to handle various output objects within a unified framework, as well as their strong theoretical foundations, which are not common characteristics in structured prediction. However, such approaches suffer from three main limitations:

- they fail to scale to large datasets, both for training and inference phases because of the inherent computational costs induced by kernel methods;
- they strongly rely on the closed-form solution induced by the square loss, and fail to extend to other losses, such as robust losses in the case of output outliers;
- they fail to learn representations from complex input data such as images or texts.

Our goal was then to design a structured prediction model incorporating the following four characteristics:

1. scalability to large datasets;
2. excess risk bound for the built estimator;
3. ability to use a wider variety of losses;
4. ability to learn representations from complex inputs.

In Chapter 3, we first focused on the input kernel in the simpler settings of multi-output regression with matrix-valued decomposable kernels, to provide a new sketching distribution - the *p-sparsified sketches* - and scale kernel machines in such settings, using Lipschitz-continuous losses. We also proved excess risk bounds for the resulting estimator and conducted experiments on synthetic and real-world joint quantile and multi-output regression. The Python implementation is available on GitHub.

Equipped with this sketching distribution, we proposed, in Chapter 4, *Sketched Input Sketched Output Kernel Regression*, a version of IOKR using sketching matrices on both the input and output kernels to accelerate it during both training and inference phases respectively. We provided an excess risk bound for any SISOKR estimator leveraging sub-Gaussian or sub-sampling sketching distributions, demonstrating close-to-optimal learning rates. Real-world experiments show that SISOKR reaches good statistical performance on a dataset intractable for IOKR. The code to reproduce the results is available on GitHub as well.

Finally, in Chapter 5, we introduced *Deep Sketched Output Kernel Regression*, a neural architecture compatible with kernel-induced losses thanks to sketching applied to the output kernel. Inspired by SISOKR, we are able to compute a small-dimensional basis within the output feature space and the DSOKR estimator then consists of a deep expansion within this basis. This allows us to use standard gradient-based methods to train any neural architecture for a wider variety of losses than the sole square one. Experiments on real-world molecular identification from text input datasets show the relevance of DSOKR and highlight the importance of incorporating deep neural networks to Output Kernel Regression, or equivalently to unlock kernel-induced losses to deep neural networks. The Python implementation is available on GitHub.

6.2 Perspectives

The work carried out in this thesis opens up many perspectives, that we discuss below.

- **Sketched Decomposable Kernel Learning.** The sketched matrix-valued decomposable kernel machines approach in Chapter 3 could be extended to the Decomposable Kernel Learning framework (Dinuzzo et al., 2011; Lim et al., 2015) by introducing another sketching matrix $R_\mathcal{Y} \in \mathbb{R}^{m_\mathcal{Y} \times d}$, with $m_\mathcal{Y} < d$, applied to the matrix $M \in \mathbb{R}^{d \times d}$. This would be of high interest for rather high dimensional output spaces, i.e. high d, and would bridge the gap between Chapter 3, focused on the input kernel, and Chapter 4, where the sketching of the output kernel is introduced.

- **Differentially private learning thanks to the p-sparsified sketches.** A major concern about machine learning nowadays is differential privacy (Dwork and Roth, 2014). Since the distribution of the p-sparsified sketches from El Ahmad et al. (2023) is independent of the data, it could be possible to make sketched kernel machines private by adding less noise than what would be needed to make a standard non-sketched kernel machine private, as in Jain and Thakurta (2013). Such an approach could be extended to structured prediction thanks to SISOKR. However, even if the p-sparsified sketches' distribution is independent of the data, note that sketching is a data-dependent kernel approximation and a line of research about differentially private kernel machines lies in the use

of data-independent random features such as Random Fourier Features, see for instance Harder et al. (2021) that focuses on kernel mean embeddings.

- **Relaxation of the attainability assumption for SISOKR.** Inspired by Ciliberto et al. (2020), an interesting theoretical line of research for SISOKR would be to relax Assumption 4.3, i.e. the attainability assumption, and to assume the target function h^* to be square-integrable. Dropping the existence assumption of an operator $H : \mathcal{H}_\mathcal{X} \to \mathcal{H}_\mathcal{Y}$ such that $h^* = H\,\psi_\mathcal{X}(\cdot)$ and $\|H\|_{\text{HS}} < \infty$ would lead to a slightly different error decomposition as in Ciliberto et al. (2020, Lemma B.2).

- **Theoretical analysis of the regularisation effect of sketching.** In the theoretical analysis of the SISOKR estimator, we concluded that it attains close-to-optimal learning rates in comparison to the non-sketched classical KRR estimator. This is due to the error decomposition we considered that writes as the sum of this standard error and the errors induced by sketching both the input and output kernels. By considering another decomposition, it would be possible to analyze to which extent and in which conditions sketching can lead to statistical improvements, as in Rudi et al. (2015) that focuses on the Nyström approximated scalar-valued kernel machines, or Brogat-Motte et al. (2022b) that focuses on a learned, and not randomly obtained, reduced-rank vector-valued estimator.

- **Even more scalable version of SISOKR.** Besides a batch approach, SISOKR could be combined with other large-scale techniques, such as preconditioning and a GPU-optimized implementation as in Meanti et al. (2020), to finally obtain a structured prediction approach scaling to datasets with billions of samples.

- **Batch version of SISOKR and DSOKR.** In Chapters 4 and 5, we showed that SISOKR scales to datasets with 60 000 training data such as Bookmarsk, and 131 382 training data such as QM9. To attain datasets with millions of samples, for instance, we could combine it with a batch approach. The aggregation of each $\hat{h}_i$ obtained by solving the surrogate regression problems over each batch can merely be their uniform average $\hat{h} = (1/B)\sum_{i=1}^{B} \hat{h}_i$, where B is the number of batches. In this case, inspired from Zhang et al. (2015), we could obtain its excess risk bound. Otherwise, the aggregation can be done at the inference step as in Section 5.3.3, or by maximizing the average of maximum scores obtained by each $\hat{h}_i$. Such an approach could also be leveraged for DSOKR.

- **Extend SISOKR and DSOKR to any Implicit Loss Embeddings.** In this thesis, we focused on Output Kernel Regression. Thus, SISOKR and DSOKR approaches could be extended to generic structured prediction surrogate methods in the ILE framework (Ciliberto et al., 2020).

- **Excess risk bound for DSOKR.** With DSOKR, we finally obtained a structured prediction model that fulfills three of the four criteria initially defined at the beginning of this thesis. The only objective left is to derive an excess risk bound of the DSOKR estimator, which is challenging due to the presence of the input neural network. Thanks to the comparison inequality, it boils down to studying the surrogate excess risk. To do so, we could build upon the SISOKR excess risk bound's proof, and in particular, the effect induced by output sketching, together with the excess risk bound of deep NN with ReLU activation functions in the non-parametric regression settings (Schmidt-Hieber, 2017).

- **End-to-end version of DSOKR.** DSOKR is a two-step approach where a pre-image problem is solved at the inference step within output candidates. While it is a strength of the method for problems where we have knowledge about the possible outcomes, as in molecular identification, it is a limitation for other problems. Building then an end-to-end DSOKR model either thanks to a direct risk minimization technique as in Belanger et al. (2017) when the output kernel is differentiable, or with a differentiable approximation (Berthet et al., 2020; Niculae and Martins, 2020) of it when it is not the case, or thanks to an inference neural network $d_\theta : \mathcal{H}_\mathcal{Y} \to \mathcal{Y}$ as in Tu and Gimpel (2018) would be of particular interest. In the latter case, DSOKR would boil down to an autoencoder whose latent space is $\mathcal{H}_\mathcal{Y}$.

- **Extension to the unsupervised settings.** DSOKR, and in particular its basis approach, could be extended to the autoencoder architecture, and consequently the unsupervised settings. In fact, assuming that the input data are structured objects, we could consider a relevant input kernel and compute the eigenfunctions of its sketched empirical covariance operator and have the first and last layer of the autoencoder computing expansions within this basis, as DSOKR's last layer does. Unlike the Kernel AutoEncoder from Laforgue et al. (2019) where all layers are functions lying in vector-valued Reproducing Kernel Hilbert Spaces, only the first and last layers would be functions taking values in a generic Hilbert space.

- **Extension of sketching the output kernel for other OKR approaches.** The sketched output kernel approach could be extended, either thanks to the induced orthogonal projector operator or small-dimensional basis within $\mathcal{H}_\mathcal{Y}$, to other OKR approaches, such as trees (Geurts et al., 2006). We started investigating the empirical benefits of sketching the output kernel in the context of kernelized trees in collaboration with Wen Yang. From theoretical perspectives, by combining the analysis of the error induced by sketching the output kernel together with the theoretical analysis of random forest (Scornet et al., 2015; Scornet, 2016), we could obtain an excess risk bound of sketched kernelized trees for instance.

- **A Python package for structured prediction.** We currently work in collaboration with *HI! PARIS* to integrate SISOKR and DSOKR models in a unified open source Python library for structured prediction.

Appendix

A Appendices for Chapter 3

A.1 Technical Proofs

In this section are gathered all the technical proofs of the results stated in chapter 3.

Notation. We recall that we assume that training data $(x_i, y_i)_{i=1}^n$ are i.i.d. realisations sampled from a joint probability density ρ. We define

$$\mathbb{E}_n[\ell_f] = \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i), y_i),$$
$$\mathbb{E}[\ell_f] = \mathbb{E}_\rho[\ell(f(X), Y].$$

For a class of functions F, the empirical Rademacher complexity (Bartlett and Mendelson, 2003) is defined as

$$\hat{R}_n(F) = \mathbb{E}\left[\sup_{f\in F}\left|\frac{2}{n}\sum_{i=1}^{n}\epsilon_i f(x_i)\right| | x_1, \dots, x_n\right],$$

where $\epsilon_1, \dots, \epsilon_n$ are independent Rademacher random variables such that $\mathbb{P}\{\epsilon_i = 1\} = \mathbb{P}\{\epsilon_i = -1\} = 1/2$. The corresponding Rademacher complexity is then defined as the expectation of the empirical Rademacher complexity

$$R_n(F) = \mathbb{E}\left[\hat{R}_n(F)\right].$$

Proof of Theorem 3.10

We first prove the first inequality in theorem 3.10 for generic Lipschitz losses.

Theorem 3.10. *Let $\tilde{f}$ as in Definition 3.3, suppose that Assumptions 3.5 to 3.9 hold, and let $C = 1 + \sqrt{6}c$, with c the constant from Assumption 3.9. Then, for any $\delta \in (0,1)$ with probability at least $1-\delta$ we have*

$$\mathbb{E}\left[\ell_{\tilde{f}}\right] \leq \mathbb{E}\left[\ell_{f_{\mathcal{H}_\mathcal{X}}}\right] + LC\sqrt{\lambda_n + \delta_n^2} + \frac{\lambda_n}{2} + 8L\sqrt{\frac{\kappa_\mathcal{X}}{n}} + 2\sqrt{\frac{8\log\left(4/\delta\right)}{n}}, \tag{3.5}$$

where $\mathbb{E}\left[\ell_f\right] = \mathbb{E}_{(X,Y)\sim\rho}\left[\ell(f(X), Y)\right]$. Furthermore, if $\ell\left(z, y\right) = \left(z-y\right)^2/2$ and $\mathcal{Y} \subset [0,1]$, with probability at least $1-\delta$ we have

$$\mathbb{E}\left[\ell_{\tilde{f}}\right] \leq \mathbb{E}\left[\ell_{f_{\mathcal{H}_\mathcal{X}}}\right] + \left(C^2 + \frac{1}{2}\right)\lambda_n + C^2\delta_n^2 + 8\frac{\kappa_\mathcal{X} + \sqrt{\kappa_\mathcal{X}}}{\sqrt{n}} + 2\sqrt{\frac{8\log\left(4/\delta\right)}{n}}. \tag{3.6}$$

Proof The proof follows that of Li et al. (2021, Theorem 3). We decompose the expected learning risk as

$$\mathbb{E}[\ell_{\tilde{f}}] - \mathbb{E}[\ell_{f_{\mathcal{H}_\mathcal{X}}}] = \mathbb{E}[\ell_{\tilde{f}}] - \mathbb{E}_n[\ell_{\tilde{f}}] + \mathbb{E}_n[\ell_{\tilde{f}}] - \mathbb{E}_n[\ell_{f_{\mathcal{H}_\mathcal{X}}}] + \mathbb{E}_n[\ell_{f_{\mathcal{H}_\mathcal{X}}}] - \mathbb{E}[\ell_{f_{\mathcal{H}_\mathcal{X}}}]. \tag{3.7}$$

We then use Bartlett and Mendelson (2003, Theorem 8) to bound $\mathbb{E}[\ell_{\tilde{f}}] - \mathbb{E}_n[\ell_{\tilde{f}}]$ and $\mathbb{E}_n[\ell_{f_{\mathcal{H}_\mathcal{X}}}] - \mathbb{E}[\ell_{f_{\mathcal{H}_\mathcal{X}}}]$.

Lemma .1. *(Bartlett and Mendelson, 2003, Theorem 8) Let* $\left\{x_i, y_i\right\}_{i=1}^n$ *be i.i.d samples from* ρ *and let* $\mathcal{H}$ *be the space of functions mapping from* $\mathcal{X}$ *to* $\mathbb{R}$*. Denote a loss function with* $l : \mathcal{Y} \times \mathbb{R} \to [0,1]$ *and recall the learning risk function for all* $f \in \mathcal{H}$ *is* $\mathbb{E}\left[l_f\right]$*, together with the corresponding empirical risk function* $\mathbb{E}_n\left[l_f\right] = (1/n)\sum_{i=1}^n l\left(y_i, f\left(x_i\right)\right)$*. Then, for a sample of size* n*, for all* $f \in \mathcal{H}$ *and* $\delta \in (0,1)$*, with probability* $1 - \delta/2$*, we have that*

$$\mathbb{E}\left[l_f\right] \leq \mathbb{E}_n\left[l_f\right] + R_n(l \circ \mathcal{H}) + \sqrt{\frac{8\log(4/\delta)}{n}} \tag{1}$$

where $l \circ \mathcal{H} = \{(x,y) \to l(y, f(x)) - l(y,0) \mid f \in \mathcal{H}\}$.

Thus, since $\tilde{f}$ lies in the unit ball $\mathcal{B}\left(\mathcal{H}_\mathcal{X}\right)$ of $\mathcal{H}_\mathcal{X}$ by Assumption 3.6, we obtain thanks to the above lemma, with a probability at least $1 - \delta$

$$\mathbb{E}\left[\ell_{\tilde{f}}\right] - \mathbb{E}_n\left[\ell_{\tilde{f}}\right] \leq R_n\left(\ell \circ \mathcal{B}\left(\mathcal{H}_\mathcal{X}\right)\right) + \sqrt{\frac{8\log(2/\delta)}{n}}. \tag{2}$$

Then, by the Lipschitz continuity of ℓ (Assumption 3.7) and point 4 of Theorem 12 from Bartlett and Mendelson (2003), we have that

$$R_n\left(\ell \circ \mathcal{B}\left(\mathcal{H}_\mathcal{X}\right)\right) \leq 2LR_n\left(\mathcal{B}\left(\mathcal{H}_\mathcal{X}\right)\right).$$

Finally, Assumption 3.8 combined with Lemma 22 from Bartlett and Mendelson (2003) then yields

$$R_n\left(\mathcal{B}\left(\mathcal{H}_\mathcal{X}\right)\right) \leq \frac{2}{n}\sqrt{\sum_{i=1}^n \mathrm{k}_\mathcal{X}\left(x_i, x_i\right)} \leq 2\sqrt{\frac{\kappa_\mathcal{X}}{n}}. \tag{3}$$

As a consequence, we obtain

$$\mathbb{E}\left[\ell_{\tilde{f}}\right] - \mathbb{E}_n\left[\ell_{\tilde{f}}\right] \leq \frac{4L\sqrt{\kappa_\mathcal{X}}}{\sqrt{n}} + \sqrt{\frac{8\log(4/\delta)}{n}}, \tag{4}$$

and the exact same result applies to $\mathbb{E}_n[\ell_{f_{\mathcal{H}_\mathcal{X}}}] - \mathbb{E}[\ell_{f_{\mathcal{H}_\mathcal{X}}}]$, by Assumption 3.6 and the opposite side of Lemma .1.

We now focus on the last quantity to bound. Let $\mathcal{H}_{\mathrm{R}_\mathcal{X}} = \left\{ f = \sum_{i=1}^n \left[\mathrm{R}_\mathcal{X}{}^\top \gamma \right]_i \mathrm{k}_\mathcal{X}\left(\cdot, x_i\right) \mid \gamma \in \mathbb{R}^{\mathrm{m}_\mathcal{X}} \right\}$. By Assumptions 3.6 and 3.7 and Jensen's inequality we have

$$\mathbb{E}_n\left[\ell_{\tilde{f}}\right] - \mathbb{E}_n\left[\ell_{f_{\mathcal{H}_\mathcal{X}}}\right] = \frac{1}{n}\sum_{i=1}^n \ell\left(\tilde{f}(x_i), y_i\right) - \frac{1}{n}\sum_{i=1}^n \ell\left(f_{\mathcal{H}_\mathcal{X}}(x_i), y_i\right) \tag{5}$$

$$\leq \frac{1}{n}\sum_{i=1}^n \ell\left(\tilde{f}(x_i), y_i\right) + \frac{\lambda_n}{2}\left\|\tilde{f}\right\|_{\mathcal{H}_\mathcal{X}}^2 - \frac{1}{n}\sum_{i=1}^n \ell\left(f_{\mathcal{H}_\mathcal{X}}(x_i), y_i\right) \tag{6}$$

$$= \inf_{\substack{f \in \mathcal{H}_{\mathrm{R}_\mathcal{X}} \\ \|f\|_{\mathcal{H}_\mathcal{X}} \leq 1}} \frac{1}{n}\sum_{i=1}^n \ell\left(f(x_i), y_i\right) - \frac{1}{n}\sum_{i=1}^n \ell\left(f_{\mathcal{H}_\mathcal{X}}(x_i), y_i\right) + \frac{\lambda_n}{2}\|f\|_{\mathcal{H}_\mathcal{X}}^2 \tag{7}$$

$$\leq \inf_{\substack{f \in \mathcal{H}_{\mathrm{R}_\mathcal{X}} \\ \|f\|_{\mathcal{H}_\mathcal{X}} \leq 1}} \frac{L}{n}\sum_{i=1}^n \left| f\left(x_i\right) - f_{\mathcal{H}_\mathcal{X}}\left(x_i\right) \right| + \frac{\lambda_n}{2} \tag{8}$$

$$\leq L \inf_{\substack{f \in \mathcal{H}_{\mathrm{R}_\mathcal{X}} \\ \|f\|_{\mathcal{H}_\mathcal{X}} \leq 1}} \sqrt{\frac{1}{n}\sum_{i=1}^n \left| f\left(x_i\right) - f_{\mathcal{H}_\mathcal{X}}\left(x_i\right) \right|^2} + \frac{\lambda_n}{2} \tag{9}$$

$$= L \sqrt{\inf_{\substack{f \in \mathcal{H}_{\mathrm{R}_\mathcal{X}} \\ \|f\|_{\mathcal{H}_\mathcal{X}} \leq 1}} \frac{1}{n}\left\| f^X - f^X_{\mathcal{H}_\mathcal{X}} \right\|_2^2} + \frac{\lambda_n}{2}, \tag{10}$$

where, for any $f \in \mathcal{H}_\mathcal{X}$, $f^X = \left(f(x_1), \ldots, f(x_n)\right) \in \mathbb{R}^n$. Let $\tilde{f}^R = \sum_{i=1}^n \left[\mathrm{R}_\mathcal{X}{}^\top \tilde{\gamma}^R \right]_i \mathrm{k}_\mathcal{X}\left(\cdot, x_i\right)$, where $\tilde{\gamma}^R$ is a solution to

$$\inf_{\gamma \in \mathbb{R}^{\mathrm{m}_\mathcal{X}}} \frac{1}{n}\left\| \mathrm{K}_\mathrm{X} \mathrm{R}_\mathcal{X}{}^\top \gamma - f^X_{\mathcal{H}_\mathcal{X}} \right\|_2^2 + \lambda_n \gamma^\top \mathrm{R}_\mathcal{X} \mathrm{K}_\mathrm{X} \mathrm{R}_\mathcal{X}{}^\top \gamma\,. \tag{11}$$

It is easy to check that $\tilde{f}^R$ is also a solution to

$$\inf_{\substack{f \in \mathcal{H}_{\mathrm{R}_\mathcal{X}} \\ \|f\|_{\mathcal{H}_\mathcal{X}} \leq \left\|\tilde{f}^R\right\|_{\mathcal{H}_\mathcal{X}}}} \frac{1}{n}\left\| f^X - f^X_{\mathcal{H}_\mathcal{X}} \right\|_2^2. \tag{12}$$

Since we have $\|\tilde{f}^R\|_{\mathcal{H}_\mathcal{X}} \leq 1$ by Assumption 3.6, it holds

$$\inf_{\substack{f \in \mathcal{H}_{\mathrm{R}_\mathcal{X}} \\ \|f\|_{\mathcal{H}_\mathcal{X}} \leq 1}} \frac{1}{n}\left\| f^X - f^X_{\mathcal{H}_\mathcal{X}} \right\|_2^2 \leq \inf_{\substack{f \in \mathcal{H}_{\mathrm{R}_\mathcal{X}} \\ \|f\|_{\mathcal{H}_\mathcal{X}} \leq \left\|\tilde{f}^R\right\|_{\mathcal{H}_\mathcal{X}}}} \frac{1}{n}\left\| f^X - f^X_{\mathcal{H}_\mathcal{X}} \right\|_2^2 \tag{13}$$

$$= \inf_{\gamma \in \mathbb{R}^{\mathrm{m}_\mathcal{X}}} \frac{1}{n}\left\| \mathrm{K}_\mathrm{X} \mathrm{R}_\mathcal{X}{}^\top \gamma - f^X_{\mathcal{H}_\mathcal{X}} \right\|_2^2 + \lambda_n \gamma^\top \mathrm{R}_\mathcal{X} \mathrm{K}_\mathrm{X} \mathrm{R}_\mathcal{X}{}^\top \gamma\,. \tag{14}$$

As a consequence,

$$\mathbb{E}_n\left[\ell_{\tilde{f}}\right] - \mathbb{E}_n\left[\ell_{f_{\mathcal{H}_\mathcal{X}}}\right] \leq L \sqrt{\inf_{\gamma \in \mathbb{R}^{\mathrm{m}_\mathcal{X}}} \frac{1}{n}\left\| \mathrm{K}_\mathrm{X}{}^\top \gamma - f^X_{\mathcal{H}_\mathcal{X}} \right\|_2^2 + \lambda_n \gamma^\top \mathrm{R}_\mathcal{X} \mathrm{K}_\mathrm{X} \mathrm{R}_\mathcal{X}{}^\top \gamma} + \frac{\lambda_n}{2}. \tag{15}$$

Finally, since $\mathrm{R}_{\mathcal{X}}$ is a K_X-satisfiable sketch matrix, using Lemma 2 from Yang et al. (2017),

$$\mathbb{E}_n\left[\ell_{\tilde{f}}\right] - \mathbb{E}_n\left[\ell_{f_{\mathcal{H}_\mathcal{X}}}\right] \leq LC\sqrt{\lambda_n + \delta_n^2} + \frac{\lambda_n}{2}, \tag{16}$$

where $C = 1 + \sqrt{6}c$ and c is a universal constant coming from K_X-satisfiable property. The desired bound is obtained by combining Equations (3.7), (4) and (16). ■

Refined analysis in the scalar case

As said in Remark 3.11, and similarly to Li et al. (2021), we can conduct a refined analysis, leading to faster convergence rates for the generalization errors, with the following additional assumption.

Assumption .2. *There is a constant B such that, for all $f \in \mathcal{H}_k$ we have*

$$\mathbb{E}\left[f - f_{\mathcal{H}_\mathcal{X}}\right]^2 \leq B\,\mathbb{E}\left[\ell_f - \ell_{f_{\mathcal{H}_\mathcal{X}}}\right]. \tag{17}$$

It has been shown that many loss functions satisfy this assumption such as Hinge loss (Steinwart and Christmann, 2008b; Bartlett et al., 2006), truncated quadratic or sigmoid loss (Bartlett et al., 2006). Under Assumptions 3.5 to 3.9 and .2, the following result holds:

Theorem .3. *We define, for $\delta \in (0,1)$, the following sub-root function $\hat{\psi}_n$*

$$\hat{\psi}_n(r) = 2LC_1\left(\frac{2}{n}\sum_{i=1}^{n}\min\left\{b_2 r, \mu_i\right\}\right)^{1/2} + \frac{C_2}{n}\log\frac{1}{\delta}, \tag{18}$$

and let $\hat{r}^\star_{\mathcal{H}_\mathcal{X}}$ be the fixed point of $\hat{\psi}_n$, i.e., $\hat{\psi}_n\left(\hat{r}^\star_{\mathcal{H}_\mathcal{X}}\right) = \hat{r}^\star_{\mathcal{H}_\mathcal{X}}$. Then, we have for all $D > 1$ and $\delta \in (0,1)$ with probability greater than $1-\delta$,

$$\mathbb{E}\left[\ell_{\tilde{f}}\right] \leq \mathbb{E}\left[\ell_{f_{\mathcal{H}_\mathcal{X}}}\right] + \frac{D}{D-1}\left(LC\sqrt{\lambda_n + \delta_n^2} + \frac{\lambda_n}{2}\right) + \frac{12D}{B}\hat{r}^\star_{\mathcal{H}_\mathcal{X}} + \frac{2C_3}{n}\log\frac{1}{\delta}, \tag{19}$$

where C is as in Theorem 3.10 and C_1, C_2, C_3 and b_2 are some constants and $\hat{r}^\star_{\mathcal{H}_\mathcal{X}}$ can be upper bounded by

$$\hat{r}^\star_{\mathcal{H}_\mathcal{X}} \leq \min_{0\leq h\leq n}\left(b_0\frac{h}{n} + \sqrt{\frac{1}{n}\sum_{i>h}\mu_i}\right), \tag{20}$$

where B and b_0 are some constants.

Hence, we see that, in order to obtain faster learning rates than theorem 3.10 as Li et al. (2021), we need to replace δ_n^2 by $\hat{r}^{\star^2}_{\mathcal{H}_\mathcal{X}}$. However, according to the expression of $\hat{\psi}_n$ and its dependencies to non-explicit constants, it appears very difficult to prove that $\left(\frac{1}{n}\sum_{i=1}^{n}\min(\hat{r}^{\star^2}_{\mathcal{H}_\mathcal{X}}, \mu_i)\right)^{1/2} \leq \hat{r}^{\star^2}_{\mathcal{H}_\mathcal{X}}$, which is a necessary condition to prove that a sketch matrix $\mathrm{R}_{\mathcal{X}}$ is K_X-satisfiable. We still prove the above result following the proof of Theorem 4 in Li et al. (2021), and leave it as an open problem to find faster rates than δ_n.

Proof of Theorem 3.16

We first recall theorem 3.16.

Theorem 3.16. *Suppose that Assumptions 3.5 to 3.9 hold, that $\mathcal{K} = \mathrm{k}_{\mathcal{X}} M$ is a decomposable kernel with M invertible, and let C as in Theorem 3.10. Then for any $\delta \in (0,1)$ with probability at least $1-\delta$ we have*

$$\mathbb{E}\left[\ell_{\tilde{f}}\right] \leq \mathbb{E}\left[\ell_{f_{\mathcal{H}}}\right] + LC\sqrt{\lambda_n + \|M\|_{\mathrm{op}}\,\delta_n^2} + \frac{\lambda_n}{2} + 8L\sqrt{\frac{\kappa_{\mathcal{X}}\,\mathrm{Tr}\left(M\right)}{n}} + 2\sqrt{\frac{8\log\left(4/\delta\right)}{n}}. \tag{3.9}$$

Furthermore, if $\ell\left(z,y\right) = \left\|z-y\right\|_2^2/2$ and $\mathcal{Y} \subset \mathcal{B}\left(\mathbb{R}^d\right)$, with probability at least $1-\delta$ we have that

$$\begin{aligned}\mathbb{E}\left[\ell_{\tilde{f}}\right] \leq \mathbb{E}\left[\ell_{f_{\mathcal{H}}}\right] + \left(C^2 + \frac{1}{2}\right)\lambda_n + C^2\|M\|_{\mathrm{op}}\,\delta_n^2 \\ + 8\,\mathrm{Tr}\left(M\right)^{1/2}\frac{\kappa_{\mathcal{X}}\left\|M\right\|_{\mathrm{op}}^{1/2} + {\kappa_{\mathcal{X}}}^{1/2}}{\sqrt{n}} + 2\sqrt{\frac{8\log\left(4/\delta\right)}{n}}.\end{aligned} \tag{3.10}$$

Here, the proof uses the same decomposition of the excess risk (eq. (3.7)) as in single output settings. Since some works (Maurer, 2016) exist to easily extend generalisation bounds of functions in scalar-valued RKHS to functions in vector-valued RKHS, the main challenge here is to derive an approximation error for the multiple output settings. Hence, let us first state the needed intermediate results that we will prove later.

Lemma .4. *For all $f \in \mathcal{H}$, such that $\|f\|_{\mathcal{H}} \leq 1$, we have $z^\top\left(\mathrm{K_X}^{-1} \otimes M^{-1}\right)z \leq 1$, where $z = \left(f(x_1)^\top, \ldots, f(x_n)^\top\right)^\top \in \mathbb{R}^{nd}$.*

We are now equipped to state the main result that generalises Lemma 2 from Yang et al. (2017).

Lemma .5. *Let $Z^\star = \left(f^\star(x_1), \ldots, f^\star(x_n)\right)^\top \in \mathbb{R}^{n\times d}$ for any $f^\star \in \mathcal{H}$ such that $\left\|f^\star\right\|_{\mathcal{H}} \leq 1$, where $\mathcal{K} = \mathrm{k}_{\mathcal{X}} M$, and $\mathrm{R}_{\mathcal{X}} \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}}\times n}$ a $\mathrm{K_X}$- satisfiable matrix. Then we have*

$$\inf_{\Gamma\in\mathbb{R}^{\mathrm{m}_{\mathcal{X}}\times d}} \frac{1}{n}\|\mathrm{K_X}\,\mathrm{R}_{\mathcal{X}}{}^\top\Gamma M - Z^\star\|_F^2 + \lambda_n\,\mathrm{Tr}\left(\mathrm{K_X}\,\mathrm{R}_{\mathcal{X}}{}^\top\Gamma M\Gamma^\top\mathrm{R}_{\mathcal{X}}\right) \leq C^2\left(\|M\|_{\mathrm{op}}\delta_n^2 + \lambda_n\right), \tag{21}$$

where $C = 1 + \sqrt{6}c$ and c is the universal constant from Definition 3.4.

Proof We adapt the proof of Lemma 2 from Yang et al. (2017) to the multidimensional case. If we are able to find a $\Gamma \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}}\times d}$ such that

$$\frac{1}{n}\|\mathrm{K_X}\,\mathrm{R}_{\mathcal{X}}{}^\top\Gamma M - Z^\star\|_F^2 + \lambda_n\,\mathrm{Tr}\left(\mathrm{K_X}\,\mathrm{R}_{\mathcal{X}}{}^\top\Gamma M\Gamma^\top\mathrm{R}_{\mathcal{X}}\right) \leq C^2\left(\|M\|_{\mathrm{op}}\delta_n^2 + \lambda_n\right), \tag{22}$$

then in particular it also holds true for the minimizer. We recall the eigendecompositions $\frac{1}{n}K = K_{norm} = UDU^\top$ and $M = V\Delta V^\top$. Then the above problem rewrites as

$$\|D\tilde{R}_{\mathcal{X}}^\top\Gamma V\Delta - \Theta^\star\|_F^2 + \lambda_n\,\mathrm{Tr}\left(\tilde{R}_{\mathcal{X}}D\tilde{R}_{\mathcal{X}}^\top\Gamma M\Gamma\right) \leq C^2\left(\|M\|_{\mathrm{op}}\delta_n^2 + \lambda_n\right), \tag{23}$$

where $\tilde{R}_{\mathcal{X}} = \mathrm{R}_{\mathcal{X}} U$ and $\Theta^\star = \frac{1}{n^{1/2}} U^\top Z^\star V$. We can rewrite $\theta^\star = (\Theta^\star_{1:}, \dots, \Theta^\star_{n:})^\top = \frac{1}{n^{1/2}}(U^\top \otimes V^\top) z^\star$, hence $\|\left(D^{-1/2} \otimes \Delta^{-1/2}\right)\theta^\star\|_2^2 = z^{\star\top}\left(\mathrm{K_X}^{-1} \otimes M^{-1}\right) z^\star$, with $z^\star = (Z^\star_{1:}, \dots, Z^\star_{n:})^\top = (f^\star(x_1)^\top, \dots, f^\star(x_n)^\top)^\top$. By lemma .4, we have that $\|\left(D^{-1/2} \otimes \Delta^{-1/2}\right)\theta^\star\|_2 \leq 1$, and using the notation $\gamma = \left(\Gamma_{1:}, \dots, \Gamma_{s:}\right)^\top \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}} d}$, we can rewrite the above problem as finding a γ such that

$$\|\theta^\star - \left(D\tilde{R}_{\mathcal{X}}^\top \otimes \Delta V^\top\right)\gamma\|_2^2 + \lambda_n \gamma^\top \left(\tilde{R}_{\mathcal{X}} D \tilde{R}_{\mathcal{X}}^\top \otimes M\right)\gamma \leq C^2 \left(\|M\|_{\mathrm{op}} \delta_n^2 + \lambda_n\right). \tag{24}$$

As in (Yang et al., 2017), we partition vector $\theta^\star \in \mathbb{R}^{nd}$ into two sub-vectors, namely $\theta_1^\star \in \mathbb{R}^{d_n d}$ and $\theta_2^\star \in \mathbb{R}^{(n-d_n)d}$, the diagonal matrix D into two blocks $D_1 \in \mathbb{R}^{d_n \times d_n}$ and $D_2 \in \mathbb{R}^{(n-d_n)\times(n-d_n)}$ and finally, under the condition $\mathrm{m}_{\mathcal{X}} > d_n$, we let $\tilde{R}_{\mathcal{X}1} \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}} \times d_n}$ and $\tilde{R}_{\mathcal{X}2} \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}} \times (n-d_n)}$ denote the left and right block of $\tilde{R}_{\mathcal{X}}$ respectively. By the $\mathrm{K_X}$-satisfiablility of $\mathrm{R}_{\mathcal{X}}$ we have

$$\|\tilde{R}_{\mathcal{X}1}^\top \tilde{R}_{\mathcal{X}1} - I_{d_n}\|_{\mathrm{op}} \leq \frac{1}{2} \quad \text{and} \quad \|\tilde{R}_{\mathcal{X}2} D_2^{1/2}\|_{\mathrm{op}} \leq c\delta_n^2. \tag{25}$$

By the first inequality, we have that $\tilde{R}_{\mathcal{X}1}^\top \tilde{R}_{\mathcal{X}1}$ is invertible. In fact, assuming that there exists $x \in \mathbb{R}^{d_n}$ such that $\|x\|_2 = 1$ and $\tilde{R}_{\mathcal{X}1}^\top \tilde{R}_{\mathcal{X}1} X = 0$, then $\left\|(\tilde{R}_{\mathcal{X}1}^\top \tilde{R}_{\mathcal{X}1} - I_{d_n})x\right\| = 1 > \frac{1}{2}$. Then, we can define

$$\hat{\gamma} = \left(\tilde{R}_{\mathcal{X}1}\left(\tilde{R}_{\mathcal{X}1}^\top \tilde{R}_{\mathcal{X}1}\right)^{-1} D_1^{-1} \otimes V\Delta^{-1}\right)\theta_1^\star. \tag{26}$$

Hence,

$$\left\|\theta^\star - (D\tilde{R}_{\mathcal{X}}^\top \otimes \Delta V^\top)\hat{\gamma}\right\|_2^2 = \left\|\theta_1^\star - (D_1 \tilde{R}_{\mathcal{X}1}^\top \otimes \Delta V^\top)\hat{\gamma}\right\|_2^2 + \left\|\theta_2^\star - (D_2 \tilde{R}_{\mathcal{X}2}^\top \otimes \Delta V^\top)\hat{\gamma}\right\|_2^2, \tag{27}$$

and we have

$$\left\|\theta_1^\star - (D_1 \tilde{R}_{\mathcal{X}1}^\top \otimes \Delta V^\top)\hat{\gamma}\right\|_2^2 = \left\|\theta_1^\star - \left(D_1 \tilde{R}_{\mathcal{X}1}^\top \otimes \Delta V^\top\right)\left(\tilde{R}_{\mathcal{X}1}(\tilde{R}_{\mathcal{X}1}^\top \tilde{R}_{\mathcal{X}1})^{-1} D_1^{-1} \otimes V\Delta^{-1}\right)\theta_1^\star\right\|_2^2 \tag{28}$$

$$= \left\|\theta_1^\star - \left(D_1 \tilde{R}_{\mathcal{X}1}^\top \tilde{R}_{\mathcal{X}1}(\tilde{R}_{\mathcal{X}1}^\top \tilde{R}_{\mathcal{X}1})^{-1} D_1^{-1} \otimes \Delta V^\top V \Delta^{-1}\right)\theta_1^\star\right\|_2^2 \tag{29}$$

$$= \|\theta_1^\star - \theta_1^\star\|_2^2 \tag{30}$$

$$= 0, \tag{31}$$

and

$$\left\|\theta_2^\star - \left(D_2\tilde{R}_{\mathcal{X}2}^\top \otimes \Delta V^\top\right)\hat{\gamma}\right\|_2 \tag{32}$$

$$= \left\|\theta_2^\star - \left(D_2\tilde{R}_{\mathcal{X}2}^\top\tilde{R}_{\mathcal{X}1}\left(\tilde{R}_{\mathcal{X}1}^\top\tilde{R}_{\mathcal{X}1}\right)^{-1}D_1^{-1}\otimes I_p\right)\theta_1^\star\right\|_2 \tag{33}$$

$$\leq \left\|\theta_2^\star\right\|_2 + \left\|\left(D_2\tilde{R}_{\mathcal{X}2}^\top\tilde{R}_{\mathcal{X}1}\left(\tilde{R}_{\mathcal{X}1}^\top\tilde{R}_{\mathcal{X}1}\right)^{-1}D_1^{-1/2}D_1^{-1/2}\otimes\Delta^{1/2}\Delta^{-1/2}\right)\theta_1^\star\right\|_2 \tag{34}$$

$$= \left\|\theta_2^\star\right\|_2 + \left\|\left(D_2\tilde{R}_{\mathcal{X}2}^\top\tilde{R}_{\mathcal{X}1}\left(\tilde{R}_{\mathcal{X}1}^\top\tilde{R}_{\mathcal{X}1}\right)^{-1}D_1^{-1/2}\otimes\Delta^{1/2}\right)\left(D_1^{-1/2}\otimes\Delta^{-1/2}\right)\theta_1^\star\right\|_2 \tag{35}$$

$$\leq \left\|\theta_2^\star\right\|_2 + \left\|\left(D_2\tilde{R}_{\mathcal{X}2}^\top\tilde{R}_{\mathcal{X}1}\left(\tilde{R}_{\mathcal{X}1}^\top\tilde{R}_{\mathcal{X}1}\right)^{-1}D_1^{-1/2}\otimes\Delta^{1/2}\right)\right\|_{\text{op}}\left\|\left(D_1^{-1/2}\otimes\Delta^{-1/2}\right)\theta_1^\star\right\|_2 \tag{36}$$

$$= \left\|\theta_2^\star\right\|_2 + \left\|D_2\tilde{R}_{\mathcal{X}2}^\top\tilde{R}_{\mathcal{X}1}\left(\tilde{R}_{\mathcal{X}1}^\top\tilde{R}_{\mathcal{X}1}\right)^{-1}D_1^{-1/2}\right\|_{\text{op}}\left\|\Delta^{1/2}\right\|_{\text{op}}\left\|\left(D_1^{-1/2}\otimes\Delta^{-1/2}\right)\theta_1^\star\right\|_2 \tag{37}$$

$$\leq \left\|\theta_2^\star\right\|_2 + \Big(\left\|D_2^{1/2}\right\|_{\text{op}}\left\|\tilde{R}_{\mathcal{X}2}D_2^{1/2}\right\|_{\text{op}}\left\|\tilde{R}_{\mathcal{X}1}\right\|_{\text{op}}\left\|\left(\tilde{R}_{\mathcal{X}1}^\top\tilde{R}_{\mathcal{X}1}\right)^{-1}\right\|_{\text{op}} \tag{38}$$

$$\cdot\left\|D_1^{-1/2}\right\|_{\text{op}}\left\|\Delta^{1/2}\right\|_{\text{op}}\left\|\left(D_1^{-1/2}\otimes\Delta^{-1/2}\right)\theta_1^\star\right\|_2\Big). \tag{39}$$

