# Event reconstruction and analysis in CMS using artificial intelligence

Polina Simkina

université
PARIS-SACLAY

# Event reconstruction and analysis in CMS using Artificial Intelligence

*Reconstruction et analyse d'événements de l'experience CMS avec Intelligence Artificielle*

École doctorale n°576 Particules, Hadrons, Énergie, Noyau, Instrumentation, Imagerie, Cosmos et Simulation (PHENIICS)
Spécialité de doctorat : Physique des particules
Graduate School : Physique
Référent : Faculté des sciences d'Orsay

Thèse préparée au **Département de Physique des Particules** (Université Paris-Saclay), sous la direction de **Julie MALCLÈS**, Ingénieure-Chercheuse, et le co-encadrement de **Mehmet Özgür ŞAHIN**, Ingénieur-Chercheur.

**Thèse soutenue à Paris-Saclay, le 20 Septembre 2023, par**

**Polina SIMKINA**

## Composition du jury

Membres du jury avec voix délibérative

| | |
|---|---|
| **Susanne GASCON-SHOTKIN**<br>Enseignante-Chercheuse, Université Claude Bernard Lyon 1 | Présidente |
| **Yann COADOU**<br>Chargé de recherche, Centre de Physique des Particules de Marseille (CPPM), Aix Marseille Université | Rapporteur & Examinateur |
| **Jessica LEVÊQUE**<br>Directrice de recherche, Université Savoie Mont Blanc | Rapportrice & Examinatrice |
| **David ROUSSEAU**<br>Directeur de recherche, Université Paris-Saclay | Examinateur |
| **Tommaso TABARELLI DE FATIS**<br>Professeur, Università degli Studi di Milano-Bicocca | Examinateur |

THESE DE DOCTORAT

NNT : 2023UPASP095

université PARIS-SACLAY

ÉCOLE DOCTORALE
Particules, hadrons, énergie et noyau : instrumentation, imagerie, cosmos et simulation (PHENIICS)

**Titre:** Reconstruction et analyse d'événements de l'experience CMS avec intelligence artificielle
**Mots clés:** apprentissage profond, intelligence artificielle, vision par ordinateur, mesures précises de temps, CMS, LHC

**Résumé:** Les récents développements en matière de matériel informatique et d'algorithmes d'apprentissage profond, combinés à de vastes ensembles de données, ont permis d'accomplir des progrès impressionnants dans le domaine de l'intelligence artificielle (IA) au cours des dernières années. Bien qu'ils n'aient été que peu étudiés dans les collisions de particules à haute énergie, les algorithmes d'apprentissage profond ont déjà démontré leur capacité à classer les particules et les événements, à estimer les variables cinématiques et à détecter des anomalies. Ces capacités sont extrêmement utiles pour l'analyse de la quantité sans précédent de collisions proton-proton attendue lors des prochaines phases de fonctionnement du Grand Collisionneur de Hadrons (LHC) au CERN.

Le détecteur CMS fera l'objet d'améliorations majeures pour faire face au nombre croissant de collisions supplémentaires par croisement du LHC ("pileup"), en bénéficiant de détecteurs plus finement segmentés et d'informations temporelles précises. L'un des sujets centraux de cette thèse est consacré au développement d'un logiciel d'acquisition de données polyvalent pour le nouveau détecteur temporel de temps de passage des particules chargées (MIP timing detector ou MTD).

Outre les améliorations matérielles, le succès de ces mises à niveau dépendra fortement de techniques de traitement et d'analyse des événements qui devront être rapides, robustes, performantes et adaptatives. Par conséquent, la majorité des travaux réalisés dans le cadre de cette thèse sont consacrés au développement et au test de nouvelles méthodes de reconstruction basées sur l'IA pour le calorimètre électromagnétique de l'expérience CMS. Ces travaux couvrent deux étapes de la chaîne complète de reconstruction des objets électromagnétiques. La première est l'évaluation des variables cinématiques à partir des signatures énergétiques laissées par des particules uniques atteignant le calorimètre. La seconde combine ces particules en un amas, appelé SuperCluster, qui permet une reconstruction précise de l'énergie des particules électromagnétiques issues de la collision. Les deux tâches sont développées séparément, et pour chacune d'entre elles, un modèle d'IA dédié est créé et ses performances sont évaluées et comparées à l'approche traditionnelle actuelle.

**Title:** Event reconstruction and analysis in CMS using artificial intelligence
**Keywords:** deep learning, artificial intelligence, computer vision, precise timing, CMS, LHC

**Abstract:** Recent developments in computer hardware and deep-learning algorithms, combined with large datasets, lead to impressive progress in artificial intelligence (AI) in the past few years. Although only marginally studied in high-energy particle collisions, deep-learning algorithms already demonstrated the ability to perform particle and event classification, estimation of kinematic variables, and anomaly detection. Those abilities are extremely useful in the analysis of the unprecedented amount of proton-proton collisions expected in the next running phases of the Large Hadron Collider (LHC) at CERN.

The CMS detector will undergo major upgrades to deal with the increasing number of additional collisions per LHC bunch crossing (pileup), benefiting from more finely segmented detectors and precise timing information, and one of the central subjects of this thesis is dedicated to the development of versatile data acquisition software for the new MIP Timing Detector.

In addition to hardware improvements, the success of these upgrades will heavily depend on fast, robust, and adaptive event processing and analysis techniques. Consequently, the majority of the work performed for this thesis is dedicated to developing and testing new AI-based reconstruction methods for the electromagnetic calorimeter of the CMS experiment. It covers two steps of the full chain of electromagnetic object reconstruction. The first one is the evaluation of the kinematic variables from the energy signatures left by standalone particles in the calorimeter. The second one combines these standalone particles into a unified object known as SuperCluster, which is crucial for accurate particle energy reconstruction. Both of the tasks are developed separately, and for each of them, a dedicated AI model is created and its performance is assessed and compared with the current traditional approach.

*Non, rien de rien*
*Non, je ne regrette rien*
*Ni le bien qu'on m'a fait*
*Ni le mal, tout ça m'est bien égal*

*Michel Vaucaire. "Non, je ne regrette rien"*
*Performed by Édith Piaf, 1960.*

# Acknowledgements

I am profoundly grateful for the invaluable support I received from all the remarkable people I encountered throughout my PhD journey. This accomplishment truly would not be possible without your help and guidance.

I want to express my genuine gratitude to the rapporteurs, Yann Coadou and Jessica Levêque, and the members of the jury — Susanne Gascon-Shotkin, David Rousseau, and Tommaso Tabarelli de Fatis. Your thoughtful review of my manuscript and the insightful comments and suggestions were immensely valuable in refining my work. I particularly enjoyed the interesting questions and discussions that we had during the defense.

To my supervisors, my deepest gratitude for your support and guidance throughout these three years. A special thank you to Julie, who graciously took on supervision after my first year. Your always positive attitude, genuine concern for both my work and personal well-being, and kindness have transformed this PhD experience into a truly incredible journey. Ozgur, thank you for being a constant presence throughout the entire three years. Your encouragement, fruitful discussions we had, and extensive help have made this journey much more exciting. And to Fabrice who, even though not officially part of the supervising team, undertook all the same responsibilities. It has been a pleasure working with you, and I will always aspire to reach the same level of competence you effortlessly demonstrate in any subject. Thank you all for helping me grow both professionally and personally!

I want to thank the CMS group at CEA — Aurore, Bruno, Federico, Gautier, Marc, Maksym, Philippe, and Serguei. Working alongside all of you has been an honor. Our lunchtime discussions were a highlight, and I appreciate the practical help and advice you've generously provided.

A big thank you to the direction of IRFU and the IT department for ensuring smooth operations. Special thanks to Nathalie, Georges, and Martine for always providing quick and efficient help with administrative matters. I also want to thank my "godmother", Vanina, for looking out for me and being open to discuss any issues or problems.

To my wonderful friend, Chiara, I have been extremely lucky to meet you along the way. It is rare to find a friend who stands by our side through all the challenging times, let alone someone genuinely cheering for every win and achievement. In you, I have found both and much more. Your friendship has made this journey a whole lot brighter, and I appreciate it more than words can convey.

To my friends from Master's – Aleksei, Andrii, Valeriia, Vidya, and Yuya, thank you for being there at every step of the way, I have been very lucky to meet you. We spent so many great moments together and without you these last three years would be, without doubt, much less memorable and fun.

To my dear friend, Juan, thank you for consistently standing by me through the highs and lows, accepting me without judgment. Your kindness and resilience helped me to see the light even during the darkest days. Your presence has made a significant difference, and I'm fortunate to call you my friend.

A sincere thank you goes to all my friends from CEA — Alexandre, Anastasia,

Chantal, Charles, Emmanuel, and Quentin. I will always remember our (sometimes extended) coffee breaks and the incredible moments we shared outside of the lab.

Special thank you to my office mate and partner in crime, Victor. You always made me laugh and your capacity to remain calm and positive helped me to go through my own challenges with ease.

To Baptiste, Masha, and Misha, your ability to create a safe space for me to share and vent, and your always thoughtful alternative perspectives on the problems I encountered, have been invaluable.

Finally, I want to express my heartfelt gratitude to my family: Anna, Valerii, Nadezhda, Tatyana, Nikolai, Dmitry, and Eleonora. Your love has been my constant source of energy and encouragement.

To my extraordinary sisters, Daria and Evangelina, you unwaveringly believed in me even on the most challenging days. I think of you and miss you every day that we are apart.

# Contents

# Résumé étendu en français

Avec la découverte du boson de Higgs en 2012 par les collaborations CMS et ATLAS, le modèle standard (MS) est complet. Il s'agit de la théorie la plus achevée de la physique des particules, capable à la fois d'expliquer les résultats expérimentaux existants et de fournir des prédictions pour des phénomènes qui n'ont pas encore été observés.

La finalisation du modèle standard nous a donné un outil remarquable pour comprendre le monde subatomique, mais a également ouvert la voie à de nouvelles découvertes. Malgré ses réalisations, de nombreux phénomènes ne s'inscrivent toujours pas dans le contexte du modèle standard. Il s'agit notamment de l'origine de l'asymétrie matière-antimatière, de l'abondance de la matière noire dans l'univers et des oscillations des neutrinos. En outre, la masse du boson de Higgs pose elle-même un problème déconcertant. Selon le modèle standard (MS), pour expliquer la masse mesurée du boson de Higgs, des corrections quantiques importantes doivent s'annuler précisément, ce qui semble hautement improbable. Ces problèmes, ainsi que d'autres problèmes inexpliqués, amènent les physiciens à penser que le MS n'est qu'une réduction d'une théorie plus vaste et plus englobante.

En conséquence, la physique expérimentale moderne des hautes énergies se concentre sur deux aspects : 1) des mesures de précision pour mieux tester la théorie MS, et 2) la recherche directe de physique au-delà du MS qui pourrait donner des indications sur de nouveaux cadres théoriques.

L'un des projets les plus connus pour répondre à ces questions sans réponse est le Grand collisionneur de hadrons (LHC), l'accélérateur le plus grand et le plus puissant du monde. Il a été construit à la frontière entre la France et la Suisse et est exploité par le CERN (Organisation européenne pour la recherche nucléaire). Le LHC est constitué d'un anneau de 27 kilomètres, où les hadrons sont accélérés à des énergies élevées et entrent ensuite en collision à quatre endroits. Chacun des points de collision est associé à une expérience particulière : deux expériences générales, ATLAS (A Toroidal LHC Apparatus) et CMS (Compact Muon Solenoid), et deux expériences dédiées aux ions lourds et à la physique du méson B, ALICE (A Large Ion Collider Experiment) et LHCb (Large Hadron Collider beauty), respectivement.

Les travaux présentés dans ce document sont réalisés à partir des données de l'expérience CMS. Celle-ci est conçue pour capturer et identifier efficacement les particules produites lors des collisions, ainsi que leurs propriétés cinématiques. En 2012, CMS a été l'une des deux expériences qui ont découvert le boson de Higgs. Actuellement, l'expérience vise à résoudre plusieurs problèmes non résolus en physique des particules. Son programme scientifique est donc très vaste et couvre différents domaines, tels que la recherche de nouvelles particules et de nouveaux processus, ainsi que les mesures de précision. La physique du boson de Higgs est l'un des points forts de l'expérience, qui étudie rigoureusement la nature de cette particule récemment découverte.

Afin d'accroître encore le potentiel de découverte, des progrès constants sont réalisés pour améliorer les performances du Grand collisionneur de hadrons (LHC) et des détecteurs, ainsi que pour créer de nouvelles méthodes de reconstruction et d'analyse

des données utilisées dans le cadre de l'expérience CMS. Ces deux efforts progressent en parallèle, s'alimentant et s'influençant mutuellement pour obtenir le système le plus performant et le plus efficace possible.

Le travail présenté dans ce manuscrit est consacré à trois projets différents : "Data Acquisition Software for the MIP Timing Detector", qui est lié au premier effort, "DeepSuperCluster model" et "DeepCluster model", dédiés à la reconstruction du calorimètre, et donc au deuxième effort.

**Logiciel d'acquisition de données pour le détecteur de temps MIP.**

Afin d'améliorer et d'élargir le potentiel de découverte du LHC, celui-ci fait l'objet d'une mise à niveau majeure qui aboutira au LHC à haute luminosité (HL-LHC), dont le démarrage est prévu en 2029. L'objectif principal est d'augmenter la luminosité instantanée d'un facteur 5 par rapport à la valeur de conception initiale ($10^{34}$ cm$^{-2}$s$^{-1}$). Il vise à fournir environ 3000 fb$^{-1}$ de luminosité intégrée sur une période d'exploitation de 10 ans.

Ce collisionneur vise à améliorer la précision des mesures des processus du modèle standard en augmentant la quantité de données enregistrées, ainsi qu'à chercher des processus nouveaux trop rares pour avoir pu être observés jusqu'à présent. Un programme de mise à niveau est en cours pour préparer les sous-systèmes de l'expérience CMS aux nouvelles conditions d'exploitation. Un aspect clé de cette mise à niveau implique l'ajout d'un nouveau détecteur de temps MIP, capable de mesurer avec précision le temps d'arrivée des particules chargées produites lors des collisions. Une partie du travail de thèse est consacrée au développement et au test du logiciel d'acquisition de données spécialement conçu pour ce détecteur.

Le MTD vise à atteindre une résolution de 30 ps au début de l'exploitation du HL-LHC, qui se dégrade lentement jusqu'à 60 ps à la fin de l'exploitation en raison du vieillissement du détecteur. Il se compose de deux parties : La couche tonneau (BTL) - fine couche cylindrique entre le trajectographe et le calorimètre électromagnétique (ECAL) couvrant la plage de pseudorapidité de $|\eta| < 1{,}48$ et une couche bouchon (ETL) - système à deux disques entre le trajectographe et le calorimètre à haute granularité (HGCAL) avec une couverture allant jusqu'à $|\eta| = 3{,}0$.

Le BTL a une surface de 38 m$^2$ et est composé de cristaux scintillants LYSO:Ce en forme de barre, associés à des photomultiplicateurs au silicium (SiPM) aux deux extrémités. Dans le cas de l'ETL, chaque disque a une surface sensible de 7,9 m$^2$. L'ETL étant exposé à des doses de rayonnement nettement plus élevées que le BTL, les détecteurs à avalanche à faible gain (LGAD) ont été choisis comme élément sensible, car ils sont capables de résister à cet environnement difficile.

Pour chacune des couches, un système d'acquisition de données (DAQ) est utilisé pour collecter les signaux des capteurs, reconstruire les informations temporelles et envoyer les données pour construire les événements finaux. Il se compose de plusieurs composants électroniques tolérants aux rayonnements dans la partie frontale et de cartes électroniques basées sur des réseaux de portes programmables (FPGA) dans la partie arrière du détecteur. Le composant clé de l'électronique frontale du BTL est la puce de lecture TOFHIR. Elle doit être capable d'effectuer la numérisation du temps et de l'énergie des MIP avec la précision requise.

Parallèlement aux composants matériels, un logiciel DAQ dédié est en cours de développement pour permettre un traitement efficace des données et la reconstruction des événements. Créé avec le langage de programmation Python, il est entièrement modulaire et fournit un outil facile à utiliser pour les tests du système et les opérations ultérieures.

Le travail effectué a été principalement consacré au développement du logiciel d'acquisition de données. Les principales contributions ont été apportées aux différentes

classes du code DAQ. De plus, l'implémentation du module de reconstruction temporelle, construit sur le code C++ existant, a été entièrement réalisée dans le cadre de cette thèse. Un autre aspect important de ce travail a été la participation active aux tests du système. Au cours de ces tests, le système d'acquisition complet a été validé et une résolution temporelle d'environ 23 ps a été obtenue, ce qui est bien en deçà de la résolution MTD requise. Les données ont été analysées à l'aide du module de reconstruction temporelle, mentionné précédemment.

Le cadre logiciel développé sera utilisé pendant les prochains faisceaux d'essai et continuera à être utilisé pendant la phase de mise en service du détecteur. Il sera encore développé et amélioré, y compris le composant pour la partie ETL, afin d'être utilisé dans les opérations MTD finales pendant le HL-LHC.

**Reconstruction d'objets électromagnétiques.**

Le second projet se concentre sur l'exploration des méthodes d'apprentissage automatique de pointe pour la reconstruction de particules dans le détecteur et l'analyse physique. Ces dernières années, le domaine de l'intelligence artificielle (IA) a connu une croissance sans précédent et, dans le cadre de l'expérience CMS, plusieurs de ces techniques ont déjà été utilisées avec succès pour diverses tâches. Au fur et à mesure que des algorithmes d'IA plus sophistiqués deviennent disponibles, il est naturel d'explorer leur application aux défis existants. Cela peut potentiellement apporter des avantages pendant les opérations en cours du LHC et offrir des perspectives prometteuses pour le futur HL-LHC.

La majeure partie du travail effectué dans le cadre de cette thèse est consacrée au développement et au test de nouvelles méthodes de reconstruction basées sur l'IA pour le calorimètre électromagnétique de l'expérience CMS. Il couvre deux étapes de la chaîne complète de reconstruction d'objets électromagnétiques.

L'objectif principal de la reconstruction du calorimètre électromagnétique est de pouvoir identifier et évaluer correctement les propriétés cinématiques des électrons et des photons. Elles peuvent être récupérées à partir des dépots d'énergie laissés dans le détecteur par ces particules. Le processus commence par la combinaison des dépôts reconstruits (PFRechits) en grappes d'énergie (PFClusters), chacune d'entre elles représentant une ou plusieurs particules qui se chevauchent. Une nouvelle méthode de reconstruction basée sur l'apprentissage automatique, appelée DeepCluster, a été entièrement développée dans le cadre de cette thèse.

Toutefois, avant d'atteindre le calorimètre, un électron ou un photon peut interagir avec le matériau devant l'ECAL, ce qui entraîne la production de plusieurs particules. Dans ce cas, un photon peut se convertir en une paire électron-positron tandis qu'un électron peut émettre des photons de bremsstrahlung. Par conséquent, plusieurs amas PFClusters, provenant d'une particule initiale, apparaîtront dans l'ECAL.

Pour reconstruire correctement l'énergie de l'électron primaire ou du photon, tous les PFClusters produits doivent être combinés en un groupe appelé SuperCluster (SC). Une partie du travail présenté dans ce manuscrit est consacrée au modèle DeepSC, dédié à la reconstruction de ces SuperClusters.

**Modèle DeepSC.**

Dans l'expérience CMS, la reconstruction SuperClustering est réalisée à l'aide d'un algorithme géométrique appelé "Mustache". Bien que cet algorithme ait montré de bonnes performances pour la reconstruction actuelle de l'ECAL, il présente encore un certain nombre de limitations, notamment en ce qui concerne le filtrage des interactions de pile-up et du bruit. Ce problème deviendra plus important au cours du Run 3 et des runs du HL-LHC en raison du vieillissement des détecteurs et de l'augmentation de la luminosité. Pour y remédier, de nouvelles approches d'apprentissage automatique sont à l'étude.

Une nouvelle méthode d'apprentissage profond pour la reconstruction du SuperClustering a été développée. Le réseau DeepSC effectue trois tâches différentes : 1) il crée un SuperCluster optimisé en classant séparément les PFClusters ; 2) il prédit le facteur d'étalonnage de l'énergie qui tient compte des pertes d'énergie dans l'ECAL ; 3) il effectue une identification des particules pour chaque SuperCluster reconstruit, en indiquant si elles proviennent d'un électron, d'un photon ou d'un hadron (jet).

La performance de la classification PFCluster peut être évaluée à partir de la résolution énergétique en comparant l'énergie reconstruite avec DeepSC et l'énergie simulée de la particule. La valeur reconstruite est estimée en additionnant toutes les énergies des PFClusters sélectionnés par le réseau. Les résultats obtenus ont été comparés à l'algorithme "Mustache", actuellement utilisé dans CMS pour la reconstruction du SuperClustering. Le réseau montre des performances supérieures, en particulier pour les régions à faible énergie et à forte accumulation. Cela met en évidence les avantages apparents du modèle DeepSC pour la reconstruction du SuperClustering, en particulier pour la fin du Run 3 et l'ère HL-LHC.

Le modèle DeepSC sera testé plus avant dans le logiciel CMS pour estimer les performances en termes de temps et d'efficacité de calcul. La comparaison entre le modèle DeepSC et l'algorithme "Mustache" avec l'application de corrections énergétiques est également étudiée. Séparément, des efforts supplémentaires seront déployés pour développer la partie du réseau consacrée à la régression énergétique.

Cette thèse s'est concentrée sur la partie du modèle DeepSC relative à l'identification des particules. Cette fonctionnalité a été entièrement développée dans le cadre de la thèse, depuis son ajout dans le modèle initial (en utilisant l'apprentissage par transfert) jusqu'à sa mise en œuvre dans le logiciel CMS global et la comparaison des résultats.

L'identification des particules a permis au modèle de prédire si le superamas reconstruit provenait d'un photon, d'un électron ou d'un hadron à partir des schémas énergétiques de l'ECAL. Le modèle présente d'excellentes performances pour la discrimination jet/photon, et parvient remarquablement à une certaine discrimination électron/photon en utilisant uniquement les informations du calorimètre.

Les performances du modèle en matière d'identification des particules pour la classification jet/photon ont été comparées à celles du réseau neuronal dense utilisé dans le flux de particules. Pour une efficacité de bruit de fond de 10%, l'efficacité du signal est améliorée avec le réseau de plus de 20%. Il s'agit d'un résultat important car l'approche par flux de particules utilise des informations supplémentaires provenant du HCAL et du tracker pour effectuer la discrimination.

Dans les perspectives d'avenir, il y a plusieurs choses qui peuvent être faites pour l'identification des particules. Tout d'abord, il est possible d'étudier la possibilité d'incorporer les données du trajectographe et du HCAL dans le modèle. Deuxièmement, la sortie de l'identification des particules peut être testée comme l'une des variables d'entrée du réseau neuronal dense du flux de particules. Dans ce cas, l'étude de validation du concept a déjà été réalisée en ajoutant la sortie du modèle DeepSC à un classificateur BDT supplémentaire, ce qui a permis d'améliorer de manière significative les résultats de la discrimination jet/photon. Cette amélioration remarquable apportée par l'ajout de la sortie DeepSC ID indique la capacité du modèle à apporter de nouvelles informations à la chaîne de reconstruction. Troisièmement, l'utilisation de la discrimination électron vs. photon peut être étudiée pour l'analyse lorsque les informations sur la trajectoire sont perdues ou mal reconstruites.

Dans l'ensemble, le modèle DeepSC présente des résultats très prometteurs pour une utilisation ultérieure dans le cadre de l'expérience CMS. Il ouvre de nouvelles possibilités pour améliorer la résolution de la reconstruction des objets électromagnétiques et la précision des analyses physiques.

**Modèle DeepCluster.**

Le processus de mesure des objets électromagnétiques (EM) dans l'ECAL implique une série d'étapes sophistiquées, commençant par la reconstruction des dépôts d'énergie dans le calorimètre et aboutissant à la formation de l'objet physique final à partir des informations obtenues.

L'objectif de l'étape de regroupement est de déterminer les propriétés cinématiques des photons et des électrons individuels entrant dans le calorimètre à partir des modèles d'énergie qu'ils laissent dans le détecteur. Le fonctionnement actuel de CMS utilise l'algorithme PFClustering, qui analyse la combinaison de cristaux calorimétriques voisins (appelés cluster) pour évaluer à la fois l'énergie et le point d'entrée de la particule initiale.

Si l'approche traditionnelle s'est avérée efficace et fournit une excellente résolution pour la reconstruction de l'énergie et de la position, sa capacité à distinguer avec précision deux photons proches est limitée, ce qui peut poser plusieurs problèmes. Par exemple, l'algorithme PFClustering peine à différencier les photons isolés ($\gamma$) des pions neutres ($\pi^0$), car ces derniers produisent deux photons qui imitent le schéma énergétique d'un $\gamma$ isolé dans le calorimètre. Un autre exemple est la recherche de la désintégration du boson de Higgs exotique $H \to aa \to 4\gamma$ [1], où $a$ est une particule scalaire ou pseudoscalaire légère. Dans ce cas, les deux photons sont souvent reconstruits comme une seule particule dans le calorimètre.

L'objectif de ce travail est d'aborder la limitation discutée de l'algorithme PFClustering tout en améliorant la performance en termes d'énergie et de résolution de position. Pour ce faire, nous avons développé un nouveau modèle d'apprentissage automatique, appelé DeepCluster, qui exploite des techniques avancées d'apprentissage profond telles que les réseaux neuronaux convolutifs (CNN) et les réseaux neuronaux graphiques (GNN).

Le modèle DeepCluster est une technique de reconstruction innovante conçue spécifiquement pour les particules électromagnétiques dans l'ECAL. Il a été créé et entièrement développé dans le cadre de cette thèse et représente la majorité du travail effectué.

Pour évaluer les méthodes développées et comparer les résultats avec l'approche traditionnelle (PFClustering), une simulation simplifiée de l'ECAL a été créée. Trois implémentations différentes du modèle DeepCluster sont testées. Une approche de réseau en deux étapes basée sur des architectures convolutives et graphiques donne les résultats les plus optimaux et surmonte toutes les difficultés rencontrées.

Le modèle optimisé final est testé sur les ensembles de données à un et deux photons ainsi que sur l'ensemble de données électroniques à particules multiples. Dans tous ces cas, le DeepCluster présente des performances supérieures à celles de la méthode PFClustering en termes de résolution en position et en énergie, ainsi que de rejet du bruit de fond et d'efficacité du signal. Plus particulièrement, l'efficacité du signal pour l'ensemble de données à deux photons obtenue avec le modèle DeepCluster est de 97,0% alors qu'elle n'est que de 82,0% avec la méthode PFClustering.

La principale limite de l'approche traditionnelle est sa difficulté inhérente à distinguer les particules étroitement espacées, ce qui entraîne une dégradation des performances dans l'identification des particules $\pi^0$. En revanche, le modèle DeepCluster donne d'excellents résultats en reconstruisant environ deux fois plus de particules $\pi^0$. Ce résultat ouvre des perspectives prometteuses pour l'application du modèle DeepCluster à la reconstruction dans l'ECAL.

Au cours des étapes de développement suivantes, le modèle DeepCluster sera intégré dans le logiciel global de l'expérience CMS afin d'être testé avec le vrai détecteur ECAL dans des conditions de physique plus réalistes.

# Introduction

With the discovery of the Higgs boson in 2012 by the CMS and ATLAS collaborations, the Standard Model (SM) was completed. Being the most successful Particle Physics theory, it was able to both explain the existing experimental results and provide predictions for the phenomena that were not yet observed.

The finalization of the SM gave us a remarkable tool for understanding the subatomic world but also opened an opportunity for new discoveries. Despite its achievements, there remain numerous phenomena that do not fit within the context of the SM. Among the highlighted subjects are the origin of matter-antimatter asymmetry, the abundance of dark matter in the Universe, and neutrino oscillations. Additionally, the mass of the Higgs boson itself presents a puzzling problem. According to the SM, to account for the measured mass of the Higgs, large quantum corrections have to precisely cancel each other, a task that seems highly improbable. These and other unexplained problems lead physicists to believe that the SM constitutes only a fraction of a larger more encompassing theory. Consequently, the current scientific challenges lie in the field of physics Beyond Standard Model (BSM), where research is motivated by open questions and observable phenomena that can not be explained by SM.

The CMS experiment aims to address several of these unresolved problems. Currently, it has a very broad scientific program covering various areas, such as the search for new particles and processes and precision measurement of the SM. A significant focus of the experiment is the Higgs boson physics, where the nature of this newly found particle is being rigorously studied.

To further increase the potential for discoveries, continuous advancements are being made in both boosting the performance of the Large Hadron Collider (LHC) and the detectors; and the creation of novel data reconstruction and analysis methods employed within the CMS experiment. These two efforts progress in parallel, mutually driving and influencing each other to achieve the most high-performing and effective system possible.

The first effort resulted in the High-Luminosity LHC (HL-LHC) project, scheduled to start in 2029. This improved collider aims to enhance the measurement precision of the standard model processes by increasing the amount of recorded data. An associated upgrade program is being carried out to prepare the sub-systems of the CMS experiment for the new operating conditions. A key aspect of this upgrade involves the addition of a novel MIP Timing Detector, capable of accurately measuring the arrival time of charged particles produced during collisions. A portion of the thesis work is dedicated to the development and testing of the data acquisition software specifically designed for this detector.

The second endeavor focuses on the exploration of state-of-the-art machine-learning methods for detector reconstruction and physics analysis. In recent years, the field of artificial intelligence (AI) has experienced unprecedented growth, and within the CMS experiment, several of these techniques have already been successfully employed for various tasks. As more sophisticated AI algorithms become available, it is natural to explore their application to existing challenges. This can potentially bring benefits during

the ongoing LHC operations and also offer promising prospects for future HL-LHC.

The majority of the work performed for this thesis is dedicated to developing and testing new AI-based reconstruction methods for the electromagnetic calorimeter of the CMS experiment. It covers two steps of the full chain of electromagnetic object reconstruction. The first one is the evaluation of the kinematic variables from the energy signatures left by standalone particles in the detector. The second one combines these standalone particles into a unified object known as SuperCluster, which is crucial for accurate particle energy reconstruction. Both of the tasks are developed separately, and for each of them, a dedicated AI model is created and its performance is assessed and compared with the current traditional approach.

This thesis is structured as follows. The overview of the Standard Model theory is given in Chapter 1. Chapter 2 covers the description of the Large Hadron Collider and the CMS experiment. The data acquisition software for the MIP Timing Detector is detailed in Chapter 3. Chapter 4 discusses the field of Artificial Intelligence and models relevant to the work performed during the thesis. SuperClustering reconstruction with AI is presented in Chapter 5. Chapter 6 covers the standalone particle reconstruction in the ECAL with deep learning. Finally, the conclusion and the outlook are discussed.

# 1 The Standard Model and the Higgs boson

The Standard Model of Particle Physics is an elegant theoretical framework describing the known particles and their fundamental interactions (with the exception of gravity) [2]. It was developed through several stages in the second half of the $20^{\text{th}}$ century and the final constituent predicted by the SM – the Higgs boson — was discovered in 2012 [3], [4].

Even though the SM has been successfully confirmed with multiple experiments (such as the discovery of neutral currents in 1973, the discovery of the $W$ and $Z$ bosons in 1983, etc.), there is a number of remaining puzzles that still can not be explained in the context of the SM [5]. Among the most prominent ones are:

- **Matter-antimatter asymmetry**. During the Big Bang, equal amounts of matter and anti-matter should have been produced. However, currently, in the observable Universe, matter largely prevails, and the reason for it is yet to be uncovered.
- **Dark energy**. From observational evidence (e.g. [6], [7], [8]), it is known that the Universe expands with acceleration rather than at a constant rate. This effect can only be theoretically explained by introducing a new form of energy, called "dark energy". Even though it represents approximately 68% of the mass-energy content of the observable Universe, its nature still remains a mystery.
- **Dark matter**. Various astrophysical observations (e.g. gravitational lensing [9]) also indicate the presence of an alternative type of matter, called "dark matter", in the Universe. Unlike usual baryonic matter, it does not interact with light through electromagnetic forces, and, thus, is hard to detect. According to the current standard model of cosmology "lambda-CDM" [10], dark matter should account for around 25% of all the energy-matter content in the observable Universe. However, the SM does not contain any candidates for dark matter particles.
- **The hierarchy problem**. The discrepancy between the parameters of the weak force and the gravity can not be explained by the SM as well. It is still unclear why the mass of the Higgs boson ($\sim 125$ GeV) is so much smaller than the Planck mass ($\sim 10^{19}$ GeV). The measured mass of the Higgs can be explained within the SM only if very precise fine-tuning cancellation between the quantum contributions appears, which is theoretically highly unlikely.

Accordingly, modern experimental high-energy physics focuses on two aspects: 1) precision measurements to further test the SM theory, and 2) searching directly for physics beyond the SM that may indicate new theoretical frameworks.

In this Chapter, the SM and the Higgs boson are described. In Section 1.1 a general overview of the SM is given. Section 1.2 examines theoretical aspects of the SM, including

the Spontaneous Symmetry Breaking mechanism. Finally, in Section 1.3 studies of the Higgs boson at the CMS experiment are presented, including a brief overview of two analyses particularly relevant to the work presented further in this document.

## 1.1 Overview of the Standard Model

The Standard Model describes all known elementary particles and three fundamental forces (electromagnetic, weak, and strong) [11]. The elementary particles are the ones that, to our knowledge, do not have an internal structure. The overview of the particles constituting the Standard Model and their properties (mass, electrical charge, spin) is given in Fig. 1.1. In general, they can be divided into two groups: fermions and bosons.

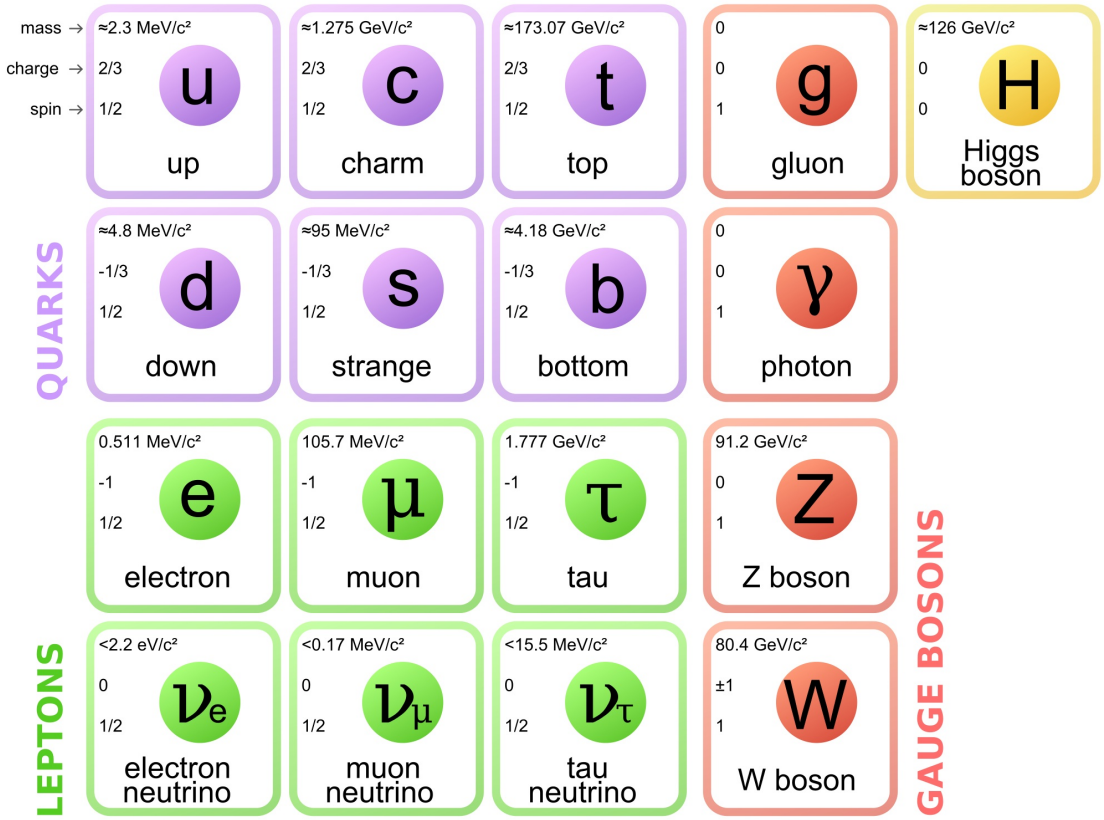



Figure 1.1: Diagram representing particle content of the Standard Model [12].

### Fermions

Fermions represent the building blocks of matter. They have a spin of $1/2$ and can be further classified into *leptons* (electron $e$, muon $\mu$, tau-lepton $\tau$, electron neutrino $\nu_e$, muon neutrino $\nu_\mu$, tau neutrino $\nu_\tau$ and their anti-particles) and *quarks* (up $u$, down $d$, strange $s$, charm $c$, top $t$, bottom $b$ and their anti-particles). Fermions follow the Pauli exclusion principle [13]: two or more identical particles with half-integer spins can not occupy the same quantum state within a quantum system simultaneously.

The quarks along with electrons, muons, and tau-leptons possess electromagnetic charge and, thus, can interact through electromagnetic interactions. All fermions can participate in weak interactions as well.

Additionally, quarks possess another quantum number called *color*, on which the strong interaction operates. It is denoted as $q_i$ and can be of three types: $i = 1, 2, 3$. The

elementary quarks are confined in composite objects called *hadrons*. They are further classified as *baryons*, usually consisting of three quarks (e.g. proton $\sim uud$), and *mesons*, usually made of a pair quark-antiquark (e.g. neutral pion $\pi^0 \sim u\bar{u}$ or $\sim d\bar{d}$). The hadrons are colorless objects: for baryons, it is achieved with a combination of three different colors, and for mesons with quark carrying color and anti-quark – anti-color.

#### Gauge Bosons

All the fundamental interactions within the SM are mediated by exchanging elementary particles known as *gauge bosons*. They have a spin = 1 and are the following:

1. Photon $\gamma$ for electromagnetic interactions. It is a massless particle with zero electrical charge.

2. Three vector bosons $W^{\pm}$, $Z$ are the corresponding intermediate bosons of the weak interactions. They are massive particles, $W^{\pm}$ bosons can carry a negative or positive electrical charge and $Z$ boson is electrically neutral.

3. Eight gluons $g_{\alpha}$ mediate the strong interaction among quarks. They are massless, electrically neutral, and carry color quantum numbers.

#### Fundamental interactions

The range and strength of three fundamental interactions are as follows:

1. Electromagnetic interaction has an infinite range as it is mediated by a massless boson. The strength is governed by the fine structure constant $\alpha = \frac{1}{137}$.

2. Weak interaction has a short range of $10^{-16}$ cm due to the massive intermediate bosons. The strength is given by Fermi constant $G_F = 1.167 \text{ x } 10^{-5} \text{ GeV}^{-2}$.

3. Strong interaction does not have an infinite range even though the associated bosons are massless. Due to the color confinement, it is $10^{-13}$ cm. The strength is determined by the size of the strong coupling constant $\alpha_s$. The value of the constant varies with energy from asymptotically large values $\sim$1 at $E \sim 1$ GeV to the vanishing asymptotic limit $\alpha_s \to 0$ at $E \gtrsim 1000$ GeV. For the energy scale of the $Z$ boson mass $\alpha_s(M_Z^2) = 0.1181$.

Above the unification energy limit ($\sim$ 100 GeV), electromagnetic and weak interactions are merged into a single electroweak force.

#### Higgs boson

In addition to the gauge bosons, there is another fundamental particle known as the Higgs boson: it possesses a spin of 0 and does not carry any electric or color charge. Unlike the gauge bosons, the Higgs boson does not mediate any fundamental force.

The Higgs boson is associated with a quantum field (*the Higgs field*), which permeates all of space according to the SM. Fermions and $W^{\pm}$, $Z$ bosons acquire their masses through this field. This process is further described in detail in Section 1.2.4.

## 1.2 Theoretical formulation

Mathematically, the Standard Model can be formulated as a quantum field theory that is based on the gauge symmetry $SU(3)_C$ x $SU(2)_L$ x $U(1)_Y$ [14]. In this group, $SU(3)_C$ represents the symmetry group of the strong interaction, and $SU(2)_L$ x $U(1)_Y$ is the

symmetry group of the unified electroweak interaction. After electroweak symmetry breaking, the group is broken into $U(1)_{EM}$, the group of electromagnetism, residual symmetry.

The gauge theory based on the $U(1)_{EM}$ is called *Quantum Electrodynamics (QED)*, the gauge theory based on $SU(3)_C$ is *Quantum Chromodynamics (QCD)*, and the *Electroweak Theory* corresponds to $SU(2)_L$ x $U(1)_Y$.

**Standard Model Lagrangian**

The dynamics and the interactions of the elementary particles can be presented in the form of the SM Lagrangian. In a compact version, it can be written as follows

$$\mathcal{L}_{\text{SM}} = -\frac{1}{4}F^{\mu\nu}F_{\mu\nu} \tag{1.1}$$
$$+ i\bar{\psi}\not{D}\psi + \text{h.c} \tag{1.2}$$
$$+ \psi_i y_i j \psi_j \phi + \text{h.c.} \tag{1.3}$$
$$+ |D^\mu \phi|^2 - V(\phi), \tag{1.4}$$

where the 1st line represents the kinetic and self-interacting terms of the gauge bosons, the 2nd line contains kinetic and interaction terms of fermions, the 3rd line describes the interaction between the matter fields and the Higgs field, the 4th line relates to the dynamic of the Higgs sector, and h.c. stands for Hermitian conjugates. In this Section, all of the parts of the SM Lagrangian are deconstructed and discussed in detail.

**The Gauge Principle**

The gauge symmetry is a local symmetry, meaning that continuous parameters of the transformation depend on the space-time coordinates [15]. This is an important aspect for the SM formulation, as promoting a global symmetry to a local one transforms a free theory (particles or fields do not interact with each other) into an interacting one. In order to keep the symmetry under a local transformation, new vector boson fields, also known as gauge fields, must be introduced.

The application of the described Gauge Principle will be further shown in the context of building QED, QCD, and electroweak theories.

**Dirac equation**

For the mathematical formulation of the SM, it is also crucial to introduce the Dirac equation, which describes free massive particles with spin 1/2 [15]. The Lagrangian for such a particle using the Dirac spinors $\psi$ (a 4-component column vector representing the particle's quantum state as a function of space-time coordinates) can be written as

$$\mathcal{L} = \bar{\psi}(x)(i\not{\partial} - m)\psi(x), \qquad \not{\partial} \equiv \partial_\mu \gamma^\mu, \tag{1.5}$$

where $m$ is the mass of a particle and $\gamma^\mu$ are the Dirac gamma matrices.

The corresponding equation of motion, also known as the Dirac equation, is

$$(i\not{\partial} - m)\psi(x) = 0 \tag{1.6}$$

### 1.2.1 Quantum Electrodynamics

QED is a quantum field theory that describes the interactions of charged particles with the electromagnetic field. It is a gauge theory that can be built starting from the

Lagrangian, described in Eq. (1.5), where $\psi$ represents the field of charged fermions.

This Lagrangian is invariant under the global $U(1)$ transformation:

$$\psi \to e^{iQ\theta}\psi, \qquad \bar{\psi} \to \bar{\psi}e^{-iQ\theta}, \qquad \partial_\mu\psi \to e^{iQ\theta}\partial_\mu\psi \tag{1.7}$$

where $\theta$ is a continuous parameter and $Q$ is a charge of the particle.

In order to include the QED interactions, first, the transformation must be promoted to local:

$$\psi \to e^{iQ\theta(x)}\psi, \qquad \bar{\psi} \to \bar{\psi}e^{-iQ\theta(x)} \qquad \partial_\mu\psi \to e^{iQ\theta(x)}\partial_\mu\psi + iQ\left(\partial_\mu\theta(x)\right)e^{iQ\theta(x)}\psi \tag{1.8}$$

Compared to Eq. (1.7), an extra term appears in the gauge transformation due to $\partial_\mu\theta(x) \neq 0$. As a consequence, the considered Lagrangian is not invariant anymore.

To restore the invariance, a gauge vector boson field or a photon field $A_\mu(x)$ is introduced. It interacts with the field $\psi$ and transforms under the $U(1)$ gauge transformations as follows:

$$A_\mu \to A_\mu - \frac{1}{e}\partial_\mu\theta(x) \tag{1.9}$$

Using this new field, the gauge-invariant Lagrangian can be built by replacing the normal derivative $\partial_\mu$ with a special so-called covariant derivative $D_\mu$, defined as

$$D_\mu\psi \equiv \left(\partial_\mu - ieQA_\mu\right)\psi \tag{1.10}$$

It can be shown that under the local $U(1)$ transformation, it changes in the same way as the field:

$$D_\mu\psi \to e^{iQ\theta(x)}D_\mu\psi \tag{1.11}$$

Finally, in order to account for the photon field propagation, a kinetic term should be added. It also must be gauge invariant and it is defined as

$$F_{\mu\nu} = \partial_\mu A_\nu - \partial_\nu A_\mu \tag{1.12}$$

The total Lagrangian of QED is

$$\mathcal{L}_{QED} = \psi(x)(i\not{D} - m)\psi(x) - \frac{1}{4}F_{\mu\nu}(x)F^{\mu\nu}(x) \tag{1.13}$$

The interaction, representing the force mediation through photon, enters the Lagrangian as the following term within $\bar{\psi}i\not{D}\psi$:

$$\bar{\psi}eQA_\mu\gamma^\mu\psi \tag{1.14}$$

### 1.2.2 Quantum Chromodynamics

QCD is a gauge theory for strong interactions [16]. In this case, the gauge symmetry is the local color transformations.

Quarks, denoted as $q_i$ with three different colors $i = 1, 2, 3$, form the fundamental representation of the corresponding $SU(3)_c$ group. Gluons, denoted ad $g_\alpha$ with $\alpha = 1, \ldots, 8$, are the gauge boson particles. There are 8 of them, corresponding to the number of generators of $SU(3)$.

QCD can be built similarly to QED, starting from Lagrangian in Eq. (1.5), which represents free quarks in this case. It must be slightly re-formed to include all possible colors and, for a single flavour, it can be written as

$$\mathcal{L} = \sum_{i=1}^{3} \overline{q_i} \left( i\not{\partial} - m_q \right) q_i, \tag{1.15}$$

The global $SU(3)$ transformation is

$$q_i \longrightarrow q_i' = U_{ij} q_j, \qquad UU^\dagger = U^\dagger U = \mathbb{I}, \tag{1.16}$$

where the three-dimensional unitary matrix $U$ with $det\ U = 1$ can be represented as:

$$U\left(\varepsilon_a\right) = e^{-i\sum_{a=1}^{8} \varepsilon_a \frac{\lambda_a}{2}}, \tag{1.17}$$

where $\varepsilon_a$ are the parameters of the transformation and $\lambda_a$ are so-called Gell-Mann matrices that generate $SU(3)$ rotations.

The theory is further promoted to a local one by demanding $\varepsilon_a = \varepsilon_a(x)$. Following the gauge principle, to achieve the invariance of the Lagrangian, eight vector gluon fields $A_\mu^\alpha(x)$ must be introduced. The covariant derivative in this case is

$$D_\mu q \equiv \left( \partial_\mu - i g_s \left( \frac{\lambda_\alpha}{2} \right) A_\mu^\alpha \right) q, \tag{1.18}$$

where

$$q = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} \tag{1.19}$$

The final QCD Lagrangian, also containing the kinetic term for the gluon fields, is

$$\mathcal{L}_{QCD} = \sum_q \bar{q}(x) \left( i\not{D} - m_q \right) q(x) - \frac{1}{4} F_{\mu\nu}^\alpha(x) F_\alpha^{\mu\nu}(x) \tag{1.20}$$

The gauge interactions among the quarks and gluons are contained in the $\bar{q} i\not{D} q$ term:

$$\bar{q} g_s \frac{\lambda^\alpha}{2} A_\mu^\alpha \gamma^\mu q \tag{1.21}$$

An important difference between QED and QCD is that the latter also allows the self-interaction between its gauge bosons, the gluons.

### 1.2.3 Electroweak Theory

The electroweak theory is based on the gauge symmetry of the $SU(2)_L$ x $U(1)_Y$ group, which unifies weak and electromagnetic interactions [17]. $SU(2)_L$ is the weak isospin group that acts only on the left-handed component of the fermion field and $U(1)_Y$ is the weak hypercharge group.

The handedness (left-handed or right-handed) of a particle is defined by means of the chirality operator $\gamma_5 = i\gamma_0\gamma_1\gamma_2\gamma_3$:

$$\psi_L = \frac{1}{2}\left(1 - \gamma_5\right)\psi, \qquad \psi_R = \frac{1}{2}\left(1 + \gamma_5\right)\psi \tag{1.22}$$

The $SU(2)_L$ transformation is defined as

$$U_L \equiv e^{i\frac{\sigma_i}{2}\alpha^i} \tag{1.23}$$

The generator of $SU(2)_L$ group is the *weak isospin* $T_i = \frac{\sigma_i}{2}$, where $\sigma_i$ are the Pauli

matrices. The corresponding gauge bosons are denoted as $W^i_\mu$, $i = 1, 2, 3$.

Usually, only the third component of $T_3$ is considered. Left-handed fermions have $T_3 = \pm\frac{1}{2}$ (by convention, the sign is the same as the electric charge), and right-handed fermions have $T_3 = 0$. In the case of the anti-fermions, the chirality and the sign of $T_3$ are reversed.

Under $SU(2)_L$ transformation, the right-handed fermions transform as singlets and the left-handed fermions transform as doublets:

$$\psi_R \to \psi_R, \quad \psi_L \to U_L \psi_L \tag{1.24}$$

In this notation, the electron-neutrino pair, for instance, can be introduced as

$$\psi_L(x) = \begin{pmatrix} \nu_e \\ e^- \end{pmatrix}_L, \quad \psi_R(x) = \nu_{eR}, \quad \psi'_R(x) = e^-_R \tag{1.25}$$

The *weak hypercharge* $Y$ is the quantum number that relates the electrical charge $Q$ and the third component of weak isospin $T_3$ through the formula:

$$Q = T_3 + \frac{1}{2}Y \tag{1.26}$$

The corresponding gauge boson of $U(1)_Y$ is $B_\mu$.

Building the electroweak theory at first follows the same steps as for QED and QCD. The global symmetry is promoted to the local one, and the invariant Lagrangian is achieved by using the covariant derivative:

$$D_\mu = \partial_\mu - ig\vec{T}\vec{W}_\mu - ig'\frac{Y}{2}B_\mu \tag{1.27}$$

where $g$ is a constant, called weak isospin coupling, and $g'$ is a constant, called weak hypercharge coupling.

Applied to the right- and left-handed fields, it takes the form:

$$D_\mu \psi_L = \left(\partial_\mu - ig\frac{\vec{\sigma}}{2} \cdot \vec{W}_\mu + ig'\frac{1}{2}B_\mu\right)\psi_L; \quad D_\mu \psi_R = \left(\partial_\mu + ig'B_\mu\right)\psi_R \tag{1.28}$$

Accordingly, the interaction term of the resulting Lagrangian can be written as

$$\mathcal{L}_f = \bar{\psi_L}(x) i\not{D} \psi_L(x) + \bar{\psi_R}(x) i\not{D} \psi_R(x) + \bar{\psi'_R}(x) i\not{D} \psi'_R(x) \tag{1.29}$$

It can be further divided into two parts, representing charged- and neutral-current interactions:

- **Charged-current interaction**. The charged-current interactions entering the Eq. (1.29) can be described using the physical gauge bosons $W^\pm$, which is defined through gauge fields as

$$W^\pm_\mu = \frac{1}{\sqrt{2}}\left(W^1_\mu \mp iW^2_\mu\right) \tag{1.30}$$

  The associated charge-current Lagrangian term is

$$\mathcal{L}_{CC} = \frac{g}{2\sqrt{2}}\left(W^+_\mu \overline{\psi_L}\gamma^\mu \sigma^+ \psi_L + W^-_\mu \overline{\psi_L}\gamma^\mu \sigma^- \psi_L\right), \tag{1.31}$$

where the new Pauli matrices are defined as

$$\sigma^{\pm} = \frac{1}{\sqrt{2}}\left(\sigma^1 \pm i\sigma^2\right) \tag{1.32}$$

- **Neutral-current interaction**. Equation (1.29) also contains interaction with the neutral gauge fields $W_\mu^3$ and $B_\mu$.

  The physical $Z_\mu$ and $A_\mu$ fields, corresponding to the $Z$ boson and the photon, can be obtained from the neutral gauge fields by applying a rotation by the weak mixing angle $\theta_W$:

$$\begin{pmatrix} A_\mu \\ Z_\mu \end{pmatrix} \equiv \begin{pmatrix} \cos\theta_W & \sin\theta_W \\ -\sin\theta_W & \cos\theta_W \end{pmatrix} \begin{pmatrix} B_\mu \\ W_\mu^3 \end{pmatrix} \tag{1.33}$$

  The associated neutral current terms are

$$\begin{aligned} \mathcal{L}_{NC}^{Z} &= \overline{\psi_L}\gamma^\mu Z_\mu \left(g\frac{\sigma_3}{2}\cos\theta_w - g'\frac{Y}{2}\sin\theta_w\right)\psi_L \\ \mathcal{L}_{NC}^{\gamma} &= \overline{\psi_L}\gamma^\mu A_\mu \left(g\frac{\sigma_3}{2}\sin\theta_w + g'\frac{Y}{2}\cos\theta_w\right)\psi_L \end{aligned} \tag{1.34}$$

Finally, the kinetic term for the gauge fields can be added as

$$\mathcal{L}_g = -\frac{1}{4}W_a^{\mu\nu}W_{\mu\nu}^a - \frac{1}{4}B^{\mu\nu}B_{\mu\nu}, \tag{1.35}$$

where the strength fields are defined as

$$\begin{aligned} W_{\mu\nu}^i &= \partial_\mu W_\nu^i - \partial_\nu W_\mu^i + g\epsilon^{ijk}W_\mu^j W_\nu^k \\ B_{\mu\nu} &= \partial_\mu B_\nu - \partial_\nu B_\mu, \end{aligned} \tag{1.36}$$

where $\epsilon^{ijk}$ is the Levi-Civita tensor. This term also incorporates the self-interaction among the gauge fields.

In this formulation, the mass term for the gauge bosons is forbidden as it breaks the invariance of the Lagrangian. Fermionic masses are also not possible, because they would communicate the right- and left-handed fields, which have different transformation properties.

However, from the experimental evidence, it is known that the physical $W^\pm$ and $Z$ bosons, as well as fermions (such as electrons), are massive. In theory, they obtain masses through the spontaneous symmetry-breaking mechanism, which is explained in the following section.

### 1.2.4 Spontaneous Symmetry Breaking and the Higgs Mechanism

In order to include the required masses in the electroweak theory without spoiling the gauge invariance, a Spontaneous Symmetry Breaking (SSB) is introduced.

The symmetry is said to be spontaneously broken if the Lagrangian describing the system is invariant under these symmetry transformations, but the vacuum solution is not.

### Goldstone bosons

In the context of the SM, the following complex scalar field with its Lagrangian is considered:

$$\mathcal{L} = \partial_\mu \phi^\dagger \partial^\mu \phi - V(\phi), \quad V(\phi) = \mu^2 \phi^\dagger \phi + h\left(\phi^\dagger \phi\right)^2 \tag{1.37}$$

The Lagrangian is invariant under the global transformation

$$\phi(x) \longrightarrow \phi'(x) \equiv e^{i\theta}\phi(x) \tag{1.38}$$

In the ground state (or vacuum) the potential can have two possibilities depending on the sign of $\mu^2$:

1. $\mu^2 > 0$: The potential has only one trivial minimum $\phi = 0$. It is a usual situation with a single ground state. In this case, the potential describes a single scalar particle with mass $\mu$ and coupling $h$.

2. $\mu^2 < 0$: The minimum of the potential is obtained when the following conditions are satisfied

$$|\phi_0| = \sqrt{\frac{-\mu^2}{2h}} \equiv \frac{v}{\sqrt{2}} > 0, \quad V(\phi_0) = -\frac{h}{4}v^4 \tag{1.39}$$

The corresponding potential is shown in Fig. 1.2. The newly introduced parameter $v$ is also known as the vacuum expectation value (VEV). This scenario is more interesting as it corresponds to the infinite number of states of minimum energy $\phi_0(x) = \frac{v}{\sqrt{2}}e^{i\theta}$, represented by the rim in Fig. 1.2. By choosing a particular ground state, for example, $\theta = 0$, the symmetry gets spontaneously broken.

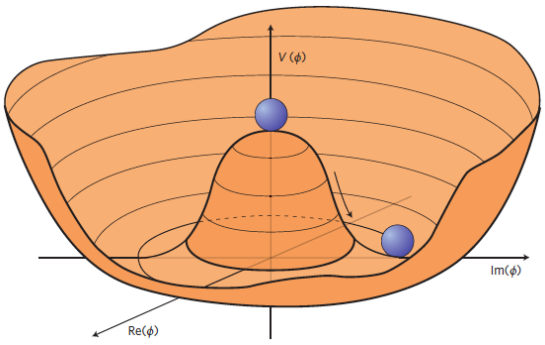



Figure 1.2: A "Mexican hat" potential that leads to spontaneous symmetry breaking [18].

The complex scalar potential can be further parameterized as

$$\phi(x) \equiv \frac{1}{\sqrt{2}}\left[v + \varphi_1(x) + i\varphi_2(x)\right], \tag{1.40}$$

where $\varphi_1(x)$ and $\varphi_2(x)$ are real fields. The potential in this case:

$$V(\phi) = V(\phi_0) - \mu^2\varphi_1^2 + hv\varphi_1\left(\varphi_1^2 + \varphi_2^2\right) + \frac{h}{4}\left(\varphi_1^2 + \varphi_2^2\right)^2 \tag{1.41}$$

Thus, $\varphi_1$ describes a state with mass $m^2 = -2\mu^2$ and $\varphi_2$ is massless. The latter is the result of a Goldstone theorem: the spontaneous breaking of continuous global symmetry is always accompanied by the appearance of one or more massless scalar particles, known as Goldstone bosons.

### Higgs mechanism

The Higgs mechanism appears in the scenario of spontaneous symmetry-breaking in the case of gauge invariance.

In order to maintain the invariance under $SU(2)_L$ x $U(1)_Y$ symmetry group, the Lagrangian from Eq. (1.37) can be re-written as

$$\mathcal{L}_S = (D_\mu \phi)^\dagger D^\mu \phi - \mu^2 \phi^\dagger \phi - h\left(\phi^\dagger \phi\right)^2 \quad \left(h > 0, \mu^2 < 0\right), \tag{1.42}$$

where

$$D_\mu \phi = \left[\partial_\mu + ig\vec{T}\vec{W}_\mu + \frac{1}{2} ig' B_\mu\right] \phi \tag{1.43}$$

The $SU(2)_L$ doublet of complex scalar fields is presented as:

$$\phi(x) \equiv \begin{pmatrix} \phi^{(+)}(x) \\ \phi^{(0)}(x) \end{pmatrix} \tag{1.44}$$

Similarly to the global symmetry case, an infinite number of ground states can be found and by choosing one of them, a gauge symmetry gets spontaneously broken. The ground state can be chosen to be:

$$\phi_0^{(+)} = 0 \quad \phi_0^{(0)} = \frac{v}{\sqrt{2}} \tag{1.45}$$

Taking this into account, the excitations around the ground state can be parameterized as

$$\phi(x) = e^{i\frac{\sigma_i}{2}\theta^i(x)} \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ v + H(x) \end{pmatrix}, \tag{1.46}$$

where $\theta_i$ are three real fields, representing the massless Goldstone bosons, and $H(x)$ is the Higgs field.

The field is invariant under the local transformation $\phi' = e^{i\alpha(x)}\phi$. The parameter $\alpha(x)$ can be chosen to be $\alpha(x) = e^{-i\frac{\sigma_i}{2}\theta_i}$. In this case, the massless Goldstone bosons are eliminated and the scalar doublet becomes

$$\phi(x) = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ v + H(x) \end{pmatrix}, \tag{1.47}$$

Using Eq. (1.47), the covariant derivative can be written as

$$(D_\mu \phi)^\dagger D^\mu \phi \quad \overset{\theta^i = 0}{\longrightarrow} \quad \frac{1}{2}\partial_\mu H \partial^\mu H + (v + H)^2 \left\{ \frac{g^2}{4} W_\mu^\dagger W^\mu + \frac{1}{2} \frac{g^2 + g'^{\,2}}{4} Z_\mu Z^\mu \right\} \tag{1.48}$$

As a result, the vacuum expectation value of the neutral scalar has generated quadratic terms for the physical $W^\pm$ and $Z$ bosons, or, in other words, these bosons have acquired masses:

$$M_Z = \frac{\sqrt{g^2 + g'^2}}{2} v, \quad M_W = \frac{gv}{2} = M_Z \cos\theta_W \tag{1.49}$$

Moreover, by inserting Eq. (1.47) into the Lagrangian from Eq. (1.42), it can be shown that for the Higgs boson, the particle associated with the Higgs field, the mass is

$$M_H = \sqrt{-2\mu^2} = \sqrt{2h} v \tag{1.50}$$

Concerning the mass of the fermions, their mass term is defined as

$$-m\left(\bar{\psi}_L\psi_R + \bar{\psi}_R\psi_L\right) \tag{1.51}$$

It could not be previously introduced as it would break the gauge invariance. However, the mass can be generated by the Higgs field. The following gauge-invariant Yukawa Lagrangian describes the interactions between the Higgs and the fermions:

$$\mathcal{L}_{YW} = -\lambda_e \frac{v+H}{\sqrt{2}}\left(\bar{e}_R e_L + \bar{e}_L e_R\right), \tag{1.52}$$

where $\lambda_e$ is the coupling constant between an electron and a Higgs boson.
Accordingly, the mass of the electron is

$$m_e = \lambda_e \frac{v}{\sqrt{2}} \tag{1.53}$$

In a similar way, the masses of other fermions can be obtained.

Finally, the interaction terms of the Higgs sector also arise from Eqs. (1.42) and (1.52), including the self-interaction terms of the Higgs boson. The interactions of Higgs with fermions and gauge bosons are proportional to the gauge couplings and to the corresponding particle masses:

$$\begin{aligned}\mathcal{L}_H^{\text{int}} = &-\frac{M_H^2}{2v}H^3 - \frac{M_H^2}{8v^2}H^4 \\ &-\frac{m_\psi}{v}\bar{\psi}H\psi \\ &+M_W^2 W_\mu^+ W^{\mu-}\left(1+\frac{2}{v}H+\frac{1}{v^2}H^2\right) \\ &+\frac{1}{2}M_Z^2 Z_\mu Z^\mu\left(1+\frac{2}{v}H+\frac{1}{v^2}H^2\right)\end{aligned} \tag{1.54}$$

The final Lagrangian of the electroweak theory can be written as the sum of Eqs. (1.29), (1.35), (1.42), (1.52):

$$\mathcal{L}_{EW} = \mathcal{L}_f + \mathcal{L}_g + \mathcal{L}_S + \mathcal{L}_{YW} \tag{1.55}$$

## 1.3 The Higgs boson at the CMS experiment

The investigation of the SM and the search for the physics beyond it are performed in the CMS experiment that is situated at the Large Hadron Collider (LHC). This section focuses on describing the Higgs boson at CMS, which is a key point of interest in the experiment. The following Chapter 2 gives a more detailed description both of the LHC and the CMS experiment.

