

BROUILLARD Alizée
Promotion 2021
Année universitaire 2020-2021

Diplôme d'ingénieur Télécom Physique Strasbourg
Spécialité Physique et Modélisation
Master 2 Physique Subatomique et Astroparticules

Mémoire de stage de 3^e année

Trouver le méson B d'origine à l'aide d'un réseau de neurones à graphes à Belle II

Institut Pluridisciplinaire Hubert-Curien
Batiment 22, BP28, 67037 Cedex 2
23 Rue du Loess, 67200 Strasbourg

DUJANY Giulio
03.88.10.62.92
giulio.dujany@iphc.cnrs.fr
Du 1^{er} mars 2021 au 16 juillet 2021

Remerciements

Je remercie l'Institut Pluridisciplinaire Hubert-Curien (IPHC) de Strasbourg de m'avoir permis de réaliser ce stage de fin d'étude. Je remercie notamment l'équipe Belle II de Strasbourg et l'équipe de l'Institut de Technologie de Karlsruhe (KIT) pour leurs précieux conseils tout au long de mon stage, ainsi que pour leur bienveillance.

Je voudrais très particulièrement exprimer ma gratitude envers M. DUJANY Giulio pour m'avoir encadré lors de ce stage de fin d'étude. Je le remercie pour le temps consacré à répondre à mes questions ainsi que pour m'avoir guidé lorsque je rencontrais des difficultés.

Je remercie l'école Télécom Physique Strasbourg ainsi que la Faculté de Physique et Ingénierie pour leurs enseignements, sans lesquels réaliser ce stage de fin d'étude n'aurait pas été possible. Je souhaite plus précisément remercier M. LEROY Yann, mon enseignant référent au sein de l'école.

Résumé

Trouver le méson B d'origine à l'aide d'un réseau de neurones à graphes à Belle II

Pour la recherche de nouvelle physique au-delà du modèle standard, l'expérience Belle II fait des mesures de précisions pour mettre en évidence des déviations entre théorie et données expérimentales. Le collisionneur de particules SuperKEKB situé à Tsukuba au Japon fait entrer en collision un faisceau d'électrons et un faisceau de positrons aux énergies 7 GeV et 4 GeV. L'annihilation d'un électron avec un positron produit une résonance $\Upsilon(4S)$ qui se désintègre en une paire de méson chargés B^+B^- ou neutres $B^0\bar{B}^0$. Ces mésons se désintègrent eux aussi en particules intermédiaires, et ainsi de suite jusqu'à l'obtention de ce que nous appelons les particules de l'état final. L'objectif de ce stage de fin d'étude est d'implémenter un algorithme permettant d'assigner chaque particule de l'état final au méson B duquel elle est issue, à l'aide de la matrice des relations. Nous avons implémenté un réseau de neurones à graphes avec la bibliothèque PyTorch de Python. Nous travaillons sur des graphes entièrement connectés et des arbres de désintégration, qui sont des données structurées. Nous avons démontré l'efficacité de notre algorithme en travaillant sur désintégrations modélisées par la bibliothèque phasespace. Chaque désintégration représente ce que l'on appelle une topologie. Nous avons augmenté progressivement la complexité des données et mis en exergue les facteurs qui influent le plus les performances. Nous avons comparé les performances obtenues avec le réseau de neurones à graphes avec celles obtenues avec la méthode Thrust, qui est une méthode analytique traditionnellement utilisée pour résoudre ce problème. Ce stage de fin d'étude visant l'obtention du diplôme d'ingénieur de Télécom Physique Strasbourg à été réalisé à l'Institut Pluridisciplinaire Hubert-Curien à Strasbourg. Le travail a été effectué entre le 1er mars 2021 et le 16 juillet 2021.

Mots clés : IPHC, Belle II, méson B , Modèle Standard, SuperKEKB, résonance $\Upsilon(4S)$, particules de l'état final, matrice des relations, réseau de neurones à graphes, PyTorch, phasespace, topologies, arbres de désintégration, Thrust, performances.

Abstract

Find the native B meson with graph neural network at Belle II

To find new physics beyond Standard Model, the Belle II experiment performs precision measurements to bring to light deviations between theory and experimental data. SuperKEKB collider, located in Tsukuba in Japan, collides electron and positron beams with the energies 7 GeV and 4 GeV. The annihilation of an electron and a positron produces the $\Upsilon(4S)$ resonance which decays to a pair of charged mesons B^+B^- or neutral mesons $B^0\bar{B}^0$. These mesons decay to intermediate particles, and so on, up to the so-called final state particles. The purpose of this end-of-course internship is to implement an algorithm which assigns each particle to its native B meson, with the matrix of relations. We implemented a graph neural network with the library PyTorch of Python because we work with fully connected graph and disintegration tree, which are structured data. We showed the efficiency of our algorithm working on disintegration modelled by the phasespace library. Each disintegration is a topology. We gradually increased the complexity of the data and highlighted the factors which affect the most the efficiency. We compared the performances obtained with the neural graph network and the Thrust method, which is traditionally used to solve this problem. This end-of-course internship for the Telecom Physique Strasbourg engineering degree took place at the Institut Pluridisciplinaire Hubert-Curien in Strasbourg. This work was conducted between March first 2021 and July 16th 2021.

Key words : IPHC, Belle II, B meson, Standard Model, SuperKEKB, $\Upsilon(4S)$ resonance, final state particles, matrix of relations, graph neural network, PyTorch, phasespace, topologies, disintegration trees, Thrust, performances.

Table des matières

Remerciements	ii
Résumé	iii
Abstract	iv
Liste des figures	viii
Glossaire	1
Abréviations	7
Introduction	9
Chapitre 1 Institut Pluridisciplinaire Hubert-Curien	10
1.1 Présentation de l'institut	10
1.2 Présentation des différents départements	11
1.2.1 Département Ecologie, Physiologie et Ethologie (DEPE)	11
1.2.2 Département Recherches Subatomiques (DRS)	11
1.2.3 Département Sciences Analytiques (DSA)	11
1.2.4 Département Radiobiologie, Hadronthérapie et Imagerie Moléculaire (DRHIM)	12
1.2.5 Institut national de la physique nucléaire et de physique des particules (IN2P3)	12
1.3 Organigramme	13
1.3.1 Organigramme de l'Institut Pluridisciplinaire Hubert-Curien	13
1.4 Présentation de Belle II	14
1.4.1 Collaboration Belle II	14
1.4.2 Organigramme de Belle II à l'IPHC	14
1.4.3 Expérience	14
1.4.4 Détecteur	15

Chapitre 2 Contexte scientifique	16
2.1 Modèle standard de la physique des particules	16
2.2 Apprentissage profond	17
2.2.1 Définitions, intérêt et motivation	17
2.2.2 Réseau de neurones traditionnel	17
2.2.3 Applications des réseaux de neurones	19
2.2.4 Réseau de neurones à graphes	19
Chapitre 3 Cadre du projet	22
3.1 Projet et objectifs de travail	22
3.2 Livrables	23
3.3 Calendrier	24
Chapitre 4 Méthodologie	25
4.1 Réseau de neurones à graphes géométrique	25
4.2 Matrice des quadri-impulsions	25
4.3 Matrice des relations	26
4.4 Génération des données	27
4.5 Optimisation : hyperparamètres et métrique	28
4.6 Outil de comparaison : méthode Thrust	29
4.6.1 Principe de la méthode	29
4.6.2 Implémentation	30
Chapitre 5 Résultats et discussions	32
5.1 Étude des données	32
5.1.1 Grande énergie de transfert	32
5.1.2 Faible énergie de transfert	33
5.2 Première étude : désintégrations génériques	35
5.3 Deuxième étude : désintégrations réelles de la résonance $\Upsilon(4S)$	37
5.3.1 Désintégrations entraînées individuellement et validées individuellement	37
5.3.2 Désintégrations entraînées globalement et validées individuellement	40
5.3.3 Comparaison entre désintégrations entraînées individuellement/globalement et validées individuellement	43
5.3.4 Désintégrations entraînées globalement et validées globalement	44
Conclusion et perspectives	46
Bibliographie	47
Annexe A Pseudo-code du réseau de neurones à graphes	48
Annexe B Performance 3o3	50

Annexe C Performance 4o4	51
Annexe D Performance 5o5	52
Annexe E Performances de validation : exemples	53
Annexe F Performances pour différentes topologies. Entraînement individuel et validation individuelle	56
Annexe G Performances pour différentes topologies. Entraînement global et validation individuelle	57