We now bound all terms involved in (38) and (39). Since $\|\left(D^{-1/2}\otimes\Delta^{-1/2}\right)\theta^\star\|_2 \leq 1$, then $\|\left(D_1^{-1/2}\otimes\Delta^{-1/2}\right)\theta_1^\star\|_2 \leq 1$ and,

$$\|\theta_2^\star\|_2^2 = \sum_{i=1}^{d}\sum_{j=d_n+1}^{n}\left(\theta_{2_{ji}}^\star\right)^2 \tag{40}$$

$$\leq \delta_n^2\|M\|_{\text{op}}\sum_{i=1}^{d}\frac{1}{\Delta_{ii}}\sum_{j=d_n+1}^{n}\frac{n\left(\theta_{2_{ji}}^\star\right)^2}{\sigma_j(\mathrm{K_X})} \tag{41}$$

$$\leq \delta_n^2\|M\|_{\text{op}}\sum_{i=1}^{d}\sum_{j=1}^{n}\frac{n\left(\theta_{2_{ji}}^\star\right)^2}{\sigma_j(\mathrm{K_X})\Delta_{ii}} \tag{42}$$

$$= \delta_n^2\|M\|_{\text{op}}\left\|\left(D^{-1/2}\otimes\Delta^{-1/2}\right)\theta^\star\right\|_2^2 \tag{43}$$

$$\leq \delta_n^2\|M\|_{\text{op}}, \tag{44}$$

since $\sigma_j(\mathrm{K_X})/n \leq \delta_n^2$, for all $j \geq d_n + 1$ and $\Delta_{ii} \leq \|M\|_{\text{op}}$ for all $1 \leq i \leq d$. Moreover, since $\|\tilde{R}_{\mathcal{X}1}^\top\tilde{R}_{\mathcal{X}1} - I_{d_n}\|_{\text{op}} \leq \frac{1}{2}$, $\|\tilde{R}_{\mathcal{X}1}^\top\tilde{R}_{\mathcal{X}1}\|_{\text{op}} \leq \frac{3}{2}$, then $\|\tilde{R}_{\mathcal{X}1}\|_{\text{op}} \leq \sqrt{\frac{3}{2}}$. Besides, for all $x \in \mathbb{R}^{d_n}$ such that $\|x\|_2 = 1$, we have

$$|\|\tilde{R}_{\mathcal{X}1}^\top\tilde{R}_{\mathcal{X}1}x\|_2 - 1| = |\|\tilde{R}_{\mathcal{X}1}^\top\tilde{R}_{\mathcal{X}1}x\|_2 - \|x\|_2| \leq \left\|\left(\tilde{R}_{\mathcal{X}1}^\top\tilde{R}_{\mathcal{X}1} - I_{d_n}\right)x\right\|_2 \leq \frac{1}{2}, \tag{45}$$

Then, we obtain that $\|\tilde{R}_{\mathcal{X}1}^\top \tilde{R}_{\mathcal{X}1} x\|_2 - 1 \geq -\frac{1}{2}$ and then $\|\tilde{R}_{\mathcal{X}1}^\top \tilde{R}_{\mathcal{X}1} x\|_2 \geq \frac{1}{2}$, taking x the eigenvector of $\tilde{R}_{\mathcal{X}1}^\top \tilde{R}_{\mathcal{X}1}$ corresponding to its smallest eigenvalue, we obtain that $\|(\tilde{R}_{\mathcal{X}1}^\top \cdot \tilde{R}_{\mathcal{X}1})^{-1}\|_{\text{op}}^{-1} \geq \frac{1}{2}$, and finally $\|\left(\tilde{R}_{\mathcal{X}1}^\top \tilde{R}_{\mathcal{X}1}\right)^{-1}\|_{\text{op}} \leq 2$. Moreover we have

$$\|D_1^{-1/2}\|_{\text{op}} \leq \frac{1}{\delta_n}, \tag{46}$$

$$\|D_2^{1/2}\|_{\text{op}} \leq \delta_n, \tag{47}$$

$$\|\tilde{R}_{\mathcal{X}2} D_2^{1/2}\|_{\text{op}} \leq c\delta_n. \tag{48}$$

Thus,

$$\left\|\theta^\star - \left(D_2 \tilde{R}_{\mathcal{X}2}^\top \otimes \Delta V^\top\right)\hat{\gamma}\right\|_2 \leq \left(\delta_n^2 \|M\|_{\text{op}}\right)^{1/2} + \delta_n c\delta_n \left(\frac{3}{2}\right)^{1/2} 2\frac{1}{\delta_n}\|M\|_{\text{op}}^{1/2} \tag{49}$$

$$= \left(\delta_n^2 \|M\|_{\text{op}}\right)^{1/2} \left(1 + c\sqrt{6}\right) \tag{50}$$

Finally,

$$\left\|\theta^\star - \left(D\tilde{R}_{\mathcal{X}}^\top \otimes \Delta V^\top\right)\hat{\gamma}\right\|_2^2 \leq \delta_n^2 \|M\|_{\text{op}} \left(1 + c\sqrt{6}\right)^2. \tag{51}$$

Furthermore, looking into the second term,

$$\hat{\gamma}^\top \left(\tilde{R}_{\mathcal{X}} D \tilde{R}_{\mathcal{X}}^\top \otimes M\right)\hat{\gamma} = \|\left(D^{1/2}\tilde{R}_{\mathcal{X}}^\top \otimes \Delta^{1/2} V^\top\right)\hat{\gamma}\|_2^2 \tag{52}$$

$$= \|\left(D_1^{1/2}\tilde{R}_{\mathcal{X}1}^\top \otimes \Delta^{1/2} V^\top\right)\hat{\gamma}\|_2^2 + \|\left(D_2^{1/2}\tilde{R}_{\mathcal{X}2}^\top \otimes \Delta^{1/2} V^\top\right)\hat{\gamma}\|_2^2 \tag{53}$$

$$= \|\left(D_1^{-1/2} \otimes \Delta^{-1/2}\right)\theta_1^\star\|_2^2 \tag{54}$$

$$+ \|\left(D_2^{1/2}\tilde{R}_{\mathcal{X}2}^\top \tilde{R}_{\mathcal{X}1}\left(\tilde{R}_{\mathcal{X}1}^\top \tilde{R}_{\mathcal{X}1}\right)^{-1} D_1^{-1} \otimes \Delta^{-1/2}\right)\theta_1^\star\|_2^2 \tag{55}$$

$$\leq 1 + \left(\|\tilde{R}_{\mathcal{X}2} D_2^{1/2}\|_{\text{op}}^2 \|\tilde{R}_{\mathcal{X}1}\|_{\text{op}}^2 \|\left(\tilde{R}_{\mathcal{X}1}^\top \tilde{R}_{\mathcal{X}1}\right)^{-1}\|_{\text{op}}^2\right. \tag{56}$$

$$\left. \cdot \|D_1^{-1/2}\|_{\text{op}}^2 \|\left(D_1^{-1/2} \otimes \Delta^{-1/2}\right)\theta_1^\star\|_2^2\right) \tag{57}$$

$$\leq 1 + c^2\delta_n^2 \frac{3}{2} 4 \frac{1}{\delta_n^2} \tag{58}$$

$$= 1 + 6c^2 \tag{59}$$

$$= \left(1 + \sqrt{6}c\right)^2 - 2\sqrt{6}c \tag{60}$$

$$\leq \left(1 + \sqrt{6}c\right)^2. \tag{61}$$

Finally, we obtain that

$$\left\|\theta^\star - \left(D\tilde{R}_{\mathcal{X}}^\top \otimes \Delta V^\top\right)\hat{\gamma}\right\|_2^2 + \lambda_n \hat{\gamma}^\top \left(\tilde{R}_{\mathcal{X}} D \tilde{R}_{\mathcal{X}}^\top \otimes M\right)\hat{\gamma} \leq \left(1 + \sqrt{6}c\right)^2 \left(\|M\|_{\text{op}}\delta_n^2 + \lambda_n\right), \tag{62}$$

and as a conclusion

$$\inf_{\Gamma \in \mathbb{R}^{m_{\mathcal{X}} \times d}} \frac{1}{n}\|\mathrm{K_X}\,\mathrm{R}_{\mathcal{X}}{}^\top \Gamma M - Z^\star\|_F^2 + \lambda_n \operatorname{Tr}\left(\mathrm{K_X}\,\mathrm{R}_{\mathcal{X}}{}^\top \Gamma M \Gamma^\top \mathrm{R}_{\mathcal{X}}\right) \leq C^2 \left(\|M\|_{\text{op}}\delta_n^2 + \lambda_n\right), \tag{63}$$

where $C = 1 + \sqrt{6}c$. ■

Now, as for the proof of theorem 3.10, let us prove equation first inequality in theorem 3.16.

Proof For any function in $\mathcal{B}\left(\mathcal{H}\right) = \{f \in \mathcal{H} : \|f\|_{\mathcal{H}} \leq 1\}$, lemma .1 still holds, then

$$\mathbb{E}\left[\ell_f\right] \leq \mathbb{E}_n\left[\ell_f\right] + R_n(l \circ \mathcal{B}\left(\mathcal{H}\right)) + \sqrt{\frac{8\log(2/\delta)}{n}}. \tag{64}$$

Then, using Corollary 1 from Maurer (2016), we have that:

$$R_n(\ell \circ \mathcal{B}\left(\mathcal{H}\right)) \leq \sqrt{2}L\mathcal{R}_n(\mathcal{B}\left(\mathcal{H}\right)), \tag{65}$$

where

$$\mathcal{R}_n(F) = \mathbb{E}\left[\sup_{f\in F}\left|\frac{2}{n}\sum_{i=1}^{n}\sum_{j=1}^{d}\epsilon_{ij}f(x_i)_j\right| \mid x_1,\ldots,x_n\right] \tag{66}$$

$$= \mathbb{E}\left[\sup_{f\in F}\left|\frac{2}{n}\sum_{i=1}^{n}\left\langle \boldsymbol{\epsilon}_i, f(x_i)\right\rangle_{\mathbb{R}^d}\right| \mid x_1,\ldots,x_n\right], \tag{67}$$

where $\epsilon_{11},\dots,\epsilon_{np}$ are nd independent Rademacher variables, and for all $1 \le i \le n$, $\boldsymbol{\epsilon}_i = \left(\epsilon_{i1},\dots,\epsilon_{id}\right)^\top$. Hence

$$\mathcal{R}_n(\mathcal{B}\left(\mathcal{H}\right)) = \mathbb{E}\left[\sup_{\|f\|_{\mathcal{H}}\le 1}\left|\frac{2}{n}\sum_{i=1}^{n}\left\langle \boldsymbol{\epsilon}_i, f(x_i)\right\rangle_{\mathbb{R}^d}\right| \mid x_1,\dots,x_n\right] \tag{68}$$

$$= \mathbb{E}\left[\sup_{\|f\|_{\mathcal{H}}\le 1}\left|\left\langle \frac{2}{n}\sum_{i=1}^{n}\mathcal{K}_{x_i}\boldsymbol{\epsilon}_i, f\right\rangle_{\mathcal{H}}\right| \mid x_1,\dots,x_n\right] \tag{69}$$

$$\le \frac{2}{n}\,\mathbb{E}\left[\left\|\sum_{i=1}^{n}\mathcal{K}_{x_i}\boldsymbol{\epsilon}_i\right\|_{\mathcal{H}}^2 \mid x_1,\dots,x_n\right]^{1/2} \tag{70}$$

$$= \frac{2}{n}\,\mathbb{E}\left[\sum_{i,j=1}^{n}\left\langle \boldsymbol{\epsilon}_i, \mathcal{K}(x_i,x_j)\boldsymbol{\epsilon}_j\right\rangle_{\mathbb{R}^d} \mid x_1,\dots,x_n\right]^{1/2} \tag{71}$$

$$= \frac{2}{n}\,\mathbb{E}\left[\sum_{i,j=1}^{n}\mathrm{k}_{\mathcal{X}}(x_i,x_j)\left\langle \boldsymbol{\epsilon}_i, M\boldsymbol{\epsilon}_j\right\rangle_{\mathbb{R}^d} | x_1,\dots,x_n\right]^{1/2} \tag{72}$$

$$= \frac{2}{n}\left(\sum_{i,j=1}^{n}\mathrm{k}_{\mathcal{X}}(x_i,x_j)\sum_{i',j'=1}^{d}\mathbb{E}\left[M_{i'j'}\epsilon_{ii'}\epsilon_{jj'}|x_1,\dots,x_n\right]\right)^{1/2} \tag{73}$$

$$= \frac{2}{n}\left(\sum_{i=1}^{n}\mathrm{k}_{\mathcal{X}}(x_i,x_i)\sum_{i'=1}^{d}M_{i'i'}\right)^{1/2} \tag{74}$$

$$= \frac{2}{n}\left(\mathrm{Tr}\left(\mathrm{K}_{\mathrm{X}}\otimes M\right)\right)^{1/2} \tag{75}$$

$$\mathcal{R}_n(\mathcal{B}\left(\mathcal{H}\right)) \le \frac{2}{n^{1/2}}\,{\kappa_{\mathcal{X}}}^{1/2}\,\mathrm{Tr}\left(M\right)^{1/2}. \tag{76}$$

Finally, for any function $f \in \mathcal{B}\left(\mathcal{H}\right)$, for all $\delta \in (0,1)$, we have for a probability at least $1-\delta$,

$$\left|\mathbb{E}\left[\ell_f\right] - \mathbb{E}_n\left[\ell_f\right]\right| \le 4L\sqrt{\frac{2\,\kappa_{\mathcal{X}}}{n}\,\mathrm{Tr}\left(M\right)} + 2\sqrt{\frac{8\log(2/\delta)}{n}}. \tag{77}$$

Now, for the approximation error term, we proceed as in the proof of Theorem 3.10. Let $\mathcal{H}_{\mathrm{R}_\mathcal{X}} = \left\{ f = \sum_{i=1}^n \mathrm{k}_\mathcal{X}(\cdot, x_i) M \left[\mathrm{R}_\mathcal{X}{}^\top \tilde{\Gamma} \right]_i \mid \gamma \in \mathbb{R}^{\mathrm{m}_\mathcal{X} \times d} \right\}$. By Assumptions 3.6 and 3.7 and Jensen's inequality,

$$\mathbb{E}_n\left[\ell_{\tilde{f}}\right] - \mathbb{E}_n\left[\ell_{f_\mathcal{H}}\right] = \frac{1}{n}\sum_{i=1}^n \ell\left(\tilde{f}(x_i), y_i\right) - \frac{1}{n}\sum_{i=1}^n \ell\left(f_\mathcal{H}(x_i), y_i\right) \tag{78}$$

$$\leq \frac{1}{n}\sum_{i=1}^n \ell\left(\tilde{f}(x_i), y_i\right) + \frac{\lambda_n}{2}\left\|\tilde{f}\right\|_\mathcal{H}^2 - \frac{1}{n}\sum_{i=1}^n \ell\left(f_\mathcal{H}(x_i), y_i\right) \tag{79}$$

$$= \inf_{\substack{f \in \mathcal{H}_{\mathrm{R}_\mathcal{X}} \\ \|f\|_\mathcal{H} \leq 1}} \frac{1}{n}\sum_{i=1}^n \ell\left(f(x_i), y_i\right) - \frac{1}{n}\sum_{i=1}^n \ell\left(f_\mathcal{H}(x_i), y_i\right) + \frac{\lambda_n}{2}\|f\|_\mathcal{H}^2 \tag{80}$$

$$\leq \inf_{\substack{f \in \mathcal{H}_{\mathrm{R}_\mathcal{X}} \\ \|f\|_\mathcal{H} \leq 1}} \frac{L}{n}\sum_{i=1}^n \left\| f\left(x_i\right) - f_\mathcal{H}\left(x_i\right) \right\|_2 + \frac{\lambda_n}{2} \tag{81}$$

$$\leq L \inf_{\substack{f \in \mathcal{H}_{\mathrm{R}_\mathcal{X}} \\ \|f\|_\mathcal{H} \leq 1}} \sqrt{\frac{1}{n}\sum_{i=1}^n \left\| f\left(x_i\right) - f_\mathcal{H}\left(x_i\right) \right\|_2^2} + \frac{\lambda_n}{2} \tag{82}$$

$$= L \sqrt{\inf_{\substack{f \in \mathcal{H}_{\mathrm{R}_\mathcal{X}} \\ \|f\|_\mathcal{H} \leq 1}} \frac{1}{n}\left\| f^X - f_\mathcal{H}^X \right\|_F^2} + \frac{\lambda_n}{2}, \tag{83}$$

where, for any $f \in \mathcal{H}$, $f^X = \left(f(x_1), \ldots, f(x_n)\right)^\top \in \mathbb{R}^{n \times d}$. Let $\tilde{f}^R = \sum_{i=1}^n \mathrm{k}_\mathcal{X}(\cdot, x_i) M \left[\mathrm{R}_\mathcal{X}{}^\top \tilde{\Gamma}^R \right]_i$, where $\tilde{\Gamma}^R$ is a solution to

$$\inf_{\Gamma \in \mathbb{R}^{\mathrm{m}_\mathcal{X} \times d}} \frac{1}{n}\left\| \mathrm{K}_\mathrm{X} \mathrm{R}_\mathcal{X}{}^\top \Gamma M - f_\mathcal{H}^X \right\|_F^2 + \lambda_n \operatorname{Tr}\left(\mathrm{K}_\mathrm{X} \mathrm{R}_\mathcal{X}{}^\top \Gamma M \Gamma^\top \mathrm{R}_\mathcal{X} \right). \tag{84}$$

It is easy to check that $\tilde{f}^R$ is also a solution to

$$\inf_{\substack{f \in \mathcal{H}_{\mathrm{R}_\mathcal{X}} \\ \|f\|_\mathcal{H} \leq \left\|\tilde{f}^R\right\|_\mathcal{H}}} \frac{1}{n}\left\| f^X - f_\mathcal{H}^X \right\|_F^2. \tag{85}$$

Since we have $\|\tilde{f}^R\|_\mathcal{H} \leq 1$ by Assumption 3.6, it holds

$$\inf_{\substack{f \in \mathcal{H}_{\mathrm{R}_\mathcal{X}} \\ \|f\|_\mathcal{H} \leq 1}} \frac{1}{n}\left\| f^X - f_\mathcal{H}^X \right\|_F^2 \leq \inf_{\substack{f \in \mathcal{H}_{\mathrm{R}_\mathcal{X}} \\ \|f\|_\mathcal{H} \leq \left\|\tilde{f}^R\right\|_\mathcal{H}}} \frac{1}{n}\left\| f^X - f_\mathcal{H}^X \right\|_F^2 \tag{86}$$

$$= \inf_{\Gamma \in \mathbb{R}^{\mathrm{m}_\mathcal{X} \times d}} \frac{1}{n}\left\| \mathrm{K}_\mathrm{X} \mathrm{R}_\mathcal{X}{}^\top \Gamma M - f_\mathcal{H}^X \right\|_F^2 + \lambda_n \operatorname{Tr}\left(\mathrm{K}_\mathrm{X} \mathrm{R}_\mathcal{X}{}^\top \Gamma M \Gamma^\top \mathrm{R}_\mathcal{X} \right). \tag{87}$$

As a consequence, we have

$$\mathbb{E}_n[\ell_{\tilde{f}}] - \mathbb{E}_n[\ell_{f_\mathcal{H}}] \leq L \sqrt{\inf_{\Gamma \in \mathbb{R}^{\mathrm{m}_\mathcal{X} \times d}} \frac{1}{n}\| \mathrm{K}_\mathrm{X} \mathrm{R}_\mathcal{X}{}^\top \Gamma M - f_\mathcal{H}^X \|_F^2 + \lambda_n \operatorname{Tr}\left(\mathrm{K}_\mathrm{X} \mathrm{R}_\mathcal{X}{}^\top \Gamma M \Gamma^\top \mathrm{R}_\mathcal{X} \right)} + \frac{\lambda_n}{2}. \tag{88}$$

Finally, by lemma .5 and eq. (77), we obtain the result stated. ■

Furthermore, we give the proof of the second claim, i.e. the excess risk bound for kernel ridge multi-output regression.

Proof We now assume that the outputs are bounded, hence, without loss of generality, $\mathcal{Y} \subset \mathcal{B}\left(\mathbb{R}^d\right)$. First, we prove Lipschitz-continuity of the square loss under Assumptions 3.6 and 3.8. Let $g : z \in \mathcal{H}\left(\mathcal{X}\right) \mapsto \frac{1}{2}\left\|z-y\right\|_2^2$. We have that $\nabla g(z) = z - y$, and hence $\left\|\nabla g(z)\right\|_2 \leq \left\|f(x)\right\|_2 + 1$, for some $f \in \mathcal{H}$ and $x \in \mathcal{X}$. By Assumptions 3.6 and 3.8 and Cauchy-Schwartz inequality, it is easy to check that

$$\left\|f\left(x\right)\right\|_2^2 \leq \left(\kappa_{\mathcal{X}} \left\|M\right\|_{\text{op}} \left\|f(x)\right\|_2^2\right)^{1/2}, \tag{89}$$

which gives us that $\left\|f\left(x\right)\right\|_2 \leq {\kappa_{\mathcal{X}}}^{1/2} \left\|M\right\|_{\text{op}}^{1/2}$ and then $\left\|\nabla g(z)\right\|_2 \leq {\kappa_{\mathcal{X}}}^{1/2} \left\|M\right\|_{\text{op}}^{1/2} + 1$. We finally obtain that

$$\left|\ell\left(f\left(x\right), y\right) - \ell\left(f'\left(x'\right), y\right)\right| \leq \left(\kappa_{\mathcal{X}}{}^{1/2} \left\|M\right\|_{\text{op}}^{1/2} + 1\right) \left|\left|f\left(x\right) - f'\left(x'\right)\right|\right|_2. \tag{90}$$

We can then obtain the same generalisation bounds as above. Finally, looking at the approximation term,

$$\begin{aligned}
\mathbb{E}_n\left[l_{\tilde{f}}\right] - \mathbb{E}_n\left[l_{f_{\mathcal{H}}}\right] &= \frac{1}{2n}\left\|\tilde{f}^X - Y\right\|_2^2 - \frac{1}{2n}\left\|f_{\mathcal{H}}^X - Y\right\|_2^2 && (91)\\
&\leq \frac{1}{2n}\left\|\tilde{f}^X - f_{\mathcal{H}}^X\right\|_2^2 && (92)\\
&\leq \inf_{\substack{f \in \mathcal{H}_{\mathrm{R}_{\mathcal{X}}} \\ \|f\|_{\mathcal{H}} \leq 1}} \frac{1}{2n}\left\|f^X - f_{\mathcal{H}}^X\right\|_2^2 + \frac{\lambda_n}{2} && (93)\\
&\leq \inf_{\gamma \in \mathbb{R}^{\mathrm{m}\mathcal{X}}} \frac{1}{n}\left\|\mathrm{K_X}\,\mathrm{R}_{\mathcal{X}}{}^{\top}\gamma - f_{\mathcal{H}}^X\right\|_2^2 + \lambda_n \gamma^{\top}\mathrm{R}_{\mathcal{X}}\,\mathrm{K_X}\,\mathrm{R}_{\mathcal{X}}{}^{\top}\gamma + \frac{\lambda_n}{2} && (94)\\
&\leq \left(C^2 + \frac{1}{2}\right)\lambda_n + C^2\delta_n^2. && (95)
\end{aligned}$$

Here again, as in second claim of theorem 3.10, we can directly use bound (21) and then, in combination with (77), we obtain the stated second claim in theorem 3.16. ■

Finally, we here prove Lemma .4.

Proof Let $f \in \mathcal{H}$ such that $\|f\|_{\mathcal{H}} \leq 1$ and $z = \left(f(x_1)^{\top}, \ldots, f(x_n)^{\top}\right)^{\top} \in \mathbb{R}^{nd}$. We define the linear operator $S_{X,\mathcal{K}} : \mathcal{H} \to \mathbb{R}^{nd}$ such that $S_{X,\mathcal{K}}(f) = \left(f(x_1)^{\top}, \ldots, f(x_n)^{\top}\right)^{\top}$ for all $f \in \mathcal{H}$. Then for all $f \in \mathcal{H}$ and $z = \left(z_1^{\top}, \ldots, z_n^{\top}\right)^{\top} \in \mathbb{R}^{nd}$ we have

$$\left\langle S_{X,\mathcal{K}}(f), z\right\rangle_{\mathbb{R}^{nd}} = \sum_{i=1}^{n}\left\langle f(x_i), z_i\right\rangle_{\mathbb{R}^d} = \sum_{i=1}^{n}\left\langle f, \mathcal{K}_{x_i} z_i\right\rangle_{\mathcal{H}} = \left\langle f, \sum_{i=1}^{n}\mathcal{K}_{x_i} z_i\right\rangle_{\mathcal{H}} = \left\langle f, S_{X,\mathcal{K}}^{\#}(z)\right\rangle_{\mathcal{H}}. \tag{96}$$

Hence

$$z^\top \left(\mathrm{K_X}^{-1} \otimes M^{-1}\right) z = \left\langle \left(\mathrm{K_X} \otimes M\right)^{-1} S_{X,\mathcal{K}}(f), S_{X,\mathcal{K}}(f) \right\rangle_{\mathbb{R}^{nd}} \tag{97}$$

$$= \left\langle S^{\#}_{X,\mathcal{K}} \left(\left(\mathrm{K_X} \otimes M\right)^{-1} S_{X,\mathcal{K}}(f) \right), f \right\rangle_{\mathcal{H}}. \tag{98}$$

We recall the eigendecompositions of $\mathrm{K_X}$ and M

$$\mathrm{K_X} = U\left(nD\right)U^\top = \sum_{i=1}^{n} \sigma_i(\mathrm{K_X}) u_i u_i^\top \tag{99}$$

$$M = V \Delta V^\top = \sum_{j=1}^{d} \sigma_j(M) v_j v_j^\top. \tag{100}$$

Then,

$$\mathrm{K_X} \otimes M = \left(\sum_{i=1}^{n} \sigma_i(\mathrm{K_X}) u_i u_i^\top \right) \otimes \left(\sum_{j=1}^{d} \sigma_j(M) v_j v_j^\top \right) \tag{101}$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{d} \sigma_i(\mathrm{K_X}) \sigma_j(M) \left(u_i u_i^\top\right) \otimes \left(v_j v_j^\top\right) \tag{102}$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{d} \sigma_i(\mathrm{K_X}) \sigma_j(M) \left(u_i\right) \otimes \left(v_j\right) \left(u_i^\top\right) \otimes \left(v_j^\top\right) \tag{103}$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{d} \sigma_i(\mathrm{K_X}) \sigma_j(M) \left(u_i\right) \otimes \left(v_j\right) \left(\left(u_i\right) \otimes \left(v_j\right)\right)^\top, \tag{104}$$

and for all $1 \leq i, i' \leq n$ and $1 \leq j, j' \leq d$, if $\left(i, i'\right) \neq \left(j, j'\right)$, then $\left(u_i\right) \otimes \left(v_j\right)^\top \left(\left(u_{i'}\right) \otimes \left(v_{j'}\right)\right) = 0$ and otherwise $\left(u_i\right) \otimes \left(v_j\right)^\top \left(\left(u_{i'}\right) \otimes \left(v_{j'}\right)\right) = 1$. Then, this allows to show that the operator norm of a Kronecker product is the product of the operator norms and that

$$\left(\mathrm{K_X} \otimes M\right)^{-1} = \sum_{i=1}^{n} \sum_{j=1}^{d} \left(\sigma_i(\mathrm{K_X}) \sigma_j(M)\right)^{-1} \left(u_i\right) \otimes \left(v_j\right) \left(\left(u_i\right) \otimes \left(v_j\right)\right)^\top. \tag{105}$$

We define, for all $1 \leq i \leq n$ and $1 \leq j \leq d$,

$$\varphi_{ij} = \frac{1}{\sqrt{\sigma_i(\mathrm{K_X}) \sigma_j(M)}} \sum_{l=1}^{n} u_{i_l} \mathcal{K}_{x_l} v_j. \tag{106}$$

Let $\mathcal{H}_n = \text{span}\left(\left(\mathcal{K}_{x_i} v_j\right)_{1\leq i\leq n, 1\leq j\leq d}\right)$ and $\Phi_n = \text{span}\left(\left(\varphi_{ij}\right)_{1\leq i\leq n, 1\leq j\leq d}\right)$. By definition, $\Phi_n \subseteq \mathcal{H}_n$ and we show that the φ_{ij}s are orthonormal,

$$\left\langle \varphi_{ij}, \varphi_{i'j'} \right\rangle_{\mathcal{H}} = \left\langle \frac{1}{\sqrt{\sigma_i(\mathrm{K_X})\sigma_j(M)}} \sum_{l=1}^{n} u_{i_l} \mathcal{K}_{x_l} v_j, \frac{1}{\sqrt{\sigma_{i'}(\mathrm{K_X})\sigma_{j'}(M)}} \sum_{l'=1}^{n} u_{i'_{l'}} \mathcal{K}_{x_{l'}} v_{j'} \right\rangle_{\mathcal{H}_\mathcal{K}} \tag{107}$$

$$= \frac{1}{\sqrt{\sigma_i(\mathrm{K_X})\sigma_j(M)}} \frac{1}{\sqrt{\sigma_{i'}(\mathrm{K_X})\sigma_{j'}(M)}} \sum_{l,l'}^{n} u_{i_l} u_{i'_{l'}} \left\langle \mathcal{K}_{x_l} v_j, \mathcal{K}_{x_{l'}} v_{j'} \right\rangle_{\mathcal{H}_\mathcal{K}} \tag{108}$$

$$= \frac{1}{\sqrt{\sigma_i(\mathrm{K_X})\sigma_j(M)}} \frac{1}{\sqrt{\sigma_{i'}(\mathrm{K_X})\sigma_{j'}(M)}} \sum_{l,l'}^{n} u_{i_l} u_{i'_{l'}} \left\langle v_j, \mathcal{K}_{x_l, x_{l'}} v_{j'} \right\rangle_{\mathbb{R}^d} \tag{109}$$

$$= \frac{1}{\sqrt{\sigma_i(\mathrm{K_X})\sigma_j(M)}} \frac{1}{\sqrt{\sigma_{i'}(\mathrm{K_X})\sigma_{j'}(M)}} \sum_{l,l'}^{n} u_{i_l} u_{i'_{l'}} \mathrm{k}_\mathcal{X}(x_l, x_{l'}) \left\langle v_j, M v_{j'} \right\rangle_{\mathbb{R}^d} \tag{110}$$

$$= \frac{1}{\sqrt{\sigma_i(\mathrm{K_X})\sigma_j(M)}} \frac{1}{\sqrt{\sigma_{i'}(\mathrm{K_X})\sigma_{j'}(M)}} \sum_{l,l'}^{n} u_{i_l} u_{i'_{l'}} \mathrm{k}_\mathcal{X}(x_l, x_{l'}) \sigma_{j'}(M) \left\langle v_j, v_{j'} \right\rangle_{\mathbb{R}^d} \tag{111}$$

$$= 0 \quad \text{if} \quad j \neq j'. \tag{112}$$

Otherwise, if $j = j'$,

$$\left\langle \varphi_{ij}, \varphi_{i'j} \right\rangle_{\mathcal{H}} = \frac{1}{\sqrt{\sigma_i(\mathrm{K_X})}} \frac{1}{\sqrt{\sigma_{i'}(\mathrm{K_X})}} \sum_{l,l'}^{n} u_{i_l} u_{i'_{l'}} \mathrm{k}_\mathcal{X}(x_l, x_{l'}) \tag{113}$$

$$= \frac{1}{\sqrt{\sigma_i(\mathrm{K_X})}} \frac{1}{\sqrt{\sigma_{i'}(\mathrm{K_X})}} \left\langle \mathrm{K_X} u_i, u_{i'} \right\rangle_{\mathbb{R}^n} \tag{114}$$

$$= \frac{1}{\sqrt{\sigma_i(\mathrm{K_X})}} \frac{1}{\sqrt{\sigma_{i'}(\mathrm{K_X})}} n\sigma_i(\mathrm{K_X}) \left\langle u_i, u_{i'} \right\rangle_{\mathbb{R}^n} \tag{115}$$

$$= 0 \quad \text{if} \quad i \neq i'. \tag{116}$$

Hence, $\left\langle \varphi_{ij}, \varphi_{i'j'} \right\rangle_{\mathcal{H}} = 0$ if $\left(i, i'\right) \neq \left(j, j'\right)$ and if $\left(i, i'\right) = \left(j, j'\right)$,

$$\left\langle \varphi_{ij}, \varphi_{ij} \right\rangle_{\mathcal{H}} = 1. \tag{117}$$

Finally, $\Phi_n \subseteq \mathcal{H}_n$ and $\dim\left(\Phi_n\right) = nd = \dim\left(\mathcal{H}_n\right)$, hence $\left(\varphi_{ij}\right)_{1\leq i\leq n, 1\leq j\leq d}$ yields an orthonormal basis of $\mathcal{H}_n$. As a consequence, all $f \in \mathcal{H}$ can be decomposed as $f = f_1 + f_2$, with $f_1 \in \mathcal{H}_n$ and $f_2 \in \mathcal{H}_n^{\perp}$. Thus, for all $y \in \mathbb{R}^d$, y can be written as $y = \sum_{j=1}^d y_j v_j$ and

$$\begin{aligned}
\left\langle S_{X,\mathcal{K}}(f), z\right\rangle_{\mathbb{R}^{nd}} &= \sum_{i=1}^n \left\langle f(x_i), z_i\right\rangle_{\mathbb{R}^d} && (118)\\
&= \sum_{i=1}^n \sum_{j=1}^d z_{i_j} \left\langle f(x_i), v_j\right\rangle_{\mathbb{R}^d} && (119)\\
&= \sum_{i=1}^n \sum_{j=1}^d z_{i_j} \left\langle f, \mathcal{K}_{x_i} v_j\right\rangle_{\mathcal{H}} && (120)\\
&= \sum_{i=1}^n \sum_{j=1}^d z_{i_j} \left\langle f_1, \mathcal{K}_{x_i} v_j\right\rangle_{\mathcal{H}} + \sum_{i=1}^n \sum_{j=1}^d z_{i_j} \left\langle f_2, \mathcal{K}_{x_i} v_j\right\rangle_{\mathcal{H}} && (121)\\
&= \sum_{i=1}^n \sum_{j=1}^d z_{i_j} \left\langle f_1, \mathcal{K}_{x_i} v_j\right\rangle_{\mathcal{H}} && (122)\\
&= \left\langle S_{X,\mathcal{K}}(f_1), z\right\rangle_{\mathbb{R}^{nd}}. && (123)
\end{aligned}$$

Hence, let $f \in \mathcal{H}$ such that $\|f\|_{\mathcal{H}} \leq 1$, written as $f = \sum_{i=1}^n \sum_{j=1}^d f_{ij}\varphi_{ij} + f^{\perp}$, with $f_{ij} \in \mathbb{R}$ for all $1 \leq i \leq n$ and $1 \leq j \leq d$ and such that $\sum_{i=1}^n \sum_{j=1}^d f_{ij}^2 \leq 1$ and $f^{\perp} \in \mathcal{H}_n^{\perp}$ such that $\|f^{\perp}\|_{\mathcal{H}} \leq 1$ (since $\|f\|_{\mathcal{H}} = \sum_{i=1}^n \sum_{j=1}^d f_{ij}^2 + \|f^{\perp}\|_{\mathcal{H}} \leq 1$), we have that

$$S_{X,\mathcal{K}}(f) = \sum_{i=1}^n \sum_{j=1}^d f_{ij} S_{X,\mathcal{K}}\left(\varphi_{ij}\right), \tag{124}$$

and, for all $1 \leq l \leq n$,

$$\begin{aligned}
\varphi_{ij}(x_l) &= \frac{1}{\sqrt{\sigma_i(\mathrm{K_X})\sigma_j(M)}} \sum_{l'=1}^n u_{i_{l'}} \mathrm{k}_{\mathcal{X}}(x_{l'}, x_l) M v_j && (125)\\
&= \sqrt{\frac{\sigma_j(M)}{\sigma_i(\mathrm{K_X})}}\, \mathrm{K_X}_{l:}^{\top} u_i v_j, && (126)
\end{aligned}$$

and then

$$S_{X,\mathcal{K}}\left(\varphi_{ij}\right) = \sqrt{\frac{\sigma_j(M)}{\sigma_i(\mathrm{K_X})}} \left(\mathrm{K_X}\, u_i\right) \otimes v_j = \sqrt{\sigma_i(\mathrm{K_X})\sigma_j(M)}\, u_i \otimes v_j. \tag{127}$$

Finally,

$$S_{X,\mathcal{K}}(f) = \sum_{i=1}^n \sum_{j=1}^d f_{ij} \left(\sigma_i(\mathrm{K_X})\sigma_j(M)\right)^{1/2} u_i \otimes v_j. \tag{128}$$

Besides,

$$\left(\mathrm{K_X}\otimes M\right)^{-1} S_{X,\mathcal{K}}(f) = \left(\sum_{i=1}^{n}\sum_{j=1}^{d}\left(\sigma_i(\mathrm{K_X})\sigma_j(M)\right)^{-1}\left(u_i\right)\otimes\left(v_j\right)\left(\left(u_i\right)\otimes\left(v_j\right)\right)^{\top}\right) \tag{129}$$

$$\times\left(\sum_{i'=1}^{n}\sum_{j'=1}^{d} f_{i'j'}\left(\sigma_{i'}(\mathrm{K_X})\sigma_{j'}(M)\right)^{1/2} u_{i'}\otimes v_{j'}\right) \tag{130}$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{d} f_{ij}\left(\sigma_i(\mathrm{K_X})\sigma_j(M)\right)^{-1/2} u_i\otimes v_j. \tag{131}$$

Then,

$$S^{\#}_{X,\mathcal{K}}\left(\left(\mathrm{K_X}\otimes M\right)^{-1} S_{X,\mathcal{K}}(f)\right) = \sum_{i=1}^{n}\sum_{j=1}^{d} f_{ij}\left(\sigma_i(\mathrm{K_X})\sigma_j(M)\right)^{-1/2} S^{\#}_{X,\mathcal{K}}\left(u_i\otimes v_j\right) \tag{132}$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{d} f_{ij}\left(\sigma_i(\mathrm{K_X})\sigma_j(M)\right)^{-1/2}\sum_{i'=1}^{n}\mathcal{K}_{x_i}(u_{i_{i'}}v_j), \tag{133}$$

and finally,

$$\left\langle S^{\#}_{X,\mathcal{K}}\left(\left(\mathrm{K_X}\otimes M\right)^{-1} S_{X,\mathcal{K}}(f)\right), f\right\rangle_{\mathcal{H}} \tag{134}$$

$$= \sum_{i,i'=1}^{n}\sum_{j,j'=1}^{d}\sum_{l=1}^{n} f_{ij}f_{i'j'}\left(\sigma_i(\mathrm{K_X})\sigma_j(M)\right)^{-1/2} u_{i_l}\left\langle\mathcal{K}_{x_l}v_j, \varphi_{i'j'}\right\rangle_{\mathcal{H}} \tag{135}$$

$$= \sum_{i,i'=1}^{n}\sum_{j,j'=1}^{d} f_{ij}f_{i'j'}\left\langle\left(\sigma_i(\mathrm{K_X})\sigma_j(M)\right)^{-1/2}\sum_{l=1}^{n} u_{i_l}\mathcal{K}_{x_l}v_j, \varphi_{i'j'}\right\rangle_{\mathcal{H}} \tag{136}$$

$$= \sum_{i,i'=1}^{n}\sum_{j,j'=1}^{d} f_{ij}f_{i'j'}\left\langle\varphi_{ij}, \varphi_{i'j'}\right\rangle_{\mathcal{H}_{\mathcal{K}}} \tag{137}$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{d} f_{ij}^2 \tag{138}$$

$$\leq 1. \tag{139}$$

Thus, we do have the ellipse constraint

$$\|\left(\mathrm{K_X}^{-1/2}\otimes M^{-1/2}\right)z\|_2 \leq 1. \tag{140}$$

■

Proof of Theorem 3.19

We first recall theorem 3.19.

Theorem 3.19. *Let $\mathrm{R}_{\mathcal{X}}$ be a p-sparsified sketching matrix. Then, there are some universal constants $C_0, C_1 > 0$ and a constant $c(p)$, increasing with p, such that for $\mathrm{m}_{\mathcal{X}} \geq \max\left(C_0 d_n/p^2, \delta_n^2 n\right)$ and with a probability at least $1 - C_1 e^{-\mathrm{m}_{\mathcal{X}} c(p)}$, the sketch $\mathrm{R}_{\mathcal{X}}$ is K_{X}-satisfiable for $c = \frac{2}{\sqrt{p}}\left(1 + \sqrt{\log\left(5\right)}\right) + 1$.*

First claim of K_{X}-satisfiability. Let us now prove the first claim (l.h.s. of eq. (3.4)) of the K_{X}-satisfiability for p-SR and p-SG sketches. It is articulated around the following two lemmas.