### 1.3.1 Higgs production and decay channels

#### Production channels

The Higgs boson is produced through different channels in proton-proton collisions at the LHC [19]. The Higgs production cross sections for $m_H = 125$ GeV as a function of the center-of-mass energy ($\sqrt{s}$) are shown in Fig. 1.3 (left).

The leading order Feynman diagrams contributing to the Higgs production are shown in Fig. 1.4 and they are the following:

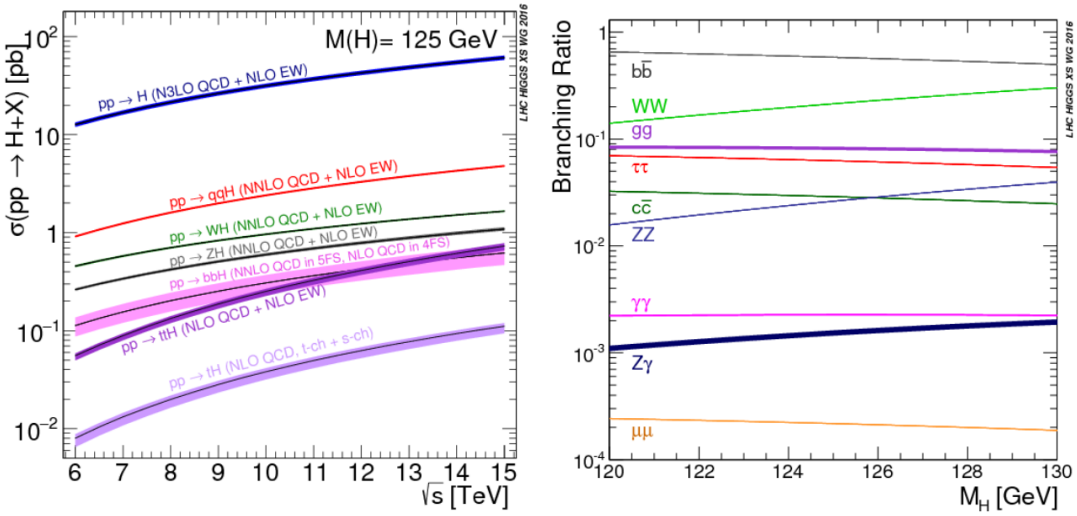



Figure 1.3: On the left: the SM Higgs boson production cross sections as a function of the center of mass energy, $\sqrt{s}$, for $pp$ collisions for the Higgs boson mass of $m_H = 125$ GeV. On the right: The branching ratios for the main decays of the SM Higgs boson near $m_H = 125$ GeV [19].

1. The dominant one is the gluon fusion ($ggH$), presented in Fig. 1.4(a). It amounts to 87% of the total production at $\sqrt{s} = 13$ TeV.

2. The second largest contribution comes from the Vector Boson Fusion (VBF), shown in Fig. 1.4(b). It accounts for approximately 7% of the total production.

3. The third one is the associated production of a Higgs boson with a Vector boson (VH) in Fig. 1.4(c). It represents 4% of the total production at $\sqrt{s} = 13$ TeV.

4. The fourth cross-section is the production of a Higgs boson with associated pair of $t$-quarks ($t\bar{t}H$). It accounts for 1% of the total production.

### Decay channels

The Higgs boson has a short lifetime of about 1.6 x $10^{-22}$ s. It is not directly detected at the CMS experiment but rather through its decay products. As it interacts with all the massive elementary particles, there are several modes of decay; the branching ratios for the dominant ones are shown in Fig. 1.3 (right) as a function of the Higgs boson mass.

Despite their small cross-section, the channels $H \to \gamma\gamma$ and $H \to ZZ^* \to 4l$ have the most significant sensitivity because they benefit from several advantages:

- the final state can be fully reconstructed,
- the mass can be measured with high precision (photons, electrons, and muons being precisely measured in CMS),
- the backgrounds are moderate.

These production processes and the various possible decay channels result in different final state signatures and can be used to study the Higgs boson couplings to different particles.

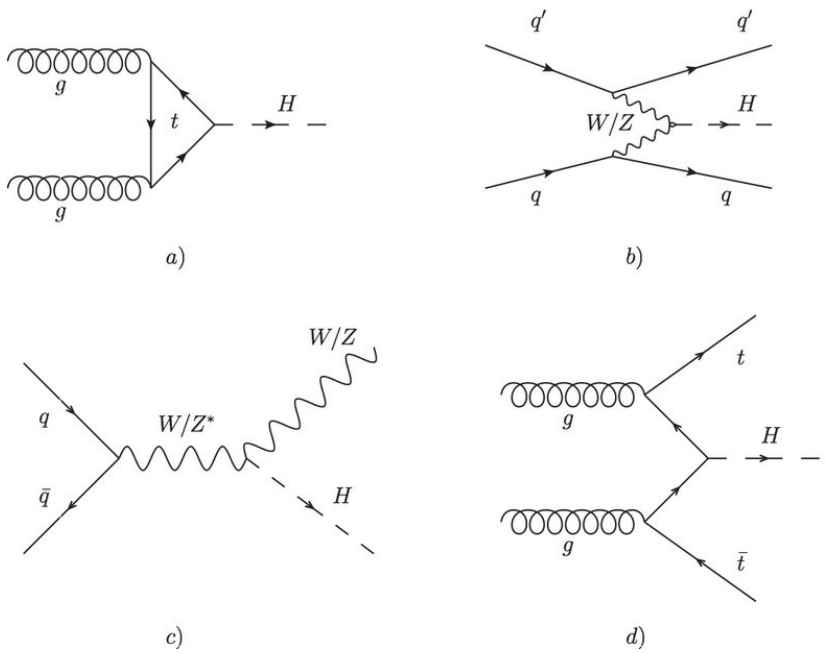



Figure 1.4: Generic Feynman diagrams contributing to the Higgs production in (a) gluon fusion, (b) weak-boson fusion, (c) associated production with a gauge boson, and (d) associated production with top quarks [20].

### 1.3.2 State of the art in CMS

The success of the SM was asserted with the discovery of a new particle with a mass $\approx$ 125 GeV announced on the 4$^{\text{th}}$ of July 2012 by the CMS [3] and ATLAS [4] collaborations. The particle's mass was precisely determined through the analysis of its decay channels, specifically $H \to \gamma\gamma$ and $H \to ZZ^* \to 4l$.

The data used for the discovery was obtained during the operational period called Run 1 (more details on the LHC operations can be found in Chapter 2). By the end of Run 1, the different analyses of the potential couplings of this particle to fermions (e.g. $H \to \tau^+\tau^-$, $H \to b^+b^-$) further provided consistent evidence for this particle to be an SM Higgs boson.

The investigation of the Higgs boson continued with the data collected during Run 2 with the center-of-mass energy of 13 TeV. The significance of Run 2 regarding Higgs boson physics is enormous. At present, all production modes have been observed, including $VH$ (e.g. [21]) and $t\bar{t}H$ (e.g. [22]) during Run 2. Moreover, many Higgs boson decay modes have also been detected, such as $ZZ$, $WW$, $\gamma\gamma$, $\tau\tau$ [23], and $bb$ [24], the latter two during Run 2. Additionally, measurements of differential cross sections and cross sections in STXS kinematic bins [25] have been performed across all observed decay channels.

The signal strength parameters are further summarized in Fig. 1.5 for observed production (left) and decay (right) channels [26]. The signal strength is proportional to $\sigma_i B^f$, where $\sigma_i$ is the production cross-section and $B^f$ is the decay branching fraction. Fits are performed under different assumptions: per production channel signal strengths ($\mu_i = \sigma_i/\sigma_i^{\text{SM}}$ with $\mathcal{B}^f = \mathcal{B}^f_{\text{SM}}$), shown in Fig. 1.5 (left), and per decay mode signal strengths ($\mu^f = \mathcal{B}^f/\mathcal{B}^f_{\text{SM}}$, with $\sigma_i = \sigma_i^{\text{SM}}$), presented in Fig. 1.5 (right). The plots are obtained from the combination of results from available Run 2 analyses.

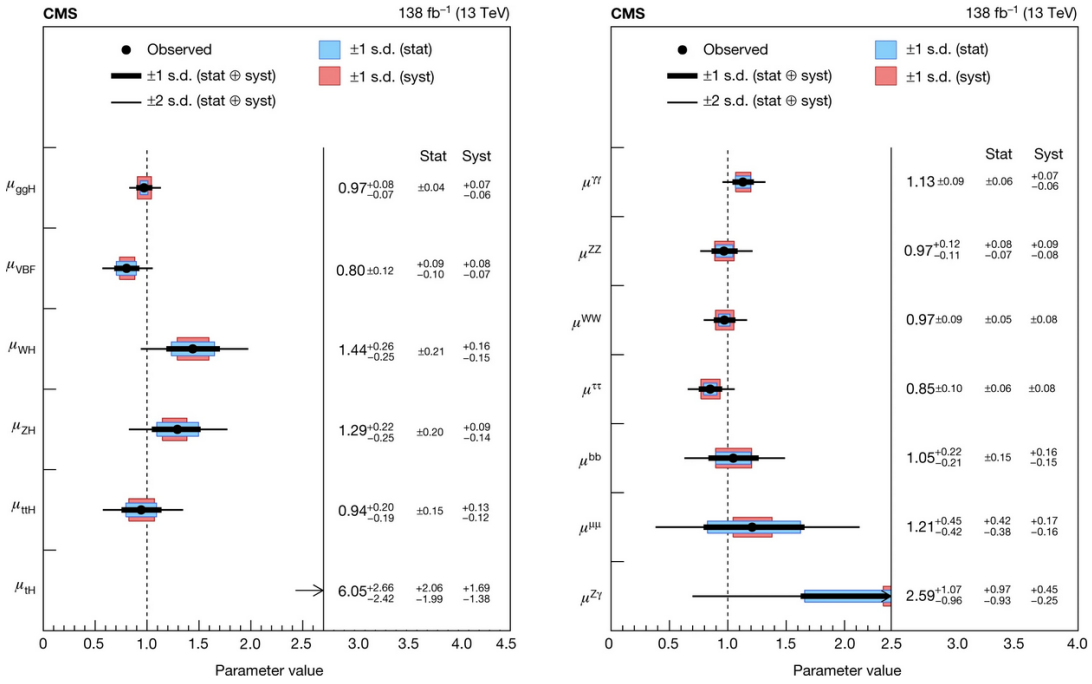



Figure 1.5: The agreement with the SM predictions for production modes and decay channels. Signal-strength parameters extracted for various production modes $\mu_i$, assuming $B^f = (B^f)_{SM}$ (left), and decay channels $\mu_f$, assuming $\sigma_i = (\sigma_i)_{SM}$ (right) [26].

Concerning the diphoton decay channel, various physics analyses based on Run 2 data were performed leading to remarkable results. Some of them are

- The first observation of $t\bar{t}H$ production channel [22] and the measurement of its CP properties [27].
- The most precise measurement of the Higgs boson mass [28]:

$$m_H = 125.38 \pm 0.11\ (stat.) \pm 0.08\ (syst.)\ GeV \tag{1.56}$$

- The inclusive and differential fiducial production cross sections measurement of the Higgs boson [29]. The inclusive cross-section was measured to be

$$\sigma_{\text{fid}} = 73.4^{+6.1}_{-5.9}\ \text{fb} \tag{1.57}$$

- The simplified template cross-section analysis [30].
- The search for Higgs boson pair production in the channel with two photons and two b-quarks [31].

All the reported results have been consistent with the SM so far. However, the first two runs only correspond to $\approx 5\%$ of the full projected dataset of LHC. Currently, Run 3 is ongoing, which will be followed by the High-Luminosity LHC operations. The gathered data will enable even more comprehensive and fundamental measurements to explore new physics.

### 1.3.3 Example physics analyses

The work presented in this document is mostly dedicated to improving the electromagnetic object reconstruction in the CMS experiment. It can benefit all of the physics analyses

as the measurement of photons and electrons is ubiquitously important. As a result, the performed work does not target any specific analysis but its potential can be underlined for two selected ones:

- $H \to \gamma\gamma$.

  The first one relates to precision measurements in the diphoton decay channel. The two-photon decay channel is one of the most important channels for precision measurements of the Higgs boson, as the two well-reconstructed photons in the final state provide a narrow invariant mass peak (mass resolution $\sim 1\%$). The main goals in this channel are measurements of production cross-sections in the different production modes (inclusively and differentially), allowing, for example, to constraint the Higgs boson couplings to other particles, but also the measurements of its mass or its CP properties in production [30].

  The possible improvements concern, first of all, the photons' energy and position resolutions, which affect the statistical precision of those measurements. Moreover, a large part of the backgrounds in these measurements comes from jets misidentified as photons (multi-jet and more importantly gamma+jet production). Currently, an identification based on Boosted Decision Tree (BDT) using shower shapes and isolation variables is used (described in Chapter 2), the distribution of the discriminating variable for signal and the background [30] is shown in Fig. 1.6. Further enhancing the photon/jet discrimination could also have an impact on the precision of these results.

  Chapter 5 presents a novel reconstruction algorithm based on Machine Learning that can both improve the mass measurement of the diphoton pair and mitigate the background coming from jets by introducing a particle ID feature. The output of the model could be further used as an input to the BDT discriminator in order to improve its results.

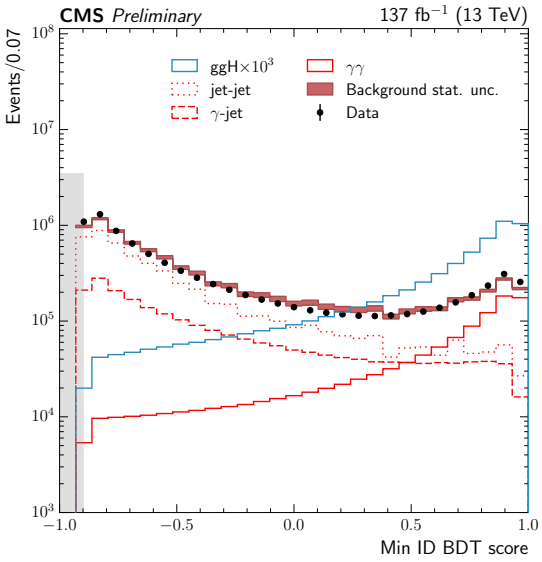



Figure 1.6: The distribution of the diphoton BDT score in events with an invariant mass in the range $100 < m_{\gamma\gamma} < 180$ GeV, for data events passing the preselection (black points), and for simulated background events (red band) [30].

- $H \to AA \to 4\gamma$. A second example is the search for the exotic Higgs decay $H \to AA \to 4\gamma$, where $A$ is a hypothetical new scalar particle decaying into two

photons [1], [32]. This type of decay enters multiple extensions of the SM such as minimal composite Higgs models, two-Higgs-doublet-like models, etc. Moreover, it has a particular interest in searches for axion-like particles. When the $A$ mass decreases, the two photons from the $A$ decay are closer and closer and can be misreconstructed as a single photon-like object. The work presented in Chapter 6 is specifically tailored to distinguish close-by photons, and, thus, can be particularly valuable in the context of this analysis.

# 2 The Large Hadron Collider and the CMS experiment

The Large Hadron Collider (LHC) [33] is the largest and most powerful accelerator in the world. It was built on the border between France and Switzerland and it is operated by CERN (L'Organisation Européen pour la Recherche Nucléaire). The LHC consists of a 27-kilometer ring, where hadrons are accelerated to high energies and subsequently collided at four locations. Each of the collision points is associated with a particular experiment: two general-purpose experiments, ATLAS (A Toroidal LHC Apparatus) and CMS (Compact Muon Solenoid), and two experiments dedicated to heavy ions and B meson physics, ALICE (A Large Ion Collider Experiment), and LHCb (Large Hadron Collider beauty), respectively.

The CMS experiment [34] is a general-purpose particle detector. It is designed to efficiently capture and identify the particles produced as a result of the collisions, along with their associated kinematic properties. In 2012 CMS was one of the two experiments that discovered the Higgs boson. Currently, the physics scope of CMS is to probe the standard model of particle physics and search for physics beyond the standard model.

In this chapter, an overview of the LHC complex is provided in Section 2.1. Afterward, Section 2.2 presents a general description of the CMS experiment and its main components. Section 2.3 covers in detail one of the sub-detectors of the CMS experiment – the electromagnetic calorimeter (ECAL). It is followed by a discussion of the offline reconstruction of electromagnetic particles in Section 2.4. An emphasis in this Chapter is placed on the ECAL reconstruction as it plays a crucial role in the subsequent work presented in this document. Finally, in Section 2.5 the High-Luminosity upgrade of LHC and a new MIP timing detector (MTD) are presented.

## 2.1 The Large Hadron Collider

The principal motivation for building the LHC collider was to investigate the nature of electroweak symmetry breaking, which could be done by searching for the Higgs boson [35]. Its discovery laid the foundation for a broad "Higgs program" at the LHC, covering a wide range of measurements aimed at uncovering the properties of this elusive particle [5]. Furthermore, the collider opens an opportunity to address other fundamental questions in particle physics: it enables precision studies of Quantum Chromodynamics (QCD), electroweak interactions, flavor physics, and the search for phenomena beyond the Standard Model (BSM).

The accelerator lies in a tunnel with a circumference of 26.7 km, situated at a depth ranging from 45 to 175 m, which was originally constructed for the Large Electron-Positron (LEP) experiment. The LHC program is mainly carried out with proton-proton collisions but it can also accelerate beams of heavy ions. The energy of the proton beams

and the design luminosity ($L = 10^{34}$ cm$^{-2}$s$^{-1}$) have been chosen in order to study physics at the TeV energy scale.

The operation of LHC started in 2010 (first physics data) with the collisions at the center-of-mass energy ($\sqrt{s}$) of 7 TeV. It marked the beginning of the first data-taking period called Run 1 with maximum $\sqrt{s} = 8$ TeV. Following a Long Shutdown in 2013-2015, the second operational Run 2 started, reaching $\sqrt{s} = 13$ TeV. Another Long Shutdown 2 was carried out between 2018 and 2022 to maintain and upgrade the accelerator complex. On 22 April 2022, the LHC resumed operations, commencing the Run 3 data-taking period with the center-of-mass energy of 13.6 TeV.

The LHC will run in the current configuration until the end of 2025. After this, a Long Shutdown 3 (LS3) will start that will last for 3 years. During this time, the LHC will undergo a major upgrade to start a new era — High-Luminosity (HL) LHC. The full timeline of the collider is shown in Fig. 2.1.

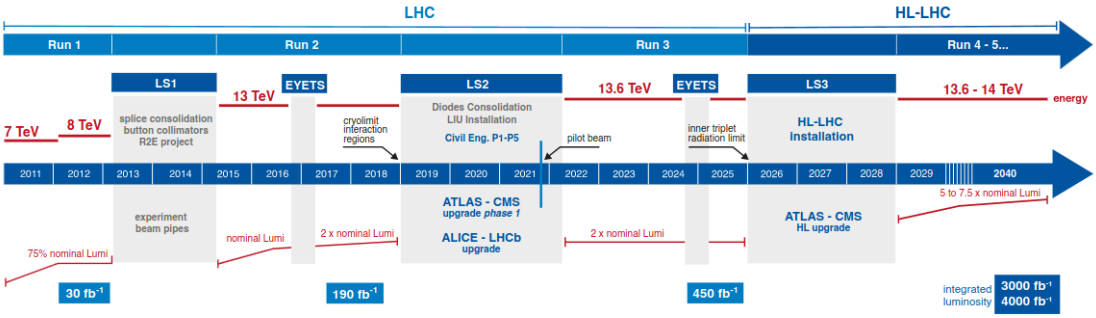



Figure 2.1: The timeline of the current LHC and the following HL-LHC [36].

### 2.1.1 Operation details

The LHC machine is part of a large complex comprising multiple accelerators [35]. At each step, the energy of the proton beam is gradually increased before it is injected into the main accelerator. The complete chain is illustrated in Fig. 2.2 and the process is as follows:

- Firstly, negative hydrogen ions ($H^-$) are generated by feeding hydrogen gas into an ion source. These $H^-$ ions are then injected into the Linear Accelerator (LINAC4), where their energy is increased to 160 MeV.
- In the subsequent step, the electrons are stripped from hydrogen ions as they pass through a thin carbon foil. This procedure leaves only nuclei containing a single proton. The resulting proton beam is injected into the Proton Synchrotron Booster (PSB), where it is further accelerated to 2 GeV.
- The proton beam is then passed to the Proton Synchrotron (PS), where its energy is increased to 26 GeV.
- The final stage before injection into the LHC involves the Super Proton Synchrotron (SPS), which pushes the energy of the beam to 450 GeV.

Within the LHC, protons are separated into two counter-rotating beams and further accelerated to their peak energy, a process that takes approximately 20 minutes. It is done using metallic chambers containing an electromagnetic field known as radiofrequency cavities (RF). In total, there are 16 RF cavities located at 4 different points that give an increase in proton energy of 0.5 MeV/turn. Under normal operation conditions, the beams can circulate for several hours before they have to be re-filled.

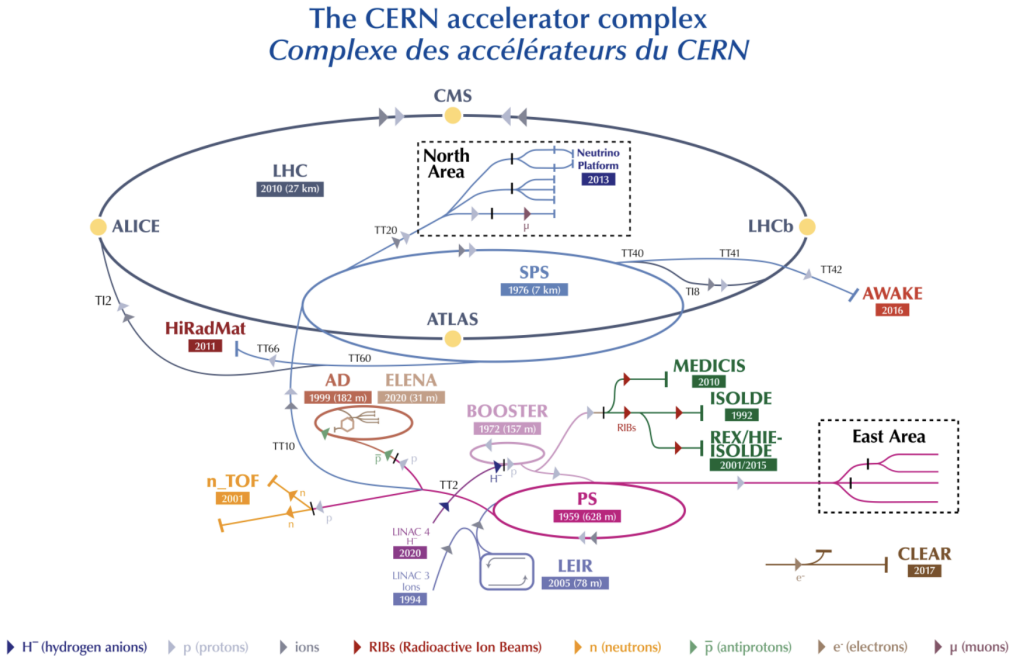



Figure 2.2: Scheme of the CERN accelerator complex [37].

In the accelerator, the proton beams travel inside two tubes kept at an ultrahigh vacuum to mitigate unwanted collisions. They are guided by a powerful magnetic field created by about 10 000 copper-clad niobium-titanium superconducting magnets. These magnets include 1 232 dipole magnets, which bend the beam, and 392 quadrupole magnets, which keep the particles focused in narrow beams. Additionally, other quadruples compress the beams right before the interaction points, while magnets of higher multiple orders correct minor imperfections in the magnetic field. The superconducting state is achieved at a temperature of -271.3°C; in order to provide such conditions, approximately 96 tonnes of superfluid helium-4 are used.

### 2.1.2 Beam parameters

Within the beams, protons are organized into bunches, with each bunch containing $10^{11}$ protons under nominal conditions. They are prepared along the injection chain and are separated by a time interval of $\Delta\tau \approx 25$ ns. This results in collisions occurring between two beams at discrete intervals with a frequency of approximately 40 MHz. The gaps between the bunches are used for synchronization, calibration data acquisition, and providing resets to front-end electronics. Overall the LHC accommodates 2808 bunches during each fill.

The number of events per second produced as a result of a collision for a given process is determined by a cross-section ($\sigma$) of this process and instantaneous luminosity $L$:

$$N_{\text{event}} = L\sigma_{\text{event}} \tag{2.1}$$

The luminosity depends only on the beam parameters and for a Gaussian beam distribution is calculated according to:

$$L = \frac{N_b^2 n_b f_{\text{rev}} \gamma_r}{4\pi\epsilon_n \beta*} F, \tag{2.2}$$

where $N_b$ is the number of particles per bunch, $n_b$ the number of bunches per beam, $f_{rev}$ the revolution frequency, $\gamma_r$ the relativistic gamma factor, $\epsilon_n$ the normalized transverse beam emittance, $\beta*$ the beta function at the collision point, and $F$ the geometric luminosity reduction due to the crossing angle at the interaction point.

Along the fill, the instantaneous luminosity progressively decreases due to beam losses during the collisions, necessitating re-filling. The total amount of data delivered at the experiments is given by integrated luminosity over time ($\int_0^t Ldt$).

Luminosity is a crucial parameter of the LHC machine, and to maximize the potential for new discoveries, it should be as high as possible. However, larger luminosity results in multiple proton-proton collisions, where interesting high-energy collisions (hard) are contaminated by additional unwanted interactions (soft) [38]. The latter is called *pileup*, and it pollutes the final state of interest complicating the reconstruction process. In order to mitigate pileup, special techniques are used, and several of them will be discussed further in Chapter 3 and Chapter 5.

### 2.1.3 High-Luminosity LHC

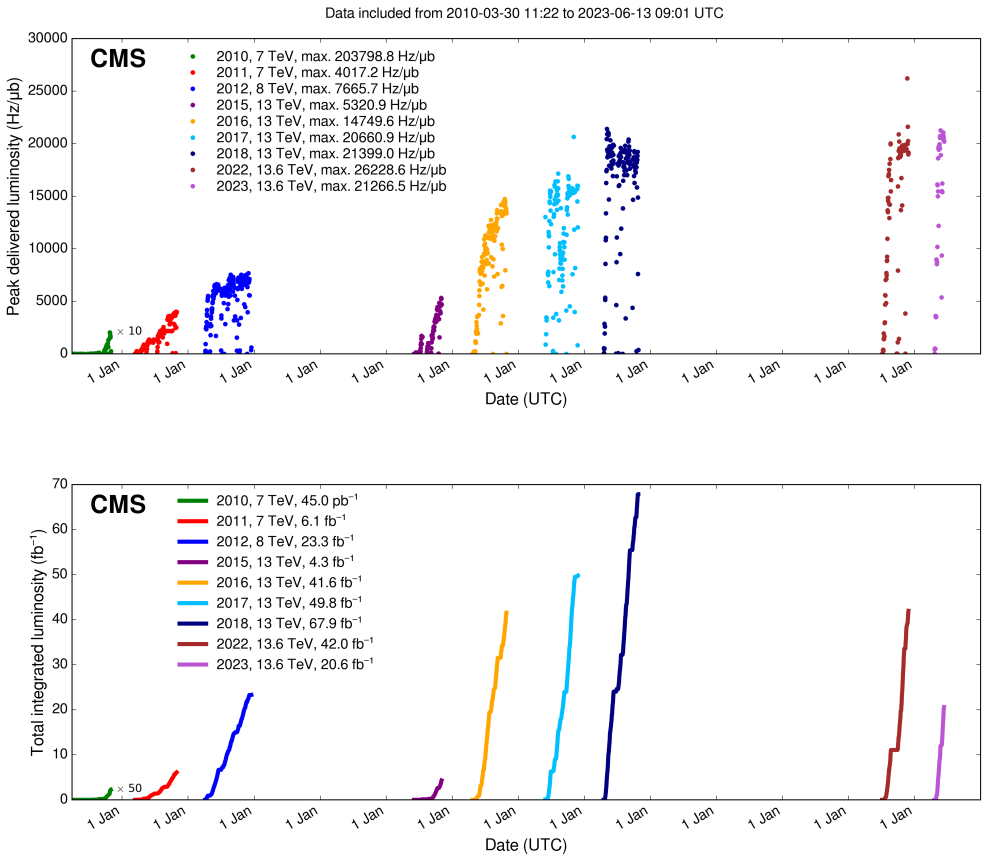



Figure 2.3: Peak instantaneous luminosity (top) and total integrated luminosity (bottom) delivered to CMS versus day during Run 1, Run 2, and Run 3 [39].

To enhance and expand the discovery potential of the LHC, it is undergoing a major upgrade [40]. The primary objective is to increase the instantaneous luminosity by a factor of 5 compared to the original design value ($10^{34}$ cm$^{-2}$s$^{-1}$). It aims to deliver approximately 3000 fb$^{-1}$ of integrated luminosity over the operation period of 10 years. For comparison, the target value for Run 3 after 4 years is 450 fb$^{-1}$. The peak delivered luminosity and the integrated luminosity values for different periods of LHC

operations are presented in Fig. 2.3. The upcoming collider era after the upgrade is called High-Luminosity (HL) LHC.

While the upgrade enables new possibilities for discoveries and precision measurements, it also presents unprecedented technical challenges for the experiments. One major problem arises from the significant increase in the number of pileup events: from an average of 30 interactions during Run 3 to ≈140-200 during the HL-LHC.

To address this issue, the CMS experiment has already undergone various upgrades to its detectors during LS2 that will further continue during LS3 aimed to maintain or improve the performance in this new environment. Additionally, novel detectors, such as a MIP Timing Detector (MTD), are being developed with the goal to mitigate the effects of pileup. Further details about HL-LHC upgrades and MTD can be found in Section 2.5.

## 2.2 The CMS experiment

The Compact Muon Solenoid (CMS) experiment is a general-purpose detector situated in an underground cavern in Cessy, France. It is associated with one of the largest scientific collaborations worldwide, involving approximately 6,000 members from nearly 60 countries. The physics scope of CMS is to probe the standard model of particle physics and search for physics beyond the standard model using proton-proton and lead-lead collisions at center-of-mass energy ranging from 7 TeV (first collisions in 2010) to 13.6 TeV (collisions recorded since July 2022).

The CMS detector is relatively compact, measuring 28.7 meters in length, 15.0 meters in width, and 15.0 meters in height, with a weight of approximately 14,000 tons. It consists of various sub-detectors, arranged in cylindrical layers around the beam axis, each serving a distinct purpose.

Starting from the interaction point, the produced particles first enter the *tracker*. The CMS detector is immersed in a strong magnetic field of 3.8 T, which bends the trajectories of charged particles and, as a result, enables the measurement of the electric charges and momenta of these particles in the tracker. Electrons and photons are captured by the electromagnetic calorimeter ($ECAL$), where their energies and directions can be determined from electromagnetic showers. Both charged and neutral hadrons can also initiate hadronic showers within the ECAL, which are subsequently absorbed in the hadron calorimeter ($HCAL$). These showers are used to reconstruct the energies and directions of the corresponding particles. Muons can traverse the detectors with little to no interaction and are measured in the tracker and *muon detectors*, which represent the final layer of the CMS experiment.

### 2.2.1 Coordinate system

The coordinate system adopted by the CMS experiment is shown in Fig. 2.4. Its origin is centered at the collision point, with the $z$-axis pointing along the beam direction, the $y$-axis pointing vertically upward, and the $x$-axis pointing toward the center of the LHC.

Given the cylindrical shape of the detector, polar coordinates are also commonly used. The azimuthal angle $\phi$ is measured from the $x$-axis in the $x - y$ plane. The radial coordinate in this plane is denoted as $r$. The polar angle $\theta$ is measured from the $z$-axis. However, a more frequently used spatial coordinate is the *pseudorapidity*, which is defined as follows:

$$\eta = -\ln \tan \frac{\theta}{2} \tag{2.3}$$

The pseudorapidity values $\eta$ corresponding to different $\theta$ angles are shown in Fig. 2.5.

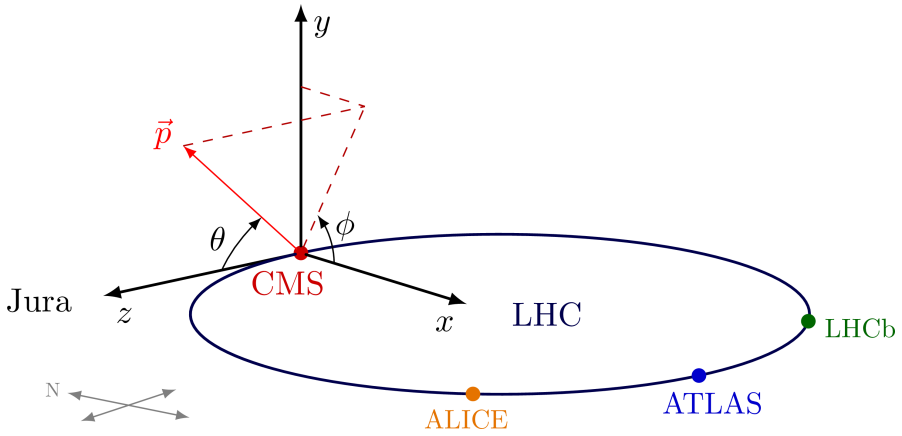



Figure 2.4: The CMS coordinate system [41].

The reason to use pseudorapidity comes from the composite nature of the protons, resulting in the interactions taking place at the level of their constituents (partons). These colliding partons carry distinct longitudinal momentum fractions of the initial protons causing their center-of-mass frames to have varying longitudinal boosts. The difference in pseudorapidity is Lorentz invariant along longitudinal boosts. Hence, if the emission angle of each particle is defined by $(\eta, \phi)$, the spatial separation between two particles can be expressed as $\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$ and is invariant with respect to boosts along the beam axis.

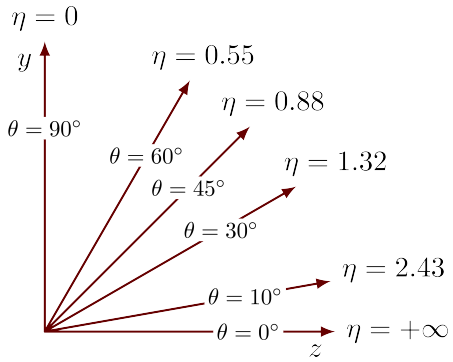



Figure 2.5: Illustration of the relation between the pseudorapidity $\eta$ and the polar angle $\theta$ [41].

Similarly, due to the unknown fraction of the momenta of the partons, the transverse momentum ($p_T$) and energy ($E_\mathrm{T}$) of the produced particles with respect to the beam axis are preferred. For a particle with mass $m$ and momentum $p$ it can be estimated as follows:

$$p_T = \frac{p}{\cosh(\eta)} \tag{2.4}$$

$$E_\mathrm{T} = \sqrt{m^2 + p_T^2} \tag{2.5}$$

Since the collisions occur along the beam axis, the final transverse momentum should be zero. The imbalance of energy in the transverse plane is denoted by $E_T^{\mathrm{miss}}$, which represents the missing transverse energy.

### 2.2.2 Detector structure

The structure of the CMS experiment is depicted in a schematic illustration in Fig. 2.6. In this section, a brief description of each sub-system of the full detector is presented.

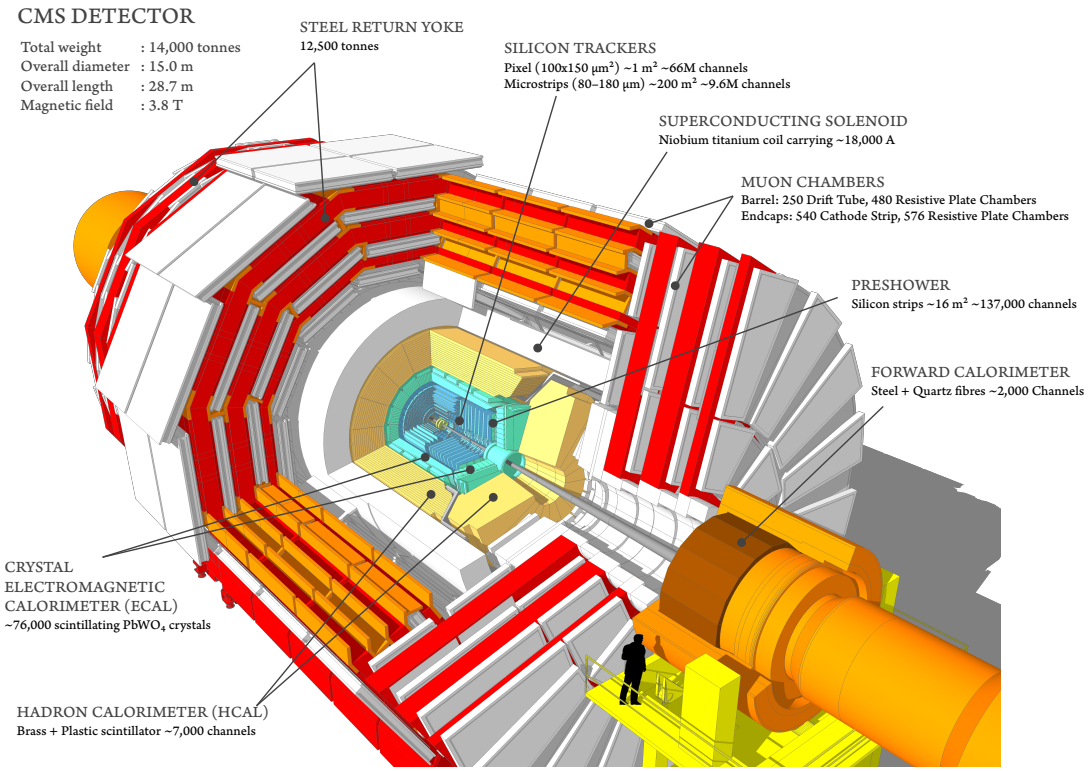



Figure 2.6: The CMS detector overview [42].

#### Solenoid magnet

The central feature of the CMS apparatus is the niobium-titanium superconducting solenoid magnet with a length of 12.5 m and a radius of 3.15 m. It generates an axial and uniform magnetic field of 3.8 T. To minimize the amount of material in front of the calorimeters while accommodating the tracker inside, the magnet is positioned after the HCAL. Outside the solenoid coil, the magnetic field is contained by a return yoke consisting of three layers of steel interleaved with four muon detector planes.

#### Tracking system

The inner tracker system is a cylindrically-shaped detector placed directly around the interaction point. It has an outer radius of 1.20 m and a length of 5.6 m, which covers the pseudorapidity range up to $|\eta| = 2.5$. It is divided into two parts: the pixel tracker and the silicon microstrip detector.

The pixel detector represents the innermost part of the tracking system [43], located in a particularly harsh radiation environment. It is divided into 1,856 segmented sensor modules, each consisting of 160 x 416 pixels. The total silicon area is 1.9 $m^2$ and the standard pixel size is 100 x 150 $\mu m^2$.

The pixel tracker is surrounded by four layers consisting of 10 cm x 180 $\mu$m silicon strips, followed by the six layers of 25 m x 180 $\mu$m strips [44].

The system provides robust charged-particle trajectories (*tracks*) and detailed location of particle collision (*vertex*) reconstructions from the signals in the tracker. The fine

granularity of the detector enables the separation of closely-spaced particle tracks. For a high $p_T$ track, (100 GeV) the $p_T$ resolution is about 1-2% in the central region ($|\eta| < 1.6$) and a bit worse in the endcaps due to the shorter lever arm of these tracks in the $x - y$ plane of the tracker [45].

The tracker represents a significant amount of material in front of the calorimeters. At $|\eta| \approx 1.5$, the probability for a photon to convert or for an electron to emit a bremsstrahlung photon by interacting with this material is approximately 85%. Similarly, a hadron can experience a nuclear interaction with a 20% probability before reaching the ECAL. These complications must be considered in the final object reconstruction process and are further discussed in this chapter for the case of electromagnetic particles.

#### Electromagnetic calorimeter (ECAL)

The ECAL detector is primarily designed to reconstruct the energy and direction of electrons and photons. It is a homogeneous calorimeter made from $PbWO_4$ scintillating crystals. Its detailed characteristics are further described in Section 2.3.

#### Hadron calorimeter (HCAL)

The HCAL is designed to measure the energy deposited by hadrons. It is divided into 4 major parts [46]: HCAL Barrel (HB) with $|\eta| < 1.3$, HCAL Endcap (HE) covering $1.3 < \eta < 3.0$, HCAL Outer (HO) placed outside the solenoid volume with $|\eta| < 1.4$, and HCAL Forward (HF) extending the coverage up to $|\eta| = 5.0$.

The HB and HE detectors are sampling calorimeters with brass used as an absorber and plastic scintillator as the active material. The HO calorimeter, which serves as a "tail-catcher" for hadronic showers, uses the same active material, and the steel return yoke and magnet material as an absorber. The HF is a Cherenkov calorimeter based on a steel absorber and quartz fibers.

The combined (ECAL + HCAL) calorimeter energy resolution was measured in a pion test beam with energy between 2 and 350 GeV [47] and is given by the formula:

$$\frac{\Delta E}{E} = \frac{84.7\%}{\sqrt{E}} \bigoplus 7.4\% \tag{2.6}$$

#### Muon detectors

The muon gas-ionization chambers represent the outermost layer of the CMS detector, their primary goal is to identify muons and measure their momenta. The muon planes are placed outside the solenoid volume and consist of three components [48]: drift tube (DT) chambers that cover the region $|\eta| < 1.2$, cathode strip chambers (CSC) detect muons in the region $0.9 < |\eta| < 2.4$, and a system of resistive plate chambers (RPC) covering $|\eta| < 1.9$.

The muon momentum resolution varies with pseudorapidity, but overall muon momentum resolution of 5-8% at 10 GeV, 20-40% at 1 TeV is obtained when using only the muon system. The numbers improve significantly to 1-1.5% at 10 GeV, 6-17% at 1 TeV when the combination of the inner tracker and outer muon system is used [49].

### 2.2.3 Trigger and Data Acquisition system

A dedicated trigger and data acquisition system is used to select potentially interesting events from the collisions, which are then stored for subsequent analysis. The triggering process consists of two levels [50]:

- **Level 1 (L1)** is a hardware trigger that uses information from the ECAL and muon RPCs to determine the selection or rejection of events. At this level, the event rate is decreased from 40 MHz to 100 kHz. Events that successfully pass this trigger are forwarded to the Data Acquisition System, where information from the 16 million channels in the CMS sub-detector systems is combined by the event builder to form a single event.
- **High-Level Trigger (HLT)** is a software-based trigger that performs a more advanced analysis based on a simplified version of the full event reconstruction. It further reduces the event rate to approximately 1 kHz, enabling storage and subsequent offline analysis of the data.

## 2.3 Electromagnetic calorimeter

The ECAL plays a vital role in the reconstruction of photons, electrons, and the measurement of jet energies and missing transverse momentum [51]. It is a hermetic and homogeneous calorimeter, consisting of two main parts:

- The **barrel section (EB)** covers the pseudorapidity region of $0 < |\eta| < 1.479$ and has an inner radius of 129 cm. The barrel granularity is 360-fold in $\phi$ and (2 x 85)-fold in $\eta$, resulting in 61,200 crystals. These crystals are arranged in 36 identical supermodules. To avoid acceptance gaps, the crystals are tilted by a small angle of 3° with respect to the line from the nominal interaction vertex.
- The two **endcaps (EE)** cover the region of $1.479 < |\eta| < 3.0$ and are placed at a distance of 314 cm from the vertex. Each endcap is divided into two halves or Dees, comprising 3,662 crystals. These crystals are arranged into 5x5 units called supercrystals.

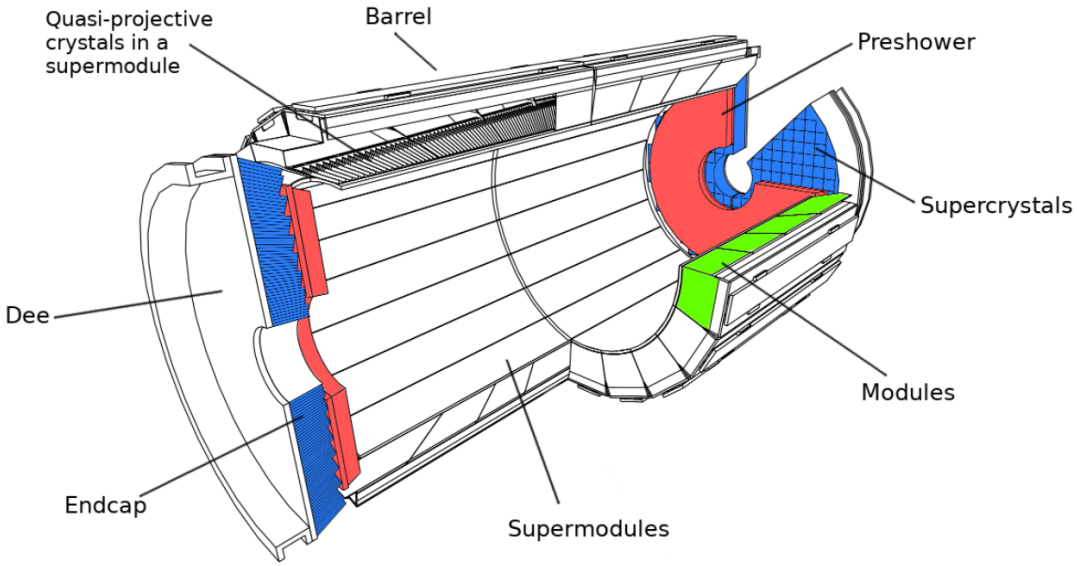



Figure 2.7: The ECAL detector layout [52].

In addition, a preshower detector with a much finer granularity is installed in front of each endcap disk. The preshower is a sampling calorimeter made from SiPb. Originally, its purpose was to discriminate prompt photons (coming directly from the interaction

vertex) from those originating from $\pi^0$ decays. However, the discriminative capabilities of the detector can not be fully exploited due to the parasitic signals created by a large number of $\pi^0$ produced by hadron interactions in the tracker. Thus, in the reconstruction algorithm, the energy deposited in the preshower is simply added to the objects detected in the main parts of ECAL.

The schematic layout of the ECAL detector is shown in Fig. 2.7.

### 2.3.1 Electromagnetic showers

The energy and direction of the photons and electrons can be determined from the energy patterns they leave within the calorimeter. When electromagnetic (EM) particles interact with the detector, they generate a shower of secondary particles with progressively decreasing energies [53]. Despite the complexity of shower development, the interactions of electrons and photons with matter follow well-understood quantum electrodynamics (QED) processes (shown in Fig. 2.8 for different energies), allowing the main parameters of the showers to be parameterized [54].

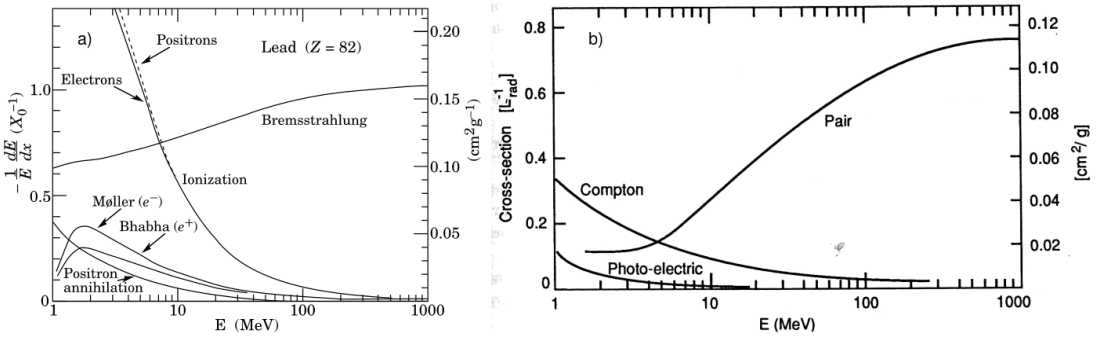



Figure 2.8: Fractional energy lost per radiation length in lead by electrons and positrons as a function of electron/positron energy (left) and photon interaction cross-section in lead as a function of energy (right) [54].

In the case of high-energy particles ($> 1$ GeV), electrons primarily lose their energy through bremsstrahlung, emitting secondary photons, while photons create an electron and a positron through pair production. These secondary particles continue the creation through the same mechanism, giving rise to a shower of particles with decreasing energies. The process stops when the energy of the electron component reaches the critical energy ($E_c$), which can be defined as the energy at which the electron ionization losses and bremsstrahlung losses become equal. Below $E_c$, electrons dissipate their energy by ionization rather than generating new shower particles.

The main parameters of an EM shower are its longitudinal and transverse sizes. The first one can be characterized using the radiation length, which depends on the material properties:

$$X_0 \left(\mathrm{g/cm^2}\right) \simeq \frac{716 \text{ g cm}^{-2} A}{Z(Z+1)\ln(287/\sqrt{Z})}, \tag{2.7}$$

where Z and A are the atomic number and atomic molar mass of the material, respectively. The radiation length represents the average distance that an electron needs to travel in a material before its energy is reduced by $1/e$ of its initial values.

The transverse size of the EM shower comes from the multiple scattering of secondary particles. It is characterized by the Molière radius:

$$R_M \left(\mathrm{g/cm^2}\right) \simeq 21 \text{ MeV} \frac{X_0}{E_c(\mathrm{MeV})} \tag{2.8}$$

It represents the radius of the cylinder that contains 90% of the shower energy deposition on average.

### 2.3.2 Crystal parameters and photodetectors

The energy measurement in the ECAL relies on detecting the light produced by electromagnetic (EM) showers, which are recorded as energy deposits in the calorimeter crystals. The choice of material for the ECAL is crucial to ensure that the majority of the EM shower's energy is contained within the detector.

In the CMS experiment, lead tungstate ($PbWO_4$) scintillating crystals are used for ECAL. The advantages of this material are:

1. Short radiation length ($X_0 = 0.89$ cm) and Moliere radius (2.2 cm) allow for the construction of a compact detector with high granularity, enabling precise energy measurements and spatial resolution.

2. Fast signal production: Approximately 80% of the scintillation light is generated within a time window of 25 ns, ensuring prompt and efficient signal detection.

3. Radiation hardness: The lead tungstate crystals exhibit resilience to radiation levels of up to 10 Mrad, making them suitable for the challenging radiation environment of the CMS experiment. However, the crystal transparency to light varies with irradiation and has to be constantly monitored. The system is briefly discussed in Section 2.3.4.

The parameters of the crystals used in different regions of the ECAL are the following:

- Barrel: each crystal covers a $\Delta\eta$ x $\Delta\phi$ region of 0.0175 x 0.0175. It corresponds to a crystal front face area of 22 x 22 $mm^2$, which matches the Moliere radius. The length of the crystal is 230 mm or about 25.8 $X_0$, which is sufficient to contain up to 98% of the shower produced by EM particles with energies up to 1 TeV.

- Endcaps: each crystal has a front face cross-section of 28.6 x 28.6 $mm^2$ and a length of 220 mm (24.7 $X_0$).

Due to the relatively low light yield of lead tungstate crystals (30 photons/MeV), high-gain photodetectors are employed to enhance the signal. In the barrel region, silicon avalanche photodiodes (APDs) are used, while in the endcaps it is done by vacuum phototriodes (VPTs). They are specifically designed to be resistant to the high radiation levels and the strong magnetic field of CMS.

### 2.3.3 Performance

The ECAL resolution can be parametrized as follows [51]:

$$\left(\frac{\sigma}{E}\right)^2 = \left(\frac{a}{\sqrt{E}}\right)^2 + \left(\frac{\sigma_n}{E}\right)^2 + c^2, \tag{2.9}$$

where

- $a$ is the stochastic term. It incorporates the statistical fluctuations in shower development and in its detection.

- $\sigma_n$ is the noise term. It includes the contributions from the readout electronics.

- $c$ is the constant term, which covers the energy leakage from the back of the calorimeter, non-uniformity of the longitudinal light collection, and imperfections in channel-to-channel inter-calibrations.

These terms were initially estimated in the test beam with no magnetic field, no dead material in front of the calorimeter, perfect inter-calibration, and stability of the single-channel responses as described in Ref. [55]. The corresponding values were as follows: $a = 0.03$ GeV$^{-\frac{1}{2}}$, $\sigma_n = 40$ MeV per channel, and $c = 0.0030$.

However, the ECAL resolution degrades over time. The noise term increases due to the rise in the dark current of the APD (VPT) caused by irradiation and the constant term worsens due to increased transparency losses.

For the work presented in this document, the estimations of the parameters for Eq. (2.9) are taken from the values in the barrel obtained during Run 2 and subsequent previsions for Run 3. The constant term is chosen to be 0.0035 based on the estimation for Run 2 (Fig. 2.9 (left)), which is also the value expected for Run 3. This plot is obtained by fitting the $Z \to ee$ invariant mass, using an unbinned likelihood comparing the invariant mll lineshape after reconstruction to the one from a parametrized model with the energy scale and resolution of both electrons as parameters. The expected noise level at the end of Run 3 operations at $|\eta| \approx 1$ is taken, it is estimated from Fig. 2.9 (right) and equals to 167 MeV per channel. The stochastic term did not change.

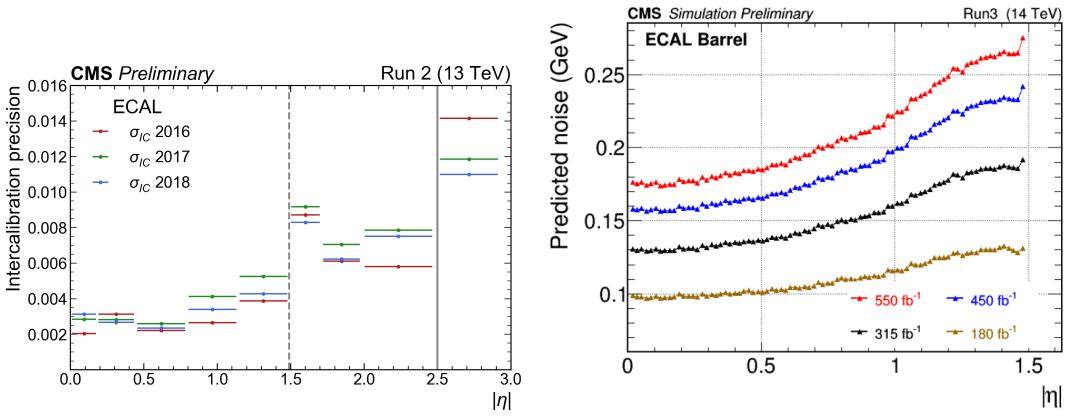



Figure 2.9: On the left: residual miscalibration of the ECAL channel inter-calibration, as a function of pseudorapidity with the dataset recorded during LHC Run 2 [56]. The dashed line represents the end of the ECAL Barrel ($|\eta| < 1.49$), while the full line refers to the end of the tracker acceptance ($|\eta| < 2.5$). On the right: ECAL average transverse noise projection for Run 3 calculated in GeV as a function of the Barrel and Endcap $\eta$ [57].

### 2.3.4 Laser monitoring system

The ECAL crystals both lose the transparency with irradiation and recover it when there are no collisions or when they are heated. The transparency losses occur due to the creation of defects in the crystal structure when irradiated, known as color centers, that further influence the scintillation light transmission in the crystals. In order to maintain the excellent energy resolution performance of the ECAL, a dedicated laser monitoring system is developed that tracks these losses.

The system works by injecting a laser signal into the crystals and in reference PN diodes, and comparing their responses. More details can be found in Ref. [58]. To

ensure the prompt reconstruction of data, it must operate in a quasi-online manner, providing transparency corrections for each crystal at approximately 40-minute intervals and completing the process within 48 hours.

## 2.4 Offline reconstruction of electromagnetic objects

Electrons and photons play a crucial role in the CMS experiment, and their reconstruction is carried out with high precision and efficiency. These electromagnetic (EM) particles deposit the majority of their energy within the ECAL and, in addition, electrons leave a trace in the tracker. From this information, the energy and direction of electrons and photons can be determined.

### Particle Flow

The reconstruction of EM particles is fully integrated into the CMS global event description framework, called *Particle Flow (PF)* [59]. This approach aims to optimally combine the information from the sub-detectors to identify and characterize all final-state particles, including charged or neutral hadrons, photons, electrons, and muons. Each of these particles leaves a distinct signature in the CMS experiment, schematically represented in Fig. 2.10 and briefly discussed in Section 2.2.

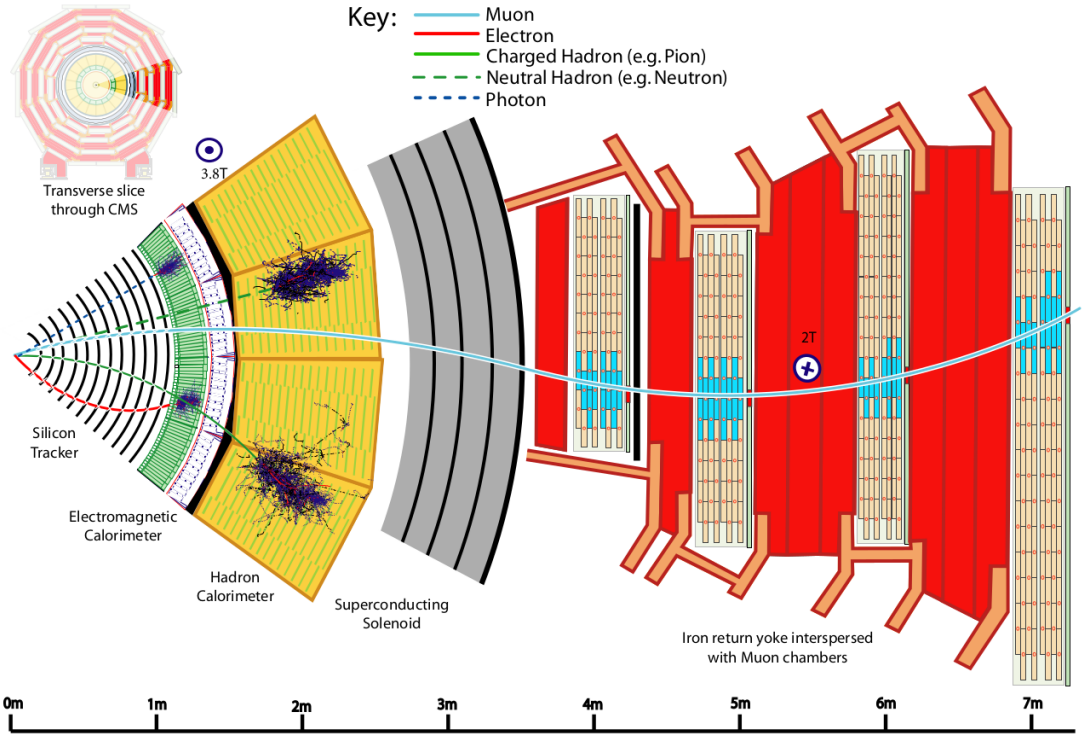



Figure 2.10: Slice of CMS in the transverse view and experimental signature of particles in the sub-detectors [59].

The basic elements of the PF framework are trajectories of the charged particles reconstructed in the tracker and muon detectors, known as *tracks*, and the *clusters* of energy deposited in ECAL and HCAL. These elements are associated with each other by a geometrical connection (*link*) in the ($\eta$, $\phi$) plane. Based on the characteristic signatures of the different particles in the CMS experiment, a simplified description of their identification is as follows:

- Muons are identified through the presence of tracks in both the tracker and muon detectors.
- Neutral hadrons are identified as clusters in HCAL and ECAL that are not associated with any tracks.
- Charged hadrons can have clusters in both the HCAL and ECAL with a linked track.
- The presence of ECAL clusters without a link to a track indicates a photon. Photons can also convert to electron-positron pair before reaching the ECAL, effectively creating a signature in the tracker as well. This aspect is further discussed in more detail.
- Electrons are identified by the presence of clusters in the ECAL that are connected to a charged track. The reconstruction of electron tracks uses a tracking algorithm, known as the Gaussian Sum Filter (GSF), which takes into account the change in the trajectory due to the emission of the bremsstrahlung photons.

The PF framework produces a list of identified particles that can be used directly to reconstruct higher-level objects such as jets and $\tau$-leptons, calculate the missing transverse momentum, and quantify the isolation of an energetic particle. An example of jet reconstruction from its constituents is shown in Fig. 2.11. The presented jet has a transverse momentum of 65 GeV and is made of five particles: two charged hadrons (a $\pi^+$ and a $\pi^-$), two photons (from a $\pi^0$ decay), and one neutral hadron (a $K^0_L$). Track $T_1$ is linked to the ECAL cluster $E_1$ and to the HCAL clusters $H_1$ and $H_2$, while track $T_2$ is linked only to the HCAL clusters $H_2$ and $H_1$. These elements form two PF blocks: the first one ($T_1$, $E_1$, $H_1$) corresponds to $\pi^-$ and the second one ($T_2$, $H_2$) to $\pi^+$. The other three ECAL clusters are not linked to any tracks or clusters and each creates its own PF block, corresponding to two photons and the neutral kaon.

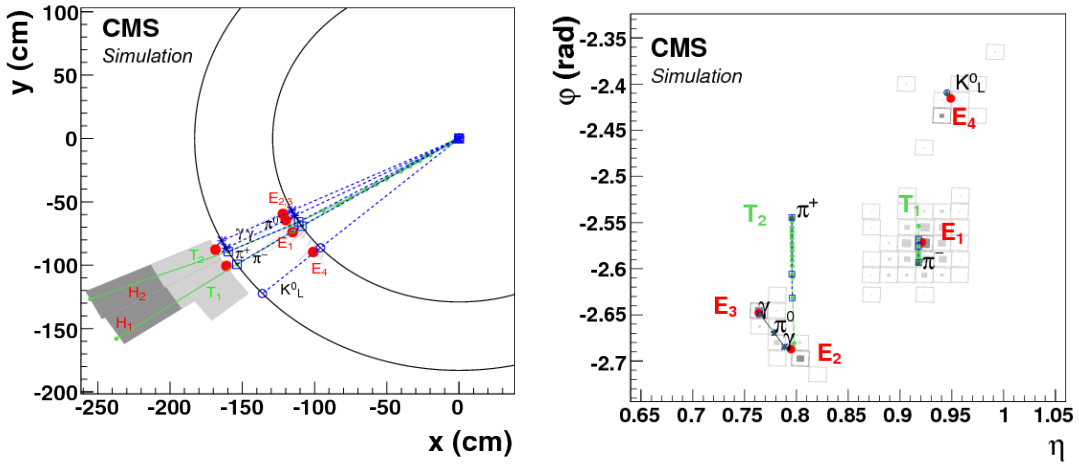



Figure 2.11: Event display of a jet made of five particles in the $(x, y)$ view (left) and in the $(\eta, \phi)$ view on the ECAL surface (left) [59]. On the left plot, ECAL and HCAL detectors are represented as circles centered around the interaction point. The $K^0_L$, the $\pi^-$, and the two photons from $\pi^0$ decay are detected as four well-separated ECAL clusters denoted as $E_{1,2,3,4}$. The $\pi^+$ does not create a cluster in the ECAL. The two charged pions are reconstructed as charged-particle tracks $T_{1,2}$, appearing as vertical solid lines in the $(\eta, \phi)$ views and circular arcs in the $(x, y)$ view.