Liste des figures

1.1	Plan de l'IPHC	10
1.2	Organigramme de l'IPHC	13
1.3	Organigramme de Belle II à l'IPHC	14
1.4	Détecteur Belle II avec ses principaux sous-détecteurs.	15
2.1	Modèle standard de la physique des particules.	16
2.2	Réseau de neurones de type MLP.	18
2.3	Schématisation des attributs.	20
2.4	Schématisation de la mise à jour du réseau de neurones à graphes.	20
3.1	Arbre de désintégration de la résonance $\Upsilon(4S)$	23
4.1	Graphe entièrement connecté entre les particules de l'état final.	25
4.2	Matrice des quadri-impulsions pour six particules de l'état final.	26
4.3	Matrice des relations pour six particules de l'état final.	26
4.4	En haut : arbre de désintégration générique. En bas : tableau répertoriant les différentes masses des particules génériques.	28
4.5	Proportion des particules de l'état final dans une collision $e^+e^- \rightarrow \Upsilon(4S)$	28
4.6	Schématisation de la méthode Thrust. A gauche : axe T bien choisi. A droite : axe T mal choisi.	30
4.7	Représentation tridimensionnelle de la méthode Trust pour une désintégration de type 4o4 pour une énergie de transfert égale à 8542 MeV^2	31
5.1	Distribution des quadri-impulsions des particules de l'état final pour dix désintégrations possédant une grande énergie de transfert (8200 MeV^2).	33
5.2	Schéma représentant l'influence d'une grande énergie de transfert sur une désintégration.	33
5.3	Distribution des quadri-impulsions des particules de l'état final pour dix désintégrations possédant une faible énergie de transfert (5950 MeV^2).	34
5.4	Schéma représentant l'influence d'une faible énergie de transfert sur une désintégration.	34

5.5	Performances du réseau de neurones pour les topologies 3o3 (en haut à gauche), 4o4 (en haut à droite) et 5o5 (en bas au milieu) en fonction de $q^2/\text{Masse}^2(\text{racine})$. Chaque groupe de données possède 5000 évènements pour l'entraînement et 5000 évènements pour la validation.	36
5.6	Influence du nombre de particules de l'état final et du minimum sur la performance de la validation.	38
5.7	Influence de l'écart topologie et du minimum sur la performance de la validation.	39
5.8	Influence du maximum et du minimum sur la performance de la validation.	40
5.9	Influence du nombre de particules de l'état final et du minimum sur la performance de la validation.	41
5.10	Influence de l'écart topologie et du minimum sur la performance de la validation.	42
5.11	Influence du maximum et du minimum sur la performance de la validation.	42
5.12	Performances de validation individuelles obtenues pour toutes les topologies, entraînement global (rond) ou individuel (croix).	43
5.13	Différence des performances de validation individuelles obtenues entre l'entraînement global et l'entraînement individuel, pour chaque topologie.	44
A.1	Structure du réseau de neurones à graphes.	49
E.1	Performance de validation d'un groupe de données de topologie 3o3. FSP : six particules de l'état final.	53
E.2	Performance de validation d'un groupe de données de topologie 4o3. FSP : sept particules de l'état final.	53
E.3	Performance de validation d'un groupe de données de topologie 4o4. FSP : huit particules de l'état final.	54
E.4	Performance de validation d'un groupe de données. FSP : entre quatre et six. 2500 évènements par topologie.	54
E.5	Performance de validation d'un groupe de données. FSP : entre quatre et dix. 2500 évènements par topologie.	55

Glossaire

PARTIE PHYSIQUE

A

Annihilation : désintégration totale, la masse se transformant en énergie.

Antimatière : en physique des particules, l'antimatière est l'ensemble des antiparticules.

Antiparticule : type de particule du modèle standard de masse et spin égaux à ceux de la particule correspondante, mais de nombres quantiques opposés.

Astroparticule : également appelé physique des astroparticules, est un jeune domaine de l'astrophysique qui étudie les particules d'origine cosmique présentes dans l'espace.

B

Baryon : en physique des particules, le baryon est une particule composite formée de trois quarks. Les baryons les plus connus sont le proton et le neutron.

Boson de Higgs : particule élémentaire dont l'existence a été confirmée en 2012. Le boson de Higgs constitue l'un des piliers du modèle standard de la physique des particules.

Boson de jauge : particule subatomique de spin égal à un. La famille des bosons inclut des particules élémentaires : les photons, les gluons, les bosons Z et W (ce sont les quatre bosons de jauge du modèle standard. Les bosons élémentaires sont vecteurs de forces.

D

Désintégration générique : modèle de désintégration représentant des particules ayant des masses en unités arbitraires. Une désintégration générique ne représente pas une véritable désintégration mais permet seulement de la schématiser.

Désintégration rare : désintégration dont la probabilité d'occurrence est faible, inférieure ou égal à 10^{-5} .

Détecteur de Tcherenkov : détecteur de particules. Si la vitesse de la particule dans ce milieu dépasse celle de la lumière dans ce même milieu, alors un rayonnement apparaît ; c'est ce que l'on appelle l'effet Tcherenkov.

E

Electron : particule élémentaire de charge négative.

F

Fermion : en physique des particules, le fermion est une particule de spin demi-entier. Le fermion peut-être une particule élémentaire, comme l'électron, ou une particule composite comme le proton. Les fermions élémentaires regroupent deux familles : les leptons et les quarks.

H

Hadron : particule composite interagissant avec l'interaction forte. Nous distinguons les fermions, auquel cas ils sont appelés baryons, et les bosons, auquel cas ils sont appelés mésons.

I

Information dE/dx : information sur la quantité d'énergie perdue par une particule par ionisation au sein d'un détecteur.

Interactions élémentaires : il existe quatre interactions responsables de tous les phénomènes physiques observés dans l'Univers. Chacune de ces interactions se manifeste par une force dite fondamentale. Les quatre forces fondamentales sont l'interaction nucléaire forte, l'interaction nucléaire faible, l'interaction électromagnétique et l'interaction gravitationnelle.

L

Lepton : en physique des particules, le lepton est une particule élémentaire de spin $\frac{1}{2}$ qui n'est pas sensible à l'interaction forte. La famille des leptons est constituée des électrons, des muons, des tauons, des neutrinos respectifs (neutrino électronique, muonique et tauique) ainsi que des antiparticules de toutes les précédentes.

Luminosité : paramètre utilisé pour les collisionneurs de particules. La luminosité est proportionnelle au nombre de collisions à l'intérieur du détecteur. Elle s'exprime en unité de surface et de temps.

M

Matière noire : catégorie de matière hypothétique qui rend compte de certaines observations astrophysiques.

Méson : en physique des particules, le méson désigne une particule composite composée de deux particules élémentaires : un quark et un antiquark.

Méson B : méson composé d'un antiquark bottom et d'un quark up B^+ , down B^0 , strange B_s ou charm B_c .

Modèle Standard : théorie qui concerne l'interaction électromagnétique, l'interaction nucléaire faible et forte, ainsi que la classification de toutes les particules subatomiques connues.

N

Nouvelle physique au-delà du modèle standard : recherche d'autres modèles pour compléter, étendre le modèle standard afin de répondre à des problématiques telles que l'asymétrie entre matière et antimatière ou la matière noire. On peut citer comme exemple la théorie de la supersymétrie.

P

Particule élémentaire : en physique des particules, une particule élémentaire est une particule insécable, qui n'est pas divisible en d'autres particules.

Particule composite : particule non élémentaire, regroupement de particules élémentaires.

Physique du tau : physique qui consiste à étudier le lepton tau.

Physique nucléaire : branche de la physique qui étudie le noyau atomique et la façon dont il interagit lorsqu'une particule arrive à sa proximité.

Physique Subatomique : branche de la physique qui étudie les constituants élémentaires de la matière et les rayonnements, ainsi que leurs interactions. On l'appelle également physique des particules ou physique des hautes énergies car de nombreuses particules sont instables, n'existent pas à l'état naturel et ne sont détectées que lors de collisions à hautes énergie dans les accélérateurs de particules.

Positon : antiparticule de l'électron, de charge positive. Parfois appelé « positron » (terme anglais).

Q

Quadri-impulsion : pour une particule, la quadri-impulsion ou le quadri-moment est un vecteur réunissant l'énergie E , et les trois composantes de l'impulsion : P_x selon x , P_y selon y et P_z selon z .

Quark : en physique des particules, le quark est une particule élémentaire. Les quarks s'associent entre eux pour former les hadrons. Les quarks s'attirent entre eux par l'interaction forte. Il existe six quarks : up, down, charm, strange, top et bottom. Chaque quark possède une charge électrique et une charge de couleur.

R

Résonance : particule très instable qui se désintègre quasiment instantanément.

T

Tau : particule élémentaire de la famille des leptons.