Lemma .6. *Let $M \in \mathbb{R}^{d\times d}$ be a symmetric matrix, $\varepsilon \in (0,1)$, and $\mathcal{C}_\varepsilon$ be an ε-cover of $\mathcal{B}^d$. Then we have*

$$\left\|M\right\|_{\mathrm{op}} \leq \frac{1}{1-2\varepsilon-\varepsilon^2} \sup_{v\in\mathcal{C}_\varepsilon} \left|\langle v, Mv\rangle\right|. \tag{141}$$

Proof Let M, ε and $\mathcal{C}_\varepsilon$ as in Lemma .6. Let $u \in \mathcal{B}^d$. By definition, there exist $v \in \mathcal{C}_\varepsilon$ and $w \in \mathcal{B}^d$ such that $u = v + \varepsilon w$. We thus have

$$\langle u, Mu\rangle = \langle v, Mv\rangle + 2\varepsilon\langle v, Mw\rangle + \varepsilon^2\langle w, Mw\rangle. \tag{142}$$

Taking the supremum on both sides of (142) we obtain

$$\begin{aligned}
\sup_{u\in\mathcal{B}^d} |\langle u, Mu\rangle| &= \sup_{v\in\mathcal{C}_\varepsilon, w\in\mathcal{B}^d} \left(|\langle v, Mv\rangle| + 2\varepsilon|\langle v, Mw\rangle| + \varepsilon^2|\langle w, Mw\rangle|\right) && (143)\\
&\leq \sup_{v\in\mathcal{C}_\varepsilon} |\langle v, Mv\rangle| + 2\varepsilon \sup_{v\in\mathcal{C}_\varepsilon, w\in\mathcal{B}^d} |\langle v, Mw\rangle| + \varepsilon^2 \sup_{w\in\mathcal{B}^d} |\langle w, Mw\rangle| && (144)\\
&\leq \sup_{v\in\mathcal{C}_\varepsilon} |\langle v, Mv\rangle| + 2\varepsilon \sup_{v'\in\mathcal{B}^d, w\in\mathcal{B}^d} |\langle v', Mw\rangle| + \varepsilon^2\|M\|_{\mathrm{op}} && (145)\\
&= \sup_{v\in\mathcal{C}_\varepsilon} |\langle v, Mv\rangle| + \left(2\varepsilon + \varepsilon^2\right)\|M\|_{\mathrm{op}}, && (146)
\end{aligned}$$

or again

$$\|M\|_{\mathrm{op}} \leq \frac{1}{1-2\varepsilon-\varepsilon^2} \sup_{v\in\mathcal{C}_\varepsilon} \left|\langle v, Mv\rangle\right|. \tag{147}$$

■

Lemma .7. *Let $\mathrm{R}_{\mathcal{X}} \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}}\times n}$ be a p-SR or a p-SG sketch. Let $v \in \mathcal{B}^n$, then for every $t > 0$, we have*

$$\mathbb{P}\left\{\left|\|\mathrm{R}_{\mathcal{X}} v\|_2^2 - \|v\|_2^2\right| > \frac{4}{p}\sqrt{\frac{2t}{\mathrm{m}_{\mathcal{X}}}} + \frac{4t}{\mathrm{m}_{\mathcal{X}} p}\right\} \leq 2e^{-t}. \tag{148}$$

Proof The proof of Lemma .7 is largely adapted from the proof of Theorem 2.13 in Boucheron et al. (2013). Let $\mathrm{R}_{\mathcal{X}} \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}}\times n}$ be a p-SR or a p-SG sketch, and $v \in \mathcal{B}^n$. It is easy to check that for all $i \leq \mathrm{m}_{\mathcal{X}}$ we have $\mathbb{E}\left[\left[\mathrm{R}_{\mathcal{X}} v\right]_i^2\right] = \frac{1}{\mathrm{m}_{\mathcal{X}}}\|v\|_2^2$, such that

$$\left|\|\mathrm{R}_{\mathcal{X}} v\|_2^2 - \|v\|_2^2\right| = \left|\sum_{i=1}^{\mathrm{m}_{\mathcal{X}}}\left(\left[\mathrm{R}_{\mathcal{X}} v\right]_i^2 - \frac{1}{\mathrm{m}_{\mathcal{X}}}\|v\|_2^2\right)\right|. \tag{149}$$

The proof then consists in applying Bernstein's inequality (Boucheron et al., 2013, Theorem 2.10) to the random variables $\left[\mathrm{R}_{\mathcal{X}} v\right]_i^2$. We now have to find some constants ν and c such that $\sum_{i=1}^{\mathrm{m}_{\mathcal{X}}} \mathbb{E}\left[\left[\mathrm{R}_{\mathcal{X}} v\right]_i^4\right] \leq \nu$ and

$$\sum_{i=1}^{s} \mathbb{E}\left[\left[\mathrm{R}_{\mathcal{X}} v\right]_i^{2q}\right] \leq \frac{q!}{2} \nu c^{q-2} \quad \text{for all } q \geq 3. \tag{150}$$

From (3.14) and (3.15), it is easy to check that the $\mathrm{R}_{\mathcal{X}ij}$ are independent and $1/(\mathrm{m}_{\mathcal{X}} p)$ sub-Gaussian. Then, for all $\lambda \in \mathbb{R}$, we have

$$\mathbb{E}\left[\exp\left(\lambda\left[\mathrm{R}_{\mathcal{X}} v\right]_i\right)\right] = \mathbb{E}\left[\exp\left(\lambda \sum_{j=1}^{n} \mathrm{R}_{\mathcal{X}ij} v_j\right)\right] \tag{151}$$

$$= \prod_{j=1}^{n} \mathbb{E}\left[\exp\left(\lambda \mathrm{R}_{\mathcal{X}ij} v_j\right)\right] \tag{152}$$

$$\leq \exp\left(\frac{\lambda^2}{2\,\mathrm{m}_{\mathcal{X}} p} \|v\|_2^2\right) \tag{153}$$

$$\leq \exp\left(\frac{\lambda^2}{2\,\mathrm{m}_{\mathcal{X}} p}\right). \tag{154}$$

The random variable $\left[\mathrm{R}_{\mathcal{X}} v\right]_i$ is therefore $1/(\mathrm{m}_{\mathcal{X}} p)$ sub-Gaussian, and Theorem 2.1 from Boucheron et al. (2013) yields that for every integer $q \geq 2$ it holds

$$\mathbb{E}\left[\left[\mathrm{R}_{\mathcal{X}} v\right]_i^{2q}\right] \leq \frac{q!}{2} 4 \left(\frac{2}{\mathrm{m}_{\mathcal{X}} p}\right)^q \leq \frac{q!}{2} \left(\frac{4}{\mathrm{m}_{\mathcal{X}} p}\right)^q. \tag{155}$$

Choosing $q = 2$, we obtain

$$\sum_{i=1}^{\mathrm{m}_{\mathcal{X}}} \mathbb{E}\left[\left[\mathrm{R}_{\mathcal{X}} v\right]_i^4\right] \leq \sum_{i=1}^{\mathrm{m}_{\mathcal{X}}} \left(\frac{4}{\mathrm{m}_{\mathcal{X}} p}\right)^2 = \frac{16}{\mathrm{m}_{\mathcal{X}} p^2}, \tag{156}$$

such that we can choose $\nu = 16/(\mathrm{m}_{\mathcal{X}} p^2)$ and $c = 4/(\mathrm{m}_{\mathcal{X}} p)$. Applying Theorem 2.10 from Boucheron et al. (2013) to the random variables $[\mathrm{R}_{\mathcal{X}} v]_i^2$ finally gives that for any $t > 0$ it holds

$$\mathbb{P}\left\{\left|\|\mathrm{R}_{\mathcal{X}} v\|_2^2 - \|v\|_2^2\right| > \frac{4}{p}\sqrt{\frac{2t}{\mathrm{m}_{\mathcal{X}}}} + \frac{4t}{\mathrm{m}_{\mathcal{X}} p}\right\} \leq 2e^{-t}. \tag{157}$$

■

Proof [**Proof of the first claim of the** K_{X}**-satisfiability.**] Let $\mathrm{K}_{\mathrm{X}} \in \mathbb{R}^{n \times n}$ be a Gram matrix, and $\mathrm{R}_{\mathcal{X}} \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}} \times n}$ be a p-SR or a p-SG sketch. Recall that we want to prove that there exists $c_0 > 0$ such that

$$\mathbb{P}\left\{\left\|U_1^\top \mathrm{R}_{\mathcal{X}}{}^\top \mathrm{R}_{\mathcal{X}} U_1 - I_{d_n}\right\|_{\mathrm{op}} > \frac{1}{2}\right\} \leq 2e^{-c_0 \mathrm{m}_{\mathcal{X}}}, \tag{158}$$

where $\mathrm{K}_X/n = UDU^\top$ is the SVD of K_X, and $U_1 \in \mathbb{R}^{n\times d_n}$ contains the left part of U. Let $\varepsilon \in (0,1)$, and $\mathcal{C}_\varepsilon = \{v^1,\dots,v^{\mathcal{N}_\varepsilon}\}$ be an ε-cover of $\mathcal{B}^{d_n}$. We know that such a covering exists with cardinality $\mathcal{N}_\varepsilon \leq \left(1+\frac{2}{\varepsilon}\right)^{d_n}$, see e.g., Matoušek (2013). Let $Q = U_1^\top \mathrm{R}_\mathcal{X}^\top \mathrm{R}_\mathcal{X} U_1 - I_{d_n}$, applying Lemma .6, we have

$$\mathbb{P}\left\{\|Q\|_{\mathrm{op}} > \frac{1}{2}\right\} \leq \mathbb{P}\left\{\sup_{i\leq\mathcal{N}_\varepsilon}\left|\langle v^i, Qv^i\rangle\right| > \frac{1-2\varepsilon-\varepsilon^2}{2}\right\} \tag{159}$$

$$\leq \sum_{i\leq\mathcal{N}_\varepsilon}\mathbb{P}\left\{\left|\langle v^i, Qv^i\rangle\right| > \frac{1-2\varepsilon-\varepsilon^2}{2}\right\} \tag{160}$$

$$= \sum_{i\leq\mathcal{N}_\varepsilon}\mathbb{P}\left\{\left|\|\mathrm{R}_\mathcal{X} w^i\|_2^2 - \|w^i\|_2^2\right| > \frac{1-2\varepsilon-\varepsilon^2}{2}\right\}, \tag{161}$$

where $w^i = U_1 v^i \in \mathcal{B}^n$. Now, by Lemma .7, for any $w \in \mathcal{B}^n$, we have

$$\mathbb{P}\left\{\left|\|\mathrm{R}_\mathcal{X} w\|_2^2 - \|w\|_2^2\right| > \frac{4}{p}\sqrt{\frac{2t}{\mathrm{m}_\mathcal{X}}} + \frac{4t}{\mathrm{m}_\mathcal{X} p}\right\} \leq 2e^{-t}. \tag{162}$$

Let $\mathrm{m}_\mathcal{X} \geq 32t/(\alpha^2 p^2)$, for some $\alpha \leq 1$. Then, we have $\frac{4}{p}\sqrt{\frac{2t}{\mathrm{m}_\mathcal{X}}} + \frac{4t}{\mathrm{m}_\mathcal{X} p} \leq \alpha + \frac{\alpha^2 p}{8} \leq 2\alpha$, and therefore

$$\mathbb{P}\left\{\left|\|\mathrm{R}_\mathcal{X} w\|_2^2 - \|w\|_2^2\right| > 2\alpha\right\} \leq 2e^{-t}. \tag{163}$$

If we take $\alpha = (1-2\varepsilon-\varepsilon^2)/4$, we obtain

$$\mathbb{P}\left\{\left|\|\mathrm{R}_\mathcal{X} w\|_2^2 - \|w\|_2^2\right| > \frac{1-2\varepsilon-\varepsilon^2}{2}\right\} \leq 2e^{-t} \tag{164}$$

as long as $\mathrm{m}_\mathcal{X} \geq \frac{512t}{p^2(1-2\varepsilon-\varepsilon^2)^2}$. Now, let $t = \frac{p^2(1-2\varepsilon-\varepsilon^2)^2}{1024}\mathrm{m}_\mathcal{X} + \log(\mathcal{N}_\varepsilon)$, and $\mathrm{m}_\mathcal{X} \geq 1024 \cdot \frac{\log(1+2/\varepsilon)}{p^2(1-2\varepsilon-\varepsilon^2)^2} d_n$. We do have

$$\frac{512t}{p^2(1-2\varepsilon-\varepsilon^2)^2} = \frac{\mathrm{m}_\mathcal{X}}{2} + \frac{512}{p^2(1-2\varepsilon-\varepsilon^2)^2}\log(\mathcal{N}_\varepsilon) \leq \frac{\mathrm{m}_\mathcal{X}}{2} + \frac{\mathrm{m}_\mathcal{X}}{2} = \mathrm{m}_\mathcal{X}, \tag{165}$$

such that

$$\mathbb{P}\left\{\left|\|\mathrm{R}_\mathcal{X} w\|_2^2 - \|w\|_2^2\right| > \frac{1-2\varepsilon-\varepsilon^2}{2}\right\} \leq 2e^{-t} = \frac{2e^{-c_0\mathrm{m}_\mathcal{X}}}{\mathcal{N}_\varepsilon}, \tag{166}$$

where $c_0 = \frac{p^2(1-2\varepsilon-\varepsilon^2)}{1024}$. Plugging this result into (161), we get that as soon as $\mathrm{m}_\mathcal{X} \geq 1024\frac{\log(1+2/\varepsilon)}{p^2(1-2\varepsilon-\varepsilon^2)^2} d_n$ it holds

$$\mathbb{P}\left\{\|Q\|_{\mathrm{op}} > \frac{1}{2}\right\} \leq 2e^{-c_0\mathrm{m}_\mathcal{X}}. \tag{167}$$

Finally, we can tune ε to optimize the lower bound on $\mathrm{m}_\mathcal{X}$. If we take $\varepsilon = 0.1$, we obtain $\mathrm{m}_\mathcal{X} \geq 5120 d_n/p^2$, and $c_0 \geq p^2/2560$. ■

Second claim of K_{X}-satisfiability.

We now turn to the proof of the second claim (r.h.s. of eq. (3.4)) of the K_{X}-satisfiability for p-SR and p-SG sketches. It builds upon the following two intermediate results, about the concentration of Lipschitz functions of Rademacher or Gaussian random variables.

Lemma .8. *Let $K > 0$, and let $X_1, \dots, X_n$ be independent real random variables with $\left|X_i\right| \leq K$ for all $1 \leq i \leq n$. Let $F : \mathbb{R}^n \to \mathbb{R}$ be a L-Lipschitz convex function. Then, there exist $C, c > 0$ such that for any λ one has*

$$\mathbb{P}\{|F(X) - \mathbb{E}\,F(X)| \geq K\lambda\} \leq C' \exp\left(-c'\lambda^2/L^2\right). \tag{168}$$

Lemma .9. *Let $X_1, \dots, X_n$ be i.i.d. standard Gaussian random variables. Let $F : \mathbb{R}^n \to \mathbb{R}$ be a L-Lipschitz function. Then, there exist $C, c > 0$ such that for any λ one has*

$$\mathbb{P}\{|F(X) - \mathbb{E}\,F(X)| \geq \lambda\} \leq C' \exp\left(-c'\lambda^2/L^2\right). \tag{169}$$

The above two lemmas are taken from Tao (2012), see Theorems 2.1.12 and 2.1.13 therein, but are actually well-known results in the literature. In particular, Lemma .8 is adapted from Talagrand's inequality (Talagrand, 1995), while Lemma .9 is stated as Theorem 5.6 in Boucheron et al. (2013), with explicit constants. We however choose the writing by Tao (2012) in order to be consistent with the Rademacher case.

Remark .10. *Note that thanks to lemma .8, we are even able to prove K_{X}-satisfiability for any sketch matrix $\mathrm{R}_{\mathcal{X}}$ whose entries are i.i.d. centered and reduced bounded random variables.*

Proof [**Proof of the second claim of the K_{X}-satisfiability.**] Let $\mathrm{K}_{\mathrm{X}} \in \mathbb{R}^{n\times n}$ be a Gram matrix, and $\mathrm{R}_{\mathcal{X}} \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}}\times n}$ be a p-SR or a p-SG sketch. Recall that we want to prove that there exist positive constants $c, c_1, c_2 > 0$ such that

$$\mathbb{P}\left\{\left\|\mathrm{R}_{\mathcal{X}}\, U_2 D_2^{1/2}\right\|_{\mathrm{op}} > c\delta_n\right\} \leq c_1 e^{-c_2\,\mathrm{m}_{\mathcal{X}}}, \tag{170}$$

where $\mathrm{K}_{\mathrm{X}}/n = UDU^\top$ is the SVD of K_{X}, $U_2 \in \mathbb{R}^{n\times(n-d_n)}$ is the right part of U, and $D_2 \in \mathbb{R}^{(n-d_n)\times(n-d_n)}$ is the right bottom part of D. Note that we have $\mathrm{R}_{\mathcal{X}}\, U_2 D_2^{1/2} = \mathrm{R}_{\mathcal{X}}\, U\bar{D}^{1/2}$, where $\bar{D} = \mathrm{diag}\left(0_{d_n}, D_2\right) \in \mathbb{R}^{n\times n}$. Following Yang et al. (2017), we have

$$\left\|\mathrm{R}_{\mathcal{X}}\, U\bar{D}^{1/2}\right\|_{\mathrm{op}} = \sup_{u\in\mathcal{B}^{\mathrm{m}_{\mathcal{X}}}, v\in\mathcal{E}} |\langle u, \mathrm{R}_{\mathcal{X}}\, v\rangle|, \tag{171}$$

where $\mathcal{E} = \left\{v \in \mathbb{R}^n : \exists\, w \in \mathcal{S}^{n-1}, v = U\bar{D}^{1/2}w\right\}$. Now, let $u^1, \dots u^{\mathcal{N}}$ be a 1/2-cover of $\mathcal{B}^{\mathrm{m}_{\mathcal{X}}}$. We know that such a covering exists with cardinality $\mathcal{N} \leq 5^{\mathrm{m}_{\mathcal{X}}}$. We then have

$$\begin{aligned}
\left\|\mathrm{R}_{\mathcal{X}}\, U\bar{D}^{1/2}\right\|_{\mathrm{op}} &= \sup_{u\in\mathcal{B}^{\mathrm{m}_{\mathcal{X}}}, v\in\mathcal{E}} |\langle u, \mathrm{R}_{\mathcal{X}}\, v\rangle| && (172)\\
&\leq \max_{i\leq\mathcal{N}} \sup_{v\in\mathcal{E}} \left|\left\langle u^i, \mathrm{R}_{\mathcal{X}}\, v\right\rangle\right| + \frac{1}{2} \sup_{u\in\mathcal{B}^{\mathrm{m}_{\mathcal{X}}}, v\in\mathcal{E}} |\langle u, \mathrm{R}_{\mathcal{X}}\, v\rangle| && (173)\\
&= \max_{i\leq\mathcal{N}} \sup_{v\in\mathcal{E}} \left|\left\langle u^i, \mathrm{R}_{\mathcal{X}}\, v\right\rangle\right| + \frac{1}{2} \left\|\mathrm{R}_{\mathcal{X}}\, U\bar{D}^{1/2}\right\|_{\mathrm{op}}, && (174)
\end{aligned}$$

and rearranging implies that

$$\left\|\mathrm{R}_{\mathcal{X}}\, U\bar{D}^{1/2}\right\|_{\mathrm{op}} \leq 2\max_{i\leq\mathcal{N}}\sup_{v\in\mathcal{E}}\left|\left\langle u^i, \mathrm{R}_{\mathcal{X}}\, v\right\rangle\right|. \tag{175}$$

Hence, for every $c > 0$ we have

$$\mathbb{P}\left(\left\|\mathrm{R}_{\mathcal{X}}\, U_2 D_2^{1/2}\right\|_{\mathrm{op}} > c\delta_n\right) \leq \mathbb{P}\left(\max_{i\leq\mathcal{N}}\sup_{v\in\mathcal{E}}\left|\left\langle u^i, \mathrm{R}_{\mathcal{X}}\, v\right\rangle\right| > \frac{c}{2}\delta_n\right) \tag{176}$$

$$\leq \sum_{i\leq\mathcal{N}} \mathbb{P}\left\{\sup_{v\in\mathcal{E}}\left|\left\langle u^i, \mathrm{R}_{\mathcal{X}}\, v\right\rangle\right| > \frac{c}{2}\delta_n\right\}. \tag{177}$$

Now, recall that

$$\mathrm{R}_{\mathcal{X}} = \frac{1}{\sqrt{\mathrm{m}_{\mathcal{X}}\, p}}\, B \circ R, \tag{178}$$

where $B \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}}\times n}$ is filled with i.i.d. Bernoulli random variables with parameter p, $R \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}}\times n}$ is filled with i.i.d. Rademacher or Gaussian random variables for p-SR and p-SG sketches respectively, and $\circ$ denotes the Hadamard (termwise) matrix product. The next step of the proof consists of controlling the right-hand side of eq. (177) by showing that, conditionally on B, we have Lipschitz functions of Rademacher or Gaussian random variables, whose deviations can be bounded using Lemmas .8 and .9. Therefore, from now on we assume B to be fixed, and only consider the randomness with respect to R. Let $u \in \mathcal{B}^{\mathrm{m}_{\mathcal{X}}}$, and define $F\colon \mathbb{R}^{\mathrm{m}_{\mathcal{X}}\times n} \to \mathbb{R}$ as

$$F(R) = \frac{1}{\sqrt{\mathrm{m}_{\mathcal{X}}\, p}}\sup_{v\in\mathcal{E}}\left|\left\langle u, (B\circ R)v\right\rangle\right|. \tag{179}$$

It is direct to check that F is a convex function. Moreover, we have

$$\sqrt{\mathrm{m}_{\mathcal{X}}\, p}\, F(R) = \sup_{v\in\mathcal{E}}|\langle u, (B\circ R)v\rangle| \tag{180}$$

$$= \sup_{v\in\mathcal{S}^{n-1}}|\langle u, (B\circ R)U\bar{D}^{1/2}v\rangle| \tag{181}$$

$$= \sup_{v\in\mathcal{S}^{n-1}}|\langle \bar{D}^{1/2}U^\top(B\circ R)^\top u, v\rangle| \tag{182}$$

$$= \left\|\bar{D}^{1/2}U^\top(B\circ R)^\top u\right\|_2. \tag{183}$$

Thus, for any R, R' we have

$$\sqrt{\mathrm{m}_{\mathcal{X}}\, p}\left|F\left(R\right) - F\left(R'\right)\right| = \left|\left\|\bar{D}^{1/2}U^\top(B\circ R)^\top u\right\|_2 - \left\|\bar{D}^{1/2}U^\top(B\circ R')^\top u\right\|_2\right| \tag{184}$$

$$\leq \left\|\bar{D}^{1/2}U^\top\left(B\circ(R-R')\right)^\top u\right\|_2 \tag{185}$$

$$\leq \left\|\bar{D}^{1/2}\right\|_{\mathrm{op}}\left\|U^\top\right\|_{\mathrm{op}}\left\|B\circ(R-R')\right\|_{\mathrm{op}}\|u\|_2 \tag{186}$$

$$\leq \delta_n\left\|B\circ(R-R')\right\|_F \tag{187}$$

$$\leq \delta_n\left\|R-R'\right\|_F, \tag{188}$$

such that F is $\sqrt{\delta_n^2/(\mathrm{m}_{\mathcal{X}}\, p)}$-Lipschitz. Moreover, we have

$$\sqrt{\mathrm{m}_{\mathcal{X}}\, p}\ \mathbb{E}\Big[F(R)\Big] = \mathbb{E}\left[\left\|D_2^{1/2} U_2^\top (B \circ R)^\top u\right\|_2\right] \tag{189}$$

$$\leq \sqrt{\mathbb{E}\Big[u^\top (B \circ R) U_2 D_2 U_2^\top (B \circ R)^\top u\Big]} \tag{190}$$

$$= \sqrt{\sum_{k,k'=1}^{\mathrm{m}_{\mathcal{X}}} u_k u_{k'}\, \mathbb{E}\Big[\Big[(B \circ R) U_2 D_2 U_2^\top (B \circ R)^\top\Big]_{kk'}\Big]} \tag{191}$$

$$= \sqrt{\sum_{k,k'=1}^{\mathrm{m}_{\mathcal{X}}} \sum_{l,l'=1}^{n} u_k u_{k'} [U_2 D_2 U_2^\top]_{ll'}\, \mathbb{E}\Big[[B \circ R]_{kl} [B \circ R]_{k'l'}\Big]} \tag{192}$$

$$= \sqrt{\sum_{k=1}^{\mathrm{m}_{\mathcal{X}}} \sum_{l=1}^{n} B_{kl}^2 u_k^2 [U_2 D_2 U_2^\top]_{ll}} \tag{193}$$

$$\leq \sqrt{\mathrm{Tr}(D_2)}, \tag{194}$$

which implies

$$\mathbb{E}\Big[F(R)\Big] \leq \sqrt{\frac{n}{\mathrm{m}_{\mathcal{X}}\, p}} \sqrt{\frac{\sum_{j=d_n+1}^{n} \mu_j}{n}} \leq \sqrt{\frac{n}{\mathrm{m}_{\mathcal{X}}\, p}} \sqrt{\frac{1}{n} \sum_{j=d_n+1}^{n} \min(\mu_j, \delta_n^2)} \leq \sqrt{\frac{\delta_n^2}{p}}, \tag{195}$$

where we have used the definition of δ_n^2 and the assumption $\mathrm{m}_{\mathcal{X}} \geq \delta_n^2 n$. Coming back to eq. (177), we obtain

$$\mathbb{P}\left\{\left\|\mathrm{R}_{\mathcal{X}}\, U_2 D_2^{1/2}\right\|_{\mathrm{op}} > c\delta_n\right\} \leq 5^{\mathrm{m}_{\mathcal{X}}}\ \mathbb{E}\left[\mathbb{P}\left\{\sup_{v \in \mathcal{E}} \left|\left\langle u, \mathrm{R}_{\mathcal{X}}\, v\right\rangle\right| > \frac{c}{2}\delta_n \,\middle|\, B\right\}\right] \tag{196}$$

$$= 5^{\mathrm{m}_{\mathcal{X}}}\ \mathbb{E}\left[\mathbb{P}\left\{F(R) > \frac{c}{2}\delta_n\right\}\right] \tag{197}$$

$$\leq 5^{\mathrm{m}_{\mathcal{X}}}\ \mathbb{E}\left[\mathbb{P}\left\{F(R) - \mathbb{E}[F(R)] > \delta_n\left(\frac{c}{2} - \frac{1}{\sqrt{p}}\right)\right\}\right] \tag{198}$$

$$\leq C\, 5^{\mathrm{m}_{\mathcal{X}}} \exp\left(-c'\left(\frac{c}{2} - \frac{1}{\sqrt{p}}\right)^2 \delta_n^2 \frac{\mathrm{m}_{\mathcal{X}}\, p}{\delta_n^2}\right) \tag{199}$$

$$\leq C \exp\left(-c'\left(\left(\frac{c}{2} - \frac{1}{\sqrt{p}}\right)^2 p - \log(5)\right)\mathrm{m}_{\mathcal{X}}\right), \tag{200}$$

where eq. (198) comes from the upper bound on $\mathbb{E}[F(R)]$ we derived in eq. (195), and eq. (199) derives from Lemmas .8 and .9 applied to the function F whose Lipschitz constant has been established in eq. (188). Therefore, taking $c = \frac{2}{\sqrt{p}}\left(1 + \sqrt{\log\left(5\right)}\right) + 1$, we have

$$\mathbb{P}\left\{\left\|\mathrm{R}_{\mathcal{X}}\, U_2 D_2^{1/2}\right\|_{\mathrm{op}} > c\delta_n\right\} \leq c_1 e^{-c_2 \mathrm{m}_{\mathcal{X}}} \tag{201}$$

with $c_1 = C'$ and $c_2 = c'\left(\sqrt{p\log\left(5\right)} + \frac{p}{4}\right)$. ■

Proof of Proposition 3.17

We prove Proposition 3.17 thanks to duality properties.

Proposition 3.17. *Solving Problem (3.3) is equivalent to solving*

$$\min_{\omega\in\mathbb{R}^{\mathrm{p_X}}} \frac{1}{n}\sum_{i=1}^{n} \ell\left(\omega^\top \tilde{\psi}_{\mathcal{X}}\left(x_i\right), y_i\right) + \frac{\lambda_n}{2}\|\omega\|_2^2, \tag{3.11}$$

where $\tilde{\psi}_{\mathcal{X}}(x) = \tilde{D}_{\mathrm{p_X}}^{-1/2}\tilde{U}_{\mathrm{p_X}}^\top \mathrm{R}_{\mathcal{X}}\left(\mathrm{k}_{\mathcal{X}}\left(x, x_1\right), \ldots, \mathrm{k}_{\mathcal{X}}\left(x, x_n\right)\right)^\top \in \mathbb{R}^{\mathrm{p_X}}$.

Proof Since problems (3.3) and (3.11) are convex problems under Slater's constraints, strong duality holds and we will show that they admit the same dual problem

$$\min_{\zeta\in\mathbb{R}^n} \sum_{i=1}^{n} \ell_i^\star\left(-\zeta_i\right) + \frac{1}{2\lambda_n n}\zeta^\top \mathrm{K_X}\mathrm{R}_{\mathcal{X}}{}^\top(\mathrm{R}_{\mathcal{X}}\mathrm{K_X}\mathrm{R}_{\mathcal{X}}{}^\top)^\dagger \mathrm{R}_{\mathcal{X}}\mathrm{K_X}\zeta\,, \tag{3.12}$$

where $\ell_i^\star$ denotes the Fenchel-Legendre transform of $\ell_i : y \in \mathbb{R} \mapsto \ell(y, y_)$ for any $i \le n$. First, we compute dual problem of (3.3), that can be rewritten

$$\min_{\gamma\in\mathbb{R}^{\mathrm{m}_{\mathcal{X}}}, u\in\mathbb{R}^n} \sum_{i=1}^{n} \ell_i\left(u_i\right) + \frac{\lambda_n n}{2}\gamma^\top \mathrm{R}_{\mathcal{X}}\mathrm{K_X}\mathrm{R}_{\mathcal{X}}{}^\top\gamma \tag{202}$$

$$\text{s.t. } u = \mathrm{K_X}\mathrm{R}_{\mathcal{X}}{}^\top\gamma. \tag{203}$$

Therefore the Lagrangian writes

$$\mathcal{L}(\gamma, u, \zeta) = \sum_{i=1}^{n} \ell_i\left(u_i\right) + \frac{\lambda_n n}{2}\gamma^\top \mathrm{R}_{\mathcal{X}}\mathrm{K_X}\mathrm{R}_{\mathcal{X}}{}^\top\gamma + \sum_{i=1}^{n}\zeta_i(u_i - [\mathrm{K_X}\mathrm{R}_{\mathcal{X}}{}^\top\gamma]_i) \tag{204}$$

$$= \sum_{i=1}^{n} \ell_i\left(u_i\right) + \frac{\lambda_n n}{2}\gamma^\top \mathrm{R}_{\mathcal{X}}\mathrm{K_X}\mathrm{R}_{\mathcal{X}}{}^\top\gamma + \sum_{i=1}^{n}\zeta_i u_i - \zeta^\top \mathrm{K_X}\mathrm{R}_{\mathcal{X}}{}^\top\gamma. \tag{205}$$

Differentiating with respect to γ and using the definition of the Fenchel-Legendre transform, one gets

$$g(\zeta) = \inf_{\gamma\in\mathbb{R}^{\mathrm{m}_{\mathcal{X}}}, u\in\mathbb{R}^n} \mathcal{L}(\gamma, u, \zeta) \tag{206}$$

$$= \sum_{i=1}^{n} \inf_{u_i\in\mathbb{R}}\left\{\ell_i\left(u_i\right) + \zeta_i u_i\right\} + \inf_{\gamma\in\mathbb{R}^{\mathrm{m}_{\mathcal{X}}}}\left\{\frac{\lambda_n n}{2}\gamma^\top \mathrm{R}_{\mathcal{X}}\mathrm{K_X}\mathrm{R}_{\mathcal{X}}{}^\top\gamma - \zeta^\top \mathrm{K_X}\mathrm{R}_{\mathcal{X}}{}^\top\gamma\right\} \tag{207}$$

$$= \sum_{i=1}^{n} -\ell_i^\star\left(-\zeta_i\right) - \frac{1}{2\lambda_n n}\zeta^\top \mathrm{K_X}\mathrm{R}_{\mathcal{X}}{}^\top(\mathrm{R}_{\mathcal{X}}\mathrm{K_X}\mathrm{R}_{\mathcal{X}}{}^\top)^\dagger \mathrm{R}_{\mathcal{X}}\mathrm{K_X}\zeta \tag{208}$$

together with the equality $\mathrm{R}_{\mathcal{X}}\mathrm{K_X}\mathrm{R}_{\mathcal{X}}{}^\top\tilde{\gamma} = \frac{1}{\lambda_n n}\mathrm{R}_{\mathcal{X}}\mathrm{K_X}\tilde{\zeta}$, implying $\tilde{\gamma} = \frac{1}{\lambda_n n}(\mathrm{R}_{\mathcal{X}}\mathrm{K_X}\mathrm{R}_{\mathcal{X}}{}^\top)^\dagger \cdot \mathrm{R}_{\mathcal{X}}\mathrm{K_X}\tilde{\zeta}$, where $\tilde{\zeta} \in \mathbb{R}^n$ is the solution of the following dual problem

$$\min_{\zeta\in\mathbb{R}^n} \sum_{i=1}^{n} \ell_i^\star\left(-\zeta_i\right) + \frac{1}{2\lambda_n n}\zeta^\top \mathrm{K_X}\mathrm{R}_{\mathcal{X}}{}^\top(\mathrm{R}_{\mathcal{X}}\mathrm{K_X}\mathrm{R}_{\mathcal{X}}{}^\top)^\dagger \mathrm{R}_{\mathcal{X}}\mathrm{K_X}\zeta\,. \tag{3.12}$$

Then, we compute dual problem of (3.11), that can be rewritten

$$\min_{\omega\in\mathbb{R}^{\mathrm{pX}},u\in\mathbb{R}^n}\sum_{i=1}^{n}\ell\left(u_i,y_i\right)+\frac{\lambda_n n}{2}\|\omega\|_2^2 \tag{209}$$

$$\text{s.t. } u=\mathrm{K_X}\,\mathrm{R}_{\mathcal{X}}{}^{\top}\tilde{U}_{\mathrm{p_X}}\tilde{D}_{\mathrm{p_X}}^{-1/2}\omega. \tag{210}$$

Therefore the Lagrangian writes

$$\mathcal{L}(\omega,u,\zeta)=\sum_{i=1}^{n}\ell_i\left(u_i\right)+\frac{\lambda_n n}{2}\|\omega\|_2^2+\sum_{i=1}^{n}\zeta_i(u_i-[\mathrm{K_X}\,\mathrm{R}_{\mathcal{X}}{}^{\top}\tilde{U}_{\mathrm{p_X}}\tilde{D}_{\mathrm{p_X}}^{-1/2}\omega]_i) \tag{211}$$

$$=\sum_{i=1}^{n}\ell_i\left(u_i\right)+\frac{\lambda_n n}{2}\|\omega\|_2^2+\sum_{i=1}^{n}\zeta_i^{\top}u_i-\omega^{\top}\tilde{D}_{\mathrm{p_X}}^{-1/2}\tilde{U}_{\mathrm{p_X}}^{\top}\mathrm{R}_{\mathcal{X}}\mathrm{K_X}\,\zeta. \tag{212}$$

Differentiating with respect to ω and using the definition of the Fenchel-Legendre transform, one gets

$$g(\zeta)=\inf_{\omega\in\mathbb{R}^{\mathrm{pX}},u\in\mathbb{R}^n}\mathcal{L}(\omega,u,\zeta) \tag{213}$$

$$=\sum_{i=1}^{n}\inf_{u_i\in\mathbb{R}}\left\{\ell_i\left(u_i\right)+\zeta_i u_i\right\}+\inf_{\omega\in\mathbb{R}^{\mathrm{pX}}}\left\{\frac{\lambda_n n}{2}\|\omega\|_2^2-\omega^{\top}\tilde{D}_{\mathrm{p_X}}^{-1/2}\tilde{U}_{\mathrm{p_X}}^{\top}\mathrm{R}_{\mathcal{X}}\mathrm{K_X}\,\zeta\right\}. \tag{214}$$

We have that

$$\frac{\partial}{\partial\omega}\left(\|\omega\|_2^2\right)=2\omega \tag{215}$$

$$\frac{\partial}{\partial\omega}\left(\omega^{\top}\tilde{D}_{\mathrm{p_X}}^{-1/2}\tilde{U}_{\mathrm{p_X}}^{\top}\mathrm{R}_{\mathcal{X}}\mathrm{K_X}\,\zeta\right)=\tilde{D}_{\mathrm{p_X}}^{-1/2}\tilde{U}_{\mathrm{p_X}}^{\top}\mathrm{R}_{\mathcal{X}}\mathrm{K_X}\,\zeta, \tag{216}$$

Then, setting the gradient to zero, we obtain

$$\tilde{\omega}=\frac{1}{\lambda_n n}\,\tilde{D}_{\mathrm{p_X}}^{-1/2}\tilde{U}_{\mathrm{p_X}}^{\top}\mathrm{R}_{\mathcal{X}}\mathrm{K_X}\,\tilde{\zeta}. \tag{217}$$

Hence, putting it into the Lagrangian,

$$-\frac{1}{\lambda_n n}\zeta^{\top}\mathrm{K_X}\,\mathrm{R}_{\mathcal{X}}{}^{\top}\tilde{U}_{\mathrm{p_X}}\tilde{D}_{\mathrm{p_X}}^{-1/2}\tilde{D}_{\mathrm{p_X}}^{-1/2}\tilde{U}_{\mathrm{p_X}}^{\top}\mathrm{R}_{\mathcal{X}}\mathrm{K_X}\,\zeta=-\frac{1}{\lambda_n n}\mathrm{K_X}\,\mathrm{R}_{\mathcal{X}}{}^{\top}\left(\mathrm{R}_{\mathcal{X}}\mathrm{K_X}\,\mathrm{R}_{\mathcal{X}}{}^{\top}\right)^{\dagger}\mathrm{R}_{\mathcal{X}}\mathrm{K_X}\,\zeta, \tag{218}$$

and

$$\frac{1}{2\lambda_n n}\zeta^{\top}\mathrm{K_X}\,\mathrm{R}_{\mathcal{X}}{}^{\top}\tilde{U}_{\mathrm{p_X}}\tilde{D}_{\mathrm{p_X}}^{-1/2}\tilde{D}_{\mathrm{p_X}}^{-1/2}\tilde{U}_{\mathrm{p_X}}^{\top}\mathrm{R}_{\mathcal{X}}\mathrm{K_X}\,\zeta=\frac{1}{2\lambda_n n}\mathrm{K_X}\,\mathrm{R}_{\mathcal{X}}{}^{\top}\left(\mathrm{R}_{\mathcal{X}}\mathrm{K_X}\,\mathrm{R}_{\mathcal{X}}{}^{\top}\right)^{\dagger}\mathrm{R}_{\mathcal{X}}\mathrm{K_X}\,\zeta. \tag{219}$$

Hence, $\tilde{\zeta}\in\mathbb{R}^n$ is the solution to the following dual problem

$$\min_{\zeta\in\mathbb{R}^n}\sum_{i=1}^{n}\ell_i^{\star}\left(-\zeta_i\right)+\frac{1}{2\lambda_n n}\zeta^{\top}\mathrm{K_X}\,\mathrm{R}_{\mathcal{X}}{}^{\top}(\mathrm{R}_{\mathcal{X}}\mathrm{K_X}\,\mathrm{R}_{\mathcal{X}}{}^{\top})^{\dagger}\mathrm{R}_{\mathcal{X}}\mathrm{K_X}\,\zeta\,. \tag{3.12}$$

Finally, since both problems are convex and strong duality holds, we obtain through KKT conditions

$$\tilde{\omega} = \tilde{D}_{\mathrm{p_X}}^{-1/2} \tilde{U}_{\mathrm{p_X}}^{\top} \left(\mathrm{R}_{\mathcal{X}} \mathrm{K_X} \mathrm{R}_{\mathcal{X}}^{\top} \right) \tilde{\gamma} \tag{220}$$

$$= \tilde{D}_{\mathrm{p_X}}^{-1/2} \tilde{U}_{\mathrm{p_X}}^{\top} \tilde{U} \tilde{D} \tilde{U}^{\top} \tilde{\gamma} \tag{221}$$

$$= \underbrace{\left(\tilde{D}_{\mathrm{p_X}}^{1/2} 0_{\mathrm{p_X} \times \mathrm{m}_{\mathcal{X}} - \mathrm{p_X}} \right)}_{\mathrm{p_X} \times \mathrm{m}_{\mathcal{X}}} \underbrace{\tilde{U}^{\top}}_{\mathrm{m}_{\mathcal{X}} \times \mathrm{m}_{\mathcal{X}}} \underbrace{\tilde{\gamma}}_{\mathrm{m}_{\mathcal{X}} \times 1} \tag{222}$$

and

$$\min_{\gamma \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}}}} \frac{1}{n} \sum_{i=1}^{n} \ell([\mathrm{K_X} \mathrm{R}_{\mathcal{X}}^{\top} \gamma]_i, y_i) + \frac{\lambda_n}{2} \gamma^{\top} \mathrm{R}_{\mathcal{X}} \mathrm{K_X} \mathrm{R}_{\mathcal{X}}^{\top} \gamma \tag{223}$$

$$= \min_{\omega \in \mathbb{R}^{\mathrm{p_X}}} \sum_{i=1}^{n} \ell\left(\omega^{\top} \tilde{\psi}_{\mathcal{X}}\left(x_i\right), y_i \right) + \frac{\lambda_n}{2} \|\omega\|_2^2 . \tag{224}$$