**Overview of the ECAL reconstruction**

In more detail, the reconstruction of EM particles in ECAL is done through several sequentially applied algorithms [60]:

1. The amount of energy deposited in each crystal of the calorimeter is reconstructed as *PFRechits*, described in Section 2.4.1.

2. An energy cluster, called *PFCluster*, is formed by grouping together PFRechits overpassing a pre-defined energy threshold. Each PFCluster represents either a single particle or a combination of overlapping particles. Further described in Section 2.4.2.

3. Due to the presence of material in front of the ECAL, electrons can undergo bremsstrahlung, and photons can convert into electron-positron pairs before reaching the calorimeter. In these cases, multiple electromagnetic showers and corresponding PFClusters originating from a single initial particle will appear in the ECAL. A specific algorithm ("Mustache") is used to combine these PFClusters into a single object, called SuperCluster. This step is necessary to correctly measure the energy of the primary electron or photon. The details of the SuperClustering are discussed in Section 2.4.3.

Furthermore, additional PFClusters can be recovered by using the tracking information: dedicated algorithms are employed to identify PFClusters compatible with the tracks coming from photon conversion or resulting from electron bremsstrahlung. The new object called a *"refined" SuperCluster*, combines the Mustache SuperClusters, GSF tracks, and additional tracks associated with electrons, as well as conversion tracks and their associated PFClusters. It serves as a baseline for the formation of electron and photon candidates.

To account for the energy losses due to shower leakage, dead channels, and intermodule gaps, a multivariate technique, a semi-parametric Boosted Decision Tree, is further used to estimate the energy corrections that should be applied to SuperClusters.

On the level of particle flow reconstruction, loose selection criteria are applied to the EM object candidates to separate the prompt photons and electrons from the misidentified hadrons and non-isolated $e/\gamma$. The objects passing these criteria that have an associated GSF track are marked as electrons and without it as photons. The list of obtained EM particles serves as a basis for the majority of the analysis. Further, tighter selection criteria can be applied during the analysis to separate prompt $e/\gamma$ from their background sources.

### 2.4.1 PFRechits

The signals produced by the $PbWO_4$ crystals and shaped by ECAL electronics can extend over several hundred nanoseconds. Due to the fact that the timing gap between two bunches is $\approx 25$ ns, the signals from neighboring bunch crossings (BX) can overlap with each other, which is known as *out-of-time (OOT)* pileup. It results in increased electronic noise and degraded energy resolution of the calorimeter.

The length of the signal needs to be short enough to avoid integrating pileup and long enough to capture as much scintillation signal (and its reflections in the crystal) as possible. In the front-end electronics of the ECAL, 12-bit analog-to-digital converters (ADCs) are employed to sample the analog signals from the detectors (APDs, VPTs) at a rate of 40 MHz. For each trigger received, ten consecutive samples are stored, taken at intervals of 25 ns.

In order to suppress OOT pileup a "multifit" algorithm is employed [61]. It uses a template fit with $N_{BX}$ parameters, comprising one in-time (IT) and up to nine OOT amplitudes (up to 5 occurring before IT pulse and up to 4 after IT pulse). The fit minimizes $\chi^2$ defined as:

$$\chi^2 = \left( \sum_{j=0}^{N_{\mathrm{BX}}} A_j \vec{p}_j - \vec{S} \right)^T \mathbf{C}^{-1} \left( \sum_{j=0}^{N_{\mathrm{BX}}} A_j \vec{p}_j - \vec{S} \right), \tag{2.10}$$

where the vector $\vec{S}$ comprises the 10 readout samples, $s_i$, after having subtracted the pedestal value, $\vec{p}_j$ are the pulse templates for each BX, and $A_j$, which are obtained by the fit, are the signal pulse amplitudes in ten consecutive BXs, with $A_5$ corresponding to the IT BX. The total covariance matrix $C$ includes the correlation of the noise and the signal between the different time samples.

The $\chi^2$ is minimized iteratively for each crystal in the considered event. Examples of fitted amplitude shapes are shown in Fig. 2.12.

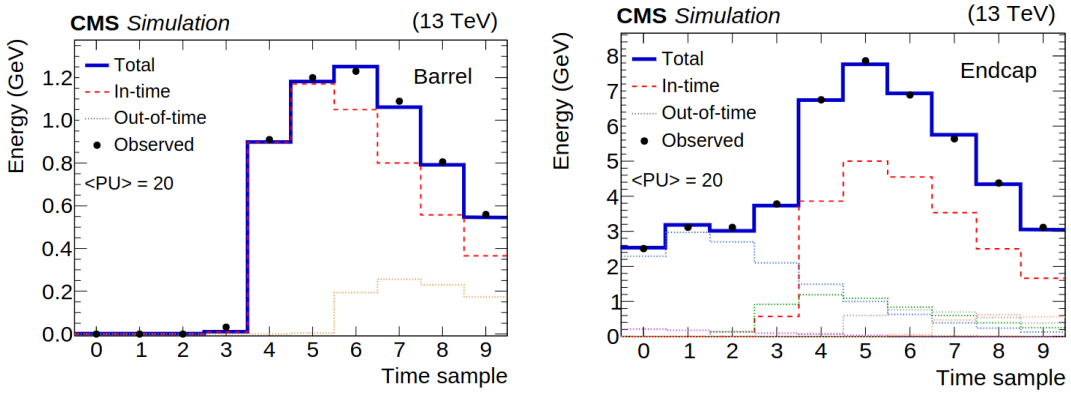



Figure 2.12: Two examples of fitted pulses for simulated events with 20 average pileup interactions and 25 ns bunch spacing. Signals from individual crystals are shown. They arise from a $p_T = 10$ GeV photon shower in the barrel (left) and in an endcap (right) [61].

### 2.4.2 PFClustering

The reconstructed PFRechits that pass a pre-defined energy threshold ($E_{\mathrm{thr}}^{\mathrm{rechit}}$) are further combined together into PFClusters, from which the positions and the energies of the corresponding particles can be evaluated [59].

The algorithm follows several steps. First, all the potential PFClusters have to be formed:

1. The crystals with deposit exceeding a certain energy threshold ($E_{\mathrm{thr}}^{\mathrm{seed}}$) and larger than the energy of their adjacent cells (either sharing a side or a corner) are selected as *seeds*. They serve as a starting point for growing one cluster.

2. Each chosen seed is combined with its eight neighboring cells to create a *topological cluster*.

3. The topological clusters are grown by aggregating all the cells with at least a corner or a side in common with a cell already in a cluster. For a crystal to be included in a topological cluster, it must exceed another specified energy threshold ($E_{\mathrm{thr}}^{\mathrm{gather}}$).

One topological cluster may contain multiple seeds. That can potentially indicate that the energy deposits coming from different particles are superimposed in the cells. To

accurately reconstruct the kinematic properties of the original particles, it is necessary to correctly attribute the energy proportions. A dedicated algorithm is used for this purpose:

1. The algorithm goes through all topological clusters, which contain more than one seed. Each seed will give rise to a PFCluster.

   a) In the first iteration, the crystal center of each seed within the topological cluster is taken as the initial position approximation of each PFCluster.

   b) The expected energy fraction $f_{ji}$ measured in the cell at position $\vec{c_j}$ arising from the $i_{\text{th}}$ energy deposit ($i_{\text{th}}$ particle) is calculated according to Eq. (2.11).

$$f_{ji} = \frac{E_i \mathrm{e}^{-(\vec{c}_j - \vec{\mu}_i)^2/\left(2\sigma_c^2\right)}}{\sum_{k=1}^{N} E_k \mathrm{e}^{-(\vec{c}_j - \vec{\mu}_k)^2/(2\sigma_c^2)}}, \tag{2.11}$$

   where $N$ is the total number of seeds within the topological cluster, $E_i$ is the energy amplitude for $i_{\text{th}}$ PFCluster, evaluated from Eq. (2.12) (for the first estimation, a sum of all energy deposits in the topological cluster is taken), $\vec{\mu_i}$ is the estimated coordinates of the $i_{\text{th}}$ PFCluster, $\sigma_c$ is a Gaussian width (constant value for a specific calorimeter, estimated from simulation [59]), $\vec{c_j}$ is the position of the $j_{\text{th}}$ cell.

   c) The energy amplitudes ($E_i$) and PFCluster positions ($\vec{\mu}$) are updated according to Eqs. (2.12) and (2.13).

$$E_i = \sum_{j=1}^{M} f_{ji} A_j \tag{2.12}$$

$$\vec{\mu}_i = \frac{1}{E_i} \sum_{j=1}^{M} f_{ji} A_j \vec{c}_j, \tag{2.13}$$

   where $M$ is the total number of cells in a topological cluster, $A_j$ is the deposited energy in $j_{\text{th}}$ cell, $\vec{c_j}$ is the position of the $j_{\text{th}}$ cell.

2. Steps 1b and 1c are iteratively repeated with newly calculated parameters until convergence, which is achieved when the difference $\Delta R$ between updated coordinates and coordinates predicted at the previous step is $< 10^{-8}$.

3. The final coordinates are evaluated according to Eq. (2.14), where the position of the cells is additionally weighted by the logarithm of deposited energy to account for the longitudinal and transverse spread of an EM shower. The energy is calculated by summing all of the cells inside the topological cluster, taking into account defined energy fractions for each identified PFCluster.

$$\vec{\mu_i} = \frac{\sum_{j=1}^{M} w_j \vec{c_j}}{\sum_{j=1}^{M} w_j}, \qquad w_j = w_0 + \ln A_j - \ln \sum_{k=1}^{M} A_k, \tag{2.14}$$

   where $w_0$ is a parameter of the calorimeter, $A_i$ is deposited energy in $i_{\text{th}}$ cell, $\vec{c_i}$ is the position of the $i_{\text{th}}$ cell.

These steps are schematically shown in Fig. 2.13.

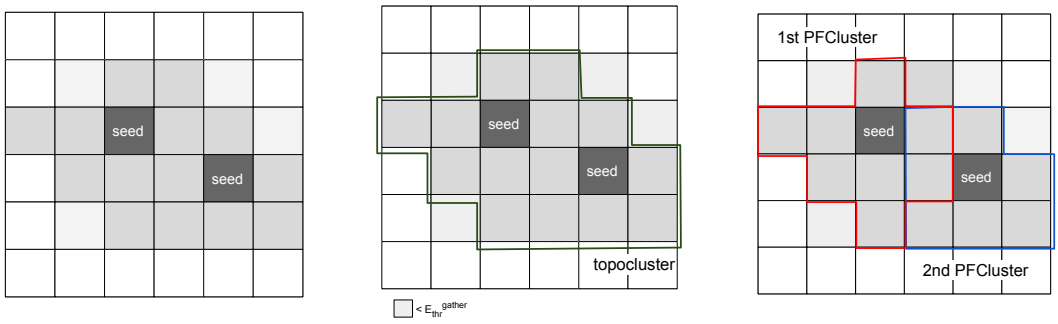



Figure 2.13: Schematic illustration of the PFClustering algorithm. a) The seeds are identified as local maximums with energy deposits $> E_{\text{thr}}^{\text{seed}}$. b) From the seeds, topological clusters are grown by combining the cells sharing a side or a corner with the crystals already in the clusters and passing the threshold $E_{\text{thr}}^{\text{gather}}$. c) Finally, PFClusters are identified by assigning the energy fractions.

### Parameters tuning for Run 3

The parameters of the PFClustering algorithm have been tuned in order to make the algorithm robust to the increased noise during Run 3 [57]. The tuning was performed on a simulated photon sample without including the tracker and pileup. The foreseen conditions of the detector for the accumulated luminosity of 450 fb$^{-1}$ were considered.

The obtained thresholds are $\eta$-dependent and for both $E_{\text{thr}}^{\text{rechit}}$ and $E_{\text{thr}}^{\text{seed}}$ equal to $3\sigma_n$ ($4\sigma_n$) for $|\eta| \leq 2.5$ ($|\eta| > 2.5$) while the $E_{\text{thr}}^{\text{gather}}$ has been eliminated. In order to maintain the resolution performance, the threshold parameters will be updated every year during the Run 3 operation to account for the evolution of the noise $\sigma_n$.

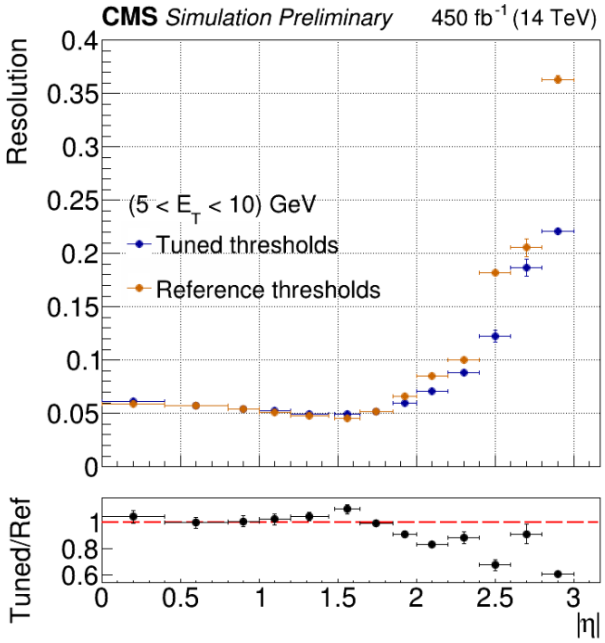



Figure 2.14: Performance of the tuning PFClustering algorithm [57].

The performance of the tuned PFClustering algorithm is presented in Fig. 2.14.

It shows the resolution, which is measured as the relative width of the ratio between reconstructed and simulated energy, extracted from a maximum likelihood fit to a double-sided crystal ball. No regression is applied to the reconstructed PFClusters. The performance is compared to the reference thresholds, which correspond to Run 2 operations.

The remaining parameters of the PFClustering algorithm are estimated from the simulation and equal to $\sigma_c = 1.5$ cm and $w_0 = 4.2$.

### 2.4.3 "Mustache" SuperClustering

Both electron and photon have a high probability to start showering before reaching the ECAL; the total thickness of the inner tracker material expressed in units of the radiation lengths is shown in Fig. 2.15. As a result, the original object may consist of several electrons and/or photons.

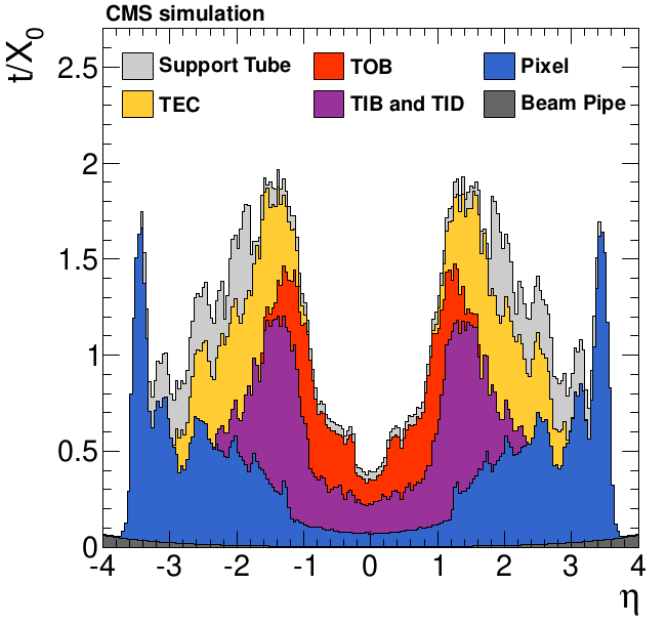



Figure 2.15: Total thickness $t$ of the inner tracker material expressed in units of radiation lengths $X_0$ as a function of pseudorapidity. The acronyms TIB, TID, TOB, and TEC stand for "tracker inner barrel", "tracker inner disks", "tracker outer barrel", and "tracker endcaps" respectively [62].

In order to account for photon conversion and bremsstrahlung losses, after all the PFClusters are formed, they have to be combined in a SuperCluster (SC). A traditional algorithm, called "Mustache", is used for this purpose and it follows several steps:

1. The list of the reconstructed PFClusters in ECAL is ordered by transverse energy $E_{\mathrm{T}}$.

2. At each iteration, a *seed PFCluster* is chosen as the one with the highest energy that is still not assigned to any SC. It also must pass the threshold of $E_{\mathrm{T}} > 1$ GeV. The *seed PFCluster* serves as the basis to grow an SC.

3. All the PFClusters that fall in a specified geometric area ("window") around the seed PFCluster are considered as a part of the corresponding SC. This area has a shape resembling a "mustache": due to the presence of the solenoidal magnetic field, the PFCluster spread is larger in the $\phi$ direction than in $\eta$. The size of the

mustache region depends on $E_{\mathrm{T}}$ of the seed PFCluster since the particles with higher $p_{\mathrm{T}}$ are less bent by the magnetic field. An example of a mustache SC is presented in Fig. 2.16 for simulated electrons with $1 < E_{\mathrm{T}}^{\mathrm{seed}} < 10$ GeV. In the case of photons, the shape is similar.

These steps are iteratively repeated until there are no more available PFClusters with $E_{\mathrm{T}} > 1$ GeV.

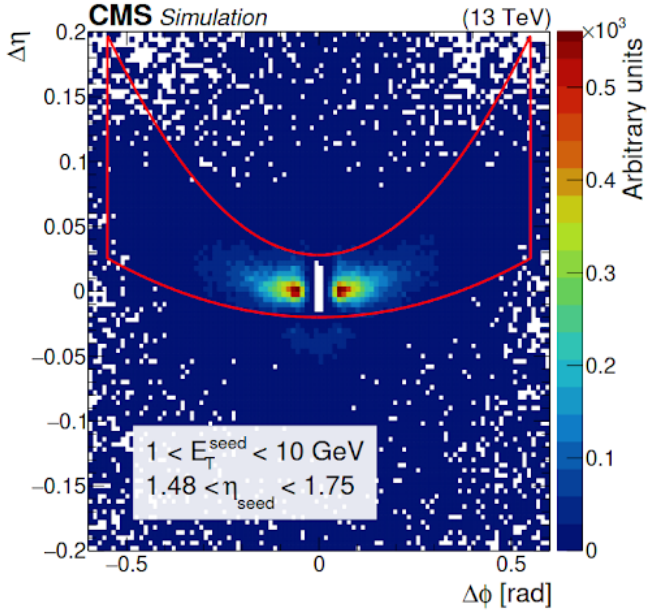



Figure 2.16: Distribution of $\Delta\eta = \eta_{\mathrm{seed-cluster}} - \eta_{\mathrm{cluster}}$ versus $\Delta\phi = \phi_{\mathrm{seed-cluster}} - \phi_{\mathrm{cluster}}$ for simulated electrons with $1 < E_T^{\mathrm{seed}} < 10$ GeV and $1.48 < \eta_{\mathrm{seed}} < 1.75$. The z-axis represents the occupancy of the number of clusters matched with the simulation (requiring to share at least 1% of the simulated electron energy) around the seed. The red line contains approximately the set of clusters selected by the mustache algorithm [60].

The main advantage of this algorithm is its very high efficiency as it is able to capture even low-energy clusters. However, there are multiple effects that degrade its performance in terms of energy reconstruction:

- Energy lost before reaching the ECAL and in detector gaps;
- Energy leakage out of the back of the ECAL and in dead channels;
- The use of finite energy thresholds to suppress noise in the detector electronics;
- Energy deposited by pileup interactions.

To mitigate the effect of these issues, a multivariate regression technique is used to define an energy correction. It is described in detail in the following section.

### Parameters tuning for Run 3

The shapes of the parabolas for the "Mustache" algorithms are parameterized with 10 numbers. The optimization of these parameters for Run 3 operation was done such that the area of the mustache encompasses at least 98% of the true EM energy. The

evaluation was performed on a photon sample simulated with pileup and with detector conditions corresponding to a total luminosity of 180 fb$^{-1}$.

The obtained fit is shown in Fig. 2.17 along with the results previously used for Run 2.

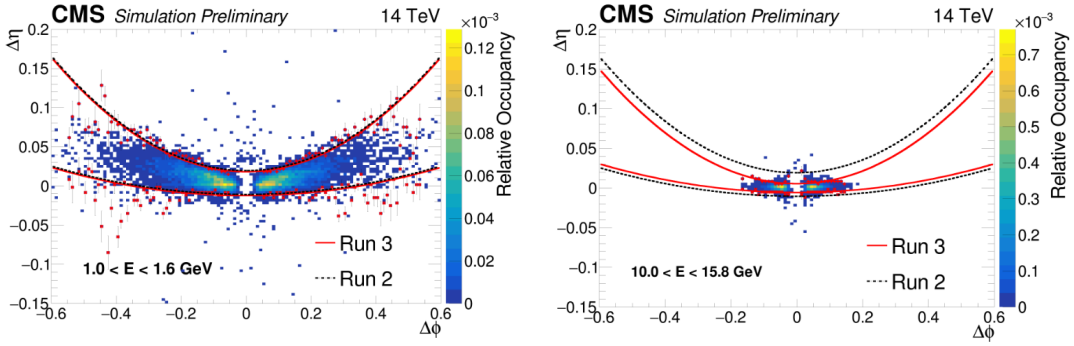



Figure 2.17: Mustache parameters fit for Run 3 operations compared to the ones used during Run 2 [57].

### 2.4.4 Energy regression

The aim of energy regression is to account for the energy losses in the ECAL and achieve the best prediction of the true energy. In order to do it, a correction factor denoted as $y$ is estimated and further applied to the energy predicted by the SuperClustering algorithm. It is calculated by dividing the true energy ($E_{\text{true}}$) by the reconstructed energy ($E_{\text{reco}}$) as shown in Eq. (2.15).

$$y = \frac{E_{\text{true}}}{E_{\text{reco}}} \tag{2.15}$$

The estimation of the energy correction for an SC is done through a machine learning algorithm called Boosted Decision Tree (BDT), which is discussed in Chapter 4. The implemented method not only predicts the required $y$ coefficient but also estimates the energy resolution of individual objects. Thirty input variables are used for regression, as listed in Table 2.1. These variables include the number of PFClusters within the SC, the energy reconstructed by the "Mustache" algorithm, the SC's width in both the $\eta$ and $\phi$ directions, and various other parameters related to the energy deposits of crystals within the seed PFCluster. The input vector is further denoted as $\vec{x}$.

During the training phase, the BDT algorithm splits the data samples into various final leaves, and the energy correction distributions of these leaves are fitted to a probability density function represented by a double-sided Crystal Ball (DSCB). It has a Gaussian core with power law tails on both sides, its definition is presented in Eq. (2.16). The fitting process allows the algorithm to estimate the energy resolution for each object by analyzing the distribution of energy corrections for the corresponding leaf.

$$\begin{gathered}\text{DSCB}(y;\mu,\sigma,\alpha_{\text{L}},n_{\text{L}},\alpha_{\text{R}},n_{\text{R}}) = \text{DSCB}, \\ \text{DSCB} = \begin{cases} N\mathrm{e}^{-\frac{\xi(y)^2}{2}}, & \text{if } -\alpha_{\text{L}} \leq \xi(y) \leq \alpha_{\text{R}} \\ N\mathrm{e}^{-\frac{a_{\text{L}}^2}{2}}\left(\frac{\alpha_{\text{L}}}{n_{\text{L}}}\left(\frac{n_{\text{L}}}{\alpha_{\text{L}}} - \alpha_{\text{L}} - \xi(y)\right)\right)^{-n_{\text{L}}}, & \text{if } \xi(y) < -\alpha_{\text{L}} \\ N\mathrm{e}^{-\frac{u_{\text{R}}^2}{2}}\left(\frac{\alpha_{\text{R}}}{n_{\text{R}}}\left(\frac{n_{\text{R}}}{\alpha_{\text{R}}} - \alpha_{\text{R}} + \xi(y)\right)\right)^{-n_{\text{R}}}, & \text{if } \xi(y) > \alpha_{\text{R}} \end{cases}\end{gathered} \tag{2.16}$$

| Parameter | Description |
|---|---|
| $n_{\mathrm{cl}}$ | Number of PFClusters inside the SuperCluster. |
| $E_{\mathrm{reco}}$ | Energy reconstructed by the "Mustache" algorithm. |
| $\eta_{\mathrm{width}}, \phi_{\mathrm{width}}$ | Width in $\eta$ and $\phi$ of the SuperCluster. |
| $E_{3x3}$ | The sum of energies in 3x3 matrix around the most energetic crystal in the seed PFCluster. |
| $E_{\mathrm{seed}}$ | Energy of the seed PFCluster. |
| $E_{\mathrm{max}}$, $E_{\mathrm{2nd}}$ | Largest and second largest energy deposit in a crystal within the seed PFCluster. |
| $E_{\mathrm{LR}}$ | The energy deposit difference between the left and the right crystals in relation to the highest energy crystal in the seed PFCluster. |
| $E_{\mathrm{TB}}$ | The energy deposit difference between the top and the bottom crystals in relation to the highest energy crystal in the seed PFCluster. |
| $cov_{\mathrm{i}\eta\mathrm{i}\eta}$, $cov_{\mathrm{i}\eta\mathrm{i}\phi}$, $cov_{\mathrm{i}\phi\mathrm{i}\phi}$ | Covariance values between the PFRechits spread in the SuperCluster in different directions. The covariance for two variables X and Y, each with sample size N is defined as $\mathrm{cov}(\mathrm{X},\mathrm{Y}) = \sum_{i=1}^{N} \frac{(\mathrm{x}_i - \bar{x})(\mathrm{y}_i - \bar{y})}{N}$, where $\bar{x}$ and $\bar{y}$ are the mean values of the variables. |
| $n_{\mathrm{vertices}}$ | Number of vertices. |
| $\Delta R^{\mathrm{max}}_{\mathrm{PFcluster}}$ | The maximum distance between the seed PFCluster and the PFClusters in the respective SuperCluster. |
| $\Delta\phi^{\mathrm{maxDR}}_{\mathrm{PFCluster}}$, $\Delta\eta^{\mathrm{maxDR}}_{\mathrm{cluster}}$ | The maximum distance in $\phi$ and $\eta$ between the seed PFCluster and the PFClusters in the respective SuperCluster. |
| $E^{\mathrm{maxDR}}_{\mathrm{PFCluster}}$ | The reconstructed energy of the PFCluster with the maximum distance from the seed PFCluster. |
| $E^{1,2,3}_{\mathrm{PFCluster}}$ | Three highest reconstructed energies of the PFClusters in the SuperCluster. |
| $\Delta\phi^{1,2,3}_{\mathrm{PFCluster}}$ | The $\phi$-position difference between the seed PFCluster and the three PFClusters in the SuperCluster with the highest reconstructed energies. |
| $\Delta\eta^{1,2,3}_{\mathrm{PFCluster}}$ | The $\eta$-position difference between the seed PFCluster and the three clusters in the SuperCluster with the highest reconstructed energies. |
| $i\eta_{\mathrm{seed}}$, $i\phi_{\mathrm{seed}}$, $\eta_{\mathrm{seed}}$ | $i\eta$, $i\phi$, $\eta$ – positions of the seed PFCluster. |

Table 2.1: The regression input variables for the SuperClustering energy correction.

where $N$ is the normalization constant, $\xi(y) = (y - \mu)/\sigma$, the variables $\mu$ and $\sigma$ are the parameters of the Gaussian core, and the $\alpha_{\mathrm{R}}$ ($\alpha_{\mathrm{L}}$) and $\mathrm{n_R}$ ($\mathrm{n_L}$) are parameters that control the right (left) tails of the function.

For each SC, $\mu$ and $\sigma$ are predicted by the regression algorithm, where $\mu$ is the estimate of the energy correction coefficient and $\sigma$ is the estimate of the per-object energy resolution that includes the effect from imperfect crystal calibrations. Both $\mu$ and $\sigma$ are predicted as the functions of the input vector $\vec{x}$.

The BDT model is optimized for photons and electrons. Moreover, in the case of electrons, an additional step is performed to combine the information from the SC and the track, resulting in the final energy correction estimation. A weighted combination of the two independent measurements is:

$$E^{\text{reco}}_{\text{combined}} = \frac{E_{\text{ECAL}}/\sigma_E^2 + p_{\text{tracker}}/\sigma_p^2}{1/\sigma_E^2 + 1/\sigma_p^2}, \tag{2.17}$$

where $E_{\text{ECAL}}$ and $\sigma_E$ are the ECAL measurements of the energy and the energy resolution of the SC of the electron corrected with the step 1 and 2 regressions, respectively, and $p_{\text{tracker}}$ with $\sigma_p$ are the momentum magnitude and momentum resolution measured by the electron tracking algorithm. This step improves the energy of the electrons at low $E_{\text{T}}$. For the particles with $E_{\text{T}} > 200$ GeV, only SC energy is used.

The effect of different regression steps for electrons is shown in Fig. 2.18, where the ratio of the most probable true (or generated) energy to reconstructed energy $E_{\text{gen}}/E_{\text{reco}}$ for different $p_T$ bins are presented.

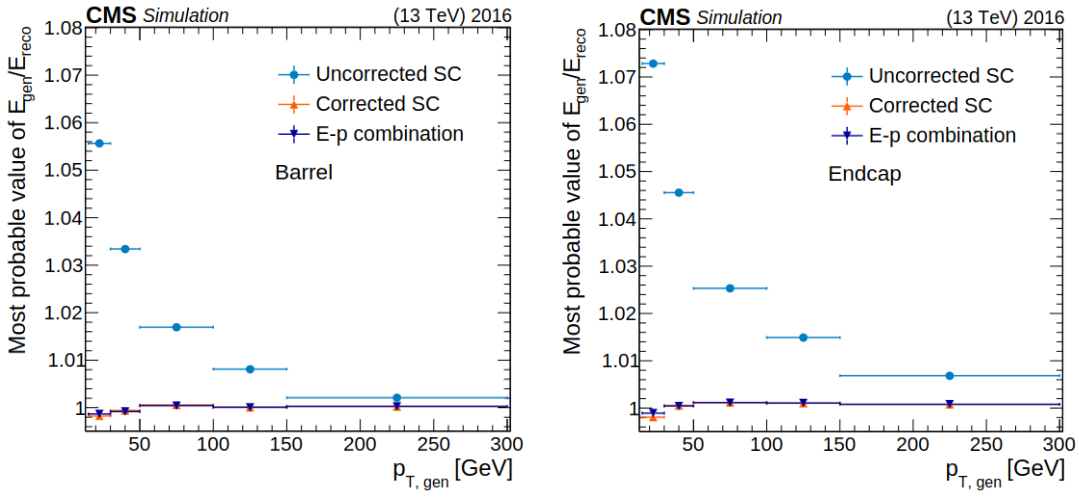



Figure 2.18: Most probable value of the ratio of true to reconstructed electron energy, as a function of electron $p_T$ with and without the regression corrections, evaluated using 2016 Monte-Carlo samples for barrel (left) and endcap (right) [59].

### 2.4.5 Electromagnetic object selection in particle flow

It is crucial to correctly separate prompt electrons and photons from hadrons and non-isolated $e/\gamma$, which form jets. The misidentified objects can potentially degrade the energy resolution of the reconstructed jets.

The primary sources of background for photons and electrons consist of the following:

- For prompt electrons, the background can originate from photon conversions, hadrons misidentified as electrons, and secondary electrons from semileptonic decays of $b$ or $c$ quarks.
- For prompt photons, the most important background arises from jets fragmenting into light mesons ($\pi^0$, $\eta$), which decay into two photons. For the high-energy mesons, these secondary photons will be nearly collinear and, as a result, hard to distinguish from a single photon in the calorimeter.

In the particle flow reconstruction, a loose identification selection is used for electrons and photons. The algorithms use various inputs including:

- Kinematic variables.
- Shower-shape variables.
- Energy deposits in the sub-detectors.
- Isolation energy sums. They are constructed from the sum of the reconstructed energy in a cone ($\Delta R = \sqrt{\Delta\phi^2 + \Delta\eta^2} = 0.3$) around electrons and photons in ECAL and HCAL detectors.
- Hadronic over electromagnetic ratio. It is defined as the ratio between the energy deposited in the HCAL in a coned with $\Delta R = 0.15$ and the energy of EM object.
- Tracker variable, indicating whether a hit is present in the innermost layer of the tracker.

For photons, a dedicated Dense Neural Network (DNN) is implemented while electrons use a BDT. The basic elements of the objects that did not pass the selection criteria are released and further considered in the formation of jets.

## 2.5 HL-LHC upgrade and MIP Timing Detector

The main goal of the HL-LHC upgrade is to significantly increase the amount of gathered data in order to facilitate new physics searches, Higgs boson coupling measurements, and other precision tests of the Standard Model (SM). The target is to achieve an integrated luminosity of 3000 fb$^{-1}$ over a period of 10 years, which is about 10 times larger compared to the current data-taking operations.

The high-luminosity collider will present unprecedented challenges for the detectors due to the increased levels of radiation and a higher pileup. Absorbed dose in the CMS cavern after an integrated luminosity of 3000 fb$^{-1}$ is presented in Fig. 2.19 and pileup conditions are further discussed in Section 2.5.2. To effectively address both of these issues, the CMS sub-systems will undergo major upgrades known as *phase II upgrades*. The primary objective is to maintain the excellent performance of the CMS detector in terms of efficiency, resolution, and background rejection in a high-luminosity environment. The installation of the upgraded systems started during LS2 (phase I upgrades) and will be completed during LS3.

### 2.5.1 Overview of the CMS phase II upgrades

To understand the effects of the harsh operating environment of HL-LHC on the detectors and to outline the required upgrades, CMS has made a major effort in simulations. In order to benchmark these simulations, the performance of the current detectors under irradiation was estimated from test beam measurements and previously observed radiation damages. In this section, a brief overview of the selected envisioned updates in the CMS experiment is presented.

#### Tracker system

As the tracker is the closest detector to the interaction point, it will suffer major radiation damage and it must be completely replaced during phase II upgrades [64]. Moreover, the new detector is required to have a very high resolution to be able to correctly associate the produced tracks to their vertices in order to handle the increased levels of pileup. It will consist of two sub-detectors: the Outer Tracker (OT) made from silicon modules and the Inner Tracker (IT) comprising silicon pixel modules. To maintain the track

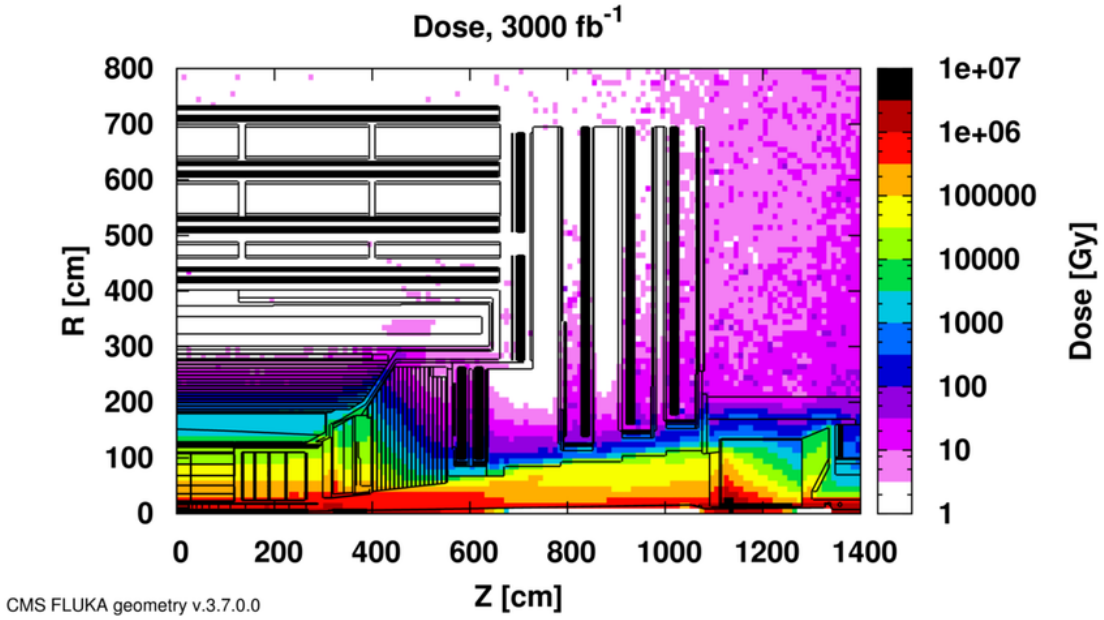



Figure 2.19: Absorbed dose in the CMS cavern after an integrated luminosity of 3000 fb$^{-1}$. $R$ is the transverse distance from the beamline and $Z$ is the distance along the beamline from the Interaction Point at $Z$=0 [63].

reconstruction performance in the new conditions, the granularities of the outer tracker and the pixel detector will be respectively increased by a factor of 4 and 6 compared to the current operations. For the outer tracker, it will be done by shortening the length of the silicon sensor strips, while for the pixel detector by using smaller pixels and thinner sensors. The new system will extend to the forward region and will be able to reconstruct the tracks up to approximately $|\eta| = 4$.

### Calorimeter Barrel

In the ECAL Barrel (EB), the front-end electronics will be replaced, and for both ECAL and HCAL Barrels (HB), the off-detector electronics will be changed as well [65]. It is done in order to accommodate new Level-1 trigger requirements on increased latency (from about 4 $\mu$s to a maximum of 12.5 $\mu$s) and data rate, provide timing measurements with much better precision (from a resolution of about 120 ps currently to a resolution of less than 50 ps), and help mitigate the increasing noise from the photodetectors.

### New Endcap Calorimeter

Electromagnetic and hadronic endcaps will be significantly damaged by radiation during the current operations as they are placed in the forward regions [66]. Consequently, both of them will be replaced by a new detector called the High-Granularity Calorimeter (HGCAL) or a new endcap calorimeter (EC). It will cover the pseudorapidity region of $1.5 < \eta < 3$ and will provide excellent transverse and longitudinal segmentation.

The electromagnetic part (CE-E) of the HGCAL occupies 25 $X_0$ and consists of 26 layers of Cu/CuW/Pb absorbers interleaved with silicon sensors used as an active material. The hadronic part has 21 layers of steel absorbers interleaved with silicon sensors in the more demanding radiation regions and scintillating tiles in the outer regions. The schematic illustration of HGCAL is shown in Fig. 2.20.

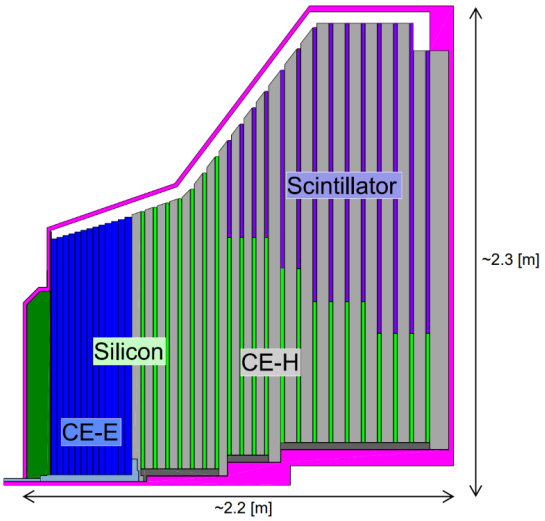



Figure 2.20: Schematic illustration of the High-Granularity Calorimeter (HGCAL) [67].

### General

Extensive upgrades are also performed for the electronics of the detectors, as well as the trigger, data acquisition, and luminosity measurement systems.

Moreover, an additional detector, called Minimum Ionizing Particle (MIP) Timing Detector (MTD), is being developed for the HL-LHC phase. As its name suggests it will enable the precise timing measurements of MIPs. More details regarding this new detector are presented in the following section.

## 2.5.2 Precision timing and MIP Timing Detector

As previously mentioned, the significant increase in pileup during the HL-LHC era poses a major challenge for the detectors. The hard interactions of interest to CMS, which are dedicated to probing physics at the energies from a few GeV to several TeV, will be accompanied by an average of 140-200 additional interactions [68]. The tracks and energy deposits coming from these "soft" collisions will degrade the identification and reconstruction of the interactions of interest.

The quality of the reconstruction of the final physics objects can be enhanced by discarding the charged tracks and energy deposits in calorimeters that do not originate from the primary vertex of interest (associated with the hard collision). Currently, the procedure relies on spatial vertex separation and it will be further adapted to HL-LHC by using a new high granularity tracker system.

However, the tracks in the detector can also arise from displaced sources, such as secondary interactions, decays of particles in flight, and resolution tails. Simulation studies indicate that a relatively large spatial window of 1 mm needs to be considered to achieve optimal reconstruction, resulting in substantial contamination of tracks from pileup into the primary vertex. This is demonstrated by the line density, which represents the number of collision vertices per mm, shown in Fig. 2.21.

To mitigate this effect and attain the level of vertex purity achieved during Run 2, timing measurements can be added to the reconstruction process. It is beneficial due to the fact that the individual collisions in one bunch crossing do not happen simultaneously as the beam has a longitudinal spread. They rather occur in the time interval with a root mean square (RMS) extent of 180-200 ps within a 25 ns bunch crossing. By introducing the fourth dimension, time, into the vertex reconstruction, the procedure

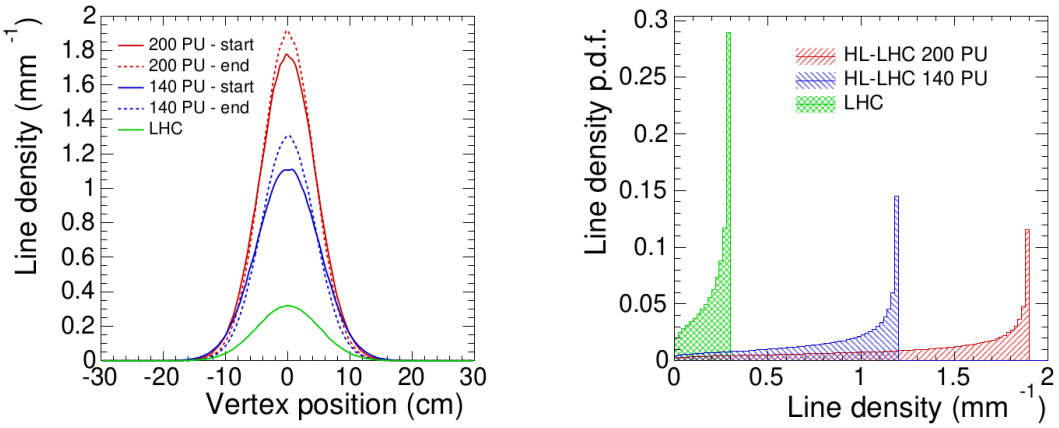



Figure 2.21: Left: Distribution of the vertices along the beam direction at the LHC (Run 1 and early Run 2) with ≈30 pileup interactions and HL-LHC with 140 and 200 pileup interactions. The solid (dashed) line refers to the start (end) of the fill. Adjustments in the focusing of the beam cause the $z$ distributions to become narrower at the end of the fill. Right: Probability density functions of the line density along the beam axis for the pileup of about 30 and for pileup 140 and 200. The modes of the three distributions are 0.3, 1.2, and 1.9 mm$^{-1}$ and their means are 0.2, 0.9, and 1.4 mm$^{-1}$, respectively [68].

can be significantly improved. This is illustrated in the event display in Fig. 2.22, which presents simulated and reconstructed vertices in a bunch crossing with 200 pileup interactions. According to simulation, instances of vertex merging are reduced from 15% in space to 1% in space-time.

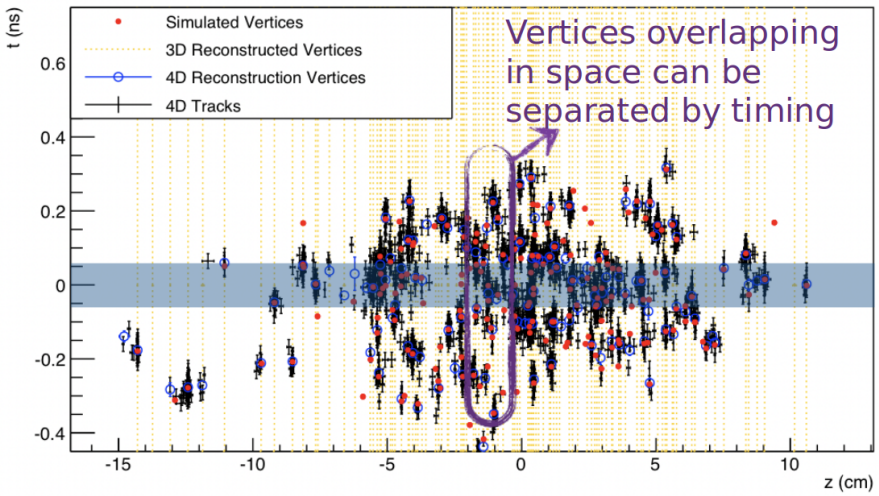



Figure 2.22: Simulated and reconstructed vertices in a bunch crossing with 200 pileup interactions assuming an MTD detector with ≈30 ps timing resolution [68].

In the phase II upgrade of the CMS, the timing measurement for neutral particles will be incorporated in calorimeters while for charged particles a new MIP Timing Detector (MTD) is being developed. The upgraded barrel parts of the calorimeters, as well as the new HGCAL detector, will be able to measure the time of arrival of the particles with a resolution of the order of 30-50 ps. It will allow better pileup mitigation by associating the particle showers with their primary vertices not only based on the spatial coordinates but also using the timing information.

The MTD will help to reconstruct the time at which the collision vertex occurred by associating tracks from a vertex to hits and their corresponding times in the detector. This process allows to eliminate other tracks that point approximately towards the vertex but arrive at the wrong time, effectively disregarding their contribution to that specific collision. As the name indicates, the new detector will be able to perform precise time of arrival measurements even for minimum ionizing particles (MIPs), which are challenging to reconstruct due to their low energy loss in the material. The goal is to achieve a timing resolution of 30-40 ps at the start of HL-LHC, which will slowly degrade to 50-60 ps by the end of operations due to the radiation damage. The reduction in the pileup using the MTD detector is quantified in Fig. 2.23.

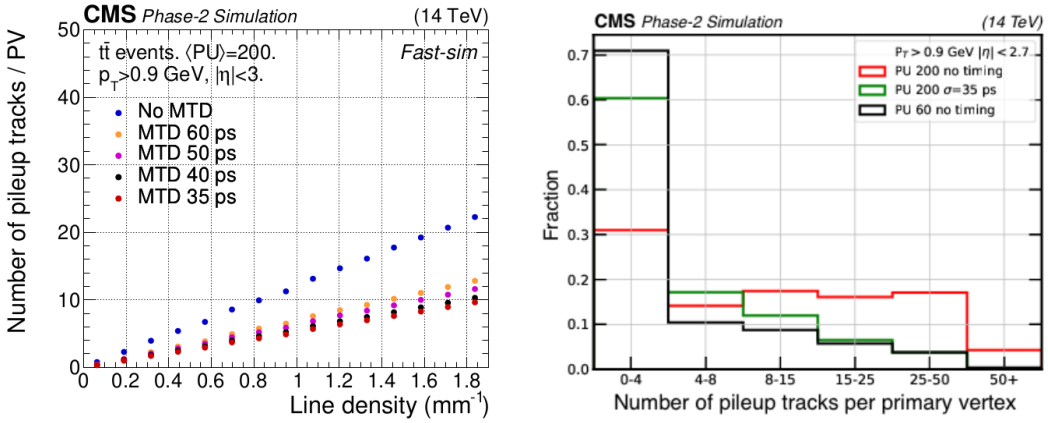



Figure 2.23: Left: number of pileup tracks incorrectly associated with the hard interaction vertex as a function of the collision line density for different time resolutions. Right: distribution of the number of incorrectly associated tracks [68].

The removal of pileup tracks significantly enhances the reconstruction of the final objects. Notably, it improves the identification efficiency of isolated leptons and photons by eliminating additional tracks within the isolation cones. The performance of b-jet identification, which relies on vertex reconstruction, is enhanced as well. Moreover, the reconstruction of other jets and missing transverse momentum is also improved. In more detail, the impact of the MTD on the physics program in the CMS experiment is discussed in Ref. [68].

In addition to maintaining the data quality, the MTD brings new capabilities to the CMS experiment:

- Time-of-flight (TOF) measurements between the collisions and particle arrival to the detector enable the search for long-lived particles (LLPs) by estimating TOF-based particle ID.
- The timing information can also be used to differentiate between low-momentum charged hadrons, such as pions, kaons, or protons, opening up possibilities for unique flavour physics studies in heavy-ion collisions.

### MIP Timing detector

The MTD is a hermetic detector that will be placed between the tracker and the calorimeters. It is divided into barrel and endcap regions. The schematic layout of the MTD is shown in Fig. 2.24.

- The **Barrel Timing Layer (BTL)** is a thin cylindrical detector that will cover the pseudorapidity range of $|\eta| < 1.48$. It is made from LYSO:Ce scintillating

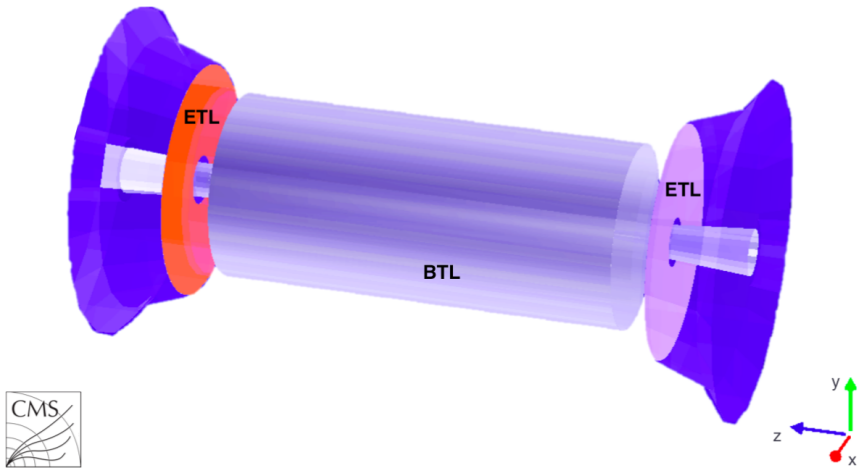



Figure 2.24: A schematic layout of the MIP Timing Detector created in Geant4 [68].

crystals that are shaped in bars of about 5.7 cm in length ($\phi$ direction) and a width of 3 mm ($z$ direction). The thickness varies along the barrel length with 3.7 mm for $|\eta| < 0.7$; 3.0 mm for $0.7 \leq |\eta| \leq 1.1$; and 2.4 mm for $|\eta| > 1.1$ to maintain the same slant depth crossed by particles coming from the primary vertex. Each crystal is accompanied by two silicon photomultipliers (SiPMs) that provide the readout of both ends of the bar. This is done in order to improve the resolution by eliminating the time delay coming from the light traveling along the crystal.

- The **Endcap Timing Layer (ETL)** is a two-disk system that will be situated between the tracker and the HGCAL. It will cover a pseudorapidity range of $1.48 < |\eta| < 3.0$. For the readout, MIP-sensitive silicon devices are used, called Low-Gain Avalanche Detectors (LGADs), as the SiPMs do not have sufficient radiation tolerance for most of the pseudorapidity range.

Further details on the data acquisition system of the MTD are presented in Chapter 3.

# 3 Data Acquisition Software for the MIP Timing Detector

A new MIP timing detector (MTD) is planned to be included in the CMS experiment as part of the phase II upgrades. It is designed to measure the time of arrival of minimum ionizing particles (MIPs), providing a new dimension (time) for disentangling pileup interactions. The detector also enables unique opportunities for Long-Lived Particle (LLP) searches and flavor physics in heavy-ion collisions.

The MTD aims to achieve the resolution of 30 ps at the start of the HL-LHC, slowly degrading to 60 ps at the end of operation due to the aging of the detector. It consists of two parts: Barrel Timing Layer (BTL) – thin cylindrical layer between the tracker and electromagnetic calorimeter (ECAL) covering the pseudorapidity range of $|\eta| < 1.48$ and an Endcap Timing Layer (ETL) – two-disk system between the tracker and high-granularity calorimeter (HGCAL) with the coverage up to $|\eta| = 3.0$. More details on the HL-LHC, phase II upgrades, and MTD can be found in Section 2.5.

The BTL has a surface area of 38 $m^2$ and is composed of bar-shaped LYSO:Ce scintillating crystals paired with silicon photomultipliers (SiPMs) at both ends. In the case of the ETL, each disk has a sensitive area of 7.9 $m^2$. As the ETL is exposed to significantly higher radiation doses compared to the BTL, the Low Gain Avalanche Detectors (LGAD) are chosen as the sensitive element because they are able to withstand this harsh environment.

For each of the layers, a data acquisition (DAQ) system is employed that collects signals from sensors, reconstructs timing information, and sends data to build final events. It consists of multiple radiation-tolerant electronic components in the front-end, and electronic boards based on Field Programmable Gate Arrays (FPGAs) in the back-end of the detector. Along with the hardware components, a dedicated DAQ software is currently being developed to enable efficient data handling and event reconstruction.

In this chapter, the DAQ software for the MTD is presented. In Section 3.1 overview of the hardware components of the MTD DAQ system is given. Afterward, the MTD DAQ software is discussed in Section 3.2. Only components related to the BTL layer are presented as the ETL software is still in the early development stage. Finally, Section 3.3 discusses the system tests performed with the described BTL DAQ hardware and software.

## 3.1 An overview of the MTD DAQ system

For the BTL, a fundamental detecting element is a scintillating crystal bar with two attached SiPMs. When a MIP traverses the crystal it produces a number of optical photons along its track. This number is proportional to the crystal light yield (number of photons generated per MeV of energy deposit). A fraction of the photons is detected at each SiPMs that converts them to photoelectrons and further amplifies the electrical

signal. From this signal, a measurement of the time at which the MIP crossed the detector, called a *time stamp*, can be obtained. The time stamp is produced for each SiPMs and to achieve the correct estimation of the time of arrival, an average value is taken: $t_{\text{ave}} = (t_{\text{left}} + t_{\text{right}})/2$.

The signal is read out by the front-end electronics and sent to the back-end, where the received data is processed and further passed to the event builder.

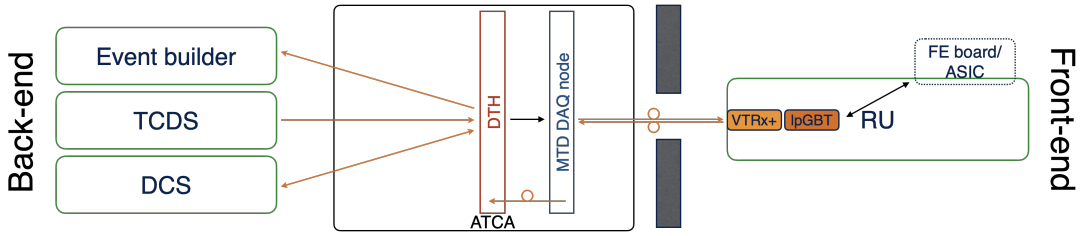



Figure 3.1: The data acquisition system (DAQ) of the MTD. The key part of the front-end electronic is the TOFHIR chip (for BTL) or ETROC chip (for ETL) that read out data from the sensors. The chips are organized into Readout Units (RU). The data from the chips is passed to the back-end through two low-power Giga-Bit transceivers (lpGBT) associated to two Versatile Link Plus (VTRx+). The information from the back-end, such as trigger signals, and fast and slow control signals from the trigger control distribution system (TCDS2), and detector control system (DCS), is transmitted to the front-end through lpGBTs as well. The back-end electronics infrastructure is based on advanced telecommunications architecture (ATCA). One central "hub" board (DTH) is connected to all the "node" boards (MTD DAQ node), each hosting two Field Programmable Gate Arrays (FPGAs). They process the acquired data and further sent it to the event builder.

The basic building block of MTD front-end electronics is called Readout Unit (RU). In the case of the BTL, one RU consists of 12 Front-End (FE) cards, each hosting 2 readout chips, called TOFHIR (Time-of-flight, High rate). One chip can read out signals from 32 SiPMs. The FE cards are connected to the back-end system through two low-power Giga-Bit transceivers (lpGBT) [69], placed on the Concentrator Card (CC).

For the ETL, LGADs are grouped in arrays of 16x32 sensors. Each module is read out by two ASICs, called Endcap Timing Readout Chips (ETROCs). A dedicated on-detector board called a service hybrid, provides power and readout services to the modules.

The back-end DAQ boards, hosting two FPGAs, are connected to the CCs (for BTL) or to service hybrids (for ETL) and are used to process the data and reconstruct the timing measurements. The schematic illustration of the DAQ system from front-end to back-end is shown in Fig. 3.1.

### 3.1.1 Front-end electronics

The key components of the front-end electronics are the TOFHIR readout chip for the BTL and ETROC readout chip for the ETL. Their main purpose is to perform digitization of passing MIPs with the required precision. The design of the TOFHIR chip is derived from the one used in the TOF-PET applications [70] and further adapted to the high signal rate and harsh radiation environment of HL-LHC.

The readout is performed upon receiving the trigger signal from other CMS subsystems. The front-end electronics is designed to receive these signals and transmit the corresponding data from each readout chip to the back-end systems through high-speed

optical links. The sampling clock with 40 MHz frequency is distributed to the readout system in order to synchronize it with the LHC bunch frequency.

The data from readout chips is transmitted to the back-end via Versatile Links+ (VL+), consisting of radiation-tolerant multi-gigabit communication ASICs (lpGBT) and radiation-tolerant optical transceivers (VTRx+ [71]) capable of handling the data rate. Each lpGBT handles communication with 24 TOFHIR chips in the case of the BTL or ETROC chips in the case of the ETL. The bidirectional links are chosen as the lpGBTs must be also able to transmit information, such as trigger signals, and fast and slow control signals from the trigger control distribution system (TCDS2), and detector control system (DCS), to the front-end. Additionally, slow monitoring signals such as temperature and voltage readings are collected from the on-detector electronics with GBT-SCA chips (two on each CC).

### 3.1.2 Back-end electronics

The back-end electronics infrastructure is based on the advanced telecommunications architecture (ATCA), centrally used in the CMS experimnet. The ATCA create can host two central "hub" boards, connected to the "node" boards. One of the two central hub slots is not used and is reserved for future development possibilities. Another one is occupied by the data trigger hub (DTH400) board, which provides an interface to the central TCDS and DAQ systems. A single DTH400 can transmit data at a rate of 400 Gb/s.

The MTD DAQ boards are placed in the "node" slots. They provide the unpacking and processing of the data received from the front-end. For the MTD detector, a board called Serenity [72] is chosen, which is developed within the CMS collaboration to handle the increased data rates as part of the phase II upgrades. Serenity consists of three elements: an ATCA-carrier card that provides common board services (power, clocking, electrical interconnections, etc.), daughter cards that host data-processing elements (FPGAs), and a framework of generic, flexible firmware and software. Serenity is capable of carrying two high-speed, high-capacity FPGAs, enabling real-time data reconstruction of the timing information, and 144 bidirectional links.

## 3.2 DAQ software

Dedicated DAQ software for MTD is currently being developed in order to efficiently operate with the system described in Section 3.1. At the time of the presented document, only the BTL electronics prototype components are advanced enough to be included in the software framework.

### 3.2.1 TOFHIR

#### Overview

The TOFHIR ASIC (application-specific integrated circuit) must be capable of performing the digitization of time and energy of the passing MIPs with the required precision and at the required rate. It is designed to operate in the radiation environment up to 10 Mrad. The chip also features a test pulse injector, which can be used to test and calibrate the TOFHIR.

Following the ionization originating from a passing charged particle, photons arrive in time following an exponential distribution with a decay time of 40 ns. The measurement of the arrival time of the MIP signals is performed based on the rising edge of the resulting photoelectron signal.

The MIPs are defined as particles with an average energy deposit $E > 1$ MeV, which are randomly distributed over the bunch crossing. All particles depositing energy $E < 1$ MeV are considered to be background and should be rejected. In order to do so, a minimum signal strength required to be classified as a MIP, called MIP threshold, is set (*trigger T2*).

However, the most precise timing measurements are provided by the arrival of the first photon. Another threshold, called timing threshold (*trigger T1*), is set that determines the minimum signal strength required to trigger the timing measurement. It is typically below the MIP threshold and ranges between 5 and 50 photoelectrons.

If the signal is identified as a MIP, the amplitude and two times (T1 and T2) corresponding to the timing threshold and MIP threshold are recorded as a hit. The processing of the input signals is performed within 25 ns, such that the chip is able to detect particles from different bunch crossings.

**Output data**

TOFHIR chip consists of 32 channels, each containing independent amplifiers, discriminators, time-to-digital (TDC) converters, and charge-to-digital (QDC) converters. The simplified description of obtaining the data from TOFHIR chips is presented in this section. A more detailed overview can be found in [68].

In the input stage of the chip, the current signal ($I_{IN}$) from the SiPMs sensors is transformed into low impedance input $R_{IN}$ and replicated into 3 branches: $T$, $E$, and $Q$.

The signals from the $T$ and $E$ branches are passed through two different TDCs, providing the timing measurements. The signal from $Q$ branch is passed to a QDC that measures the integrated charge using the trigger $Q$ (obtained from the input signal information).

The TDC provides two values:

- A coarse counter, counting the number of cycles of the external 160 MHz reference clock (LHC clock).
- A fine counter, providing fine timing measurements within one period of the reference clock.

1. *T1 measurement (branch T)* The fine time measurements are evaluated as follows: on the rising edge of trigger T1 it starts collecting the current $\mathrm{I_{TAC}}$. The integration continues until a falling edge and a rising edge of the reference clock. The stored voltage is digitized as t1fine. The coarse time estimation t1coarse is provided by the global counter associated with the reference clock.

   In the first approximation, the timing measurement T1 can be calculated as $\tau_1 =$ t1coarse - $\frac{\mathrm{t1fine}}{\mathrm{I_{TAC}}}$.

2. *T2 measurement (branch E).* The T2 measurements are performed identically to T1 but using trigger T2.

3. *Q measurement (branch Q).* The signal integration is triggered by trigger Q and closes either when trigger B (defines the end of the event) is zero and the minimum integration time has been met or when the maximum integration time has been reached.

A schematic illustration of the integration operations is shown in Fig. 3.2.

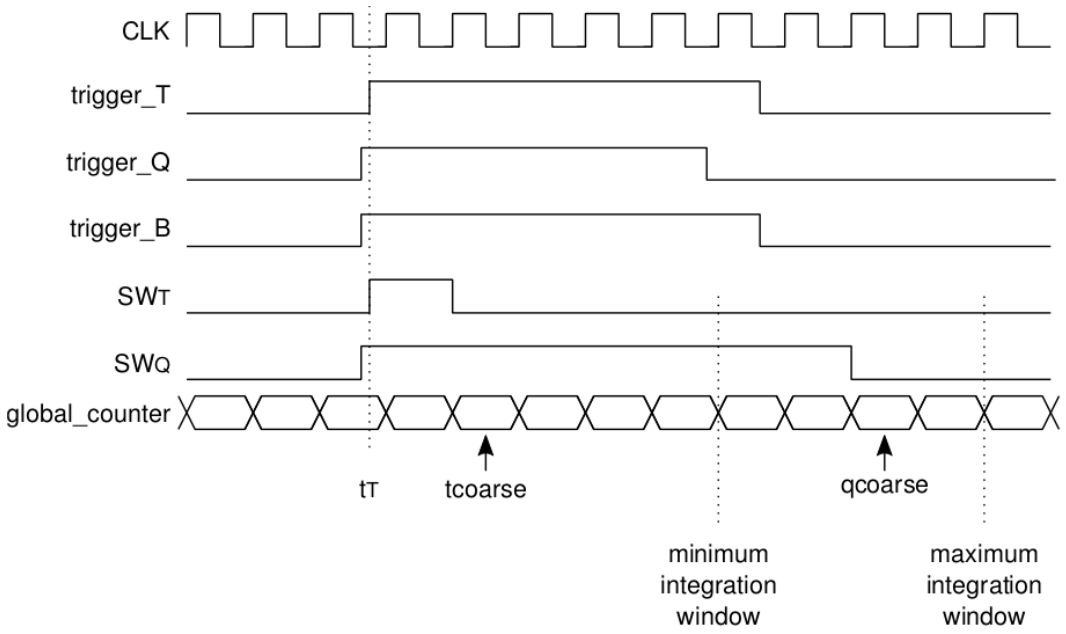



Figure 3.2: TAC and QDC operation.