Théorie de la relativité générale : théorie relativiste de la gravitation. Elle décrit l'influence de la présence de matière et plus généralement d'énergie, sur le mouvement des astres en tenant en compte des principes de la relativité restreinte.

V

Vertex : point symbolisant le lieu où se déroule une interaction entre particules.

PARTIE APPRENTISSAGE AUTOMATIQUE

A

Apprentissage profond : branche de l'intelligence artificielle. L'apprentissage automatique permet aux algorithmes d'apprendre à partir de données, sans intervention humaine, grâce à des approches mathématiques et statistiques.

Apprentissage profond : type d'apprentissage automatique. Très souvent, l'algorithme prend la forme d'un réseau de neurones. Le terme profond signifie que l'algorithme d'apprentissage possède un grand nombre de couches cachées, contrairement à un algorithme traditionnel.

Apprentissage supervisé : type d'apprentissage automatique qui apprend grâce à des étiquettes préalablement attribuées aux données d'apprentissage.

B

Biais : vecteur ajouté aux données dans chaque neurone. Les biais permettent de construire les combinaisons linéaires entre les données d'entrées. Ils sont initialisés à des valeurs aléatoires, qui seront progressivement ajustées pour augmenter les performances.

« **Bruit** » dans les données : lorsque l'on considère un groupe de données, nous pouvons distinguer deux types de données : les données pertinentes pour résoudre un problème et celle non pertinentes. Ces données non pertinentes sont appelées bruit car elles nuisent à la performance d'un algorithme.

C

Classification : classement des données en fonction des étiquettes. La classification considère des étiquettes discrètes.

Couche cachée : couche située entre la couche d'entrée et la couche de sortie. Chaque couche cachée reçoit en entrée les données situées en sortie de la couche précédente.

Couche d'entrée : première couche du réseau de neurones. Cette couche reçoit les informations brutes.

Couche de sortie : dernière couche du réseau de neurones. C'est cette couche qui nous donne le résultat.

D

Descente de gradient : algorithme itératif ayant pour but de minimiser ou maximiser une fonction.

E

Entraînement : étape de l'apprentissage automatique durant laquelle l'algorithme apprend des données, en ajustant les biais et les poids synaptiques.

F

Fonction d'activation : fonction mathématique non linéaire.

Fonction de perte : fonction qui traduit la différence entre les prédictions et les valeurs réelles.

G

Graphes : ensemble de nœuds interconnectés par des liens.

H

Hyperparamètres du réseau de neurones : paramètres externes au processus d'entraînement. Il définit les propriétés de l'entraînement, tel que le nombre d'époques ou le taux d'apprentissage. Il reste statique durant l'entraînement.

I

Intelligence artificielle : caractérisation des algorithmes qui permettent de simuler l'intelligence humaine. Le terme « intelligence » fait référence au fait que l'algorithme doit apprendre par lui-même, avoir un certain comportement, sans intervention humaine, tandis que le terme « artificielle » fait référence à l'usage d'ordinateurs.

N

Neurone (artificiel) : fonction de transfert composée d'une combinaison linéaire des données d'entrées et d'une fonction d'activation prenant en argument cette combinaison linéaire.

Nombre d'époques : nombre d'itérations du réseau de neurones. Le nombre d'époques correspond au nombre de fois que le réseau de neurones prend en compte chaque donnée.

P

Paramètres du réseau de neurones : paramètres internes au réseau de neurones. Il va évoluer durant l'ensemble du processus d'entraînement durant la rétropropagation.

Perceptron multicouche : type de réseau de neurones organisé en plusieurs couches. L'information y circule de la couche d'entrée vers la couche de sortie uniquement. C'est un réseau de neurones à propagation directe.

Performance d'entraînement : pourcentage de prédictions parfaitement en accord avec les valeurs réelles lors de la phase d'entraînement.

Performance de validation : pourcentage de prédictions parfaitement en accord avec les valeurs réelles lors de la phase de validation.

Poids synaptique : matrice utilisée pour calculer le produit scalaire avec le vecteur des entrées. Les coefficients de la matrice sont initialisés à des valeurs aléatoires, qui seront progressivement ajustées pour augmenter les performances.

Prédiction : dans un réseau de neurones, le résultat en sortie est appelé la prédiction.

Propagation directe : transmission des données de la couche d'entrée vers la couche de sortie.

R

Reconstruction de jet : reconstruction de la chaîne de désintégration (position des vertex notamment) de quarks.

Régression : classement des données en fonction des étiquettes. La régression considère des étiquettes continues.

Réseau de neurones à graphes : réseau de neurones prenant comme données d'entrée des graphes. Ce type de réseau de neurones est particulièrement adapté pour les données structurées.

Réseau de neurones convolutifs : réseau de neurones reposant principalement sur la convolution. Ce type de réseau de neurones est particulièrement adapté à la classification d'images.

Réseau de neurones récurrents : réseau de neurones qui possède des connexions récurrentes. Il est constitué d'unités qui se répètent. Ce type de réseau de neurones est particulièrement

adapté à la reconnaissance automatique de la parole, de l'écriture ou plus généralement à la reconnaissance de formes.

Rétropropagation : méthode pour calculer le gradient de l'erreur pour chaque neurone d'un réseau de neurones, de la couche de sortie vers la couche d'entrée. La technique consiste à corriger les erreurs selon l'importance de la contribution de chaque élément aux erreurs.

T

Taux d'apprentissage : hyperparamètre du réseau de neurone, il permet de moduler la correction apportée lors de la descente de gradient.

Topologie : structure d'une désintégration.

V

Validation : étape de l'apprentissage automatique durant laquelle l'algorithme compare ses prédictions aux valeurs réelles.

Abréviations

IPHC : Institut Pluridisciplinaire Hubert-Curien

KIT : Institut de Technologie de Karlsruhe

DEPE : Département d'Écologie, Physiologie et Éthologie

AVEC : Adaptation des VERTébrés marins aux Changements climatiques

EPE : Éthologie et Physiologie Évolutive

PAGRAS : Adaptations Physiologique à la Gravité et Santé

ADAGE : ADapation des Animaux et Gestions Environnementales

DRS : Département de Recherches Subatomiques

DSA : Département de Sciences Analytiques

DRHIM : Département de radiobiologie, Hadronthérapie et Imagerie Moléculaire

CNRS : Centre National de Recherche Scientifique

ALICE : A Large Ion Collider Experiment , expérience sur un grand collisionneur d'ions

CMS : Compact Muon Solenoid, solénoïde compact à muons

OGMA : Ondes Gravitationnelles et Messagers pour l'Astronomie

PICSEL : Physics with Integrated Cmos Sensors and ELelectron, Physique avec un détecteur cmos intégré et électron

DeSIs : Dosimétrie, Simulation, Intrumentation

LSMBO : Spectrométrie de masse Bio-organique

SynPA : Synthèse Pour l'Analyse

RePSeM : Reconnaissance et procédés de séparation moléculaire

CAMBAP : Chimie analytique des molécules bioactives et Pharmagnosie

IN2P3 : Institut nation de la physique nucléaire et de physique des particules

IJCLab : Irene Joliot-Curie Laboratoire des deux infinis

CPPM : Centre de Physique des Particules de Marseille

PXD : Silicon Pixel Detector, détecteur pixelisé de silicium

SVD : Silicon Vertex Detector, détecteur de vertex en silicium

CDC : Central Drift Chamber, chambre de dérive centrale

TOP : Time-Of-Propagation

A-RICH : Aerogel Ring Imaging CHerenkov

ECL : Electromagnetic Calorimeter, calorimeter électromagnétique

KLM : KL-Muon Détecteur : détecteur de muons

MLP : Multilayer Perceptron, perceptron multicouche

graFEI : graph Full Event Interpretation, Interprétation en événements entiers à l'aide de graphes.

FSP : Final State Particles, particules de l'état final

Introduction

Le modèle standard de la physique des particules est la théorie la plus complète qui, à ce jour, explique les interactions qui existent entre matière et forces. Cependant, il existe encore des questions que ce modèle laisse sans réponse, comme l'asymétrie qui existe entre matière et antimatière. Pour tenter de trouver des réponses, l'expérience Belle II fait des mesures de précision pour mettre en évidence des déviations entre théorie et données expérimentales dans le cas de désintégrations rares. Pour cela, le collisionneur de particules SuperKEKB fait entrer en collision des électrons et des positrons.