■

A.2 On relaxing Assumption 3.6

In this section, we detail the discussion about relaxing Assumption 3.6, i.e. the restriction of the hypothesis set to the unit ball of the RKHS. Assumption 3.6 is a classical assumption in kernel literature to apply generalisation bounds based on the Rademacher complexity of a bounded ball of an RKHS. Moreover, it is also useful in our case to derive an approximation error bound, describing how $\mathrm{K_X}$-satisfiability of a sketch matrix allows to obtain a good approximation of the minimiser of the risk. However, let us discuss the consequences of relaxing this assumption. Indeed, all we need is a bound on the norm of the estimators $\tilde{f}$ – minimiser of the regularised ERM sketched problem – and $\tilde{f}^R$ – minimiser of the regularised ERM sketched denoised KRR problem. By definition, noting $\mathcal{H}_{\mathrm{R}_{\mathcal{X}}} = \left\{ f = \sum_{i=1}^{n} \left[\mathrm{R}_{\mathcal{X}}^{\top} \gamma \right]_i \mathrm{k}_{\mathcal{X}}\left(\cdot, x_i\right) \mid \gamma \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}}} \right\}$, we have that

$$\tilde{f} = \operatorname*{arg\,min}_{f \in \mathcal{H}_{\mathrm{R}_{\mathcal{X}}}} \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i) + \frac{\lambda_n}{2} \|f\|_{\mathcal{H}_{\mathcal{X}}}^2 . \tag{225}$$

Hence,

$$\frac{\lambda_n}{2} \|\tilde{f}\|_{\mathcal{H}_{\mathcal{X}}}^2 \leq \frac{1}{n} \sum_{i=1}^{n} \ell(\tilde{f}(x_i), y_i) + \frac{\lambda_n}{2} \|\tilde{f}\|_{\mathcal{H}_{\mathcal{X}}}^2 \leq \frac{1}{n} \sum_{i=1}^{n} \ell(0, y_i) \leq 1 , \tag{226}$$

if we assume that $\max_{1 \leq i \leq n} \ell(0, y_i) \leq 1$ to simplify the derivations. As a consequence, we obtain that

$$\|\tilde{f}\|_{\mathcal{H}_{\mathcal{X}}} \leq \sqrt{\frac{2}{\lambda_n}} . \tag{227}$$

Similarly, we have that

$$\tilde{f}^R = \operatorname*{arg\,min}_{f \in \mathcal{H}_{\mathrm{R}_{\mathcal{X}}}} \frac{1}{n} \left\| f^X - f_{\mathcal{H}_{\mathcal{X}}}^X \right\|_2^2 + \frac{\lambda_n}{2} \|f\|_{\mathcal{H}_{\mathcal{X}}}^2 , \tag{228}$$

that gives

$$\|\tilde{f}^R\|_{\mathcal{H}_\mathcal{X}} \leq \left(\frac{1}{\lambda_n n} \left\| f_{\mathcal{H}_\mathcal{X}} \right\|^2_{\mathcal{H}_\mathcal{X}} \right)^{1/2}. \tag{229}$$

By Assumptions 3.6 and 3.8,

$$\begin{aligned} \frac{1}{n} \left\| f_{\mathcal{H}_\mathcal{X}} \right\|^2_{\mathcal{H}_\mathcal{X}} &= \frac{1}{n} \sum_{i=1}^{n} \left\langle f_{\mathcal{H}_\mathcal{X}}, \mathrm{k}_\mathcal{X}\left(\cdot, x_i\right) \right\rangle_{\mathcal{H}_\mathcal{X}} && (230) \\ &\leq \frac{1}{n} \sum_{i=1}^{n} \left\| f_{\mathcal{H}_\mathcal{X}} \right\|^2_{\mathcal{H}_\mathcal{X}} \mathrm{k}_\mathcal{X}\left(x_i, x_i\right) && (231) \\ &\leq \kappa_\mathcal{X}, && (232) \end{aligned}$$

and finally

$$\|\tilde{f}^R\|_{\mathcal{H}_\mathcal{X}} \leq \sqrt{\frac{\kappa_\mathcal{X}}{\lambda_n}}. \tag{233}$$

Remark .11. *Note that in the multiple output settings, we obtain*

$$\|\tilde{f}^R\|_{\mathcal{H}} \leq \sqrt{\frac{\kappa_\mathcal{X} \operatorname{Tr}(M)}{\lambda_n}}. \tag{234}$$

We are now equipped to derive the generalisation error bound $\mathbb{E}\left[\ell_{\tilde{f}}\right] - \mathbb{E}_n\left[\ell_{\tilde{f}}\right]$ and the approximation error bound $\mathbb{E}_n\left[\ell_{\tilde{f}}\right] - \mathbb{E}_n\left[\ell_{f_{\mathcal{H}_\mathcal{X}}}\right]$. We first focus on the generalisation bound, and following the proof given in Appendix A.1 and given the new norm upper bound $\sqrt{\frac{2}{\lambda_n}}$, for any $\delta \in (0,1)$, with probability $1 - \delta/2$, we have that

$$\mathbb{E}\left[\ell_{\tilde{f}}\right] - \mathbb{E}_n\left[\ell_{\tilde{f}}\right] \leq \frac{4L\sqrt{2\,\kappa_\mathcal{X}}}{\sqrt{\lambda_n n}} + \sqrt{\frac{8\log(4/\delta)}{n}}. \tag{235}$$

This dependence in $1/\sqrt{\lambda_n}$ shows that, as expected by a regularisation penalty, with a fixed n, when λ_n increases, the generalisation bound decreases and then we obtain a better generalisation performance. However, this behaviour does not reflect completely the role of λ_n, since there exists a tradeoff between overfitting and underfitting, and then it should not be set too large. We now focus on the approximation error

bound. As in Appendix A.1, we obtain that

$$\mathbb{E}_n\left[\ell_{\tilde{f}}\right] - \mathbb{E}_n\left[\ell_{f_{\mathcal{H}_\mathcal{X}}}\right] = \frac{1}{n}\sum_{i=1}^{n}\ell\left(\tilde{f}(x_i), y_i\right) - \frac{1}{n}\sum_{i=1}^{n}\ell\left(f_{\mathcal{H}_\mathcal{X}}(x_i), y_i\right) \tag{236}$$

$$= \inf_{\substack{f\in\mathcal{H}_{\mathrm{R}_\mathcal{X}} \\ \|f\|_{\mathcal{H}_\mathcal{X}}\leq\left\|\tilde{f}\right\|_{\mathcal{H}_\mathcal{X}}}} \frac{1}{n}\sum_{i=1}^{n}\ell\left(f(x_i), y_i\right) - \frac{1}{n}\sum_{i=1}^{n}\ell\left(f_{\mathcal{H}_\mathcal{X}}(x_i), y_i\right) \tag{237}$$

$$\leq \inf_{\substack{f\in\mathcal{H}_{\mathrm{R}_\mathcal{X}} \\ \|f\|_{\mathcal{H}_\mathcal{X}}\leq\left\|\tilde{f}\right\|_{\mathcal{H}_\mathcal{X}}}} \frac{L}{n}\sum_{i=1}^{n}\left|f\left(x_i\right) - f_{\mathcal{H}_\mathcal{X}}\left(x_i\right)\right| \tag{238}$$

$$\leq L \inf_{\substack{f\in\mathcal{H}_{\mathrm{R}_\mathcal{X}} \\ \|f\|_{\mathcal{H}_\mathcal{X}}\leq\left\|\tilde{f}\right\|_{\mathcal{H}_\mathcal{X}}}} \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left|f\left(x_i\right) - f_{\mathcal{H}_\mathcal{X}}\left(x_i\right)\right|^2} \tag{239}$$

$$= L \sqrt{\inf_{\substack{f\in\mathcal{H}_{\mathrm{R}_\mathcal{X}} \\ \|f\|_{\mathcal{H}_\mathcal{X}}\leq\left\|\tilde{f}\right\|_{\mathcal{H}_\mathcal{X}}}} \frac{1}{n}\left\|f^X - f^X_{\mathcal{H}_\mathcal{X}}\right\|_2^2}, \tag{240}$$

where, for any $f \in \mathcal{H}_\mathcal{X}$, $f^X = \left(f(x_1), \ldots, f(x_n)\right) \in \mathbb{R}^n$. Let $\tilde{f}^R = \sum_{i=1}^{n}\left[\mathrm{R}_\mathcal{X}{}^\top \tilde{\gamma}^R\right]_i \mathrm{k}_\mathcal{X}\left(\cdot, x_i\right)$, where $\tilde{\gamma}^R$ is a solution to

$$\inf_{\gamma\in\mathbb{R}^{\mathrm{m}_\mathcal{X}}} \frac{1}{n}\left\|\mathrm{K}_\mathrm{X}\mathrm{R}_\mathcal{X}{}^\top\gamma - f^X_{\mathcal{H}_\mathcal{X}}\right\|_2^2 + \lambda_n \gamma^\top \mathrm{R}_\mathcal{X}\mathrm{K}_\mathrm{X}\mathrm{R}_\mathcal{X}{}^\top\gamma. \tag{241}$$

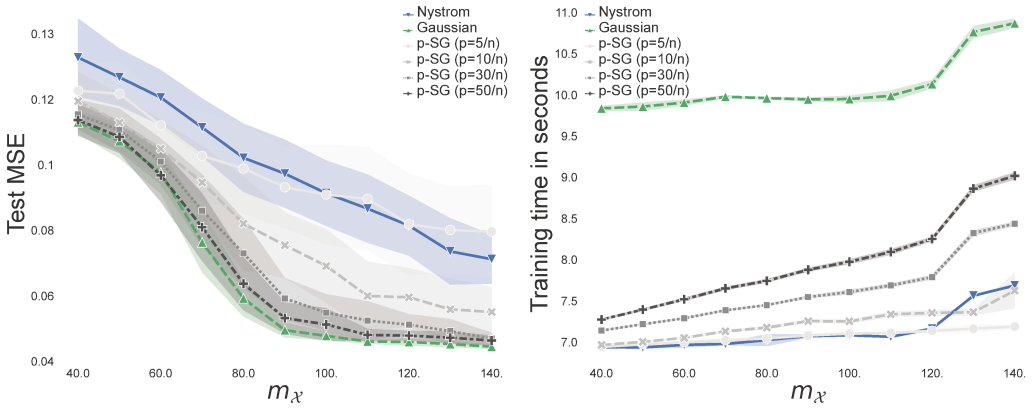
It is easy to check that $\tilde{f}^R$ is also a solution to

$$\inf_{\substack{f\in\mathcal{H}_{\mathrm{R}_\mathcal{X}} \\ \|f\|_{\mathcal{H}_\mathcal{X}}\leq\left\|\tilde{f}^R\right\|_{\mathcal{H}_\mathcal{X}}}} \frac{1}{n}\left\|f^X - f^X_{\mathcal{H}_\mathcal{X}}\right\|_2^2. \tag{242}$$

Now, comparing eq. (240) and eq. (242), as done in Appendix A.1, essentially boils down to comparing $\left\|\tilde{f}\right\|_{\mathcal{H}_\mathcal{X}}$ and $\left\|\tilde{f}^R\right\|_{\mathcal{H}_\mathcal{X}}$, which is a highly nontrivial question. In particular, the upper bounds (227) and (233) are not informative enough. Another solution could consist in adding $\frac{\lambda_n}{2}\left\|\tilde{f}\right\|_{\mathcal{H}_\mathcal{X}}$ to eq. (236). However, the upper bound (227) then transforms this term into a constant bias. This can be explained as eq. (227) is very crude. Instead, having $\left\|\tilde{f}\right\|_{\mathcal{H}_\mathcal{X}}$ bounded by λ_n^α for $\alpha \geq -1/2$ would be enough to exhibit a bias term that vanishes as λ_n goes to 0. Note that it would still degrade the tradeoff with the generalisation term. Hence, if generalisation errors can be dealt with when removing Assumption 3.6, it is much more complex to control eq. (236).

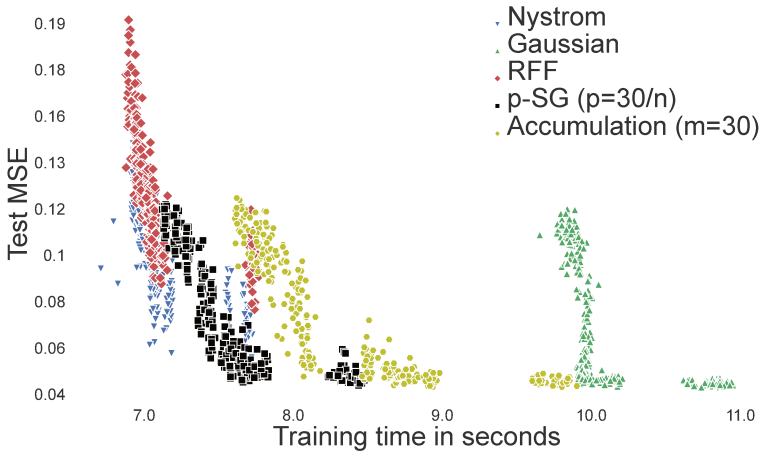
A.3 Additional experiments and details

In this section, we bring some additional experiments and details.



(a) Test relative MSE w.r.t. $m_{\mathcal{X}}$ with κ-Huber.

(b) Training time (seconds) w.r.t. $m_{\mathcal{X}}$ with κ-Huber.



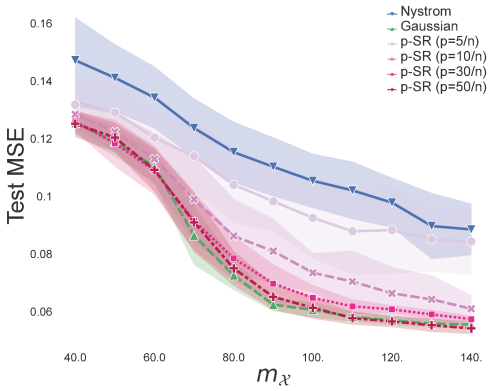
(c) Test relative MSE w.r.t. training times with κ-Huber.

Figure A.1: Trade-off between Accuracy and Efficiency for p-SG sketches with κ-Huber loss on synthetic dataset.

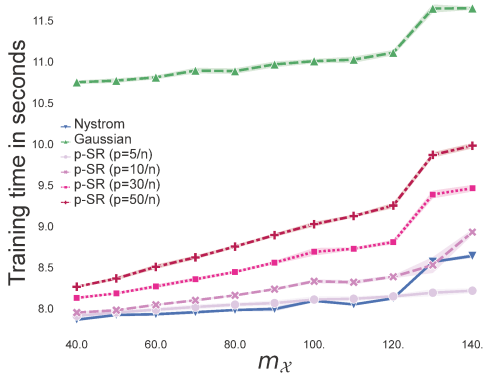
Simulated dataset for single output regression

First, we report the plots obtained with κ-Huber for p-SG sketches (see Figure A.1) and note that we observe a behaviour similar to p-SR sketches when varying p and in comparison to other types of sketching and RFFs. However, we see that the MSE obtained is slightly worse than p-SR sketches. An explanation might be that, in a very sparse regime, i.e. very low p, a p-SG sketch is too different than a Gaussian sketch, making it lose some good statistical properties of Gaussian sketches. We however observe that the larger p is, the smaller the statistical performance between p-SR and p-SG sketches.

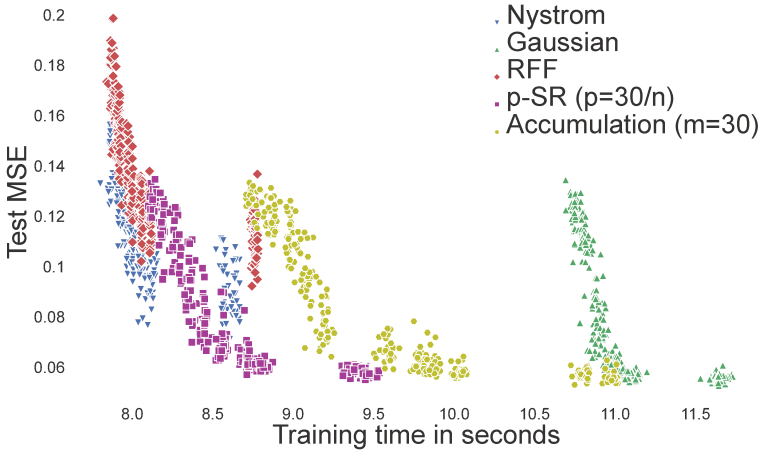
We then report in the following the corresponding plots obtained with ϵ-SVR, that witnesses the same phenomenon observed earlier with κ-Huber about the interpolation between Nyström method and Gaussian sketching while varying the probability of being different than 0 p, with p-SR sketches (see Figure A.2) and p-SG sketches (see Figure A.3).



(a) Test relative MSE w.r.t. $m_{\mathcal{X}}$ with ϵ-SVR.



(b) Training time (seconds) w.r.t. $m_{\mathcal{X}}$ with ϵ-SVR.



(c) Test relative MSE w.r.t. training times with ϵ-SVR.

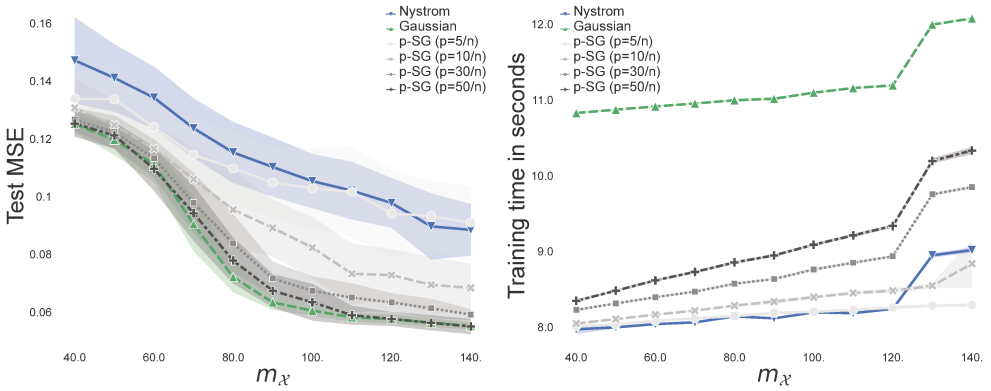
Figure A.2: Trade-off between Accuracy and Efficiency for p-SR sketches with ϵ-SVR loss on synthetic dataset.

Multi-Output Regression on real datasets

We here first a brief presentation on River Flow and Supply Chain Management:

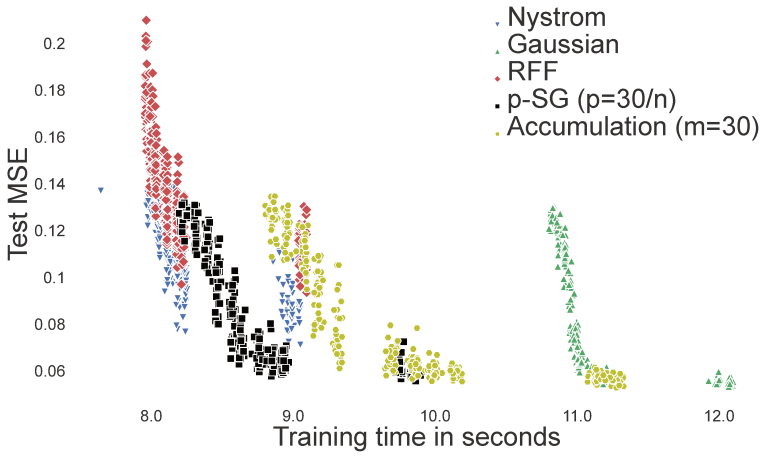
Table .1: Numbers of training samples (n), test samples (n_{te}), features (q) and targets (d).

Dataset	n	n_{te}	q	d
Boston	354	152	13	5
otoliths	3780	165	4096	5
rf1	4108	5017	64	8
rf2	4108	5017	576	8
scm1d	8145	1658	280	16
scm20d	7463	1503	61	16



(a) Test relative MSE w.r.t. $m_{\mathcal{X}}$ with ϵ-SVR.

(b) Training time (seconds) w.r.t. $m_{\mathcal{X}}$ with ϵ-SVR.



(c) Test relative MSE w.r.t. training times with ϵ-SVR.

Figure A.3: Trade-off between Accuracy and Efficiency for p-SG sketches with ϵ-SVR loss on synthetic dataset.

1. River Flow datasets aim at predicting the river network flows for 48 hours in the future at specific locations. These locations are 8 sites in the Mississippi River network in the United States and were obtained from the US National Weather Service. Dataset rf2 extends rf1 since it contains additional precipitation forecast information for each of the 8 sites.

2. The datasets scm1d and sm20d come from the Trading Agent Competition in Supply Chain Management (TAC SCM) tournament from 2010. More details about data preprocessing can be found in Groves and Gini (2015). The dataset contains prices of products on specific days, and the task is to predict the next day's mean price (scm1d) or mean price for 20 days in the future (scm20d) for each product in the simulation.

To conduct these experiments, the train-test splits used are the ones available at `http://mulan.sourceforge.net/datasets-mtr.html`. Besides, we used a multi-output Kernel Ridge Regression framework, an input Gaussian kernel, and an operator $M = I_d$. We selected regularisation parameter λ_n and bandwidth of kernel σ^2 via a 5-fold cross-validation. Results are averages over 30 replicates for sketched models.

Table .2: Test RRMSE and ARRMSE for different methods on the MTR datasets. For decomposable kernel-based models, loss here is square loss and $s = 100$ when performing Sketching.

Dataset	Targets	w/o Sketch	$20/n_{tr}$-SR	$20/n_{tr}$-SG	Acc. $m = 20$	CountSketch	SOTA
rf1	Mean	**0.575**	0.584 ± 0.003	0.583 ± 0.003	0.592 ± 0.001	**0.575 ± 0.0005**	[0.091, 0.983]
	chsi2	0.351	0.356 ± 0.005	0.357 ± 0.004	0.361 ± 0.002	**0.350 ± 0.002**	[0.033, 0.797]
	nasi2	1.085	1.124 ± 0.003	1.124 ± 0.003	**1.082 ± 0.0004**	1.110 ± 0.0003	[0.376, 1.951]
	eadm7	0.388	0.397 ± 0.004	0.398 ± 0.003	**0.387 ± 0.001**	**0.387 ± 0.001**	[0.039, 1.019]
	sclm7	0.660	0.659 ± 0.008	0.661 ± 0.005	0.681 ± 0.002	**0.648 ± 0.002**	[0.047, 1.503]
	clkm7	0.283	**0.281 ± 0.001**	0.282 ± 0.001	0.293 ± 0.0005	**0.281 ± 0.0004**	[0.031, 0.587]
	vali2	0.614	0.633 ± 0.008	0.635 ± 0.010	0.656 ± 0.003	**0.611 ± 0.003**	[0.037, 0.571]
	napm7	**0.593**	0.628 ± 0.020	0.614 ± 0.016	0.627 ± 0.003	0.601 ± 0.003	[0.038, 0.909]
	dldi4	0.629	**0.597 ± 0.004**	**0.597 ± 0.003**	0.646 ± 0.001	0.614 ± 0.002	[0.029, 0.534]
rf2	Mean	**0.578**	0.671 ± 0.009	0.656 ± 0.006	0.796 ± 0.006	0.715 ± 0.011	[0.095, 1.103]
	chsi2	**0.318**	0.382 ± 0.016	0.358 ± 0.010	0.478 ± 0.006	0.426 ± 0.013	[0.034, 0.737]
	nasi2	1.099	1.084 ± 0.005	1.092 ± 0.006	**1.018 ± 0.003**	1.036 ± 0.002	[0.384, 3.143]
	eadm7	**0.342**	0.390 ± 0.013	0.369 ± 0.007	0.456 ± 0.004	0.417 ± 0.010	[0.040, 0.737]
	sclm7	**0.610**	0.719 ± 0.030	0.672 ± 0.021	0.948 ± 0.014	0.852 ± 0.030	[0.049, 0.970]
	clkm7	**0.311**	0.328 ± 0.009	0.330 ± 0.009	0.614 ± 0.005	0.436 ± 0.006	[0.041, 0.891]
	vali2	**0.712**	0.960 ± 0.044	0.894 ± 0.043	0.890 ± 0.017	0.939 ± 0.028	[0.047, 0.956]
	napm7	**0.589**	0.812 ± 0.014	0.831 ± 0.017	1.110 ± 0.023	0.856 ± 0.032	[0.039, 0.617]
	dldi4	**0.646**	0.696 ± 0.010	0.701 ± 0.011	0.855 ± 0.004	0.761 ± 0.007	[0.032, 0.770]
scm1d	Mean	**0.418**	0.422 ± 0.002	0.423 ± 0.001	0.423 ± 0.001	0.420 ± 0.001	[0.330, 0.457]
	lbl	**0.358**	0.365 ± 0.003	0.364 ± 0.002	0.367 ± 0.001	0.363 ± 0.001	[0.294, 0.409]
	mtlp2	**0.352**	0.360 ± 0.003	0.362 ± 0.003	0.362 ± 0.001	0.358 ± 0.001	[0.308, 0.436]
	mtlp3	**0.409**	0.419 ± 0.003	0.416 ± 0.002	0.417 ± 0.001	0.416 ± 0.002	[0.315, 0.442]
	mtlp4	**0.417**	0.427 ± 0.002	0.426 ± 0.003	0.426 ± 0.001	0.423 ± 0.002	[0.325, 0.461]
	mtlp5	0.495	**0.491 ± 0.006**	0.492 ± 0.006	0.502 ± 0.002	0.492 ± 0.003	[0.349, 0.530]
	mtlp6	0.534	**0.524 ± 0.008**	0.527 ± 0.006	0.537 ± 0.002	0.527 ± 0.002	[0.347, 0.540]
	mtlp7	0.531	**0.519 ± 0.008**	0.523 ± 0.006	0.534 ± 0.002	0.523 ± 0.003	[0.338, 0.526]
	mtlp8	0.542	**0.536 ± 0.010**	0.540 ± 0.008	0.547 ± 0.002	0.537 ± 0.003	[0.345, 0.504]
	mtlp9	**0.385**	0.395 ± 0.003	0.395 ± 0.002	0.390 ± 0.001	0.390 ± 0.002	[0.323, 0.456]
	mtlp10	**0.389**	0.398 ± 0.003	0.397 ± 0.003	0.394 ± 0.002	0.394 ± 0.001	[0.339, 0.456]
	mtlp11	**0.424**	0.430 ± 0.003	0.429 ± 0.003	0.426 ± 0.001	0.426 ± 0.001	[0.327, 0.445]
	mtlp12	**0.420**	0.422 ± 0.003	0.421 ± 0.004	0.423 ± 0.001	0.421 ± 0.002	[0.350, 0.466]
	mtlp13	**0.349**	0.358 ± 0.004	0.354 ± 0.004	0.351 ± 0.001	0.351 ± 0.001	[0.322, 0.419]
	mtlp14	**0.347**	0.364 ± 0.004	0.363 ± 0.003	0.350 ± 0.001	0.355 ± 0.002	[0.356, 0.472]
	mtlp15	**0.361**	0.371 ± 0.004	0.370 ± 0.003	0.363 ± 0.001	0.364 ± 0.001	[0.314, 0.406]
	mtlp16	**0.376**	0.382 ± 0.003	0.384 ± 0.003	**0.376 ± 0.001**	0.378 ± 0.001	[0.322, 0.407]
scm20d	Mean	0.755	0.754 ± 0.003	0.754 ± 0.003	**0.753 ± 0.001**	0.754 ± 0.002	[0.394, 0.763]
	lbl	**0.613**	0.618 ± 0.002	0.618 ± 0.002	0.614 ± 0.001	**0.613 ± 0.001**	[0.356, 0.678]
	mtlp2a	**0.628**	0.635 ± 0.002	0.634 ± 0.003	0.632 ± 0.001	0.631 ± 0.002	[0.352, 0.688]
	mtlp3a	**0.603**	0.608 ± 0.002	0.608 ± 0.003	0.607 ± 0.001	0.605 ± 0.002	[0.363, 0.683]
	mtlp4a	**0.635**	0.645 ± 0.002	0.645 ± 0.003	0.644 ± 0.001	0.638 ± 0.002	[0.374, 0.730]
	mtlp5a	**0.974**	0.977 ± 0.008	0.977 ± 0.007	0.978 ± 0.003	0.975 ± 0.006	[0.413, 0.846]
	mtlp6a	**0.981**	0.986 ± 0.009	0.992 ± 0.008	1.002 ± 0.004	0.989 ± 0.008	[0.424, 0.843]
	mtlp7a	**0.996**	1.001 ± 0.008	1.004 ± 0.007	1.005 ± 0.006	1.000 ± 0.009	[0.404, 0.833]
	mtlp8a	0.995	0.997 ± 0.010	0.997 ± 0.011	1.008 ± 0.005	**0.994 ± 0.005**	[0.407, 0.851]
	mtlp9a	0.708	0.704 ± 0.003	0.702 ± 0.003	**0.698 ± 0.001**	0.705 ± 0.002	[0.382, 0.737]
	mtlp10a	0.718	0.722 ± 0.004	0.722 ± 0.004	**0.716 ± 0.001**	0.723 ± 0.003	[0.413, 0.753]
	mtlp11a	0.729	0.730 ± 0.003	0.729 ± 0.003	**0.725 ± 0.001**	0.728 ± 0.002	[0.402, 0.769]
	mtlp12a	0.720	0.718 ± 0.004	0.717 ± 0.004	**0.712 ± 0.002**	0.716 ± 0.003	[0.429, 0.787]
	mtlp13a	0.711	0.703 ± 0.005	0.699 ± 0.004	**0.697 ± 0.001**	0.705 ± 0.003	[0.400, 0.751]
	mtlp14a	0.683	0.673 ± 0.004	0.670 ± 0.003	**0.668 ± 0.001**	0.675 ± 0.002	[0.411, 0.779]
	mtlp15a	0.684	0.674 ± 0.004	0.671 ± 0.004	**0.666 ± 0.001**	0.678 ± 0.002	[0.384, 0.727]
	mtlp16a	0.689	0.677 ± 0.005	0.676 ± 0.005	**0.672 ± 0.001**	0.682 ± 0.003	[0.386, 0.754]

We compare our non-sketched framework with the sketched one, and we furthermore compare our p-sparsified sketches with Accumulation sketch from Chen and Yang (2021a) and CountSketch from Clarkson and Woodruff (2017). Moreover, we report the range of results obtained by SOTA methods available at Spyromitros-Xioufis et al. (2016). All results in terms of Test RRMSE are reported in Table .2, we see that our p-sparsified sketches allow to ally statistical and computational performance, since we maintain an accuracy of the same order as without sketching, and these sketches outperform Accumulation in terms of training times (see table 3.4). In comparison to SOTA, our framework does not compete with the best results obtained in Spyromitros-Xioufis et al. (2016), but almost always remains within the range of results obtained with SOTA methods.

B Appendices for Chapter 4

B.1 Notations And Definitions

In this section, we remind some important notations and definitions.

Setting. In the following, we consider $\mathcal{X}$ and $\mathcal{Y}$ to be Polish spaces. We denote by ρ the unknown data distribution on $\mathcal{X} \times \mathcal{Y}$. We denote by $\rho_{\mathcal{X}}$ and $\rho_{\mathcal{Y}}$ the marginal distributions of the inputs and outputs, respectively.

Linear algebra notation. For an operator A, $A^{\#}$ is its adjoint, $\sigma_{\max}(A)$ its largest eigenvalue, and $\sigma_k(A)$ its k^{th} largest eigenvalue (if A admits an eigendecomposition). Let $\mathcal{B}\left(E\right)$ be the space of bounded linear operators in a separable Hilbert space E, given positive semi-definite operators $A, B \in \mathcal{B}\left(E\right)$, $A \preceq B$ if $B - A$ is positive semidefinite. For any $t > 0$ and $A : E \to E$, $A_t = A + tI_E$. Let M be a matrix, $M_{i:}$ denotes its i^{th} row and $M_{:j}$ its j^{th} column, and $M^{\dagger}$ denotes its Moore-Penrose inverse.

Notation for simplified bounds. To keep the dependencies of a bound only in the parameters of interest, for $a, b \in \mathbb{R}$ we note $a \lesssim b$ as soon as there exists a constant $c > 0$ independents of the parameters of interest such that $a \leq c \times b$.

Least-squares notation. For any function $h : \mathcal{X} \to \mathcal{H}_{\mathcal{Y}}$, its least-squares expected risk is given by

$$\mathcal{E}\left(h\right) = \mathbb{E}_{\rho}\left[\left\|h\left(x\right) - \psi_{\mathcal{Y}}\left(y\right)\right\|_{\mathcal{H}_{\mathcal{Y}}}^{2}\right]. \tag{243}$$

The measurable minimizer of $\mathcal{E}$ is given by $h^*(x) = \mathbb{E}_{\rho\left(y|x\right)}\left[\psi_{\mathcal{Y}}\left(y\right)\right]$ (Ciliberto et al., 2020, Lemma A.2).

RKHS notation. We denote by $\mathcal{H}_{\mathcal{X}}$ and $\mathcal{H}_{\mathcal{Y}}$ the RKHSs associated to the input $\mathrm{k}_{\mathcal{X}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and output $\mathrm{k}_{\mathcal{Y}} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ kernels, respectively. We denote by $\psi_{\mathcal{X}} : \mathcal{X} \to \mathcal{H}_{\mathcal{X}}$ and $\psi_{\mathcal{Y}} : \mathcal{Y} \to \mathcal{H}_{\mathcal{Y}}$ the canonical feature maps $\psi_{\mathcal{X}}(x) = \mathrm{k}_{\mathcal{X}}(x, .)$ and $\psi_{\mathcal{Y}}(y) = \mathrm{k}_{\mathcal{Y}}(y, .)$, respectively. We denote by $\mathcal{H}$ the vv-RKHS associated to the operator-valued kernel $\mathcal{K} = kI_{\mathcal{H}_{\mathcal{Y}}}$. We denote $\hat{h} \in \mathcal{H}$ the KRR estimator trained with n couples $(x_i, y_i)_{i=1}^{n}$ i.i.d. from ρ.

Kernel ridge operators. We define the following operators.

- $S : f \in \mathcal{H}_{\mathcal{X}} \mapsto \langle f, \psi_{\mathcal{X}}(\cdot)\rangle_{\mathcal{H}_{\mathcal{X}}} \in L^2\left(\mathcal{X}, \rho_{\mathcal{X}}\right)$
- $T : f \in \mathcal{H}_{\mathcal{Y}} \mapsto \langle f, h^*(\cdot)\rangle_{\mathcal{H}_{\mathcal{Y}}} \in L^2\left(\mathcal{X}, \rho_{\mathcal{X}}\right)$
- $\mathrm{C}_{\mathcal{X}} = \mathbb{E}_x\left[\psi_{\mathcal{X}}(x) \otimes \psi_{\mathcal{X}}(x)\right]$ and $\mathrm{C}_{\mathcal{Y}} = \mathbb{E}_y\left[\psi_{\mathcal{Y}}(y) \otimes \psi_{\mathcal{Y}}(y)\right]$,
- $\widehat{\mathrm{C}}_{\mathrm{X}} = \frac{1}{n}\sum_{i=1}^{n} \psi_{\mathcal{X}}(x_i) \otimes \psi_{\mathcal{X}}(x_i)$ and $\widehat{\mathrm{C}}_{\mathrm{Y}} = \frac{1}{n}\sum_{i=1}^{n} \psi_{\mathcal{Y}}(y_i) \otimes \psi_{\mathcal{Y}}(y_i)$,
- $\mathrm{S}_{\mathrm{X}} : f \in \mathcal{H}_{\mathcal{X}} \mapsto \frac{1}{\sqrt{n}}\left(f\left(x_1\right), \ldots, f\left(x_n\right)\right)^{\top} \in \mathbb{R}^n$,

- $\mathrm{S_X}^{\#} : \alpha \in \mathbb{R}^n \mapsto \frac{1}{\sqrt{n}} \sum_{i=1}^n \alpha_i \, \psi_{\mathcal{X}}(x_i) \in \mathcal{H}_{\mathcal{X}}$,
- $\mathrm{S_Y} : f \in \mathcal{H}_{\mathcal{Y}} \mapsto \frac{1}{\sqrt{n}} \Big(f\left(y_1\right), \ldots, f\left(y_n\right) \Big)^\top \in \mathbb{R}^n$,
- $\mathrm{S_Y}^{\#} : \alpha \in \mathbb{R}^n \mapsto \frac{1}{\sqrt{n}} \sum_{i=1}^n \alpha_i \, \psi_{\mathcal{Y}}(y_i) \in \mathcal{H}_{\mathcal{X}}$,

Sketching operators.

- We denote $\mathrm{R}_{\mathcal{X}} \in \mathbb{R}^{\mathrm{m}_{\mathcal{X}} \times n}$ and $\mathrm{R}_{\mathcal{Y}} \in \mathbb{R}^{\mathrm{m}_{\mathcal{Y}} \times n}$ the input and output sketch matrices with $\mathrm{m}_{\mathcal{X}} < n$ and $\mathrm{m}_{\mathcal{Y}} < n$,
- $\widetilde{\mathrm{C}}_{\mathrm{X}} = \mathrm{S_X}^{\#} \mathrm{R}_{\mathcal{X}}{}^\top \mathrm{R}_{\mathcal{X}} \mathrm{S_X}$ and $\widetilde{\mathrm{C}}_{\mathrm{Y}} = \mathrm{S_Y}^{\#} \mathrm{R}_{\mathcal{Y}}{}^\top \mathrm{R}_{\mathcal{Y}} \mathrm{S_Y}$,
- $\widetilde{\mathrm{K}}_{\mathrm{X}} = \mathrm{R}_{\mathcal{X}} \mathrm{K_X} \mathrm{R}_{\mathcal{X}}{}^\top$ and $\widetilde{\mathrm{K}}_{\mathrm{Y}} = \mathrm{R}_{\mathcal{Y}} \mathrm{K_Y} \mathrm{R}_{\mathcal{Y}}{}^\top$.

B.2 Preliminary Results

In this section, we present useful preliminary results about kernel ridge operators and sketching properties, as well as the proof Proposition 4.2 that gives the expressions of the SISOKR estimator.

Useful kernel ridge operators properties. The following results hold true.

- $\widehat{\mathrm{C}}_{\mathrm{X}} = \frac{1}{n} \sum_{i=1}^n \psi_{\mathcal{X}}(x_i) \otimes \psi_{\mathcal{X}}(x_i) = \mathrm{S_X}^{\#} \mathrm{S_X}$ and $\widehat{\mathrm{C}}_{\mathrm{Y}} = \frac{1}{n} \sum_{i=1}^n \psi_{\mathcal{Y}}(y_i) \otimes \psi_{\mathcal{Y}}(y_i) = \mathrm{S_Y}^{\#} \mathrm{S_Y}$,
- $\mathrm{K_X} = n \mathrm{S_X} \mathrm{S_X}^{\#}$ and $\mathrm{K_Y} = n \mathrm{S_Y} \mathrm{S_Y}^{\#}$,
- Under the attainability assumption (Ciliberto et al., 2020, Lemma B.2, B.4, B.9) show that:
  - For all $x \in \mathcal{X}$, $\hat{h}(x) = \widehat{H} \, \psi_{\mathcal{X}}(x)$, where $\widehat{H} = \mathrm{S_Y}^{\#} \mathrm{S_X} \, {\widehat{\mathrm{C}}_{\mathrm{X}\lambda}}^{-1}$.
  - $\mathbb{E}[\|\hat{h}(x) - h^*(x)\|^2]^{1/2} = \|(\widehat{H} - H) S^{\#}\|_{\mathrm{HS}}$.

Useful sketching properties. We remind some useful notations and provide the expression of $\widetilde{\mathrm{P}}_{\mathrm{Z}}$, leading to the expression of the SISOKR estimator.

Expression of $\widetilde{\mathrm{P}}_{\mathrm{Z}}$. Let $\left\{ \left(\sigma_i(\widetilde{\mathrm{K}}_{\mathrm{Z}}), \tilde{\mathbf{u}}_i^Z \right), i \in [\mathrm{m}_{\mathcal{Z}}] \right\}$ be the eigenpairs of $\widetilde{\mathrm{K}}_{\mathrm{Z}}$ ranked in descending order of eigenvalues, $\mathrm{p_Z} = \mathrm{rank}\left(\widetilde{\mathrm{K}}_{\mathrm{Z}} \right)$, and for all $\tilde{e}_i^Z = \sqrt{\frac{n}{\sigma_i(\widetilde{\mathrm{K}}_{\mathrm{Z}})}} \mathrm{S_Z}^{\#} \mathrm{R}_{\mathcal{Z}}{}^\top \tilde{\mathbf{u}}_i^Z$, for all $1 \leq i \leq \mathrm{p_Z}$.