After the full procedure is performed, three measurements are obtained: time of arrival (TOA) from T1; time over threshold (TOT), or the width of the pulse from T2 - T1; and energy of the particle from Q.

The calibration of the signals is performed using a test pulse and an alternative DAQ system called FEB-D [73] in order to obtain the final physics measurements: time of arrival (TOA), time over threshold (TOT), and energy of the particles.

The TOT is used for time-walk corrections of the TOA measurement. Time walk stands for the tendency of a threshold crossing time of a pulse to shift as a function of its pulse height.

### 3.2.2 Software structure

The versatile framework for MTD DAQ Software is created with Python programming language [74], it is fully modular and provides an easy user-friendly tool for system tests and further operations.

The overview of the framework is shown in Fig. 3.3. For each of the chips described in Section 3.1, a separate object is created. *lpGBT*, *SCA*, and *VTRxp* chips are defined as classes that inherit functionality from a common *chip* object, which provides a connection to the hardware (*EMP FPGA*). To enable simple user-friendly usage, an I2C (Inter-Integrated Circuit) communication protocol is employed for the chips.

The chips provide the following functionality:

- lpGBT: enables communication between TOFHIR and the back-end.
- SCA: distributes control and monitoring signals to the on-detector front-end electronics and performs monitoring operations.
- VTRxp: provides data transmission.

A dedicated object (*ROC*) is created for the TOFHIR chip that is able to perform chip *configuration* and *calibration*, and enables the *data readout*. Following the system, the ROCs are organized in the *readout unit*, which provides an appropriate *mapping*,

indicating the position of the TOFHIR chips on the boards. Finally, a *conf parser* object supplies configuration files and allows connection to the global CMS DAQ system.

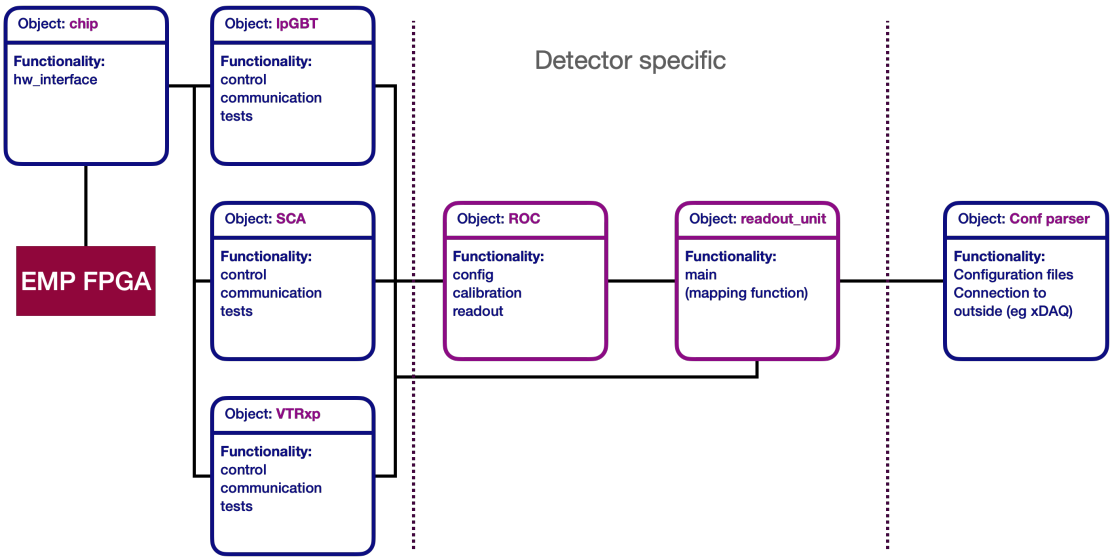



Figure 3.3: The overview of the DAQ software framework.

**Timing reconstruction**

A dedicated module has been developed to facilitate the full timing reconstruction using data obtained from the TOFHIR chip. Building upon the existing C++ framework for the TOF-PET chip, this module has been implemented in Python, ensuring easy integration with the overall MTD DAQ Software. Notably, during the code development process, certain errors of the basis code were detected and corrected, resulting in the recovery of 7% of previously missing events.

## 3.3 DAQ system tests

The tests of the DAQ system, described in Section 3.1, are carried out with software from Section 3.2. They have two main targets:

- Measure the timing resolution that can be achieved with the readout TOFHIR chip.
- Validate the full data acquisition chain along with the developed software.

In this section, the test stand setup used for the tests is described along with the obtained results.

### 3.3.1 Test stand setup

The test setup consists of two TOFHIR chips mounted on the FE board that is connected to a CC. In these tests, an alternative back-end board, called KCU105, is used as the DAQ software enabling the usage of the Serenity board is still under development. The images of the test stand are shown in Fig. 3.4.

The measurements are performed with an ultraviolet (UV) laser that shines either directly on naked SiPMs or on LYSO:CE crystals with attached SiPMs. The laser signal is split using optical breakouts in order to provide a synchronized signal for two different

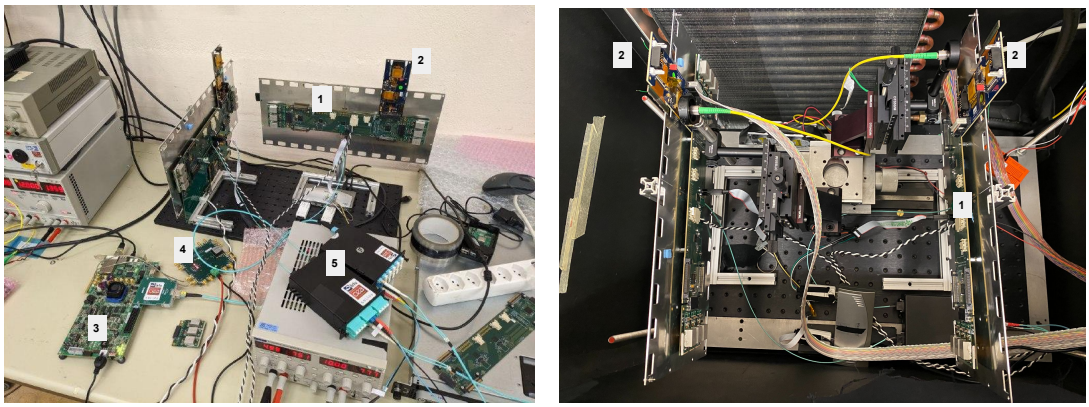



Figure 3.4: 1 - Concentrator Card v1 2 - Front-end board with 2 TOFHIR ASICs 3 - Back-end Board 4 - Clock generator 5 - Optical breakouts

SiPMs (or crystals) that are read out with two TOFHIR readout channels. The frequency of the laser can be adjustable but it usually operates at 50 kHz with a period $\Delta T = 2$ $\mu$s. Overall, approximately 51,000 events are recorded.

The whole test setup is placed in a cold box that prevents external light contamination and can be potentially cooled off to -30°, which is the operating temperature of the MTD.

The measurement involves determining the relative time difference between two TOFHIR channels that are triggered with a synchronized signal. In this case, the channel-to-channel timing resolution serves as an approximate estimation of how accurately the chip can measure the difference between the TOAs of two separate particles.

### 3.3.2 Results

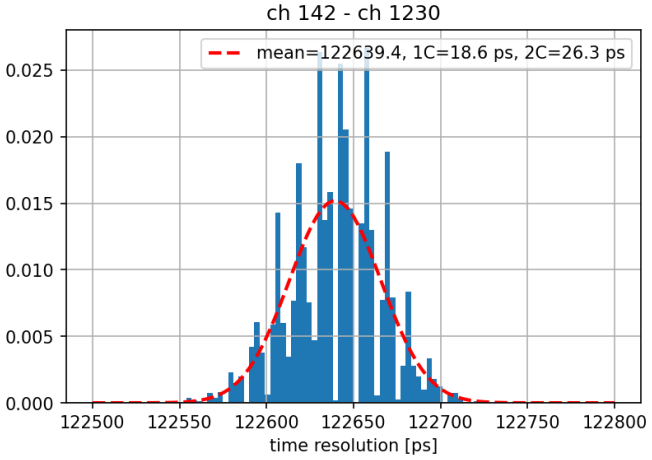



Figure 3.5: Time resolution results obtained with the TOFHIR chip during the test of the system before time-walk corrections

The results obtained from the experimental setup described above are shown in Fig. 3.5. The plot presents the timing resolution, obtained for the difference between TOA measurements of two TOFHIR channels (ch 142 as a reference and ch 1230). This is achieved by grouping together the events from different channels when they occur within the same pre-defined timing window, which can be adjusted. In these measurements, the timing window $\Delta\tau = \Delta T/4 = 500$ ns is chosen.

The timing resolution results are $\sigma_t^2 = 26$ ps for channel-to-channel measurements and, accordingly, $\sigma_t^1 = \sigma_t^2/\sqrt{2}$ =19 ps for one channel.

To mitigate the effects of time-walk, a correction is additionally applied. It is estimated by plotting the mean timing resolution in bins of TOT ratio between two channels versus this TOT ratio, illustrated in Fig. 3.6. Overall 50 bins are used over the range [-0.05, 0.05]. $2^{\text{nd}}$-order polynomial fit is applied to the distribution in order to obtain the correction values.

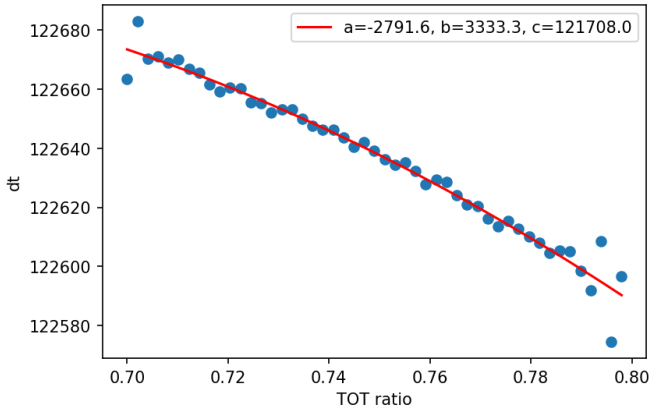



Figure 3.6: The distribution of the mean ΔTOA (dt) vs. TOT ratio between two channels is presented, and a polynomial fit is applied in order to obtain the time-walk correction parameters.

The final timing resolution is estimated from the corrected TOA distribution:

$$\Delta\text{TOA}^{\text{corr}} = \Delta\text{TOA}^{\text{uncorr}} - (a\text{TOT}_{\text{ratio}}{}^2 + b\text{TOT}_{\text{ratio}} + c), \tag{3.1}$$

where parameters $a$, $b$, and $c$ are estimated from the fit, ΔTOA is the TOA difference between two channels, and $\text{TOT}_{\text{ratio}}$ is the TOT ratio between two channels.

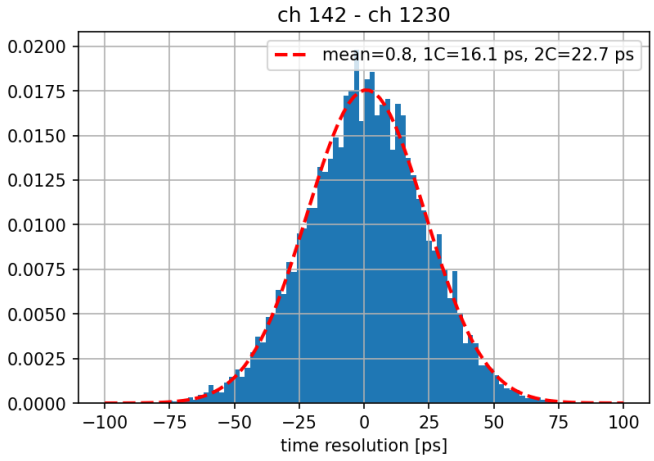



Figure 3.7: Time resolution results obtained with the TOFHIR chip during the test of the system after the time-walk corrections.

The updated results are presented in Fig. 3.7. The final resolution for channel-to-channel measurements is $\sigma_t^2 = 23$ ps and for single channel $\sigma_t^2 = 16$ ps, which is meeting the precision requirements specified for the MTD.

## 3.4 Conclusion and perspectives

In conclusion, this Chapter has presented the DAQ system for the new MIP Timing Detector and the DAQ software framework specifically designed for the Barrel Timing Layer.

The performed work presented in this Chapter was mostly dedicated to the development of the software. The major contributions were made especially to the creation of *chip*, *lpGBT*, and *SCA* classes of the DAQ framework. The implementation of the timing reconstruction module, built upon the existing C++ code, was fully performed as part of this thesis. Another important aspect of this work was active participation in the system tests. During these tests, the full acquisition system has been validated and a timing resolution of approximately 23 ps has been achieved, which is well within the required MTD resolution. The data was analyzed using the timing reconstruction module, mentioned previously.

The developed software framework will be used during the upcoming test beams and will continue to be employed for the detector's commissioning phase. It will be further developed and improved, including the component for the ETL part, in order to be used in the final MTD operations during HL-LHC.

# 4 Artificial Intelligence

The field of Artificial Intelligence (AI) has seen rapid growth in the last decade primarily due to the increased accessibility and manageability of large-scale data and the adoption of Graphics Processing Units (GPUs) [75]. This trend is clearly represented by the number of AI papers published over the years: 120k articles in 2019, which is nearly 12 times more compared to 2000 [76]. Both academic and industrial communities have been vastly applying innovative AI methods to their work due to the advantages that they offer.

The goals outlined for Particle Physics in general and for the CMS experiment in particular (see Chapter 2) require unprecedented performance for all the steps of the physics analysis chain: from data reconstruction at the detector level to the evaluation of the final physics objects. Adapting AI techniques for these applications presents both challenges and immense opportunities. AI methods not only have the potential to outperform traditional algorithms but also significantly reduce processing time.

Consequently, to facilitate the research, members of the CMS collaboration adopt increasingly sophisticated AI algorithms for data reconstruction and analysis. For instance, neural networks have already been broadly employed for the analysis of the data collected during Run 2 (e.g. [30], [77]). However, with the availability of more advanced algorithms that leverage low-level information (e.g. [78], [79]), there still remains the possibility for improvement.

This chapter provides an overview of the AI (or Machine Learning (ML)) field. It starts with an introduction to ML and covers fundamental concepts, such as model training, loss functions, hyperparameters, etc. Specific ML models, including Boosted Decision Trees, Convolutional and Graph Neural Networks, are explored in detail as they play a crucial role in the work presented in this document. Additionally, widely used ML techniques are discussed, such as Bayesian optimization and transfer learning, and Self-Attention layers are briefly introduced.

## 4.1 Machine Learning overview

### 4.1.1 Types of Machine Learning algorithms

The main difference between the traditional algorithms and ML ones is the ability of the latter to "self-learn". Unlike traditional algorithms that rely on explicitly defined instructions, ML models can autonomously uncover meaningful patterns in data, which they can further use to solve specific problems [80].

Despite the existence of numerous different ML algorithms and their classification methods, this section focuses on the most common approach to categorizing them. It is primarily based on the way the algorithm learns the necessary information and the type

of task it aims to solve.

### Supervised, unsupervised, and reinforcement learning

The classification of ML algorithms is done by the way the data in the task is represented and, consequently, how models learn from it [81]. There are three main groups in this case: supervised, unsupervised, and reinforcement learning.

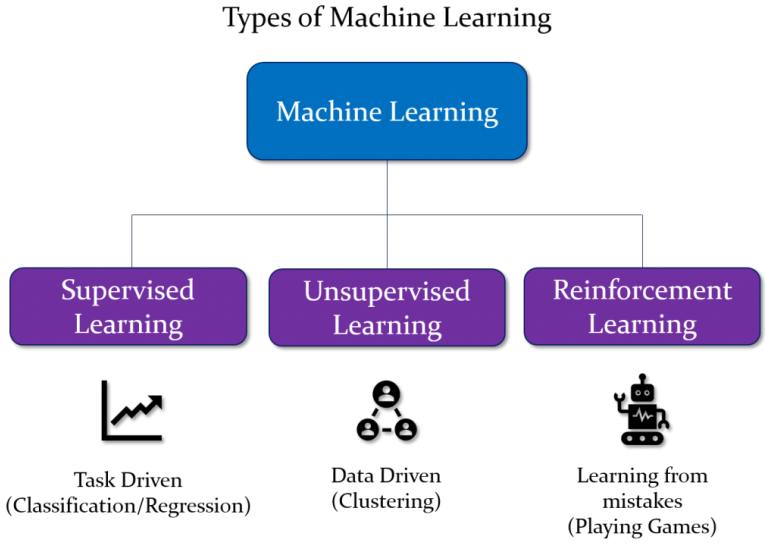



Figure 4.1: Types of Machine Learning algorithms [82].

1. **Supervised** learning.

   For supervised learning, it is obligatory to have a labeled dataset, meaning each of the samples (one entry to the dataset) has a specific target (label). The goal of a model is to learn to predict this desired *target* (output Y) based on several examples of *feature vectors* (inputs X). Essentially, a supervised algorithm aims to implicitly infer a function f: X $\rightarrow$ Y in order to be able to accurately predict outputs on the data that it has not seen before [81].

   Supervised learning is widely used for such applications as image classification, price prediction, spam detection, etc.

2. **Unsupervised** learning.

   In contrast, unsupervised learning utilizes unlabelled datasets [81], [83]. From the provided examples, the model aims to uncover similar hidden patterns or data groupings. In this case, there is no pre-defined target as in supervised learning, and the algorithm needs to solve the task without relying on human-created guidance.

   Unsupervised learning is commonly used for anomaly detection, recommendation systems, image generation, etc.

3. **Reinforcement** learning.

   For reinforcement learning, the dataset does not have to be labeled as well. The learning process employs a trial-and-error approach: good or correct actions are rewarded while negative behavior is punished. The goal of the learner is to find the best set of actions that yields the highest cumulative reward [84].

   Reinforcement learning is used for natural language processing, self-driving cars, gaming, etc.

In this document, only supervised learning algorithms are used.

#### Regression vs. classification tasks

Supervised learning models can be further categorized based on the type of task they aim to solve: classification or regression [81], [85].

- **Classification** problems require a model to predict to which category (can be two or more) each sample in the dataset belongs based on the input vector of features. The output in this case is usually a probability distribution over all possible classes, indicating the likelihood of each sample to belong to a particular category.
- In case of a **regression** problem, an ML model aims to predict a continuous numerical output ($Y$) from an input value ($X$).

### 4.1.2 Machine Learning model pipeline

In order to solve any task with an ML model, it is necessary to follow a series of steps. The initial and imperative step involves outlining the conditions of the problem. More explicitly, the desired output, as well as the metrics measuring the performance, must be defined in advance. Once the goal of the algorithm is clear the following stages are usually employed: 1) data preparation, 2) model creation, 3) model training, validation, and test.

#### Data preparation

Firstly, a suitable dataset needs to be prepared. It is a critical step for any ML development because the performance of the model directly depends on the quality of the data it is trained on. Once the relevant data is gathered, several preprocessing procedures can be applied to further enhance its impact on the training.

Common preprocessing techniques include *data cleaning*, which involves deleting missing or erroneous data, removing duplicates or irrelevant samples, and excluding outliers where possible. Another useful technique is *data normalization*, which means scaling the values of features to a similar range. This can bring further advantages in terms of performance and training stability [86].

#### Model creation

The second step is choosing an appropriate algorithm for a given problem. There are no explicit rules for selecting a suitable ML model, and it usually requires experience and creativity. However, there are a couple of factors that must be considered before making the final choice. Some important factors include:

- *Size of the dataset.* For example, neural networks tend to perform well on large datasets, but if the amount of training data is limited, simpler algorithms (e.g. K-nearest neighbors or decision trees) may be more advantageous.
- *Training and evaluation times.* Before selecting a model, it is essential to consider the time required for training as well as running the model in a production environment.

**Model training, validation, and test**

Finally, the training and evaluation of the model is performed. To ensure the robustness of the algorithm, the initial data is divided into three different datasets, each serving a specific purpose:

- A *training* dataset is used for the model to learn from. The goal of the model is to learn as much useful information from the data as possible. However, ML algorithms can suffer from a problem that is called *"overfitting"*. In this case, a model learns the specific features of the training dataset that helps to correctly predict the target but it is not able to generalize, resulting in a poor performance on new data.

- A *validation* dataset is used to mitigate overfitting. It contains data that the model has not seen during the training, and, thus, can serve to impartially evaluate the performance at the development stage. Even though the validation dataset can not directly impact the training process, it can influence the choice of the inner parameters of the model. As a result, it still introduces bias as the set of these parameters can be selected in a way to achieve the best performance specifically on the validation (and training) dataset.

- A *test* dataset is used for final performance estimation. Once the model development is complete, it is evaluated using the test dataset, which does not contain samples used during training or validation. The reported performance of any ML algorithm is usually the one obtained during the testing stage.

It is also necessary to introduce the concept of *loss function* as it plays an important role in a lot of ML models both for training (e.g. neural networks), validation, and test stages. The loss function evaluates how well the model can predict the data, it quantifies the difference between the predicted and the target values. These functions can be also separated into the ones used for classification and regression tasks. The loss functions employed for the work presented in this document are:

1. Classficitaion tasks.

   - Cross-entropy loss

   $$\mathcal{L} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k} t_j \log(p_j), \tag{4.1}$$

   where $n$ is the number of data points, $k$ is the number of classes, $t_j$ is the truth label, and $p_j$ is the predicted likelihood for a considered sample.
   In the case where $j = 2$, the loss is called binary cross-entropy:

   $$\mathcal{L} = -\frac{1}{n}\sum_{i=1}^{n}[t\log(p) + (1-t)\log(1-p)] \tag{4.2}$$

   - Focal loss
   A modification of a standard cross-entropy loss can be used in case of a class imbalance (different classes have significantly varying statistics) in the dataset [87]. The focal loss is defined as:

   $$\mathcal{L} = -\frac{1}{n}\sum_{i=1}^{n}(1-p)^{\gamma}\log(p) \tag{4.3}$$

Adding a modulating factor $(1-p)^{\gamma}$ enables the model to focus on hard, misclassified examples.

2. Regression tasks.

   - Mean Squared Error (MSE) loss

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}(y-\hat{y})^2, \tag{4.4}$$

   where $n$ is the number of data points, $y$ is the predicted values, and $\hat{y}$ is the target values.

   - Mean Absolute Error (MAE) loss

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}|y-\hat{y}|, \tag{4.5}$$

   where $n$ is the number of data points, $y$ is the predicted values, and $\hat{y}$ is the target values.

## 4.2 Boosted Decision Trees

### 4.2.1 Decision trees

A *decision tree* is a common supervised learning algorithm used both for regression and classification [88]. It has a structure of a tree with root, decision, and leaf nodes as illustrated in Fig. 4.2. The goal of the algorithm is to predict the target value ($y$) by learning simple cut-based decision rules inferred from the input feature vector ($x$). At each decision node, the data is split into two sub-sets based on one of the features and pre-defined criteria. The process is consecutively repeated until a particular stopping condition is reached.

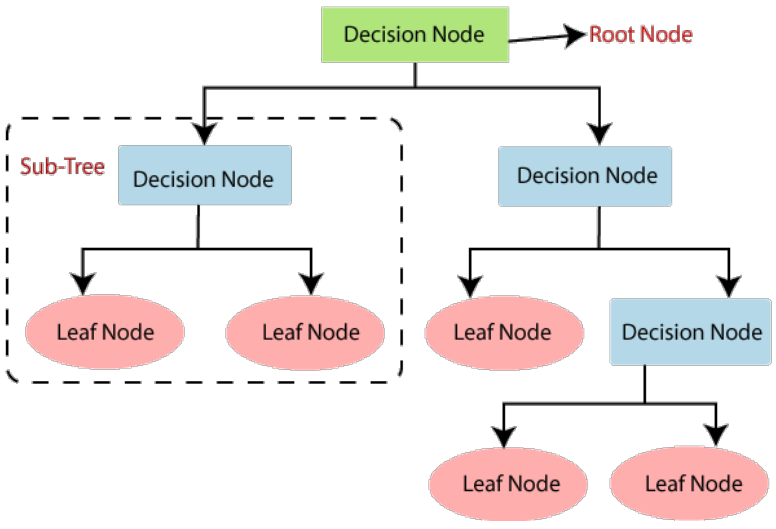



Figure 4.2: Schematic illustration of a decision tree [89].

**Splitting criteria**

In order to grow a tree, the selection of a specific feature and its splitting criteria is required at each decision node [90], [88]. This feature is used to divide the data into two daughter nodes. It is done by minimizing a function that evaluates how well each test condition can separate the samples into classes (for classification tasks). Among the most commonly used functions are:

1. Classification criteria

   - Information gain. The information gain is defined as the difference in entropy before and after performing a data split. The formula for entropy for a given dataset is defined in Eq. 4.6.

     $$H(Q_m) = -\sum_k p_{mk} \log(p_{mk}), \tag{4.6}$$

     where $Q_m$ is data at a defined node $m$ and $p_{mk}$ is the proportion of class k observations in node m and it is defined in Eq. 4.7.

     $$p_{mk} = \frac{1}{n_m} \sum_{y \in Q_m} I(y = k), \tag{4.7}$$

     where $n_m$ is the number of samples at node $m$ and the function $I(y = k)$ returns 1 if the predicted label $y$ matches the true class value $k$; otherwise, it returns 0.

     If all the samples in a node belong to the same class, the entropy will equal zero. Thus, the feature and the criteria that give the lowest amount of entropy after the splitting or, in other words, the highest information gain will be used.

   - Gini impurity. Gini impurity represents the probability to misclassify a randomly selected sample within a given node, assuming it is labeled based on the node class distribution. The formula to calculate the Gini impurity is given in Eq. 4.8.

     $$H(Q_m) = \sum_k p_{mk}(1 - p_{mk}), \tag{4.8}$$

     Similar to the entropy, if the classification is perfect in the node, the Gini impurity will be zero. Thus, the feature is selected in a way to minimize this criterion.

2. Regression criteria

   - Mean Squared Error (MSE). For the continuous values, the MSE is one of the most common criteria. It is defined as follows:

     $$H(Q_m) = \frac{1}{n_m} \sum_{y \in Q_m} (y - \overline{y}_m)^2, \tag{4.9}$$

     where $Q_m$ is data at a defined node $m$ with $n_m$ samples and $\overline{y}_m$ is the mean value at the given node. In this case, the predicted value of the leaf node will be the mean value of all the samples in it ($\overline{y}_m$).

   - Mean Absolute Error (MAE). The MAE is similar to MSE but instead of the mean value, the predicted estimate for each node will be the median$(y)_m$.

The error is defined as follows:

$$H(Q_m) = \frac{1}{n_m} \sum_{y \in Q_m} |y - \text{median}(y)_m|, \tag{4.10}$$

where $Q_m$ is data at a defined node $m$ with $n_m$ samples.

**Stopping criteria**

Another important condition for a decision tree is a stopping criteria. If the tree is allowed to grow indefinitely, the splitting can continue until there is only one sample at each leaf node. Clearly, such a tree is overfitted and has no use. Thus, the criteria to be satisfied in order to stop the splitting and declare the node to be terminal have to be defined. They can include:

- minimum leaf size — the minimum amount of samples required to be in each daughter node after the splitting.
- maximum tree depth — the maximum number of layers that a tree can have.
- minimum samples leaf — the minimum amount of samples the node should have to continue its splitting.

**Advantages and disadvantages**

Decision trees are commonly used for different applications in ML because of their advantages. Among the main ones is the fact that they are easily interpretable and require little data preparation. However, as mentioned earlier, decision trees can suffer from overfitting. Even when the stopping criteria are employed, the algorithm can still create an over-complex tree that is not able to generalize to the new data. Different techniques can be used to mitigate the overfitting, such as pruning [91], for example. In this document, I will focus on another technique, known as the ensemble method called Boosted Decision Tree.

### 4.2.2 Gradient Boosting

Gradient-boosted decision trees are ML models that combine multiple weak learners (decision trees) into a new better-performing algorithm (strong model) [92], [93]. In this case, decision trees are added sequentially, with each subsequent algorithm attempting to minimize the error of the previous one. In order to do so, when a new tree is added, it fits on the gradient (or error) obtained as a result of the precedent model.

Formally, a strong model $F$ can be considered that tries to solve a regression task using MSE loss. $F_k$ is the model at step k that should be improved at the next iteration. It can be done by adding a weak learner $h_k$ such that:

$$F_{k+1} = F_k + h_k = \hat{y}, \tag{4.11}$$

where $\hat{y}$ is the target value. The formula can be re-written in the following way:

$$h_k = \hat{y} - F_k \tag{4.12}$$

Now, the subsequent decision tree would be trained on the residual $h_k$ instead of $\hat{y}$. Taking into account Eq. 4.4 with $y = F(x_i)$, it can be shown that the residuals for a

given model are proportional to negative gradients:

$$-\frac{\partial \mathcal{L}_{MSE}}{\partial F(x_i)} = \frac{2}{n}\sum_{i}^{n}(\hat{y}_i - F(x_i)) = \frac{2}{n}h_k(x_i) \tag{4.13}$$

Thus, the method can be generalized by taking an appropriate loss function for the task and performing the subsequent training on the obtained gradients.

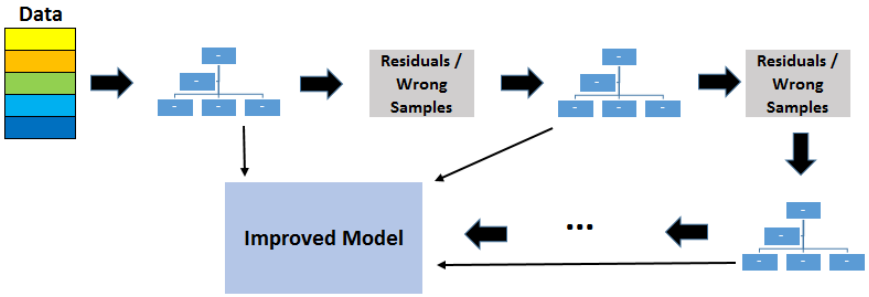



Figure 4.3: Schematic illustration of a gradient-boosted decision tree [94].

### Hyperparameters

Apart from the hyperparameters described for the decision tree, for this model, there are two extra values:

- *Learning rate.* To further mitigate overfitting, a learning rate parameter can be introduced, which controls how fast the strong model learns. The decision tree is multiplied by a small value ($\alpha$, typically of the order of 0.1) before adding it to the strong model. In other words, Eq. 4.11 is changed as follows:

$$F_{k+1} = F_k + \alpha h_k, \tag{4.14}$$

- *Number of estimators.* A maximum number of trees allowed in the ensemble.

The final algorithm aggregates the results from each step. Such an approach creates a stable and highly efficient model. Gradient-boosted decision trees are especially popular ML algorithms in high-energy physics.

## 4.3 Neural networks

A neural network (NN) is a type of ML algorithm that was inspired by connections of biological neurons in the brain [95]. They are used ubiquitously both for regression and classification tasks due to their outstanding performance, especially in applications to image recognition, speech recognition, and language processing.

The first and simplest type of NN that was developed is a feedforward neural network (FNN). It consists of three parts: an input layer, one or more hidden layers, and an output layer. Unlike in other types of networks (such as recurrent NN), in this case, the information flows only in one direction: from the input through the hidden layers to the output. Each layer is comprised of artificial neurons as illustrated in Fig. 4.4. They can take one or more inputs and produce a single or multiple outputs.

The learning process of the network consists of two stages:

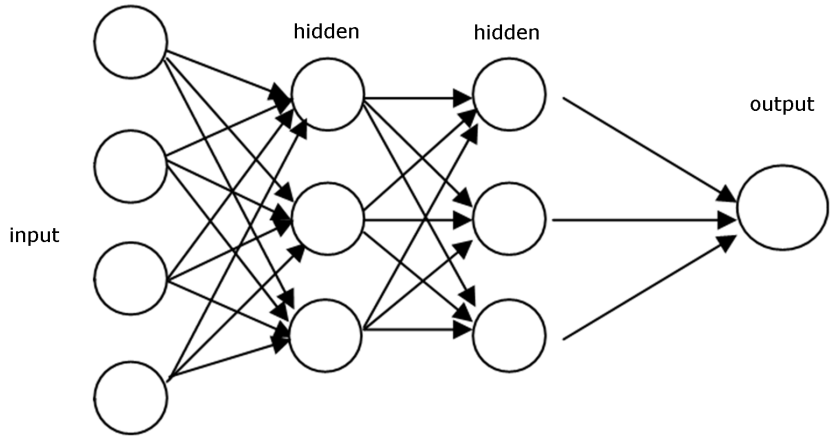



Figure 4.4: Schematic illustration of a feedforward neural network with an input layer, two hidden layers, and an output layer [96].

1. *Forward pass* — a step, where the information from the input layer traverses hidden layers to obtain the required output.
2. *Backpropagation* — a method to update the parameters of the NN to make predictions of the output layer more accurate.

In this section, all the mentioned concepts will be discussed in more detail taking an FNN as a basic example. Moreover, the methods to further optimize the performance of NN will be described along with a commonly used technique called "transfer learning" that is further used in the work presented in this document.

### 4.3.1 Forward pass

#### Artificial neuron

To understand how forward pass is executed, first, a single artificial neuron can be considered. Its inner structure is illustrated in Fig. 4.5.

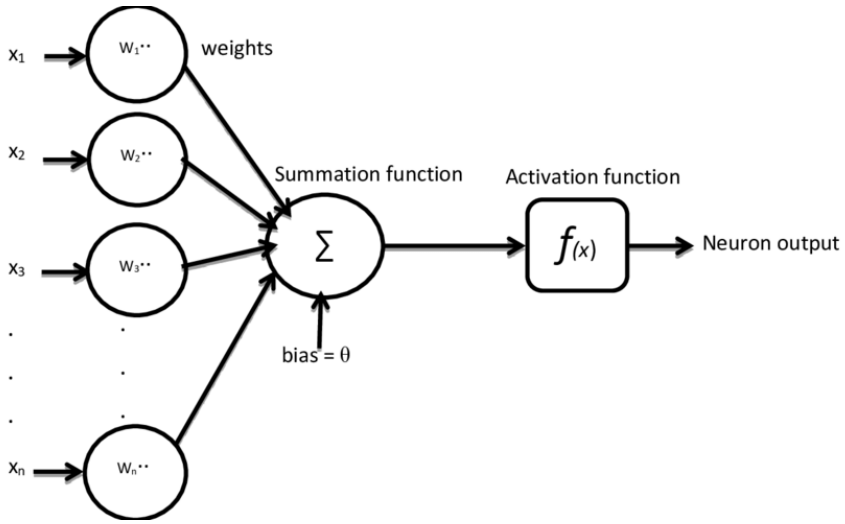



Figure 4.5: Schematic illustration of an artificial neuron. It takes one or more inputs, calculates their weighted sum, and applies an activation function [97].

The neuron executes three steps:

1. Each input is multiplied by a specific *weight*:

$$\begin{aligned} x_1 &\rightarrow w_1 \cdot x_1 \\ x_2 &\rightarrow w_2 \cdot x_2 \\ &\dots \end{aligned} \tag{4.15}$$

   A weight is a real number that represents the importance of a respective input to the output.

2. The weighted inputs are summed up and an extra parameter – *bias* – is added:

$$\sum_i^N (x_i \cdot w_i) + b \tag{4.16}$$

   The weights and biases are the *trainable parameters* of the network.

3. Finally, a special function ($\sigma$) called "activation function" is applied to introduce non-linearity between inputs and outputs and to adjust the output to fit into the necessary scale:

$$\sigma\left(\sum_i^N (x_i \cdot w_i) + b\right) \tag{4.17}$$

**Activation functions**

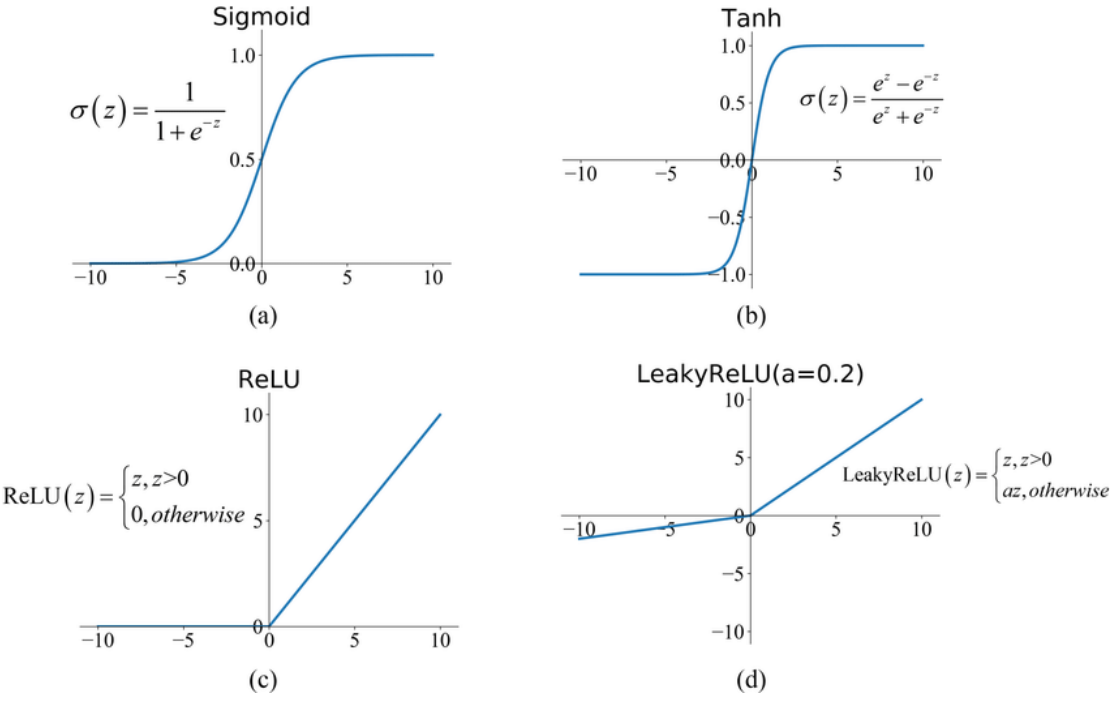



Figure 4.6: Most commonly used activation functions for neural networks [98].

The activation functions are chosen based on their performance and the particular task the NN is trying to solve. Among the most commonly employed ones are [99]:

- *Sigmoid*. It is usually used in applications where a probability is required as an output because the function exists between (0,1). Its mathematical representation and distribution are shown in Fig. 4.6 (a).

- *Hyperbolic tangent (tanh).* It has a similar shape to a sigmoid function but the range is between (-1,1). It is represented in Fig. 4.6 (b).
- *Rectified Linear Unit (ReLU).* ReLU is the most commonly used activation function for NNs. It is more computationally efficient compared to sigmoid and tanh as only a certain number of neurons will have a non-zero output. It is shown in Fig. 4.6 (c).
- *Leaky ReLU.* ReLU activation function can suffer from a so-called "dying ReLU problem": if one of the nodes reaches a negative value it will stay at zero for the whole training, which can significantly degrade the performance. To mitigate this issue, a Leaky ReLU can be used that is depicted in Fig. 4.6 (d).

#### Feedforward network

A feedforward network has several artificial neurons connected together. The input layer consists of a feature vector (for example, it can be the intensities of image pixels) that is processed by hidden layers that aim to uncover underlying structures and connections to predict the most accurate output. Each new hidden layer takes the output of the previous one, effectively increasing the level of abstraction of the decision-making.

Formally, the process can be described with the following equation:

$$h^{l+1} = \sigma(W^T h^l + b), \tag{4.18}$$

where $h^l$ is the output vector of a hidden layer $l$, $W$ is a weight matrix, $b$ is a vector of biases, and $\sigma$ is an activation function. It is repeated until the output layer is reached.

The most notable feature of a neural network is that it can automatically devise the weights and the biases that will result in the best performance using backpropagation and gradient descent.

### 4.3.2 Backpropagation and gradient descent

After the forward pass is executed, the output of the network is compared to the target value using an appropriate loss function. The goal of the network training is to minimize this loss function by adjusting the weights and biases of the model. In order to see how this process is done, a simple example of an FNN shown in Fig. 4.7 can be considered.

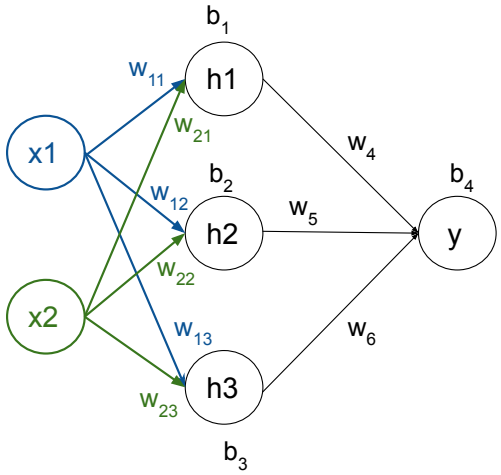



Figure 4.7: Example of a simple feedforward neural network.

In this case, the loss is a function of all trainable parameters of the network:

$$\mathcal{L}(w_{11}, w_{12}, w_{13}, w_{21}, w_{22}, w_{23}, w_4, w_5, w_6, b_1, b_2, b_3, b_4) \tag{4.19}$$

To evaluate how the loss function evolves with the change of weight $w_{11}$, a partial derivative ($\frac{\partial \mathcal{L}}{\partial w_{11}}$) needs to be calculated. It can be done using the chain rule:

$$\frac{\partial \mathcal{L}}{\partial w_{11}} = \frac{\partial \mathcal{L}}{\partial y} \cdot \frac{\partial y}{\partial w_{11}} \tag{4.20}$$

The output $y$ can be also represented as a multivariable function:

$$y = f(w_4 h_1 + w_5 h_2 + w_6 h_3 + b_4) \tag{4.21}$$

Accordingly its partial derivative with respect to $w_{11}$ is:

$$\frac{\partial y}{\partial w_{11}} = \sum_{i=1}^{3} \frac{\partial y}{\partial h_i} \cdot \frac{\partial h_i}{w_{11}} \tag{4.22}$$

As only $h_1$ depends on $w_{11}$, it can be simplified as:

$$\frac{\partial y}{\partial w_{11}} = \frac{\partial y}{\partial h_1} \cdot \frac{\partial h_1}{w_{11}} \tag{4.23}$$

And the loss function partial derivative can be re-written as:

$$\frac{\partial \mathcal{L}}{\partial w_{11}} = \frac{\partial \mathcal{L}}{\partial y} \cdot \frac{\partial y}{\partial h_1} \cdot \frac{\partial h_1}{\partial w_{11}} \tag{4.24}$$

The required derivatives $\frac{\partial y}{\partial h_1}$ and $\frac{\partial h_1}{\partial w_{11}}$ are calculated according to the formulas:

$$\frac{\partial y}{\partial h_1} = w_4 \cdot f'(w_4 h_1 + w_5 h_2 + w_6 h_3 + b_4), \tag{4.25}$$

$$\frac{\partial h_1}{\partial w_{11}} = x_1 \cdot f'(w_{11} x_1 + w_{12} x_2 + b_1), \tag{4.26}$$

where $f'$ is the derivative of the activation function.

This process is called *backpropagation*. In a similar way, the partial derivatives can be calculated for all the weights and biases of the model and further extrapolated to more complex networks.

After the relation between the change in the weight and loss function is derived, a *gradient* descent can be applied. The goal is to find the minimum of the loss function by adjusting the weight in the direction opposite to the gradient as illustrated in Fig. 4.8.

Mathematically, for the considered case it can be represented as follows:

$$w_{11}^{\text{new}} = w_{11} - \alpha \frac{\partial \mathcal{L}}{\partial w_{11}}, \tag{4.27}$$

where $w_{11}^{\text{new}}$ is the new updated weight and $\alpha$ is a parameter called "learning rate" that controls how fast the learning is performed.

In a more general case, the equation is written as:

$$W_{\text{new}} = W_{\text{previous}} - \alpha \nabla_W \mathcal{L}, \tag{4.28}$$

where $W_{\text{new}}, W_{\text{previous}}$ are updated and old trainable parameters respectively and $\nabla_W \mathcal{L}$

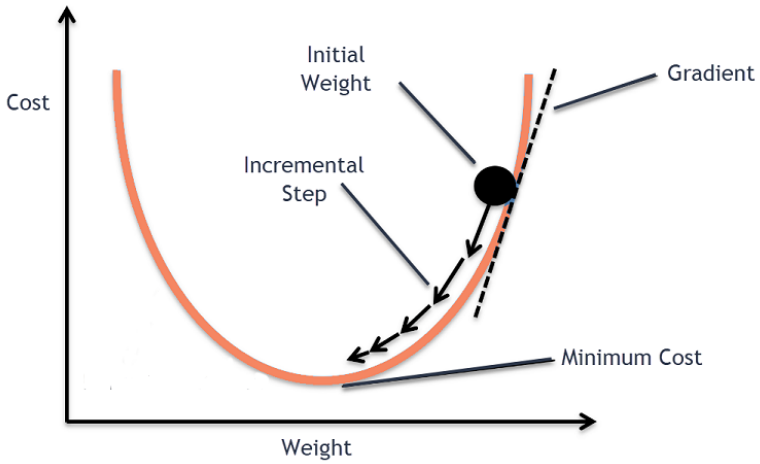



Figure 4.8: Schematic illustration of a gradient descent, which aims to find the minimum of the loss (cost) function.

is a vector of partial derivatives.

In the previous equations, $\mathcal{L}$ is represented as an average loss function on all of the training data samples:

$$\mathcal{L} = \frac{1}{n} \sum_{x} \mathcal{L}_x, \tag{4.29}$$

where $x$ represents one data sample and $n$ is the overall number of samples.

However, the datasets used to train neural networks can be extremely large. Thus, calculating the gradient for the whole dataset before updating the weight can be inefficient and computationally exhaustive. In order to mitigate this problem, a technique called *mini-batch gradient descent* can be used. In this case, the parameters of the model are updated on a small random subset of a training dataset called a *"batch"*.

In the context of this document, other extended versions of the gradient descent optimization algorithm are used:

- Adaptive momentum estimation (Adam) [100]. This algorithm also uses subsets to perform the training. The difference is that the Adam optimizer adapts the learning rate to each trainable parameter of the model instead of keeping it constant. It is one of the most commonly used algorithms in deep learning.
- LAMB [101]. This optimization algorithm was specifically designed to train the data on very large batch sizes.

### 4.3.3 Optimization techniques

#### Hyperparameters

The training process described above is dedicated to finding the best possible set of trainable parameters of the model. However, there is another type of internal variable, called *"hyperparameters"* that can not be inferred by the model itself. These values are used to control the learning process and have to be selected with human input.

Hyperparameters play a crucial role in neural networks: they affect model performance, training and inference times, and computational cost. They can be divided into two groups: 1) parameters determining the structure of the network and 2) parameters controlling how the network is trained.

1. Structure of the model.

- **Number of hidden layers and nodes**. These are one of the most important hyperparameters determining the final performance of the network. The balance has to be found between using too many layers and nodes, which can result in overfitting, and not using enough, which leads to sub-optimal performance.

2. Network training.

   - **Learning rate**. This parameter has already been mentioned when discussing gradient descent in relation to Eqs. (4.27) and (4.28). It controls how fast the training is performed. If the learning rate is too small, the training will take a long time. On the other hand, if it is too high, the minimum of the loss function might not be achieved.
   - **Number of epochs**. This parameter defines how many times all the samples of the training dataset went through the model. A number of epochs is chosen based on the validation loss: when it is not improving for several epochs, the training is stopped.
   - **Batch size**. This parameter has also been mentioned before when discussing the gradient descent. It controls how many samples the model processes before updating weights and biases. The large batch size can cause problems in terms of computational cost and memory limits, while the small batch size takes a longer time to train and can introduce more noise in the gradient calculation, thus, preventing the network to reach the optimal minimum.
   - **Dropout rate**. It is one of the regularization techniques that helps to mitigate overfitting [102]. The idea is that each individual node of the model can be excluded from a training cycle with a probability (1-$p$), where $p$ is a dropout rate.

The best set of hyperparameters is usually inferred from the performance estimated on the validation dataset. As mentioned earlier, to avoid the associated bias, the final results for the models are presented on a separate test dataset.

#### Bayesian optimization

There are various methods that can be used to find the optimal hyperparameters, such as manual search or random grid search. In the work presented in this document, Bayesian hyperparameter optimization [103] is chosen as the preferred approach.

It is an automatic technique, where the algorithm aims to construct a probabilistic model that establishes a relationship between hyperparameters and the loss function evaluated on the validation dataset. By doing so, the Bayesian algorithm can identify the promising regions within the hyperparameter space that are worth exploring. As the algorithm refines its prediction with each training iteration, it progressively improves the selection of hyperparameters, leading to the best model performance.

### 4.3.4 Transfer learning

Transfer learning (TL) is another widely-used technique for neural networks. Its key concept involves leveraging knowledge gained from solving a specific broad problem to tackle a different but related task. It can be especially useful in applications where the amount of data for a more narrowly focused problem is limited.

Within the context of this document, TL is employed to introduce a new target value into the model while maintaining excellent performance achieved for already existing targets (see Chapter 5). For this purpose, the following steps are executed:

1. A model is trained and optimized on a large dataset to predict specific targets that are relevant to the final problem.

2. The weights and biases that are associated with the components of the model that are unrelated to the final problem are "frozen", meaning they will not change during the subsequent training.

3. The model is then trained on the new dataset that is specifically tailored to address the final problem, enabling it to adapt and solve the task effectively.

## 4.4 Selected types of Neural Networks

In this section two types of neural networks will be discussed: *convolutional neural networks (CNNs)*, which are primarily used for image recognition tasks, and *graph neural networks (GNNs)*, which can operate on graph-structured data. Additionally, a brief introduction to the Self-Attention layers is given. These algorithms are further implemented in Chapters 5 and 6 for calorimeter reconstruction.

### 4.4.1 Convolutional Neural Networks

CNN is a type of artificial neural network that is specifically designed to process data that has a grid-like structure (e.g. an image) [104]. It is dominantly used for various applications in computer vision, like medical image analysis, recommender systems, image segmentation and classification, etc.

The immense popularity of CNNs is explained by their superior ability to identify complex patterns in images and make predictions based on them. A CNN architecture typically consists of three main building blocks, each dedicated to a specific task: 1) convolutional layer, 2) pooling layer, and 3) fully-connected layer. The first two perform the feature extraction while the last one is used to obtain the final output (classification or regression target). Multiple convolutional and pooling layers are usually stacked together followed by one or several fully connected layers to achieve the final architecture. An example of a basic CNN for image classification is presented in Fig. 4.9.

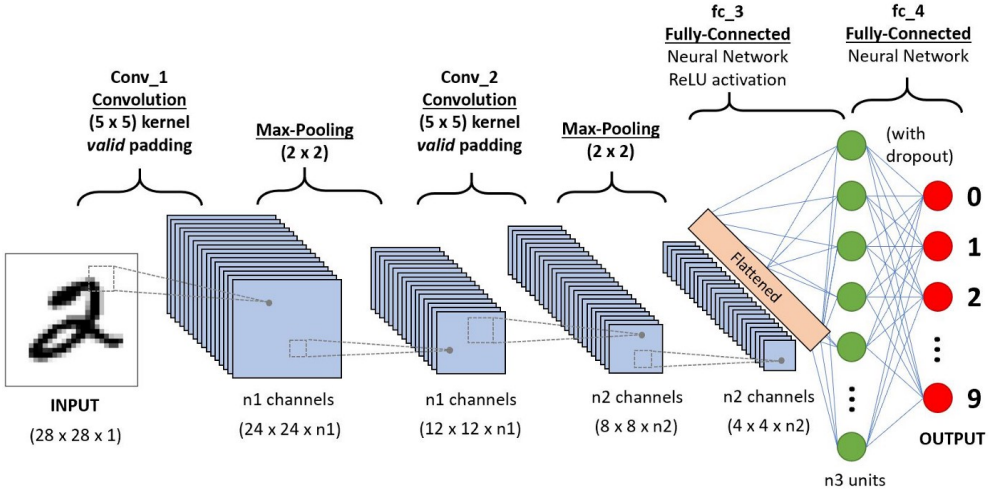



Figure 4.9: Schematic illustration of a convolutional neural network architecture used for image classification [105].

### Convolutional layer

A convolutional layer is an essential component of any CNN architecture. Its purpose is to extract valuable features from an image using a linear operation known as convolution.

A convolutional layer takes an image as input and performs a dot product between its local field and a special matrix with trainable parameters called a *"kernel"* or *"filter"*. The size of a kernel is a hyperparameter and it usually varies from 3x3 to 7x7. After one region of an image has been processed, a kernel moves to the next area. The distance between two successive kernel positions is also a hyperparameter of the layer known as a *"stride"*. The values of the stride are usually kept small at one or two pixels. As the kernel progresses, it processes different regions of the image, creating a feature map that indicates the presence of specific features. This process is illustrated in Fig. 4.10. Following the convolution, an activation function is applied to introduce non-linearity.

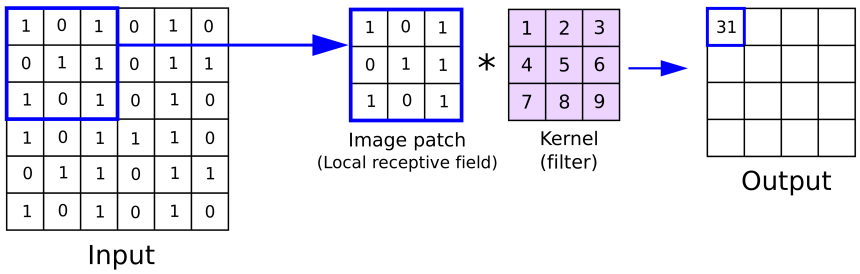



Figure 4.10: Schematic illustration of a convolution operation between an image and a kernel [106].

The key aspect of a convolutional layer is weight sharing, where the same kernel is used across all positions of the image. This approach enables the network to efficiently search for a specific feature within the input. By employing multiple different kernels, feature maps are generated to represent various characteristics of the image. The overall number of kernels is another hyperparameter of a convolutional layer.

Furthermore, several convolutional layers can be stacked on top of each other. In this case, as one layer feeds its output to the next one the model progressively captures increasingly complex and abstract features of the input image.

### Pooling layer

A pooling layer provides a downsampling operation to reduce the dimensionality of the feature maps created after the convolution. A pooling operation involves sliding a two-dimensional window over the feature map and summarising the features falling within the area covered by the window. There are no learnable parameters in the pooling layer, as the size of the window and the stride are hyperparameters. There are two types of pooling operations:

1. **Max Pooling**. The maximum element is chosen from the area of the feature map covered by the pooling window.

2. **Average Pooling**. Computes the average of the elements of the feature map covered by the pooling window.

An example of a pooling layer is presented in Fig. 4.11.

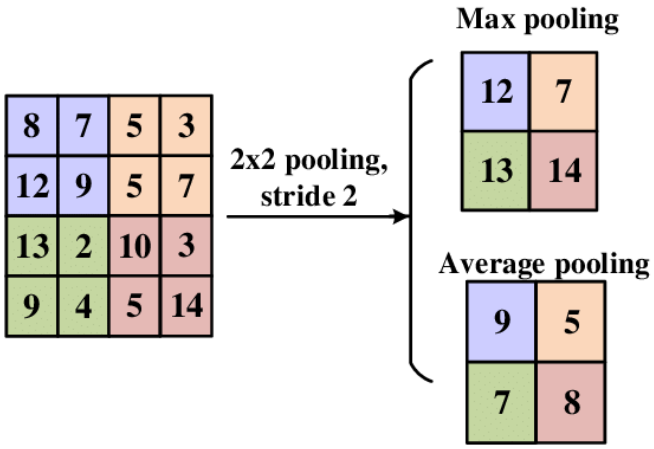



Figure 4.11: An illustration of a pooling operation performed with Max and Average pooling [107].

The main benefits of using a pooling layer are: it reduces the number of trainable parameters and it makes the model more robust to small variations in the feature positions in the image as it produces a summary of each processed region.

#### Fully-connected layer

The output of the final convolutional or pooling layer is typically flattened in a one-dimensional vector (sometimes called a vector of *latent features*) and passed through a single or several fully-connected (or *dense*) layers. These layers were already previously discussed in the context of an FNN: each input is connected with an output by a learnable weight.

The final dense layer is constructed in a way that fits the desired target. For example, for a binary classification task, the final dense layer consists of one node with a sigmoid activation function. In this case, a binary cross-entropy loss function should be used. The training of a CNN is done in a similar way as for an FNN using backpropagation and gradient descent to obtain the learnable parameters of the convolutional and dense layers.

### 4.4.2 Graph Neural Networks

The Graph Neural Network (GNN) is a relatively new but rapidly evolving model designed to operate and analyze graph-structured data [108]. The key distinctive feature of GNN is the message-passing (MP) technique, enabling the exchange of information among network inputs, as schematically shown in Fig. 4.12.


Figure 4.12: Illustration of message-passing procedure [109].

GNNs recently gained significant popularity in high-energy physics applications

(e.g. [110], [111]) due to the following factors :

- GNNs can effectively handle sparse data coming from complex detector geometries.
- They can be applied to non-Euclidean data with variable input sizes.

### Definitions

Before exploring how GNN operates, it is essential to define a graph and related concepts.

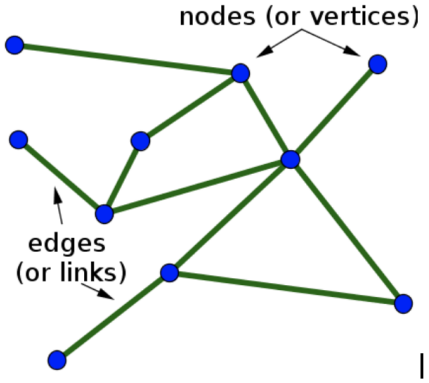



Figure 4.13: Schematic representation of a graph [112].

A graph is a mathematical structure consisting of nodes that contain object features and edges that reflect the relationships between the nodes. Mathematically, a graph can be represented by a $N \times D$ feature matrix $X$, where $N$ is the number of nodes and $D$ is the number of features. The relationship between nodes can be given by an adjacency matrix ($A$) — a square matrix with elements indicating whether the pairs of nodes are connected in the graph. An adjacency matrix can also incorporate additional information, such as the distance between the nodes instead of only the presence of the edge.

### Architecture

The main goal of a GNN is to extract information from the input graph and provide predictions for individual nodes, edges, or the entire graph. Multiple architecture choices exist for graph representation, depending on the specific problem and data characteristics. One of the widely used models is the Graph Convolutional Network (GCN) [113]. In this case, to incorporate the message-passing feature, the forward pass from Eq. 4.18 should be modified to account for the adjacency matrix as follows:

$$f(H^{(l)}, A) = \sigma(AH^{(l)}W^{(l)}) \tag{4.30}$$

Introducing the adjacency matrix to the equation means that all the features of neighboring and connected nodes are aggregated. However, in this formulation, the features of the graph node in consideration are not directly involved. To address this, a self-loop is included by adding an identity matrix ($I$) to $A$. Furthermore, the adjacency matrix should be normalized to ensure the preservation of the scale of the feature vectors. In GCN, it is done using the following equation:

$$D^{-\frac{1}{2}}\hat{A}D^{-\frac{1}{2}}, \tag{4.31}$$

where $\hat{A} = A + I$ and $D$ is the diagonal node degree matrix of $\hat{A}$.

Taking these adjustments into account, the final propagation rule becomes:

$$f(H^{(l)}, A) = \sigma(D^{-\frac{1}{2}}\hat{A}D^{-\frac{1}{2}}H^{(l)}W^{(l)}) \tag{4.32}$$

With each message-passing layer, the network expands its knowledge of the nodes, effectively enabling information sharing between more distant neighbors.

In other implementations of graph neural networks, different variants of message-passing can be introduced, such as summing, averaging, or concatenating the neighboring features.

#### Graph Highway Network

GCNs are effective and efficient models used for graph-structured data. However, they suffer from a so-called "over-smoothing" problem, when after stacking multiple GCN layers (e.g. 2 or 3 layers), the learned latent features from different graph nodes converge to alike vectors as a result of learning the information about the neighbors.

In order to mitigate this issue, a modified version of GCN, called *Graph Highway Network (GHN)* is introduced. It uses *gating units* to automatically balance the trade-off between maintaining the specifications of a selected node and aggregating the knowledge about the neighbors. Such type of architecture allows to use deep (multiple layers) neural networks on graphs without degrading the performance. A detailed description of a GHN can be found in Ref. [114].

### 4.4.3 Self-Attention layers

*Self-Attention (SA)* layers, initially developed for language modeling tasks, are another neural network architecture that recently gained popularity. The key feature of these layers is the ability to help the network to focus more on specific parts of the input that are most correlated with the desired output.

A simplified description of the self-attention mechanism for a task with multiple ($n$) input vectors is as follows:

1. Three distinct matrices called **key**, **query**, and **value** are created. The weights of these matrices are learnable parameters of the network.

2. For each input vector, representations for the key ($Q$), query ($K$), and value ($V$) are obtained by multiplying the vector with the corresponding matrix.

3. For each input vector:

   - Attention scores are evaluated as a dot product between the key representation $Q$ of the considered vector and query representations $K$ of each of the input vectors $i$ (including itself) and further passed through a softmax function:

   $$\text{Attention}_i(Q, K_i) = \text{softmax}\left(QK_i^T\right) \tag{4.33}$$

   - The values representation $V$ of the input vectors are multiplied by the respective attention scores, and the resulting vectors are summed up:

   $$\sum_i^n V_i \cdot \text{Attention}_i(Q, K_i)_i \tag{4.34}$$

4. The procedure is repeated for each of the input vectors, and the output of the Self-Attention layer is the updated weighted vectors.

A detailed description of Self-Attention layers can be found in Ref. [115]. Self-Attention layers are beneficial because they allow neural networks to put more importance on relevant parts of the input, emphasizing key relationships, and, as a result, improving their ability to capture complex patterns in various tasks.

# 5 SuperClustering reconstruction in the ECAL with Deep Learning

The main purpose of the ECAL reconstruction is to be able to identify and correctly evaluate the kinematic properties of electrons and photons. They can be retrieved from the energy patterns left in the detector by these particles. The process starts by combining the reconstructed deposits (PFRechits) into clusters of energy (PFClusters), each of them representing single or multiple overlapping particles. A detailed description of the current algorithm is presented in Chapter 2, and a DeepCluster model created for this task is introduced in Chapter 6.