Ce stage se concentre plus particulièrement sur la résonance $\Upsilon(4S)$ formée par l'annihilation d'un électron et d'un positron aux énergies 7 GeV et 4 GeV. Cette résonance se désintègre en une paire de mésons $B^0\bar{B}^0$ ou B^+B^- , qui eux-mêmes se désintègrent en d'autres particules intermédiaires. Les dernières particules issues de ces désintégrations sont appelées particules de l'état final. L'objectif principal de ce stage est d'implémenter un algorithme qui assignera une particule de l'état final au méson B duquel elle est issue. Pour cela, nous utiliserons l'apprentissage profond, et plus précisément un réseau de neurones à graphes.

Ce rapport est divisé en plusieurs parties. La présentation de l'IPHC où s'est déroulé ce stage se trouve dans la partie 1. La présentation du contexte scientifique ainsi que quelques bases élémentaires sur l'apprentissage profond et les réseaux de neurones à graphes se trouvent dans le chapitre 2. La présentation du projet se trouve dans la partie 3. La méthodologie employée est détaillée dans le chapitre 4. Enfin, les résultats et leurs discussions sont exposés dans le chapitre 5. Nous avons comparé les résultats obtenus avec le réseau de neurones à graphes avec une méthode traditionnellement utilisée dans ce domaine appelée méthode Thrust.

Ce stage s'est tenu dans le respect des règles en vigueur en raison de la crise sanitaire. La période de stage s'est déroulée principalement en télétravail, avec quelques jours sur le site de l'Institut Pluridisciplinaire Hubert-Curien (IPHC) à Strasbourg. La majorité des réunions se sont déroulées en visioconférence.

Institut Pluridisciplinaire Hubert-Curien

1.1 Présentation de l'institut

L'Institut Pluridisciplinaire Hubert-Curien (IPHC) est une unité mixte de recherche du Centre National de la Recherche Scientifique (CNRS) et de l'Université de Strasbourg créé le 1er janvier 2006. Durant mon stage, le directeur de l'IPHC est R. Barillon [1].

L'objectif était de créer un institut à partir de différentes compétences scientifiques pour développer des programmes pluridisciplinaires de très haut niveau. En 2006, l'IPHC fut composé de trois départements :

- le Département d'Ecologie, Physiologie et Ethologie (DEPE)
- le Département de Recherches Subatomiques (DRS)
- le Département des Sciences Analytiques (DSA)

En 2016, le Département Radiobiologie, Hadronthérapie et Imagerie Moléculaire (DRHIM) fut créé.

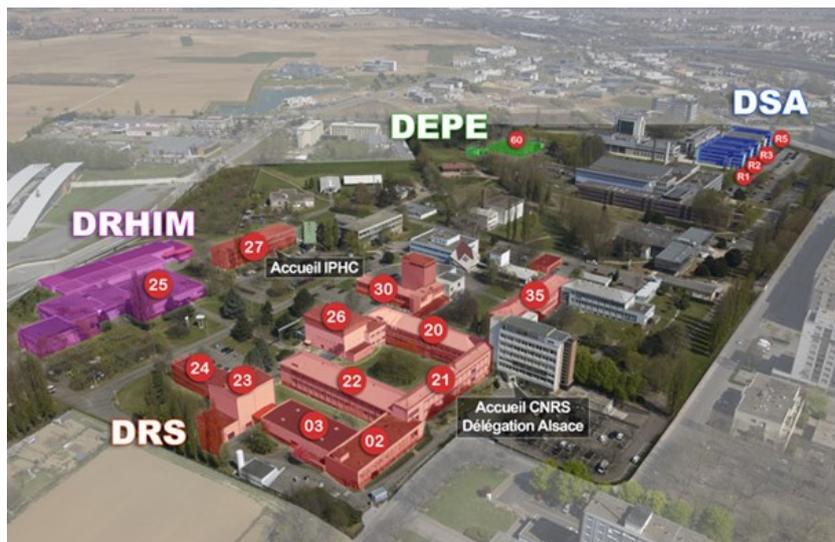


FIGURE 1.1 – Plan de l'IPHC

L'IPHC regroupe près de 160 ingénieurs, techniciens et personnels administratifs. Les quatre départements de l'IPHC sont situés dans dix-sept bâtiments sur le Campus de Cronenbourg et une équipe du DSA est située à la Faculté de pharmacie à Illkirch.

1.2 Présentation des différents départements

1.2.1 Département Ecologie, Physiologie et Ethologie (DEPE)

L'objectif principal de recherche du groupe DEPE est de comprendre les stratégies mises en place au cours de l'évolution pour permettre aux espèces animales de faire face aux contraintes et aux modifications actuelles de leur environnement naturel. C. Hibold est le responsable de ce département. Le DEPE possède trois groupes de travail :

- AVEC : adaptation des vertébrés marins aux changements environnementaux
- EPE : éthologie et physiologie évolutive
- PAGRAS : adaptations physiologiques à la gravité et santé
- ADAGE : adaptation des animaux et gestion environnementales

Ce département possède également des supports techniques, par exemple la métrologie ou la spectroscopie de masse.

1.2.2 Département Recherches Subatomiques (DRS)

L'objectif principal de ce groupe est la recherche fondamentale en physique nucléaire et en physique des particules. S. Courtin est le responsable de ce département. Parmi les axes de recherche du DRS, nous trouvons :

- Théorie
- Du noyau aux étoiles
- Du Big Bang aux particules (ALICE, Belle II, CMS, Neutrino, OGMA, PICSEL)
- Énergie, environnement et dosimétrie (données nucléaires pour les réacteurs, radiochimie, DeSIs)

Ce département possède des équipes techniques, par exemple l'équipe d'instrumentation des accélérateurs.

1.2.3 Département Sciences Analytiques (DSA)

Ce département a pour objectif l'élaboration de nouvelles architectures complexes permettant la reconnaissance ionique, le transport et le transfert, ainsi que la séparation et l'analyse de mélanges complexes de macromolécules biologiques. E. Marchioni est le responsable de ce département. Ce département est partagé en quatre axes :

- LSMBO : Spectrométrie de masse bio-organique
- SynPA : Synthèse pour l'analyse
- RePSeM : Reconnaissance et procédés de séparation moléculaire

-
- CAMBAP : Chimie analytique des molécule bioactives et pharmacognosie. Ce département possède des équipes techniques.

1.2.4 Département Radiobiologie, Hadronthérapie et Imagerie Moléculaire (DRHIM)

Ce département est né autour du projet Cyrce. Le Cyclotron Cyrce est un accélérateur de particules de 24 MeV qui permet de produire des radio-isotopes pour le diagnostic, le suivi médicamenteux ou le traitement médical. P. Laquerriere est le responsable de ce département.

Les axes de recherche sont :

- Radiobiologie
- Hadronthérapie
- Imagerie Moléculaires

Ce département possède des équipes techniques.

1.2.5 Institut national de la physique nucléaire et de physique des particules (IN2P3)

Le DRS travaille en collaboration avec l'Institut national de la physique nucléaire et de physique des particules [2] (IN2P3). L'IN2P3 est l'une des tutelles de l'IPHC, avec l'Université de Strasbourg. Cet institut coordonne depuis 50 ans à l'échelle nationale des recherches de l'infiniment grand avec l'étude de la cosmologie et des astroparticules, à l'infiniment petit avec la physique nucléaire et la physique des particules élémentaires. R. Pain est le directeur de l'IN2P3. Dans le cadre de notre stage, nous utilisons une des plateformes de IN2P3, à savoir le Centre de Calcul (abrégé « CC ») de l'IN2P3 de Lyon. Le CC-IN2P3 conçoit et opère un ensemble de services, en particulier un système de stockage de masse et de moyens de trainement de grandes masses de données.