Proposition 4.1 (Expression of the orthogonal projector)**.** *The $\tilde{e}_i^Z$s are the eigenfunctions, associated to the eigenvalues $\sigma_i(\widetilde{\mathrm{K}}_{\mathrm{Z}})/n$ of $\widetilde{\mathrm{C}}_{\mathrm{Z}}$. Furthermore, let $\widetilde{\mathcal{H}}_{\mathcal{Z}} = \mathrm{span}\left(\tilde{e}_1^z, \ldots, \tilde{e}_{\mathrm{p_Z}}^z \right)$, the orthogonal projector $\widetilde{\mathrm{P}}_{\mathrm{Z}}$ onto $\widetilde{\mathcal{H}}_{\mathcal{Z}}$ writes as*

$$\widetilde{\mathrm{P}}_{\mathrm{Z}} = (\mathrm{R}_{\mathcal{Z}} \mathrm{S_Z})^{\#} \left(\mathrm{R}_{\mathcal{Z}} \mathrm{S_Z} (\mathrm{R}_{\mathcal{Z}} \mathrm{S_Z})^{\#} \right)^{\dagger} \mathrm{R}_{\mathcal{Z}} \mathrm{S_Z} \, . \tag{4.8}$$

Proof For $1 \leq i \leq \mathrm{p_Z}$

$$\widetilde{\mathrm{C}}_{\mathrm{Z}}\ \tilde{e}_i^Z = \mathrm{S_Z}^{\#}\mathrm{R_Z}^{\top}\mathrm{R_Z}\mathrm{S_Z}\left(\sqrt{\frac{n}{\sigma_i(\widetilde{\mathrm{K}}_{\mathrm{Z}})}}\mathrm{S_Z}^{\#}\mathrm{R_Z}^{\top}\tilde{\mathbf{u}}_i^Z\right) \tag{244}$$

$$= \sqrt{\frac{n}{\sigma_i(\widetilde{\mathrm{K}}_{\mathrm{Z}})}}\mathrm{S_Z}^{\#}\mathrm{R_Z}^{\top}\left(\frac{1}{n}\widetilde{\mathrm{K}}_{\mathrm{Z}}\right)\tilde{\mathbf{u}}_i^Z \tag{245}$$

$$= \frac{1}{\sqrt{n\sigma_i(\widetilde{\mathrm{K}}_{\mathrm{Z}})}}\mathrm{S_Z}^{\#}\mathrm{R_Z}^{\top}\sigma_i(\widetilde{\mathrm{K}}_{\mathrm{Z}})\tilde{\mathbf{u}}_i^Z \tag{246}$$

$$= \frac{\sigma_i(\widetilde{\mathrm{K}}_{\mathrm{Z}})}{n}\tilde{e}_i^Z\,. \tag{247}$$

Moreover, we verify that $\mathrm{span}\left(\tilde{e}_1^Z,\ldots,\tilde{e}_{\mathrm{p_Z}}^Z\right)$ forms an orthonormal basis. Let $1 \leq i,j \leq \mathrm{p_Z}$,

$$\left\langle \tilde{e}_i^Z, \tilde{e}_j^Z \right\rangle_{\mathcal{H}_{\mathcal{X}}} = \left\langle \sqrt{\frac{n}{\sigma_i(\widetilde{\mathrm{K}}_{\mathrm{Z}})}}\mathrm{S_Z}^{\#}\mathrm{R_Z}^{\top}\tilde{\mathbf{u}}_i^Z, \sqrt{\frac{n}{\sigma_j(\widetilde{\mathrm{K}}_{\mathrm{Z}})}}\mathrm{S_Z}^{\#}\mathrm{R_Z}^{\top}\tilde{\mathbf{u}}_j^Z \right\rangle_{\mathcal{H}_{\mathcal{Z}}} \tag{248}$$

$$= \frac{n}{\sqrt{\sigma_i(\widetilde{\mathrm{K}}_{\mathrm{Z}})\sigma_j(\widetilde{\mathrm{K}}_{\mathrm{Z}})}}\tilde{\mathbf{u}}_i^{Z\top}\mathrm{R_Z}\mathrm{S_Z}\mathrm{S_Z}^{\#}\mathrm{R_Z}^{\top}\tilde{\mathbf{u}}_j^Z \tag{249}$$

$$= \frac{n}{\sqrt{\sigma_i(\widetilde{\mathrm{K}}_{\mathrm{Z}})\sigma_j(\widetilde{\mathrm{K}}_{\mathrm{Z}})}}\tilde{\mathbf{u}}_i^{Z\top}\left(\frac{1}{n}\widetilde{\mathrm{K}}_{\mathrm{Z}}\right)\tilde{\mathbf{u}}_j^Z \tag{250}$$

$$= \frac{\sigma_j(\widetilde{\mathrm{K}}_{\mathrm{Z}})}{\sqrt{\sigma_i(\widetilde{\mathrm{K}}_{\mathrm{Z}})\sigma_j(\widetilde{\mathrm{K}}_{\mathrm{Z}})}}\tilde{\mathbf{u}}_i^{Z\top}\tilde{\mathbf{u}}_j^Z \tag{251}$$

$$= \delta_{ij}\,, \tag{252}$$

where $\delta_{ij} = 0$ if $i \neq j$, and 1 otherwise.

Finally, it is easy to check that the orthogonal projector onto $\mathrm{span}\left(\tilde{e}_1^Z,\ldots,\tilde{e}_{\mathrm{p_Z}}^Z\right)$, i.e. $\widetilde{\mathrm{P}}_{\mathrm{Z}} : f \in \mathcal{H}_{\mathcal{Z}} \mapsto \sum_{i=1}^{\mathrm{p_Z}} \left\langle f, \tilde{e}_i^Z \right\rangle_{\mathcal{H}_{\mathcal{Z}}} \tilde{e}_i^Z$ rewrites as

$$\widetilde{\mathrm{P}}_{\mathrm{Z}} = n\mathrm{S_Z}^{\#}\mathrm{R_Z}^{\top}\widetilde{\mathrm{K}}_{\mathrm{Z}}^{\dagger}\mathrm{R_Z}\mathrm{S_Z} = (\mathrm{R_Z}\mathrm{S_Z})^{\#}\left(\mathrm{R_Z}\mathrm{S_Z}(\mathrm{R_Z}\mathrm{S_Z})^{\#}\right)^{\dagger}\mathrm{R_Z}\mathrm{S_Z}\,. \tag{253}$$

■

Remark .12. *With $\mathrm{R}_{\mathcal{X}}$ a sub-sampling matrix, we recover the linear operator L_m introduced in Yang et al. (2012) for the study of Nyström approximation and its eigendecomposition. Moreover, we also recover the projection operator P_m from Rudi et al. (2015) and follow the footsteps of the proposed extension "Nyström with sketching matrices".*

Algorithm. We here give the proof of Proposition 4.2 that provides an expression of the SISOKR estimator $\tilde{h}$ as a linear combination of the $\psi_{\mathcal{Y}}(y_i)$s.

Proposition 4.2 (Expression of SISOKR). $\forall\, x \in \mathcal{X}$,

$$\tilde{h}(x) = \sum_{i=1}^{n} \tilde{\alpha}_i(x)\, \psi_{\mathcal{Y}}\left(y_i\right), \tag{4.9}$$

where $\tilde{\alpha}(x) = \mathrm{R}_{\mathcal{Y}}{}^{\top}\, \widetilde{\Omega}\, \mathrm{R}_{\mathcal{X}}\, \mathrm{k}_{\mathrm{X}}^{x}$ *and*

$$\widetilde{\Omega} = \widetilde{\mathrm{K}}_{\mathrm{Y}}{}^{\dagger}\, \mathrm{R}_{\mathcal{Y}}\, \mathrm{K}_{\mathrm{Y}}\, \mathrm{K}_{\mathrm{X}}\, \mathrm{R}_{\mathcal{X}}{}^{\top} (\mathrm{R}_{\mathcal{X}}\, \mathrm{K}_{\mathrm{X}}{}^{2}\, \mathrm{R}_{\mathcal{X}}{}^{\top} + n\lambda\, \widetilde{\mathrm{K}}_{\mathrm{X}})^{\dagger}, \tag{4.10}$$

with $\widetilde{\mathrm{K}}_{\mathrm{X}} = \mathrm{R}_{\mathcal{X}}\, \mathrm{K}_{\mathrm{X}}\, \mathrm{R}_{\mathcal{X}}{}^{\top}$ *and* $\widetilde{\mathrm{K}}_{\mathrm{Y}} = \mathrm{R}_{\mathcal{Y}}\, \mathrm{K}_{\mathrm{Y}}\, \mathrm{R}_{\mathcal{Y}}{}^{\top}$.

Proof Recall that $\tilde{h}(x) = \widetilde{\mathrm{P}}_{\mathrm{Y}}\, \mathrm{S}_{\mathrm{Y}}{}^{\#}\, \mathrm{S}_{\mathrm{X}}\, \widetilde{\mathrm{P}}_{\mathrm{X}} (\widetilde{\mathrm{P}}_{\mathrm{X}}\, \mathrm{S}_{\mathrm{X}}{}^{\#}\, \mathrm{S}_{\mathrm{X}}\, \widetilde{\mathrm{P}}_{\mathrm{X}} + \lambda I_{\mathcal{H}_{\mathcal{X}}})^{-1}\, \psi_{\mathcal{X}}(x)$. By lemma .13 and especially (257), we obtain that

$$\tilde{h}(x) = \sqrt{n}\, \widetilde{\mathrm{P}}_{\mathrm{Y}}\, \mathrm{S}_{\mathrm{Y}}{}^{\#}\, \mathrm{K}_{\mathrm{X}}\, \mathrm{R}_{\mathcal{X}}{}^{\top} \left(\mathrm{R}_{\mathcal{X}}\, \mathrm{K}_{\mathrm{X}}{}^{2}\, \mathrm{R}_{\mathcal{X}}{}^{\top} + n\lambda\, \mathrm{R}_{\mathcal{X}}\, \mathrm{K}_{\mathrm{X}}\, \mathrm{R}_{\mathcal{X}}{}^{\top}\right)^{\dagger} \mathrm{R}_{\mathcal{X}}\, \mathrm{S}_{\mathrm{X}}\, \psi_{\mathcal{X}}(x). \tag{254}$$

Finally, by lemma .14, we have that $\tilde{h}(x) = \sum_{i=1}^{n} \tilde{\alpha}_i(x)\, \psi_{\mathcal{Y}}\left(y_i\right)$ where

$$\tilde{\alpha}(x) = \mathrm{R}_{\mathcal{Y}}{}^{\top}\, \widetilde{\mathrm{K}}_{\mathrm{Y}}{}^{\dagger}\, \mathrm{R}_{\mathcal{Y}}\, \mathrm{K}_{\mathrm{Y}}\, \mathrm{K}_{\mathrm{X}}\, \mathrm{R}_{\mathcal{X}}{}^{\top} (\mathrm{R}_{\mathcal{X}}\, \mathrm{K}_{\mathrm{X}}{}^{2}\, \mathrm{R}_{\mathcal{X}}{}^{\top} + n\lambda\, \widetilde{\mathrm{K}}_{\mathrm{X}})^{\dagger}\, \mathrm{R}_{\mathcal{X}}\, \mathrm{k}_{\mathrm{X}}^{x}. \tag{255}$$

■

Before stating and proving lemmas .13 and .14, and similarly to Rudi et al. (2015), let $\mathrm{R}_{\mathcal{X}}\, \mathrm{S}_{\mathrm{X}} = U \Sigma V^{\#}$ be the SVD of $\mathrm{R}_{\mathcal{X}}\, \mathrm{S}_{\mathrm{X}}$ where $U : \mathbb{R}^{\mathrm{p_X}} \to \mathbb{R}^{\mathrm{m}_{\mathcal{X}}}$, $\Sigma : \mathbb{R}^{\mathrm{p_X}} \to \mathbb{R}^{\mathrm{p_X}}$, $V : \mathbb{R}^{\mathrm{p_X}} \to \mathcal{H}_{\mathcal{X}}$, and $\Sigma = \mathrm{diag}(\sigma_1(\mathrm{R}_{\mathcal{X}}\, \mathrm{S}_{\mathrm{X}}), \ldots, \sigma_{\mathrm{p_X}}(\mathrm{R}_{\mathcal{X}}\, \mathrm{S}_{\mathrm{X}}))$ with $\sigma_1(\mathrm{R}_{\mathcal{X}}\, \mathrm{S}_{\mathrm{X}}) \geq \ldots \geq \sigma_{\mathrm{p_X}}(\mathrm{R}_{\mathcal{X}}\, \mathrm{S}_{\mathrm{X}}) > 0$, $UU^{\top} = I_{\mathrm{p_X}}$ and $V^{\#}V = I_{\mathrm{p_X}}$. We are now ready to prove the following lemma for the expansion induced by input sketching.

Lemma .13. *Let* $\widetilde{H} = \widetilde{\mathrm{P}}_{\mathrm{Y}}\, \mathrm{S}_{\mathrm{Y}}{}^{\#}\, \mathrm{S}_{\mathrm{X}}\, \widetilde{\mathrm{P}}_{\mathrm{X}} (\widetilde{\mathrm{P}}_{\mathrm{X}}\, \mathrm{S}_{\mathrm{X}}{}^{\#}\, \mathrm{S}_{\mathrm{X}}\, \widetilde{\mathrm{P}}_{\mathrm{X}} + \lambda I_{\mathcal{H}_{\mathcal{X}}})^{-1}$. *The following two expansions hold true*

$$\widetilde{H} = \widetilde{\mathrm{P}}_{\mathrm{Y}}\, \mathrm{S}_{\mathrm{Y}}{}^{\#}\, \mathrm{S}_{\mathrm{X}}\, \tilde{\eta}(\widehat{\mathrm{C}}_{\mathrm{X}}), \tag{256}$$

where $\tilde{\eta}(\widehat{\mathrm{C}}_{\mathrm{X}}) = V(V^{\#}\, \widehat{\mathrm{C}}_{\mathrm{X}}\, V + \lambda I_{\mathcal{H}_{\mathcal{X}}})^{-1}\, V^{\#}$ *and for algorithmic purposes*

$$\widetilde{H} = \sqrt{n}\, \widetilde{\mathrm{P}}_{\mathrm{Y}}\, \mathrm{S}_{\mathrm{Y}}{}^{\#}\, \mathrm{K}_{\mathrm{X}}\, \mathrm{R}_{\mathcal{X}}{}^{\top} \left(\mathrm{R}_{\mathcal{X}}\, \mathrm{K}_{\mathrm{X}}{}^{2}\, \mathrm{R}_{\mathcal{X}}{}^{\top} + n\lambda\, \mathrm{R}_{\mathcal{X}}\, \mathrm{K}_{\mathrm{X}}\, \mathrm{R}_{\mathcal{X}}{}^{\top}\right)^{\dagger} \mathrm{R}_{\mathcal{X}}\, \mathrm{S}_{\mathrm{X}}. \tag{257}$$

Proof Let us prove (256) first.

$$\begin{aligned}
\tilde{H} &= \widetilde{\mathrm{P}}_{\mathrm{Y}}\, \mathrm{S}_{\mathrm{Y}}{}^{\#}\, \mathrm{S}_{\mathrm{X}}\, \widetilde{\mathrm{P}}_{\mathrm{X}} (\widetilde{\mathrm{P}}_{\mathrm{X}}\, \mathrm{S}_{\mathrm{X}}{}^{\#}\, \mathrm{S}_{\mathrm{X}}\, \widetilde{\mathrm{P}}_{\mathrm{X}} + \lambda I_{\mathcal{H}_{\mathcal{X}}})^{-1} && (258)\\
&= \widetilde{\mathrm{P}}_{\mathrm{Y}}\, \mathrm{S}_{\mathrm{Y}}{}^{\#}\, \mathrm{S}_{\mathrm{X}}\, VV^{\#} (VV^{\#}\, \mathrm{S}_{\mathrm{X}}{}^{\#}\, \mathrm{S}_{\mathrm{X}}\, VV^{\#} + \lambda I_{\mathcal{H}_{\mathcal{X}}})^{-1} && (259)\\
&= \widetilde{\mathrm{P}}_{\mathrm{Y}}\, \mathrm{S}_{\mathrm{Y}}{}^{\#}\, \mathrm{S}_{\mathrm{X}}\, V(V^{\#} \hat{\mathrm{C}}_{\mathcal{X}}\, V + \lambda I_{\mathcal{H}_{\mathcal{X}}})^{-1}\, V^{\#} && (260)\\
&= \widetilde{\mathrm{P}}_{\mathrm{Y}}\, \mathrm{S}_{\mathrm{Y}}{}^{\#}\, \mathrm{S}_{\mathrm{X}}\, \tilde{\eta}(\widehat{\mathrm{C}}_{\mathrm{X}}), && (261)
\end{aligned}$$

using the so-called push-through identity $(I + UV)^{-1} U = U(I + VU)^{-1}$.

Now, we focus on proving (257). First, we have that

$$\tilde{H} = \widetilde{\mathrm{P}}_{\mathrm{Y}}\, \mathrm{S}_{\mathrm{Y}}{}^{\#}\, \mathrm{S}_{\mathrm{X}}\, V(V^{\#}\, \widehat{\mathrm{C}}_{\mathrm{X}\lambda}\, V)^{\dagger} V^{\#}. \tag{262}$$

Then, using the fact that U has orthonormal columns, $U^\top$ has orthonormal rows and Σ is a full-rank matrix, together with the fact that $UU^\top = I_{\mathrm{p_X}}$ and $V^\# V = I_{\mathrm{p_X}}$, we have that,

$$\tilde{H} = \widetilde{\mathrm{P}}_{\mathrm{Y}} \mathrm{S_Y}^\# \mathrm{S_X} V\Sigma U^\top \left(U\Sigma V^\# \widehat{\mathrm{C}}_{\mathrm{X}\lambda} V\Sigma U^\top\right)^\dagger U\Sigma V^\#. \tag{263}$$

Then, since $\mathrm{R}_{\mathcal{X}} \mathrm{S_X} = U\Sigma V^\#$,

$$\tilde{H} = \widetilde{\mathrm{P}}_{\mathrm{Y}} \mathrm{S_Y}^\# \mathrm{S_X} (\mathrm{R}_{\mathcal{X}} \mathrm{S_X})^\# \left(\mathrm{R}_{\mathcal{X}} \mathrm{S_X} \left(\widehat{\mathrm{C}}_{\mathrm{X}} + \lambda I_{\mathcal{H}_{\mathcal{X}}}\right) (\mathrm{R}_{\mathcal{X}} \mathrm{S_X})^\#\right)^\dagger \mathrm{R}_{\mathcal{X}} \mathrm{S_X}\,. \tag{264}$$

Finally, using the fact that $\widehat{\mathrm{C}}_{\mathrm{X}} = \mathrm{S_X}^\# \mathrm{S_X}$ and $\mathrm{K_X} = n \mathrm{S_X} \mathrm{S_X}^\#$, we obtain that

$$\widetilde{H} = \sqrt{n} \widetilde{\mathrm{P}}_{\mathrm{Y}} \mathrm{S_Y}^\# \mathrm{K_X} \mathrm{R}_{\mathcal{X}}{}^\top \left(\mathrm{R}_{\mathcal{X}} \mathrm{K_X}^2 \mathrm{R}_{\mathcal{X}}{}^\top + n\lambda \mathrm{R}_{\mathcal{X}} \mathrm{K_X} \mathrm{R}_{\mathcal{X}}{}^\top\right)^\dagger \mathrm{R}_{\mathcal{X}} \mathrm{S_X}\,. \tag{265}$$

■

Now we state and prove the lemma for the expansion induced by output sketching.

Lemma .14. *For all $x \in \mathcal{X}$, for any $h \in \mathcal{H}$ that writes as $h(x) = \sqrt{n} \widetilde{\mathrm{P}}_{\mathrm{Y}} \mathrm{S_Y}^\# \alpha(x)$ with $\alpha : \mathcal{X} \to \mathbb{R}^n$, then $h(x) = \sum_{i=1}^n \mathrm{R}_{\mathcal{Y}}{}^\top \widetilde{\mathrm{K}}_{\mathrm{Y}}{}^\dagger \mathrm{R}_{\mathcal{Y}} \mathrm{K_Y} \alpha(x) \psi_{\mathcal{Y}}(y_i)$.*

Proof

$$\begin{aligned} h(x) &= \sqrt{n} \widetilde{\mathrm{P}}_{\mathrm{Y}} \mathrm{S_Y}^\# \alpha(x) && (266)\\ &= \sqrt{n} \mathrm{S_Y}^\# \mathrm{R}_{\mathcal{Y}}{}^\top \widetilde{\mathrm{K}}_{\mathrm{Y}}{}^\dagger \mathrm{R}_{\mathcal{Y}} \left(n \mathrm{S_Y} \mathrm{S_Y}^\#\right) \alpha(x) && (267)\\ &= \sqrt{n} \mathrm{S_Y}^\# \mathrm{R}_{\mathcal{Y}}{}^\top \widetilde{\mathrm{K}}_{\mathrm{Y}}{}^\dagger \mathrm{R}_{\mathcal{Y}} \mathrm{K_Y} \alpha(x) && (268)\\ &= \sum_{i=1}^n \mathrm{R}_{\mathcal{Y}}{}^\top \widetilde{\mathrm{K}}_{\mathrm{Y}}{}^\dagger \mathrm{R}_{\mathcal{Y}} \mathrm{K_Y} \alpha(x) \psi_{\mathcal{Y}}(y_i). && (269) \end{aligned}$$

■

B.3 SISOKR Excess Risk Bound

In this section, we provide the proof of theorem 4.7 which gives a bound on the excess risk of the proposed approximated regression estimator with both input and output sketching (SISOKR).

Theorem 4.7 (SISOKR excess risk bound)**.** *Let $\delta \in (0,1]$, $n \in \mathbb{N}$ such that $\lambda = n^{-1/(1+\gamma_{\mathcal{X}})} \geq \frac{9\kappa_{\mathcal{X}}{}^2}{n} \log(\frac{n}{\delta})$. Under Assumptions 4.3 to 4.6, with probability $1-\delta$ we have*

$$\mathbb{E}_x \left[\|\tilde{h}(x) - h^*(x)\|^2_{\mathcal{H}_{\mathcal{Y}}}\right]^{\frac{1}{2}} \leq S(n,\delta) + c_2 A^{\psi_{\mathcal{X}}}_{\rho_{\mathcal{X}}}(\widetilde{P}_{\mathrm{X}}) + A^{\psi_{\mathcal{Y}}}_{\rho_{\mathcal{Y}}}(\widetilde{P}_{\mathrm{Y}}), \tag{4.12}$$

where $S(n,\delta) = c_1 \log(4/\delta) n^{-\frac{1}{2(1+\gamma_{\mathcal{X}})}}$ and

$$A^{\psi_{\mathcal{Z}}}_{\rho_{\mathcal{Z}}}(\widetilde{P}_{\mathrm{Z}}) = \mathbb{E}_z \left[\|(\widetilde{\mathrm{P}}_{\mathrm{Z}} - I_{\mathcal{H}_{\mathcal{Z}}}) \psi_{\mathcal{Z}}(z)\|^2_{\mathcal{H}_{\mathcal{Z}}}\right]^{\frac{1}{2}}, \tag{4.13}$$

with $c_1, c_2 > 0$ constants independent of n and δ.

Proof Our proofs consist of decompositions and then applications of the probabilistic bounds given in Appendix B.5.

We have

$$\mathbb{E}[\|\tilde{h}(x) - h^*(x)\|^2]^{1/2} = \|(\tilde{H} - H)S^{\#}\|_{\mathrm{HS}} \tag{270}$$

with $\tilde{H} = \widetilde{P}_Y \, \mathrm{S_Y}^{\#} \, \mathrm{S_X} \, \tilde{\eta}(\widehat{\mathrm{C}}_\mathrm{X})$.

Then, defining $H_\lambda = H \, \mathrm{C}_\mathcal{X}(\mathrm{C}_\mathcal{X} + \lambda I)^{-1}$, we decompose

$$\tilde{H} - H = \widetilde{P}_Y \left(\mathrm{S_Y}^{\#} \mathrm{S_X} - H_\lambda \widehat{\mathrm{C}}_\mathrm{X} \right) \tilde{\eta}(\widehat{\mathrm{C}}_\mathrm{X}) + \widetilde{P}_Y H_\lambda \left(\widehat{\mathrm{C}}_\mathrm{X} \, \tilde{\eta}(\widehat{\mathrm{C}}_\mathrm{X}) - I_{\mathcal{H}_\mathcal{X}} \right) + \left(\widetilde{P}_Y H_\lambda - H \right) \tag{271}$$

such that

$$\|(\tilde{H} - H)S^{\#}\|_{\mathrm{HS}} \leq (A) + (B) + (C) \tag{272}$$

with

$$(A) = \left\| \left(\mathrm{S_Y}^{\#} \mathrm{S_X} - H_\lambda \widehat{\mathrm{C}}_\mathrm{X} \right) \tilde{\eta}(\widehat{\mathrm{C}}_\mathrm{X}) \mathrm{C}_\mathcal{X}^{1/2} \right\|_{\mathrm{HS}} \tag{273}$$

$$(B) = \left\| H_\lambda \left(\widehat{\mathrm{C}}_\mathrm{X} \, \tilde{\eta}(\widehat{\mathrm{C}}_\mathrm{X}) - I_{\mathcal{H}_\mathcal{X}} \right) \mathrm{C}_\mathcal{X}^{1/2} \right\|_{\mathrm{HS}} \tag{274}$$

$$(C) = \left\| (\widetilde{P}_Y H_\lambda - H) \, \mathrm{C}_\mathcal{X}^{1/2} \right\|_{\mathrm{HS}} \tag{275}$$

Then, from lemmas .15 to .17, we obtain

$$\|(\tilde{H} - H)S^{\#}\|_{\mathrm{HS}} \leq 2\sqrt{3} M \log(4/\delta) n^{-\frac{1}{2(1+\gamma_\mathcal{X})}} + 2\sqrt{3} \|H\|_{\mathrm{HS}} \|(I - \widetilde{P}_\mathrm{X}) \, \mathrm{C}_\mathcal{X}^{1/2}\|_{\mathrm{op}} \tag{276}$$

$$+ \mathbb{E}_y \left[\left\| \left(\widetilde{\mathrm{P}}_\mathrm{Y} - I_{\mathcal{H}_\mathcal{Y}} \right) \psi_\mathcal{Y}(y) \right\|_{\mathcal{H}_\mathcal{Y}}^2 \right]^{1/2}. \tag{277}$$

Then, notice that

$$\left\| (I - \widetilde{P}_\mathrm{X}) \, \mathrm{C}_\mathcal{X}^{1/2} \right\|_{\mathrm{op}} \leq \left\| (I - \widetilde{P}_\mathrm{X}) \, \mathrm{C}_\mathcal{X}^{1/2} \right\|_{\mathrm{HS}} \tag{278}$$

$$= \mathbb{E}_x \left[\left\| \left(\widetilde{\mathrm{P}}_\mathrm{X} - I_{\mathcal{H}_\mathcal{X}} \right) \psi_\mathcal{X}(x) \right\|_{\mathcal{H}_\mathcal{X}}^2 \right]^{1/2}. \tag{279}$$

We conclude by defining

$$c_1 = 2\sqrt{3} M, \tag{280}$$

$$c_2 = 2\sqrt{3} \|H\|_{\mathrm{HS}}. \tag{281}$$

■

Lemma .15 (Bound (A)). *Let $\delta \in [0,1]$, $n \in \mathbb{N}$ sufficiently large such that $\lambda = n^{-1/(1+\gamma)} \geq \frac{9\kappa_\mathcal{X}^2}{n} \log(\frac{n}{x})$ Under our set of assumptions, the following holds with probability at least $1 - \delta$*

$$(A) \leq 2M \log(4/\delta) n^{-\frac{1}{2(1+\gamma_\mathcal{X})}}. \tag{282}$$

where the constant M depends on $\kappa_\mathcal{Y}, \|H\|_{\mathrm{HS}}, \delta$.

Proof

We have

$$(A) \leq \underbrace{\left\| \left(S_Y{}^{\#} S_X - H_\lambda \widehat{C}_X \right) C_{\mathcal{X}\lambda}{}^{-1/2} \right\|_{HS}}_{(A.1)} \times \underbrace{\| C_{\mathcal{X}\lambda}{}^{1/2} \tilde{\eta}(\widehat{C}_X) C_{\mathcal{X}}{}^{1/2} \|_{op}}_{(A.2)} \tag{283}$$

Moreover, we have

$$(A.2) \leq \| \widehat{C}_{X\lambda}{}^{1/2} \tilde{\eta}(\widehat{C}_X) \widehat{C}_{X\lambda}{}^{1/2} \|_{op} \| \widehat{C}_{X\lambda}{}^{-1/2} C_{\mathcal{X}\lambda}{}^{1/2} \|_{op}^2 \| C_{\mathcal{X}\lambda}{}^{-1/2} C_{\mathcal{X}}{}^{1/2} \|_{op} \tag{284}$$

$$\leq \| \widehat{C}_{X\lambda}{}^{1/2} \tilde{\eta}(\widehat{C}_X) \widehat{C}_{X\lambda}{}^{1/2} \|_{op} \| \widehat{C}_{X\lambda}{}^{-1/2} C_{\mathcal{X}\lambda}{}^{1/2} \|_{op}^2 \tag{285}$$

because $\| C_{\mathcal{X}\lambda}{}^{-1/2} C_{\mathcal{X}}{}^{1/2} \|_{op} \leq 1$.

Finally, by using the probabilistic bounds given in lemmas .20 and .21, and Lemma .27, we obtain

$$(A) \leq 2M \log(4/\delta) n^{-\frac{1}{2(1+\gamma_{\mathcal{X}})}}. \tag{286}$$

■

Lemma .16 (Bound (B)). *If* $\frac{9}{n} \log \frac{n}{\delta} \leq \lambda \leq \| C_{\mathcal{X}} \|_{op}$, *then with probability* $1 - \delta$

$$(B) \leq 2\sqrt{3} \|H\|_{HS} (\lambda^{1/2} + \| (I - \widetilde{P}_X) C_{\mathcal{X}}{}^{1/2} \|_{op}) \tag{287}$$

Proof

We do a similar decomposition than in Rudi et al. (2015, Theorem 2):

$$\widehat{C}_X \tilde{\eta}(\widehat{C}_X) - I_{\mathcal{H}_\mathcal{X}} = \widehat{C}_{X\lambda} \tilde{\eta}(\widehat{C}_X) - \lambda \tilde{\eta}(\widehat{C}_X) - I_{\mathcal{H}_\mathcal{X}} \tag{288}$$

$$= (I - \widetilde{P}_X) \widehat{C}_{X\lambda} \tilde{\eta}(\widehat{C}_X) + \widetilde{P}_X \widehat{C}_{X\lambda} \tilde{\eta}(\widehat{C}_X) - \lambda \tilde{\eta}(\widehat{C}_X) - I_{\mathcal{H}_\mathcal{X}} \tag{289}$$

$$= (I - \widetilde{P}_X) \widehat{C}_{X\lambda} \tilde{\eta}(\widehat{C}_X) - \lambda \tilde{\eta}(\widehat{C}_X) - (\widetilde{P}_X - I_{\mathcal{H}_\mathcal{X}}), \tag{290}$$

as $\widetilde{P}_X \widehat{C}_{X\lambda} \tilde{\eta}(\widehat{C}_X) = \widetilde{P}_X$.

Then, we have

$$(B) \leq \left\| H_\lambda \right\|_{HS} \left\| \left(\widehat{C}_X \tilde{\eta}(\widehat{C}_X) - I_{\mathcal{H}_\mathcal{X}} \right) C_{\mathcal{X}}{}^{1/2} \right\|_{op} \tag{291}$$

$$\leq \left\| H_\lambda \right\|_{HS} \left(\left\| (I - \widetilde{P}_X) \widehat{C}_{X\lambda} \tilde{\eta}(\widehat{C}_X) C_{\mathcal{X}}{}^{1/2} \right\|_{op} + \lambda \left\| \tilde{\eta}(\widehat{C}_X) C_{\mathcal{X}}{}^{1/2} \right\|_{op} + \left\| (\widetilde{P}_X - I_{\mathcal{H}_\mathcal{X}}) C_{\mathcal{X}}{}^{1/2} \right\|_{op} \right) \tag{292}$$

But,

$$\|H_\lambda\|_{\mathrm{HS}} \leq \left\|H\left(\mathrm{C}_{\mathcal{X}}\,{\mathrm{C}_{\mathcal{X}\lambda}}^{-1} - I_{\mathcal{H}_{\mathcal{X}}}\right)\right\|_{\mathrm{HS}} + \|H\|_{\mathrm{HS}} \tag{293}$$

$$= \left\|H\left(\mathrm{C}_{\mathcal{X}} - \mathrm{C}_{\mathcal{X}\lambda}\right){\mathrm{C}_{\mathcal{X}\lambda}}^{-1}\right\|_{\mathrm{HS}} + \|H\|_{\mathrm{HS}} \tag{294}$$

$$= \lambda\left\|H\,{\mathrm{C}_{\mathcal{X}\lambda}}^{-1}\right\|_{\mathrm{HS}} + \|H\|_{\mathrm{HS}} \tag{295}$$

$$\leq 2\|H\|_{\mathrm{HS}}. \tag{296}$$

And,

$$\left\|(I-\widetilde{P}_X)\widehat{\mathrm{C}}_{\mathrm{X}\lambda}\,\tilde{\eta}(\widehat{\mathrm{C}}_{\mathrm{X}})\,{\mathrm{C}_{\mathcal{X}}}^{1/2}\right\|_{\mathrm{op}} \leq \left\|(I-\widetilde{P}_X)\,{\widehat{\mathrm{C}}_{\mathrm{X}\lambda}}^{1/2}\right\|_{\mathrm{op}}\left\|{\widehat{\mathrm{C}}_{\mathrm{X}\lambda}}^{1/2}\,\tilde{\eta}(\widehat{\mathrm{C}}_{\mathrm{X}})\,{\widehat{\mathrm{C}}_{\mathrm{X}\lambda}}^{1/2}\right\|_{\mathrm{op}} \tag{297}$$

$$\cdot\left\|{\widehat{\mathrm{C}}_{\mathrm{X}\lambda}}^{-1/2}\,{\mathrm{C}_{\mathcal{X}}}^{1/2}\right\|_{\mathrm{op}}. \tag{298}$$

And,

$$\left\|(I-\widetilde{P}_X)\,{\widehat{\mathrm{C}}_{\mathrm{X}\lambda}}^{1/2}\right\|_{\mathrm{op}} \leq \left\|(I-\widetilde{P}_X)C_{X\lambda}^{1/2}\right\|_{\mathrm{op}}\left\|{\mathrm{C}_{\mathcal{X}\lambda}}^{-1/2}\,{\widehat{\mathrm{C}}_{\mathrm{X}\lambda}}^{1/2}\right\|_{\mathrm{op}}. \tag{299}$$

And,

$$\left\|(I-\widetilde{P}_X)\,{\mathrm{C}_{\mathcal{X}\lambda}}^{1/2}\right\|_{\mathrm{op}} \leq \left\|(I-\widetilde{P}_X)\,{\mathrm{C}_{\mathcal{X}}}^{1/2}\right\|_{\mathrm{op}} + \lambda^{1/2}. \tag{300}$$

Moreover,

$$\left\|\lambda\tilde{\eta}\left(\widehat{\mathrm{C}}_{\mathrm{X}}\right){\mathrm{C}_{\mathcal{X}}}^{1/2}\right\|_{\mathrm{op}} \leq \lambda\left\|{\widehat{\mathrm{C}}_{\mathrm{X}\lambda}}^{-1/2}\right\|_{\mathrm{op}}\left\|{\widehat{\mathrm{C}}_{\mathrm{X}\lambda}}^{1/2}\,\tilde{\eta}\left(\widehat{\mathrm{C}}_{\mathrm{X}}\right){\widehat{\mathrm{C}}_{\mathrm{X}\lambda}}^{1/2}\right\|_{\mathrm{op}} \tag{301}$$

$$\cdot\left\|{\widehat{\mathrm{C}}_{\mathrm{X}\lambda}}^{-1/2}\,{\mathrm{C}_{\mathcal{X}\lambda}}^{1/2}\right\|_{\mathrm{op}}\left\|{\mathrm{C}_{\mathcal{X}\lambda}}^{-1/2}\,{\mathrm{C}_{\mathcal{X}}}^{1/2}\right\|_{\mathrm{op}} \tag{302}$$

$$\leq \lambda^{1/2}\left\|{\widehat{\mathrm{C}}_{\mathrm{X}\lambda}}^{1/2}\,\tilde{\eta}\left(\widehat{\mathrm{C}}_{\mathrm{X}}\right){\widehat{\mathrm{C}}_{\mathrm{X}\lambda}}^{1/2}\right\|_{\mathrm{op}}\left\|{\widehat{\mathrm{C}}_{\mathrm{X}\lambda}}^{-1/2}\,{\mathrm{C}_{\mathcal{X}\lambda}}^{1/2}\right\|_{\mathrm{op}}. \tag{303}$$

Conclusion. Using the probabilistic bounds given in Lemmas .21, .22, and Lemma .27, we obtain

$$(B) \leq 4\sqrt{3}\|H\|_{\mathrm{HS}}\left(\lambda^{1/2} + \left\|(I-\widetilde{P}_X)\,{\mathrm{C}_{\mathcal{X}}}^{1/2}\right\|_{\mathrm{op}}\right) \tag{304}$$

■

Lemma .17 (Bound (C))**.** *We have*

$$(C) \leq \mathbb{E}_y\left[\left\|\left(\widetilde{\mathrm{P}}_{\mathrm{Y}} - I_{\mathcal{H}_y}\right)\psi_y(y)\right\|_{\mathcal{H}_y}^2\right]^{1/2} + \lambda^{1/2}\left\|H\right\|_{\mathrm{HS}}. \tag{305}$$

Proof We have

$$(C) = \left\|(\widetilde{P}_Y H(I_{\mathcal{H}_\mathcal{X}} - \lambda \mathrm{C}_\mathcal{X} \lambda^{-1}) - H)\mathrm{C}_\mathcal{X}^{1/2}\right\|_{\mathrm{HS}} \tag{306}$$

$$\leq \left\|(\widetilde{P}_Y - I_{\mathcal{H}_\mathcal{Y}})H\,\mathrm{C}_\mathcal{X}^{1/2}\right\|_{\mathrm{HS}} + \lambda^{1/2}\left\|H\right\|_{\mathrm{HS}} \tag{307}$$

$$= \mathbb{E}[\|(\widetilde{P}_Y - I_{\mathcal{H}_\mathcal{Y}})h^*(x)\|^2_{\mathcal{H}_\mathcal{Y}}]^{1/2} + \lambda^{1/2}\left\|H\right\|_{\mathrm{HS}}. \tag{308}$$

We conclude the proof as follows. Using the fact that $h^*(x) = \mathbb{E}_{\rho(y|x)}\left[\psi_\mathcal{Y}\left(y\right)\right]$, the linearity of $\widetilde{\mathrm{P}}_\mathrm{Y} - I_{\mathcal{H}_\mathcal{Y}}$ and the convexity of $\|\cdot\|^2_{\mathcal{H}_\mathcal{Y}}$, by the Jensen's inequality we obtain that

$$\mathbb{E}_x\left[\left\|\left(\widetilde{\mathrm{P}}_\mathrm{Y} - I_{\mathcal{H}_\mathcal{Y}}\right)h^*(x)\right\|^2_{\mathcal{H}_\mathcal{Y}}\right] = \mathbb{E}_x\left[\left\|\left(\widetilde{\mathrm{P}}_\mathrm{Y} - I_{\mathcal{H}_\mathcal{Y}}\right)\mathbb{E}_{\rho(y|x)}\left[\psi_\mathcal{Y}\left(y\right)\right]\right\|^2_{\mathcal{H}_\mathcal{Y}}\right] \tag{309}$$

$$= \mathbb{E}_x\left[\left\|\mathbb{E}_{\rho(y|x)}\left[\left(\widetilde{\mathrm{P}}_\mathrm{Y} - I_{\mathcal{H}_\mathcal{Y}}\right)\psi_\mathcal{Y}\left(y\right)\right]\right\|^2_{\mathcal{H}_\mathcal{Y}}\right] \tag{310}$$

$$\leq \mathbb{E}_x\left[\mathbb{E}_{\rho(y|x)}\left[\left\|\left(\widetilde{\mathrm{P}}_\mathrm{Y} - I_{\mathcal{H}_\mathcal{Y}}\right)\psi_\mathcal{Y}\left(y\right)\right\|^2_{\mathcal{H}_\mathcal{Y}}\right]\right] \tag{311}$$

$$= \mathbb{E}_y\left[\left\|\left(\widetilde{\mathrm{P}}_\mathrm{Y} - I_{\mathcal{H}_\mathcal{Y}}\right)\psi_\mathcal{Y}(y)\right\|^2_{\mathcal{H}_\mathcal{Y}}\right]. \tag{312}$$

■

B.4 Sketching Reconstruction Error

We provide here a bound on the reconstruction error of a sketching approximation.