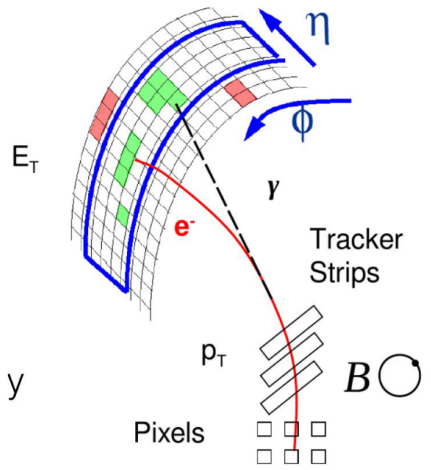



Figure 5.1: Schematic illustration of an electron emitting a bremsstrahlung photon before entering the calorimeter [116].

However, before reaching the calorimeter, an electron or photon may interact with the material in front of the ECAL, resulting in the production of multiple particles. In this case, a photon may convert to an electron-positron pair while an electron may emit bremsstrahlung photons. As a result, multiple PFClusters, originating from one initial particle, will appear in the ECAL (more details in Chapter 2).

To correctly reconstruct the energy of the primary electron or photon, all the produced PFClusters must be combined into a group called a SuperCluster (SC). In the CMS experiment, it is done using a geometrical algorithm called "Mustache", described in Chapter 2.

Even though this algorithm has shown good performance for the current ECAL reconstruction, it still has a number of limitations, particularly, in filtering pileup interactions and noise. This issue will become more significant during Run 3 and subsequent HL-LHC due to the aging of the detectors and increased luminosity. In order to address it, novel Machine Learning approaches are being explored.

In this chapter, I will present an innovative DeepSC model, developed for SuperClustering reconstruction in ECAL. It is primarily based on Graph Neural Networks (GNNs) and Self-Attention (SA) layers. In the first section, the details of the model development and training are presented, including the dataset that is used and the architecture of the DeepSC network. The performance of the model in terms of energy resolution along with a comparison to the "Mustache" algorithm is presented in Section 5.2. Additionally, DeepSC network is able to predict whether the reconstructed SC originated from a

photon, electron, or jet. This new feature is referred to as particle identification (ID) and its development, associated model training, and obtained results are presented in Section 5.3.

## 5.1 DeepSC model and datasets description

The significant drawback of the "Mustache" algorithm comes from the fact that it is an entirely geometrical approach, meaning that all PFClusters within a specified area are aggregated to form an SC. It results in the contamination of energy of the final object with noise and pileup. As discussed in Chapter 2, an ML technique (Boosted Decision Tree) has been already implemented in order to improve energy resolution. However, leveraging more advanced Deep Learning methods, specifically operating on low-level information such as PFRechits, can provide additional benefits.

In particular, GNNs are considered for the SuperClustering reconstruction. Their main advantages for the presented task are:

- GNNs can easily process varying input sizes, which is important as the number of PFClusters, determining the input size for the model, is different for each SC.
- GNNs enable communication between different PFClusters through message-passing, providing additional information to the network and improving overall performance.

Another widespread technique used in the DeepSC development is Self-Attention (SA) layer. It helps the network to focus on the most important information.

The DeepSC model takes the features of all PFClusters within a predefined area as input. Unlike the "Mustache" algorithm, the network performs a selection of the PFClusters in order to create an optimal SC. One of the outputs of the model is the *PFCluster score*, indicating the likelihood for each PFCluster to belong to the SC.

Furthermore, the model can perform additional tasks such as energy correction prediction, which can be used instead of applying a BDT on top of the created SC. This aspect is not considered in the work presented in this document as it is still under development.

Another output of the model corresponds to the particle ID. It aims to classify the reconstructed SCs into one of the following groups: originated from prompt photons, prompt electrons, or hadrons forming jets. Even though, the DeepSC model does not specifically target jet reconstruction, the identification of PFClusters coming from it can significantly reduce the background for prompt photons and electrons.

In this section, the dataset used for training and evaluation of the model is described, including details about the initial data characteristics and the specific input format for the model. After it, the model architecture is presented, with each component of the network thoroughly discussed.

### 5.1.1 Parameters of the datasets

To train the DeepSC model and test its performance, several dedicated datasets are generated.

The first one is created to target the SC reconstruction. Electrons and photons are simulated using a full CMS Monte Carlo simulation at 14 TeV. They are generated uniformly in pseudorapidity and transverse momentum $p_{\mathrm{T}} = [1, 500]$ GeV, the corresponding distributions are shown in Fig. 5.2. Pileup is simulated by adding secondary interactions uniformly distributed in the range 55 to 75 as presented in Fig. 5.3.

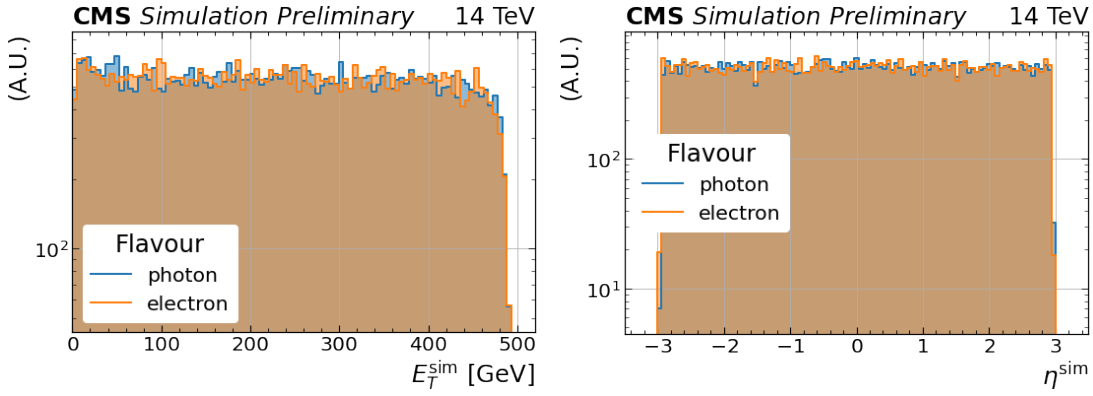



Figure 5.2: Distribution of $E_T$ (left) and $\eta$ position (right) for electrons and photons in the simulated dataset produced for the DeepSC model. Events are created using a full CMS Monte Carlo simulation at 14 TeV.

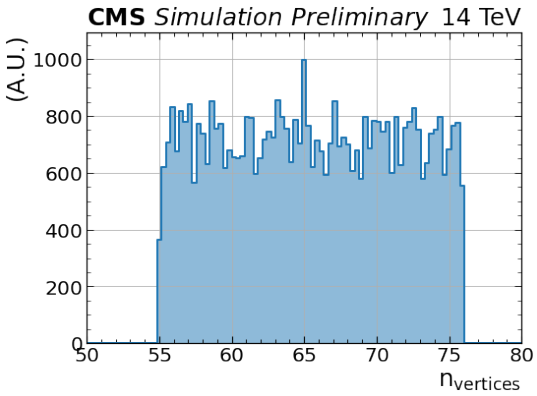



Figure 5.3: Distribution of the number of pileup vertices in the simulated dataset produced for the DeepSC model. Events are created using a full CMS Monte Carlo simulation at 14 TeV.

The second dataset is created specifically to develop and test the particle ID feature of the DeepSC model. The samples of this dataset consist of jets, generated by using quarks and gluons as initial simulated particles, the occurrence of different parton flavours is shown in Fig. 5.4. The $t$-quarks are intentionally removed as the main goal is the light jet versus photon discrimination.

The jet dataset is also created using a full CMS Monte Carlo simulation at 14 TeV. The transverse momentum for the generated partons ranges from 1 to 500 GeV, its distribution is shown in Fig. 5.5 (left). Initially, two different datasets are created: the first one $p_T = [1, 100]$ GeV and the second with $p_T = [1, 500]$ GeV. They are combined together to obtain larger statistics, which explains the distribution shape in Fig 5.5 (left). The parton $p_T$ does not play a crucial role in the model development and testing, a more important variable is the resulting hadron $E_T$, shown in Fig 5.5 (right). During the hadronization process, hadrons with low energies are produced more often, which can be seen from their $E_T$ distribution. The pileup levels are the same as for the $e/\gamma$ dataset: from 55 to 75 additional interactions. To be able to test the discrimination between prompt photons and non-isolated ones, every sample in the jet dataset is required to have at least one photon pair coming from a $\pi^0$.

The simulated transverse energy distributions for photon, electron and jet datasets are shown in Fig. 5.5 (right) as well. The discrepancies between photon, electron, and

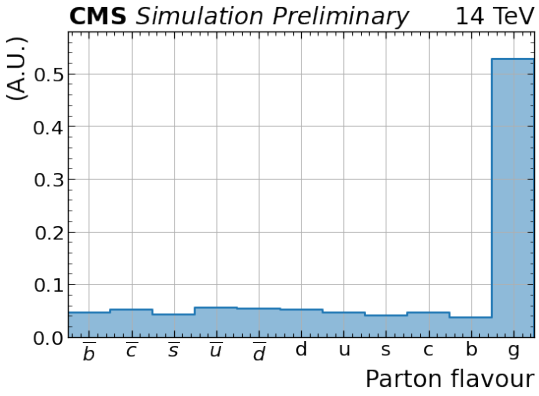



Figure 5.4: Distribution of the flavours of the generated partons in the jet dataset.

hadron $E_{\mathrm{T}}$ spectra are further addressed in Section 5.3.

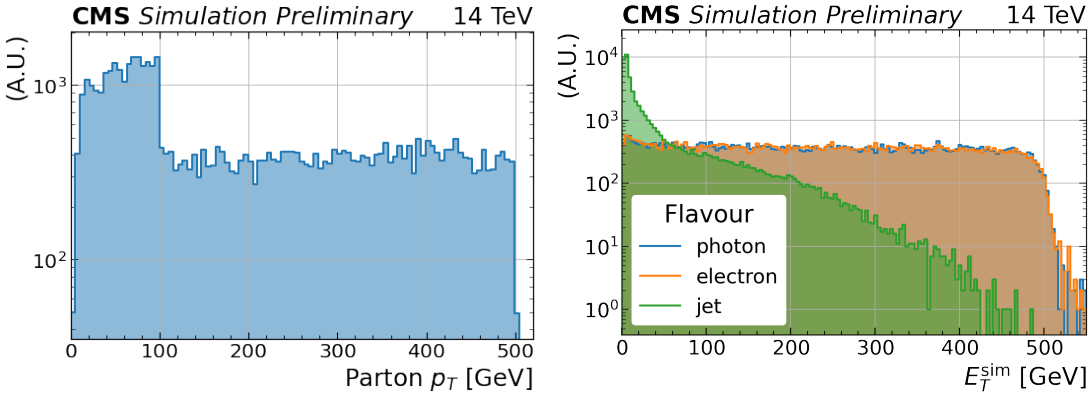



Figure 5.5: On the left: $p_{\mathrm{T}}$ distribution of the simulated partons for the jet dataset. On the right: distribution of simulated transverse energy $E_{\mathrm{T}}$ for hadrons in the jet dataset and for electrons and photons in the expanded $e/\gamma$ dataset.

### The truth definition

The DeepSC model, being a supervised machine learning algorithm, requires accurate identification of the ground truth information. In the case of an SC, it means correctly associating all the PFClusters in the ECAL to the particle they originated from (it can be an electron, photon, or hadron), while excluding the ones representing noise and pileup. It is a non-trivial task as energy deposits coming from different particles can potentially overlap within the same PFCluster.

The first step is to develop a procedure to associate each PFCluster with a generated particle. The goal is to achieve the best possible resolution between energy, evaluated by summing up all the associated PFClusters, and simulated energy of the generated particle.

To perform this association, a new object called a *CaloParticle* is introduced [117]. CaloParticle is defined for electron, photon, hadron, and also out-of-time pileup particle. It tracks the EM shower produced from the simulated particles and records the information about the energy deposits (*simEnergy*) in each ECAL crystal in the form of *simHits*. This information allows precise matching between PFClusters and their corresponding CaloParticles.

Each CaloParticle can produce multiple PFClusters and, consequently, be matched

to several of them, while in this method each PFCluster can only have one associated CaloParticle, even if it contains overlapping energy deposits.

The matching is performed as follows:

- For each **PFCluster**:
  1. For each CaloParticle $k$ containing at least one simHit in a PFCluster, a special matching variable called simFraction is calculated according to Eq. (5.1). It represents the energy fraction left by the CaloParticle $k$ in the considered PFCluster.
  $$\text{simFraction}_k = \frac{\sum_n E_k^{\text{simHit}}}{E_k^{\text{total}}}, \tag{5.1}$$
  where $n$ is the total number of simHits from CaloParticle $k$ in a considered PFCluster, $E_k^{\text{simHit}}$ is the amount of energy deposited by CaloParticle $k$ in the crystal $n$, $E_k^{\text{total}}$ is the total energy deposited in the calorimeter by CaloParticle $k$.
  2. The CaloParticles are sorted based on the simFraction.
  3. The PFCluster is associated to the CaloParticle with the highest simFraction.
- For each **CaloParticle**:
  1. The list of the associated PFClusters based on the criteria previously described is kept. All non-assigned PFClusters are considered to be noise.
  2. The PFClusters are sorted by simFraction.
  3. The PFCluster with the highest simFraction is called *caloSeed*.

As a result of the association procedure, each PFCluster is assigned to a specific CaloParticle, and, in turn, for each CaloParticle a caloSeed is defined. The caloSeed and all the PFClusters associated to the same CaloParticle form an object. The PFClusters that did not pass a pre-defined simFraction threshold are excluded from it. This is done in order to eliminate PFClusters with most of the energy coming from PU or noise. The threshold depends on the energy and position of the caloSeed and is optimized based on a collection of 1M caloSeeds.

From this constructed object the energy of the initial simulated particle can be evaluated by summing up the energies of the included PFClusters. The achieved estimate is compared with the simulated particle energy in order to estimate the ideal resolution.

These results are further compared with the raw energy resolution obtained from Mustache SuperClustering in Fig. 5.6. The estimations performed with the truth-matched PFClusters represent the maximum improvement in the resolution that the DeepSC model can possibly achieve with the described truth definition.

### The training dataset

After the truth definition has been established, a training dataset can be created. Supplying the full ECAL calorimeter as an input to the model will be inefficient and computationally exhaustive, Instead, each input to the DeepSC consists of a caloSeed and PFClusters that fall in the defined rectangular window around it. The dimensions of this window are always chosen to be larger than the area used for the "Mustache" algorithm.

The size of the window depends on the position of the caloSeed in the $\eta$ coordinate. The distributions of $|\Delta\eta|$, $|\Delta\phi|$, the distance from the caloSeed to the window border, are presented in Fig. 5.7. These parameters are defined from the truth-generated information.

The truth labeling of each sample of this dataset proceeds as follows:

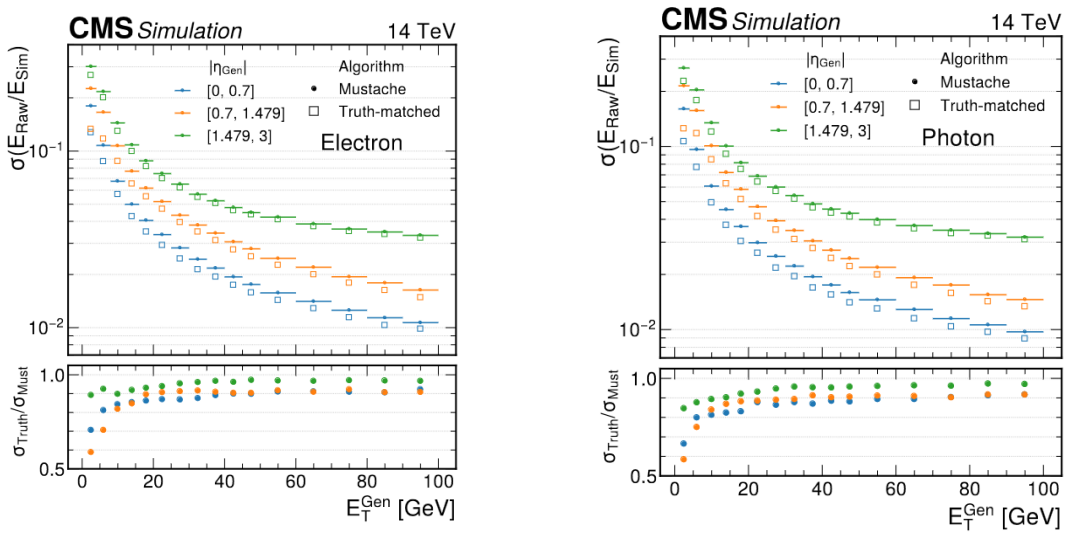



Figure 5.6: The best possible energy resolution that can be obtained with the truth matching compared to the uncorrected energy prediction of the Mustache algorithm for electron (left) and photon (right). Shown in bins of the energy of the generated particle ($E_{\mathrm{T}}^{\mathrm{Gen}}$) [117].

- The list of all PFClusters is ordered by $E_{\mathrm{T}}$.
- The iteration is performed on all the possible caloSeeds (PFClusters with $E_{\mathrm{T}} >$ 1 GeV):
  - If the considered PFCluster is associated with a CaloParticle with at least 1% SimFraction, then it is saved as matched to the CaloParticle and discarded otherwise. The PFCluster is also excluded if it already belongs to a window defined by another caloSeed.
  - If the PFCluster passes the previous criterion, it is considered as a caloSeed and a new window is created around it.
  - All the PFClusters falling inside this window are saved.
- For each PFCluster in the window a truth association is done:
  - If the PFCluster is not associated to the same CaloParticle, it is labeled as out-the-SC.
  - Otherwise, if the PFCluster passes the simFraction threshold, it is labeled as in-the-SC.

For each training sample, the parameters of the caloSeed and all the PFClusters falling in the associated window represent input to the network:

- **PFCluster information**:
  - energy: $E$, transverse energy: $E_{\mathrm{T}}$, coordinates in the ECAL: ($\eta$, $\phi$, $z$), number of crystals, caloSeed flag.
  - Information relative to the seed: $\Delta\eta$, $\Delta\phi, \Delta E$, $\Delta E_{\mathrm{T}}$
  - List of PFRechits for each PFCluster, including their position and deposited energy.
- **Summary window features**: the minimum, maximum and average values of $E_{\mathrm{T}}$, $E$, $\Delta\eta$, $\Delta\phi$, $\Delta E$, $\Delta E_{\mathrm{T}}$ of all the PFClusters in the window.

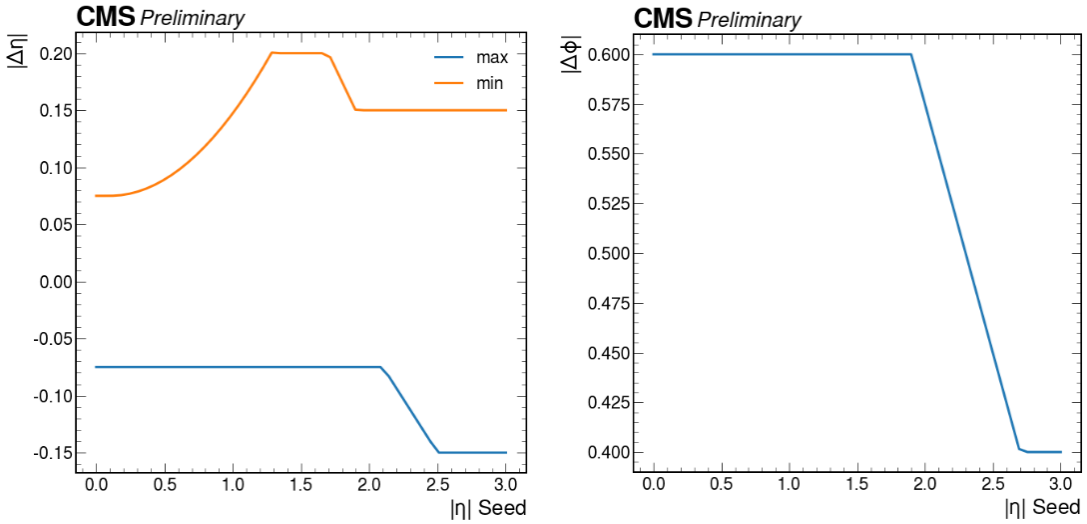



Figure 5.7: Parameters defining the detector window dimensions used to select PFClusters around a seed: $|\Delta\eta|$ (on the left) and $|\Delta\phi|$ (on the right) depending on the $\eta$-position of the seed [117]

### 5.1.2 Model architecture

The DeepSC model is the first ML method developed for SuperClustering reconstruction in CMS. The goal of the network is threefold:

1. Identify which PFClusters should be grouped together to form an optimal SC. Represented by PFCluster classification output in the model.

2. Simultaneously predict the energy correction coefficient in order to improve the resolution. Represented by energy calibration output in the model.

3. Perform particle ID for the reconstructed SC. Represented by window classification output in the model.

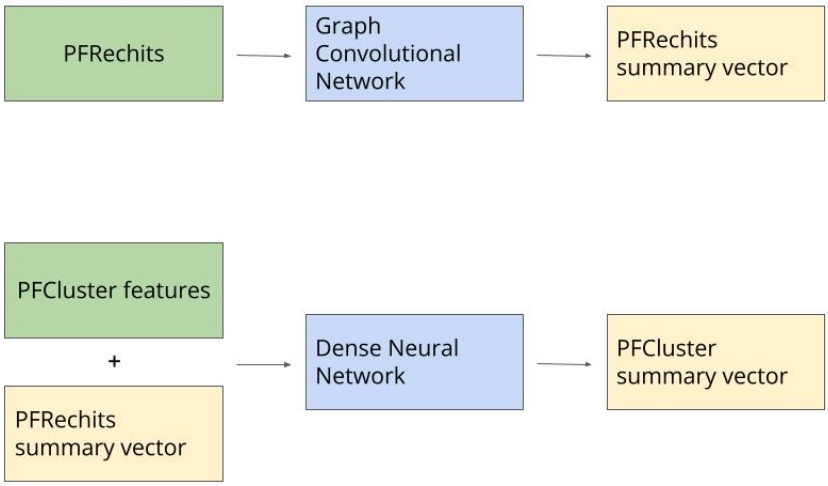
In this document, the focus is on the 1$^{st}$ and the 3$^{rd}$ tasks as the energy correction prediction is still being investigated.

To accomplish these goals, a complex network architecture is developed based on GNNs and SA layers. The full architecture is presented in Fig. 5.8. In order to better understand the role of each of its components, a detailed decoupling of the parts of the model is discussed further:

1. *PFRechits encoding.* All the PFRechits in the input window are processed by a Graph Convolutional Network (GCN) layer, which creates PFRechits summary vectors. One vector is made for each of the PFClusters and it represents the encoded information about PFRechits. The GCN is used in this case to enable communication between PFRechits from close-by PFClusters.

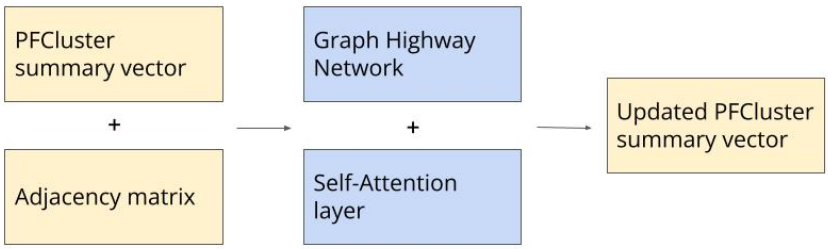
2. *PFCluster features encoding.* PFRechits summary vectors from the 1$^{st}$ step are combined with input PFCluster features and passed through a simple Dense Neural Network (DNN). The resulting vector represents encoded summary information about each of the PFCluster in the window.

3. *Graph building and message-passing.* The next step is to enable information sharing between neighboring PFClusters in the window through message passing. In order

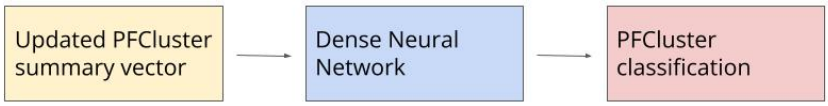



to do this, a graph has to be introduced or, more specifically, its nodes and edges must be defined.

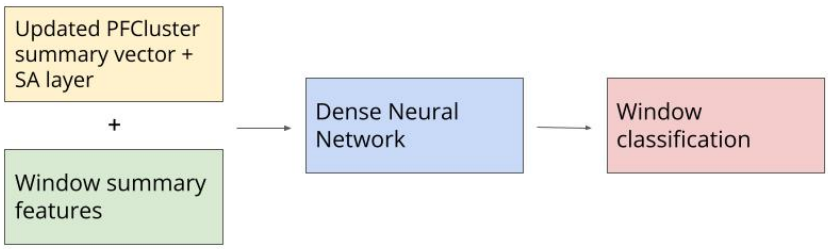
In this model, the nodes of the graph are PFClutsters and the edges (or connections between the nodes) are decoded in the adjacency matrix, which is constructed from the Euclidean distance between PFClusters. The message passing is performed by Graph Highway Network (GHN) and the SA layer that process the created graph. As a result, the PFCluster summary vectors are updated to include information about the neighboring graph nodes.



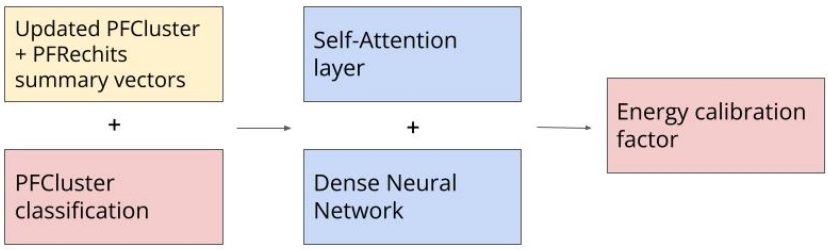

4. *PFCluster classification output.* To obtain the final output for each of the PFClusters, the updated PFCluster summary vectors from step 3 are passed through a DNN.




5. *Window classification output.* For the window classification, the updated PFCluster summary vectors from step 3 are additionally passed through another SA layer in order to encode information from all the PFClusters in the window in a single vector. It is further combined with the input window summary features and passed through a DNN.




6. *Energy calibration factor.* For energy calibration output, the updated PFCluster summary vectors are combined with the PFRechits summary vector from step 1 and cluster classification output and passed through another SA layer and a DNN.




In this kind of architecture every part of the model has a specific and dedicated role:

- Dense layers are used to find the patterns in the data and extract the summary vectors or to provide the necessary output dimensions
- SA layers help the network to combine information from multiple close-by nodes in one vector by focusing on the most important features.
- GCN and GHN are used to enable message-passing between the close-by PFClusters.

The final outputs of the model are 1) **for each PFCluster in the input window**: a probability score to belong to an SC, which is compared to the true labeled information (out-the-SC or in-the-SC); 2) **for each window**: energy calibration factor and window score indicating from which type of caloParticle an SC has originated from.

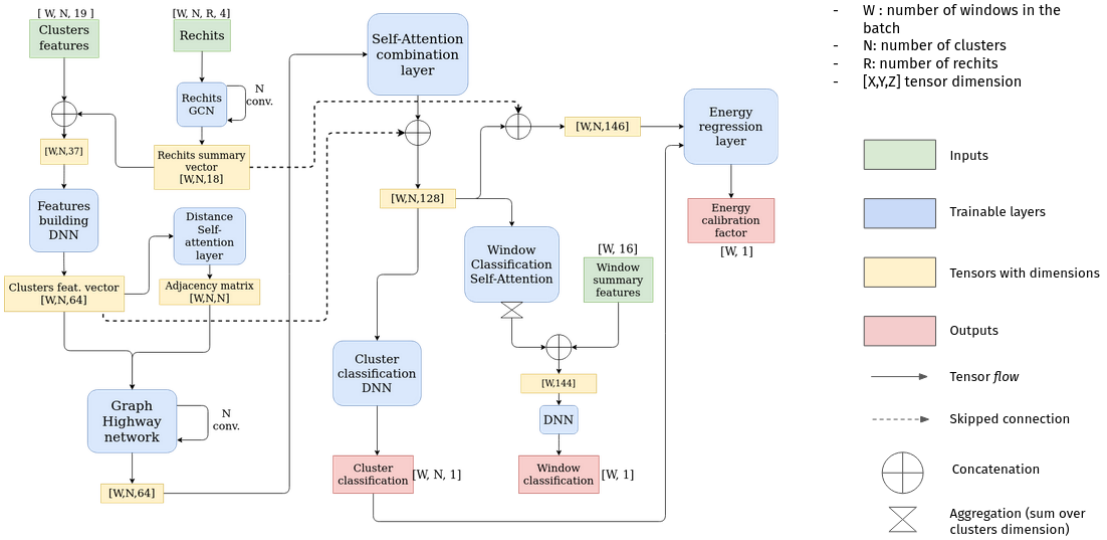



Figure 5.8: DeepSC model architecture. The input to the network consists of selected features and rechits of the clusters that fall into a predefined geometrical window. Using dense layers the latent features are extracted from the initial input, they are processed and combined together by different types of graph architectures: Graph Convolutional Network (GCN) and Graph Highway Network (GHN). Self-Attention layers are used as well to help the network with focusing on the most important features/inputs. The final outputs are the following: information on whether each of the clusters belongs to the SuperCluster (cluster classification), the type of particle from which the SuperCluster originated (window classification), and energy correction (energy calibration factor) [117].

## 5.2 Performance for the energy resolution

The primary objective of the new model is to accurately identify clusters belonging to the SC while effectively filtering out noise and PU. The dataset used for this task comprises 2M electron samples and 2M photon samples. They are divided into training, validation, and test sets, with proportions of 50%, 30%, and 20% respectively. The network is trained using Adam optimizer with a batch size of 512. The binary cross-entropy loss for the PFCluster classification and the categorical cross-entropy for window classification are used. As the jet dataset is not used in this training, the window classification is performed between photon, electron, and noise. The training and validation losses are shown in Fig. 5.9.

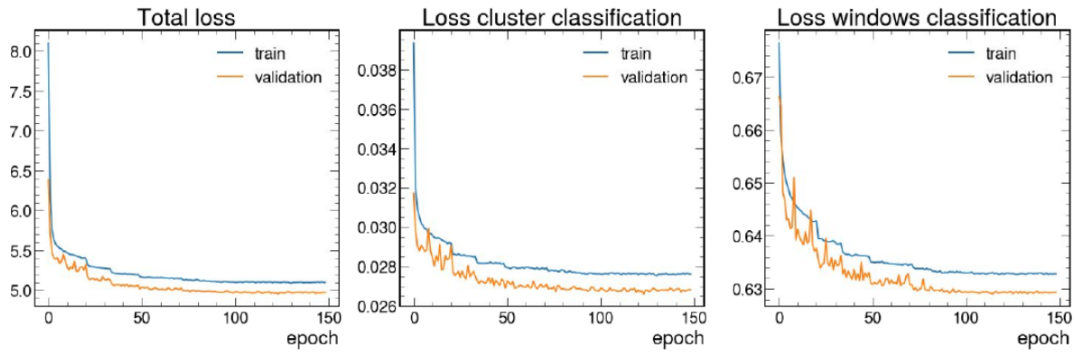



Figure 5.9: Training and validation losses for the DeepSC model [117].

The performance of the model in terms of energy resolution is compared to the "Mustache" algorithm. The evaluation is done using raw energy results, without any additional energy corrections applied to either of the algorithms.

The resolution is calculated by analyzing the distribution of the ratio between the summed energies of all PFClusters selected by the algorithm and the simulated particle energy. The resolution is determined by $\sigma = [\text{quantile}(0.84) - \text{quantile}(0.16)]/2$. The results are presented in bins of $E_{\mathrm{T}}$ of the generated particle and $\eta$ in Fig 5.10 and in bins of $E_{\mathrm{T}}^{\mathrm{Gen}}$ and number of PFClusters in the window in Fig. 5.11 for both electrons and photons. Across all the presented results, the DeepSC model consistently outperforms the "Mustache" algorithm. The advantage is the clearest for low energies, where "Mustache" reconstruction is contaminated by PU and noise.

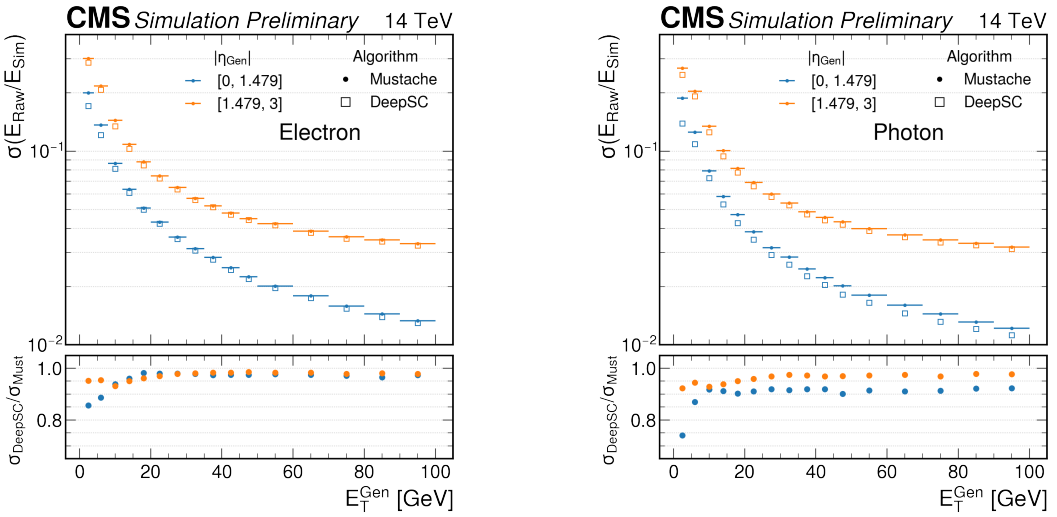



Figure 5.10: Energy resolution comparison between the DeepSC and "Mustache" algorithms in bins of generated particle energy $E_{\mathrm{T}}^{\mathrm{Gen}}$ and $\eta$ for electrons (right) and photons (left) [117].

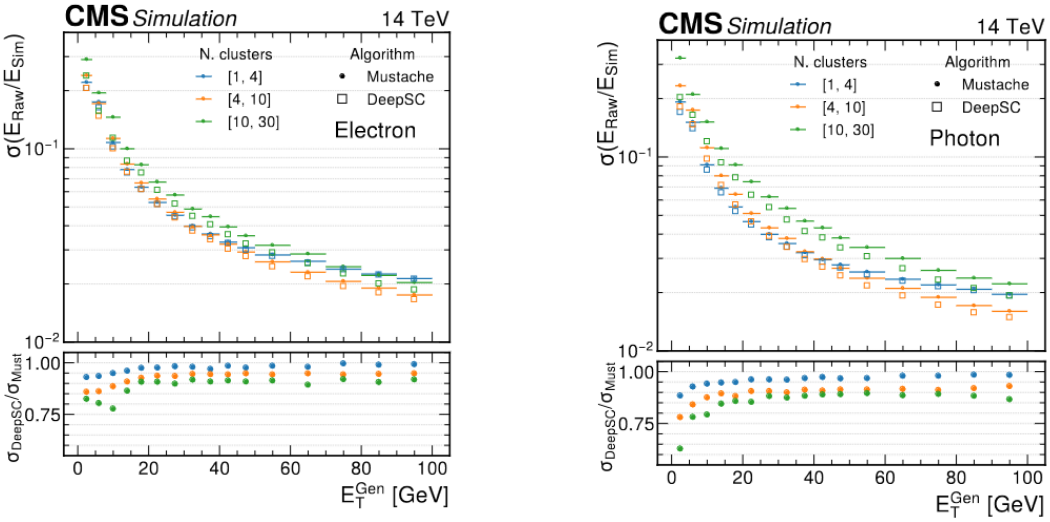



Figure 5.11: Energy resolution comparison between the DeepSC and "Mustache" algorithms in bins of generated particle energy $E_{\mathrm{T}}^{\mathrm{Gen}}$ and number of PFClusters in the window for electrons (right) and photons (left) [117].

Furthermore, the results are also shown for various simulated PU levels, as presented in Fig. 5.12. The DeepSC algorithm achieves greater robustness to the PU, particularly at low energies.

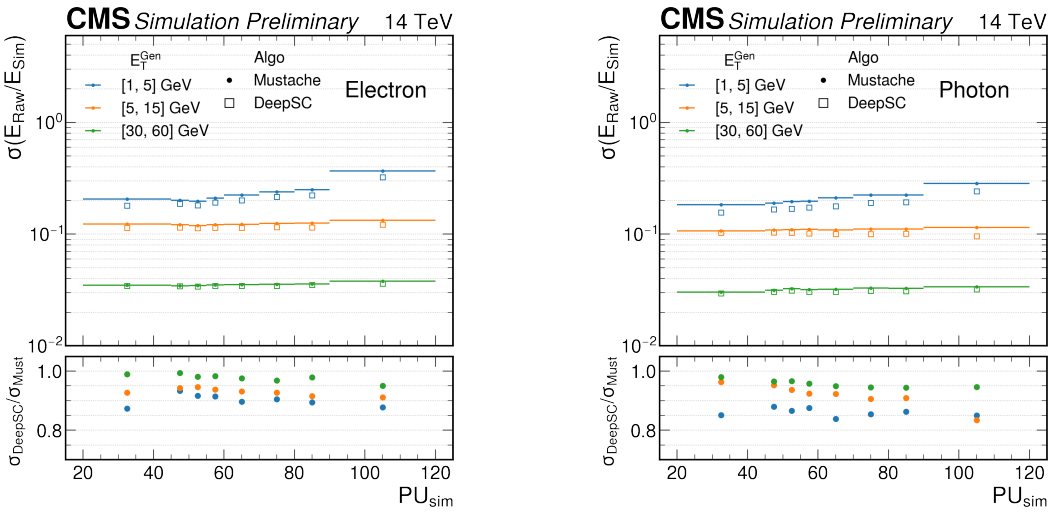



Figure 5.12: Energy resolution comparison between the DeepSC and "Mustache" algorithms in bins of the number of pileup vertices for electrons (right) and photons (left) [117].

## 5.3 Particle flavour identification

After the excellent results for the energy resolution with the DeepSC model were achieved, the network was further developed to include particle ID, classifying SCs between photons, electrons, and jets.

As discussed in Chapter 2, current $e/\gamma$ selection in particle flow is done based on various ML methods: a BDT for electrons and a DNN for photons. Both of these models use high-level information such as the final object $p_{\mathrm{T}}$, its shower spreads, its energy deposits in different detectors, etc.

In the DeepSC model, the particle ID is performed using low-level information such as PFRechits. Moreover, the classification of the objects is performed only based on their signatures in the ECAL without taking into account other detectors (tracker, HCAL).

Two primary objectives drive the development of the particle ID in the DeepSC:

1. **Identification of PFClusters originating from jets**. The SuperClustering algorithms are specifically designed for reconstructing electrons and photons and are not applied to jets. However, hadrons can deposit energy in the ECAL, serving as a background for prompt photons and electrons (discussed in more detail in Chapter 2). The DeepSC model aims to help mitigate this background with the new feature.

2. **Discrimination between photons and electrons**. The presence of a distinct track signature in the pixel tracker serves as the main indicator for discriminating between photons and electrons as discussed in Chapter 2. The DeepSC model does not aim to substitute this approach but rather provides additional information. It can be especially beneficial for cases where the track information is absent or not reconstructed properly.

### 5.3.1 Training details

For the particle ID development, the model uses the jet dataset and high-energy $e/\gamma$ dataset with equal proportions between all three flavours. Overall, the samples are distributed as 900k/200k/100k for the training, validation, and test datasets respectively.

The input windows are created in the same way as discussed in Section 5.1.1, where hadrons represent caloParticles for the jet dataset. The model target for particle ID is a hot-encoded vector of size 3 ([1, 0, 0] for jet, [0, 1, 0] for electrons, and [0, 0, 1] for photon). The output of the model in this case is called window classification and, in accordance with the target, it is a vector of size 3 as well, where each entry evaluates a window score, indicating the likelihood of an SC to originate from a hadron (jet score), electron (electron score), and photon (photon score), respectively. Each of these values can vary from 0 to 1 and they sum up to 1.

As the DeepSC model has been previously optimized on the $e/\gamma$, the transfer learning technique (see Chapter 4) is used for particle ID training. In this way, the energy resolution performance for EM particles is not degraded while adding the new feature. This process is carried out in three steps:

1. **PFClustering classification training**. The model is only trained on the $e/\gamma$ samples as described in the previous section. In this way, the optimal network weights for the PFCluster classification are obtained.

2. **Freezing the weights**. The weights corresponding to the network components responsible for PFCluster classification and energy regression are frozen. This means that they will remain unchanged during subsequent training stages.

3. **Particle ID training**. The second training phase is performed using the combined dataset comprising photons, electrons, and jets. During this training, only the network weights directly related to window classification are optimized. This approach results in the best performance for energy resolution and particle ID.

Table 5.1 shows a comprehensive overview of the model's architecture, the number of parameters in each layer, and whether they are trainable for particle ID.

| **Layer** | $N_{\text{weights}}$ | **Trainable** |
|---|---|---|
| GCN (1) + DNN (2) + SA-layer (3) | 30,775 | no |
| GHN (3) | 33,344 | no |
| DNN (4) | 12,481 | no |
| SA-layer (5) | 41,344 | yes |
| DNN (5) | 3,427 | yes |
| SA-layer (6) + DNN (6) | 47,349 | no |

Table 5.1: The overview of the layers in the DeepSC model. Numbers of weights for each layer are given and whether they are trainable for particle ID. The numbers in the parentheses after the name of the layer indicate the step of the model description from Section 5.1.2.

The model is trained for 10 epochs using Adam optimizer (learning rate = 0.001) and a batch size of 512. It does not require a lot of epochs as it has been already optimized

and only a small amount of network weights changes. Categorical cross-entropy is used as a loss function for the window classification.

The results of the model training are presented in Fig. 5.13. These plots illustrate the distribution of the window scores output for different flavors: jet scores for $e$, $\gamma$, and jet datasets (left); and photon scores for $e$, $\gamma$ datasets (right).

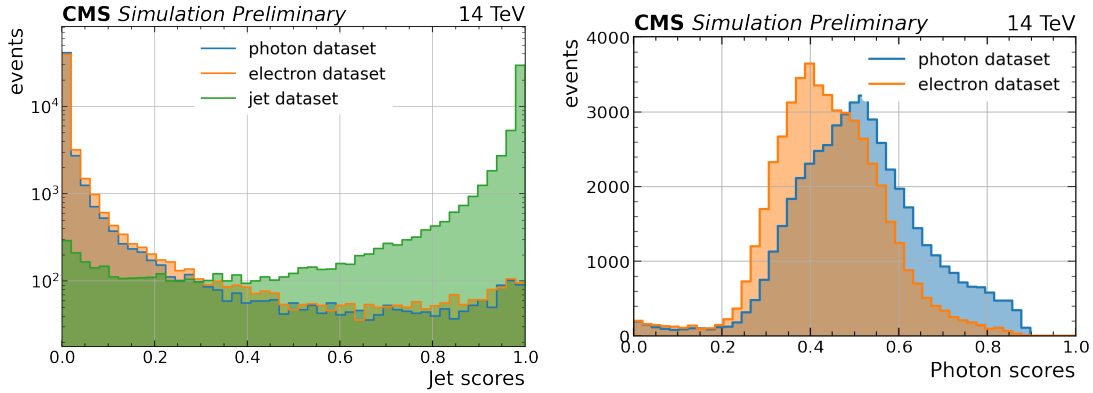



Figure 5.13: Jet (photon) likelihood score distributions obtained by the DeepSC model applied on the photon, electron, and jet (photon and electron) datasets presented on the right (left).

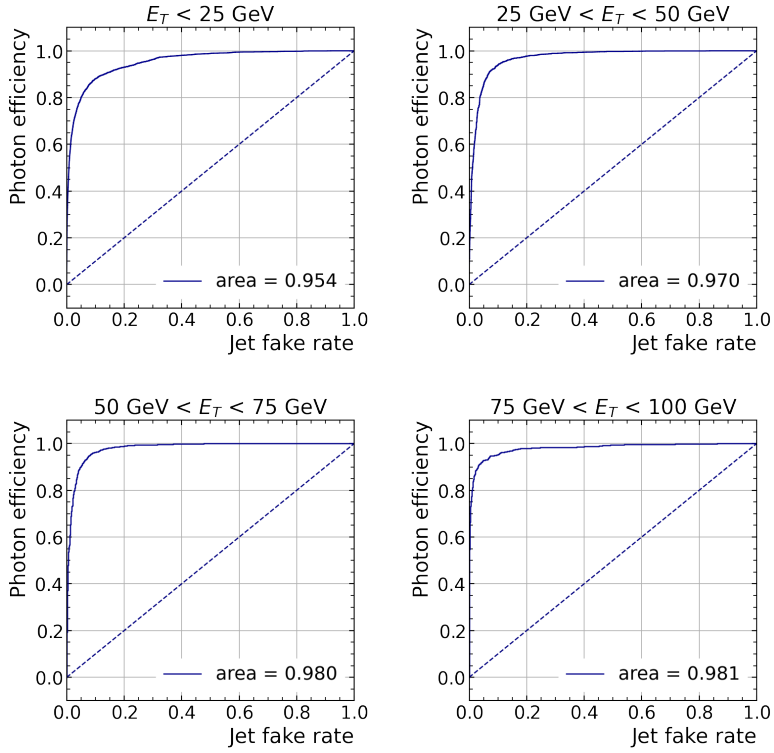



Figure 5.14: ROC curves obtained from the classification of the SuperClusters originated from photon against jet performed with DeepSC model. The discrimination variable is chosen to be the jet score. The ROC curves are shown in bins of the transverse energy $E_{\mathrm{T}}$ of the caloParticle (photon or hadron).

The discrimination capabilities of the DeepSC model are quantified using Receiver Operating Characteristic (ROC) curves [118]. They represent the true positive rate (TPR, rate of correctly identified samples or signal efficiency) against the false positive

rate (FPR, rate of misidentified samples or fake rate) at different thresholds set on the predicted window scores. The area under the ROC curve (AUC) provides an evaluation of performance along all the possible thresholds. For the perfect classifier AUC = 1 or 100%. The ROC curves for photon vs. jet discrimination in different caloSeed $E_{\mathrm{T}}$ bins are presented in Fig. 5.14, where the jet score is used as a discriminating variable. Similarly, the ROC curves for photon vs. electron discrimination are shown in Fig. 5.15, where the photon score is used as a discriminating variable.

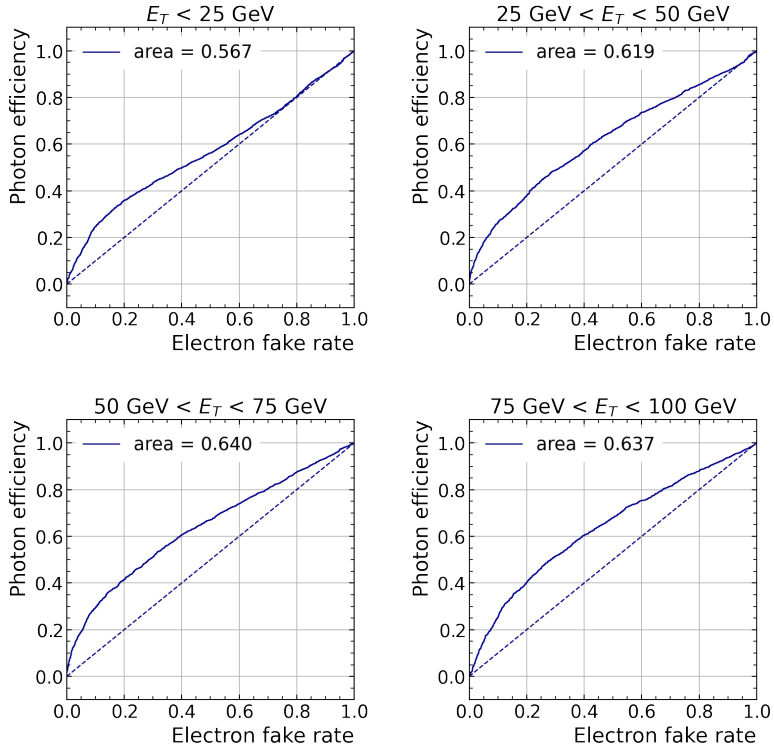



Figure 5.15: ROC curves obtained from the classification of the SuperClusters originated from photon against electron performed with DeepSC model. The discrimination variable is chosen to be the photon score. The ROC curves are shown in bins of the transverse energy $E_{\mathrm{T}}$ of the caloParticle (photon or electron).

The DeepSC model shows excellent performance for photon vs. jet discrimination: AUC > 95% for all energy bins between 1 and 100 GeV with the maximum value at 98% for $E_{\mathrm{T}}$ = [50, 75] GeV. Distinguishing between electrons and photons is more challenging for the model considering that only information from the ECAL is taken into account. Nevertheless, the DeepSC network still demonstrates noteworthy discrimination in this case: AUC > 56% for all energy bins between 1 and 100 GeV with the maximum value at 64% for $E_{\mathrm{T}}$ = [50, 75] GeV.

### 5.3.2 Dataset energy re-weighting

One potential problem for the particle ID that has to be addressed is the difference in the $E_{\mathrm{T}}$ spectra between $e/\gamma$ and jet datasets. The hadron $E_{\mathrm{T}}$ distribution in Fig. 5.5 (right) is skewed to the low energies, while for photons and electrons, the distribution is flat.

This discrepancy will further translate into the total energy of the input, calculated as the sum of energies of all PFClusters in the selected input window, as shown in Fig. 5.16 (left). Consequently, the network may take into account this information when making

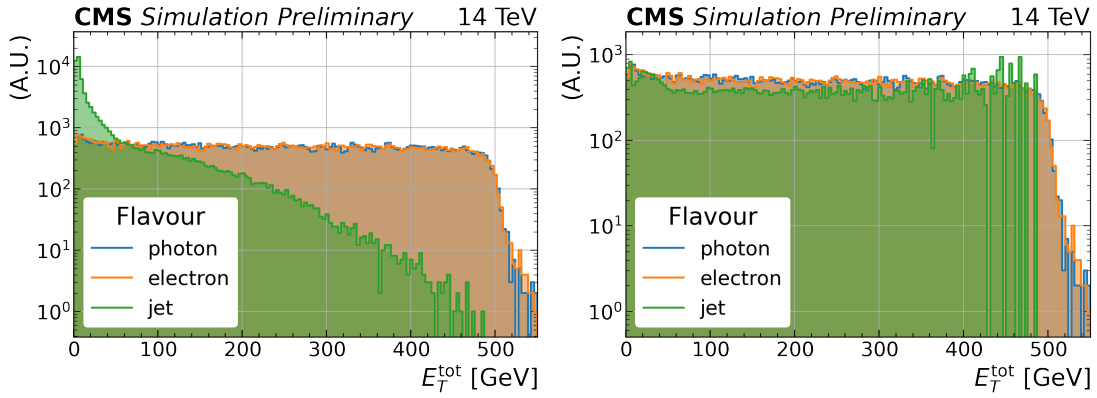



Figure 5.16: The distributions of the transverse energy sum ($E_{\mathrm{T}}^{\mathrm{tot}}$) of PFClusters in input windows for the DeepSC model. They are shown for photon, electron, and jet samples without (left) and with (right) re-weighting coefficients applied on the jet dataset.

a decision about the particle ID, and for any low-energy SC, it will be more likely to assign a higher jet score.

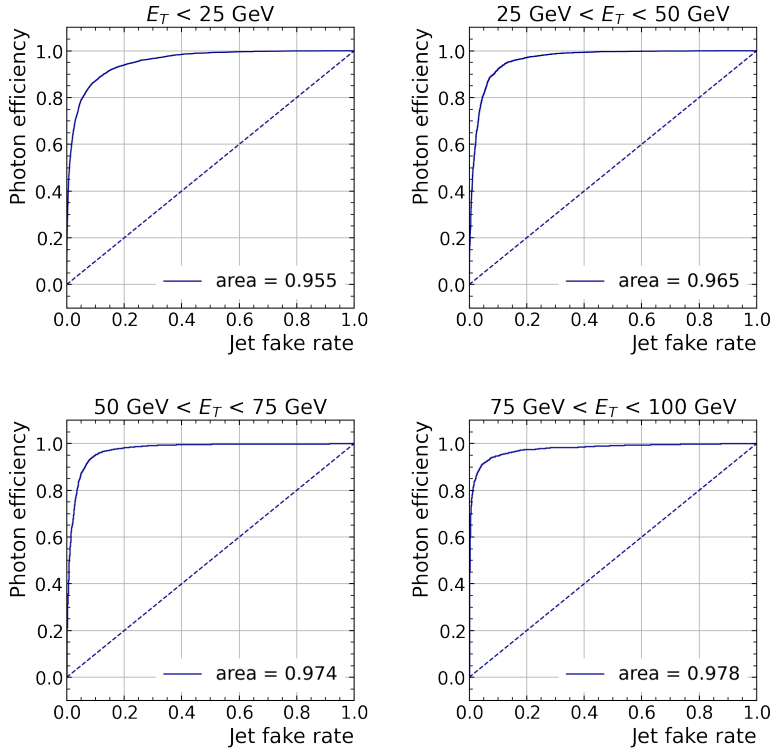



Figure 5.17: ROC curves obtained from the classification of the SuperClusters originated from photon against jet performed with DeepSC model. The jet dataset is re-weighted for the training based on the $E_{\mathrm{T}}$ of the caloSeed. The discrimination variable is chosen to be the jet score. The ROC curves are shown in bins of the transverse energy $E_{\mathrm{T}}$ of the caloParticle (photon or hadron).

In order to avoid this bias and create a robust network, the re-weighting coefficients for jet samples are calculated in bins of the caloSeed $E_{\mathrm{T}}$. For a specific energy range $\Delta$,

the coefficient is estimated as follows:

$$w_\Delta = \frac{N_\Delta^{\text{ele}}}{N_\Delta^{\text{jet}}}, \tag{5.2}$$

where $N_\Delta^{\text{ele}}$, $N_\Delta^{\text{jet}}$ are the numbers of samples with caloSeed energy falling in $\Delta$ in the electron and jet datasets, respectively.

The re-weighted dataset obtained by applying the appropriate coefficients from Eq. (5.2) on the samples is shown in Fig. 5.16 (right). The new $E_{\text{T}}^{\text{tot}}$ distributions are similar for all three flavours.

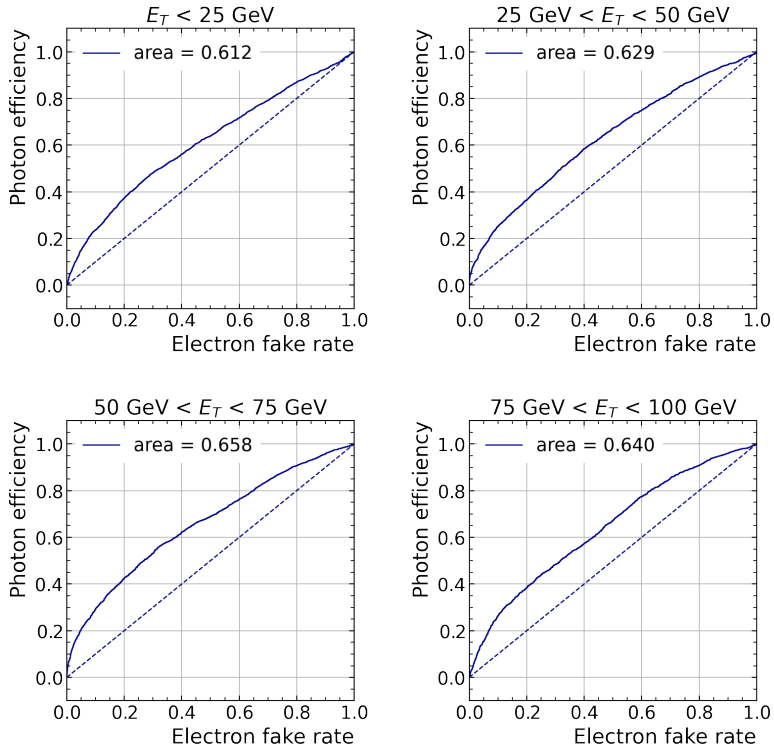



Figure 5.18: ROC curves obtained from the classification of the SuperClusters originated from photon against electron performed with DeepSC model. The jet dataset is re-weighted for the training based on the $E_{\text{T}}$ of the caloSeed. The discrimination variable is chosen to be the photon score. The ROC curves are shown in bins of the transverse energy $E_{\text{T}}$ of the caloParticle (photon or electron).

These coefficients are incorporated into the model as the weights of the loss function for each of the jet data samples. The model is subsequently re-trained. The obtained ROC curves are shown in Fig. 5.17 for jets vs. photons and in Fig. 5.18 for photons vs. electrons. The performance slightly differs compared to the baseline non-weighted training: for electron vs. photon it is improved with weights by 7%, 1.5%, 3% and 0.5% for $E_{\text{T}} < 25$ GeV, $E_{\text{T}} = [25, 50]$ GeV, $E_{\text{T}} = [50, 75]$ GeV and $E_{\text{T}} = [75, 100]$ GeV respectively. The discrepancy for the jet vs. photon discrimination is less than 1% for all energy bins.

### 5.3.3 Comparison with particle flow selection

The final DeepSC model trained for particle ID is incorporated into the software of the CMS experiment and further compared to the currently used DNN for photon vs. jet discrimination (discussed in Section 2.4.5). Both models are evaluated on the two

additional datasets (consisting of photons and jets), in which particles are generated with $p_{\mathrm{T}} = [1, 100]$ GeV, and the pileup levels are from 55 to 75. The samples undergo the full chain of reconstruction in the CMS software. The ROC curves for both models in bins of the reconstructed particle $p_{\mathrm{T}}$ are presented in Fig. 5.19.

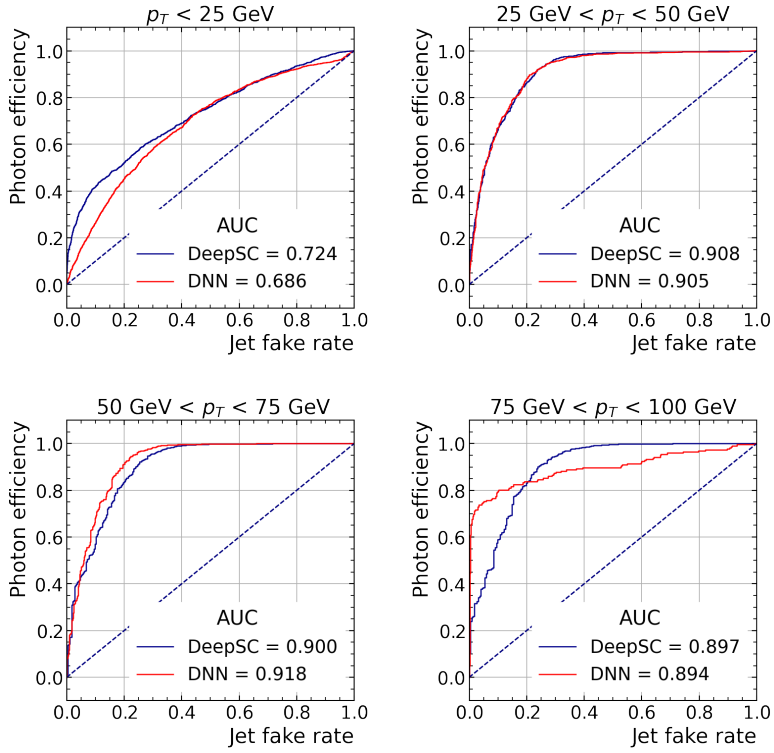



Figure 5.19: ROC curves obtained from the classification of the SuperClusters originated from photon against jet performed with DeepSC model and particle flow DNN. The discrimination variable for the DeepSC model is chosen to be the jet score. The ROC curves are shown in bins of the transverse momentum $p_{\mathrm{T}}$ of the caloParticle (photon or hadron).

The DeepSC model slightly outperforms DNN in three energy bins: by 5%, 0.3%, and 0.3% for $p_{\mathrm{T}} < 25$ GeV, $p_{\mathrm{T}} = [25, 50]$ GeV, and $p_{\mathrm{T}} = [75, 100]$ GeV, respectively. For $p_{\mathrm{T}} = [50, 75]$ GeV the classification obtained from the DeepSC model is worse by 2%. It is an outstanding result as the DNN of particle flow additionally uses information from the tracker and HCAL, unlike the DeepSC model.

The window scores of the DeepSC can be further used as one of the input variables for the selection algorithms either at the level of particle flow reconstruction or physics analysis. In order to demonstrate the potential of this approach, three additional BDTs are trained for particle classification. They use a common set of input variables (similar to the particle flow DNN as discussed in Section 2.4.5): $\phi$, $\eta$, hadronic over electromagnetic ratio, isolation variables, and shower shapes. Each of them incorporates different additional inputs:

1. *BDT with PFDNN*: the output of the particle flow DNN classification.
2. *BDT with DeepSC ID*: the window scores from DeepSC.
3. *BDT with both*: both particle flow DNN variable and window scores from DeepSC.

The results are presented in Fig. 5.20. The obtained AUC values are the following: 86.7% for BDT with PFDNN, 90.7% for BDT with DeepSC ID, and 91.1% for BDT with

both. For a background efficiency of 10%, the signal efficiency is improved by more than 20%. A remarkable improvement brought by adding the DeepSC ID output indicates the ability of the model to bring new information to the reconstruction chain.

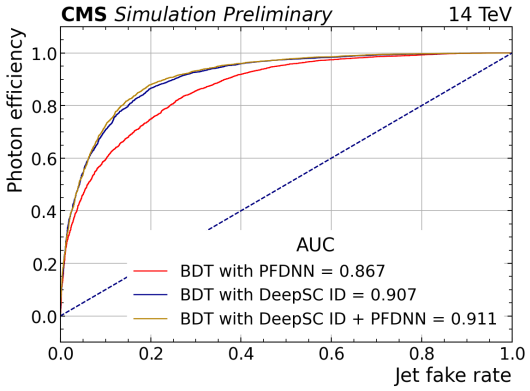



Figure 5.20: ROC curves obtained from three BDTs classifying of photons against jets. The BDTs are trained with a common set of input parameters and, additionally, the output of the particle flow DNN classification (BDT with PFDNN), the window scores from DeepSC (BDT with DeepSC IDs), both of these variables (BDT with both).

## 5.4 Conclusion and perspectives

A novel deep learning method for SuperClustering reconstruction has been developed. The DeepSC network performs three different tasks: 1) it creates an optimized SuperCluster by separately classifying PFClusters; 2) it predicts the energy calibration factor that accounts for the energy losses in the ECAL; 3) it performs a particle ID for each reconstructed SuperCluster, indicating whether it has originated from an electron, photon or jet.

The performance of the PFCluster classification can be evaluated from the energy resolution between the energy reconstructed with DeepSC and the simulated energy of the particle. The reconstructed value is estimated by summing all the energies of the PFClusters selected by the network. The obtained results were compared with the "Mustache" algorithm, currently used in CMS for SuperClustering reconstruction. The network shows superior performance, especially for the low-energy and high-pileup regions. It highlights the apparent benefits of the DeepSC model for SuperClustering reconstruction, especially for the end of Run 3 and HL-LHC era.

The DeepSC model will be further tested inside the CMS software to estimate the performance in terms of timing and computing efficiency. The comparison between the DeepSC model and the "Mustache" algorithm with the application of energy corrections is being investigated as well. Separately, additional effort will be put into developing the energy regression part of the network.