1.3 Organigramme

1.3.1 Organigramme de l'Institut Pluridisciplinaire Hubert-Curien

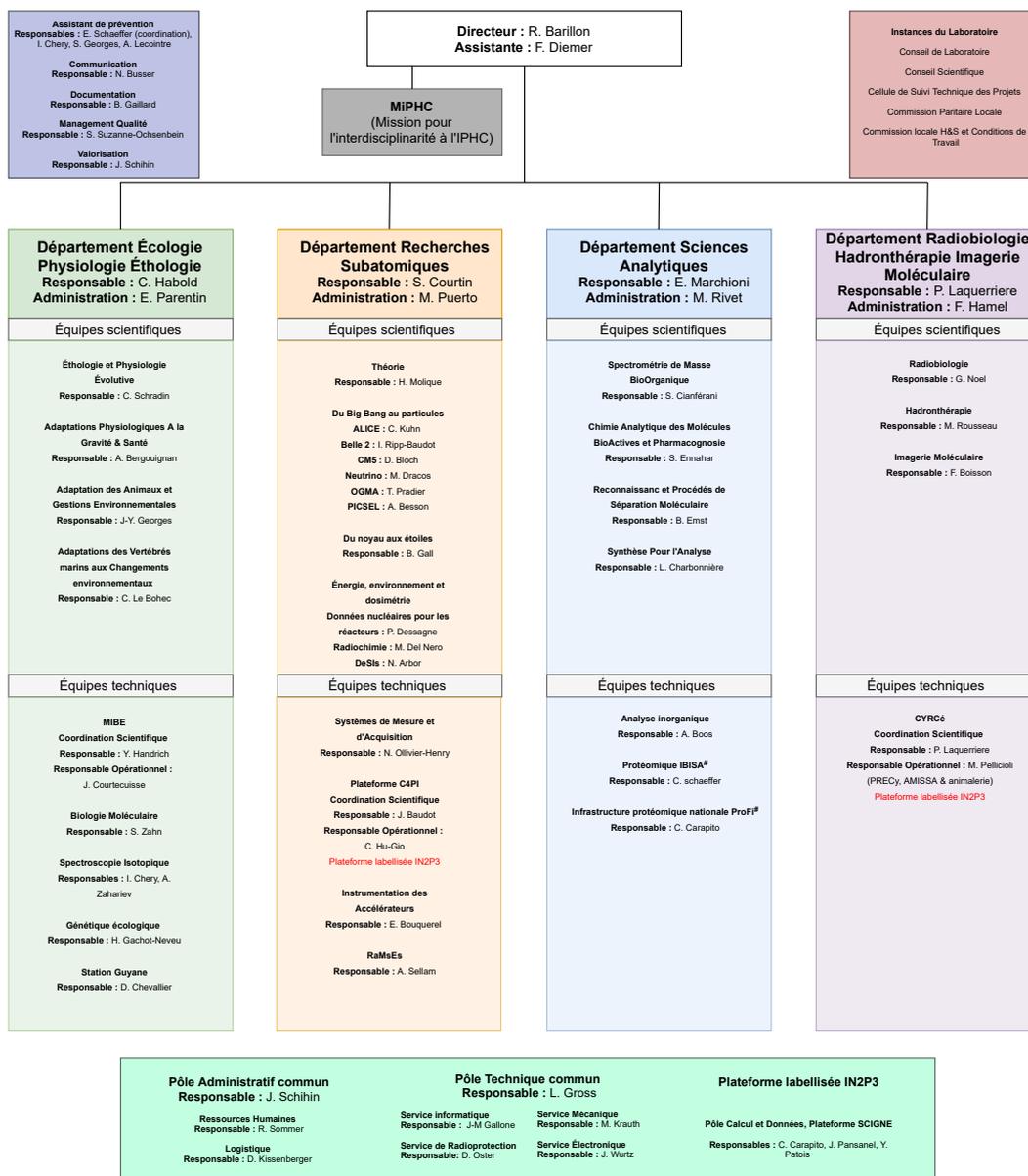


FIGURE 1.2 – Organigramme de l'IPHC

1.4 Présentation de Belle II

1.4.1 Collaboration Belle II

Le 11 octobre 2017 [3], la France est entrée dans la collaboration Belle II qui compte actuellement vingt-trois autres pays. L'expérience Belle II est une expérience de physique des particules qui enregistre depuis 2018 les collisions entre électrons et positons au sein du collisionneur SuperKEKB au Japon.

A ce jour, le groupe français de Belle II compte une vingtaine de scientifiques du Laboratoire des deux infinis Irene Joliot-Curie (IJCLab) à Orsay, du Centre de Physique des Particules de Marseille (CPPM) et de l'IPHC à Strasbourg. Les chercheurs des trois laboratoires sont impliqués dans l'analyse des collisions réalisée au SuperKEKB. Plus particulièrement, le groupe de l'IPHC est impliqué dans la reconstruction des traces des particules chargées.

1.4.2 Organigramme de Belle II à l'IPHC

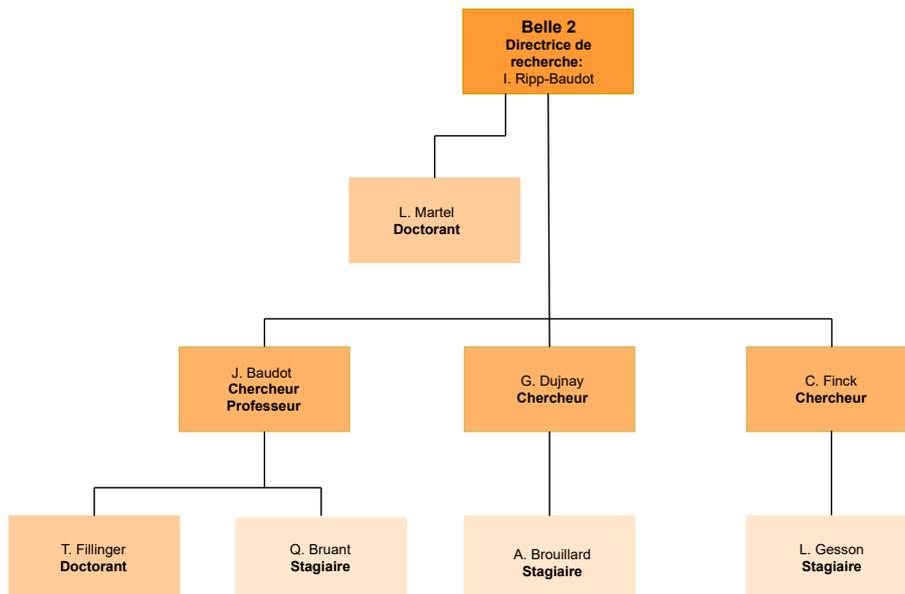


FIGURE 1.3 – Organigramme de Belle II à l'IPHC

1.4.3 Expérience

L'expérience Belle II [4] succède à l'expérience Belle [5]. Parmi ses objectifs tels que la physique du τ ou la recherche de matière noire, la collaboration Belle II étudie la désintégration du méson B , notamment les désintégrations rares. L'expérience Belle II se déroule au SuperKEKB [6], collisionneur de particules SuperKEKB situé à Tsukuba au Japon. SuperKEKB fait entrer en collision un faisceau d'électrons à une énergie de 7 GeV et un faisceau de positons à une énergie de 4 GeV. L'annihilation d'un électron et d'un positon pour ces énergies produit une

résonance $\Upsilon(4S)$, qui est un méson composé d'un couple quark/anti-quark bottom/anti-bottom $b\bar{b}$. Dans au moins 96% des cas, cette résonance se désintègre en une paire de mésons $B^0\bar{B}^0$ ou B^+B^- . Chacun de ces mésons se désintègre à son tour en particules élémentaires (pion π , kaon K , muon μ , proton p , électron e , neutrino ν , etc). L'ensemble de ces particules sont nommées particules de l'état final. Ce sont ces particules de l'état final qui sont détectées par le détecteur de Belle II, à l'exception des neutrinos.

1.4.4 Détecteur

Le détecteur Belle II est un spectromètre qui collecte efficacement les données des collisions entre électrons et positons effectuées par l'accélérateur SuperKEKB. Le détecteur Belle II se compose de plusieurs sous-détecteurs [7] :

- un détecteur de vertex composé de deux couches de capteurs en silicium pixélisé (PXD) et de quatre couches de capteurs à bande de silicium double face (SVD). Ce détecteur permet de reconstruire la position des vertex de désintégration des mésons B
- une chambre de dérive centrale (CDC) qui mesure les trajectoires, les impulsions et les informations dE / dx des particules chargées
- deux détecteurs de Tchérenkov : un compteur de temps de propagation (TOP) et un autre compteur à imagerie annulaire avec aérogel radiateur (A-RICH), qui permettent l'identification de particules
- Un calorimètre électromagnétique (ECL), qui permet la séparation des électrons et des pions. Il mesure également l'énergie des photons et des pions
- un flux-retour de fer situé à l'extérieur de la bobine qui est instrumenté pour détecter les mésons K_L^0 et identifier les muons (KLM)

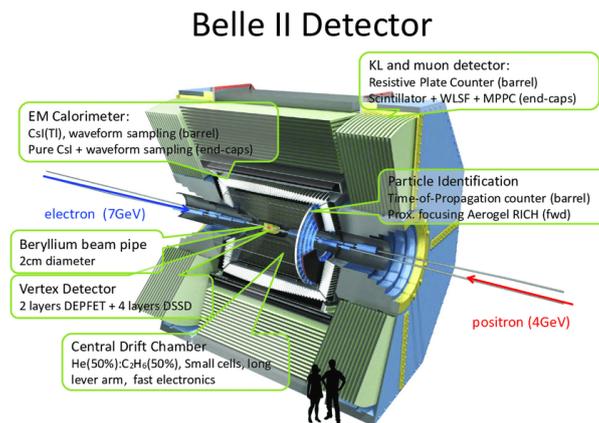


FIGURE 1.4 – Détecteur Belle II avec ses principaux sous-détecteurs.