Theorem 4.9 (sub-Gaussian sketching reconstruction error)**.** *For $\delta \in \left(0, 1/e\right]$, $n \in \mathbb{N}$ sufficiently large such that $\frac{9}{n}\log(n/\delta) \leq n^{-\frac{1}{1+\gamma_\mathcal{Z}}} \leq \|\mathrm{C}_\mathcal{Z}\|_{\mathrm{op}}/2$, then if*

$$\mathrm{m}_\mathcal{Z} \geq c_4 \max\left(\nu_\mathcal{Z}^2\, n^{\frac{\gamma_\mathcal{Z}+\mu_\mathcal{Z}}{1+\gamma_\mathcal{Z}}}, \nu_\mathcal{Z}^4 \log\left(1/\delta\right)\right), \tag{4.14}$$

with probability $1-\delta$ we have

$$\mathbb{E}_z\left[\|(\widetilde{\mathrm{P}}_\mathrm{Z} - I_{\mathcal{H}_\mathcal{Z}})\,\psi_\mathcal{Z}(z)\|^2_{\mathcal{H}_\mathcal{Z}}\right] \leq c_3\, n^{-\frac{1-\gamma_\mathcal{Z}}{1+\gamma_\mathcal{Z}}}, \tag{4.15}$$

where $c_3, c_4 > 0$ are constants independents of $n, \mathrm{m}_\mathcal{Z}, \delta$.

Proof For $t > 0$, we have

$$\mathbb{E}_z\left[\left\|\left(\widetilde{\mathrm{P}}_Z - I_{\mathcal{H}_\mathcal{Z}}\right)\psi_\mathcal{Z}(z)\right\|_{\mathcal{H}_\mathcal{Z}}^2\right] = \mathrm{Tr}\left(\left(\widetilde{\mathrm{P}}_Z - I_{\mathcal{H}_\mathcal{Z}}\right)\mathbb{E}_z\left[\psi_\mathcal{Z}(z)\otimes\psi_\mathcal{Z}(z)\right]\right) \tag{313}$$

$$= \left\|\left(\widetilde{\mathrm{P}}_Z - I_{\mathcal{H}_\mathcal{Z}}\right)\mathrm{C}_\mathcal{Z}{}^{1/2}\right\|_{\mathrm{HS}}^2 \tag{314}$$

$$\leq \left\|\left(\widetilde{\mathrm{P}}_Z - I_{\mathcal{H}_\mathcal{Z}}\right)\widehat{\mathrm{C}}_{Z\mathrm{t}}{}^{1/2}\right\|_{\mathrm{op}}^2 \left\|\widehat{\mathrm{C}}_{Z\mathrm{t}}{}^{-1/2}\mathrm{C}_{\mathcal{Z}\mathrm{t}}{}^{1/2}\right\|_{\mathrm{op}}^2 \left\|\mathrm{C}_{\mathcal{Z}\mathrm{t}}{}^{-1/2}\mathrm{C}_\mathcal{Z}{}^{1/2}\right\|_{\mathrm{HS}}^2. \tag{315}$$

Lemma .21 gives that, for $\delta \in (0,1)$, if $\frac{9}{n}\log\left(\frac{n}{\delta}\right) \leq t \leq \left\|\mathrm{C}_\mathcal{Z}\right\|_{\mathrm{op}}$, then with probability $1-\delta$

$$\left\|\widehat{\mathrm{C}}_{Z\mathrm{t}}{}^{-1/2}\mathrm{C}_{\mathcal{Z}\mathrm{t}}{}^{1/2}\right\|_{\mathrm{op}}^2 \leq 2. \tag{316}$$

Moreover, since $\left\|\mathrm{C}_{\mathcal{Z}\mathrm{t}}{}^{-1/2}\mathrm{C}_\mathcal{Z}{}^{1/2}\right\|_{\mathrm{HS}}^2 = \mathrm{Tr}\left(\mathrm{C}_{\mathcal{Z}\mathrm{t}}{}^{-1}\mathrm{C}_\mathcal{Z}\right) = \mathrm{d}_{\mathrm{eff}}^\mathcal{Z}(t)$, lemma .25 gives that

$$\left\|\mathrm{C}_{\mathcal{Z}\mathrm{t}}{}^{-1/2}\mathrm{C}_\mathcal{Z}{}^{1/2}\right\|_{\mathrm{HS}}^2 \leq \mathrm{Q}_\mathcal{Z}\, t^{-\gamma_\mathcal{Z}}. \tag{317}$$

Then, using the Lemma .18, and multiplying the bounds, gives

$$\mathbb{E}_y\left[\left\|\left(\widetilde{\mathrm{P}}_Z - I_{\mathcal{H}_\mathcal{Y}}\right)\psi_\mathcal{Z}(z)\right\|_{\mathcal{H}_\mathcal{Z}}^2\right] \leq 6\,\mathrm{Q}_\mathcal{Z}\, t^{1-\gamma_\mathcal{Z}}. \tag{318}$$

Finally, choosing $t = n^{-\frac{1}{1+\gamma_\mathcal{Z}}}$, defining $c_3 = 6\,\mathrm{Q}_\mathcal{Z}$ and $c_4 = 576\mathfrak{C}^2\,\mathrm{b}_\mathcal{Z}\,\mathrm{Q}_\mathcal{Z}$, and noticing $\mathcal{N}_\mathcal{Z}^\infty(t) \leq \mathrm{b}_\mathcal{Z}\,\mathrm{Q}_\mathcal{Z}\, t^{-(\gamma_\mathcal{Z}+\mu_\mathcal{Z})}$ (from lemmas .25 and .26), allows to conclude the proof.

■

Lemma .18. *Let $\mathcal{N}_\mathcal{Z}^\infty(t)$ be as in Definition .24. For all $\delta \in \left(0, 1/e\right]$, $\frac{9}{n}\log\left(\frac{n}{\delta}\right) \leq t \leq \left\|\mathrm{C}_\mathcal{Z}\right\|_{\mathrm{op}} - \frac{9}{n}\log\left(\frac{n}{\delta}\right)$ and $\mathrm{m}_\mathcal{Z} \geq \max\left(432\mathfrak{C}^2\,\nu_\mathcal{Z}{}^2\,\mathcal{N}_\mathcal{Z}^\infty(t), 576\mathfrak{C}^2\,\nu_\mathcal{Z}{}^4\log\left(1/\delta\right)\right)$, with probability at least $1-\delta$,*

$$\left\|\left(\widetilde{\mathrm{P}}_Z - I_{\mathcal{H}_\mathcal{Z}}\right)\widehat{\mathrm{C}}_{Z\mathrm{t}}{}^{1/2}\right\|_{\mathrm{op}}^2 \leq 3t. \tag{319}$$

Proof Using Propositions 3 and 7 from Rudi et al. (2015), we have, for $t > 0$,

$$\left\|\left(\widetilde{\mathrm{P}}_Z - I_{\mathcal{H}_\mathcal{Z}}\right)\widehat{\mathrm{C}}_{Z\mathrm{t}}{}^{1/2}\right\|_{\mathrm{op}}^2 \leq \frac{t}{1-\beta_\mathcal{Z}(t)}, \tag{320}$$

with $\beta_\mathcal{Z}(t) = \sigma_{\max}\left(\widehat{\mathrm{C}}_{Z\mathrm{t}}{}^{-1/2}\left(\widehat{\mathrm{C}}_Z - \widetilde{\mathrm{C}}_Z\right)\widehat{\mathrm{C}}_{Z\mathrm{t}}{}^{-1/2}\right)$.

Now, applying lemma .19, with the condition

$$\mathrm{m}_\mathcal{Z} \geq \max\left(432\mathfrak{C}^2\,\nu_\mathcal{Z}{}^2\,\mathcal{N}_\mathcal{Z}^\infty(t), 576\mathfrak{C}^2\,\nu_\mathcal{Z}{}^4\log\left(1/\delta\right)\right), \tag{321}$$

we obtain $\beta_{\mathcal{Z}}(t) \leq 2/3$, which gives

$$\left\| \left(\widetilde{\mathrm{P}}_{\mathrm{Z}} - I_{\mathcal{H}_{\mathcal{Y}}} \right) \widehat{\mathrm{C}}_{\mathrm{Zt}}^{1/2} \right\|_{\mathrm{op}}^2 \leq 3t \,. \tag{322}$$

■

Lemma .19. *Let* $\mathrm{R}_{\mathcal{Z}}$ *be as in Definition 4.8 and* $\mathcal{N}_{\mathcal{Z}}^{\infty}(t)$ *as in Definition .24. For all* $\delta \in \left(0, 1/e\right]$, $\frac{9}{n}\log\left(\frac{n}{\delta}\right) \leq t \leq \left\|\mathrm{C}_{\mathcal{Z}}\right\|_{\mathrm{op}} - \frac{9}{n}\log\left(\frac{n}{\delta}\right)$ *and* $\mathrm{m}_{\mathcal{Z}} \geq \max\left(6\mathcal{N}_{\mathcal{Z}}^{\infty}(t), \log\left(1/\delta\right)\right)$, *with probability at least* $1-\delta$,

$$\left\| \widehat{\mathrm{C}}_{\mathrm{Zt}}^{-1/2} \left(\widehat{\mathrm{C}}_{\mathrm{Z}} - \widetilde{\mathrm{C}}_{\mathrm{Z}} \right) \widehat{\mathrm{C}}_{\mathrm{Zt}}^{-1/2} \right\|_{\mathrm{op}} \leq \mathfrak{C} \frac{2\sqrt{2}\, \nu_{\mathcal{Z}} \sqrt{6\mathcal{N}_{\mathcal{Z}}^{\infty}(t)} + 8\, {\nu_{\mathcal{Z}}}^2 \sqrt{\log\left(1/\delta\right)}}{\sqrt{\mathrm{m}_{\mathcal{Z}}}}, \tag{323}$$

where $\mathfrak{C}$ *is a universal constant independent of* $\mathcal{N}_{\mathcal{Z}}^{\infty}(t)$, δ *and* $\mathrm{m}_{\mathcal{Z}}$.

Proof We define the following random variables

$$W_i = \sqrt{\frac{\mathrm{m}_{\mathcal{Z}}}{n}} \sum_{j=1}^{n} (\mathrm{R}_{\mathcal{Z}})_{ij} \widehat{\mathrm{C}}_{\mathrm{Zt}}^{-1/2} \psi_{\mathcal{Z}}(z_j) \in \mathcal{H}_{\mathcal{Z}} \quad \text{for } i = 1, \dots \mathrm{m}_{\mathcal{Z}} \,. \tag{324}$$

In order to use the concentration bound given in Theorem .23, we show that the W_is are i.i.d. weakly square integrable centered random vectors with covariance operator Σ, sub-Gaussian, and pre-Gaussian.

The W_is are weakly square integrable. Let $u \in \mathcal{H}_{\mathcal{Z}}$ and $v = \widehat{\mathrm{C}}_{\mathrm{Zt}}^{-1/2} u$, we have that $\langle W_i, u \rangle_{\mathcal{H}_{\mathcal{Z}}} = \sqrt{\frac{\mathrm{m}_{\mathcal{Z}}}{n}} \sum_{j=1}^{n} (\mathrm{R}_{\mathcal{Z}})_{ij} v(z_j)$. Hence, using the definition of a sub-Gaussian sketch, we have

$$\left\| \langle W_i, u \rangle_{\mathcal{H}_{\mathcal{Z}}} \right\|_{L_2(\mathbb{P})}^2 = \mathbb{E}_{\mathrm{R}_{\mathcal{Z}}} \left[|\langle W_i, u \rangle_{\mathcal{H}_{\mathcal{Z}}}|^2 \right] \tag{325}$$

$$= \frac{1}{n} \sum_{j=1}^{n} v(z_j)^2 \tag{326}$$

$$< +\infty \,. \tag{327}$$

The W_is are sub-Gaussian. Let $c \in \mathbb{R}$, using the independence and sub-Gaussianity of the $R_{z_{ij}}$, we have

$$\mathbb{E}_{\mathrm{R}_\mathcal{Z}}\left[\exp\left(c\langle W_i, u\rangle_{\mathcal{H}_\mathcal{Z}}\right)\right] = \mathbb{E}_{\mathrm{R}_\mathcal{Z}}\left[\exp\left(\sum_{j=1}^{n} c\sqrt{\frac{\mathrm{m}_\mathcal{Z}}{n}} R_{z_{ij}} v(z_j)\right)\right] \tag{328}$$

$$= \prod_{j=1}^{n} \mathbb{E}_{\mathrm{R}_\mathcal{Z}}\left[\exp\left(c\sqrt{\frac{\mathrm{m}_\mathcal{Z}}{n}} R_{z_{ij}} v(z_j)\right)\right] \tag{329}$$

$$\leq \prod_{j=1}^{n} \exp\left(\frac{c^2\, \mathrm{m}_\mathcal{Z}\, v(z_j)^2}{2n} \frac{{\nu_\mathcal{Z}}^2}{\mathrm{m}_\mathcal{Z}}\right) \tag{330}$$

$$= \exp\left(\frac{c^2\, {\nu_\mathcal{Z}}^2}{2n} \sum_{j=1}^{n} v(z_j)^2\right) \tag{331}$$

$$= \exp\left(\frac{c^2\, {\nu_\mathcal{Z}}^2}{2} \left\|\langle W_i, u\rangle_{\mathcal{H}_\mathcal{Z}}\right\|^2_{L_2(\mathbb{P})}\right). \tag{332}$$

Hence, $\langle W_i, u\rangle_{\mathcal{H}_\mathcal{Z}}$ is a $\frac{1}{2}\,{\nu_\mathcal{Z}}^2 \left\|\langle W_i, u\rangle_{\mathcal{H}_\mathcal{Z}}\right\|^2_{L_2(\mathbb{P})}$-sub-Gaussian random variable. Then, the Orlicz condition of sub-Gaussian random variables gives

$$\mathbb{E}_{\mathrm{R}_\mathcal{Z}}\left[\exp\left(\frac{\langle W_i, u\rangle^2_{\mathcal{H}_\mathcal{Z}}}{8\,{\nu_\mathcal{Z}}^2 \left\|\langle W_i, u\rangle_{\mathcal{H}_\mathcal{Z}}\right\|^2_{L_2(\mathbb{P})}}\right) - 1\right] \leq 1\,. \tag{333}$$

We deduce that

$$\left\|\langle W_i, u\rangle_{\mathcal{H}_\mathcal{Z}}\right\|_{\varphi_2} \leq 2\sqrt{2}\,\nu_\mathcal{Z} \left\|\langle W_i, u\rangle_{\mathcal{H}_\mathcal{Z}}\right\|_{L_2(\mathbb{P})}. \tag{334}$$

We conclude that the W_is are sub-Gaussian with $B = 2\sqrt{2}\,\nu_\mathcal{Z}$.

The W_is are pre-gaussian. We define $Z = \sqrt{\frac{\mathrm{m}_\mathcal{Z}}{n}} \sum_{j=1}^{n} G_j\, \widehat{\mathrm{C}}_{\mathrm{Zt}}{}^{-1/2}\, \psi_\mathcal{Z}(z_j)$, with $G_j \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, 1/\mathrm{m}_\mathcal{Z}\right)$. Z is a Gaussian random variable that admits the same covariance operator as the W_is. So, the W_i are pre-Gaussian.

Applying concentration bound. Because the W_is are i.i.d. weakly square integrable centered random variables, we can apply theorem .23, and by using also lemma .30, and the condition $\mathrm{m}_\mathcal{Z} \geq \max\left(6\mathcal{N}^\infty_\mathcal{Z}(t), \log\left(1/\delta\right)\right)$, we obtain

$$\left\|\widehat{\mathrm{C}}_{\mathrm{Zt}}{}^{-1/2}\left(\widehat{\mathrm{C}}_\mathrm{Z} - \widetilde{\mathrm{C}}_\mathrm{Z}\right)\widehat{\mathrm{C}}_{\mathrm{Zt}}{}^{-1/2}\right\|_{\mathrm{op}} \leq \mathfrak{C}\, \frac{2\sqrt{2}\,\nu_\mathcal{Z}\sqrt{6\mathcal{N}^\infty_\mathcal{Z}(t)} + 8\,{\nu_\mathcal{Z}}^2\sqrt{\log\left(1/\delta\right)}}{\sqrt{\mathrm{m}_\mathcal{Z}}}. \tag{335}$$

■

B.5 Probabilistic Bounds

In this section, we provide all the probabilistic bounds used in our proofs. In particular, we restate bounds from other works for the sake of providing a self-contained work. We order them in the same order of appearance in our proofs.

Lemma .20 (Bound $(A.1) = \left\| \left({S_Y}^{\#} S_X - H_\lambda \widehat{C}_X \right) {C_{\mathcal{X}\lambda}}^{-1/2} \right\|_{HS}$ (Ciliberto et al., 2020, Theorem B.10)). *Let $\delta \in [0,1]$, $n \in \mathbb{N}$ sufficiently large such that $\lambda = n^{-1/(1+\gamma_\mathcal{X})} \geq \frac{9{\kappa_\mathcal{X}}^2}{n} \log(\frac{n}{\delta})$ Under our set of assumptions, the following holds with probability at least $1-\delta$*

$$(A.1) \leq M \log(4/\delta) n^{-\frac{1}{2(1+\gamma_\mathcal{X})}} \tag{336}$$

where the constant M depends on $\kappa_\mathcal{Y}, \|H\|_{HS}, \delta$.

Proof This lemma can be obtained from Ciliberto et al. (2020, Theorem B.10), by noticing that the bound of Theorem B.10 is obtained by upper bounding the sum of $(A.1)$ and a positive term, such that the bound of Ciliberto et al. (2020, Theorem B.10) is an upper bound of $(A.1)$.

Lemma .21 (Bound $\left\| {\widehat{C}_{Z\lambda}}^{-1/2} {C_{\mathcal{Z}\lambda}}^{1/2} \right\|_{op}$ (Rudi et al., 2013, Lemma 3.6)). *If $\frac{9}{n} \log \frac{n}{\delta} \leq \lambda \leq \|C_\mathcal{Z}\|_{op}$, then we have with probability $1-\delta$*

$$\|{\widehat{C}_{Z\lambda}}^{-1/2} {C_{\mathcal{Z}\lambda}}^{1/2}\|_{op} \leq \sqrt{2}. \tag{337}$$

■

Lemma .22 (Bound $\left\| {C_{\mathcal{Z}\lambda}}^{-1/2} {\widehat{C}_{Z\lambda}}^{1/2} \right\|_{op}$). *If $\frac{9}{n} \log \frac{n}{\delta} \leq \lambda \leq \|C_\mathcal{Z}\|_{op}$, then with probability $1-\delta$*

$$\left\| {C_{\mathcal{Z}\lambda}}^{-1/2} {\widehat{C}_{Z\lambda}}^{1/2} \right\|_{op} \leq \sqrt{\frac{3}{2}}. \tag{338}$$

Proof We have

$$\left\| {C_{\mathcal{Z}\lambda}}^{-1/2} {\widehat{C}_{Z\lambda}}^{1/2} \right\|_{op} = \left\| {C_{\mathcal{Z}\lambda}}^{-1/2} \widehat{C}_{Z\lambda} {C_{\mathcal{Z}\lambda}}^{-1/2} \right\|_{op}^{1/2} \tag{339}$$

$$= \left\| I + {C_{\mathcal{Z}\lambda}}^{-1/2} (\widehat{C}_Z - C_\mathcal{Z}) {C_{\mathcal{Z}\lambda}}^{-1/2} \right\|_{op}^{1/2} \tag{340}$$

$$\leq \left(1 + \left\| {C_{\mathcal{Z}\lambda}}^{-1/2} (\widehat{C}_Z - C_\mathcal{Z}) {C_{\mathcal{Z}\lambda}}^{-1/2} \right\|_{op} \right)^{1/2} \tag{341}$$

$$\leq \sqrt{\frac{3}{2}} \tag{342}$$

with probability at least $1-\delta$, where the last inequality is from Rudi et al. (2013, Lemma 3.6).

Theorem .23 (sub-Gaussian concentration bound (Koltchinskii and Lounici, 2017, Theorem 9)). *Let $W, W_1, \dots, W_m$ be i.i.d. weakly square integrable centered random vectors in a separable Hilbert space $\mathcal{H}_\mathcal{Z}$ with covariance operator Σ. If W is sub-Gaussian and pre-Gaussian, then there exists a constant $\mathfrak{C} > 0$ such that, for all $\tau \geq 1$, with probability at least $1 - e^{-\tau}$,*

$$\|\hat{\Sigma} - \Sigma\| \leq \mathfrak{C}\|\Sigma\| \left(B \sqrt{\frac{\mathbf{r}(\Sigma)}{m}} \vee \frac{\mathbf{r}(\Sigma)}{m} \vee B^2 \sqrt{\frac{\tau}{m}} \vee B^2 \frac{\tau}{m} \right), \tag{343}$$

where $B > 0$ is the constant such that $\left\| \langle W, u \rangle_{\mathcal{H}_\mathcal{Y}} \right\|_{\varphi_2} \leq B \left\| \langle W, u \rangle_{\mathcal{H}_\mathcal{Y}} \right\|_{L_2(\mathbb{P})}$ for all $u \in \mathcal{H}_\mathcal{Z}$.

■

B.6 Auxiliary Results And Definitions

Definition .24. *For $t > 0$, we define the random variable*

$$\mathcal{N}(z, t) = \langle \psi_\mathcal{Z}(z), {\mathrm{C}_{\mathcal{Z}\mathrm{t}}}^{-1} \psi_\mathcal{Z}(z) \rangle_{\mathcal{H}_\mathcal{Z}} \tag{344}$$

with $z \in \mathcal{Z}$ distributed according to $\rho_\mathcal{Z}$ and let

$$\mathrm{d}_{\mathrm{eff}}^{\mathcal{Z}}(t) = \mathbb{E}_z \left[\mathcal{N}(z, t) \right] = \mathrm{Tr}\left(\mathrm{C}_\mathcal{Z} {\mathrm{C}_{\mathcal{Z}\mathrm{t}}}^{-1} \right), \quad \mathcal{N}_\mathcal{Z}^\infty(t) = \sup_{z \in \mathcal{Z}} \mathcal{N}(z, t). \tag{345}$$

We note $\mathcal{N}_\mathcal{X}^\infty, \mathrm{d}_{\mathrm{eff}}^{\mathcal{X}}(t), \gamma_\mathcal{X}, \mathrm{Q}_\mathcal{Y}, \mathcal{N}_\mathcal{Y}^\infty, \mathrm{d}_{\mathrm{eff}}^{\mathcal{Y}}(t), \gamma_\mathcal{Y}, \mathrm{Q}_\mathcal{Y}$ for the input and output kernels $\mathrm{k}_\mathcal{X}, k_\mathcal{Y}$, respectively.

Lemma .25. *When Assumption 4.5 holds then we have*

$$\mathrm{d}_{\mathrm{eff}}^{\mathcal{Z}}(t) \leq \mathrm{Q}_\mathcal{Z}\, t^{-\gamma_\mathcal{Z}}. \tag{346}$$

Proof We have

$$\begin{aligned} \mathrm{d}_{\mathrm{eff}}^{\mathcal{Z}}(t) &= \mathrm{Tr}\left(\mathrm{C}_\mathcal{Z} {\mathrm{C}_{\mathcal{Z}\mathrm{t}}}^{-1} \right) && (347) \\ &\leq \mathrm{Tr}\left({\mathrm{C}_\mathcal{Z}}^{\gamma_\mathcal{Z}} \right) \| {\mathrm{C}_\mathcal{Z}}^{1-\gamma_\mathcal{Z}} {\mathrm{C}_{\mathcal{Z}\mathrm{t}}}^{-1} \|_{\mathrm{op}} && (348) \\ &\leq \mathrm{Q}_\mathcal{Z}\, t^{-\gamma_\mathcal{Z}}. && (349) \end{aligned}$$

■

Lemma .26. *When Assumption 4.6 holds then we have*

$$\mathcal{N}_\mathcal{Z}^\infty(t) \leq \mathrm{b}_\mathcal{Z}\, \mathrm{d}_{\mathrm{eff}}^{\mathcal{Z}}(t) t^{-\mu_\mathcal{Z}}. \tag{350}$$

Proof We have

$$\mathcal{N}_{\mathcal{Z}}^{\infty}(t) = \sup_{z\in\mathcal{Z}} \langle \psi_{\mathcal{Z}}(z), {C_{\mathcal{Z}t}}^{-1}\psi_{\mathcal{Z}}(z)\rangle_{\mathcal{H}_{\mathcal{Z}}} \tag{351}$$

$$\leq b_{\mathcal{Z}} \operatorname{Tr}({C_{\mathcal{Z}t}}^{-1} {C_{\mathcal{Z}}}^{1-\mu_{\mathcal{Z}}}) \tag{352}$$

$$\leq b_{\mathcal{Z}} \operatorname{Tr}({C_{\mathcal{Z}t}}^{-1} C_{\mathcal{Z}}) \| {C_{\mathcal{Z}t}}^{-\mu_{\mathcal{Z}}} \|_{\mathrm{op}} \tag{353}$$

$$\leq b_{\mathcal{Z}}\, d_{\mathrm{eff}}^{\mathcal{Z}}(t) t^{-\mu_{\mathcal{Z}}}. \tag{354}$$

■

We recall the following deterministic bound.

Lemma .27 (Bound $\|{\widehat{C}_{X\lambda}}^{1/2} \tilde{\eta}(\widehat{C}_X) {\widehat{C}_{X\lambda}}^{1/2}\|_{\mathrm{op}}$ (Rudi et al., 2015, Lemma 8))**.** *For any* $\lambda > 0$,

$$\|{\widehat{C}_{X\lambda}}^{1/2} \tilde{\eta}(\widehat{C}_X) {\widehat{C}_{X\lambda}}^{1/2}\|_{\mathrm{op}} \leq 1. \tag{355}$$

We introduce some notations and definitions from Koltchinskii and Lounici (2017). Let W be a centered random variable in $\mathcal{H}_{\mathcal{Z}}$, W is weakly square integrable if and only if $\left\|\langle W, u\rangle_{\mathcal{H}_{\mathcal{Z}}}\right\|_{L_2(\mathbb{P})}^2 := \mathbb{E}\left[|\langle W, u\rangle_{\mathcal{H}_{\mathcal{Z}}}|^2\right] < +\infty$, for any $u \in \mathcal{H}_{\mathcal{Z}}$. Moreover, we define the Orlicz norms. For a convex nondecreasing function $\varphi : \mathbb{R}_+ \to \mathbb{R}_+$ with $\varphi(0) = 0$ and a random variable η on a probability space $\left(\Omega, \mathcal{A}, \mathbb{P}\right)$, the φ-norm of η is defined as

$$\|\eta\|_{\varphi} = \inf\left\{C > 0 : \mathbb{E}\left[\varphi\left(|\eta|/C\right)\right] \leq 1\right\}. \tag{356}$$

The Orlicz φ_1- and φ_2-norms coincide to the functions $\varphi_1(u) = e^u - 1, u \geq 0$ and $\varphi_2(u) = e^{u^2} - 1, u \geq 0$. Finally, Koltchinskii and Lounici (2017) introduces the definitions of sub-Gaussian and pre-Gaussian random variables in a separable Banach space E. We focus on the case where $E = \mathcal{H}_{\mathcal{Z}}$.

Definition .28. *A centered random variable X in $\mathcal{H}_{\mathcal{Z}}$ will be called sub-Gaussian iff, for all $u \in \mathcal{H}_{\mathcal{Z}}$, there exists $B > 0$ such that*

$$\left\|\langle X, u\rangle_{\mathcal{H}_{\mathcal{Z}}}\right\|_{\psi_{\mathcal{X}_2}} \leq B \left\|\langle X, u\rangle_{\mathcal{H}_{\mathcal{Z}}}\right\|_{L_2(\mathbb{P})}. \tag{357}$$

Definition .29. *A weakly square integrable centered random variable X in $\mathcal{H}_{\mathcal{Z}}$ with covariance operator Σ is called pre-Gaussian iff there exists a centered Gaussian random variable Y in $\mathcal{H}_{\mathcal{Z}}$ with the same covariance operator Σ.*

Lemma .30 (Expectancy, covariance, and intrinsic dimension of the W_is)**.** *Defining* $W_i = \sqrt{\frac{m_{\mathcal{Z}}}{n}} \sum_{j=1}^{n} (R_{\mathcal{Z}})_{ij} {\widehat{C}_{Zt}}^{-1/2} \psi_{\mathcal{Z}}(z_j) \in \mathcal{H}_{\mathcal{Z}}$ *for* $i = 1, \dots m_{\mathcal{Z}}$ *where* $R_{\mathcal{Z}}$ *is a sub-Gaussian sketch, the following hold true*

$$\mathbb{E}_{R_{\mathcal{Z}}}\left[W_i\right] = 0 \tag{358}$$

$$\Sigma = \mathbb{E}_{R_{\mathcal{Z}}}\left[W_i \otimes W_i\right] = {\widehat{C}_{Zt}}^{-1/2} \widehat{C}_Z {\widehat{C}_{Zt}}^{-1/2} \tag{359}$$

$$\widehat{\Sigma} = \frac{1}{m_{\mathcal{Z}}} \sum_{i=1}^{m_{\mathcal{Z}}} \langle f, W_i\rangle_{\mathcal{H}_{\mathcal{Z}}} W_i = {\widehat{C}_{Zt}}^{-1/2} \widetilde{C}_Z {\widehat{C}_{Zt}}^{-1/2} \tag{360}$$

and for $\delta \in \left(0,1\right)$*, if* $\frac{9}{n}\log\left(\frac{n}{\delta}\right) \leq t \leq \left\|\mathrm{C}_{\mathcal{Z}}\right\|_{\mathrm{op}} - \frac{9}{n}\log\left(\frac{n}{\delta}\right)$*, then with probability* $1-\delta$

$$r\left(\Sigma\right) = \frac{\mathbb{E}_{\mathrm{R}_{\mathcal{Z}}}\left[\left\|X_i\right\|_{\mathcal{H}_{\mathcal{Z}}}\right]^2}{\left\|\Sigma\right\|_{\mathrm{op}}} \leq 6\mathcal{N}_{\mathcal{Z}}^{\infty}(t). \tag{361}$$

Proof First, it is straightforward to check that

$$\frac{1}{\mathrm{m}_{\mathcal{Z}}}\sum_{i=1}^{\mathrm{m}_{\mathcal{Z}}}\langle f, W_i\rangle_{\mathcal{H}_{\mathcal{Z}}} W_i = \widehat{\mathrm{C}}_{\mathrm{Zt}}{}^{-1/2}\widetilde{\mathrm{C}}_{\mathrm{Z}}\widehat{\mathrm{C}}_{\mathrm{Zt}}{}^{-1/2}. \tag{362}$$

Then, since $\mathbb{E}_{\mathrm{R}_{\mathcal{Z}}}\left[(\mathrm{R}_{\mathcal{Z}})_{i:}\right] = 0$,

$$\mathbb{E}_{\mathrm{R}_{\mathcal{Z}}}\left[W_i\right] = \sqrt{\frac{\mathrm{m}_{\mathcal{Z}}}{n}}\widehat{\mathrm{C}}_{\mathrm{Zt}}{}^{-1/2}\mathrm{S_Z}{}^{\#}\,\mathbb{E}_{\mathrm{R}_{\mathcal{Z}}}\left[(\mathrm{R}_{\mathcal{Z}})_{i:}\right] = 0. \tag{363}$$

Then,

$$\begin{aligned}
\left(W_i \otimes W_i\right) f &= \langle f, W_i\rangle_{\mathcal{H}_{\mathcal{Z}}} W_i && (364)\\
&= \langle f, \sqrt{\mathrm{m}_{\mathcal{Z}}}\widehat{\mathrm{C}}_{\mathrm{Zt}}{}^{-1/2}\mathrm{S_Z}{}^{\#}(\mathrm{R}_{\mathcal{Z}})_{i:}\rangle_{\mathcal{H}_{\mathcal{Z}}}\sqrt{\mathrm{m}_{\mathcal{Z}}}\widehat{\mathrm{C}}_{\mathrm{Zt}}{}^{-1/2}\mathrm{S_Z}{}^{\#}(\mathrm{R}_{\mathcal{Z}})_{i:} && (365)\\
&= \mathrm{m}_{\mathcal{Z}}\left((\mathrm{R}_{\mathcal{Z}})_{i:}^{\top}\mathrm{S_Z}\widehat{\mathrm{C}}_{\mathrm{Zt}}{}^{-1/2} f\right)\widehat{\mathrm{C}}_{\mathrm{Zt}}{}^{-1/2}\mathrm{S_Z}{}^{\#}(\mathrm{R}_{\mathcal{Z}})_{i:} && (366)\\
&= \widehat{\mathrm{C}}_{\mathrm{Zt}}{}^{-1/2}\mathrm{S_Z}{}^{\#}\left(\mathrm{m}_{\mathcal{Z}}(\mathrm{R}_{\mathcal{Z}})_{i:}(\mathrm{R}_{\mathcal{Z}})_{i:}^{\top}\right)\mathrm{S_Z}\widehat{\mathrm{C}}_{\mathrm{Zt}}{}^{-1/2} f, && (367)
\end{aligned}$$

and since $\mathbb{E}_{\mathrm{R}_{\mathcal{Z}}}\left[\mathrm{m}_{\mathcal{Z}}(\mathrm{R}_{\mathcal{Z}})_{i:}(\mathrm{R}_{\mathcal{Z}})_{i:}^{\top}\right] = I_n$,

$$\begin{aligned}
\Sigma &= \mathbb{E}_{\mathrm{R}_{\mathcal{Z}}}\left[W_i \otimes W_i\right] && (368)\\
&= \widehat{\mathrm{C}}_{\mathrm{Zt}}{}^{-1/2}\mathrm{S_Z}{}^{\#}\,\mathbb{E}_{\mathrm{R}_{\mathcal{Z}}}\left[\mathrm{m}_{\mathcal{Z}}(\mathrm{R}_{\mathcal{Z}})_{i:}(\mathrm{R}_{\mathcal{Z}})_{i:}^{\top}\right]\mathrm{S_Z}\widehat{\mathrm{C}}_{\mathrm{Zt}}{}^{-1/2} && (369)\\
&= \widehat{\mathrm{C}}_{\mathrm{Zt}}{}^{-1/2}\widehat{\mathrm{C}}_{\mathrm{Z}}\widehat{\mathrm{C}}_{\mathrm{Zt}}{}^{-1/2}. && (370)
\end{aligned}$$

Then,

$$\mathbb{E}_{\mathrm{R}_\mathcal{Z}}\left[\left\|X_i\right\|_{\mathcal{H}_\mathcal{Z}}\right]^2 \leq \mathbb{E}_{\mathrm{R}_\mathcal{Z}}\left[\left\|X_i\right\|_{\mathcal{H}_\mathcal{Z}}^2\right] \quad \text{(by Jensen's inequality)} \tag{371}$$

$$= \mathrm{m}_\mathcal{Z}\,\mathbb{E}_{\mathrm{R}_\mathcal{Z}}\left[\langle \widehat{\mathrm{C}}_{\mathrm{Zt}}{}^{-1/2}\,\mathrm{S}_\mathrm{Z}{}^{\#}(\mathrm{R}_\mathcal{Z})_{i:}, \widehat{\mathrm{C}}_{\mathrm{Zt}}{}^{-1/2}\,\mathrm{S}_\mathrm{Z}{}^{\#}(\mathrm{R}_\mathcal{Z})_{i:}\rangle_{\mathcal{H}_\mathcal{Z}}\right] \tag{372}$$

$$= \frac{\mathrm{m}_\mathcal{Z}}{n}\,\mathbb{E}_{\mathrm{R}_\mathcal{Z}}\left[\langle \sum_{j=1}^{n} \mathrm{R}_{\mathcal{Z}_{\mathrm{ij}}}\,\psi_\mathcal{Z}(z_j), \sum_{l=1}^{n} \mathrm{R}_{\mathcal{Z}_{\mathrm{il}}}\,\widehat{\mathrm{C}}_{\mathrm{Zt}}{}^{-1}\,\psi_\mathcal{Z}(z_l)\rangle_{\mathcal{H}_\mathcal{Z}}\right] \tag{373}$$

$$= \frac{\mathrm{m}_\mathcal{Z}}{n}\,\mathbb{E}_{\mathrm{R}_\mathcal{Z}}\left[\sum_{j,l=1}^{n} \mathrm{R}_{\mathcal{Z}_{\mathrm{ij}}}\,\mathrm{R}_{\mathcal{Z}_{\mathrm{il}}}\langle \psi_\mathcal{Z}(z_j), \widehat{\mathrm{C}}_{\mathrm{Zt}}{}^{-1}\,\psi_\mathcal{Z}(z_l)\rangle_{\mathcal{H}_y}\right] \tag{374}$$

$$= \frac{\mathrm{m}_\mathcal{Z}}{n}\sum_{j=1}^{n}\frac{1}{\mathrm{m}_\mathcal{Z}}\langle \psi_\mathcal{Z}(z_j), \widehat{\mathrm{C}}_{\mathrm{Zt}}{}^{-1}\,\psi_\mathcal{Z}(z_j)\rangle_{\mathcal{H}_\mathcal{Z}} \tag{375}$$

$$= \mathrm{Tr}\left(\widehat{\mathrm{C}}_{\mathrm{Zt}}{}^{-1}\,\widehat{\mathrm{C}}_\mathrm{Z}\right) \tag{376}$$

$$= \left\|\widehat{\mathrm{C}}_{\mathrm{Zt}}{}^{-1/2}\,\widehat{\mathrm{C}}_\mathrm{Z}{}^{1/2}\right\|_{\mathrm{HS}}^2 \tag{377}$$

$$\leq \left\|\widehat{\mathrm{C}}_{\mathrm{Zt}}{}^{-1/2}\,\mathrm{C}_{\mathcal{Z}\mathrm{t}}{}^{1/2}\right\|_{\mathrm{op}}^2 \left\|\mathrm{C}_{\mathcal{Z}\mathrm{t}}{}^{-1/2}\,\widehat{\mathrm{C}}_\mathrm{Z}{}^{1/2}\right\|_{\mathrm{HS}}^2. \tag{378}$$

But,

$$\left\|\mathrm{C}_{\mathcal{Z}\mathrm{t}}{}^{-1/2}\,\widehat{\mathrm{C}}_\mathrm{Z}{}^{1/2}\right\|_{\mathrm{HS}}^2 = \mathrm{Tr}\left(\mathrm{C}_{\mathcal{Z}\mathrm{t}}{}^{-1}\,\widehat{\mathrm{C}}_\mathrm{Z}\right) \tag{379}$$

$$= \mathrm{Tr}\left(\mathrm{C}_{\mathcal{Z}\mathrm{t}}{}^{-1}\left(\frac{1}{n}\sum_{i=1}^{n}\psi_\mathcal{Z}(z_i)\otimes\psi_\mathcal{Z}(z_i)\right)\right) \tag{380}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\mathrm{Tr}\left(\mathrm{C}_{\mathcal{Z}\mathrm{t}}{}^{-1}\left(\psi_\mathcal{Z}(z_i)\otimes\psi_\mathcal{Z}(z_i)\right)\right) \tag{381}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\left\langle \psi_\mathcal{Z}(z_i), \mathrm{C}_{\mathcal{Z}\mathrm{t}}{}^{-1}\,\psi_\mathcal{Z}(z_i)\right\rangle_{\mathcal{H}_y} \tag{382}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\mathcal{N}(z_i,t) \tag{383}$$

$$\leq \mathcal{N}_\mathcal{Z}^\infty(t). \tag{384}$$

Then, from lemma .21, for $\delta \in \left(0,1\right)$, and $\frac{9}{n}\log\left(\frac{n}{\delta}\right) \leq t \leq \left\|\mathrm{C}_\mathcal{Z}\right\|_{\mathrm{op}}$, then with probability $1-\delta$,

$$\mathbb{E}_{\mathrm{R}_\mathcal{Z}}\left[\left\|X_i\right\|_{\mathcal{H}_\mathcal{Z}}\right]^2 \leq 2\mathcal{N}_\mathcal{Z}^\infty(t). \tag{385}$$

Then, $\left\|\Sigma\right\|_{\mathrm{op}} = \left\|\widehat{\mathrm{C}}_{\mathrm{Zt}}{}^{-1/2}\,\widehat{\mathrm{C}}_\mathrm{Z}{}^{1/2}\right\|_{\mathrm{op}}^2 \geq 1/3$ for $t \leq 2\left\|\widehat{\mathrm{C}}_\mathrm{Z}\right\|_{\mathrm{op}}$.