The focus of this thesis was on the particle ID part of the DeepSC model. This feature was fully developed as part of the thesis, starting from its addition into the initial model (using transfer learning) to its implementation in the global CMS software and comparison of the results.

The particle ID allowed the model to predict whether the reconstructed SuperCluster resulted from a photon, electron, or hadron from the energy patterns in the ECAL. The model shows excellent performance for jet vs. photon discrimination, and remarkably achieves some discrimination for electron vs. photon using only calorimeter information.

The particle ID performance of the model for jet/photon classification was compared with the dense neural network used in particle flow. The DeepSC shows compatible

performance. This is a remarkable result as the particle flow approach uses additional information from the HCAL and tracker to make the discrimination.

In future prospects, there are multiple things that can be done for particle ID. Firstly, incorporating the data from the tracker and HCAL in the model can be explored. Secondly, the output of the particle ID can be tested as one of the input variables for the dense neural network of particle flow. In this case, the proof-of-concept study was already performed by adding the DeepSC model output to an additional BDT classifier, which resulted in significantly improved jet versus photon discrimination results. Thirdly, the usage of electron vs. photon discrimination can be investigated for the analysis where track information is lost or not properly reconstructed.

Overall, the DeepSC model shows very promising results for further usage within the CMS experiment. It opens new possibilities for improving the resolution of electromagnetic object reconstruction and the precision of physics analyses.

# 6 Clustering reconstruction for standalone particles in the ECAL with Deep Learning

The process of measuring electromagnetic (EM) objects in ECAL involves a series of sophisticated steps, beginning with the reconstruction of energy deposits in the calorimeter and resulting in the formation of the final physical object from the obtained information. A detailed description of the offline reconstruction procedure can be found in Chapter 2.

The focus of this Chapter is the intermediate step called *clustering*. It aims to determine the kinematic properties of individual photons and electrons entering the calorimeter from the energy patterns they leave in the detector. The current CMS operation uses the PFClustering algorithm, which analyzes the combination of neighboring calorimeter crystals (called a cluster) to evaluate both the energy and the entry point of the initial particle.

While the traditional approach is proven to be efficient and provides excellent resolution for energy and position reconstruction, it has a limited ability to accurately distinguish two close-by photons, which can result in several issues. For example, the PFClustering algorithm struggles to differentiate isolated photons ($\gamma$) from neutral pions ($\pi^0$), as the latter produce two photons that mimic the energy pattern of an isolated $\gamma$ in the calorimeter. Another example is the search for exotic Higgs boson decay $H \to aa \to 4\gamma$ [1], where $a$ is a light scalar or pseudoscalar particle. In this case, the two photons are often reconstructed as a single particle in the calorimeter as well.

The aim of this work is to address the discussed limitation of the PFClustering algorithm while also improving the performance in terms of energy and position resolution. This is achieved by developing a novel machine learning (ML) model, called *DeepCluster*, that leverages advanced deep learning techniques such as convolutional (CNNs) and graph (GNNs) neural networks (described in Chapter 4).

The first part of this Chapter introduces a simulation of a simplified calorimeter that is used in the subsequent work. Section 6.2 presents the performance of the PFClustering algorithm on the datasets simulated with this detector. The implementation of the first version of the DeepCluster model (one-shot CNN), along with encountered setbacks, is described in Section 6.3. To cope with these setbacks, an improved DeepCluster model (two-step net), containing two consecutively applied networks, is introduced in Section 6.4 and the detailed development process is presented. First, for both of these underlying networks, CNNs are used and later the second one is changed to GNN in order to mitigate the energy overestimation. Sect ion 6.5 discusses the optimization techniques implemented to achieve the best-performing model. Finally, a comparison of the results between the DeepCluster model and the PFClustering algorithm is provided in the last section, including analysis of photon, electron, and pion datasets.

## 6.1 Dataset simulation

To develop a new reconstruction method, a dedicated simulation of a simplified ECAL detector, called a *toy calorimeter*, is used. It is created with Geant4 software [119] that provides access to all the necessary information, such as energy deposits per crystal, and enables easy control over kinematic properties (energy, direction) and particle type of the generated objects.

The detailed characteristics of the toy calorimeter are presented in Table 6.1, including the comparison to the barrel part of the ECAL. The complex cylindrical geometry of the actual detector is simplified to a rectangular shape, while the material and the size of the crystals stay identical. The toy calorimeter consists of 51 x 51 crystals that do not have any tilt.

| **Description** | **Value (toy calorimeter)** | **Value (ECAL barrel)** |
|---|---|---|
| Material | $PbWO_4$ | $PbWO_4$ |
| Crystal size | 2.2 x 2.2 x 23 $cm^3$ | 2.2 x 2.2 x 23 $cm^3$ |
| Detector size | 51 x 51 crystals | 360 x 170 crystals |
| Shape | Rectangular | Cylindrical |
| Magnetic field | None | 3.8 T |
| Material before detector | Air | ECAL tracker (silicon) |

Table 6.1: Description of the toy calorimeter characteristics and comparison to the ECAL barrel.

In this document, the term *dataset* refers to the simulated data. One entry to the dataset is called a *sample*. Each sample consists of individual *events*, representing instances of the simulated EM objects. For example, a photon dataset can consist of several samples, each of them containing two particles or two events.

Using the toy calorimeter, first, a dataset with only one photon per sample is created. In this dataset, photons ($\gamma$) are simulated perpendicularly to the detector surface. Their energies are uniformly distributed within the range of 1–100 GeV. The position at which the particle enters the calorimeter is chosen randomly, only avoiding the edges of the detector (the central part is used: 80% of the toy calorimeter). This ensures that the majority of the deposited energy remains within the detector. An event display of the Geant4 simulation is presented in Fig. 6.1, showing two different simulated photons in $(z, x)$ plane on the left and $(x, y)$ plane on the right.

In order to train and test the DeepCluster model, in the post-processing phase (using pandas [120] and numpy [121] libraries from Python programming language), two separate datasets are created from this original set, targeting different purposes:

1. **Single-photon dataset**.
   Each entry of the dataset consists of one photon. It is used both for training and as a primary check of DeepCluster performance regarding coordinate and energy resolution. It corresponds to the case of isolated particles in the calorimeter.

2. **Two-photon dataset**.
   Each entry is created by superimposing two different samples from the original set. This represents two separate photons in the calorimeter. In this dataset, only

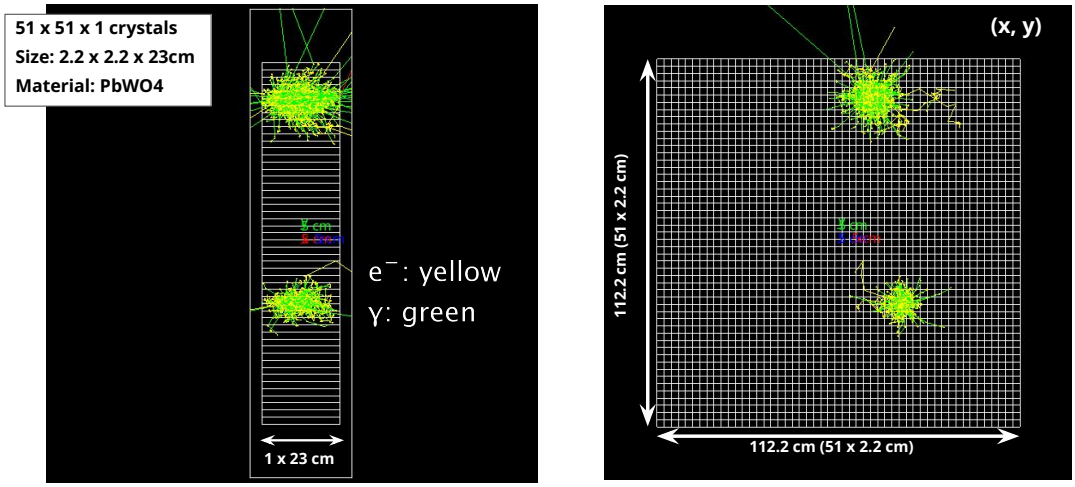



Figure 6.1: Event displays (side view $(z, x)$ on the left and front view $(x, y)$ on the right) of particle simulation done with Geant4 software. The toy calorimeter consists of 51 x 51 $PbWO_4$ crystals with a size of 2.2 x 2.2 x 23 $cm^3$. The particles are directed perpendicularly to the surface of the calorimeter. The displays show two simulated photons.

particles with positions located within a maximum distance ($\Delta R = \sqrt{\Delta x^2 + \Delta y^2}$) of 3 calorimeter crystals from each other are intentionally selected (signature mimicking $\pi^0$ decay). It is also used for training and to estimate the ability of the DeepCluster network to separate two close-by photons.

As discussed in Chapter 4, in order to avoid overfitting, three different sets should be created. In this work, the training set consists of randomly chosen and mixed 600k samples of the 1st dataset and 300k samples of the 2nd one. The validation set has 200k entries from the 1st dataset and 100k entries from the 2nd. The test set is separate for each of the cases: 100k samples for the 1st one and 50k for the 2nd one.

Moreover, to have a thorough understanding of the performance of the algorithms for various practical uses, additional datasets, only used during the evaluation stage, are created:

1. **Single-electron dataset**.
   The dataset is created in the same way as the single-photon dataset but using electrons instead.

2. **Two-electron dataset**.
   The dataset is created in the same way as the two-photon dataset but using electrons instead.

3. **Multiple-particle electron dataset**.
   Each entry is created by superimposing up to 6 electrons. Each particle has energy in the range of $E = [1, 100]$ GeV. Overall consists of 30k samples.

4. **Pion dataset**.
   To create this dataset, $\pi^0$ particles with $E = [1, 100]$ GeV, shot perpendicularly at the toy calorimeter from a distance of 130 cm ($\approx$ distance between the interaction point and ECAL inner surface) are used. Overall consists of 180k samples.

Further in this chapter, the performance of the DeepCluster model on the described additional datasets is shown, even though it is not specifically trained on them.

### Simulation validation

To validate the simulation of the toy calorimeter, the energy deposit profile is investigated and compared to the simulation of the actual ECAL. In order to do it, 1,000 electrons at 100 GeV are shot at the center of the detector, and the average deposited energy in each cell of the 5x5 crystal matrix around the central crystal is calculated. The results are shown in Fig. 6.2 (left) and they are compared to the ones achieved with the real ECAL simulation presented in Fig. 6.2 (right).

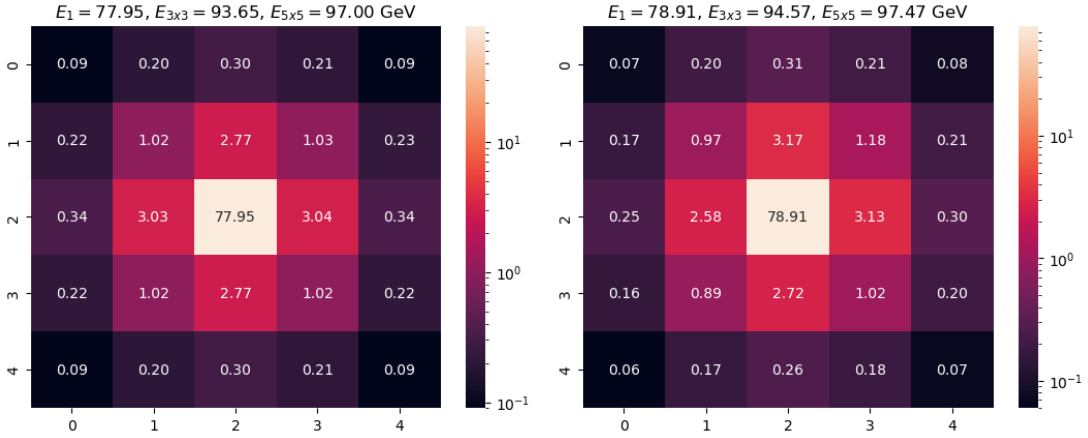



Figure 6.2: Energy deposit profiles from the toy calorimeter (left) and from Geant4 simulation of ECAL (right) [122]. The profiles are obtained with an electron beam.

The obtained results are comparable: the average ratio of energy deposits of initial electrons in the central crystal (E1), 3x3 matrix (E3), and 5x5 (E5) matrix around it are approximately 78%, 94%, 97% for toy calorimeter and 79%, 95%, 98% for ECAL simulation [122], respectively. This demonstrates that the toy calorimeter can be used as a proxy for the real ECAL.

The asymmetry in the energy deposition around the central crystal is not present for the toy calorimeter (unlike in the ECAL simulation) as it does not include crystal tilt.

### 6.1.1 Energy resolution

As the Geant4 simulation does not include a readout chain, the true deposited energy in each crystal is smeared to mimic the performance of the real detector using the parameterization of the ECAL energy resolution discussed in Chapter 2 and presented by Eq. (2.9).

Eventually, the energy of each crystal is given by:

$$E_{\text{xtal}} = E_{\text{xtal}}^{\text{true}} \times \mathcal{N}(E_{\text{xtal}}^{\text{true}}, \sigma^2), \tag{6.1}$$

where $\mathcal{N}$ is the Gaussian distribution function, $\sigma$ is the standard deviation taken from Eq. (2.9) with the parameters corresponding to Run 3 operation, $E_{\text{xtal}}^{\text{true}}$ is true energy deposits in crystals.

A cut at 50 MeV is applied on the smeared energy to mitigate the noise.

## 6.2 Performance of the PFClustering algorithm

The ECAL reconstruction must be very efficient and provide excellent resolution both for energy and position measurements of the standalone particles. To achieve this, a

dedicated PFClustering algorithm is used in the CMS experiment. It is designed to maintain high efficiency even for low-energy photons and electrons and distinguish closely spaced particles. A detailed description of all the steps of the PFClustering algorithm is presented in Chapter 2.

In order to perform a fair comparison with the DeepCluster model, the PFClustering algorithm is independently implemented outside of the general CMS software framework and is applied to the simulated datasets described in Section 6.1. The results for the position and energy reconstruction are shown in Figs. 6.3 and 6.4, respectively. Each plot demonstrates the difference between the value reconstructed with the PFClustering algorithm (reco) and the value generated in the simulation (gen). The energy distribution is presented in bins of the energy of the generated photon. The resolution is determined from these distributions as $\sigma$ = [quantile(0.84) - quantile(0.16)]/2.

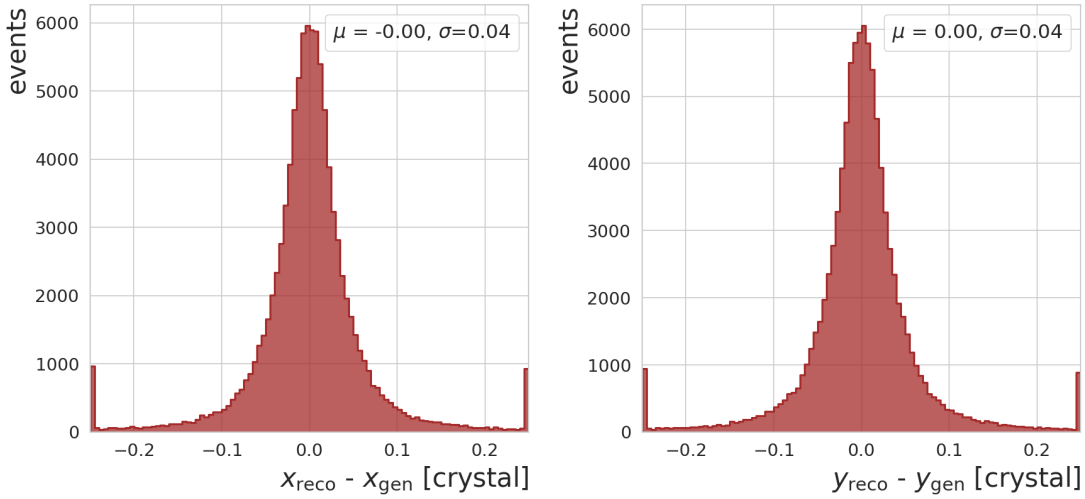



Figure 6.3: Difference between the reconstructed position ($x_{\text{reco}}$ on the left, $y_{\text{reco}}$ on the right) and the generated position of the particle ($x_{\text{gen}}$ on the left, $y_{\text{gen}}$ on the right). The results are obtained by applying the PFClustering (PF) algorithm on the single-photon test dataset. The resolutions and mean values evaluated from these distributions are reported on the plots.

### 6.2.1 Energy corrections

Energy losses can occur in a real calorimeter because of a variety of factors such as lateral and longitudinal shower leakage, intermodule gaps, and dead channels. In order to accurately determine the initial energy of the particle, correction coefficients must be applied to the raw energy predicted by PFClustering. Even though the described simulation can only account for the longitudinal shower leakage, the energy resolution is still significantly degraded due to this leakage and energy smearing as can be seen in Fig. 6.4.

In the particle flow framework, energy corrections are typically applied after the formation of final EM objects (as discussed in Chapter 2). It is done using a multivariate technique, called Boosted Decision Trees (BDT), discussed in Chapter 4.

To achieve a fair comparison of energy reconstruction performance between PFClustering and the DeepCluster model, a modified version of the particle-flow BDT is implemented (using scikit-learn [123]). The primary changes involve adjusting the input of the algorithm. This includes removing variables that are only defined for a SuperCluster and not for a PFCluster, such as the energy of the caloSeed, and adding

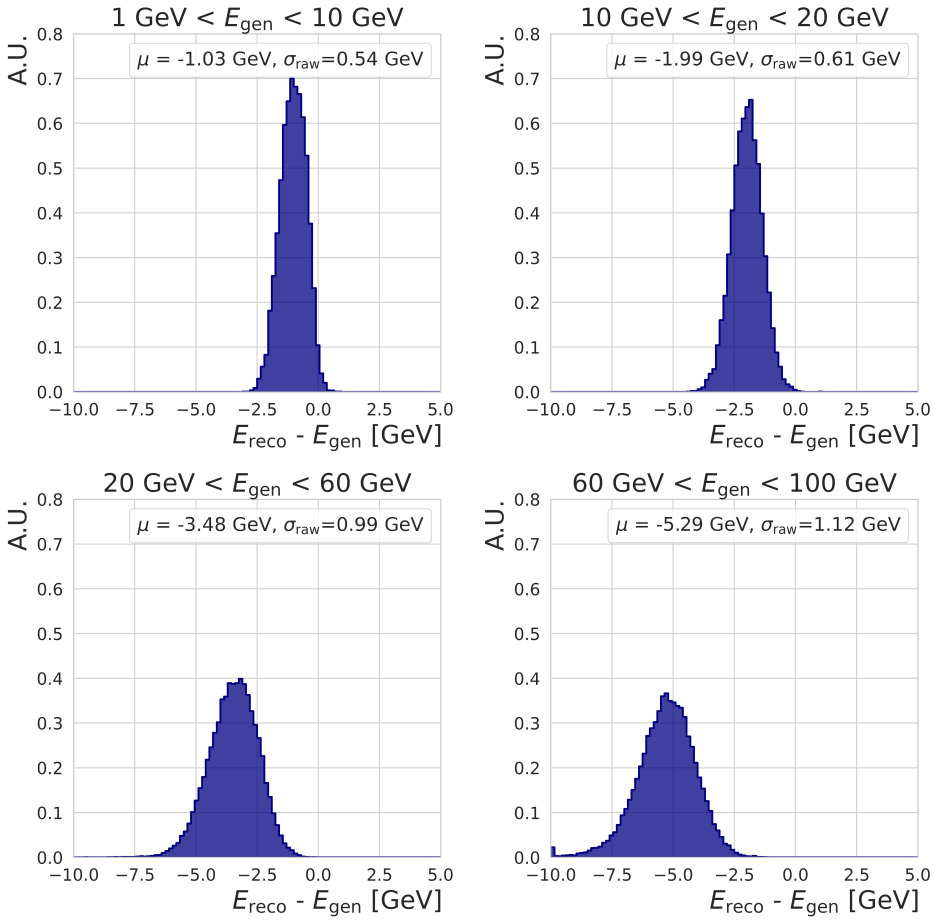



Figure 6.4: Difference between the reconstructed energy $E_{\text{reco}}$ and the generated energy of the particle $E_{\text{gen}}$. The results are obtained by applying the PFClustering algorithm on the single-photon test dataset. The distributions are presented in four bins of the generated photon energy $E_{\text{gen}}$: [1, 10] GeV, [10, 20] GeV, [20, 60] GeV, and [60, 100] GeV. The resolutions and mean values evaluated from these distributions are reported on the plots.

extra observables specific to the PFClustering step. A comprehensive list of all inputs with their brief descriptions is presented in Table 6.2, the list of input variables used in particle-flow BDT is given in Table 2.1. The output of the BDT is a correction value that needs to be applied to the raw PFClustering energy prediction.

| **Parameter** | **Description** |
|---|---|
| $E_{\text{reco}}$ | Energy predicted by the PFClustering algorithm |
| $x_{\text{reco}}$, $y_{\text{reco}}$ | Particle position predicted by the PFClustering algorithm |
| $E_{\text{max}}$, $E_{\text{2nd}}$ | Largest and second largest energy deposits in the crystals within the processed PFCluster. |
| $E_{\text{LR}}$ | The energy deposit difference between the left and right crystals in relation to the seed crystal. |
| $E_{\text{TB}}$ | The energy deposit difference between the top and bottom crystals in relation to the seed crystal. |
| $cov(X,X)$, $cov(X,Y)$, $cov(Y,Y)$ | Covariance values between the PFCluster spread in different directions. |
| $E_{\text{3x3}}$ | The sum of energies in a 3x3 matrix around the seed crystal. |

Table 6.2: Input variables to the PFClustering energy regression BDT.

The BDT is trained on a single-photon dataset with flat energy distribution between 1 and 250 GeV. The internal parameters of the model (see Chapter 4), such as *learning rate*, *number of estimators*, *minimum split*, *minimum leaf*, and *maximum depth*, are optimized to achieve the best performance using a random grid search. In this procedure, a BDT is trained with a randomly selected set of internal parameters, and the results are compared. The performance metric is a regression score ($R^2_{\text{score}}$):

$$R^2_{\text{score}} = 1 - \frac{u}{v}, \qquad u = \sum_{k=1}^{n}(E_{\text{gen}} - E_{\text{reco}})^2, \qquad v = \sum_{k=1}^{n}(E_{\text{gen}} - \bar{E}_{\text{gen}})^2, \tag{6.2}$$

where $n$ is the number of data samples, $E_{\text{gen}}$ is the true energy of the generated particle, $\bar{E}_{\text{gen}}$ is its mean estimation, and $E_{\text{reco}}$ is the reconstructed energy after the BDT correction. The model with a higher value of $R^2_{\text{score}}$ is superior, and the best possible $R^2_{\text{score}}$ is 1.

The resulting regression scores for 20 different sets of parameters are calculated. The full summary of optimization is presented in Table 6.3 and parameters corresponding to the highest $R^2_{\text{score}}$ (trial 10) are selected based on these results.

The performance in terms of energy for the final optimized model is shown in Fig. 6.5 along with the comparison to the raw PFClustering prediction. With the BDT correction, the energy resolution is drastically improved compared to the raw results: by 35%, 36%, 43%, and 26% for the energies of generated particles $E_{\text{gen}}$ = [1, 10] GeV, $E_{\text{gen}}$ = [10, 20] GeV, $E_{\text{gen}}$ = [20, 60] GeV, and $E_{\text{gen}}$ = [60, 100] GeV, respectively. Moreover,

| **trial** | **learning rate** | **number of estimators** | **minimum split** | **minimum leaf** | **maximum depth** | **score** |
|---|---|---|---|---|---|---|
| 1 | 0.20 | 100 | 20 | 5 | 20 | 0.7799 |
| 2 | 0.05 | 300 | 10 | 10 | 2 | 0.7676 |
| 3 | 0.20 | 300 | 10 | 5 | 10 | 0.7815 |
| 4 | 0.05 | 300 | 20 | 2 | 2 | 0.7756 |
| 5 | 0.20 | 300 | 2 | 10 | 5 | 0.7886 |
| 6 | 0.05 | 500 | 4 | 5 | 20 | 0.7797 |
| 7 | 0.20 | 100 | 10 | 2 | 10 | 0.7952 |
| 8 | 0.05 | 100 | 20 | 2 | 5 | 0.7874 |
| 9 | 0.20 | 500 | 20 | 2 | 5 | 0.7947 |
| 10 | 0.10 | 100 | 2 | 2 | 10 | 0.7953 |
| 11 | 0.20 | 300 | 10 | 2 | 10 | 0.7887 |
| 12 | 0.10 | 300 | 2 | 10 | 2 | 0.7754 |
| 13 | 0.10 | 300 | 4 | 10 | 10 | 0.7850 |
| 14 | 0.20 | 300 | 2 | 10 | 10 | 0.7782 |
| 15 | 0.20 | 500 | 20 | 2 | 2 | 0.7913 |
| 16 | 0.05 | 100 | 2 | 5 | 10 | 0.7913 |
| 17 | 0.10 | 300 | 4 | 5 | 5 | 0.7951 |
| 18 | 0.10 | 500 | 20 | 10 | 10 | 0.7828 |
| 19 | 0.20 | 300 | 4 | 10 | 10 | 0.7782 |
| 20 | 0.05 | 300 | 2 | 2 | 5 | 0.7952 |

Table 6.3: Results for the BDT hyperparameter optimization. The algorithm is applied on the single-photon validation dataset, and the regression score measures its performance. The parameters of the $10^{\text{th}}$ trial are chosen as optimal as they correspond to the highest value of the regression score.

the corrected energy distributions are centered around zero, unlike the ones obtained only from the PFClustering algorithm. The presented results are comparable with the estimated performance of the PFClustering reconstruction during Run 3, shown previously in Fig. 2.14.

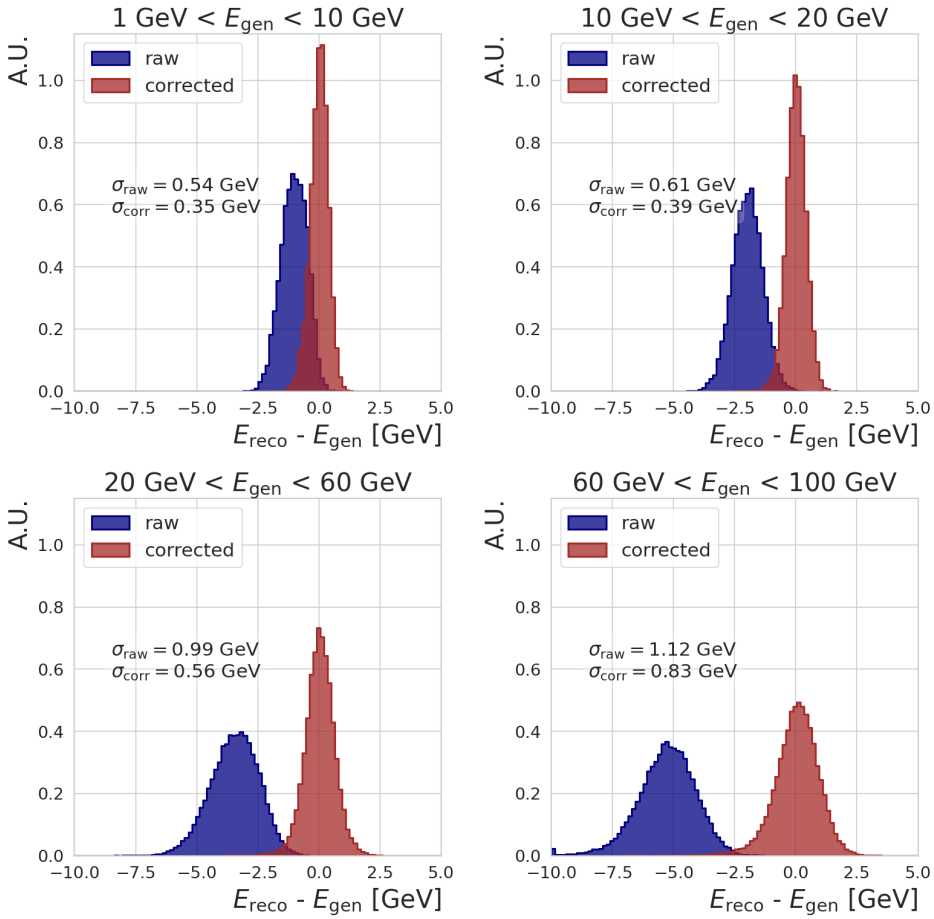



Figure 6.5: Difference between the reconstructed energy $E_{\text{reco}}$ and the generated energy of the particle $E_{\text{gen}}$. The results are obtained by applying the PFClustering algorithm (raw) and the PFClustering algorithm with additional energy regression (corrected) on the single-photon test dataset. The distributions are presented in four bins of the generated photon energy $E_{\text{gen}}$: [1, 10] GeV, [10, 20] GeV, [20, 60] GeV, and [60, 100] GeV. The resolutions evaluated from these distributions are reported on the plots.

### 6.2.2 Conclusion

The standalone version of the PFClustering algorithm is implemented for the toy calorimeter.

Despite its excellent performance, the PFClustering algorithm still has significant limitations:

1. As previously mentioned, it struggles to correctly differentiate between an isolated photon ($\gamma$) and a neutral pion ($\pi^0$). $\pi^0$ is an unstable particle (mean lifetime $\approx 8.3 \cdot 10^{-17}$ s [124]) that most likely decays into two $\gamma$. When originating from a high-energy $\pi^0$, the opening angle between the momenta of these two particles is small. In this case, their energy deposits overlap in the calorimeter, mimicking the

energy pattern of a single $\gamma$. A sketch of a neutral pion decay and its signature in the calorimeter is shown in Fig. 6.6.

2. The algorithm also produces a high background rate (misidentification of noise as a real particle) at low energies ($< 1$ GeV). Due to the increased noise and aging of the detectors in the coming LHC operations, the performance is expected to further deteriorate.

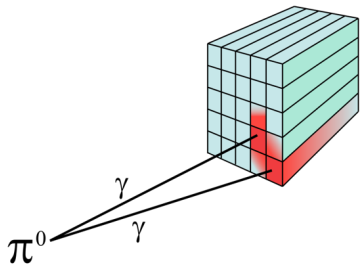



Figure 6.6: A sketch of $\pi^0$ decay to two photons and its signature in the calorimeter [125].

Currently, ML algorithms are widely used in particle physics experiments both for object reconstruction (e.g. [126]) and physics analysis (e.g. [77]) because of their often better performance compared to traditional algorithms. The goal of this work is to demonstrate that ML models and, more specifically, deep neural networks can be used for standalone particle reconstruction in ECAL. With these new techniques, the limitations of the PFClustering algorithm can be overcome and, furthermore, the position and energy resolutions can be potentially improved.

## 6.3 DeepCluster model: one-shot network

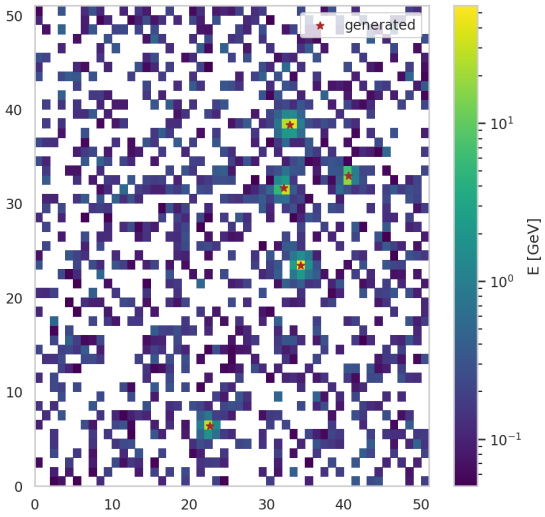
The first implementation of the DeepCluster model for the task of the standalone particle reconstruction is a one-shot CNN. This particular type of deep neural network is chosen based on the fact that the deposits in the crystals of the calorimeter can be represented as pixel intensities of an image, and CNNs are widely used to process the information from images.

The label "one-shot" in the name indicates that this CNN processes the full toy calorimeter at once. In order to do it, the network takes a matrix with a size of 51 x 51 (called *toy calorimeter window*) as an input, where each value represents an energy deposit in each particular crystal. From this information, the model aims to predict the positions and energies of all the generated particles in a given sample. An example of one input sample to the one-shot network is presented in Fig. 6.7, and the associated target values (truth information) are listed in Table 6.7.

### 6.3.1 Network architecture

The one-shot network uses a typical CNN architecture [127], where multiple convolutional layers are used followed by several dense layers, as discussed in Chapter 4. Specific parameters of this model are chosen to minimize the number of trainable weights while maintaining good performance. In the work presented in this document all the networks are implemented using TensorFlow [128] framework.

The model architecture consists of nine different convolutional layers with gradually growing filter numbers and kernel sizes. The convolutional layers extract patterns from the input image, and the resulting information is combined into one flattened vector, representing the summary features of this input. This vector is then passed through a chain of dense layers, separated into two branches: one leading to the prediction of the positions of the generated particles and another to the prediction of the energies of the




| x coordinate | y coordinate | Energy [GeV] |
|---|---|---|
| 38.34 | 33.0 | 81.6 |
| 32.9 | 40.6 | 30.9 |
| 31.6 | 32.2 | 44.3 |
| 23.5 | 34.4 | 71.4 |
| 6.4 | 22.6 | 38.0 |

Figure 6.7: On the left: an example of a toy calorimeter window, displaying a sample with five generated photons, serving as an input to a one-shot network. On the right: a table listing the generated position and energy values of the five photons in the sample, serving as the target output for a one-shot network.

generated particles. More details on the convolutional layers and their parameters can be found in Section 4.4.1 and for the dense layers in Section 4.3 of Chapter 4.

Both convolutional and dense layers use ReLU as an activation function except for the final layer, where sigmoid and tanh are chosen for position and energy, respectively. This is done in order to match the target output values, which are rescaled to [0, 1] for position and [-1, 1] for energy. A dropout of 1% is applied at each layer to mitigate overfitting. The full one-shot network architecture is shown in Fig. 6.8.

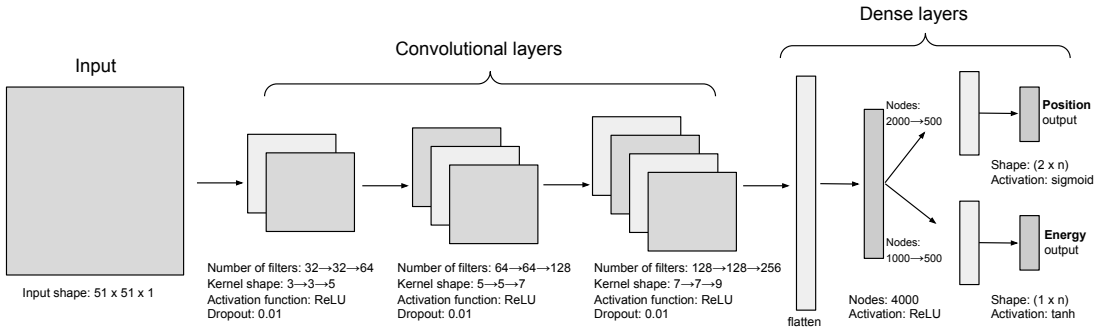



Figure 6.8: One-shot network architecture. The toy calorimeter window is passed as an input and is processed by multiple convolutional layers, resulting in one vector representing the summary features of the sample. This vector is further passed through a chain of dense layers, separated into two branches: one leading to the prediction of the positions of the generated particles and another to the prediction of the energies of the generated particles. $n$ represents the number of reconstructed particles.

The network is trained for 300 epochs with a batch size of 64, using an ADAM optimizer with a learning rate = 0.0001. Mean absolute error is chosen as the loss function both for position and energy predictions. The training parameters are discussed in detail in Section 4.3. The activation functions used in this work are shown in Fig. 4.6, and the description of the loss functions can be found in Section 4.1.2.

Various hyperparameter values are tested for this network, including the number of

kernels, their shapes, and the number of nodes in the dense layers. The results are shown for the ones providing the best performance.

### 6.3.2 Results

The performance of the DeepCluster model for position reconstruction is shown in Fig. 6.9 along with the comparison to the PFClustering algorithm. The plot demonstrates the difference between the reconstructed (reco) and the generated (gen) $x$-(on the right) and $y$-positions (on the left). The position resolution obtained with the network reconstruction is = 0.10 crystal, which is considerably higher than the one achieved with the PFClustering (0.04 crystal).

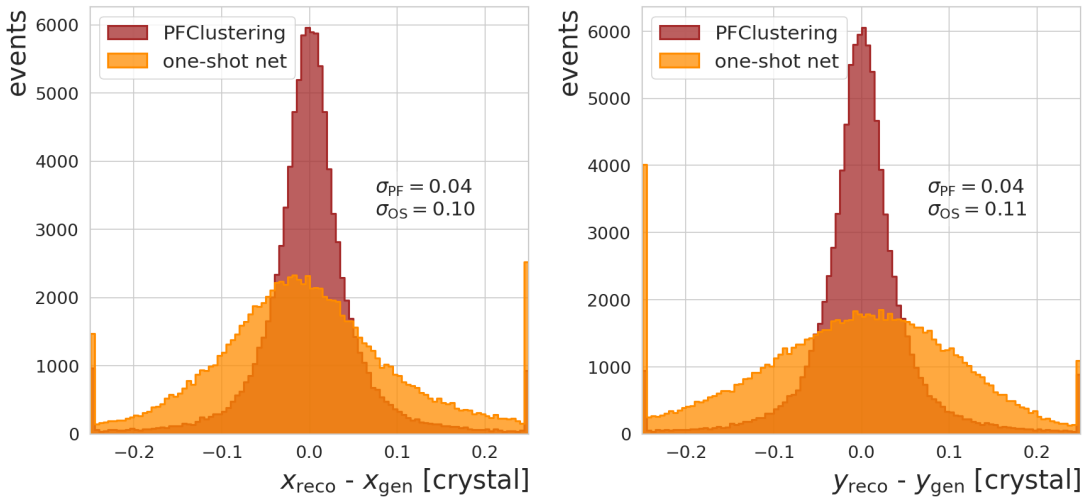



Figure 6.9: Difference between the reconstructed position ($x_{\text{reco}}$ on the left, $y_{\text{reco}}$ on the right) and the generated position of the particle ($x_{\text{gen}}$ on the left, $y_{\text{gen}}$ on the right). The results are obtained by applying the PFClustering (PF) algorithm and one-shot (OS) network on the single-photon test dataset. The resolutions evaluated from these distributions are reported on the plots.

### 6.3.3 Conclusion

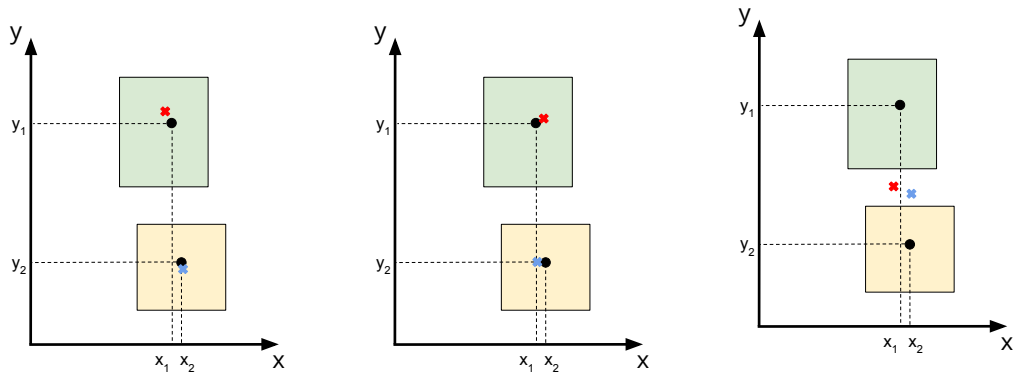
The performance of the first implementation of the DeepCluster model, presented in Fig. 6.3, shows promising results. The network is able to roughly identify the positions of the generated particles, indicating that it is a viable approach.

However, the obtained resolution is significantly worse compared to the PFClustering algorithm. After multiple attempts at improving the model, the two main issues preventing the network from performing better are:

1. **Ordering problem**. To enable a neural network to predict the positions and energies of multiple particles in a single sample, it is essential to arrange the target values in a specific order during the training process. If these values are supplied randomly, the network is unable to learn, as it needs a structured approach to predict the output and optimize the loss functions effectively. While in general, it can be done with simple sorting on $x$ positions, this type of sorting still creates confusion between associate $y$ positions when two particles are too close to each other in the $x$ coordinate.

   To better understand the issue, a simplified example can be considered. Figure 6.10 schematically shows two energy clusters green and yellow, produced by two gen-

erated photons with positions $(x_1, y_1)$ and $(x_2, y_2)$, respectively. The generated positions of the photons are very close to each other in terms of their $x$-coordinate: $|x_1 - x_2| < 0.05$ crystals. When the network encounters this sample for the first time, it predicts two positions: $(\bar{x}_1, \bar{y}_1)$ denoted as red and $(\bar{x}_2, \bar{y}_2)$ denoted as blue in Fig. 6.10 (a). From the majority of the available data samples, where the positions of different particles are well-separated in both directions, the model can easily learn that the target values are ordered based on $x$. Consequently, for the loss function, the network will associate the red prediction (lower $x$) to the first photon (represented with the green energy cluster) and the blue — to the second (represented with the yellow energy cluster). However, due to the extremely small $\Delta x$ between two photons, when the network processes this sample again, it may randomly make predictions where $\bar{x}_2 < \bar{x}_1$ as illustrated in Fig. 6.10 (b). In this case, the loss function receives the red prediction $(\bar{x}_1, \bar{y}_1)$ associated with the second photon (yellow energy cluster) and blue prediction $(\bar{x}_2, \bar{y}_2)$ associated with the first photon (green energy cluster). Finally, after trying to optimize for both of these cases, the network ends up predicting the coordinates that lie somewhere in between the positions of two generated photons, as shown in Fig. 6.10 (c).




a) The first prediction. b) The second prediction. c) The prediction after convergence.

Figure 6.10: Schematic illustration of the ordering problem.

2. **Sparsity problem**. Most of the pixels in the sample do not carry any valuable information, as the crystals are empty or contain only noise. However, as the full toy calorimeter window is passed as input to the network, the model still has to process all crystals. It leads to a very large number of trainable parameters and, as a result, the performance is degraded. Moreover, this type of network is not scalable to the real ECAL detector, which is much larger (360 x 170 crystals) compared to the toy calorimeter (51 x 51 crystals).

## 6.4 DeepCluster model: two-step network

The second development stage of the DeepCluster model aims to overcome the difficulties encountered with the one-shot CNN. In order to do it, a new approach is introduced: instead of using the full toy calorimeter window as input, it is pre-processed into smaller *seed windows* around the energetic crystals ($> 0.5$ GeV). Each of these seed windows either contains a cluster of energy deposits from a generated particle (further referred to as *energy cluster*) or just noise. In this approach, the standalone particle reconstruction task is divided into two consecutive steps:

1. Select seed windows containing energy clusters from all the generated particles and discard the ones with only noise.

2. For each selected window predict the kinematic properties of the corresponding generated particle.

These steps are performed by two separate dedicated models. The first one is called *seed-finder NN* and the second is *center-finder NN*.

This approach solves both ordering and sparsity problems encountered for one-shot architecture. As the seed windows are passed separately from each other, the network has to process only a small matrix as input at each iteration, which drastically reduces the number of trainable weights and removes the need to analyze irrelevant areas of the toy calorimeter. The windows can also be transmitted in random order, since they are independent of each other, which does not require any sorting rule.

In this section, a detailed description of seed-finder and center-finder networks is presented, along with the development steps and obtained performances.

### 6.4.1 Input and truth association

The input for the networks of the two-step approach is obtained from the toy calorimeter window for each data sample as follows:

1. All the crystals in the toy calorimeter window with energy deposits $> 0.5$ GeV are defined as potential *seed crystals*. The threshold value of 0.5 GeV is chosen to be approximately equal to $E_{\text{thr}}^{\text{seed}}$ tuned for the PFClustering algorithm for Run 3 operations.

   This definition of the seed crystal is slightly different from the one introduced for the PFClustering algorithm in Section 2.4.2, as it does not require the local maximum condition. Nevertheless, the naming is decided to be left identical as in this Chapter only seed crystals in relation to the DeepCluster network are mentioned and, thus, should not create confusion.

2. For each potential seed crystal a seed window is created. It represents a matrix of the size of 7x7 crystals, containing energy deposits of the seed crystal (the center of the matrix) and the crystals around it. An example of a seed window is shown in Fig. 6.11.

From one simulated toy calorimeter sample, multiple seed windows are created. Each seed window can contain single and overlapping energy clusters from generated particles or only noise.

In order to perform the training of the networks, the association between the seed windows and the generated particles and subsequent truth labeling has to be defined. In this work, for each seed window it is done as follows:

1. Check if the position of any generated particle in the considered sample lies within the boundaries of the seed crystal. In case such a particle is found, associate it with the seed window.

2. If the seed window is associated with a particle, it is labeled as *true seed window* and assigned three kinematic variables of the respective particle: generated position ($x_{\text{gen}}$, $y_{\text{gen}}$) and energy ($E_{\text{gen}}$). Otherwise, it is labeled as background, and no values are assigned to it.

3. The local position of the particle inside the seed window ($x_{\text{gen}}^{\text{local}}$, $y_{\text{gen}}^{\text{local}}$) is calculated by subtracting the position of the window center ($x_{\text{window}}$, $y_{\text{window}}$) from the global particle positions inside the toy calorimeter ($x_{\text{gen}}$, $y_{\text{gen}}$).

### 6.4.2 Seed-finder NN

The seed-finder NN is the first network in the two-step DeepCluster model. It is implemented as a CNN and its goal is to find the true seed windows and discard everything else (noise and not associated windows).

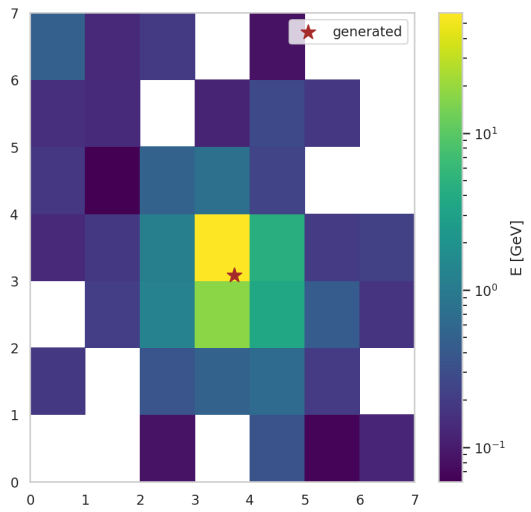



Figure 6.11: Example of a seed window around the possible seed crystal ($> 0.5$ GeV).

The network separately takes all the created windows as input and for each of them predicts a seed-finder score ($P_{\text{seedSF}}$), indicating the likelihood to be a true seed window. A threshold value ($P^{\text{thr}}_{\text{seedSF}}$) is defined and in the evaluation phase, only the windows with $P_{\text{seedSF}} > P^{\text{thr}}_{\text{seedSF}}$ are passed to the center-finder NN. The optimization of this threshold is further presented in Section 6.5.

The network architecture consists of two convolutional and two dense layers. LeakyReLU is chosen as the activation function and a dropout of 10% is applied after the first convolutions. For the output of the seed-finder NN, the sigmoid activation function is used. A detailed description of the model architecture is presented in Fig. 6.12.

The model is trained using Adam optimizer with a learning rate = 0.0001 and a batch size of 64. The early stopping is implemented for this network, where the training ends if there is no improvement in validation loss during 30 epochs. Using this technique, the network is trained for 94 epochs with epoch 64 yielding the best result. The loss function is binary cross entropy and its evolution with respect to epoch number is shown in Fig. 6.13 (left). The training parameters are discussed in detail in Section 4.3, the activation functions used in this work are shown in Fig. 4.6, and the description of the loss functions can be found in Section 4.1.2.

The distributions of the predicted seed-finder scores both for true seed windows and background evaluated on the single-photon test dataset are presented in Fig. 6.13 (right).

The main advantages that seed-finder NN brings compared to the PFClustering algorithm are:

- As the condition for the seed to be a local maximum is removed, the seed-finder NN provides a better possibility to reconstruct close-by photons.
- Seed-finder NN performs a refined seed window selection that helps to significantly eliminate the low-energy background.

### 6.4.3 Center-finder NN – Convolutional Neural Network

The center-finder NN is the second step of the DeepCluster model. The task of this network is to predict kinematics variables $x^{\text{local}}_{\text{gen}}$, $y^{\text{local}}_{\text{gen}}$, $E_{\text{gen}}$ of the generated particles from the associated seed window. The global coordinates of the particle can be further easily reconstructed from ($x^{\text{local}}_{\text{gen}}$, $y^{\text{local}}_{\text{gen}}$) and the position of the seed window ($x_{\text{window}}$, $y_{\text{window}}$).

For the first center-finder NN development, it is also implemented as a CNN. Similarly to the seed-finder NN, it takes seed windows as input. All the inputs are processed

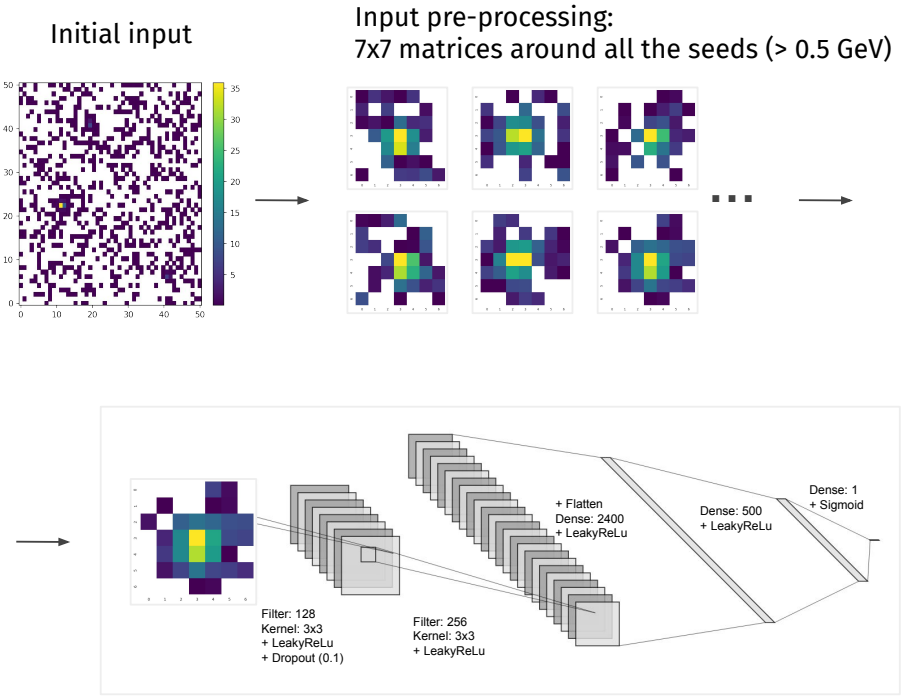



Figure 6.12: Seed-finder NN architecture. 7x7 seed windows are first selected around all possible seeds (>0.5 GeV) from the full toy calorimeter window. They are separately passed as input to the seed-finder NN. The input is processed by two convolutional layers until the vector of summary features is extracted. This vector is further passed to two dense layers, resulting in the network output: seed-finder score $P_{\text{seedSF}}$. It represents the likelihood of the input to be a true seed window. Detailed information on the number of nodes at each layer is presented in the figure.

independently from each other. In the training phase, the input consists only of the true seed windows, as only they are associated with a generated particle and, thus, have assigned kinematic variables. During the evaluation, however, the center-finder NN processes all the windows with the seed-finder scores passing $P_{\text{seed}}^{\text{thr}}$. It is further discussed in the next Section. The output of the center-finder NN is the prediction of three kinematic variables of the generated particle.

The architecture of the center-finder NN is fairly similar to the seed-finder NN: it consists of multiple convolutional layers, followed by dense layers that are divided into two parts: one resulting in coordinate prediction and another in energy prediction. ReLU activation function is applied everywhere except the output layer, where tanh and sigmoid are used for position and energy predictions, respectively. The dropout level is set to 10% everywhere except the last energy prediction layer, where it is set to 30%. The full network architecture with precise details on the number of nodes is presented in Fig. 6.14.

The network is trained for 1000 epochs with a batch size of 64 samples. For both of the outputs mean absolute error loss is used. The training is performed using Adam optimizer with a learning rate of 0.0001. The evolution of the loss function with respect to the epoch number is shown in Fig. 6.15. Epoch 974 is chosen as providing the best performance on the validation dataset. The training parameters are discussed in detail in Section 4.3. The activation functions used in this work are shown in Fig. 4.6, and the description of the loss functions can be found in Section 4.1.2.

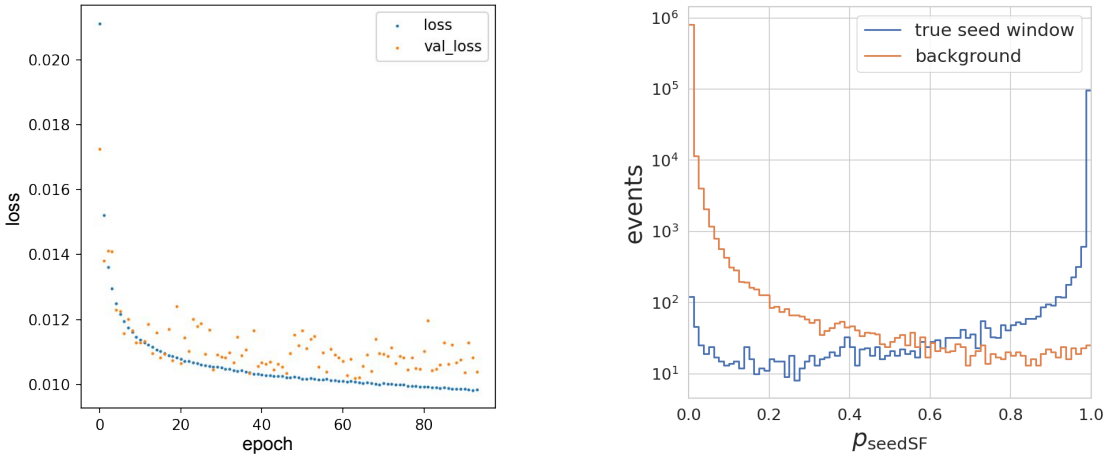



Figure 6.13: On the left: loss function evolution with respect to epoch number for the seed-finder NN training. On the right: distributions of seed-finder scores predicted by the network for true seed windows and background.

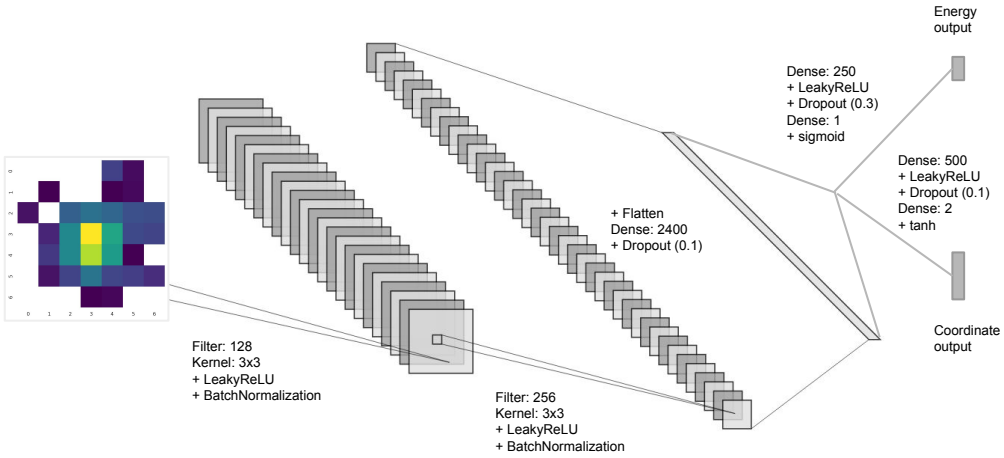



Figure 6.14: Center-finder NN architecture. The seed windows are passed independently as input to the network. The input is processed by two convolutional layers until the vector of summary features is extracted. This vector is passed through one dense layer and further sent separately to two different branches (coordinate and energy prediction). In each branch, it passes through two additional dense layers. Detailed information on the number of nodes at each layer is presented in the figure.

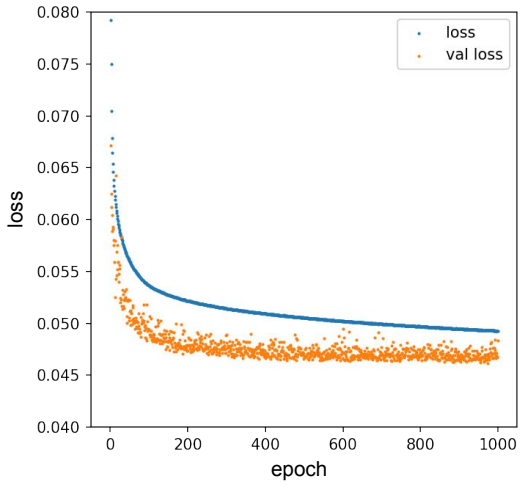



Figure 6.15: Loss function evolution with respect to epoch number for the center-finder NN training.

### 6.4.4 Evaluation and results

After both of the networks of the DeepCluster are trained, the resulting combined model is applied on the single- and two-photon test datasets to evaluate the performance.

The full reconstruction process is done according to several steps:

1. The initial samples of the test datasets are pre-processed in order to create seed windows according to the procedure from Section 6.4.1.
2. The seed windows are passed to the seed-finder NN and for each of them a seed-finder score is extracted.
3. The seed windows with $P_{\text{seedSF}} > P^{\text{thr}}_{\text{seedSF}}$ are passed to the center-finder NN.
4. The center-finder NN predicts the local coordinates $(x^{\text{local}}_{\text{reco}}, y^{\text{local}}_{\text{reco}})$ and the energy $E_{\text{reco}}$ of the potential particle from the selected seed windows.
5. The global positions in the toy calorimeter $(x_{\text{reco}}, y_{\text{reco}})$, are evaluated by summing predicted local positions $(x^{\text{local}}_{\text{reco}}, y^{\text{local}}_{\text{reco}})$ and positions of the corresponding seed windows $(x_{\text{window}}, y_{\text{window}})$.

As a result of these steps, for each selected seed window, a *reconstructed object (reco-object)* is created, which is defined by predicted kinematic values: $x_{\text{reco}}$, $y_{\text{reco}}$, and $E_{\text{reco}}$. It also possesses a seed-finder score $P_{\text{seedSF}}$. Every reco-object represents one potential particle.

To be able to evaluate the performance, a matching procedure that links the reco-objects with the true generated particles is also developed. The links inside each sample are created as follows:

- Equation (6.3) is used to calculate a "matching" variable $r_{\text{match}}$. It represents a normalized distance between the predicted values of the reco-objects and generated values of the particle in $(x, y, E)$ space.

$$r_{\text{match}} = \sqrt{\left(\frac{R_{\text{reco}} - R_{\text{gen}}}{\overline{R}}\right)^2 + \left(\frac{E_{\text{reco}} - E_{\text{gen}}}{\overline{E}}\right)^2}, \tag{6.3}$$

where $R_{\text{reco}} = \sqrt{(x_{\text{reco}})^2 + (y_{\text{reco}})^2}$ is the position of the reconstructed object, $E_{\text{reco}}$ is the energy of the reconstructed object, $R_{\text{gen}} = \sqrt{(x_{\text{gen}})^2 + (y_{\text{gen}})^2}$ is the generated position of the particle, $E_{\text{gen}}$ is the generated energy of the particle, $\overline{R}$ is the mean value of ($R_{\text{reco}}$ - $R_{\text{gen}}$) distribution, $\overline{E}$ is the mean value of ($E_{\text{reco}}$ - $E_{\text{gen}}$) distribution. These values are obtained from a preliminary truth association based only on the position.

For each reco-object in the sample:

- A matching variable is estimated between the considered reco-object and all the generated particles.
- A link between a reco-object and a particle with the minimum matching variable is created.
- Additional criteria on the Euclidian distance between the reco-object and generated particle ($\Delta R = \sqrt{(x_{\text{reco}} - x_{\text{gen}})^2 + (y_{\text{reco}} - y_{\text{gen}})^2}$) has to be satisfied. If $\Delta R > 1.5$ crystal, the link is removed. And the reco-object is further considered as background.

Each of the reco-objects is linked to only one generated particle, while the particle can be linked to multiple reco-objects, which is further referred to as *splitting*.

The same matching procedure is applied for the objects reconstructed by the DeepCluster model and PFClustering algorithm. The presented results for position and energy predictions include only pairs of linked reco-objects/generated particles. To further investigate the performance, additional variables are defined that represent the ability of the algorithm to correctly predict the properties of each sample: *signal efficiency*, *background yield*, *splitting yield*. Their description is given in Table 6.4.

| **Name** | **Description** |
|---|---|
| Signal efficiency | The number of linked reco-objects divided by the number of generated particles. |
| Splitting yield | The number of events where one particle was linked to two different reco-objects. |
| Background yield | The number of reco-objects that are not linked. |

Table 6.4: Description of the variables used for evaluation of the algorithms.

### Results

The results of the DeepCluster model and PFClutsering algorithms are evaluated on the single- and two-photon test datasets and compared with each other. For the seed-finer NN, a loose threshold of $P^{\text{thr}}_{\text{seedSF}} = 0.3$ is chosen for this evaluation as it enables high signal efficiency. A more rigorous threshold selection is done for the final DeepCluster model and it is further discussed in Section 6.5.

Performance for position reconstruction is presented in Fig. 6.16 for the single-photon dataset on the left and for the two-photon dataset on the right. Each plot demonstrates the difference between the reconstructed ($x_{\text{reco}}$) and generated ($x_{\text{gen}}$) $x$-coordinates. The results for the $y$-coordinate are omitted as they are similar to the $x$-coordinate due to the spatial symmetry of the toy calorimeter. With the new two-step approach, the DeepCluster network significantly outperforms the PFClustering algorithm. The

coordinate resolution (evaluated as $\sigma$ = [quantile(0.84) - quantile(0.16)]/2) for the two-step network (TS) is 0.02 crystal compared to 0.04 crystal for the PFClustering (PF) algorithm for the single-photon dataset and 0.03 crystal versus 0.08 crystal for the two-photon dataset.

The position reconstruction for the two-photon dataset is a more difficult task than for the single-photon as the overlapping energy clusters can lead to a non-ideal evaluation of the energy proportions for close-by generated particles and, as a result, can bias the prediction. This explains the discrepancy between the reported results for the two datasets.