En juin 2020, SuperKEKB, le nouveau collisionneur d'électrons-positons, a atteint un nouveau record du monde de luminosité instantanée, à savoir $2,4 \times 10^{34} \text{ cm}^{-2}\text{s}^{-1}$ [6].

2

Contexte scientifique

2.1 Modèle standard de la physique des particules

Le modèle standard de la physique des particules s'est imposé depuis les années 1970 comme étant la théorie majeure décrivant la matière, composée de particules élémentaires (fermions, bosons de jauge, Higgs), interagissant avec les trois forces fondamentales : l'électromagnétisme, la force nucléaire forte et la force nucléaire faible [8].

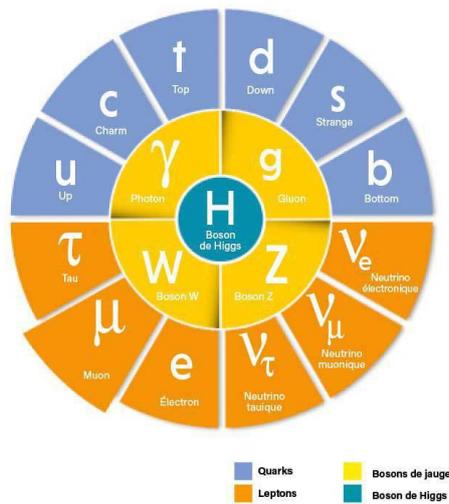


FIGURE 2.1 – Modèle standard de la physique des particules.

Malgré des résultats expérimentaux solides, cette théorie soulève encore aujourd'hui certains problèmes. Parmi eux, nous pouvons citer notamment la quatrième force fondamentale, la gravitation, décrite par la théorie de la relativité générale et incompatible avec le modèle standard. Un des enjeux majeurs du *XXI^e* siècle est de réussir à trouver une nouvelle théorie englobant le modèle standard de la physique des particules et la théorie de la relativité générale.

De plus, le modèle standard n'explique ni l'asymétrie qui existe dans l'univers entre la matière

et antimatière, ni l'existence de la matière noire. C'est pour cela qu'aujourd'hui nous recherchons une nouvelle physique, au-delà du modèle standard. Nous faisons plus particulièrement des mesures de précision pour trouver des déviations entre théorie et données expérimentales dans les désintégrations rares.

2.2 Apprentissage profond

2.2.1 Définitions, intérêt et motivation

L'apprentissage automatique est une branche de l'intelligence artificielle [9]. En se basant sur une approche mathématique et statistique, l'apprentissage automatique permet aux algorithmes d'apprendre seuls à partir de données. Généralement, l'apprentissage automatique se décompose en deux phases.

La première est une phase d'apprentissage au cours de laquelle l'algorithme va estimer un modèle à partir de données, par exemple reconnaître la présence ou non d'un chat sur une photographie. Si les données sont préalablement étiquetées, alors l'apprentissage est supervisé. Nous parlons de classification si ces étiquettes sont discrètes et de régression si ces étiquettes sont continues.

Cependant, nous ne possédons pas toujours les étiquettes associées aux données. Nous pouvons donc demander à l'algorithme de trouver une structure interne aux données et d'attribuer lui-même des étiquettes : c'est l'apprentissage non supervisé.

La deuxième phase de l'apprentissage automatique est la phase de validation où nous présentons un nouveau groupe de données au modèle et où ce dernier nous donne ses résultats, que nous appelons prédictions.

L'apprentissage profond, ou apprentissage neuronal profond, dérive de l'apprentissage automatique. Il prend le plus souvent la forme d'un réseau de neurones. Le terme profond fait référence au nombre de couches cachées dans le réseau de neurones, qui est beaucoup plus grand que dans les réseaux de neurones traditionnels. Cela lui permet notamment de s'améliorer de façon autonome.

Ainsi, l'apprentissage profond et notamment les réseaux de neurones profonds s'avèrent très efficaces dans la physique des particules, notamment pour la classification de particules comme nous allons le faire au cours de ce projet.

2.2.2 Réseau de neurones traditionnel

Principe

La conception d'un réseau de neurones artificiels repose sur un mimétisme de réseaux de neurones biologiques, comme le cerveau humain où nous pouvons identifier un neurone artificiel à un neurone biologique et le lien entre deux neurones artificiels à une synapse. Dans le réseau de neurones artificiels, les neurones sont organisés en différentes couches successives ; ce type de

réseau est un perceptron multicouche (MLP). Nous pouvons distinguer la couche d'entrée, la couche de sortie et les couches intermédiaires également appelées couches cachées. Tout comme dans le cerveau humain, les neurones sont interconnectés.

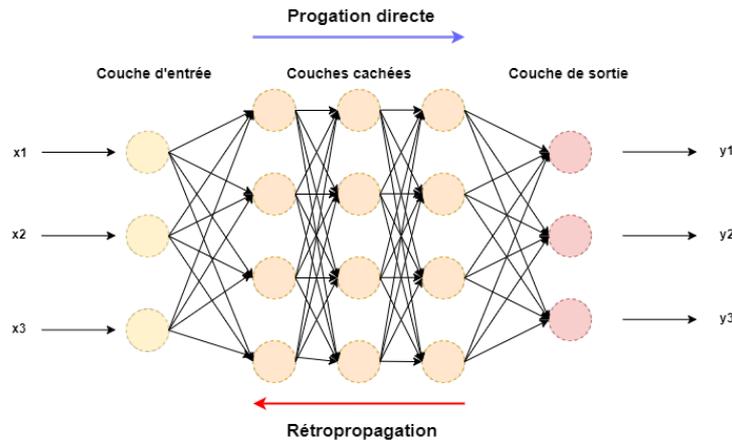


FIGURE 2.2 – Réseau de neurones de type MLP.

Fonctionnement

La couche d'entrée reçoit un groupe de données. Puis chaque neurone calcule une combinaison linéaire des entrées, c'est-à-dire qu'elle renvoie le produit scalaire entre le vecteur des entrées « x » et la matrice des poids synaptiques « W » en y ajoutant un vecteur de biais « b ».

$$x \rightarrow Wx + b \quad (2.1)$$

Au lancement du réseau de neurones, les coefficients des poids synaptiques et des biais sont initialisés à des valeurs aléatoires. Puis, après chaque couche, le réseau de neurones fait passer l'information dans une fonction d'activation qui est une fonction mathématique non linéaire.

$$Wx + b \rightarrow g(Wx + b) \quad (2.2)$$

Cette fonction d'activation est l'équivalent biologique du potentiel d'activation : si le seuil est atteint, alors le neurone concerné propagera l'information, sinon il s'éteindra. La fonction d'activation joue donc un rôle d'interrupteur pour le réseau de neurones. La tangente hyperbolique ou sigmoïde sont des exemples typiques de fonction d'activation. Chaque couche intermédiaire prend en entrée la sortie de la couche précédente, réalise la combinaison linéaire puis applique la fonction d'activation, et ce jusqu'à la couche de sortie qui nous donne la prédiction de notre modèle. C'est ce que nous appelons la propagation. Dans le cadre de l'apprentissage supervisé, à la fin de chaque prédiction, nous comparons la valeur obtenue à la valeur réelle. Nous calculons alors ce que nous appelons la fonction de perte L .

$$L = \Phi(y_{pred} - y_{rel}) \quad (2.3)$$

Puis nous ajustons les coefficients des poids synaptiques et les biais. Pour cela, nous minimisons l'erreur avec un algorithme de descente de gradient. Nous effectuons la dérivée de la fonction de perte par rapport aux poids synaptiques et aux biais.

$$W_{i+1} = W_i - \alpha \frac{\partial L}{\partial W_i} \quad (2.4)$$

α est le taux d'apprentissage. Cela s'appelle la rétropropagation. Le taux d'apprentissage indique la vitesse à laquelle les coefficients évoluent. La procédure propagation/rétropropagation se réalise un certain nombre de fois avant de s'arrêter. Le nombre d'itérations est appelé nombre d'époques. En général, le processus se déroule sur deux types de données :

- les données d'entraînement, qui sont utilisées en premier pour mettre à jour les poids et les biais, au cours de l'étape que l'on appelle "entraînement"
- les données de validation, où nous comparons les prédictions aux valeurs réelles, au cours de l'étape que l'on appelle "validation"

2.2.3 Applications des réseaux de neurones

En fonction des contraintes intrinsèques au problème que nous voulons résoudre et des données utilisées, nous devons choisir un réseau de neurones adapté [10]. Il existe beaucoup de réseaux de neurones différents. Parmi eux, les réseaux de neurones convolutifs sont particulièrement adaptés dans la reconnaissance d'image du fait de l'invariance du traitement par translation ; pour un problème de reconnaissance vocale, un réseau de neurones récurrents est plus adapté car il prend en compte l'invariance temporelle, c'est-à-dire que l'ordre des mots dans une phrase n'influence pas leur reconnaissance. Enfin, les réseaux de neurones à graphes sont utilisés pour des problèmes où nous souhaitons avoir une invariance par permutation. C'est notamment le cas lorsque les données sont structurées, par exemple des graphes. Pour la suite de ce projet, nous allons travailler avec un réseau de neurones à graphes.