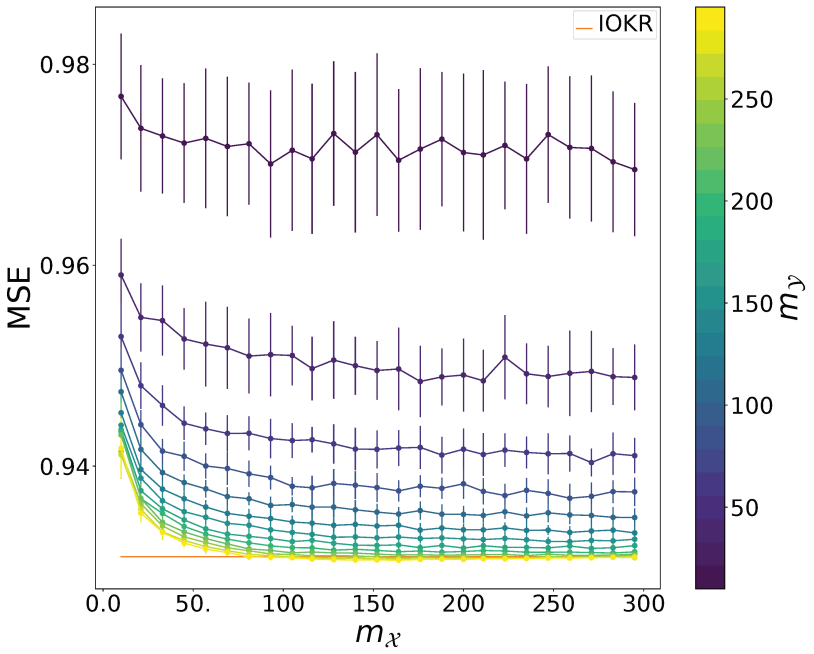


Figure B.4: Test MSE with respect to $m_{\mathcal{X}}$ and $m_{\mathcal{Y}}$ for the SISOKR model with $(2 \cdot 10^{-3})$-SR input and output sketches.

We conclude that

$$\frac{\mathbb{E}_{R_Z}\left[\left\|W_i\right\|_{\mathcal{H}_Z}\right]^2}{\left\|\Sigma\right\|_{\text{op}}} \leq 6\mathcal{N}_{\mathcal{Z}}^{\infty}(t). \tag{386}$$

Finally, in order to obtain a condition on t that does not depend on empirical quantities, we use lemma .21 which gives that, for any $\frac{9}{n}\log\left(\frac{n}{\delta}\right) \leq t' \leq \left\|C_{\mathcal{Z}}\right\|_{\text{op}}$, then $C_{\mathcal{Z}t'} \preceq 2\widehat{C}_{Zt'}$, which implies $2\left\|\widehat{C}_Z\right\|_{\text{op}} \geq \left\|C_{\mathcal{Z}}\right\|_{\text{op}} - t'$. Now, taking $t' = \frac{9}{n}\log\left(\frac{n}{\delta}\right)$, we obtain $\left\|C_{\mathcal{Z}}\right\|_{\text{op}} - \frac{9}{n}\log\left(\frac{n}{\delta}\right) \leq 2\left\|\widehat{C}_Z\right\|_{\text{op}}$.

■

B.7 Additional experiments and details

In this section, we bring some additional experiments and details.

Simulated Data Set for Least Squares Regression

We report here some results about statistical performance on the synthetic data set described in section 4.5. First, we give an additional figure showing the MSE with respect to $m_{\mathcal{X}}$ and $m_{\mathcal{Y}}$ of the SISOKR model, see Figure B.4. As reported in Figure B.5, SIOKR outperforms IOKR from $m_{\mathcal{X}} = 100$, and ISOKR obtains a very similar result to IOKR from $m_{\mathcal{Y}} = 250$.

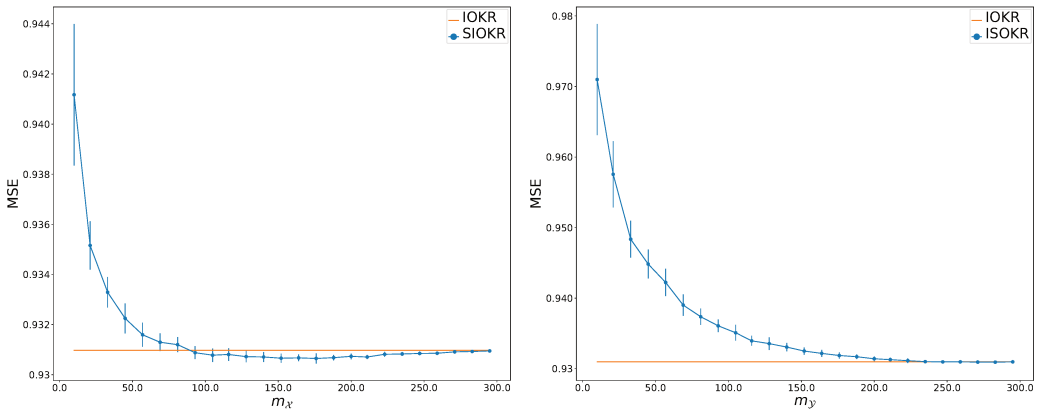


Figure B.5: Test MSE with respect to $m_{\mathcal{X}}$ and $m_{\mathcal{Y}}$ for a SIOKR (left) and ISOKR (right) model respectively with $(2 \cdot 10^{-3})$-SR input and output sketches.

More Details about Multi-Label Classification Data Set

In this section, you can find more details about training and testing sizes, the number of features of the inputs, and the number of labels to predict of Bibtex, Bookmarks, and Mediamill data sets in table .3.

Table .3: Multi-label data sets description.

Data set	n	n_{te}	$n_{features}$	n_{labels}
Bibtex	4880	2515	1836	159
Bookmarks	60000	27856	2150	298
Mediamill	30993	12914	120	101

C Appendices for Chapter 5

C.1 Gradient computations with Robust Losses

In this section, we give further details about gradient computations when using ε-insensitive ℓ_1 (ε-SVR), ℓ_2 (ε-ridge regression) and Huber (κ-Huber regression) losses. We remind that DSOKR denotes the model whose surrogate estimator is $h_\theta = g_E \circ g_W$, where g_θ is a deep neural network and W denotes its weights. For all $(x, y) \in \mathcal{X} \times \mathcal{Y}$, we denote $l(W; x, y) = \|g_E \circ g_W(x) - \psi_\mathcal{Y}(y)\|^2_{\mathcal{H}_\mathcal{Y}}$, and then, by eq. (5.11), its gradient is given by

$$\frac{\partial}{\partial W}\, l(W; x, y) = \frac{\partial}{\partial W}\, \|g_W(x)\|_2^2 - 2\frac{\partial}{\partial W}\, \tilde{\psi}_\mathcal{Y}(y)^\top g_W(x). \tag{387}$$

In the following, IOKR denotes the model whose surrogate estimator is $h_\theta = g_E \circ g_W$, where $g_W : x \mapsto W^\top \mathrm{k}_\mathrm{X}^\mathrm{x}$ and $W \in \mathbb{R}^{n \times \mathrm{p_Y}}$ is the solution to

$$\min_{W \in \mathbb{R}^{n \times \mathrm{p_Y}}} \frac{1}{n} \sum_{i=1}^{n} c\left(k_X^{x_i\top} W W^\top k_X^{x_i} - 2k_X^{x_i\top} W\, \tilde{\psi}_\mathcal{Y}(y) + \mathrm{k}_\mathcal{Y}(y, y)\right) + \lambda \operatorname{Tr}(\mathrm{K_X}\, W W^\top). \tag{5.16}$$

Moreover, we provide the following set of useful gradients:

$$\frac{\partial}{\partial W}\, \mathrm{k}_\mathrm{X}^{\mathrm{x}\top} W W^\top \mathrm{k}_\mathrm{X}^\mathrm{x} = 2\, \mathrm{k}_\mathrm{X}^\mathrm{x}\, \mathrm{k}_\mathrm{X}^{\mathrm{x}\top} W, \tag{388}$$

$$\frac{\partial}{\partial W}\, -2\, \mathrm{k}_\mathrm{X}^{\mathrm{x}\top} W\, \tilde{\psi}_\mathcal{Y}(y) = -2\, \mathrm{k}_\mathrm{X}^\mathrm{x}\, \tilde{\psi}_\mathcal{Y}(y)^\top, \tag{389}$$

$$\frac{\partial}{\partial W}\, \operatorname{Tr}(\mathrm{K_X}\, W W^\top) = 2\, \mathrm{K_X}\, W. \tag{390}$$

Hence, as for DSOKR, for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$, we denote $l(W; x, y) = \|g_E \circ g_W(x) - \psi_\mathcal{Y}(y)\|^2_{\mathcal{H}_\mathcal{Y}}$ and then, its gradient is given by

$$\frac{\partial}{\partial W}\, l(W; x, y) = 2\, \mathrm{k}_\mathrm{X}^\mathrm{x}\, \mathrm{k}_\mathrm{X}^{\mathrm{x}\top} W - 2\, \mathrm{k}_\mathrm{X}^\mathrm{x}\, \tilde{\psi}_\mathcal{Y}(y)^\top. \tag{391}$$

ε-SVR

We recall that, for $\varepsilon > 0$, the ε-insensitive ℓ_1 loss is given by

$$\ell : (z, z') \in {\mathcal{H}_\mathcal{Y}}^2 \mapsto \max(\|z - z'\|_{\mathcal{H}_\mathcal{Y}} - \varepsilon, 0). \tag{392}$$

Then, for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$, the gradient is given by

$$\frac{\partial}{\partial W}\, \ell(g_E \circ g_W(x), \psi_\mathcal{Y}(y)) = \frac{1}{2\sqrt{l(W; x, y)}} \frac{\partial}{\partial W}\, l(W; x, y) \mathbb{1}_{\{\sqrt{l(W;x,y)} > \varepsilon\}}, \tag{393}$$

where $\mathbb{1}_{\{\cdot\}}$ denotes the indicator function. One can then solve the primal ERM problem via a gradient-based method for DSOKR and IOKR models by using the expression of the gradient of l in eq. (387) and eq. (391) respectively, without forgetting the gradient of the regularisation penalty in eq. (390) for IOKR. The same applies for the ε-insensitive ℓ_2 and Huber losses.

ε-Ridge Regression

We recall that, for $\varepsilon > 0$, the ε-insensitive ℓ_2 loss is given by

$$\ell : (z, z') \in {\mathcal{H}_\mathcal{Y}}^2 \mapsto \max(\|z - z'\|_{\mathcal{H}_\mathcal{Y}} - \varepsilon, 0)^2 . \tag{394}$$

Then, for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$, the gradient is given by

$$\frac{\partial}{\partial W} \ell(g_E \circ g_W(x), \psi_\mathcal{Y}(y)) = \left(1 - \frac{\varepsilon}{\sqrt{l(W;x,y)}}\right) \frac{\partial}{\partial W} l(W;x,y) \mathbb{1}_{\{\sqrt{l(W;x,y)} > \varepsilon\}} . \tag{395}$$

κ-Huber Regression

We recall that, for $\kappa > 0$, the κ-Huber loss is given by

$$\ell : (z, z') \in {\mathcal{H}_\mathcal{Y}}^2 \mapsto \begin{cases} \frac{1}{2}\|z - z'\|^2_{\mathcal{H}_\mathcal{Y}} & \text{if } \|z - z'\|_{\mathcal{H}_\mathcal{Y}} \leq \kappa \\ \kappa\left(\|z - z'\|_{\mathcal{H}_\mathcal{Y}} - \frac{\kappa}{2}\right) & \text{otherwise} \end{cases} . \tag{396}$$

Then, for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$, the gradient is given by

$$\frac{\partial}{\partial W} \ell(g_E \circ g_W(x), \psi_\mathcal{Y}(y)) = \begin{cases} \frac{1}{2} \frac{\partial}{\partial W} l(W;x,y) & \text{if } \sqrt{l(W;x,y)} \leq \kappa \\ \frac{\kappa}{2\sqrt{l(W;x,y)}} \frac{\partial}{\partial W} l(W;x,y) & \text{otherwise} \end{cases} . \tag{397}$$

As a conclusion, the basis approach offers a way to perform gradient descent algorithms to solve the primal ERM problem with a wider variety of losses than only the square one, such as the above robust losses. Moreover, by considering an input kernel rather than a deep neural network, which finally corresponds to the shallow IOKR model, it constitutes an interesting alternative to the dual approach of Laforgue et al. (2020) presented in section 2.5.2 since leveraging sketching with duality is not straightforward, as discusses in section 3.2.3.

C.2 Graph Prediction via Output Kernel Regression

In this section, we present kernel examples to tackle graph prediction via Output Kernel Regression.

A graph G is defined by its sets of vertices V and edges E. Besides, it may contain either node labels or attributes, or edge labels, attributes, or weights. Before giving some examples of kernels dealing directly with graphs, we present examples of kernels dealing with fingerprints.

Fingerprints. Indeed, when manipulating molecules, either for molecular property prediction or molecule identification, many works use fingerprints to represent graphs (Ralaivola et al., 2005; Brouard et al., 2016a,b; Tripp et al., 2023). A fingerprint is a binary vector of length $d \geq 1$ and each entry of the fingerprint encodes the presence or absence of a substructure within the graph based on a dictionary. Hence, when using fingerprints, the problem of graph prediction becomes a high-dimensional multi-label prediction problem. A very popular kernel to handle fingerprints is the Tanimoto kernel (Tanimoto, 1958), which basically consists of an intercept over union measure between two fingerprints.

Graph kernels. In this work, we also manipulate raw graphs. Many kernels exist to handle graphs, we present a few that we will use during the experiments. For more details about these kernels and other graph kernel examples, see the documentation of the GraKet library (Siglidis et al., 2020).

Definition .31 (Vertex Histogram kernel)**.** *Let $G = (V, E)$ and $G' = (V', E')$ be two node-labeled graphs. Let $\mathcal{L} = \{1, \dots, d\}$ be the set of labels, and $\ell : v \in V \mapsto \ell(v) \in \mathcal{L}$ be the function that assigns a label for each vertex. Then, the vertex label histogram of G is a vector $f = (f_1, \dots, f_d)^\top$, such that $f_i = |\{v \in V : \ell(v) = i\}|$ for each $i \in \mathcal{L}$. Let f, f' be the vertex label histograms of G, G', respectively. The vertex histogram kernel is then defined as the linear kernel between f and f', that is*

$$\mathrm{k}_{\mathcal{Y}}(G, G') = f^\top f'. \tag{398}$$

The VH kernel needs node-labeled graphs and simply compares two graphs based on the number of nodes having each type of label. Its computation is very fast.

Definition .32 (Shortest-Path kernel (Borgwardt and Kriegel, 2005))**.** *Let $G = (V, E)$ and $G' = (V', E')$ be two graphs, and $S = (V, E_S)$ and $S' = (V', E'_{S'})$ their corresponding shortest-path graphs, i.e. the graphs where we only keep the edges contained in the shortest path between every vertex, then $E_S \subseteq E$ and $E'_{S'} \subseteq E'$. The shortest-path kernel is then defined on G and G' as*

$$\mathrm{k}_{\mathcal{Y}}(G, G') = \mathrm{k}_{\mathcal{Y}}(S, S') = \sum_{e \in E} \sum_{e' \in E'} k^{(1)}_{\mathrm{walk}}(e, e'), \tag{399}$$

where $k^{(1)}_{\mathrm{walk}}(e, e')$ is a positive definite kernel on edge walks of length 1.

The SP kernel can handle graphs either without node labels, with node labels, or with node attributes. This information, as well as the shortest path lengths, are encoded into $k^{(1)}_{\mathrm{walk}}$ whose classical choices are Dirac kernels or, more rarely, Brownian bridge kernels. The computation of the SP kernel is very expensive since it takes $\mathcal{O}(n_V)$ time, where n_V denotes the number of nodes.

We present the Neighborhood Subgraph Pairwise Distance kernel (Costa and Grave, 2010). This kernel extracts pairs of subgraphs from each graph and then compares these pairs.

Definition .33 (Neighborhood Subgraph Pairwise Distance kernel (Costa and Grave, 2010))**.** *Let $G = (V, E)$ and $G' = (V', E')$ be two node-labeled and egde-labeled graphs. For $u, v \in V$, $D(u, v)$ denotes the distance between u and v, i.e. the length of the shortest path between them, for $r \geq 1$, $\{u \in V : D(u, v) \leq r\}$ denotes the neighborhood of radius r of a vertex v, i.e. the set of vertices at a distance less than or equal to r from v, for a subset of vertices $S \subseteq V$, $E(S)$ denotes the set of edges that have both end-points in S, and we can define the subgraph with vertex set S and edge set $E(S)$. N^v_r denotes the subgraph induced by $\{u \in V : D(u, v) \leq r\}$. Let also $R_{r,d}(A_v, B_u, G)$ be a relation between two rooted graphs A_v, B_u and a graph $G = (V, E)$ that is true if and only if both A_v and B_u are in $\{N^v_r : v \in V\}$, where we require A_v, B_u to be isomorphic to some N^v_r to verify the set inclusion, and that $D(u, v) = d$. We denote with $R^{-1}(G)$ the inverse relation that yields all the pairs of rooted graphs A_v, B_u satisfying the above constraints. The neighborhood subgraph pairwise*

distance kernel is then based on the following kernel

$$k_{r,d}(G,G') = \sum_{A_v,B_u \in R_{r,d}^{-1}(G)} \sum_{A'_{v'},B'_{u'} \in R_{r,d}^{-1}(G')} \delta(A_v,A'_{v'})\delta(B_u,B'_{u'}), \tag{400}$$

where δ is 1 if its input subgraphs are isomorphic, and 0 otherwise. This counts the number of identical pairs of neighboring subgraphs of radius r at distance d between two graphs. The NSPD kernel is then defined on G and G' as

$$\mathrm{k}_{\mathcal{Y}}(G,G') = \sum_{r=0}^{r^*} \sum_{d=0}^{d^*} \hat{k}_{r,d}(G,G'), \tag{401}$$

where $\hat{k}_{r,d}$ is a normalized version of $k_{r,d}$, and r^* and d^* are hyper-parameters of the kernel.

The NSPD takes into account the edge labels, which can be of particular interest when manipulating molecules. For small values of r^* and d^*, its complexity is in practice linear in the size of the graph.

We now introduce the Weisfeiler-Lehman (WL) framework, inspired by the WL test of graph isomorphism (Weisfeiler and Leman, 1968), that operates on top of existing graph kernels. The WL algorithm replaces the label of each vertex with a multiset label consisting of the original label of the vertex and the sorted set of labels of its neighbors. The resulting multiset is then compressed into a new, short label, and this procedure is repeated for h iterations.

Definition .34 (Weisfeiler-Lehman kernel (Shervashidze et al., 2011))**.** *Let $G = (V,E)$ and $G' = (V',E')$ be two node-labeled graphs, endowed with labeling functions $\ell = \ell_0$ and $\ell' = \ell'_0$, respectively. The WL graph of G at height i is a graph G_i endowed with a labeling function ℓ_i which has emerged after i iterations of the relabeling procedure described previously. Let $\mathrm{k}_{\mathcal{Y}_{\text{base}}}$ be any kernel for graphs, called the base kernel. The WL kernel with h iterations is then defined on G and G' as*

$$\mathrm{k}_{\mathcal{Y}}(G,G') = \mathrm{k}_{\mathcal{Y}_{\text{base}}}(G_0,G'_0) + \ldots + \mathrm{k}_{\mathcal{Y}_{\text{base}}}(G_h,G'_h). \tag{402}$$

A very popular choice is the WL subtree kernel, which corresponds to choosing the VH kernel as the base kernel. Its time complexity is $\mathcal{O}(hn_E)$, where n_E denotes the number of edges, which is efficient. We call it the WL-VH kernel.

We finally present the Core kernel framework that, similarly to the WL framework, operates on top of existing graph kernels. It builds upon the notion of k-core decomposition, first introduced to study the cohesion of social networks (Seidman, 1983).

Definition .35 (Core kernel (Nikolentzos et al., 2018))**.** *Let $G = (V,E)$ and $G' = (V',E')$ be two graphs. Let $G_{\text{sub}}(S,E(S))$ be the subgraph induced by the subset of vertices $S \subseteq V$ and the set of edges $E(S)$ that have both end-points in S. Let $d_{G_{\text{sub}}}(v)$ be the degree of a vertex $v \in S$, i.e. the number of vertices that are adjacent to v in G_{sub}. The, G_{sub} is a k-core of G, denoted by C_k, if it is a maximal subgraph of G in which all vertices have a degree at least k. Let $\mathrm{k}_{\mathcal{Y}_{\text{base}}}$ be any kernel for graphs, called the base kernel. The core variant of this kernel is then defined on G and G' as*

$$\mathrm{k}_{\mathcal{Y}}(G,G') = \mathrm{k}_{\mathcal{Y}_{\text{base}}}(C_0,C'_0) + \ldots + \mathrm{k}_{\mathcal{Y}_{\text{base}}}(C_{\delta^*_{\min}},C'_{\delta^*_{\min}}), \tag{403}$$

where $\delta^*_{\min}$ *is the minimum of the degeneracies of the two graphs, and for all* $1 \leq i \leq \delta^*_{\min}$, C_i *and* C'_i *are the* i*-core subgraphs of* G *and* G'.

The time complexity of computing the k-core decomposition of a graph is $\mathcal{O}(n_V + n_E)$. Moreover, the complexity of computing the core variant of a kernel depends on its complexity, and in general, the complexity added by the core variant is not very high.

C.3 Additional experiments and details

In this section, we bring some additional experiments and details.

SMILES to Molecule

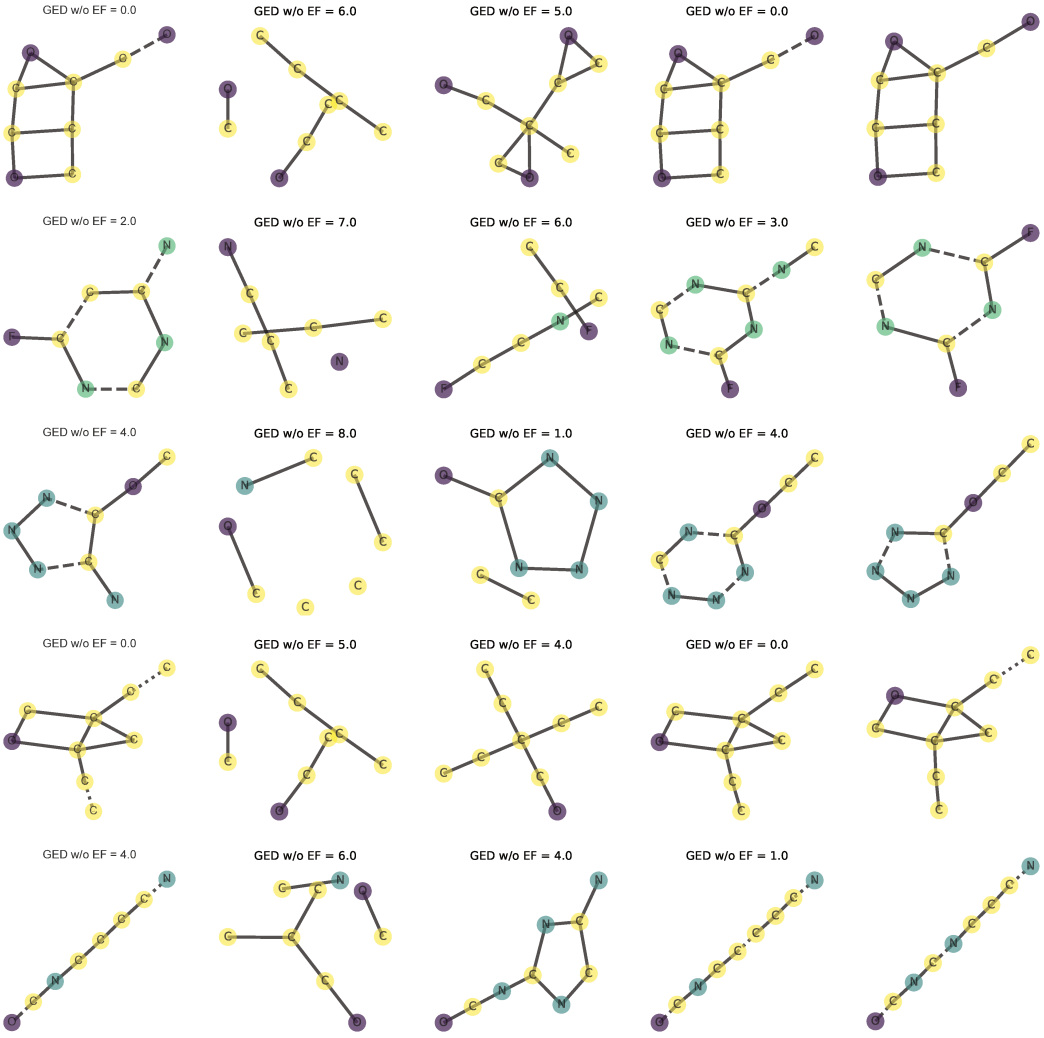
For DSOKR, we optimize the parameters of neural networks using Adam with a learning rate of 10^{-3} over 50 epochs. We adopt early stopping based on the validation set's edit distance. The number of transformer layers is chosen from $\{3, 6\}$, the model dimension is selected from $\{256, 512\}$, the number of heads is set to 8, the feed-forward network dimension is set to four times the model dimension, and the dropout probability is set to 0.2.

More examples of predictions can be found in Figure C.6.

Text to Molecule

For DSOKR, we conducted training on SciBERT for 50 epochs using the Adam optimizer with a learning rate of 3×10^{-5}. Additionally, we implemented a learning rate schedule that linearly decreases from the initial rate set by the optimizer to 0, following a warm-up period of 1000 steps where it linearly increases from 0 to the initial rate. We incorporated early stopping based on the MRR score on the validation set as well.

Figure C.7 presents the validation MRR with respect to $\mathrm{m}_\mathcal{Y}$ obtained by *Perfect h* with a Gaussian output kernel and additional values of γ. The best γ is clearly 10^{-6} since all sketching types attain the performance of the non-sketched *Perfect h*. Table .4 presents all the results gathered on ChEBI-20 with the additional Mean Rank metric. DSOKR under-performs in terms of mean rank compared with CMAM while outperforming it in terms of hits@1, attaining around 50%, and being equivalent to the ensemble CMAM methods in terms of hits@10, attaining around 88%, which means that most of the time, the correct molecule is predicted in the top rankings and even at the top position half of the time, but in the 12% left, the correct molecule falls to a high predicted rank.



(a) SISOKR (b) NNBary (c) ILE (d) DSOKR (e) True target

Figure C.6: More predicted molecules on the SMI2Mol dataset.

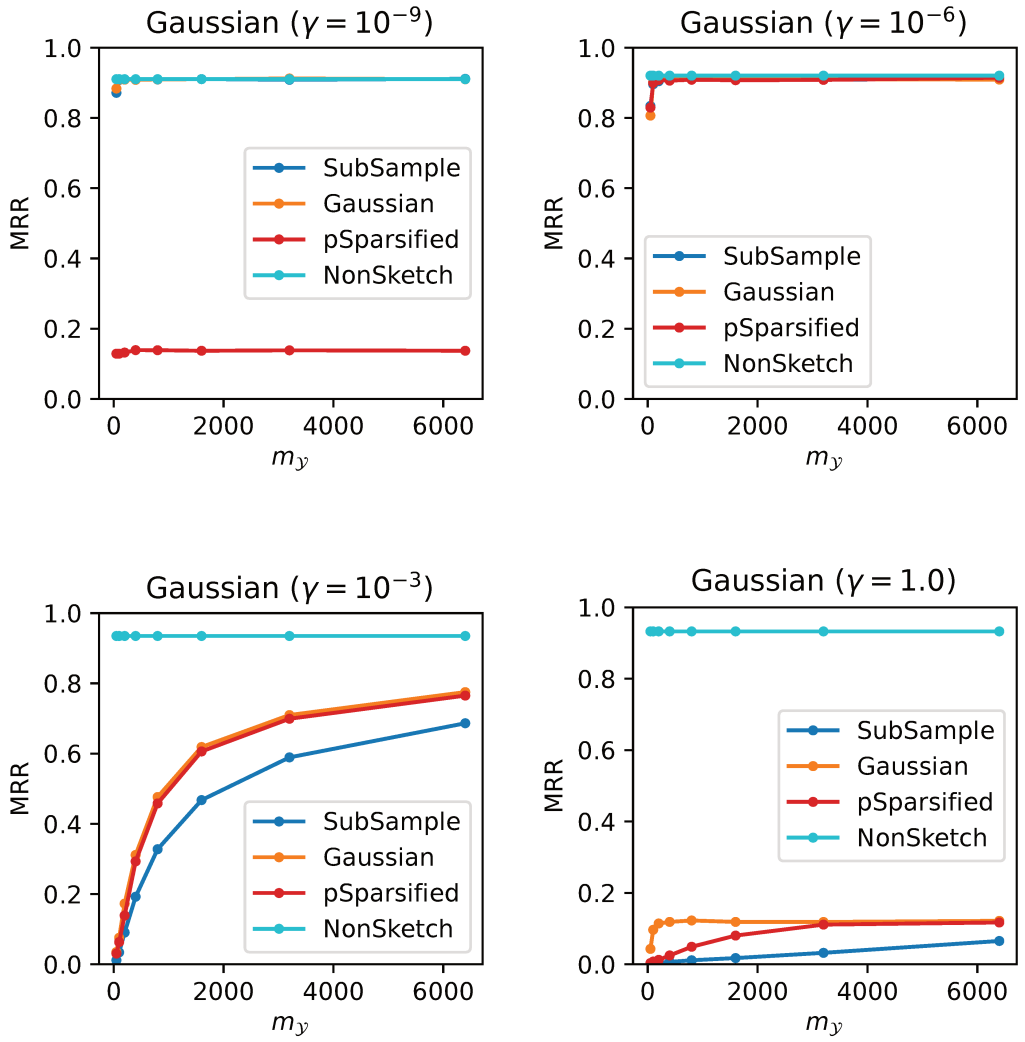


Figure C.7: The MRR scores on ChEBI-20 validation set with respect to the sketching size m_y for *Perfect h* when the output kernel is Gaussian with $\gamma \in \{10^{-9}, 10^{-6}, 10^{-3}, 1.0\}$.

Table .4: Performance of different methods on ChEBI-20 test set. All the methods based on NNs use SciBERT as input text encoder for fair comparison.

	Mean Rank ↓	MRR ↑	Hits@1 ↑	Hits@10 ↑
SISOKR	2230.48	0.015	0.4%	2.8%
SciBERT Regression	344.53	0.298	16.8%	56.9%
CMAM - MLP	23.74	0.513	34.9%	84.2%
CMAM - GCN	24.11	0.495	33.2%	82.5%
CMAM - Ensemble (MLP)	17.92	0.562	39.8%	87.6%
CMAM - Ensemble (GCN)	20.48	0.551	39.0%	87.0%
CMAM - Ensemble (MLP + GCN)	**16.28**	0.597	44.2%	**88.7**%
DSOKR - SubSample Sketch	82.92	0.624	48.2%	87.4%
DSOKR - Gaussian Sketch	91.19	0.630	49.0%	87.5%
DSOKR - Ensemble (SubSample Sketch)	76.43	**0.642**	**51.0**%	88.2%
DSOKR - Ensemble (Gaussian Sketch)	81.70	**0.642**	50.5%	87.9%
DSOKR - Ensemble (SubSample + Gaussian)	76.87	0.640	50.0%	88.3%

Bibliography

D. Achlioptas. Database-friendly random projections. In *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 274–281, 2001. page 73

A. Alacaoglu, O. Fercoq, and V. Cevher. Random extrapolation for primal-dual coordinate descent. In *Proc of the International Conference on Machine Learning (ICML)*, pages 191–201. PMLR, 2020. page 72

A. Alaoui and M. W. Mahoney. Fast randomized kernel ridge regression with statistical guarantees. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, 2015. pages 40, 77, 92, 106

M. A. Álvarez, L. Rosasco, and N. D. Lawrence. Kernels for vector-valued functions: a review. *Foundations and Trends in Machine Learning*, 4(3):195–266, 2012. pages 24, 26

W. Ammar, D. Groeneveld, C. Bhagavatula, I. Beltagy, M. Crawford, D. Downey, J. Dunkelberger, A. Elgohary, S. Feldman, V. Ha, R. Kinney, S. Kohlmeier, K. Lo, T. Murray, H.-H. Ooi, M. Peters, J. Power, S. Skjonsberg, L. L. Wang, C. Wilhelm, Z. Yuan, M. van Zuylen, and O. Etzioni. Construction of the Literature Graph in Semantic Scholar. In S. Bangalore, J. Chu-Carroll, and Y. Li, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 84–91, New Orleans - Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-3011. URL https://aclanthology.org/N18-3011. page 60

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, pages 337–404, 1950. pages 21, 65

F. Bach. Sharp analysis of low-rank kernel matrix approximations. In *Proc. of the 26th annual Conference on Learning Theory*, pages 185–209. PMLR, 2013. pages 38, 86

F. R. Bach, G. R. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *Proceedings of the twenty-first international conference on Machine learning*, page 6, 2004. page 58

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.0473. page 27

D. Bajusz, A. Rácz, and K. Héberger. *Chemical Data Formats, Fingerprints, and Other Molecular Descriptions for Database Analysis and Searching*. 12 2017. ISBN 9780124095472. doi: 10.1016/B978-0-12-409547-2.12345-5. page 14

G. Bakir, T. Hofmann, A. J. Smola, B. Schölkopf, and B. Taskar. *Predicting structured data*. The MIT Press, 2007. page 33

L. Baldassarre, L. Rosasco, A. Barla, and A. Verri. Multi-output learning via spectral filtering. *Machine Learning*, 87(3):259–301, 2012. page 26

R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3):253–263, 2008. page 73

P. L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *J. Mach. Learn. Res.*, 3:463–482, 2003. pages 46, 47, 68, 120, 121

P. L. Bartlett, O. Bousquet, and S. Mendelson. Local rademacher complexities. *Ann. Statist.*, 33(4):1497–1537, 2005. pages 49, 68

P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006. pages 66, 123

F. Bauer, S. Pereverzev, and L. Rosasco. On regularization algorithms in learning theory. *Journal of Complexity*, 23(1):52–72, 2007. page 88

D. Belanger and A. McCallum. Structured prediction energy networks. In *International Conference on Machine Learning*, pages 983–992, 2016. pages 13, 27, 34, 94

D. Belanger, B. Yang, and A. McCallum. End-to-end learning for structured prediction energy networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 429–439. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/belanger17a.html. pages 13, 27, 35, 117

S. Belharbi, R. Hérault, C. Chatelain, and S. Adam. Deep neural networks regularization for structured output prediction. *Neurocomputing*, 281, 12 2017. doi: 10.1016/j.neucom.2017.12.002. page 27

I. Beltagy, K. Lo, and A. Cohan. SciBERT: A Pretrained Language Model for Scientific Text. In K. Inui, J. Jiang, V. Ng, and X. Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1371. URL https://aclanthology.org/D19-1371. pages 60, 110

Q. Berthet, M. Blondel, O. Teboul, M. Cuturi, J.-P. Vert, and F. Bach. Learning with differentiable perturbed optimizers, 2020. pages 28, 117

G. Blanchard, O. Bousquet, and L. Zwald. Statistical properties of kernel principal component analysis. *Machine Learning*, 66(2):259–294, 2007. page 89

M. Blondel, A. F. Martins, and V. Niculae. Learning with fenchel-young losses. *Journal of Machine Learning Research*, 21(35):1–69, 2020. page 29

K. Borgwardt and H. Kriegel. Shortest-path kernels on graphs. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 8 pp.–, 2005. doi: 10.1109/ICDM.2005.132. page 172

K. Borgwardt, E. Ghisu, F. Llinares-López, L. O'Bray, and B. Rieck. Graph kernels: State-of-the-art and future challenges. *Foundations and Trends in Machine Learning*, 13(5-6):531–712, 2020. pages 99, 100

B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, page 144–152, New York, NY, USA, 1992. Association for Computing Machinery. ISBN 089791497X. doi: 10.1145/130385.130401. URL https://doi.org/10.1145/130385.130401. page 54

D. Bouche, M. Clausel, F. Roueff, and F. d'Alché Buc. Nonlinear functional output regression: A dictionary approach. In *International Conference on Artificial Intelligence and Statistics*, pages 235–243. PMLR, 2021. page 26

S. Boucheron, G. Lugosi, and P. Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013. pages 43, 73, 136, 137, 139

O. Bousquet and A. Elisseeff. Stability and generalization. *Journal of machine learning research*, 2(Mar):499–526, 2002. page 61

R. Brault, M. Heinonen, and F. Buc. Random fourier features for operator-valued kernels. In *Asian Conference on Machine Learning*, pages 110–125. PMLR, 2016. page 37

R. Brault, A. Lambert, Z. Szabo, M. Sangnier, and F. d'Alché-Buc. Infinite task learning in rkhss. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1294–1302, 2019. page 57

L. Brogat-Motte, R. Flamary, C. Brouard, J. Rousu, and F. D'Alché-Buc. Learning to Predict Graphs with Fused Gromov-Wasserstein Barycenters. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 2321–2335. PMLR, July 2022a. URL https://proceedings.mlr.press/v162/brogat-motte22a.html. page 108

L. Brogat-Motte, A. Rudi, C. Brouard, J. Rousu, and F. d'Alché Buc. Vector-valued least-squares regression under output regularity assumptions. *Journal of Machine Learning Research*, 23(344):1–50, 2022b. URL http://jmlr.org/papers/v23/21-1357.html. pages 26, 88, 95, 116

C. Brouard, F. d'Alché-Buc, and M. Szafranski. Semi-supervised penalized output kernel regression for link prediction. In *International Conference on Machine Learning (ICML)*, pages 593–600, 2011. pages 26, 28, 83, 84, 99

C. Brouard, H. Shen, K. Dührkop, F. d'Alché-Buc, S. Böcker, and J. Rousu. Fast metabolite identification with input output kernel regression. *Bioinformatics*, 32(12): 28–36, 2016a. pages 13, 26, 27, 28, 29, 46, 58, 95, 104, 171

C. Brouard, M. Szafranski, and F. D'Alché-Buc. Input output kernel regression: supervised and semi-supervised structured output prediction with operator-valued kernels. *The Journal of Machine Learning Research*, 17(1):6105–6152, 2016b. pages 14, 15, 26, 27, 28, 30, 31, 54, 84, 94, 99, 171

A. Caponnetto and E. De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007. pages 24, 45, 48, 49, 51, 52, 67, 68, 70, 88, 89

C. Carmeli, E. De Vito, and A. Toigo. Vector valued reproducing kernel hilbert spaces of integrable functions and mercer theorem. *Analysis and Applications*, 4(04):377–408, 2006. page 84

C. Carmeli, E. De Vito, A. Toigo, and V. Umanitá. Vector valued reproducing kernel hilbert spaces and universality. *Analysis and Applications*, 8(01):19–61, 2010. pages 24, 84, 88

A. Chambolle, M. J. Ehrhardt, P. Richtárik, and C.-B. Schonlieb. Stochastic primal-dual hybrid gradient algorithm with arbitrary sampling and imaging applications. *SIAM Journal on Optimization*, 28(4):2783–2808, 2018. page 72

Z. Charles and D. Papailiopoulos. Stability and generalization of learning algorithms that converge to global optima. In *International conference on machine learning*, pages 745–754. PMLR, 2018. page 61

A. Chatalic, L. Carratino, E. De Vito, and L. Rosasco. Mean nyström embeddings for adaptive compressive learning. In G. Camps-Valls, F. J. R. Ruiz, and I. Valera, editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 9869–9889. PMLR, 28–30 Mar 2022. URL https://proceedings.mlr.press/v151/chatalic22a.html. page 71

D. Chen, L. Jacob, and J. Mairal. Recurrent kernel networks. *Advances in Neural Information Processing Systems*, 32, 2019. page 61

D. Chen, L. Jacob, and J. Mairal. Convolutional kernel networks for graph-structured data. In *International Conference on Machine Learning*, pages 1576–1586. PMLR, 2020. page 61

Y. Chen and Y. Yang. Accumulations of projections—a unified framework for random sketches in kernel ridge regression. In *International Conference on Artificial Intelligence and Statistics*, pages 2953–2961. PMLR, 2021a. pages 43, 44, 66, 67, 76, 77, 92, 150

Y. Chen and Y. Yang. Fast statistical leverage score approximation in kernel ridge regression. In *International Conference on Artificial Intelligence and Statistics*, pages 2935–2943. PMLR, 2021b. page 77

K. C. Cheng and J. B. Schenkman. Testosterone metabolism by cytochrome p-450 isozymes rlm3 and rlm5 and by microsomes. metabolite identification. *Journal of Biological Chemistry*, 258(19):11738–11744, 1983. ISSN 0021-9258. doi: https://doi.org/10.1016/S0021-9258(17)44291-8. URL https://www.sciencedirect.com/science/article/pii/S0021925817442918. page 13