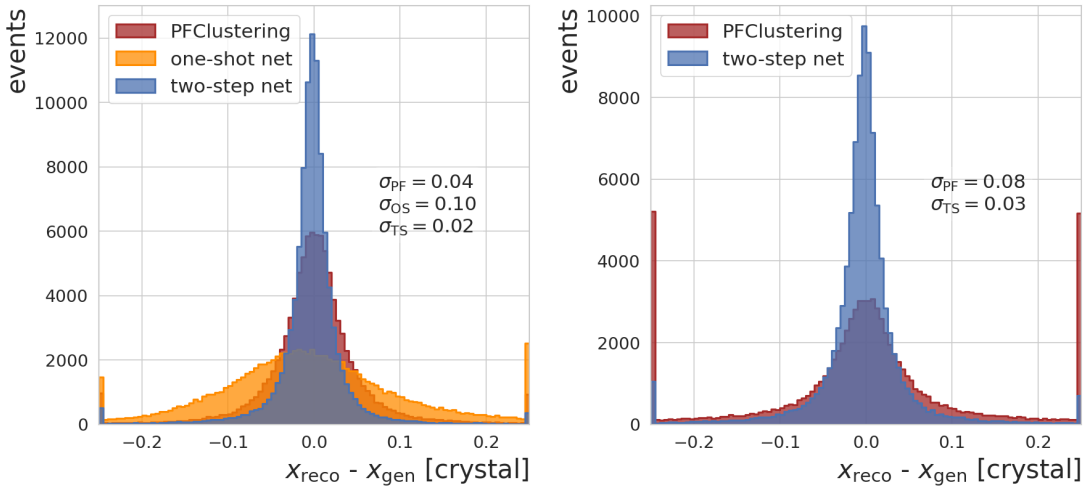



Figure 6.16: Difference between the reconstructed position $x_{\text{reco}}$ and the generated position of the particle $x_{\text{gen}}$. The results are obtained by applying the PFClustering (PF) algorithm, one-shot (OS) network, and two-step (TS) network on the single-photon (left) and two-photons (right) test datasets. The resolutions evaluated from these distributions are reported on the plots.

Performance for energy reconstruction is presented in Fig. 6.17 for the single-photon dataset and Fig. 6.18 for the two-photon dataset. Similarly to the coordinate, each plot demonstrates the difference between the reconstructed ($E_{\text{reco}}$) and generated ($E_{\text{gen}}$) energies. The results are presented in bins of the energy of the generated particle ($E_{\text{gen}}$). For both of the datasets the performance in the first two energy bins $E_{\text{gen}}$ = [1, 10] GeV and $E_{\text{gen}}$ = [10, 20] GeV is similar for the DeepCluster model and the PFClustering algorithm. For $E_{\text{gen}}$ = [20, 60] GeV and $E_{\text{gen}}$ = [60, 100] GeV the two-step network achieves slightly better results in the case of single-photons: $\sigma_{\text{TS}} = 0.52$ GeV versus $\sigma_{\text{PF}} = 0.56$ GeV and $\sigma_{\text{TS}} = 0.80$ GeV versus $\sigma_{\text{PF}} = 0.83$ GeV, respectively. In the case of two photons, it remarkably outperforms the PFClustering: $\sigma_{\text{TS}} = 0.82$ GeV versus $\sigma_{\text{PF}} = 2.59$ GeV and $\sigma_{\text{TS}} = 1.13$ GeV versus $\sigma_{\text{PF}} = 17.70$ GeV, respectively for the same energy ranges.

The same reason as discussed for the coordinate prediction explains the difference between the results for the two datasets. Moreover, the poor energy resolution obtained with the PFClustering algorithm for the two-photon dataset comes from the fact that the PFClustering energy regression is trained only on the single-photon dataset (as done in the CMS particle flow).

Finally, the signal efficiency, splitting yield, and background yield results for the DeepCluster model and the PFClustering algorithm are shown in Fig. 6.19 for the single-photon dataset on the left and for the two-photon dataset on the right. The results are presented versus the energy of the generated particle $E_{\text{gen}}$ for the signal efficiency

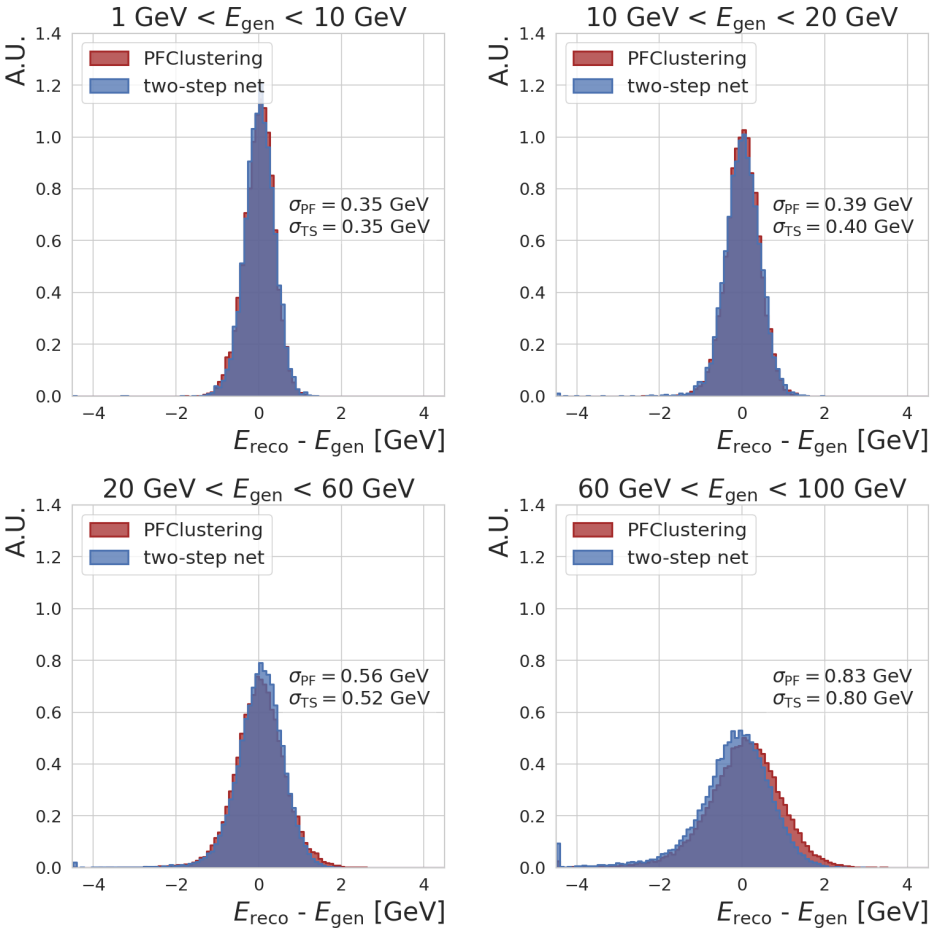



Figure 6.17: Difference between the reconstructed energy $E_{\text{reco}}$ and the generated energy of the particle $E_{\text{gen}}$. The results are obtained by applying the PFClustering (PF) algorithm and two-step (TS) network on the single-photon test dataset. The distributions are presented in four bins of the generated photon energy $E_{\text{gen}}$: [1, 10] GeV, [10, 20] GeV, [20, 60] GeV, and [60, 100] GeV. The resolutions evaluated from these distributions are reported on the plots.

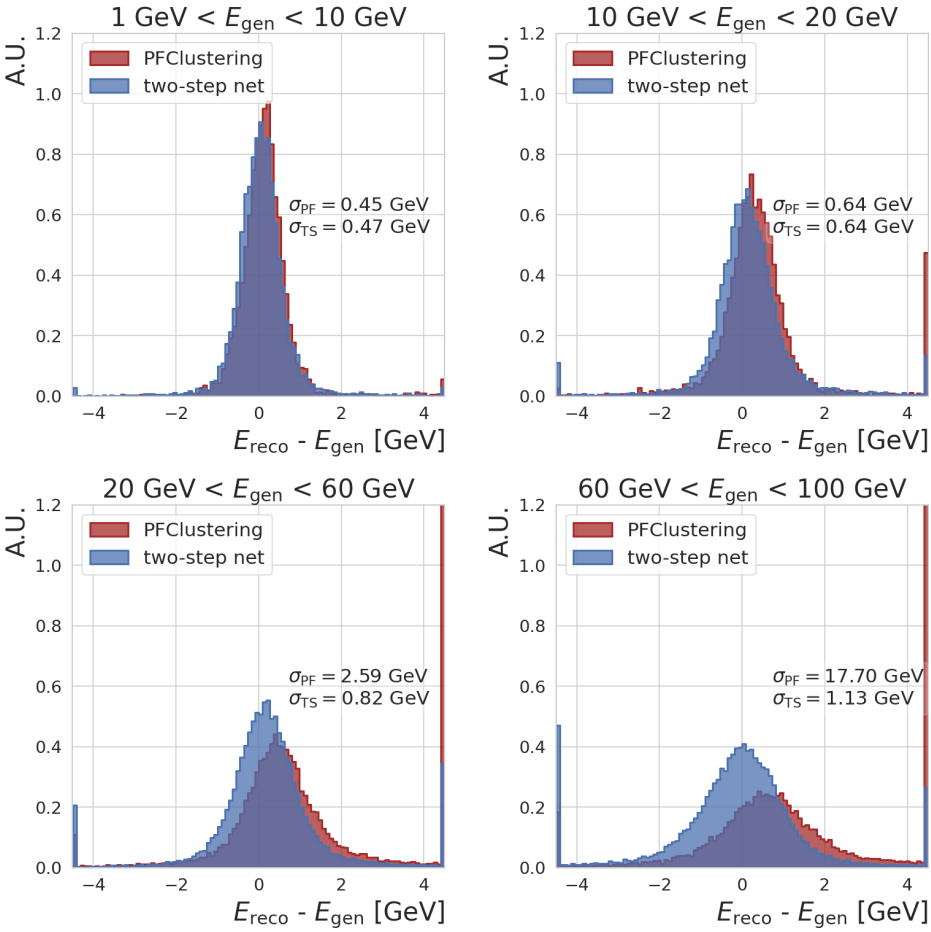



Figure 6.18: Difference between the reconstructed energy $E_{\text{reco}}$ and the generated energy of the particle $E_{\text{gen}}$. The results are obtained by applying the PFClustering (PF) algorithm and two-step (TS) network on the two-photon test dataset. The distributions are presented in four bins of the generated photon energy $E_{\text{gen}}$: [1, 10] GeV, [10, 20] GeV, [20, 60] GeV, and [60, 100] GeV. The resolutions evaluated from these distributions are reported on the plots.

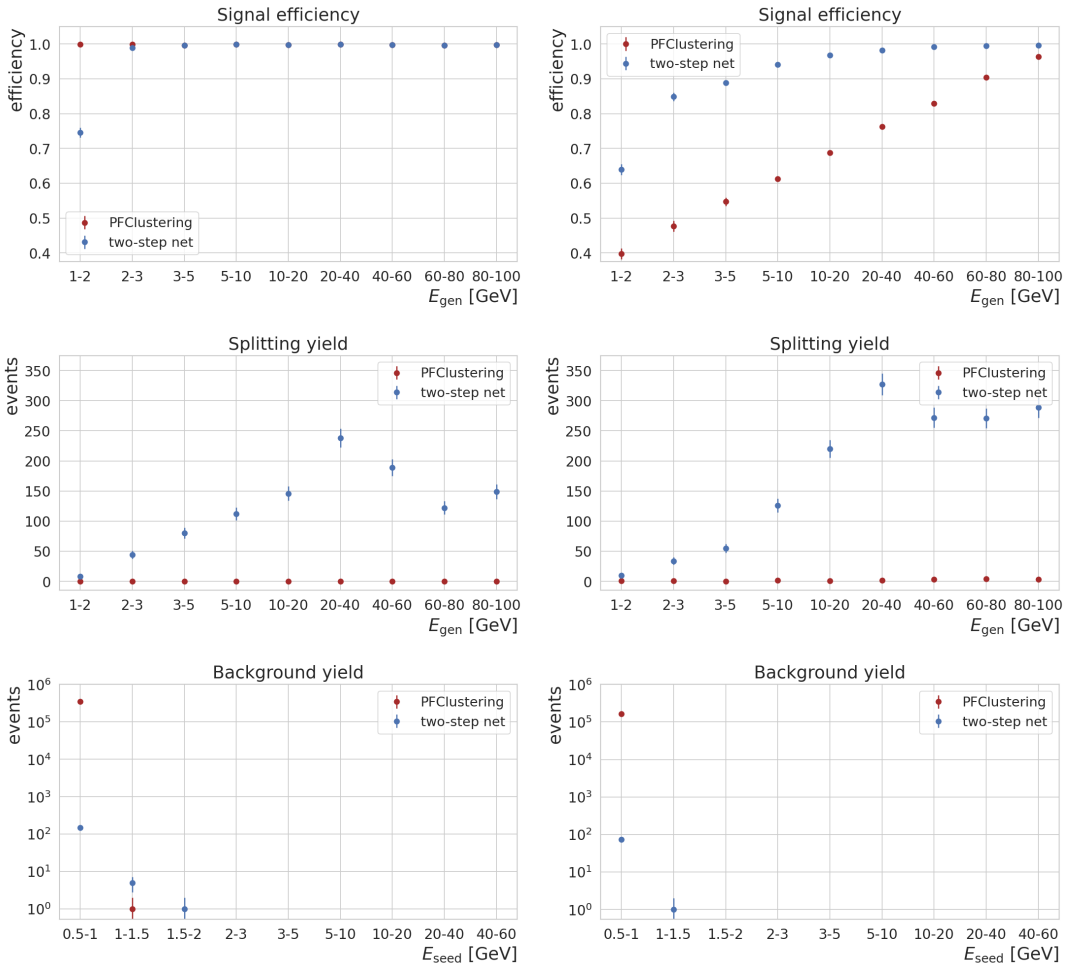



Figure 6.19: Signal efficiency (top), splitting yield (middle), and background yield (bottom) values. The results are obtained by applying the PFClustering (PF) algorithm and two-step (TS) network on the single-photon (left) and two-photon (right) test datasets.

and splitting yield, and versus the energy of the seed crystal $E_{\text{seed}}$ for the background yield.

The signal efficiency for the PFClustering algorithm is 100% for the single-photon case in the considered energy range by construction (at least one PFCluster is always reconstructed around the crystals with energy deposit $> E^{thr}_{seed}$). The DeepCluster network achieves identical performance starting from $E_{\text{gen}} = 3$ GeV. In the case of two close-by photons, the signal efficiency is considerably higher for the DeepCluster model compared to the PFClustering: starting from 64% at $E_{\text{gen}} = [1, 2]$ GeV and gradually improving to 100% for $E_{\text{gen}} = [80, 100]$ GeV while the corresponding values for the PFClustring are 40% and 96%.

For the low-energy background yield, the two-step network only incorrectly identifies $N^{\text{bkg}}_{\text{TS}} = 150$ events for single-photon dataset and $N^{\text{bkg}}_{\text{TS}} = 75$ events for two-photon dataset while the PFClustering reconstruction results in $N^{\text{bkg}}_{\text{PF}} \approx 350$k events and $N^{\text{bkg}}_{\text{PF}} \approx 165$k events, respectively. The results are reported on 100k samples for the single-photon dataset and 50k samples for the two-photon dataset.

Considering the splitting yield, the performance of the DeepCluster model is inferior to PFClustering. By construction, the PFClustering algorithm merges two very close energy deposits and, consequently, the splitting can not appear. However, this feature also prevents the algorithm to separate true energy deposits, which is crucial for $\pi^0$ reconstruction.

The aim of the subsequent DeepCluster model development is to mitigate the splitting yield while maintaining the excellent results achieved for signal efficiency. The splitting is also related to a problem called *"double-counting"*, where energy is overestimated due to the incorrect reconstruction. It is described in the following section.

**Double-counting problem**

After the full reconstruction chain of the DeepCluster model and the subsequent linking process, one generated particle can be incorrectly reconstructed as two reco-objects (splitting) as discussed in Section 6.4.4.

This issue arises as a result of the construction of the seed windows. As the local maximum condition is omitted for the seed crystal, two (or more) neighboring crystals with high energies ($> 0.5$ GeV) are both selected as seeds and give rise to two separate seed windows. In the majority of cases, where the energy deposit difference between these two neighbors is large, the seed-finder NN is able to internally identify the local maximum itself and predict a high seed-finder score $P_{\text{seedSF}}$ for the corresponding seed window (in which the local maximum is the central crystal) and a low score for its neighbor.

However, as can be seen from the splitting yield in Fig 6.19, there still occur several events, where seed-finder NN fails to correctly select between the neighboring seed windows. An especially drastic problem appears in the subset of these events, where the position of the particle is generated close to the edge of a crystal. In this case, a particle deposits large and almost equal energies in the neighboring crystals, and both of them give rise to a separate seed window. An example of created seed windows is shown in Fig. 6.20, obtained from a sample with a single generated particle with $E_{\text{gen}} = 97.7$ GeV. The seed-finder NN predicts high seed scores for both of them $P^1_{\text{seedNN}} = 0.94$ and $P^2_{\text{seedNN}} = 0.93$.

Following the reconstruction chain, these windows are separately passed to the center-finder NN. The network predicts two reco-objects with similar coordinates and energies: $(E^1_{\text{reco}}, x^1_{\text{reco}}, y^1_{\text{reco}}) = (84.0 \text{ GeV}, 19.22, 26.04)$ and $(E^2_{\text{reco}}, x^2_{\text{reco}}, y^2_{\text{reco}}) = (87.0 \text{ GeV}, 19.28, 25.97)$.

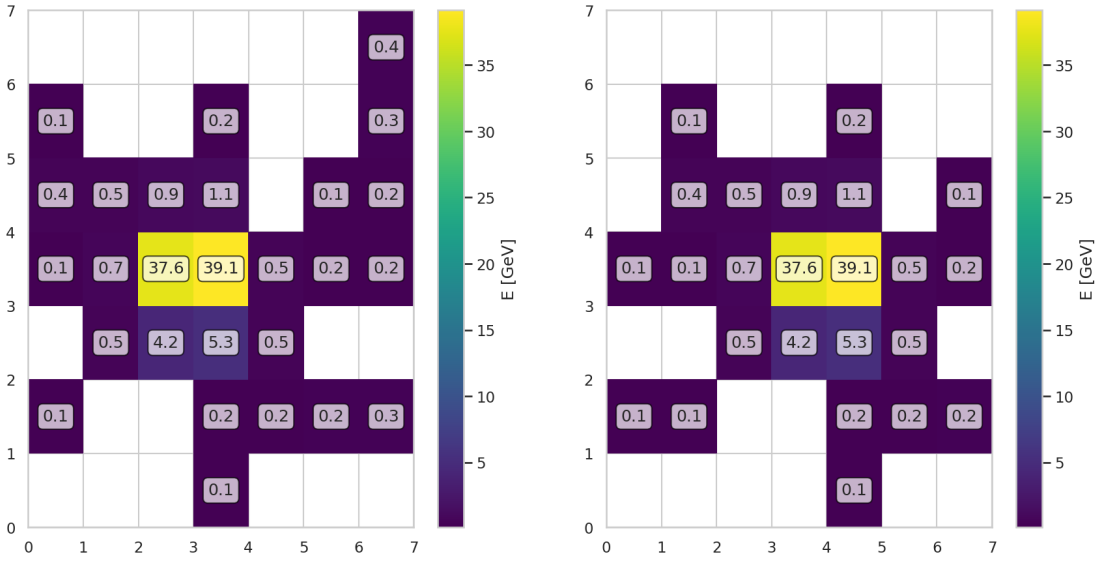



Figure 6.20: Example of the two seed windows created for a particle with a generated position close to the crystal border.

Consequently, the total energy reconstructed by the DeepCluster model for one sample ($E^{\text{tot}}_{\text{reco}}$), which is evaluated as the sum of the reconstructed energies $E_{\text{reco}}$ of all the reco-objects in the considered sample, can be significantly overestimated. This can be seen from Fig. 6.21 (left), where the distribution of the ratio between the total reconstructed energy $E^{\text{tot}}_{\text{reco}}$ and total generated energy $E^{\text{tot}}_{\text{gen}}$ (sum of the energies of all the simulated particles) for single-photon dataset is presented. The plot clearly shows an additional peak around two that comes from "double-counting".

### 6.4.5 Center-finder NN – Graph Neural Network

The DeepCluster model with a two-step approach shows excellent results and outperforms the PFClustering algorithm in the coordinate and energy resolution, signal efficiency, and background rejection. However, unlike PFClustering, the model can also create two reconstructed objects for one generated particle. The associated problem called "double-counting" leads to a considerable overestimation of the total reconstructed energy, the details are described in Section 6.4.4. The next development of the DeepCluster network addresses this issue by modifying the center-finder NN.

The splitting partially arises due to the lack of information about the input seed windows from the same sample inside the model, as each of them is processed independently. For example, in the sample described in Section 6.4.4, when one of the seed windows (illustrated in Fig. 6.20) is passed through the center-finder NN, the network is not "aware" that another, neighboring seed window exists and is also going to be processed. As a result, the center-finder NN predicts almost identical values from both of them, not knowing that the same particle is effectively being reconstructed twice.

The solution to this problem is to add a way of "communicating" between different input windows inside the network. This can be done by using GNNs as they provide a message-passing feature enabling information sharing between the inputs. The details about this type of neural network can be found in Chapter 4.

In this section, the center-finder NN, implemented as a graph neural network, is discussed, including the architecture of the model and the obtained results.

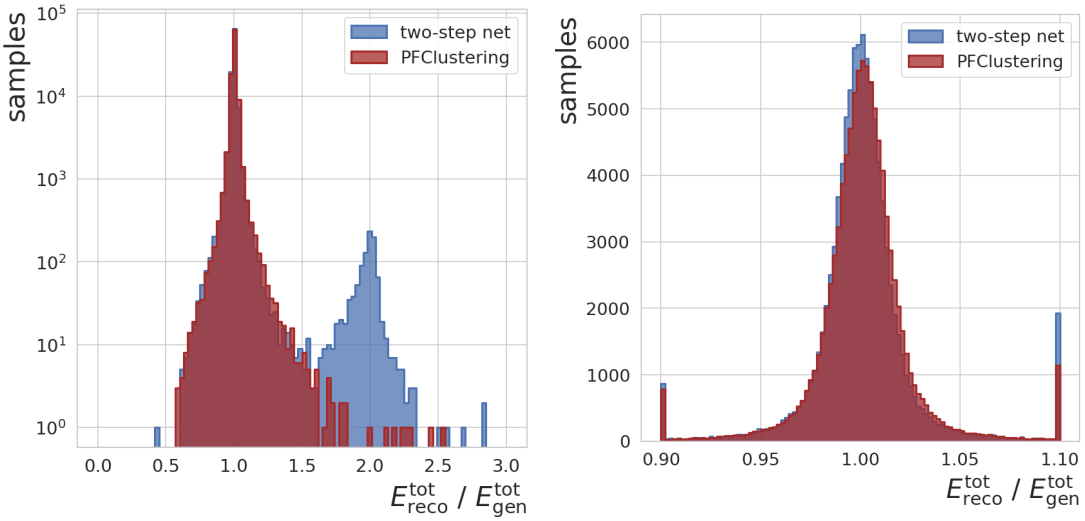



Figure 6.21: The distribution of the ratio between the total reconstructed energy $E^{\text{tot}}_{\text{reco}}$ and the total generated energy $E^{\text{tot}}_{\text{gen}}$. The results are obtained by applying the PFClustering algorithm and two-step network on the single-photon test dataset. They are shown in log (left) and linear (right) scales. Due to the double-counting problem, the second peak around two arises for the DeepCluster model.

### Network architecture and concept of message-passing

In order to enable the information sharing between the inputs, in the new implementation of the center-finder NN, the neighboring seed windows are no longer passed independently to the network. Instead, the input is created as follows:

1. The seed crystals and associated seed windows are constructed in the same way as discussed in Section 6.4.1 and a list containing pairs of seed crystal - seed window is created.

2. This list is ordered based on the energy of the seed crystal $E_{\text{seed}}$.

3. The list is processed until it becomes empty, starting from the entry with the highest $E_{\text{seed}}$:

    - The new input is initialized by the considered seed window. The current shape of the input is (1, 7, 7).
    - The distance ($\Delta R = \sqrt{\Delta x^2 + \Delta y^2}$) between the considered seed crystal and the remaining seed crystals in the list is evaluated.
    - The seed windows with $\Delta R < 3$ crystal are added to the input as well. This step changes the input shape to ($i$, 7, 7), where $i$ is the number of added seed windows.
    - All of the seed windows that make part of the input are removed from the list.

Each sample of the dataset is processed in the described way. The maximum value of $i$ is chosen to be four as two and four reco-objects can be potentially created for the samples with single and two generated photons, respectively. The maximum value of variable $i$ can be easily adjusted to higher values as well. If the seed window has $<$ 3 neighbors, the input is completed with 7x7 matrices of 0 values in order to have a constant input shape.

The four potential seed windows of the input represent the nodes of the graph. In the center-finder NN, each window $j$ is first separately processed by a chain of convolutional layers (same as for CNN implementation) in order to extract the vector of summary features $v_j$. The message-passing is implemented as the concatenation (or appending) of these vectors. It results in 4 updated vectors $\bar{v}_j$: , each of them containing information about their neighbors:

$$\bar{v}_1 = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{pmatrix}, \quad \bar{v}_2 = \begin{pmatrix} v_2 \\ v_1 \\ v_3 \\ v_4 \end{pmatrix} \quad \bar{v}_3 = \begin{pmatrix} v_3 \\ v_1 \\ v_2 \\ v_4 \end{pmatrix} \quad \bar{v}_4 = \begin{pmatrix} v_4 \\ v_1 \\ v_2 \\ v_3 \end{pmatrix} \tag{6.4}$$

The combined vectors $\bar{v}_j$ are then passed independently to a set of dense layers until the final output is extracted.

The center-finder NN in this implementation predicts 4 values for each seed window: kinematic variables ($x^j_{\text{reco}}$, $y^j_{\text{reco}}$, $E^j_{\text{reco}}$) and a new seed score $P_{\text{seedCF}}$, indicating the likelihood to be a true seed window.

In this version of the DeepCluster model, the seed-finder NN remains unchanged. It serves as an initial filter, separating signal from background, while the updated center-finder has to correct the predictions of reco-objects that potentially occur as a result of splitting. The reconstruction and evaluation procedure follows the same steps as discussed in Section 6.4.4. An extra condition is added at the end on the final reco-object: only objects with $P_{\text{seedCF}} > P^{\text{thr}}_{\text{seedCF}}$ are selected. The threshold optimization procedure is presented further in Section 6.5. The full DeepCluster model architecture is presented in Fig 6.22.

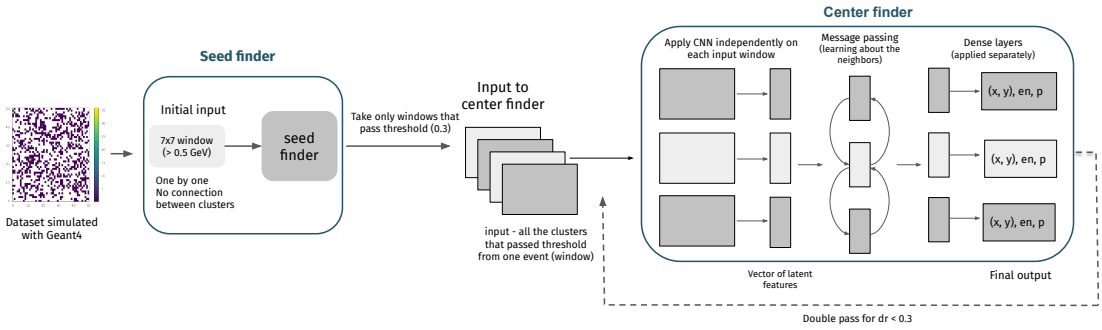



Figure 6.22: Flow chart of the DeepCluster model. 7x7 seed windows are first selected around all possible seeds (>0.5 GeV) from the full toy calorimeter window. They are separately passed as input to the seed-finder NN. The network predicts a seed-finder score $P_{\text{seedSF}}$ for each input. The selected seed windows with $P_{\text{seedSF}} > P^{\text{thr}}_{\text{seedSF}}$ are combined into groups of 4 with their neighbors and passed to center-finder NN. The network processes predicts coordinates $x_{\text{reco}}$, $y_{\text{reco}}$, energy $E_{\text{reco}}$ and a new seed score $P_{\text{seedCF}}$ for each window.

### Results

In the GNN implementation of the center-finder NN, the model receives information about all the neighboring seed windows simultaneously, instead of processing them independently. With this adjustment, the network is able to make a more informed decision of which seed windows shall receive a high seed score $P_{\text{seedCF}}$ and additionally better attribute energy fractions for different reco-objects. As a result, even if the particle

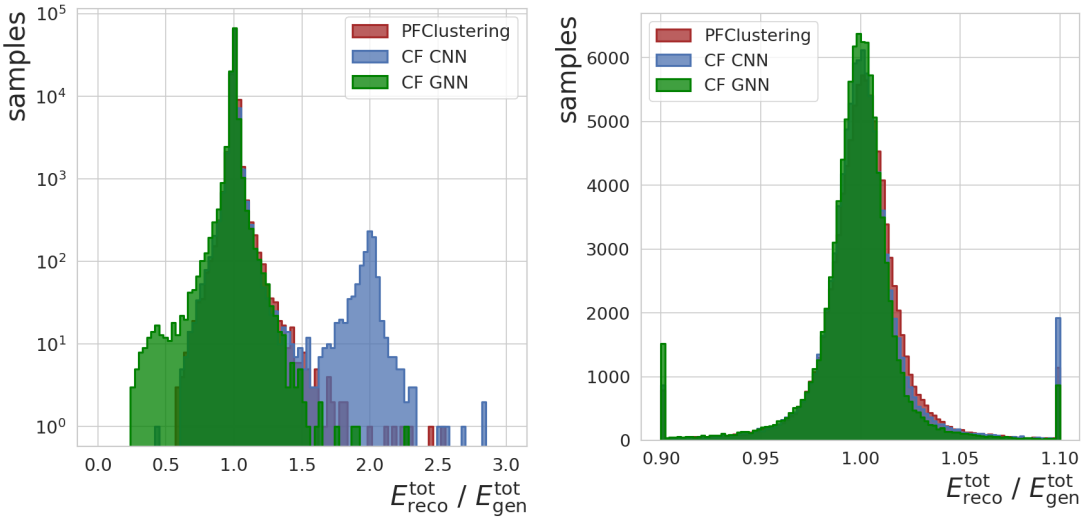



Figure 6.23: The distribution of the ratio between the total reconstructed energy $E^{\text{tot}}_{\text{reco}}$ and the total generated energy $E^{\text{tot}}_{\text{gen}}$. The results are obtained by applying the PFClustering algorithm, DeepCluster model with center-finder CNN (CF CNN) and center-finder GNN (CF GNN) on the single-photon test dataset. They are shown in log (left) and linear (right) scales. Due to the double-counting problem, the second peak around 2 arises for the CF CNN, while it is completely eliminated with CF GNN.

is reconstructed as two reco-objects, their energy predictions sum up to the generated energy, which mitigates $E^{\text{tot}}_{\text{reco}}$ overestimation.

This improvement is shown in Fig. 6.23, where the distribution of the ratio between the total reconstructed energy $E^{\text{tot}}_{\text{reco}}$ and total generated energy $E^{\text{tot}}_{\text{gen}}$ (sum of the energies of all the simulated particles) for single-photon dataset is presented. With the center-finder GNN (CF GNN) the overestimation peak around 2, which indicates double-counting, is removed.

The mitigation of the underestimation of energy with CF GNN that can be seen in these plots is further discussed in Section 6.5 – double pass.

## 6.5 Network optimization

The final version of the DeepCluster model (two-step network with center-finder GNN) is further optimized in order to achieve the best possible performance.

In this section, the selection of the loss function coefficients and $P^{\text{thr}}_{\text{seedSF}}$, $P^{\text{thr}}_{\text{seedCF}}$ thresholds are presented. It is followed by a discussion of an additional implemented technique called *"double-pass"* that improves energy reconstruction. Finally, the hyper-parameter tuning of the model is presented.

### 6.5.1 Loss function weights

The output of the center-finder GNN consists of four different values: coordinates $x_{\text{reco}}$, $y_{\text{reco}}$, energy $E_{\text{reco}}$ and seed-score $P_{\text{seedCF}}$, as discussed in Section 6.4.5.

The loss function in this case has three components (the coordinates are combined in one term): *position loss* implemented as mean absolute error, *energy loss* implemented as mean absolute error as well, and *seed loss* implemented as focal cross-entropy loss. The details on the loss functions can be found in Section 4.1.2. Each of these terms can be additionally weighted by a special coefficient (weight) in order to reduce/increase its

importance in the learning process.

Equations (6.5) show the total loss function for each input along with its constituents used to train the center-finder GNN.

$$\begin{aligned}
\mathcal{L}\text{position} &= \frac{1}{2 \cdot 4} \sum_{i=1}^{4} \left( |x^i_{\text{reco}} - x^i_{\text{gen}}| + |y^i_{\text{reco}} - y^i_{\text{gen}}| \right), \\
\mathcal{L}\text{energy} &= \frac{1}{4} \sum_{i=1}^{4} |E^i_{\text{reco}} - E^i_{\text{gen}}|, \\
\mathcal{L}\text{seed} &= -\frac{1}{4} \sum_{i=1}^{4} \alpha (1 - P^i_{\text{seedCF}})^{\gamma} \log(P^i_{\text{seedCF}}), \\
\mathcal{L}\text{total} &= k_{\text{p}} \cdot \mathcal{L}\text{position} + k_{\text{e}} \cdot \mathcal{L}\text{energy} + k_{\text{s}} \cdot \mathcal{L}\text{seed},
\end{aligned} \tag{6.5}$$

where $x^i_{\text{reco}}$, $y^i_{\text{reco}}$, $E^i_{\text{reco}}$ are the reconstructed coordinates and energy for the seed window $i$; $x^i_{\text{gen}}$, $y^i_{\text{gen}}$, $E^i_{\text{gen}}$ are the generated coordinates and energy of the associated particle; $P^i_{\text{seedCF}}$ is the predicted seed score; $\gamma$, $\alpha$ are parameters of the focal loss, chosen to be $\gamma = 2$, $\alpha = 0.25$; and $k_{\text{p}}$, $k_{\text{e}}$, $k_{\text{s}}$ are the weights attributed to the loss associated with the position loss, energy loss, and seed loss, respectively. They are further referred to as *position weight*, *energy weight*, and *seed weight*.

The goal of the selection of the loss weights is to find an optimal value that leads to the best performance by achieving convergence of all three loss terms. During the training, the position and energy validation losses are observed to always reach a relative plateau, while the validation seed loss indicates overfitting. This can be seen from Fig. 6.25, where the loss evolution versus epoch for different seed weights $k_{\text{s}}$ is presented both for training (left) and validation (right). Due to this reasoning, the loss weight optimization only focuses on $k_{\text{s}}$, while $k_{\text{p}} = 1$ and $k_{\text{e}} = 1$.

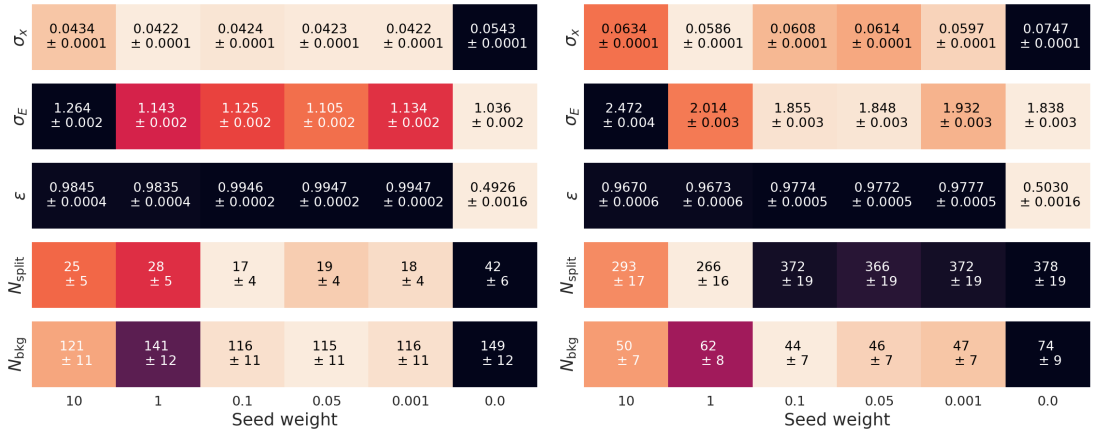



Figure 6.24: Summary values obtained for different seed weights $k_s$ for single-photon (left) and two-photon (right) datasets.

The optimal values are chosen by monitoring the loss function and comparing the performance achieved with different loss weights $k_{\text{s}}$. The summary results for various important performance metrics, such as position ($\sigma_x$) and energy ($\sigma_E$) resolutions, overall signal efficiency ($\epsilon$), the overall number of background events ($N_{\text{bkg}}$), and the overall number of splitted events ($N_{\text{split}}$), combined over the whole generated energy range $E_{\text{gen}}$, are shown in Fig. 6.24 both for single-photon and two-photon datasets. They are evaluated on the best-performing model for each $k_{\text{s}}$, meaning the results are estimated on the model taken at the end of the epoch with the lowest validation loss.

As can be seen from Fig. 6.24, the performance of the networks for different $k_{\text{s}}$

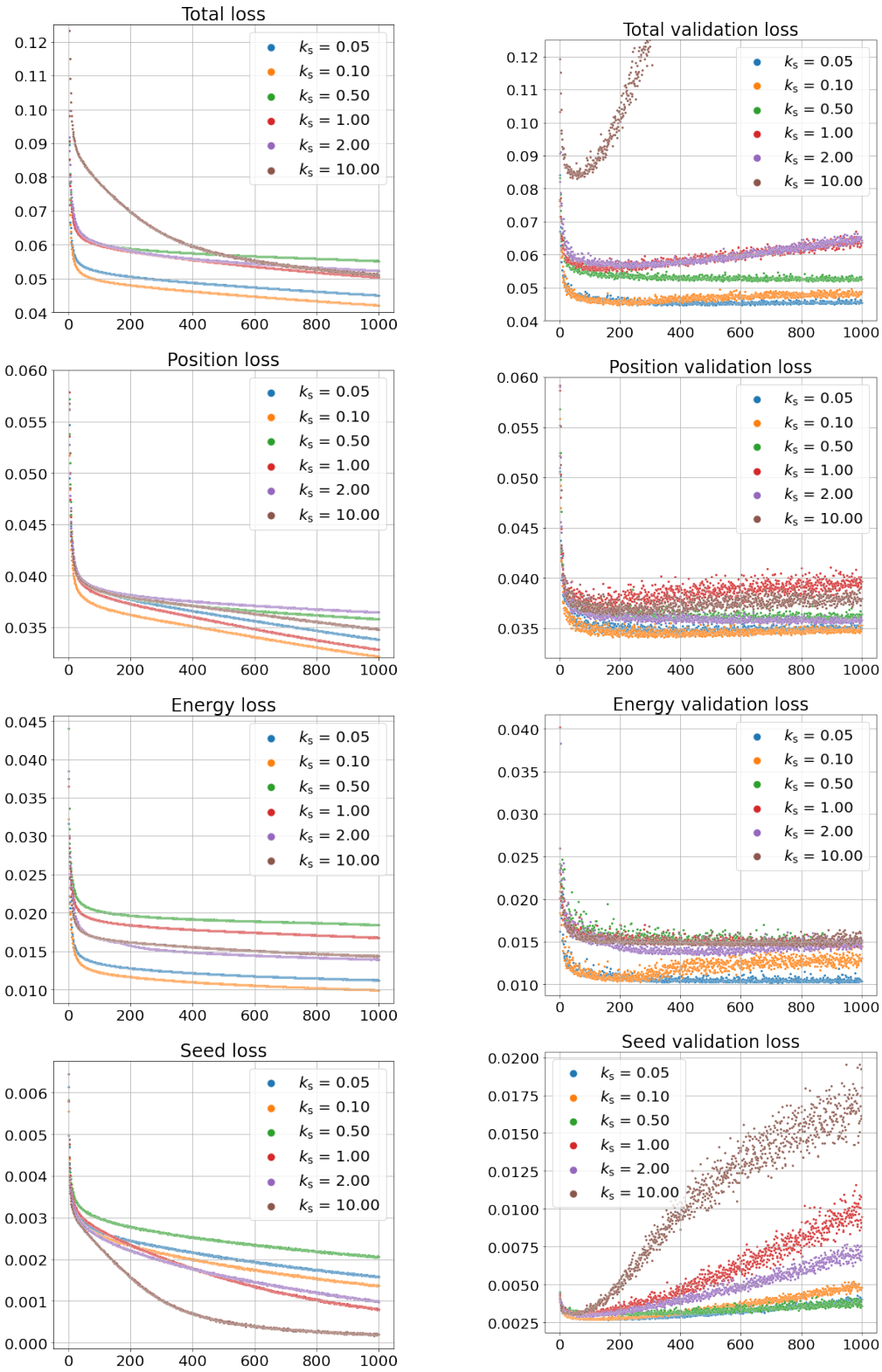



Figure 6.25: The loss functions evolution versus epoch, presented for different seed coefficients ($k_s$). The position, energy, seed, and total training losses are shown on the left, while the corresponding validation losses are shown on the right.

differs only marginally. The value $k_s = 0.05$ is selected for the seed weight based on the energy resolution results as it is the only statistically significant metric in this case. The best-performing model (lowest validation loss) for the chosen seed weight is achieved at epoch $n = 321$. At this point, all three loss terms are converged.

### 6.5.2 Seed-score thresholds

In the final DeepCluster model implementation, the seed-finder NN and the center-finder NN predict two different seed scores: $P_{\text{seedSF}}$, $P_{\text{seedCF}}$. They are both used in the evaluation phase to create an optimal collection of reco-objects by applying the respective thresholds $P^{\text{thr}}_{\text{seedSF}}$ and $P^{\text{thr}}_{\text{seedCF}}$ on them. These thresholds are optimized in order to achieve excellent resolution and high signal efficiency ($\epsilon$) while minimizing background ($N_{\text{bkg}}$) and splitting ($N_{\text{split}}$) yields.

The optimization is done on a single-photon dataset and the following performance metrics are monitored: position ($\sigma_x$) and energy ($\sigma_E$) resolutions, overall signal efficiency ($\epsilon$), the overall number of background events ($N_{\text{bkg}}$), and the overall number of splitted events ($N_{\text{split}}$), combined over the whole generated energy range $E_{\text{gen}}$. The results achieved with different $P^{\text{thr}}_{\text{seedSF}}$ and $P^{\text{thr}}_{\text{seedCF}}$ (the values are taken from 0 to 1 with a step of 0.1) are shown in Fig. 6.26.

According to these results, $P^{\text{thr}}_{\text{seedSF}} = 0.3$ and $P^{\text{thr}}_{\text{seedCF}} = 0.4$ are selected as they ensure excellent performance in terms of resolution ($\sigma_x^{(0.3,0.4)} = 0.02$ crystal, $\sigma_E^{(0.3,0.4)} = 0.56$ GeV) and high signal efficiency ($\epsilon^{(0.3,0.4)} = 99.5\%$). Although the background yield ($N_{\text{bkg}}^{(0.3,0.4)} = 153$) is more significant for the chosen thresholds compared to larger values, it is still ≈2000 times lower than those of the PFClustering method.

The splitting performance can be recovered using another optimization procedure called "double pass", which is described in the next section.

### 6.5.3 Double pass

The main goal of the DeepCluster model is to be able to reconstruct two close-by particles. The performance for this case can be estimated from the signal efficiency results for the two-photon dataset (Fig. 6.19), where the DeepCluster network achieves significantly higher efficiencies compared to PFClustering. Both for the model and the PFClustering the closer particles are to each other, the harder it is to disentangle them in the reconstruction process.

A limit on the distance between the two photons can be evaluated, indicating the minimal spatial separation when both of them can still be correctly reconstructed with the model. Figure 6.27 helps to explore this limit in more detail. It shows two different values estimated on the two different datasets:

- **Distance between generated particles**. For this value, the two-photon dataset is considered. The plot shows the distribution of generated $\Delta R$ between two particles. Only particles that are correctly reconstructed (meaning both of them have a link to a reco-object) are considered. For $\Delta R < 0.3$ crystal the amount of particles that still can be separated by the network is negligible.
- **Distance between reco-objects**. For this value, the single-photon dataset is considered. The plot shows the distribution of reconstructed $\Delta R$ between two reco-objects that are linked to the same generated particle (the case of splitted events). The majority of the events appear for $\Delta R < 0.3$.

In order to further mitigate the splitting yield while maintaining high signal efficiency for two close-by photons, a dedicated procedure called "double-pass" is implemented. For each sample, it follows several steps:

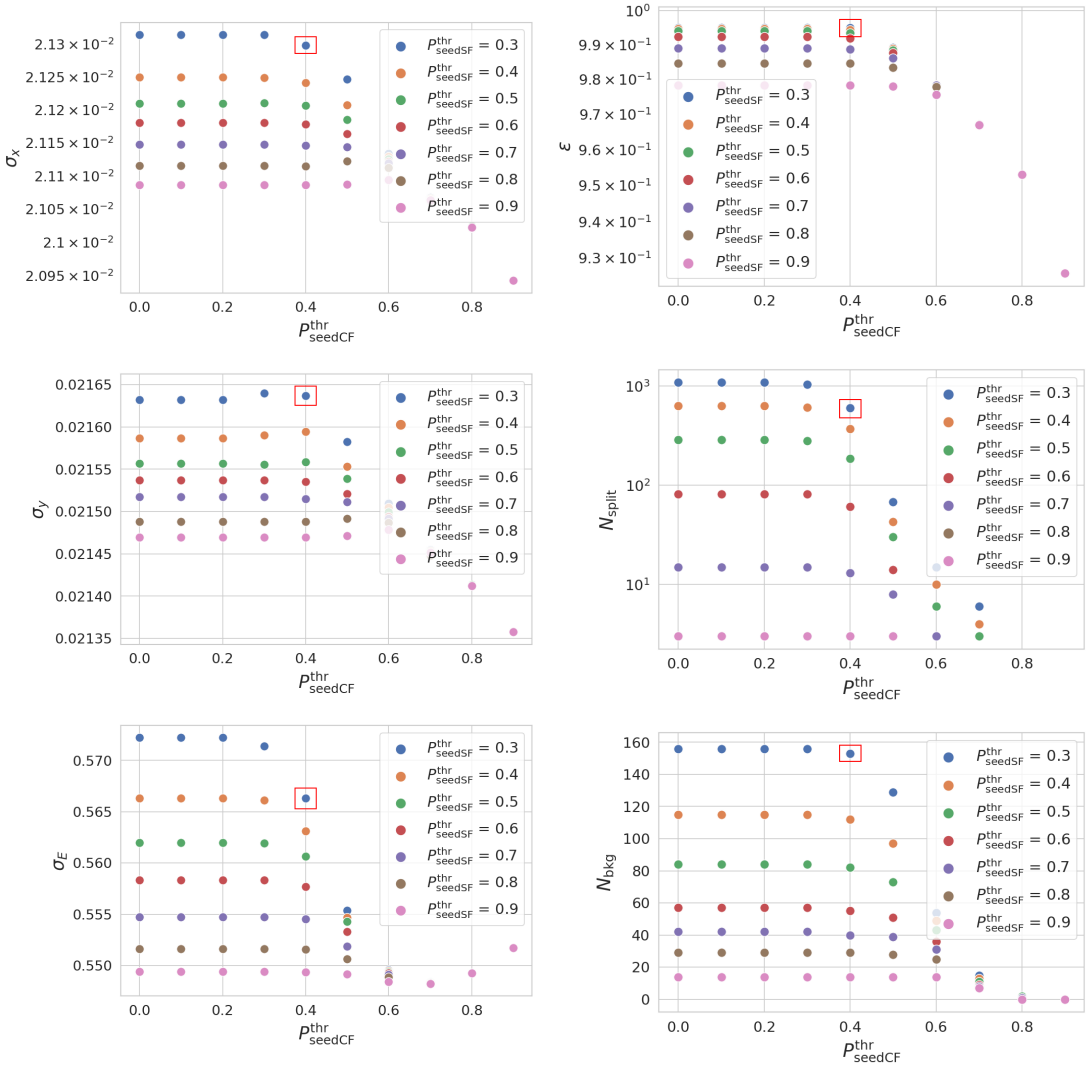



Figure 6.26: The results for performance metrics are shown for different seed-score threshold values $P^{\text{thr}}_{\text{seedCF}}$, and $P^{\text{thr}}_{\text{centerCF}}$. On the left: position $\sigma_x$ (top), $\sigma_y$ (middle), and energy $\sigma_E$ (bottom) resolutions are presented. On the right: signal efficiency (top), splitting yield (middle), and background yield (bottom) are presented. The red rectangular indicates the results for the selected threshold values $P^{\text{thr}}_{\text{seedCF}} = 0.3$ and $P^{\text{thr}}_{\text{centerCF}} = 0.4$.

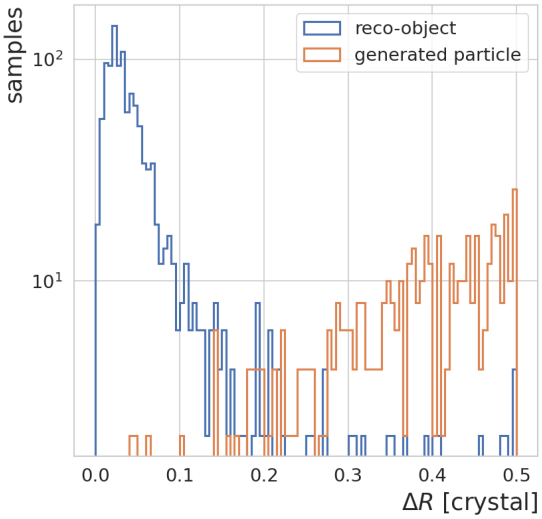



Figure 6.27: Distributions of $\Delta R$ between two generated particles in the two-photon dataset (generated particle) and $\Delta R$ between two reco-objects constructed from the single-photon dataset (reco-object).

1. All the groups of reco-objects with a distance $\Delta R < 0.3$ crystal are selected.
2. From these pairs, the reco-object with the highest reconstructed energy is chosen. The remaining reco-objects in the group are discarded.
3. The seed window associated with the chosen reco-object is passed for the second time through the center-finder NN.

In the double-pass procedure, the neighbors that are too close to each other are eliminated. It leads to an improved prediction on the single-photon dataset, as, in this case, the network only processes a single seed window without neighbors, and, by construction can not predict more than one reco-object. As previously discussed, the performance for the two-photon dataset is not significantly degraded by adding the double-pass technique, as the amount of samples for the chosen limit ($\Delta R = 0.3$ crystal) is small.

| | | $\epsilon$ | $N_{\text{split}}$ |
|---|---|---|---|
| Without Double Pass | single-photon | 99.49 | 605 |
| | two-photon | 97.59 | 824 |
| With Double Pass | single-photon | 99.47 | 34 |
| | two-photon | 96.98 | 345 |

Table 6.5: Result for signal efficiency ($\epsilon$) and splitting yield ($N_{\text{split}}$) for the DeepCluster model with and without a double pass (DS) for single- and two-photon datasets.

The effect of adding this method can be seen in Fig. 6.28, which shows the comparison of signal efficiency and splitting yield between the DeepCluster models with and without the double pass. The obtained values integrated over the full energy range $E_{\text{gen}}$ are summarised in Table 6.5. Including the double pass method largely reduces the splitting yield while only slightly affecting the signal efficiency.

The improvement in performance with the double pass can also be seen in Fig 6.29, where the distribution of the ratio between the total reconstructed energy $E_{\text{reco}}^{\text{tot}}$ and

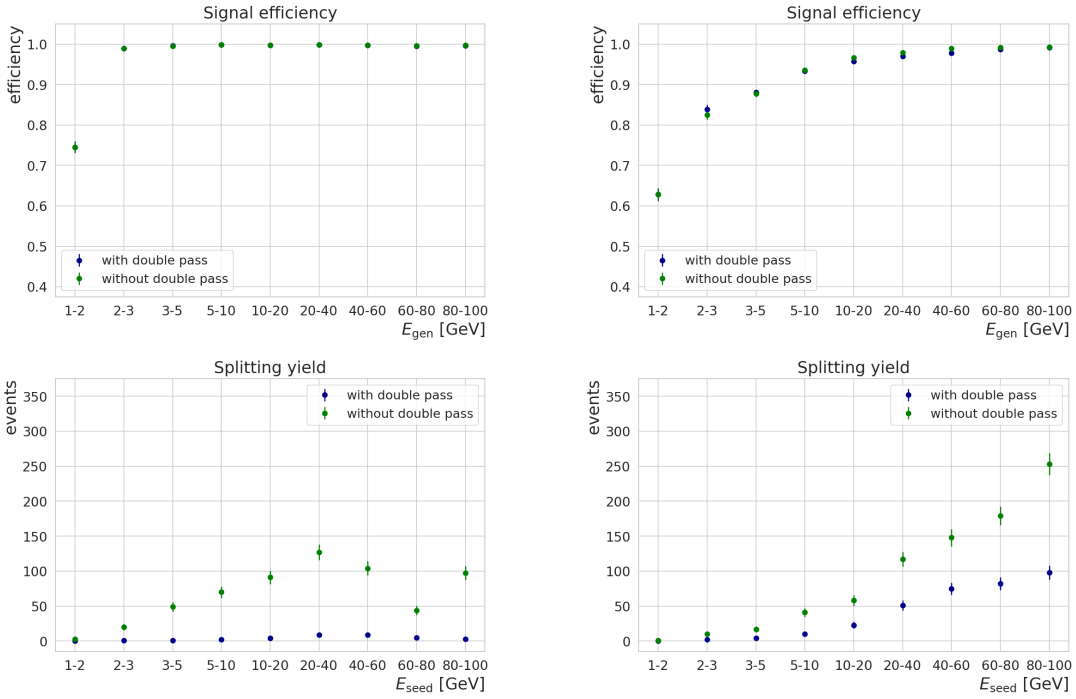



Figure 6.28: Signal efficiency (top) and splitting yield (bottom) values. The results are obtained by applying the DeepCluster model with and without the double pass technique on the single-photon (left) and two-photon (right) test datasets.

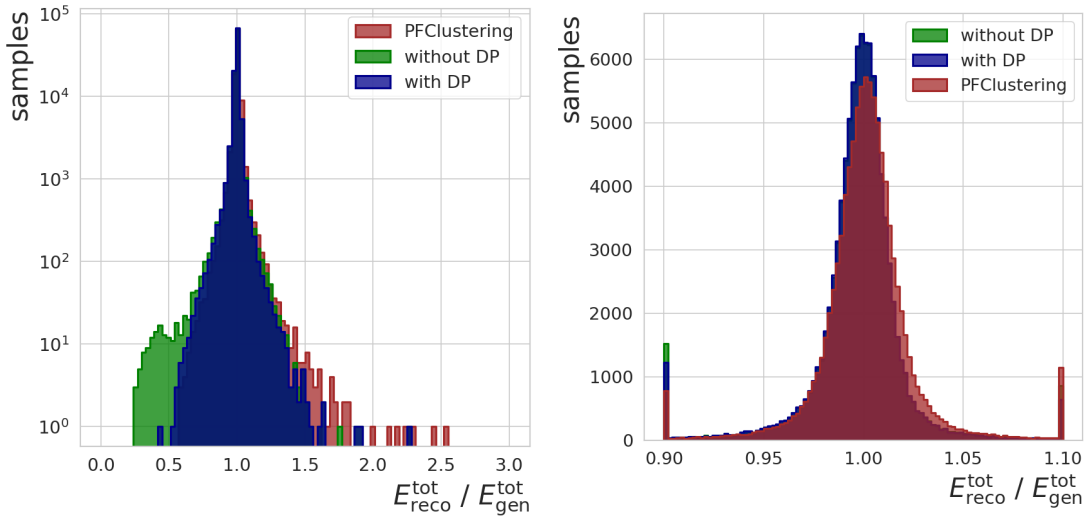



Figure 6.29: The distribution of the ratio between the total reconstructed energy $E^{\text{tot}}_{\text{reco}}$ and the total generated energy $E^{\text{tot}}_{\text{gen}}$. The results are obtained by applying the DeepCluster model with and without the double pass (DP) technique on the single-photon test dataset. They are shown in log (left) and linear (right) scales.

total generated energy $E^{\text{tot}}_{\text{gen}}$ (sum of the energies of all the simulated particles) with and without the double pass for single-photon dataset is presented. It shows that the energy underestimation is reduced with the double pass.

The effect is explained by the specifics of the DeepCluster model reconstruction. When one photon gives rise to two reco-objects, the generated energy is distributed between them. In the evaluation phase, after the center-finder score threshold $P^{\text{thr}}_{\text{seedCF}}$ is

applied, one of these reco-objects can be potentially removed, resulting in the energy underestimation, as shown in Fig 6.29. With the double pass, however, only one reco-object is produced without any energy sharing, which leads to better energy prediction.

### 6.5.4 Hyperparameter optimization

The final optimization procedure applied to the DeepCluster model is the tuning of its internal parameters (or hyperparameters) in order to achieve the best possible performance. This procedure is done using Bayesian optimization, which is discussed in detail in Section 4.3.3. The hyperparameters that are optimized and their considered ranges are presented in Table 6.6.

| **Parameter** | **Value** |
|---|---|
| 1st convolutional layer | number of filters: [16, 32, 48, 64, 80, 96, 112, 128], kernel size: [1, 3] |
| 2nd convolutional layer | number of filters: [16, 32, 48, 64, 80, 96, 112, 128], kernel size: [1, 3] |
| Common dense layers | nodes: [500, 700, 900, 1100] |
| Center dense layer | nodes: [250, 500, 750] |
| Energy dense layer | nodes: [50, 100, 150, 200, 250, 300] |
| Seed dense layer | nodes: [50, 100, 150, 200, 250, 300] |
| Dropout | [0.1, 0.3, 0.5] |

Table 6.6: The hyperparameters of the DeepCluster model and their considered ranges used by the Bayesian hyperparameter optimization.

The final chosen hyperparameters are:

- 1st convolutional layer: 128 filters with kernel size 3, batch normalization is applied.
- 2nd convolutional layer: 112 filters with kernel size 1, batch normalization is applied.
- Common dense layer: 1100 nodes, 0.1 dropout is applied.
- Center dense layer: 500 nodes, 0.3 dropout is applied.
- Energy dense layer: 100 nodes, 0.1 dropout is applied.
- Seed dense layer: 250 nodes, 0.3 dropout is applied.
- Batch size: 512.
- Learning rate: 0.0001.

The effect of the hyperparameter optimization is shown in Fig. 6.30, where the difference between $E_{\text{gen}}$ and $E_{\text{reco}}$ obtained with the DeepCluster model before (basis) and after (optimized) optimization for single-photon dataset is shown. With the optimization, the energy resolution is improved by 17%.

The final model is trained using LAMB optimizer [101]. The loss function is presented in Eq. 6.5. The model at epoch=321 is taken as it achieves the lowest value of the validation loss.

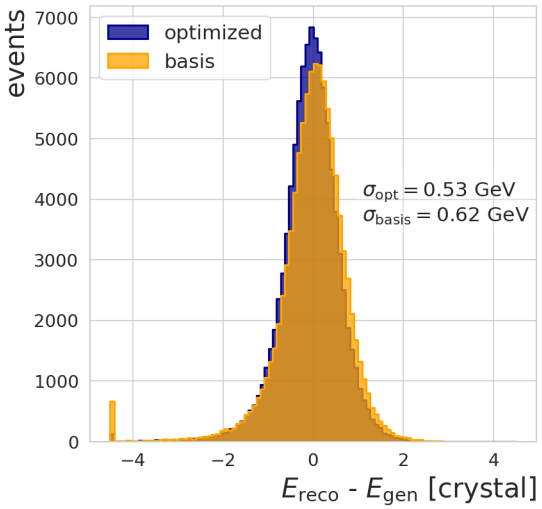



Figure 6.30: Difference between the reconstructed energy $E_{\text{reco}}$ and the generated energy of the particle $E_{\text{gen}}$. The results are obtained by applying the DeepCluster model before (basis) and after (optimized) hyperparameter optimization. The resolutions evaluated from these distributions are reported on the plots.

## 6.6 Results

In this section, the results obtained with the final optimized DeepCluster model are presented along with a comparison to the PFClustering algorithm.

### 6.6.1 Photons

All results are presented for single-photon and two-photon test datasets. The resolutions are evaluated as $\sigma$ = [quantile(0.84) - quantile(0.16)]/2.

Performance for position reconstruction is presented in Fig. 6.31 for the single-photon dataset and in Fig. 6.32 for the two-photon dataset. Performance for energy reconstruction is presented in Fig. 6.33 for the single-photon dataset and Fig. 6.34 for the two-photon dataset. Each plot demonstrates the difference between the reconstructed and generated values. The results are presented in bins of the energy of the generated particle ($E_{\text{gen}}$).

The signal efficiency, splitting yield, and background yield results for the DeepCluster model and the PFClustering algorithm are presented in Fig. 6.35 for the single-photon dataset on the left and for the two-photon dataset on the right. The results are presented versus the energy of the generated particle $E_{\text{gen}}$ for the signal efficiency and splitting yield, and versus the energy of the seed crystal $E_{\text{seed}}$ for the background yield.

The performance summary is presented in Table 6.7. The DeepCluster model outperforms the PFClustering algorithm in terms of position resolution across the full energy range both for the single- and two-photon cases. The energy reconstruction is improved with the model as well, apart from $E_{\text{gen}} = [1, 10]$ GeV and $E_{\text{gen}} = [10, 20]$ GeV for two photons, where it achieves slightly worse results compared to PFClustering.

Most notably, the signal efficiency for the two-photon dataset obtained with the DeepCluster model is 97.0% while with PFClustering it is only 82.0%.