2.2.4 Réseau de neurones à graphes

Principe

Comme dit précédemment, les réseaux de neurones à graphes sont des réseaux de neurones conçus pour recevoir des données structurées et dont le but est de déduire les relations qui existent entre les objets. Les réseaux de neurones à graphes ont été développés à la suite de nouveaux besoins de performance. Très vite, les réseaux de neurones à graphes sont devenus très populaires dans de nombreux domaines tels que la classification et la prédiction de liens. Ils sont par exemple utilisés par les algorithmes des réseaux sociaux pour mieux comprendre les relations entre les personnes ou en chimie pour déterminer la composition d'une molécule.

Dans le domaine de la physique des particules, les réseaux de neurones à graphes sont utilisés par exemple en calorimétrie pour la reconstruction de jet [11].

Fonctionnement

Le fonctionnement du réseau de neurones à graphe est différent du réseau de neurones simple. Nous définissons un graphe G par un doublet (V, E) [12] :

- V représente l'ensemble des nœuds (ou "Vertices" en anglais) du graphe. Chaque nœud v_i possède un certain nombre de caractéristiques qui lui sont propre, par exemple la quadri-impulsion d'une particule
- E représente l'ensemble des liens (ou "Edges" en anglais) du graphe. Chaque lien e_{ij} , reliant le nœud v_i au nœud v_j , possède également un certain nombre de caractéristiques, par exemple un entier représentant la relation qui existent entre les deux neurones reliés par ce lien

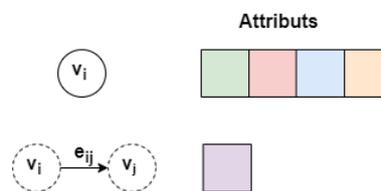


FIGURE 2.3 – Schématisation des attributs.

Le calcul s'effectue ensuite en plusieurs étapes :

- mise à jour des attributs des liens
- agrégation des attributs des liens sur chaque nœud
- mise à jour des attributs des nœuds

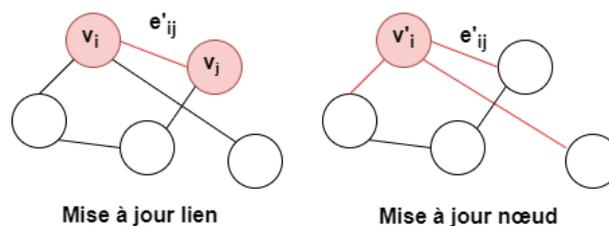


FIGURE 2.4 – Schématisation de la mise à jour du réseau de neurones à graphes.

La mise à jour des attributs des liens se fait via un réseau de neurones qui prend en entrée les attributs du lien, ainsi que les attributs des nœuds connectés par le lien.

Le réseau de neurones à graphes réalise ensuite l'agrégation des attributs des liens sur chaque nœud. Par exemple, le réseau de neurones à graphes réalise une moyenne des attributs des liens arrivant sur un nœud.

La mise à jour des attributs des nœuds se fait également via un réseau de neurones qui prend

en entrée les attributs des noeuds et l'agrégation des attributs des liens qui arrivent sur lui.

Les réseaux de neurones cités précédemment sont des MLP.

3

Cadre du projet

3.1 Projet et objectifs de travail

Dans le cadre de ce stage au sein de Belle II, nous nous intéressons plus particulièrement à la désintégration de la résonance $\Upsilon(4S)$ en deux mésons B chargés ou neutres :

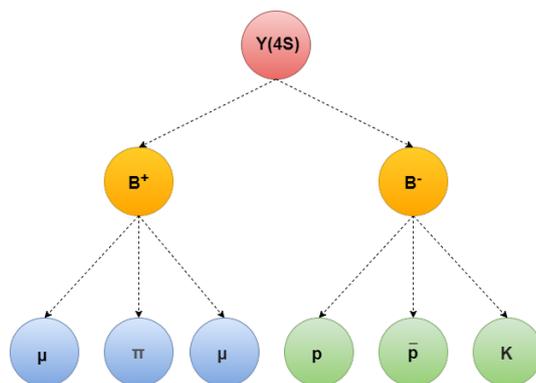
$$\Upsilon(4S) \rightarrow B^+ B^- \quad (3.1)$$

$$\Upsilon(4S) \rightarrow B^0 \bar{B}^0 \quad (3.2)$$

Chaque méson va ensuite se désintégrer en particules intermédiaires, qui vont également se désintégrer jusqu'à l'obtention des particules dites de l'état final. Les particules de l'état final les plus courantes sont les pions, les kaons, les muons, les électrons, les protons, les photons et les neutrinos. Le nombre de particules de l'état final varie, avec une moyenne autour de trois pour chaque méson B .

L'objectif principal de ce stage est de créer un algorithme qui, à partir des quadri-impulsions des particules de l'état final, attribue chaque particule de l'état final au méson B duquel elle est issue. Nous pouvons appeler cet algorithme un séparateur d'origine. C'est un problème de classification : soit la particule de l'état final appartient au premier méson B , soit elle appartient au deuxième.

Pour attribuer chaque particule de l'état final au méson B duquel elle est issue, nous utilisons un algorithme d'apprentissage profond. La motivation principale de l'utilisation de l'intelligence artificielle pour ce problème est que nous travaillons sur un nombre très important de données (de plusieurs milliers à quelques millions) et que les autres types d'algorithmes sont moins efficaces. Nous allons prendre comme exemple comparatif la méthode Thrust. Dans le cadre de ce projet, nous nous concentrons sur un nombre total de particules de l'état final compris entre quatre et dix.

FIGURE 3.1 – Arbre de désintégration de la résonance $\Upsilon(4S)$.