F. Cherfaoui, H. Kadri, and L. Ralaivola. Scalable ridge leverage score sampling for the nyström method. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4163–4167, 2022. doi: 10.1109/ICASSP43922.2022.9747039. page 40

C. Ciliberto, L. Rosasco, and A. Rudi. A consistent regularization approach for structured prediction. In *Advances in Neural Information Processing Systems (NIPS) 29*, pages 4412–4420, 2016. pages 26, 28, 29, 32, 49, 51, 83, 84

C. Ciliberto, L. Rosasco, and A. Rudi. A general framework for consistent structured prediction with implicit loss embeddings. *J. Mach. Learn. Res.*, 21(98):1–67, 2020. pages 14, 17, 26, 28, 29, 30, 32, 49, 50, 51, 70, 84, 85, 88, 89, 91, 101, 116, 151, 152, 163

K. L. Clarkson and D. P. Woodruff. Low-rank approximation and regression in input sparsity time. *J. ACM*, 63(6), jan 2017. ISSN 0004-5411. doi: 10.1145/3019134. URL https://doi.org/10.1145/3019134. pages 43, 76, 77, 150

M. B. Cohen. Nearly tight oblivious subspace embeddings by trace inequalities. In *Proceedings of the 2016 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 278–287, 2016. page 73

L. Condat. A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms. *Journal of optimization theory and applications*, 158(2):460–479, 2013. page 72

C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995. pages 23, 52, 67, 71

C. Cortes, M. Mohri, and J. Weston. A general regression technique for learning transductions. In *International Conference on Machine Learning (ICML)*, pages 153–160, 2005. pages 28, 30, 31, 83, 84, 99

C. Cortes, M. Mohri, and J. Weston. A general regression framework for learning string-to-string mappings. In *Predicting Structured Data*, 2007. URL http://www.cs.nyu.edu/~mohri/postscript/sts.pdf. pages 13, 27, 28, 100, 104

C. Cortes, M. Mohri, and A. Rostamizadeh. Algorithms for learning kernels based on centered alignment. *The Journal of Machine Learning Research*, 13(1):795–828, 2012. page 58

F. Costa and K. D. Grave. Fast Neighborhood Subgraph Pairwise Distance Kernel. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 255–262, Madison, WI, USA, 2010. Omnipress. ISBN 978-1-60558-907-7. event-place: Haifa, Israel. pages 108, 172

A. Damianou and N. D. Lawrence. Deep gaussian processes. In *Artificial intelligence and statistics*, pages 207–215. PMLR, 2013. page 59

M. Derezinski, Z. Liao, E. Dobriban, and M. W. Mahoney. Sparse sketches with small inversion bias. In *COLT*, 2021. page 76

A. Deshwal, J. R. Doppa, and D. Roth. Learning and inference for structured prediction: A unifying perspective. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*, 2019. page 33

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NACCL-HLT)*, pages 4171–4186, June 2019. doi:

10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423. page 60

F. Dinuzzo, C. Ong, P. Gehler, and G. Pillonetto. Learning output kernels with block coordinate descent. In *International Conference on Machine Learning (ICML)*, pages 49–56, 2011. pages 58, 115

P. Drineas, M. W. Mahoney, and N. Cristianini. On the nyström method for approximating a gram matrix for improved kernel-based learning. *JMLR*, 6(12), 2005. pages 38, 85

H. Drucker, C. J. Burges, L. Kaufman, A. J. Smola, and V. Vapnik. Support vector regression machines. In *Advances in neural information processing systems*, pages 155–161, 1997. pages 23, 55, 67

K. Dührkop. Deep kernel learning improves molecular fingerprint prediction from tandem mass spectra. *Bioinformatics*, 38(Supplement_1):i342–i349, 2022. page 59

C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, aug 2014. ISSN 1551-305X. doi: 10.1561/0400000042. URL https://doi.org/10.1561/0400000042. page 115

C. Edwards, C. Zhai, and H. Ji. Text2Mol: Cross-Modal Molecule Retrieval with Natural Language Queries. In M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 595–607, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.47. URL https://aclanthology.org/2021.emnlp-main.47. pages 110, 111

T. El Ahmad, P. Laforgue, and F. d'Alché Buc. Fast kernel methods for generic lipschitz losses via p-sparsified sketches. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=ry2qgRqTOw. pages 18, 90, 92, 110, 115

T. El Ahmad, L. Brogat-Motte, P. Laforgue, and F. d'Alché Buc. Sketch in, sketch out: Accelerating both learning and inference for structured prediction with kernels. In S. Dasgupta, S. Mandt, and Y. Li, editors, *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 109–117. PMLR, 02–04 May 2024. URL https://proceedings.mlr.press/v238/el-ahmad24a.html. pages 18, 101, 102, 108

T. El Ahmad, J. Yang, P. Laforgue, and F. d'Alché Buc. Deep sketched output kernel regression for structured prediction, 2024. page 18

T. Evgeniou, C. A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6(21):615–637, 2005. pages 25, 69

O. Fercoq and P. Bianchi. A coordinate-descent primal-dual algorithm with large step size and possibly nonseparable functions. *SIAM Journal on Optimization*, 29 (1):100–134, Jan 2019. page 72

S. Fischer and I. Steinwart. Sobolev norm learning rates for regularized least-squares algorithms. *J. Mach. Learn. Res.*, 21:205–1, 2020. page 88

J. H. Friedman. Multivariate adaptive regression splines. *The annals of statistics*, pages 1–67, 1991. page 77

T. Gärtner. *Kernels for Structured Data*, volume 72 of *Series in Machine Perception and Artificial Intelligence*. WorldScientific, 2008. page 99

D. M. Gentile, E. S. Tomlinson, J. L. Maggs, B. K. Park, and D. J. Back. Dexamethasone metabolism by human liver in vitro. metabolite identification and inhibition of 6-hydroxylation. *Journal of Pharmacology and Experimental Therapeutics*, 277(1):105–112, 1996. page 13

P. Geurts, L. Wehenkel, and F. d'Alché Buc. Kernelizing the output of tree-based methods. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 345–352, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933832. pages 28, 83, 99, 117

L. Giffon, S. Ayache, T. Artières, and H. Kadri. Deep networks with adaptive nyström approximation. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019. page 61

A. Gittens and M. Mahoney. Revisiting the nystrom method for improved large-scale machine learning. *Proceedings of the 30th International Conference on Machine Learning*, 28(3):567–575, 17–19 Jun 2013. page 77

A. Gittens and M. W. Mahoney. Revisiting the nyström method for improved large-scale machine learning. *The Journal of Machine Learning Research*, 17(1):3977–4041, 2016. page 40

M. Gönen and E. Alpaydin. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12(64):2211–2268, 2011. URL http://jmlr.org/papers/v12/gonen11a.html. page 58

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. pages 35, 60

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org. page 59

R. M. Gower, P. Richtárik, and F. Bach. Stochastic quasi-gradient methods: Variance reduction via jacobian sketching. *Mathematical Programming*, 188(1):135–192, 2021. page 40

C. Graber, O. Meshi, and A. Schwing. Deep structured prediction with nonlinear output transformations. *Advances in Neural Information Processing Systems*, 31, 2018. page 35

W. Groves and M. Gini. On optimizing airline ticket purchase timing. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 7(1):1–28, 2015. page 149

S. Grünewälder, G. Lever, L. Baldassarre, S. Patterson, A. Gretton, and M. Pontil. Conditional mean embeddings as regressors. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1803–1810, 2012. page 84

M. Gygli, M. Norouzi, and A. Angelova. Deep value networks learn to evaluate and iteratively refine structured outputs. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1341–1351. JMLR.org, 2017. pages 35, 94

A. A. Hagberg, D. A. Schult, and P. J. Swart. Exploring Network Structure, Dynamics, and Function using NetworkX. In G. Varoquaux, T. Vaught, and J. Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008. page 108

F. Harder, K. Adamczewski, and M. Park. Dp-merf: Differentially private mean embeddings with randomfeatures for practical privacy-preserving data generation. In *International conference on artificial intelligence and statistics*, pages 1819–1827. PMLR, 2021. page 116

D. Harrison Jr and D. L. Rubinfeld. Hedonic housing prices and the demand for clean air. *Journal of environmental economics and management*, 5(1):81–102, 1978. page 79

Y. Hashimoto, M. Ikeda, and H. Kadri. Deep learning with kernels through rkhm and the perron-frobenius operator. *Advances in Neural Information Processing Systems*, 36, 2024. page 61

J. Hastings, G. Owen, A. Dekker, M. Ennis, N. Kale, V. Muthukrishnan, S. Turner, N. Swainston, P. Mendes, and C. Steinbeck. ChEBI in 2016: Improved services and an expanding collection of metabolites. *Nucleic Acids Research*, 44(D1):D1214–D1219, 2016. ISSN 0305-1048. doi: 10.1093/nar/gkv1031. URL https://doi.org/10.1093/nar/gkv1031. page 110

H. He, G. Huang, and Y. Yuan. Asymmetric valleys: Beyond sharp and flat local minima. *Advances in neural information processing systems*, 32, 2019. page 61

P. J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, pages 73–101, 1964. pages 52, 55, 67

A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018. page 61

S. Jaeger, S. Fulle, and S. Turk. Mol2vec: Unsupervised Machine Learning Approach with Chemical Intuition. *Journal of Chemical Information and Modeling*, 58(1):27–35, 2018. ISSN 1549-9596. page 110

P. Jain and A. Thakurta. Differentially private learning with kernels. In *International conference on machine learning*, pages 118–126. PMLR, 2013. page 115

W. B. Johnson and J. Lindenstrauss. Extensions of lipschitz mappings into a hilbert space 26. *Contemporary mathematics*, 26:28, 1984. pages 41, 42

J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021. pages 13, 27, 28

H. Kadri, E. Duflos, P. Preux, S. Canu, and M. Davy. Nonlinear functional regression: a functional rkhs approach. In Y. W. Teh and M. Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 374–380. PMLR, 13–15 May 2010. page 26

H. Kadri, A. Rakotomamonjy, P. Preux, and F. Bach. Multiple operator-valued kernel learning. *Advances in Neural Information Processing Systems*, 25, 2012. page 58

H. Kadri, M. Ghavamzadeh, and P. Preux. A generalized kernel approach to structured output learning. In *International Conference on Machine Learning (ICML)*, pages 471–479, 2013a. pages 28, 100

H. Kadri, M. Ghavamzadeh, and P. Preux. A generalized kernel approach to structured output learning. In S. Dasgupta and D. McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 471–479, Atlanta, Georgia, USA, 17–19 Jun 2013b. PMLR. URL http://proceedings.mlr.press/v28/kadri13.html. pages 26, 84

H. Kadri, E. Duflos, P. Preux, S. Canu, A. Rakotomamonjy, and J. Audiffren. Operator-valued kernels for learning from functional response data. *Journal of Machine Learning Research*, 17(20):1–54, 2016. URL http://jmlr.org/papers/v17/11-315.html. page 26

S. Kim, P. A. Thiessen, E. E. Bolton, J. Chen, G. Fu, A. Gindulyte, L. Han, J. He, S. He, B. A. Shoemaker, J. Wang, B. Yu, J. Zhang, and S. H. Bryant. PubChem Substance and Compound databases. *Nucleic Acids Research*, 44(D1):D1202–D1213, 2016. ISSN 0305-1048. doi: 10.1093/nar/gkv951. URL https://doi.org/10.1093/nar/gkv951. page 110

S. Kim, J. Chen, T. Cheng, A. Gindulyte, J. He, S. He, Q. Li, B. A. Shoemaker, P. A. Thiessen, B. Yu, L. Zaslavsky, J. Zhang, and E. E. Bolton. PubChem 2019 update: improved access to chemical data. *Nucleic Acids Research*, 47(D1):D1102–D1109, 2019. ISSN 0305-1048. doi: 10.1093/nar/gky1033. URL https://doi.org/10.1093/nar/gky1033. page 110

G. Kimeldorf and G. Wahba. Some results on tchebycheffian spline functions. *Journal of mathematical analysis and applications*, 33(1):82–95, 1971. pages 23, 66

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. pages 61, 71, 107

D. P. Kingma and M. Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013. URL https://api.semanticscholar.org/CorpusID:216078090. page 60

A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. pages 13, 27, 28

R. Koenker. *Quantile regression*. Cambridge university press, 2005. pages 52, 57

R. Koenker and G. Bassett Jr. Regression quantiles. *Econometrica: journal of the Econometric Society*, pages 33–50, 1978. page 57

V. Koltchinskii and K. Lounici. Concentration inequalities and moment bounds for sample covariance operators. *Bernoulli*, 23(1):110 – 133, 2017. doi: 10.3150/15-BEJ730. URL https://doi.org/10.3150/15-BEJ730. pages 52, 90, 164, 165

V. Koltchinskii and M. Yuan. Sparsity in multiple kernel learning. *The Annals of Statistics*, 38(6), Dec 2010. ISSN 0090-5364. doi: 10.1214/10-aos825. URL http://dx.doi.org/10.1214/10-AOS825. page 58

A. Korba, A. Garcia, and F. d'Alché-Buc. A structured prediction approach for label ranking. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/b3dd760eb02d2e669c604f6b2f1e803f-Paper.pdf. pages 13, 28, 29, 99, 100, 104

S. Kpotufe and B. K. Sriperumbudur. Gaussian sketching yields a J-L lemma in RKHS. In S. Chiappa and R. Calandra, editors, *AISTATS 2020*, volume 108 of *Proceedings of Machine Learning Research*, pages 3928–3937. PMLR, 2020. pages 44, 86

S. Kumar, M. Mohri, and A. Talwalkar. Sampling methods for the nyström method. *J. Mach. Learn. Res.*, 13:981–1006, 2012. pages 40, 77

J. Lacotte and M. Pilanci. Adaptive and oblivious randomized subspace methods for high-dimensional optimization: Sharp analysis and lower bounds. *IEEE Transactions on Information Theory*, 68(5):3281–3303, 2022. pages 44, 45, 66, 86

J. Lacotte, M. Pilanci, and M. Pavone. High-dimensional optimization in adaptive random subspaces. In *Proc. of the 33rd International Conference on Neural Information Processing Systems*, pages 10847–10857, 2019. page 45

J. D. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*, 2001. URL https://api.semanticscholar.org/CorpusID:219683473. pages 27, 33

P. Laforgue, S. Clémençon, and F. d'Alché-Buc. Autoencoding any data through kernel autoencoders. In K. Chaudhuri and M. Sugiyama, editors, *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, volume 89 of *Proceedings of Machine Learning Research*, pages 1061–1069. PMLR, 2019. pages 61, 117

P. Laforgue, A. Lambert, L. Brogat-Motte, and F. d'Alché Buc. Duality in rkhss with infinite dimensional outputs: Application to robust losses. In *International Conference on Machine Learning*, pages 5598–5607. PMLR, 2020. pages 26, 27, 32, 56, 71, 96, 104, 171

A. Lambert, D. Bouche, Z. Szabo, and F. d'Alché Buc. Functional output regression with infimal convolution: Exploring the huber and ε-insensitive losses. In *International Conference on Machine Learning*, pages 11844–11867. PMLR, 2022. page 26

G. R. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine learning research*, 5(Jan):27–72, 2004. page 58

Y. LeCun, Y. Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995. page 60

P. Li, T. J. Hastie, and K. W. Church. Very sparse random projections. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 287–296, 2006. page 73

Y. Li, C. Wei, and T. Ma. Towards explaining the regularization effect of initial large learning rate in training neural networks. *Advances in neural information processing systems*, 32, 2019. page 61

Z. Li, J.-F. Ton, D. Oglic, and D. Sejdinovic. Towards a unified analysis of random fourier features. *Journal of Machine Learning Research*, 22(108):1–51, 2021. pages 44, 45, 48, 49, 67, 68, 74, 103, 120, 123

H. Lian. Nonlinear functional models for functional responses in reproducing kernel Hilbert spaces. *Canadian Journal of Statistics*, pages 597–606, 2007. page 26

N. Lim, F. d'Alché Buc, C. Auliac, and G. Michailidis. Operator-valued Kernel-based Vector Autoregressive Models for Network Inference. *Machine Learning*, 99(3):489–513, June 2015. pages 58, 115

X. V. Lin, S. Singh, L. He, B. Taskar, and L. Zettlemoyer. Multi-label learning with posterior regularization, 2014. page 94

E. Lindgren, S. Reddi, R. Guo, and S. Kumar. Efficient training of retrieval models using negative cache. *Advances in Neural Information Processing Systems*, 34:4134–4146, 2021. page 28

D. C. Liu and J. Nocedal. On the limited memory bfgs method for large scale optimization. *Math. Program.*, 45(1–3):503–528, 1989. page 61

F. Liu, W. Xu, J. Lu, G. Zhang, A. Gretton, and D. J. Sutherland. Learning deep kernels for non-parametric two-sample tests. In *International conference on machine learning*, pages 6316–6326. PMLR, 2020. page 59

M. Liu, Z. Shang, and G. Cheng. Sharp theoretical analysis for nonparametric testing under random projection. In A. Beygelzimer and D. Hsu, editors, *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 2175–2209. PMLR, 25–28 Jun 2019. page 66

H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. Text classification using string kernels. *Journal of machine learning research*, 2(Feb):419–444, 2002. page 21

J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. page 28

M. W. Mahoney et al. Randomized algorithms for matrices and data. *Foundations and Trends® in Machine Learning*, 3(2):123–224, 2011. pages 40, 101, 102

J. Mairal, P. Koniusz, Z. Harchaoui, and C. Schmid. Convolutional kernel networks, 2014. page 61

J. Matoušek. *Lectures on Discrete Geometry*. Graduate Texts in Mathematics. Springer, 2013. page 138

A. Maurer. A vector-contraction inequality for rademacher complexities. In *International Conference on Algorithmic Learning Theory*, pages 3–17. Springer, 2016. pages 47, 48, 124, 128

C. McDiarmid et al. On the method of bounded differences. *Surveys in combinatorics*, 141(1):148–188, 1989. page 47

G. Meanti, L. Carratino, L. Rosasco, and A. Rudi. Kernel methods through the roof: Handling billions of points efficiently. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 2020. pages 40, 102, 116

G. Meanti, A. Chatalic, V. R. Kostic, P. Novelli, massimiliano pontil, and L. Rosasco. Estimating koopman operators with sketching to provably learn large scale dynamical systems. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=GItLpB1vhK. page 40

C. A. Micchelli and M. Pontil. On learning vector-valued functions. *Neural computation*, 17(1):177–204, 2005. pages 24, 26, 31, 70, 84, 104

E. Moen, N. O. Handegard, V. Allken, O. T. Albert, A. Harbitz, and K. Malde. Automatic interpretation of otoliths using deep learning. *PLoS One*, 13(12):e0204713, 2018. page 79

C. Musco and C. Musco. Recursive sampling for the nyström method. *Advances in Neural Information Processing Systems*, 2017:3834–3846, 2017. pages 40, 77

J. Nelson and H. L. Nguyên. Osnap: Faster numerical linear algebra algorithms via sparser subspace embeddings. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 117–126, 2013. doi: 10.1109/FOCS.2013.21. page 76

V. Niculae and A. Martins. LP-SparseMAP: Differentiable relaxed optimization for sparse structured prediction. In H. D. III and A. Singh, editors, *Proceedings of International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7348–7359, 2020. pages 28, 117

V. Niculae, A. Martins, M. Blondel, and C. Cardie. Sparsemap: Differentiable sparse structured inference. In *International Conference on Machine Learning (ICML)*, pages 3799–3808. PMLR, 2018. page 28

G. Nikolentzos, P. Meladianos, S. Limnios, and M. Vazirgiannis. A Degeneracy Framework for Graph Similarity. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 2595–2601. International Joint Conferences on Artificial Intelligence Organization, July 2018. doi: 10.24963/ijcai.2018/360. URL https://doi.org/10.24963/ijcai.2018/360. pages 108, 173

A. Nowak, F. Bach, and A. Rudi. Consistent structured prediction with max-min margin Markov networks. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7381–7391. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/nowak20a.html. pages 33, 100

A. Nowak-Vila, F. Bach, and A. Rudi. A general theory for structured prediction with smooth convex surrogates, 2019. page 33

S. Nowozin and C. H. Lampert. Structured learning and prediction in computer vision. *Foundations and Trends in Computer Graphics and Vision*, 6(3-4):185–365, 2011. pages 33, 34

A. Ordoñez, L. Eikvil, A.-B. Salberg, A. Harbitz, S. M. Murray, and M. C. Kampffmeyer. Explaining decisions of deep neural networks used for fish age prediction. *PloS one*, 15(6):e0235013, 2020. page 79

M. Pilanci and M. J. Wainwright. Iterative hessian sketch: Fast and accurate solution approximation for constrained least-squares. *Journal of Machine Learning Research*, 17(53):1–38, 2016. page 40

L. Pillaud-Vivien, A. Rudi, and F. Bach. Statistical optimality of stochastic gradient descent on hard learning problems through multiple passes. *Advances in Neural Information Processing Systems*, 31, 2018. pages 88, 89

X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897, 2020. page 60

A. Rahimi and B. Recht. Random features for large scale kernel machines. *NIPS*, 20: 1177–1184, 01 2007. pages 36, 37, 77, 103

A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. SimpleMKL. *Journal of Machine Learning Research*, 9:2491–2521, 2008. page 58

L. Ralaivola, S. J. Swamidass, H. Saigo, and P. Baldi. Graph kernels for chemical informatics. *Neural Networks*, 18(8):1093–1110, 2005. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2005.07.009. URL https://www.sciencedirect.com/science/article/pii/S0893608005001693. Neural Networks and Kernel Methods for Structured Domains. pages 14, 95, 171

R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1, 2014. Publisher: Nature Publishing Group. page 107

D. Richards and I. Kuzborskij. Stability & generalisation of gradient descent for shallow neural networks without the neural tangent kernel. *Advances in neural information processing systems*, 34:8609–8621, 2021. page 61

H. Robbins and S. Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951. page 61

L. Ruddigkeit, R. van Deursen, L. C. Blum, and J.-L. Reymond. Enumeration of 166 Billion Organic Small Molecules in the Chemical Universe Database GDB-17. *Journal of Chemical Information and Modeling*, 52(11):2864–2875, Nov. 2012. ISSN 1549-9596. doi: 10.1021/ci300415d. URL https://doi.org/10.1021/ci300415d. Publisher: American Chemical Society. page 107

A. Rudi and L. Rosasco. Generalization properties of learning with random features. In *Advances on Neural Information Processing Systems (NeurIPS)*, pages 3215–3225, 2017. pages 37, 44, 45, 48, 52, 67, 103

A. Rudi, G. D. Canas, and L. Rosasco. On the sample complexity of subspace learning. In *Advances in Neural Information Processing Systems*, pages 2067–2075, 2013. page 163

A. Rudi, R. Camoriano, and L. Rosasco. Less is more: Nyström computational regularization. *Advances in Neural Information Processing Systems*, 28, 2015. pages 17, 38, 39, 40, 44, 45, 52, 66, 86, 87, 88, 89, 90, 91, 92, 102, 108, 110, 116, 153, 154, 157, 160, 1

A. Rudi, L. Carratino, and L. Rosasco. Falkon: an optimal large scale kernel method. In *Proceedings of the 31st International Conference on Advances on Neural Information Processing Systems (NeurIPS)*, pages 3891–3901, 2017. page 40

A. Rudi, D. Calandriello, L. Carratino, and L. Rosasco. On fast leverage score sampling and optimal learning. In *NeurIPS*, 2018a. pages 40, 77

A. Rudi, C. Ciliberto, G. Marconi, and L. Rosasco. Manifold structured prediction. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018b. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/f6185f0ef02dcaec414a3171cd01c697-Paper.pdf. page 13

D. E. Rumelhart, R. Durbin, R. Golden, and Y. Chauvin. Backpropagation: The basic theory. In *Backpropagation*, pages 1–34. Psychology Press, 2013. page 61

H. Sak, A. W. Senior, and F. Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *INTERSPEECH*, pages 338–342, 2014. page 60

M. Sangnier, O. Fercoq, and F. d'Alché Buc. Joint quantile regression in vector-valued RKHSs. In *Advances in Neural Information Processing Systems (NeurIPS)*, Barcelona, France, Dec. 2016. pages 25, 26, 57, 69, 79

M. Sangnier, O. Fercoq, and F. d'Alché-Buc. Data sparse nonparametric regression with ϵ-insensitive losses. In *Asian Conference on Machine Learning (ACML)*, pages 192–207, 2017. pages 26, 56, 57

F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009. doi: 10.1109/TNN.2008.2005605. page 60

J. Schmidt-Hieber. Nonparametric regression using deep neural networks with relu activation function. *Annals of Statistics*, 48, 08 2017. doi: 10.1214/19-AOS1875. pages 61, 116

B. Scholkopf and A. J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. Adaptive Computation and Machine Learning Series, 2018. page 20

B. Schölkopf, A. Smola, and K.-R. Müller. Kernel principal component analysis. In *International conference on Artificial Neural Networks (ICANN)*, pages 583–588. Springer, 1997. page 101

B. Schölkopf, R. Herbrich, and A. J. Smola. A generalized representer theorem. In *Computational Learning Theory: 14th Annual Conference on Computational Learning Theory, COLT 2001 and 5th European Conference on Computational Learning Theory, EuroCOLT 2001 Amsterdam, The Netherlands, July 16–19, 2001 Proceedings 14*, pages 416–426. Springer, 2001. page 66

E. Scornet. On the asymptotics of random forests. *Journal of Multivariate Analysis*, 146:72–83, 2016. page 117

E. Scornet, G. Biau, and J.-P. Vert. Consistency of random forests. *The Annals of Statistics*, 43(4):1716 – 1741, 2015. doi: 10.1214/15-AOS1321. URL https://doi.org/10.1214/15-AOS1321. page 117

S. B. Seidman. Network structure and minimum degree. *Social Networks*, 5(3):269–287, 1983. ISSN 0378-8733. doi: https://doi.org/10.1016/0378-8733(83)90028-X. URL https://www.sciencedirect.com/science/article/pii/037887338390028X. page 173

E. Senkene and A. Tempel'man. Hilbert spaces of operator-valued functions. *Lithuanian Mathematical Journal*, 13(4):665–670, 1973. pages 24, 84

J. Shawe-Taylor and N. Cristianini. *An introduction to support vector machines and other kernel-based learning methods*, volume 204. Volume, 2000. page 20

J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis*. Cambridge university press, 2004. page 20

D. Sheldon. Graphical multi-task learning, 2008. pages 25, 69

N. Shervashidze, P. Schweitzer, E. J. van Leeuwen, K. Mehlhorn, and K. M. Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(77):2539–2561, 2011. URL http://jmlr.org/papers/v12/shervashidze11a.html. pages 108, 173

G. Siglidis, G. Nikolentzos, S. Limnios, C. Giatsidis, K. Skianis, and M. Vazirgiannis. Grakel: A graph kernel library in python. *Journal of Machine Learning Research*, 21(54):1–5, 2020. URL http://jmlr.org/papers/v21/18-370.html. pages 108, 172

S. Smale and D.-X. Zhou. Learning theory estimates via integral operators and their approximations. *Constructive Approximation*, 26:153–172, 08 2007. doi: 10.1007/s00365-006-0659-y. pages 50, 83

C. G. M. Snoek, M. Worring, J. C. van Gemert, J.-M. Geusebroek, and A. W. M. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *Proceedings of the 14th ACM International Conference on Multimedia*, MM '06, page 421–430, New York, NY, USA, 2006. Association for

Computing Machinery. ISBN 1595934472. doi: 10.1145/1180639.1180727. URL https://doi.org/10.1145/1180639.1180727. page 94

E. Spyromitros-Xioufis, G. Tsoumakas, W. Groves, and I. Vlahavas. Multi-target regression via input space expansion: treating targets as inputs. *Machine Learning*, 104(1):55–98, 2016. pages 79, 150

B. K. Sriperumbudur and Z. Szabó. Optimal rates for random fourier features. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1144–1152, Cambridge, MA, USA, 2015. MIT Press. page 37

I. Steinwart and A. Christmann. Sparsity of svms that use the epsilon-insensitive loss. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21 (NeurIPS)*, pages 1569–1576. Curran Associates, Inc., 2008a. page 52

I. Steinwart and A. Christmann. *Support vector machines*. Springer Science & Business Media, 2008b. pages 20, 123

I. Steinwart and A. Christmann. Estimating conditional quantiles with the help of the pinball loss. *Bernoulli*, 17(1):211–225, 2011. page 67

I. Steinwart, D. R. Hush, C. Scovel, et al. Optimal rates for regularized least squares regression. In *COLT*, pages 79–93, 2009. page 88

N. Sterge and B. K. Sriperumbudur. Statistical optimality and computational efficiency of nystrom kernel pca. *Journal of Machine Learning Research*, 23(337):1–32, 2022. URL http://jmlr.org/papers/v23/21-0766.html. page 102

N. Sterge, B. Sriperumbudur, L. Rosasco, and A. Rudi. Gain with no pain: Efficiency of kernel-pca by nyström sampling. In S. Chiappa and R. Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 3642–3652. PMLR, 26–28 Aug 2020. URL https://proceedings.mlr.press/v108/sterge20a.html. page 102

M. Sugiyama and K. Borgwardt. Halting in random walk kernels. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/31b3b31a1c2f8a370206f111127c0dbd-Paper.pdf. page 21

S. Szedmak, J. Shawe-Taylor, and E. Parrado-Hernandez. Learning via linear operators: Maximum margin regression. *Tech. rep., Pascal Research Reports*, 01 2006. page 54

I. Takeuchi and T. Furuhashi. Non-crossing quantile regressions by svm. In *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541)*, volume 1, pages 401–406. IEEE, 2004. page 57

M. Talagrand. Concentration of measure and isoperimetric inequalities in product spaces. *Publications Mathématiques de l'Institut des Hautes Etudes Scientifiques*, 81 (1):73–205, 1995. page 139

T. Tanimoto. *An Elementary Mathematical Theory of Classification and Prediction*. International Business Machines Corporation, 1958. URL https://books.google.fr/books?id=yp34HAAACAAJ. pages 29, 100, 171

T. Tao. *Topics in random matrix theory*, volume 132. American Mathematical Soc., 2012. pages 73, 139

B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems*, volume 16. MIT Press, 2003. URL https://proceedings.neurips.cc/paper_files/paper/2003/file/878d5691c824ee2aaf770f7d36c151d6-Paper.pdf. page 27

B. Taskar, V. Chatalbashev, D. Koller, and C. Guestrin. Learning structured prediction models: a large margin approach. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, page 896–903, New York, NY, USA, 2005. Association for Computing Machinery. ISBN 1595931805. doi: 10.1145/1102351.1102464. URL https://doi.org/10.1145/1102351.1102464. pages 27, 33

P. Tossou, B. Dura, F. Laviolette, M. Marchand, and A. Lacoste. Adaptive deep kernel learning, 2019. page 59

A. Tripp, S. Bacallado, S. Singh, and J. M. Hernández-Lobato. Tanimoto random features for scalable molecular machine learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=MVOINFAKGq. page 171

J. A. Tropp, A. Yurtsever, M. Udell, and V. Cevher. Practical sketching algorithms for low-rank matrix approximation. *SIAM Journal on Matrix Analysis and Applications*, 38(4):1454–1485, Jan 2017. page 40

I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. *Machine Learning*, 07 2004. doi: 10.1145/1015330.1015341. pages 27, 33, 54

I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of machine learning research*, 6(Sep):1453–1484, 2005. pages 27, 33

L. Tu and K. Gimpel. Learning approximate inference networks for structured prediction. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1WgVz-AZ. pages 13, 35, 117

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems (NIPS)*, pages 5998–6008, 2017. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf. pages 60, 108

B. C. Vu. A splitting algorithm for dual monotone inclusions involving cocoercive operators, 2011. page 72

R. Wang, Y. Demiris, and C. Ciliberto. Structured prediction for conditional meta-learning. *Advances in Neural Information Processing Systems*, 33:2587–2598, 2020. page 13

S. Wang and Z. Zhang. Improving cur matrix decomposition and the nyström approximation via adaptive sampling. *J. Mach. Learn. Res.*, 14(1):2729–2769, jan 2013. page 77

B. Weisfeiler and A. Leman. The reduction of a graph to canonical form and the algebra which appears therein. *nti, Series*, 2(9):12–16, 1968. page 173

L. Wenliang, D. J. Sutherland, H. Strathmann, and A. Gretton. Learning deep kernels for exponential family densities. In *International Conference on Machine Learning*, pages 6737–6746. PMLR, 2019. page 59

J. Weston, O. Chapelle, V. Vapnik, A. Elisseeff, and B. Schölkopf. Kernel dependency estimation. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 897–904. MIT Press, 2003. pages 13, 27, 28, 31, 84, 99

P. Willett. Similarity-based virtual screening using 2d fingerprints. *Drug discovery today*, 11(23-24):1046–1053, 2006. page 14

C. Williams and M. Seeger. Using the nyström method to speed up kernel machines. In T. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13, pages 682–688. MIT Press, 2001. pages 38, 66, 77, 86

A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing. Deep kernel learning. In *Artificial intelligence and statistics*, pages 370–378. PMLR, 2016. page 59

D. P. Woodruff. Sketching as a tool for numerical linear algebra. *Found. Trends Theor. Comput. Sci.*, 10(1-2):1–157, 2014. doi: 10.1561/0400000060. URL https://doi.org/10.1561/0400000060. pages 40, 101, 102

T. Yang, Y.-f. Li, M. Mahdavi, R. Jin, and Z.-H. Zhou. Nyström method vs random fourier features: A theoretical and empirical comparison. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. pages 38, 39, 44, 48, 66, 71, 86, 102, 153

Y. Yang, M. Pilanci, M. J. Wainwright, et al. Randomized sketches for kernels: Fast and optimal nonparametric regression. *The Annals of Statistics*, 45(3):991–1023, 2017. pages 16, 44, 49, 65, 66, 67, 68, 70, 71, 73, 77, 78, 86, 87, 92, 102, 110, 123, 124, 125, 139

N. Zhang, S. T. Fountain, H. Bi, and D. T. Rossi. Quantification and rapid metabolite identification in drug discovery using api time-of-flight lc/ms. *Analytical chemistry*, 72(4):800–806, 2000. page 13

Y. Zhang, J. Duchi, and M. Wainwright. Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *Journal of Machine Learning Research*, 16(102):3299–3340, 2015. page 116

W. Zhao, D. Zhou, B. Cao, K. Zhang, and J. Chen. Adversarial Modality Alignment Network for Cross-Modal Molecule Retrieval. *IEEE Transactions on Artificial Intelligence*, 5(1):278–289, 2024. doi: 10.1109/TAI.2023.3254518. page 111

ECOLE DOCTORALE

Titre : Apprentissage de réseaux profonds à noyaux pour la prédiction structurée efficace et robuste

Mots clés : prédiction structurée, méthodes à noyaux, projections aléatoires, réseaux de neurones

Résumé : La prédiction d'objets structurés, tels que les graphes ou les séquences par exemple, est plus exigeante que les problèmes standards de régressions ou de classification supervisés, dans lesquels les sorties sont généralement des vecteurs de petite dimension. Cette tâche fait l'objet de beaucoup d'attention dans différents domaines, comme la biologie ou la chimie informatique. Les espaces structurés sont en général de grande dimension, discrets, et non-linéaires, ce qui complique la conceptualisation d'un modèle polyvalent, autrement dit un modèle capable de gérer différents types de sorties dans un cadre unifié, tout en bénéficiant de solides fondations théoriques.
Dans cette thèse, nous nous concentrons sur les méthodes à noyaux de substitution, et en particulier à la méthode *Input Output Kernel Regression* (*IOKR*), une approche de prédiction structurée polyvalente et théoriquement fondée utilisant l'astuce du noyau sur les espaces d'entrée et de sortie. Toutefois, cette méthode présente plusieurs limites: elle souffre de lourds coûts de calcul pendant les phases d'apprentissage et de prédiction, d'une difficulté à utiliser d'autres fonctions de perte que la quadratique (qui lui permet de bénéficier d'une solution explicite), et l'incapacité des noyaux à apprendre des représentations à partir de données d'entrée complexes comme des images ou du texte. Notre objectif est donc de concevoir un modèle utilisant un noyau de sortie passant à l'échelle de grandes bases de données, avec une borne sur son excès de risque, compatible avec une plus grande variété de fonctions de perte et capable d'apprendre des représentations à partir de données d'entrée complexes.
Dans un premier temps, nous travaillons sur le noyau d'entrée, et introduisons une nouvelle distribution de projections aléatoires sous-gaussienne, les p-*sparsified sketches*, afin de passer à l'échelle les machines à noyau matriciel décomposable utilisant des fonctions de perte lipschitziennes. Ces projections aléatoires sont linéaires et permettent de réduire la complexité calculatoire tout en maintenant de bonnes performances statistiques. De plus, nous fournissons une borne d'excès de risque de l'estimateur induit par cette approche.
Dans un second temps, nous introduisons *Sketched Input Sketched Output Kernel Regression* (*SISOKR*), une méthode basée sur *IOKR* et tirant profit des projections aléatoires sur les noyaux d'entrée et de sortie pour obtenir un estimateur structuré de rang faible. Nous prouvons une borne d'excès de risque de cet estimateur utilisant des projections aléatoires entrée/sortie sous-gaussiennes ou de sous-échantillonnage et montrons qu'il atteint une vitesse d'apprentissage proche de l'optimal. En outre, nous démontrons de solides performances empiriques de *SISOKR* sur des ensembles de données où les calculs requis par *IOKR* excèdent les capacités de la plupart des ordinateurs.
Enfin, nous proposons une architecture neuronale profonde capable de prédire dans l'espace caractéristique potentiellement de dimension infinie du noyau de sortie grâce à l'utilisation de projections aléatoires sur ce dernier. À cette fin, nous calculons la base formée par les fonctions propres de l'opérateur de covariance empirique de sortie projeté aléatoirement, et le réseau de neurones de *Deep Sketched Output Kernel Regression* (*DSOKR*) calcule par la suite une combinaison linéaire au sein de cette base et apprend ses coordonnées pendant l'entraînement. Ceci permet l'utilisation de méthodes d'optimisation à base de gradient pour n'importe quelle fonction de perte consistant en une composition de la perte quadratique et d'une fonction sous-différentiable, comme les fonctions de perte robustes standards par exemple. Ceci est également compatible avec toute sorte d'architecture neuronale, comme les transformeurs, ainsi que le confirment les expériences menées sur un problème de prédiction de molécules dont les données d'entrée sont des descriptions textuelles de ces dernières.

Title : Learning deep kernel networks: application to efficient and robust structured prediction

Keywords : structured prediction, kernel methods, sketching, neural networks

Abstract : The task of predicting structured objects, e.g. graphs or sequences, is more demanding than the standard supervised regression or classification problems, where the outputs are usually low-dimensional vectors. It has recently attracted a lot of attention in various fields, such as computational biology and chemistry. Such structured spaces are usually high-dimensional, discrete, large, and lack of linear structure, which makes it difficult to design a versatile model, i.e. a model able to deal with various output types within a unified framework, together with strong theoretical foundations.
In this thesis, we focus on surrogate kernel methods, and in particular Input Output Kernel Regression, a versatile and theoretically-funded structured prediction approach leveraging the kernel trick in both the input and output spaces. However, in practice, this method exhibits some flaws. As with other kernel-based methods, IOKR suffers from computational burdens at both the training and inference phases. Moreover, it benefits from a closed-form solution when combined with the squared loss, and it is challenging to employ a wider variety of losses. Finally, it is not efficient in handling complex inputs such as images or texts. Our goal is then to design an OKR model that is: scalable to large datasets, theoretically sound (i.e. for which excess risk bounds can be derived), compatible with a wider variety of losses, and able to learn representations from complex inputs.
In the first part of this thesis, we focus on the input kernel, and introduce a new sub-Gaussian sketching distribution, called the p-*sparsified sketches*, in order to scale-up matrix-valued decomposable kernel machines with generic Lipschitz-continuous losses. Sketching consists in manipulating random linear projections to reduce computational complexity while maintaining good statistical performance. We additionally provide an excess risk bound of the estimator induced by this approach.
In the second part, we introduce *Sketched Input Sketched Output Kernel Regression*, an IOKR-based method that leverages sketching on both the input and output kernels to induce a reduced-rank structured estimator. We derive its excess risk bound with sub-Gaussian or sub-sampling input/output sketches and show that it attains close-to-optimal learning rates. Besides, we demonstrate the strong empirical performance of SISOKR on datasets on which IOKR is intractable.
In the last part, we apply sketching on the output kernel and introduce a deep neural architecture able to predict within the possibly infinite-dimensional output kernel's feature space. Indeed, we compute the basis induced by the eigenfunctions of the sketched output empirical covariance operator, and *Deep Sketched Output Kernel Regression*'s neural network then computes an expansion within this basis and learns its coordinates during training. This unlocks the use of gradient-based methods for any loss which is the composition of the square loss with a sub-differentiable function, such as standard robust losses, and any neural architectures, such as transformers. Empirical validations of the approach are provided, in particular on a text-to-molecule dataset.

Institut Polytechnique de Paris
91120 Palaiseau, France