In a more realistic scenario, with larger numbers of particles per sample, the performance will not be significantly degraded, as can be observed from the multiple-particle electron dataset, presented in the next section. As the network does not process the whole calorimeter window, but only separated 7x7 windows, the scalability for the full

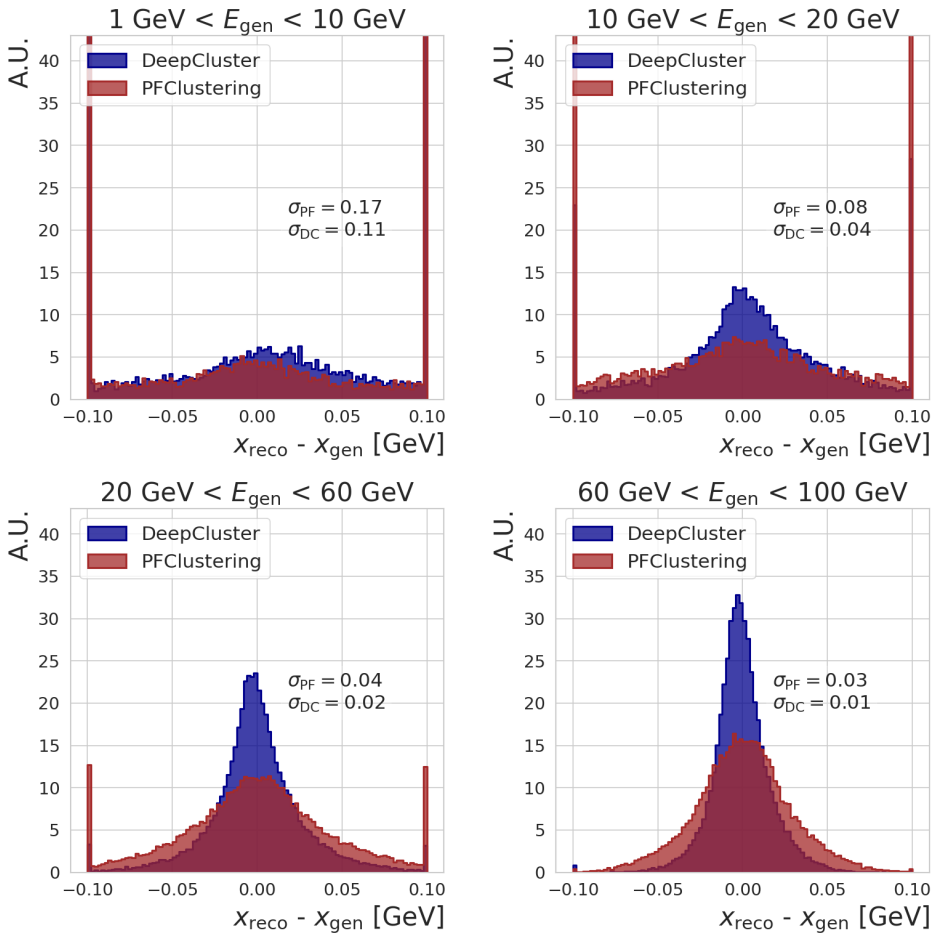



Figure 6.31: Difference between the reconstructed position $x_{\text{reco}}$ and the generated position of the particle $x_{\text{gen}}$. The results are obtained by applying the PFClustering (PF) algorithm and DeepCluster model on the single-photon test dataset. The distributions are presented in four bins of the generated photon energy $E_{\text{gen}}$: [1, 10] GeV, [10, 20] GeV, [20, 60] GeV, and [60, 100] GeV. The resolutions evaluated from these distributions are reported on the plots.

| | Single-Photon | | Two-Photon | |
|---|---|---|---|---|
| | PFClustering | DeepCluster | PFClustering | DeepCluster |
| $\sigma_x$ [crystal] | 0.04 ± 0.00 | 0.02 ± 0.00 | 0.08 ± 0.00 | 0.03 ± 0.00 |
| $\sigma_E$ [GeV] | 0.61 ± 0.00 | 0.55 ± 0.00 | 6.24 ± 0.01 | 0.92 ± 0.00 |
| $\epsilon$ | 99.8 ± 0.3 % | 99.5 ± 0.3% | 82.0 ± 0.3% | 97.0 ± 0.3% |
| $N_{\text{split}}$/100k photons | 0 | 34 ± 6 | 17 ± 4 | 345 ± 19 |
| $N_{\text{bkg}}$/100k toy samples | 350k ± 19k | 153 ± 12 | 320k ± 18k | 146 ± 12 |

Table 6.7: Performance comparison for position and energy resolutions, signal efficiency, splitting yield for 100k photons, and background yield for 100k toy simulation between PFClustering and DeepCluster algorithms for single- and two-photon datasets.

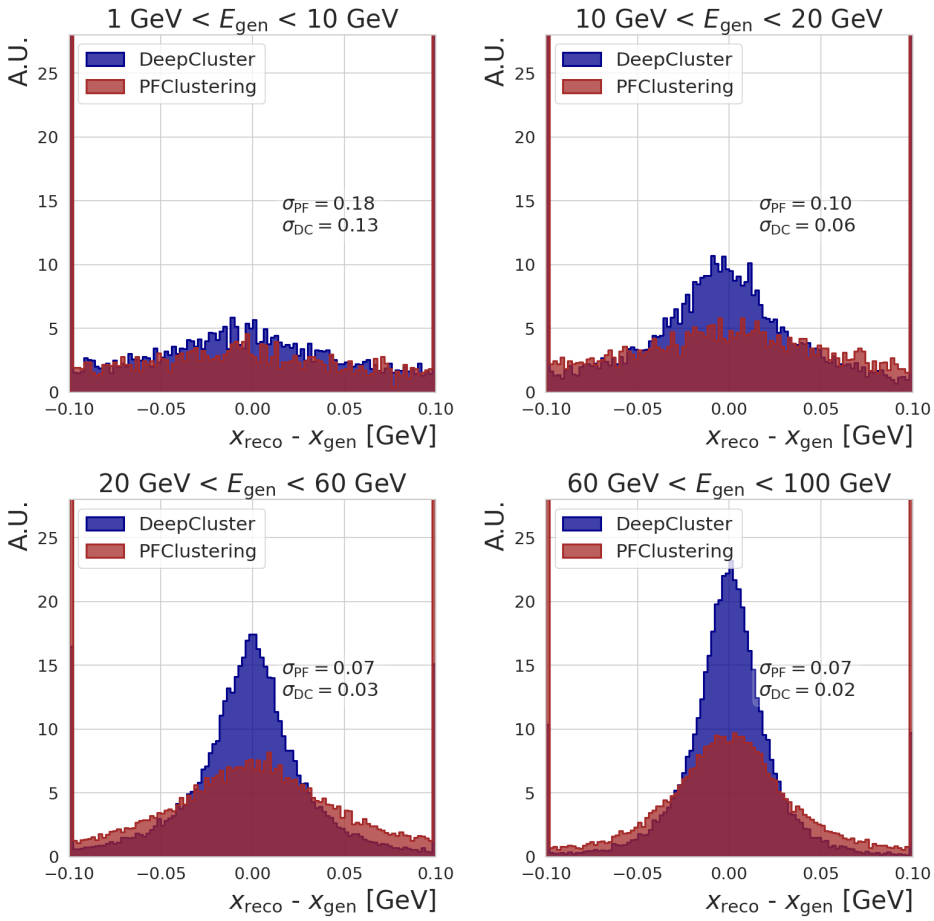



Figure 6.32: Difference between the reconstructed position $x_{\text{reco}}$ and the generated position of the particle $x_{\text{gen}}$. The results are obtained by applying the PFClustering (PF) algorithm and DeepCluster model on the two-photon test dataset. The distributions are presented in four bins of the generated photon energy $E_{\text{gen}}$: [1, 10] GeV, [10, 20] GeV, [20, 60] GeV, and [60, 100] GeV. The resolutions evaluated from these distributions are reported on the plots.

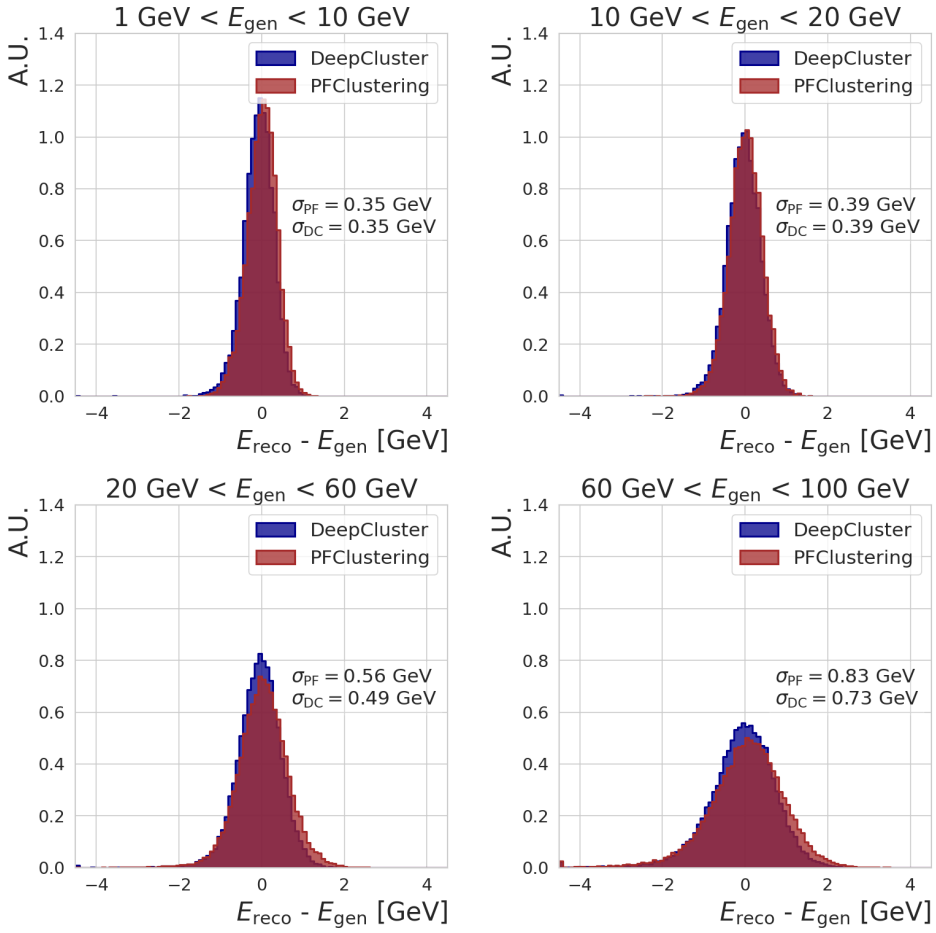



Figure 6.33: Difference between the reconstructed energy $E_{\text{reco}}$ and the generated energy of the particle $E_{\text{gen}}$. The results are obtained by applying the PFClustering (PF) algorithm and the DeepCluster model on the single-photon test dataset. The distributions are presented in four bins of the generated photon energy $E_{\text{gen}}$: [1, 10] GeV, [10, 20] GeV, [20, 60] GeV, and [60, 100] GeV. The resolutions evaluated from these distributions are reported on the plots.

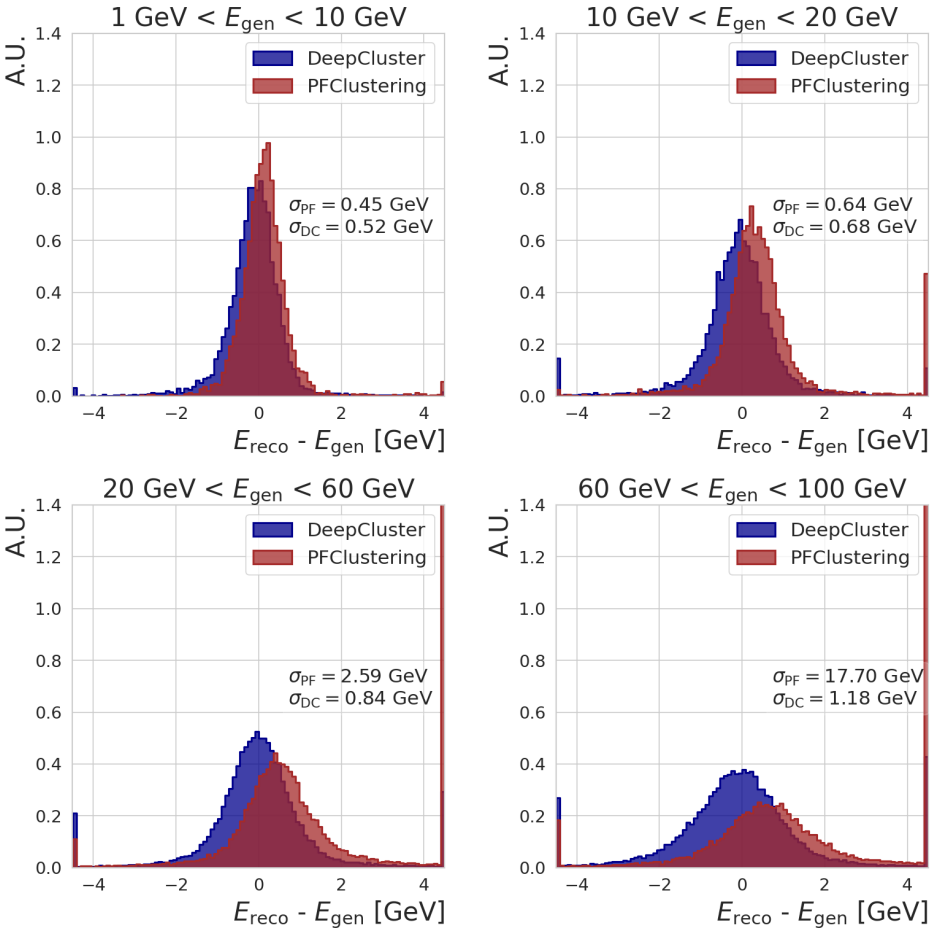



Figure 6.34: Difference between the reconstructed energy $E_{\text{reco}}$ and the generated energy of the particle $E_{\text{gen}}$. The results are obtained by applying the PFClustering (PF) algorithm and DeepCluster model on the two-photon test dataset. The distributions are presented in four bins of the generated photon energy $E_{\text{gen}}$: [1, 10] GeV, [10, 20] GeV, [20, 60] GeV, and [60, 100] GeV. The resolutions evaluated from these distributions are reported on the plots.

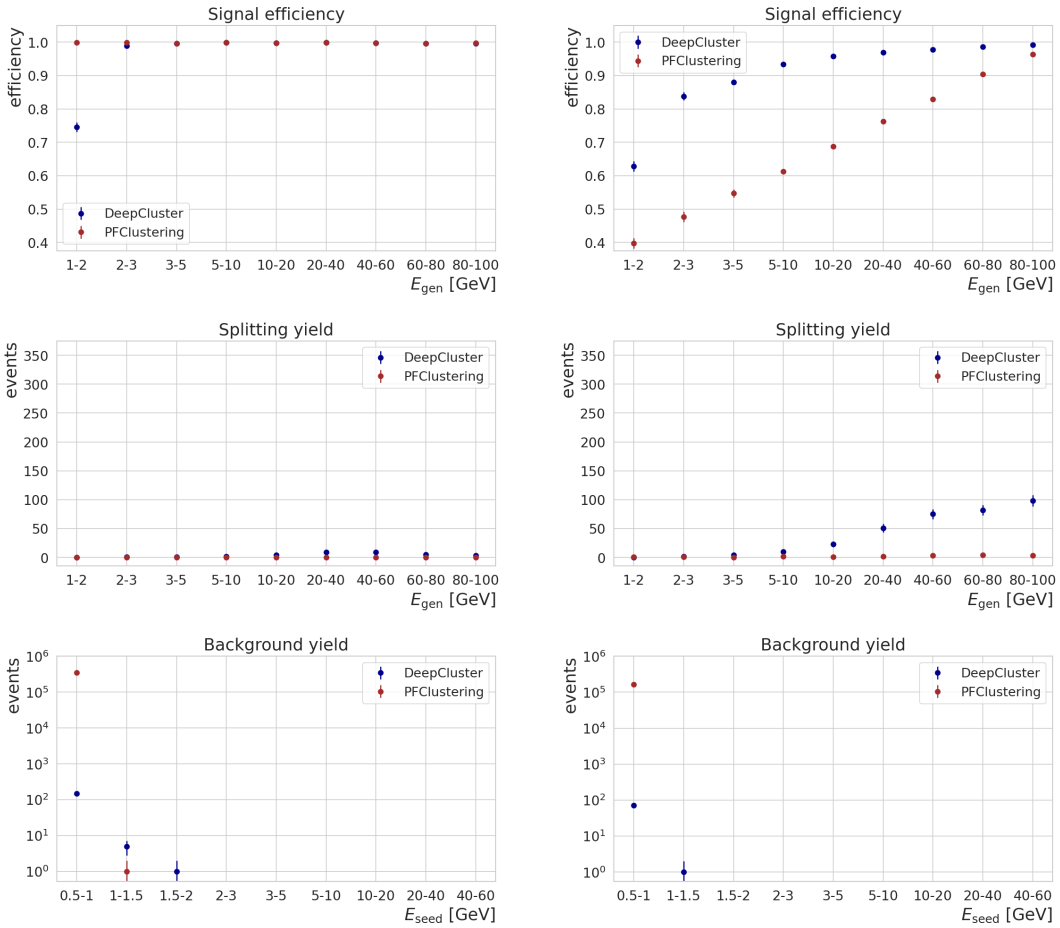



Figure 6.35: Signal efficiency (top), splitting yield (middle), and background yield (bottom) values. The results are obtained by applying the PFClustering and DeepCluster model on the single-photon (left) and two-photon (right) test datasets.

ECAL will not pose a problem as well. In this case, the particles are either far apart in the calorimeter (single-photon dataset) or in close proximity (two-photon dataset), both of which are covered during the model training.

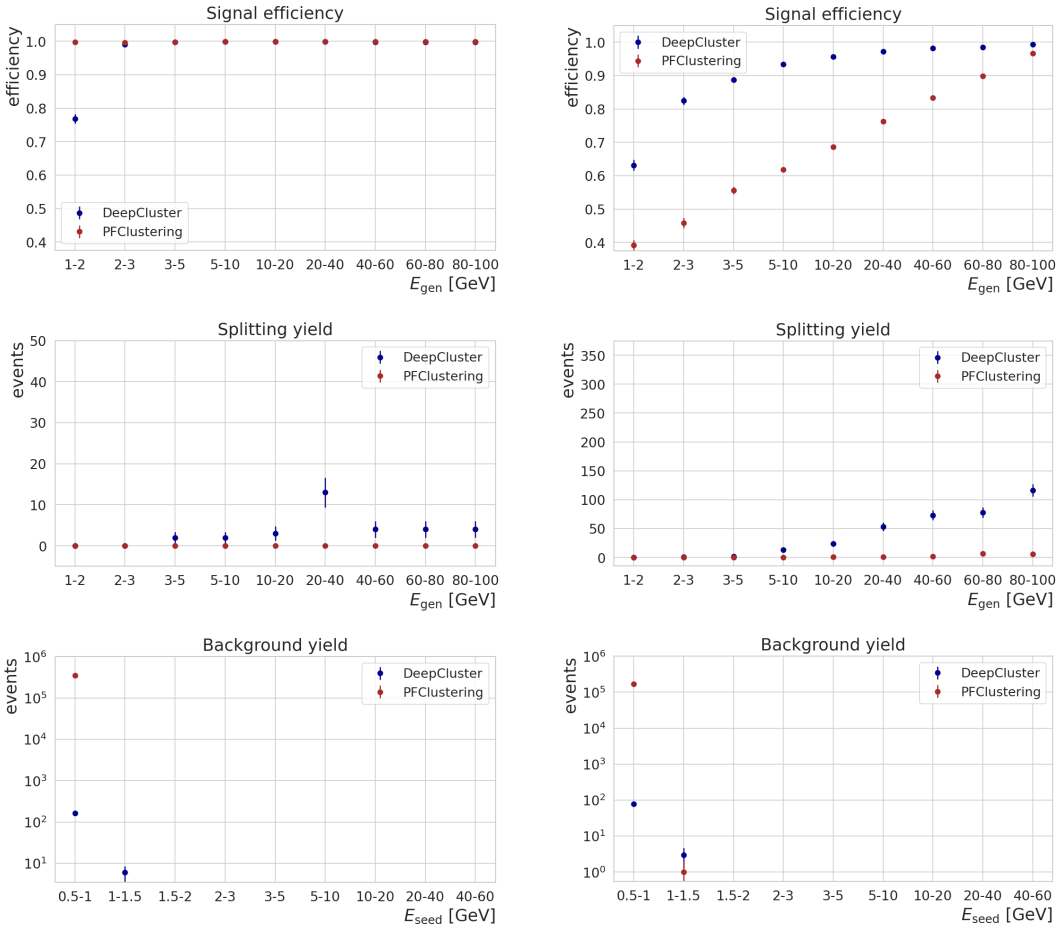



Figure 6.36: Signal efficiency (top), splitting yield (middle), and background yield (bottom) values. The results are obtained by applying the PFClustering and DeepCluster model on the single-electron (left) and two-electron (right) test datasets.

### 6.6.2 Electrons

The DeepCluster model performance on the single- and two-electron datasets is identical to the photon ones, underlining the robustness of the model. Figure 6.36 presents the results for signal efficiency, splitting yield, and background yield for the single-electron dataset on the left and the two-electron dataset on the right. The position and energy results are also very similar to the ones achieved on the photon dataset and are omitted for brevity.

The DeepCluster model is also tested on the multiple-particle electron dataset, where each sample contains up to six particles.

Performance for position and energy reconstruction is shown in Fig. 6.37. Each plot demonstrates the difference between the reconstructed and generated values.

The signal efficiency, splitting yield, and background yield results for the DeepCluster model and the PFClustering algorithm are presented in Fig. 6.38.

Even though the network is not specifically trained on the electron dataset, it still shows excellent performance, which is competitive with PFClustering. The position and energy resolution with DeepCluster reconstruction is improved by 50% and 6%, respectively. Overall (evaluated along all energy bins) signal efficiency is similar for both algorithms: 70.8% for the DeepCluster and 70.6% for the PFClustering. The splitting yield is higher for the model: 539 versus 5 for the PFClustering. The background yield is significantly reduced: 100 for the DeepCluster versus 100k for the PFClustering.

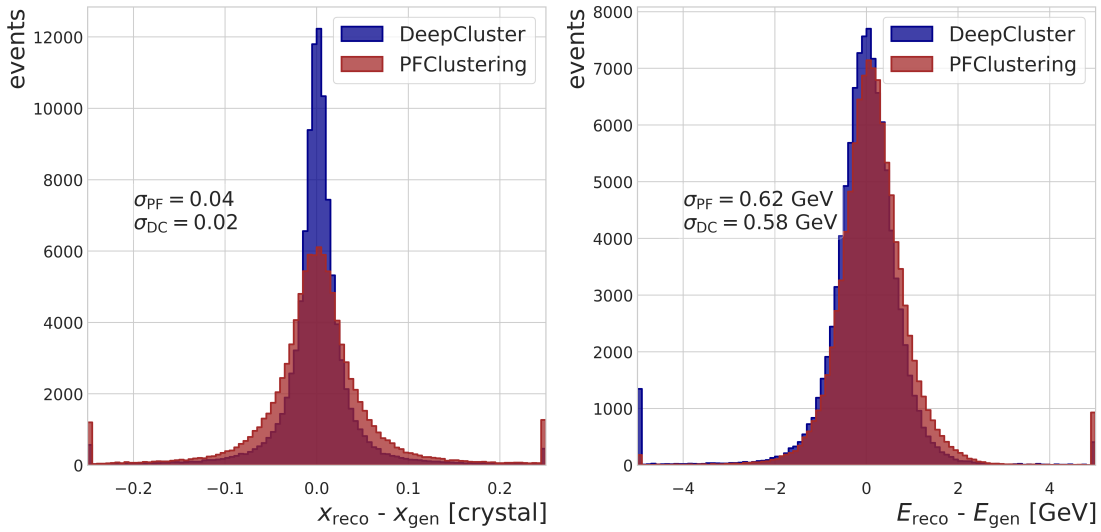



Figure 6.37: On the left: the difference between the reconstructed position $x_{\text{reco}}$ and the generated position of the particle $x_{\text{gen}}$. On the right: the difference between the reconstructed energy $E_{\text{reco}}$ and the generated energy of the particle $E_{\text{gen}}$. The results are obtained by applying the PFClustering (PF) algorithm and the DeepCluster model on the multiple-particle electron dataset. The resolutions evaluated from these distributions are reported on the plots.

### 6.6.3 Pions

Finally, the results achieved on the $\pi^0$ sample (described in Section 6) are presented. In this case, the reconstruction algorithms have to detect both of the photons produced as the result of $\pi^0$ decay and correctly define their energy. With this information, the mass of $\pi^0$ can be reconstructed as:

$$m_{\pi^0} = \sqrt{2E^1_{\text{reco}}E^2_{\text{reco}}\,(1 - cos(\theta))}, \tag{6.6}$$

where $E^1_{\text{reco}}$, $E^2_{\text{reco}}$ are the reconstructed energies of two photons and $\theta$ is the decay angle between them.

$\pi^0$ mass distributions reconstructed with the DeepCluster model and PFClustering are presented in Fig. 6.39. The results are shown in bins of the generated momentum $p_{\text{gen}}$ of $\pi^0$. The model achieves excellent results, outperforming the PFClustering $\pi^0$ detection efficiency by a factor of more than 2. Moreover, the mass resolution is significantly better with the model, which can be seen from the presented plots.

Examples of event displays are presented in Fig. 6.40 (left), where both algorithms only reconstruct one particle, Fig. 6.40 (right), where both photons are reconstructed by the network and PFClustering and Fig. 6.41, where only DeepCluster model is able to correctly identify both particles.

As in the case of electrons, the model is not specifically trained on the $\pi^0$ dataset. Moreover, the photons coming from the decay enter the toy calorimeter under different angles and not perpendicularly as in the training sample, which additionally indicates the robustness of the network.

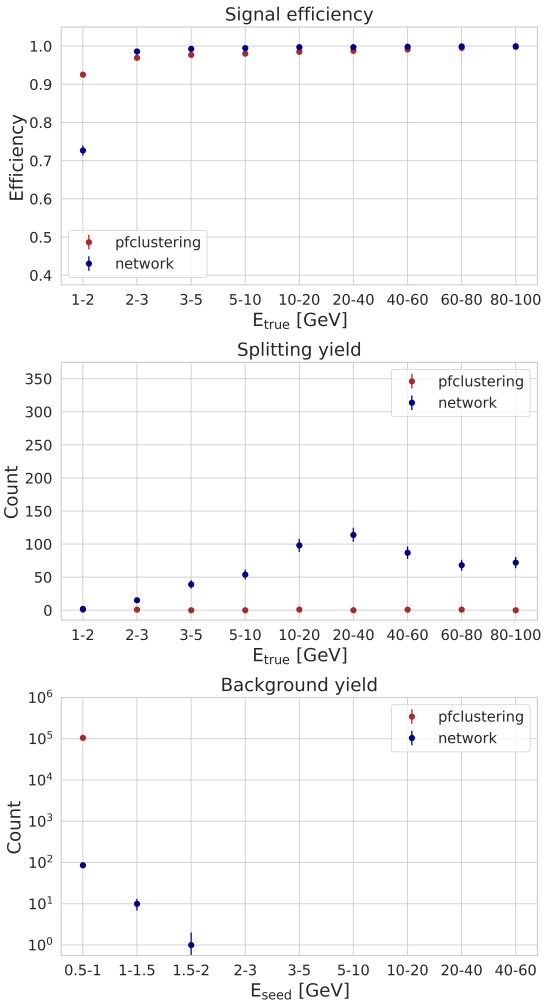



Figure 6.38: Signal efficiency (top), splitting yield (middle), and background yield (bottom) values. The results are obtained by applying the PFClustering and DeepCluster model on the multiple-particle electron dataset.

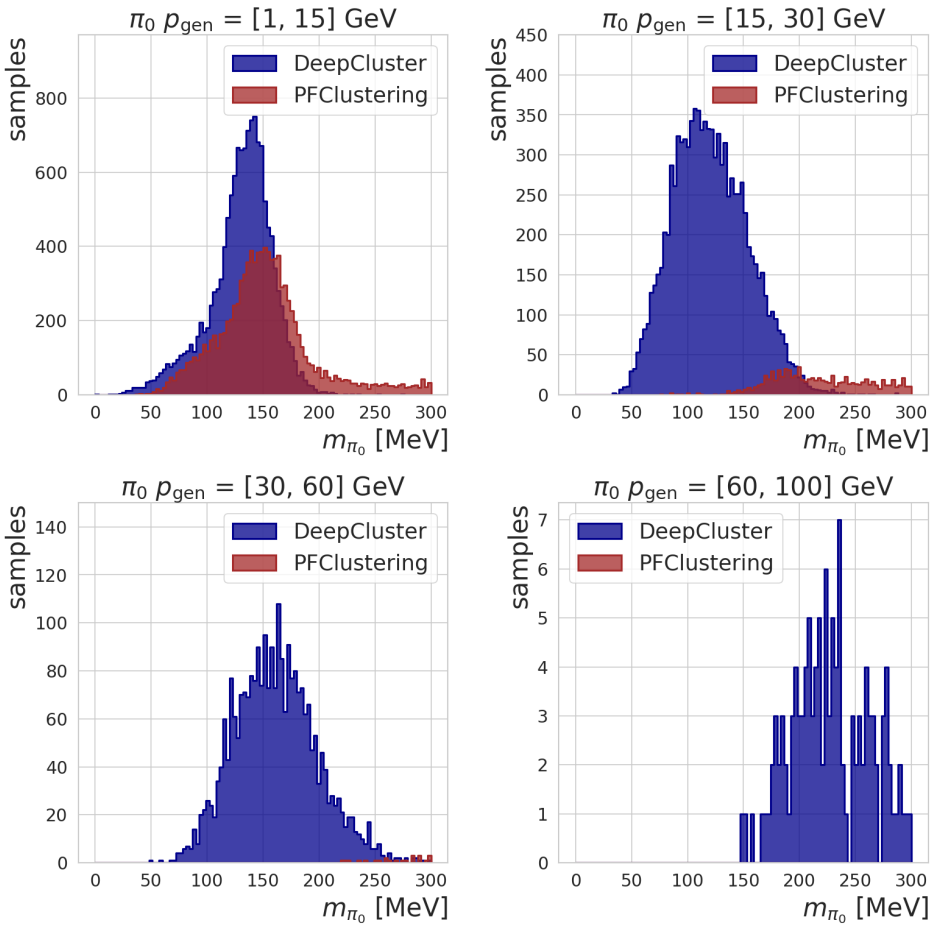



Figure 6.39: $\pi^0$ mass distribution reconstructed with DeepCluster model and PFClustering algorithm. The results are shown in bins of the generated momentum $p$ of $\pi^0$.

## 6.7 Conclusion and perspectives

This chapter focuses on the DeepCluster model, an innovative reconstruction technique designed specifically for standalone electromagnetic particles in the ECAL. It was created and fully developed as part of this thesis and represents the majority of the performed work.

To evaluate the developed methods and compare results with the traditional approach (PFClustering), a dedicated simplified simulation of the ECAL is created. Three different implementations of the DeepCluster model are tested. A two-step network approach based on convolutional and graph architectures yields the most optimal results and overcomes all encountered difficulties.

The final optimized model is tested on the single- and two-photon datasets as well as the multiple-particle electron dataset. In all of the cases, the DeepCluster shows superior performance compared to the PFClustering method in coordinate and energy resolutions, as well as background rejection and signal efficiency.

The main limitation of the traditional approach is its inherent difficulty in distinguishing between closely spaced particles, which leads to degraded performance in $\pi^0$ particle identification. In contrast, the DeepCluster model demonstrates excellent results by reconstructing approximately twice as many $\pi^0$ particles. This result holds promising prospects for the application of the DeepCluster model for ECAL reconstruction.

In subsequent development stages, the DeepCluster model will be integrated into the

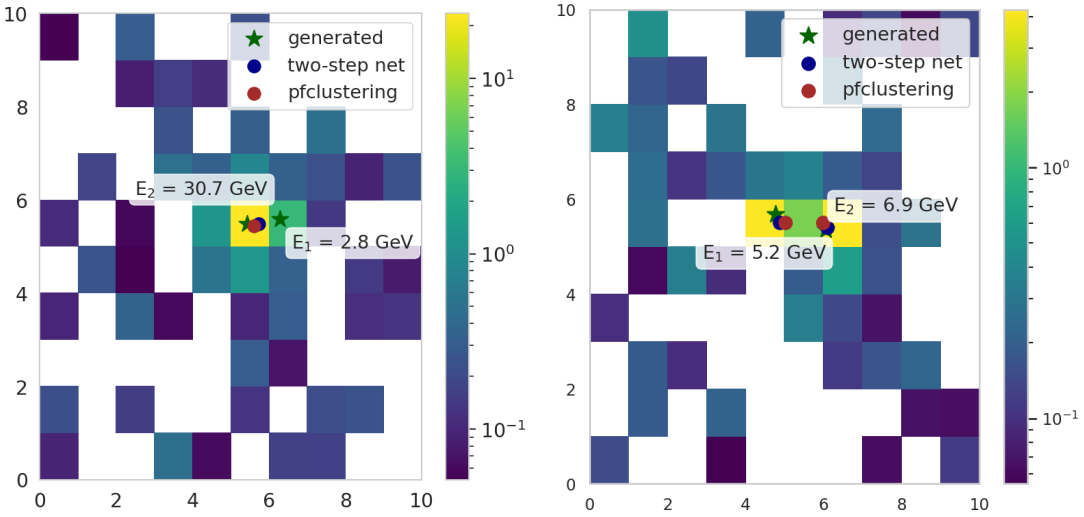



Figure 6.40: Two event displays for $\pi^0$-decay in the toy calorimeter. On the left: a case where both PFClustering and DeepCluster model can only reconstruct one particle. On the right: a case where both of the algorithms correctly reconstruct two photons.

global software of the CMS experiment to undergo testing with the real ECAL detector under more realistic physics conditions.

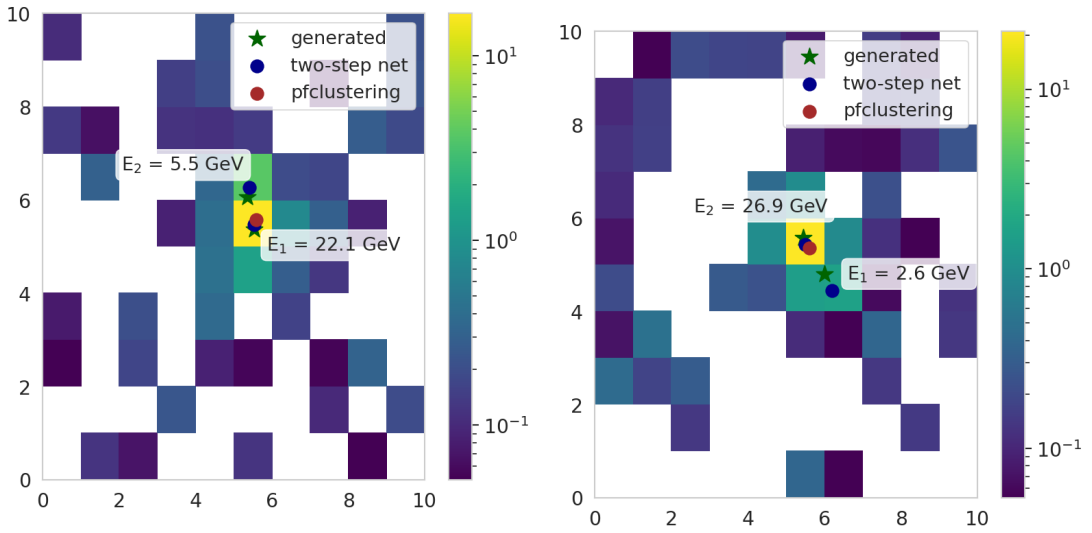



Figure 6.41: Two event displays for $\pi^0$-decay in the toy calorimeter. In these cases, only the DeepCluster model can correctly reconstruct two photons.

# Conclusion

The work performed during this thesis was dedicated to three different topics. All of them were carried out within the context of the CMS experiment and have a strong potential to benefit the further performance of the detectors and physics analysis.

The first subject was the development of the data acquisition system for the new MIP Timing Detector (MTD). Specifically, the focus was on the development of the DAQ software framework and a module used to reconstruct physical variables (time, charge) from the signal of the readout TOFHIR chip. During the subsequent system tests, the created software was validated and the timing resolution of 23 ps was achieved, which is within the bounds of the MTD requirements.

The MTD DAQ software is currently under continuous development with the target of including all the necessary features to operate with both the barrel and endcap parts of the detector. It will be further examined during the test beams and used for the detector commissioning.

The second subject was the SuperClustering reconstruction in the electromagnetic calorimeter (ECAL) of CMS. A new DeepSC model was created based on Graph Neural Networks and Self-Attention layers. The focus of this thesis was on the development of the particle identification feature — a novel approach to distinguishing electromagnetic objects and background using only ECAL information at the reconstruction step. A proof-of-concept study was performed, showing that using this extra information improves the signal efficiency of the current algorithm used in CMS particle flow by 20% for a background efficiency of 10%.

The further steps for the DeepSC and, in particular, the particle identification part may include adding it to the general reconstruction flow as an additional discriminating variable and testing its performance on real data (using $Z$ boson decaying to electrons) and its impact for analyses which might strongly benefit from it (e.g. $H \to \gamma\gamma$).

Finally, the majority of the work performed during the thesis was dedicated to the standalone electromagnetic object reconstruction in ECAL. The DeepCluster model was developed in order to overcome the limitations of the current PFClustering algorithm used for this task. It was based both on Graph and Convolutional Neural Networks. The full study from simulating a simplified calorimeter to comparing the results on $\pi^0$ samples was carried out. The new approach outperforms the PFClustering both in coordinate and energy resolution but, most importantly, in signal efficiency evaluated for two close-by photons: 97% for the DeepCluster model versus 82% for the PFClustering. It is particularly valuable for the cases with two close-by photons, as in $\pi^0$ decay or $H \to AA \to 4\gamma$ analysis as discussed in Chapter 1.

The outlook for this project includes implementing the DeepCluster model in the general software of the CMS experiment in order to test it in the real detector condition and, potentially, in application to physics analyses.

# Bibliography


[1] CMS Collaboration. "Search for exotic Higgs boson decays H $\to \mathcal{A}\mathcal{A} \to 4\gamma$ with events containing two merged diphotons in proton-proton collisions at $\sqrt{s} = 13$ TeV", 2022. URL https://arxiv.org/abs/2209.06197.

[2] S. Weinberg. "The making of the Standard Model". *The European Physical Journal C*, 34(1):5–13, may 2004. URL https://doi.org/10.1140%2Fepjc%2Fs2004-01761-1.

[3] CMS Collaboration. "Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC". *Physics Letters B*, 716(1):30–61, 2012. URL https://www.sciencedirect.com/science/article/pii/S0370269312008581.

[4] ATLAS Collaboration. "Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC". *Physics Letters B*, 716 (1):1–29, 2012. URL https://www.sciencedirect.com/science/article/pii/S037026931200857X.

[5] B. Heinemann and Y. Nir. "The Higgs program and open questions in particle physics and cosmology". *Physics-Uspekhi*, 62(9):920–930, sep 2019. URL https://doi.org/10.3367%2Fufne.2019.05.038568.

[6] S. Perlmutter, G. Alderingm, G. Goldhaber et al. "Measurements of Ω and Λ from 42 High-Redshift Supernovae". *The Astrophysical Journal*, 517(2):565, jun 1999. URL https://dx.doi.org/10.1086/307221.

[7] A. G. Riess, A. V. Filippenko, P. Challis et al. "Observational Evidence from Supernovae for an Accelerating Universe and a Cosmological Constant". *The Astronomical Journal*, 116(3):1009, sep 1998. URL https://dx.doi.org/10.1086/300499.

[8] N. Aghanim, Y. Akrami, M. Ashdown et al. "Planck 2018 results. VI. Cosmological parameters". *Astronomy & Astrophysics*, 641:A6, sep 2020. URL https://doi.org/10.1051%2F0004-6361%2F201833910.

[9] R. Massey, T. Kitching, and J. Richard. "The dark matter of gravitational lensing". *Reports on Progress in Physics*, 73(8):086901, 2010. URL https://arxiv.org/abs/1001.1739.

[10] S. Weinberg. *Cosmology*. Cosmology. OUP Oxford, 2008. ISBN 9780191523601. URL https://books.google.fr/books?id=nqQZdg020fsC.

[11] David J Griffiths. *Introduction to elementary particles; 2nd rev. version*. Physics textbook. Wiley, New York, NY, 2008.

[12] "The Standard Model of particle physics is brilliant and completely flawed". https://www.abc.net.au/news/science/2017-07-15/the-standard-model-of-particle-physics-explained/7670338, 2017. Accessed on July 24, 2023.

[13] I. G. Kaplan. "Pauli Exclusion Principle and its theoretical foundation", 2019. URL https://doi.org/10.48550/arXiv.1902.00499.

[14] M. J. Herrero. "The Standard Model", 1998. URL https://doi.org/10.48550/arXiv.hep-ph/9812242.

[15] Michael E Peskin. *An introduction to quantum field theory*. CRC press, 2018.

[16] G. Ecker. "Quantum Chromodynamics", 2006. URL https://cds.cern.ch/record/943008.

[17] A. Pich. "The Standard Model of Electroweak Interactions", 2012. URL https://doi.org/10.48550/arXiv.1201.0537.

[18] J. Ellis, M. K. Gaillard, and D. V. Nanopoulos. "A Historical Profile of the Higgs Boson. An Updated Historical Profile of the Higgs Boson", 2016. URL https://cds.cern.ch/record/2012465. Accessed on 28 July 2023.

[19] R.L. Workman et al. (Particle Data Group). "The Review of Particle Physics: Status of Higgs Boson Physics". *Prog. Theor. Exp. Phys. 2022*, 2022(8):083C01, 2022. URL https://pdg.lbl.gov/2019/reviews/rpp2018-rev-passage-particles-matter.pdf.

[20] "Higgs boson". https://grab-group.ethz.ch/research/higgs-boson.html, 2023. Accessed on July 27, 2023.

[21] CMS Collaboration. "Measurement of Higgs boson production in association with a W or Z boson in the H $\rightarrow$ WW decay channel". Technical Report CMS-PAS-HIG-19-017, CERN, Geneva, 2021. URL http://cds.cern.ch/record/2758367.

[22] CMS Collaboration. "Observation of $t\bar{t}H$ production". *Phys. Rev. Lett.*, 120 (CMS-HIG-17-035, CERN-EP-2018-064):231801, 2018. URL https://arxiv.org/abs/1804.02610.

[23] CMS Collaboration. "Observation of the Higgs boson decay to a pair of $\tau$ leptons with the CMS detector". *Phys. Lett. B*, 779(CMS-HIG-16-043, CERN-EP-2017-181): 283–316, 2018. URL https://doi.org/10.1016/j.physletb.2018.02.004.

[24] CMS Collaboration. "Observation of Higgs boson decay to bottom quarks". *Phys. Rev. Lett.*, 121:121801, 2018. URL https://arxiv.org/abs/1808.08242.

[25] N. Berger, C. Bertella, T. P. Calvet et al. "Simplified Template Cross Sections - Stage 1.1", 2019. URL https://arxiv.org/abs/1906.02754.

[26] CMS Collaboration. "A portrait of the Higgs boson by the CMS experiment ten years after the discovery". *Nature*, 607(CMS-HIG-22-001, CERN-EP-2022-039): 60–68, 2022. URL https://www.nature.com/articles/s41586-022-04892-x.

[27] CMS Collaboration. "Measurements of $t\bar{t}H$ Production and the CP Structure of the Yukawa Interaction between the Higgs Boson and Top Quark in the Diphoton Decay Channel". *Phys. Rev. Lett.*, 125(CMS-HIG-19-013, CERN-EP-2020-028): 061801, 2020. URL https://doi.org/10.1103/PhysRevLett.125.061801.

[28] CMS Collaboration. "A measurement of the Higgs boson mass in the diphoton decay channel". *Physics Letters B*, 805:135425, 2020. URL https://www.sciencedirect.com/science/article/pii/S037026932030229X.

[29] CMS Collaboration. "Measurement of the Higgs boson inclusive and differential fiducial production cross sections in the diphoton decay channel with pp collisions at $\sqrt{s} = 13$ TeV". *JHEP*, 07(CMS-HIG-19-016, CERN-EP-2022-142):091, 2023. URL https://link.springer.com/article/10.1007/JHEP07(2023)091.

[30] CMS Collaboration. "Measurements of Higgs boson production cross sections and couplings in the diphoton decay channel at $\sqrt{\mathrm{s}} = 13$ TeV". *JHEP*, 07(CMS-HIG-19-015, CERN-EP-2021-038):027, 2021. URL https://link.springer.com/article/10.1007/JHEP07(2021)027.

[31] CMS Collaboration. "Search for Higgs boson pair production in the $\gamma\gamma b\bar{b}$ final state in pp collisions at $\sqrt{\mathrm{s}} = 13$ TeV". *Physics Letters B*, 788:7–36, 2019. URL https://doi.org/10.1016/j.physletb.2018.10.056.

[32] CMS Collaboration. "Search for exotic decay of the Higgs boson into two light pseudoscalars with four photons in the final state at $\sqrt{s} = 13$ TeV". Technical Report CMS-PAS-HIG-21-003, CERN, Geneva, 2021. URL https://cds.cern.ch/record/2776775.

[33] O. Sim Brüning, P. Collier, P Lebrun et al. *LHC Design report*. CERN Yellow Reports: Monographs. CERN, Geneva, 2004. URL https://cds.cern.ch/record/782076.

[34] CMS Collaboration. CMS Physics: Technical Design Report Volume 1: Detector Performance and Software. Technical Report CERN-LHCC-2006-001, CMS-TDR-8-1, CERN, Geneva, 2006. URL http://cds.cern.ch/record/922757.

[35] L. Evans and P. Bryant. "LHC Machine". *Journal of Instrumentation*, 3(08):S08001, aug 2008. URL https://dx.doi.org/10.1088/1748-0221/3/08/S08001.

[36] HiLumi Collaboration. "The HL-LHC project". Accessed on May 15, 2023, 2023. URL https://hilumilhc.web.cern.ch/.

[37] E. Lopienska. "The CERN accelerator complex, layout in 2022", 2022. URL https://cds.cern.ch/record/2800984. Accessed on May 15, 2023.

[38] G. Soyez. "Pileup mitigation at the LHC: A theorist's view". *Physics Reports*, 803: 1–158, 2019. URL https://www.sciencedirect.com/science/article/pii/S0370157319300456.

[39] CMS Collaboration. "Public luminosity results", 2023. URL https://twiki.cern.ch/twiki/bin/view/CMSPublic/LumiPublicResults#Summary_charts_of_luminosity. Accessed on May 15, 2023.

[40] B. Schmidt. "The High-Luminosity upgrade of the LHC: Physics and Technology Challenges for the Accelerator and the Experiments". *Journal of Physics: Conference Series*, 706(2):022002, apr 2016. URL https://dx.doi.org/10.1088/1742-6596/706/2/022002.

[41] I. Neutelings. "CMS coordinate system", 2021. URL https://tikz.net/axis3d_cms/. Accessed on May 15, 2023.

[42] CMS Collaboration. "Cutaway diagrams of CMS detector", 2019. URL https://cds.cern.ch/record/2665537. Accessed on May 15, 2023.

[43] CMS Tracker Group of the CMS Collaboration. "The CMS Phase-1 pixel detector upgrade". *JINST*, 16(02):P02027, feb 2021. URL https://dx.doi.org/10.1088/1748-0221/16/02/P02027.

[44] CMS Collaboration. The CMS tracker system project: Technical Design Report. Technical Report CERN-LHCC-98-006, CMS-TDR-5, CERN, Geneva, 1997. URL http://cds.cern.ch/record/368412.

[45] CMS Collaboration. "Description and performance of track and primary-vertex reconstruction with the CMS tracker". *Journal of Instrumentation*, 9(10):P10009–P10009, oct 2014. URL https://doi.org/10.1088%2F1748-0221%2F9%2F10%2Fp10009.

[46] J. Mans, J. Anderson, B. Dahmes et al. CMS Technical Design Report for the Phase 1 Upgrade of the Hadron Calorimeter. Technical Report CERN-LHCC-2012-015, CMS-TDR-10, CERN, 2012. URL https://cds.cern.ch/record/1481837.

[47] E. Yazgan for the CMS ECAL/HCAL Collaborations. "The CMS barrel calorimeter response to particle beams from 2 to 350 GeV/c". *Journal of Physics: Conference Series*, 160(1):012056, apr 2009. URL https://dx.doi.org/10.1088/1742-6596/160/1/012056.

[48] CMS Collaboration. "Performance of the CMS muon detector and muon reconstruction with proton-proton collisions at $\sqrt{s} = 13$ TeV". *JINST*, 13(06):P06015, 2018. URL https://doi.org/10.1088/1748-0221/13/06/P06015.

[49] CMS collaboration. "The performance of the CMS muon detector in proton-proton collisions at $\sqrt{s}$= 7 TeV at the LHC". *Journal of Instrumentation*, 8(11):P11002–P11002, nov 2013. URL https://doi.org/10.1088%2F1748-0221%2F8%2F11%2Fp11002.

[50] CMS Collaboration. "The CMS trigger system". *Journal of Instrumentation*, 12 (01):P01020, jan 2017. URL https://dx.doi.org/10.1088/1748-0221/12/01/P01020.

[51] CMS Collaboration. The CMS electromagnetic calorimeter project: Technical Design Report. Technical Report CERN-LHCC-97-033, CMS-TDR-4, CERN, 1997. URL https://cds.cern.ch/record/349375?ln=en.

[52] CMS Collaboration. CMS TriDAS project: Technical Design Report, Volume 1: The Trigger Systems. Technical report, CERN, 2000. URL https://cds.cern.ch/record/706847.

[53] C. W. Fabjan and F. Gianotti. "Calorimetry for Particle Physics". *Rev. Mod. Phys.*, 75(CERN-EP-2003-075):1243–1286, 2003. URL https://cds.cern.ch/record/692252.

[54] R.L. Workman et al. (Particle Data Group). "The Review of Particle Physics: Passage of Particles Through Matter". *Prog. Theor. Exp. Phys. 2022*, 2022(8):083C01, 2022. URL https://pdg.lbl.gov/2019/reviews/rpp2018-rev-passage-particles-matter.pdf.

[55] CMS Collaboration. "Time reconstruction and performance of the CMS electromagnetic calorimeter". *Journal of Instrumentation*, 5(03):T03011, mar 2010. URL https://dx.doi.org/10.1088/1748-0221/5/03/T03011.

[56] CMS Collaboration. "ECAL Performance - Public Results", 2020. URL https://ecaldpgplots.web.cern.ch/#/ecalapproved/CMS-DP-2020_021. Accessed on May 26, 2023.

[57] CMS Collaboration. "ECAL Clustering for run 3", 2022. URL https://cds.cern.ch/record/2812783. Accessed on May 26, 2023.

[58] M. Anfreville et al. "Laser monitoring system for the CMS lead tungstate crystal calorimeter". *Nucl. Instrum. Meth. A*, 594(CERN-CMS-NOTE-2007-028):292–320, 2008. URL https://doi.org/10.1016/j.nima.2008.01.104.

[59] CMS Collaboration. "Particle-flow reconstruction and global event description with the CMS detector". *Journal of Instrumentation*, 12(10):P10003, oct 2017. URL https://dx.doi.org/10.1088/1748-0221/12/10/P10003.

[60] CMS Collaboration. "Electron and photon reconstruction and identification with the CMS experiment at the CERN LHC". *JINST*, 16(CMS-EGM-17-001, CERN-EP-2020-219):P05014, 2021. URL https://iopscience.iop.org/article/10.1088/1748-0221/16/05/P05014.

[61] CMS Collaboration. "Reconstruction of signal amplitudes in the CMS electromagnetic calorimeter in the presence of overlapping proton-proton interactions". *Journal of Instrumentation*, 15(10):P10002, oct 2020. URL https://dx.doi.org/10.1088/1748-0221/15/10/P10002.

[62] CMS Collaboration. "Description and performance of track and primary-vertex reconstruction with the CMS tracker". *Journal of Instrumentation*, 9(10):P10009–P10009, oct 2014. URL https://doi.org/10.1088%2F1748-0221%2F9%2F10%2Fp10009.

[63] A. Fagot, A. Cimmino, S. Crucy et al. "R&D towards the CMS RPC Phase-2 upgrade". *Journal of Instrumentation*, 11(09):C09017, sep 2016. URL https://dx.doi.org/10.1088/1748-0221/11/09/C09017.

[64] CMS Collaboration. The Phase-2 Upgrade of the CMS Tracker. Technical Report CERN-LHCC-2017-009, CMS-TDR-014, CERN, Geneva, 2017. URL https://cds.cern.ch/record/2272264.

[65] CMS Collaboration. The Phase-2 Upgrade of the CMS Barrel Calorimeters. Technical Report CERN-LHCC-2017-011, CMS-TDR-015, CERN, Geneva, 2017. URL https://cds.cern.ch/record/2283187.

[66] CMS Collaboration. The Phase-2 Upgrade of the CMS Endcap Calorimeter. Technical Report CERN-LHCC-2017-023, CMS-TDR-019, CERN, Geneva, 2017. URL https://cds.cern.ch/record/2293646.

[67] A. Martelli. "The CMS HGCAL detector for HL-LHC upgrade", 2017. URL https://arxiv.org/abs/1708.08234.

[68] CMS Collaboration. A MIP Timing Detector for the CMS Phase-2 Upgrade. Technical Report CERN-LHCC-2019-003, CMS-TDR-020, CERN, Geneva, 2019. URL https://cds.cern.ch/record/2667167.

[69] P. Moreira. "LpGBT specification document", 2018. URL https://espace.cern.ch/GBT-Project/LpGBT/Specifications/Forms/AllItems.aspx. Accessed on Jul 15, 2023.

[70] M. Da Rocha Rolo, R. Bugalho, F. Gonçalves et al. "TOFPET ASIC for PET applications". *Journal of Instrumentation*, 8:C02050, 02 2013. URL https://iopscience.iop.org/article/10.1088/1748-0221/8/02/C02050.

[71] F. Vasey, J. Troska, D. Ricci et al. "Versatile link plus project", 2017. URL https://espace.cern.ch/project-Versatile-Link-Plus/SitePages/Home.aspx. Accessed on July 15, 2023.

[72] A. Rose et al. "Serenity: An ATCA prototyping platform for CMS Phase-2". *PoS*, TWEPP2018(CMS-CR-2018-327):115, 2019. URL https://cds.cern.ch/record/2646388.

[73] "PETsys TOF FEB / D board (Readout System)". https://www.petsyselectronics.com/web/public/products/4, 2020. Accessed on July 10, 2023.

[74] "Python documentation". https://www.python.org/, 2023. Accessed on July 10, 2023.

[75] M. Pandey, M. Fernandez, F. Gentile et al. "The transformational role of GPU computing and deep learning in drug discovery". *Nat Mach Intell 4, 211–221*, page 211–221, 2022. URL https://doi.org/10.1038/s42256-022-00463-x.

[76] "Artificial Intelligence Index Report 2021". https://aiindex.stanford.edu/wp-content/uploads/2021/03/2021-AI-Index-Report-_Chapter-1.pdf, 2021. Accessed on April 23, 2023.

[77] CMS Collaboration. "Search for nonresonant Higgs boson pair production in final state with two bottom quarks and two tau leptons in proton-proton collisions at $\sqrt{s} = 13$ TeV", 2022. URL https://cds.cern.ch/record/2803419?ln=en.

[78] M. Stoye. "Deep learning in jet reconstruction at CMS". *J. Phys. Conf. Ser.*, 1085 (CMS-CR-2017-416):042029, 2018. URL https://iopscience.iop.org/article/10.1088/1742-6596/1085/4/042029.

[79] J. Pata, J. Duarte, F. Mokhtar et al. "Machine Learning for Particle Flow Reconstruction at CMS". *J. Phys. Conf. Ser.*, 2438(1):012100, 2023. URL https://doi.org/10.1088/1742-6596/2438/1/012100.

[80] C. Janiesch, P. Zschech, and K. Heinrich. "Machine learning and deep learning". *Electron Markets*, 31:685–695, 2021. URL https://doi.org/10.1007/s12525-021-00475-2.

[81] Y. Bengio I. Goodfellow and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[82] 3 Types of Machine Learning. https://www.newtechdojo.com/3-types-of-machine-learning/, 2020. Accessed on April 30, 2023.

[83] IBM. "What is Unsupervised learning?". https://www.ibm.com/topics/unsupervised-learning, 2023.

[84] M. Wiering and M. Van Otterlo. *Reinforcement learning*, volume 12. Springer, 2012.

[85] A. Kapoor, A. Gulli, S. Pal et al. *Deep Learning with TensorFlow and Keras: Build and deploy supervised, unsupervised, deep, and reinforcement learning models.* Packt Publishing Ltd, 2022.

[86] D. Singh and B. Singh. "Investigating the impact of data normalization on classification performance". *Applied Soft Computing*, 97:105524, 2020. URL https://www.sciencedirect.com/science/article/pii/S1568494619302947.

[87] L. Tsung-Yi, P. Goyal, R. Girshick et al. "Focal Loss for Dense Object Detection", 2018. URL https://doi.org/10.48550/arXiv.1708.02002.

[88] "Scikit-learn documentation: Decision trees". https://scikit-learn.org/stable/modules/tree.html, 2007-2023. Accessed on April 30, 2023.

[89] "Decision tree classification algorithm". https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm, 2011-2021. Accessed on April 30, 2023.

[90] IBM. "What is a Decision Tree?". https://www.ibm.com/topics/decision-trees, 2023. Accessed on April 30, 2023.

[91] H. Almuallim. "An efficient algorithm for optimal pruning of decision trees". *Artificial Intelligence*, 83(2):347–362, 1996.

[92] Y. Coadou. Boosted decision trees. In *Artificial Intelligence for High Energy Physics*, pages 9–58. WORLD SCIENTIFIC, feb 2022. URL https://doi.org/10.1142%2F9789811234033_0002.

[93] Google. "Gradient Boosted Decision Trees". https://developers.google.com/machine-learning/decision-forests/intro-to-gbdt, 2022. Accessed on May 1, 2023.

[94] "An Introduction to Gradient Boosting Decision Trees". https://www.machinelearningplus.com/machine-learning/an-introduction-to-gradient-boosting-decision-trees/, 2023. Accessed on May 1, 2023.

[95] M. A Nielsen. *Neural networks and deep learning*, volume 25. Determination press San Francisco, CA, USA, 2015.

[96] "Feed-forward and feedback networks". https://subscription.packtpub.com/book/data/9781788397872/1/ch01lvl1sec21/feed-forward-and-feedback-networks, 2023. Accessed on May 3, 2023.

[97] J. Yacim and D. Boshoff. "Impact of Artificial Neural Networks Training Algorithms on Accurate Prediction of Property Values". *Journal of Real Estate Research*, 40: 375–418, 11 2018. URL https://doi.org/10.1080/10835547.2018.12091505.

[98] J. Feng, X. He, and Q. Teng et al. "Reconstruction of porous media from extremely limited information using conditional generative adversarial networks". *Physical Review E*, 100, 09 2019. URL https://doi.org/10.1103/PhysRevE.100.033308.

[99] S. Singh S. Dubey and B. Chaudhuri. "Activation Functions in Deep Learning: A Comprehensive Survey and Benchmark", 2022. URL https://doi.org/10.48550/arXiv.2109.14545.

[100] D. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization", 2017. URL https://doi.org/10.48550/arXiv.1412.6980.

[101] Y. Yang You, J. Li, S. Reddi et al. "Large Batch Optimization for Deep Learning: Training BERT in 76 minutes", 2020. URL https://doi.org/10.48550/arXiv.1904.00962.

[102] A. Krizhevsky et al. N. Srivastava, G. Hinton. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". *Journal of Machine Learning Research*, 15(56): 1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

[103] H. Larochelle J. Snoek and R. Adams. "Practical Bayesian Optimization of Machine Learning Algorithms", 2012. URL https://doi.org/10.48550/arXiv.1206.2944.

[104] R. Do et al. R. Yamashita, M. Nishio. "Convolutional neural networks: an overview and application in radiology". *Insights into Imaging*, 9, 06 2018. URL https://doi.org/10.1007/s13244-018-0639-9.

[105] "Basic Introduction to Convolutional Neural Network in Deep Learning". https://www.analyticsvidhya.com/blog/2022/03/basic-introduction-to-convolutional-neural-network-in-deep-learning/, 2022. Accessed on May 10, 2023.

[106] A. Reynolds. "Convolutional Neural Networks (CNNs)". https://anhreynolds.com/blogs/cnn.html, 2019. Accessed on May 20, 2023.

[107] H. Yingge, I. Ali and K. Lee. "Deep Neural Networks on Chip - A Survey". *IEEE*, pages 589–592, 02 2020. URL https://ieeexplore.ieee.org/document/9070243.

[108] S. Hu et al. J. Zhou, G. Cui. "Graph Neural Networks: A Review of Methods and Applications", 2021. URL https://doi.org/10.1016/j.aiopen.2021.01.001.

[109] "Tutorial 7: Graph Neural Networks". https://uvadlc-notebooks.readthedocs.io/en/latest/tutorial_notebooks/tutorial7/GNN_overview.html, 2022. Accessed on May 25, 2023.

[110] P. Battaglia J. Shlomi and J.-R. Vlimant. "Graph neural networks in particle physics". *Machine Learning: Science and Technology*, 2(2):021001, jan 2021. URL https://doi.org/10.1088%2F2632-2153%2Fabbf9a.

[111] Y. Iiyama et al. S. Qasim, J. Kieseler. "Learning representations of irregular particle-detector geometry with distance-weighted graph networks". *The European Physical Journal C*, 79(7), jul 2019. URL https://doi.org/10.1140%2Fepjc%2Fs10052-019-7113-9.

[112] "Exciting Applications of Graph Neural Networks". https://blog.fastforwardlabs.com/2019/10/30/exciting-applications-of-graph-neural-networks.html, 2019. Accessed on May 30, 2023.

[113] T. Kipf and M. Welling. "Semi-Supervised Classification with Graph Convolutional Networks", 2017. URL https://doi.org/10.48550/arXiv.1609.02907.

[114] X. Xin, A. Karatzoglou and I. Arapakis. "Graph Highway Networks", 2020. URL https://arxiv.org/abs/2004.04635.

[115] A. Vaswani, N. Shazeer, N. Parmar et al. "Attention Is All You Need", 2017. URL https://arxiv.org/abs/1706.03762.

[116] C. Biino. "The CMS Electromagnetic Calorimeter: overview, lessons learned during Run 1 and future projections". *J. Phys.: Conf.*, 2015(Ser. 587):012001, 2015. URL https://iopscience.iop.org/article/10.1088/1742-6596/587/1/012001.

[117] D. Valsecchi. *"First evidence of VBS in semileptonic decays with $WVjj \to lvqqjj$ final state and optimization of the CMS electromagnetic calorimeter for Run III"*. PhD thesis, Milan Bicocca U. , 2022. URL https://cds.cern.ch/record/2801848. Accessed on April 15, 2023.

[118] F. Melo. *Receiver Operating Characteristic (ROC) Curve*, pages 1818–1823. Springer New York, New York, NY, 2013. ISBN 978-1-4419-9863-7. URL https://doi.org/10.1007/978-1-4419-9863-7_242.

[119] "GEANT4: A simulation toolkit". https://geant4.web.cern.ch/, 2023. Accessed on March 10, 2023.

[120] "Pandas - Python Data Analysis Library". https://pandas.pydata.org, 2023. Accessed on March 10, 2023.

[121] "Numpy documentation". https://numpy.org, 2023. Accessed on March 10, 2023.

[122] T. Frisson and P. Mine. "H4SIM, a Geant4 simulation program for the CMS ECAL supermodule", 2005. URL http://geant4.in2p3.fr/2005/Workshop/UserSession/P.Mine.pdf. Accessed on March 23, 2023.

[123] Scikit-learn: machine learning in Python. https://scikit-learn.org/stable/index.html, 2023. Accessed on March 10, 2023.

[124] R.L. Workman et al. (Particle Data Group). "The Review of Particle Physics: $\pi_0$". *Prog. Theor. Exp. Phys. 2022*, 2022(083C01), 2022. URL https://pdg.lbl.gov/2018/listings/rpp2018-list-pi-zero.pdf.

[125] T. Dorigo. "Photons And Neutral Pions". https://www.science20.com/tommaso_dorigo/photons_and_neutral_pions-255827, 2021. Accessed on March 13, 2023.

[126] M. Stoye on behalf of the CMS collaboration. "Deep learning in jet reconstruction at CMS". *Journal of Physics: Conference Series*, 1085(4):042029, sep 2018. URL https://dx.doi.org/10.1088/1742-6596/1085/4/042029.

[127] L. Zewen, F. Liu, W. Yang et al. "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects". *IEEE Transactions on Neural Networks and Learning Systems*, 33(12):6999–7019, 2022. URL https://doi.org/10.48550/arXiv.2004.02806.

[128] "Tensorflow documentation". https://www.tensorflow.org/, 2023. Accessed on March 23, 2023.