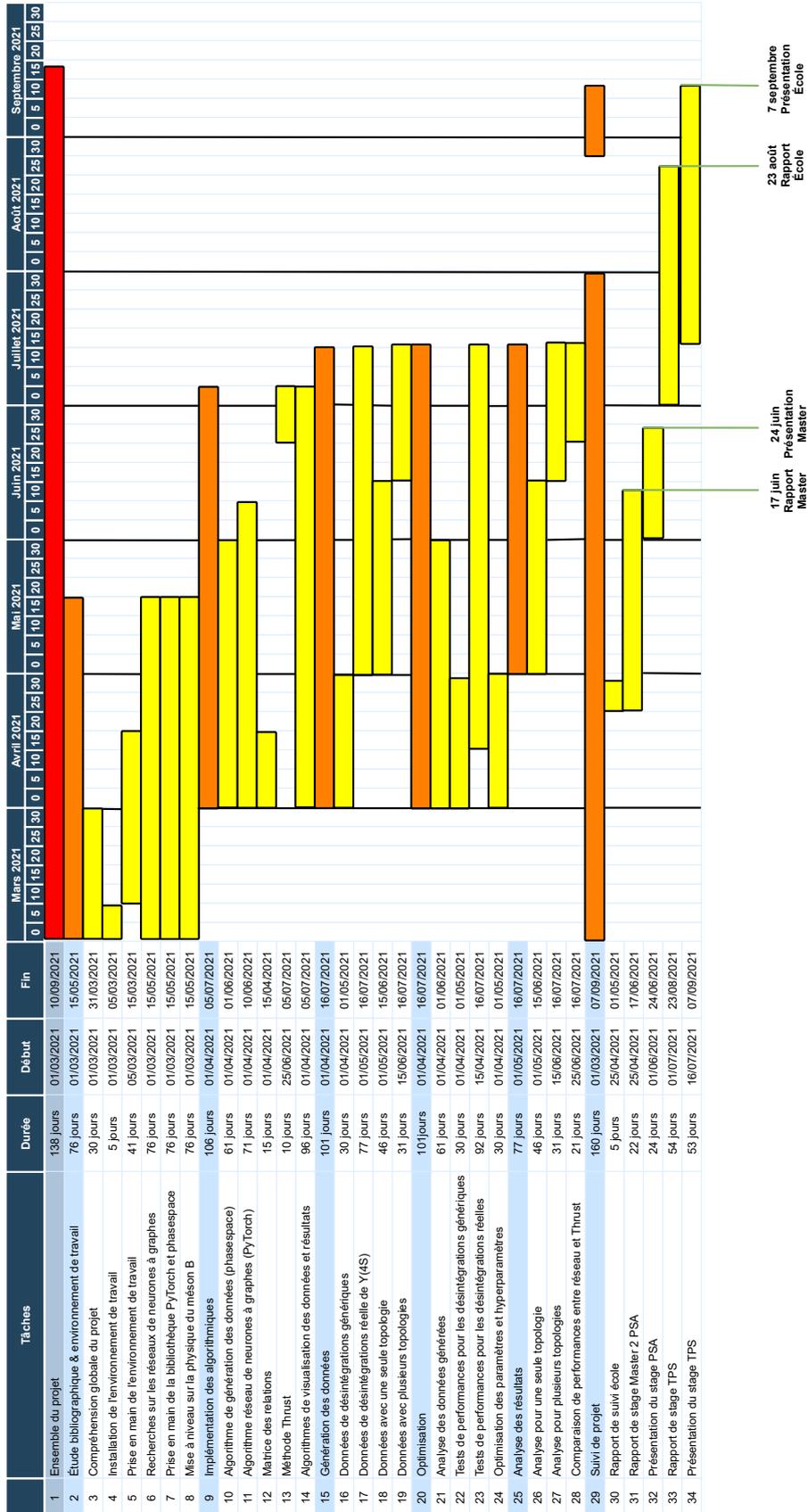
3.2 Livrables

Les livrables sont l'ensemble des scripts relatifs au réseau de neurones à graphes, l'implémentation de la méthode Thrust, les données générées au cours du projet, ainsi que les différents résultats obtenus.

Les livrables sont disponibles dans le répertoire GitHub du projet graFEI, ainsi que sur l'espace informatique dédié au projet Belle II.

Enfin, les différentes présentations réalisées tout au long du projet et les rapports sont disponibles sur l'indico de Belle II.

3.3 Calendrier



4

Méthodologie

4.1 Réseau de neurones à graphes géométrique

En ce qui concerne l'architecture de l'algorithme, nous utilisons un réseau de neurones à graphes.

Nous avons implémenté notre réseau grâce la bibliothèque PyTorch [13], et plus précisément le module PyTorch Geometric [14]. Les couches utilisées ont été construites à partir de la classe « MetaLayer » qui permet de faire les mises à jour successives des couches de nœuds et de liens.

Ce type de réseau à la particularité de prendre pour données d'entrée des graphes, c'est-à-dire des données structurées, c'est pour cela que nous avons choisi cette architecture. Dans le cadre de ce projet, nous avons utilisé un graphe entièrement connecté, qui nous donne les relations existantes entre les différentes particules de l'état final. C'est ce graphe que nous donnons en entrée du réseau de neurones. La Figure 4.1 donne un exemple de ce type de graphes :

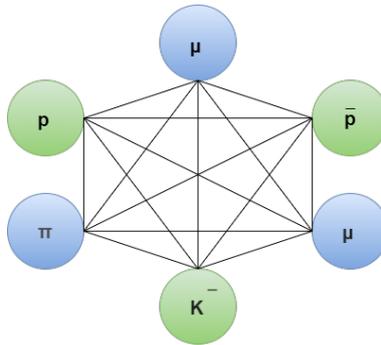


FIGURE 4.1 – Graphe entièrement connecté entre les particules de l'état final.

4.2 Matrice des quadri-impulsions

Nous définissons ici la matrice des quadri-impulsions. Cette matrice a un nombre de lignes égal au nombre de particules de l'état final et un nombre de colonnes égal à quatre pour chaque

composante de la quadri-impulsion : P_x la composante selon x , P_y la composante selon y , P_z la composante selon z et E l'énergie. La structure de la matrice des quadri-impulsions pour six particules de l'état final est montrée dans la Figure 4.2.

	P_x	P_y	P_z	E
μ	a_0	a_1	a_2	a_3
π	b_0	b_1	b_2	b_3
$\bar{\mu}$	c_0	c_1	c_2	c_3
p	d_0	d_1	d_2	d_3
\bar{p}	e_0	e_1	e_2	e_3
K	f_0	f_1	f_2	f_3

FIGURE 4.2 – Matrice des quadri-impulsions pour six particules de l'état final.

Notre algorithme est de type géométrique. Son principal intérêt est qu'il est invariant pour l'ordre des particules de l'état final mises en entrée, et que le nombre de particules de l'état final peut varier.

4.3 Matrice des relations

Dans le cadre de ce stage, nous étudions une méthode d'apprentissage. Cette méthode consiste à entraîner le réseau de neurones pour qu'il prédise ce que nous appelons la matrice des relations.

La matrice des relations est une matrice bidimensionnelle indiquant les relations qui existent entre les particules de l'état final. Le nombre de lignes est égal au nombre de particules de l'état final, et le nombre de colonnes est également égal au nombre de particules de l'état final. Dans le cas d'une désintégration à six particules de l'état final, nous obtenons la matrice de la Figure 4.3 :

	μ	π	$\bar{\mu}$	p	\bar{p}	K
μ	0	1	1	2	2	2
π	1	0	1	2	2	2
$\bar{\mu}$	1	1	0	2	2	2
p	2	2	2	0	1	1
\bar{p}	2	2	2	1	0	1
K	2	2	2	1	1	0

FIGURE 4.3 – Matrice des relations pour six particules de l'état final.

Considérons deux particules p_1 et p_2 :

- si p_1 et p_2 sont identiques, alors le coefficient 0 est placé aux intersections de ces deux particules dans la matrice des relations

- si p_1 et p_2 sont issues du même méson B , alors le coefficient 1 est placé aux intersections de ces deux particules dans la matrice des relations
- si p_1 et p_2 ne sont pas issues du même méson B , alors le coefficient 2 est placé aux intersections de ces deux particules dans la matrice des relations

Par conséquent, la matrice des relations nous permet de savoir si deux particules sont issues du même méson B ou non.

En ce qui concerne le réseau de neurones à graphes, chaque nœud possèdera quatre caractéristiques, les quatre composantes de la quadri-impulsion. Les liens, quant à eux, posséderont une seule caractéristique : la relation qui existe entre ces deux nœuds, c'est-à-dire 1 ou 2.

4.4 Génération des données

Afin de contrôler l'apprentissage de notre algorithme, nous allons tout d'abord l'entraîner sur des données dites génériques, puis nous ferons les entraînements sur les données des désintégrations réelles. Nous créons les premiers ensembles de données génériques grâce à la bibliothèque *phasespace* [15]. Cette bibliothèque nous permet entre autres de choisir la masse des différentes particules ainsi que la structure de l'arbre de désintégration. Ensuite, celle-ci génère un ensemble de désintégrations à n corps en munissant chaque particule de sa quadri-impulsion. Nous pouvons augmenter progressivement la complexité des données que nous générons. Nous faisons cela pour avoir un meilleur suivi de l'entraînement et d'optimiser au mieux les paramètres.

Le premier entraînement a été réalisé avec le cas le plus simple suivant : une particule Z se désintégrant en deux particules X et Y . La particule X se désintègre à son tour en trois particules : a , b et c . De même, la particule Y se désintègre en trois particules : d , e et f . Les différentes masses, en unités arbitraires, sont de 100 pour la particule Z , 30 pour les particules X et Y , et 5 pour toutes les particules de l'état final a, b, c, d, e et f . Les quadri-impulsions sont générées de telle sorte à ce qu'elles respectent les règles de cinématique pour les désintégrations à n corps, c'est-à-dire la conservation de la quadri-impulsion. Ce cas générique est résumé dans la Figure 4.4.

Nous appelons ce type de désintégration « 3o3 » ; le chiffre 3 représente ici le nombre de particules de l'état final originaires de la même particule. Par exemple, dans le cas où une particule X posséderait quatre particules de l'état final et la particule Y en posséderait trois, alors cette désintégration s'appellerait « 4o3 ». Nous réaliserons les entraînements sur de multiples configurations de désintégrations, notamment 3o3, 4o4 et 5o5.

Puis nous entraînerons notre réseau sur des données possédant les vraies particules finales, dans les proportions réelles indiquées dans la Figure 4.5 [16].

Lorsque les entraînements réalisés sur ces données génériques seront suffisamment satisfaisants, nous entraînerons le réseau de neurones avec des ensembles de données Monte-Carlo de l'expérience Belle II.

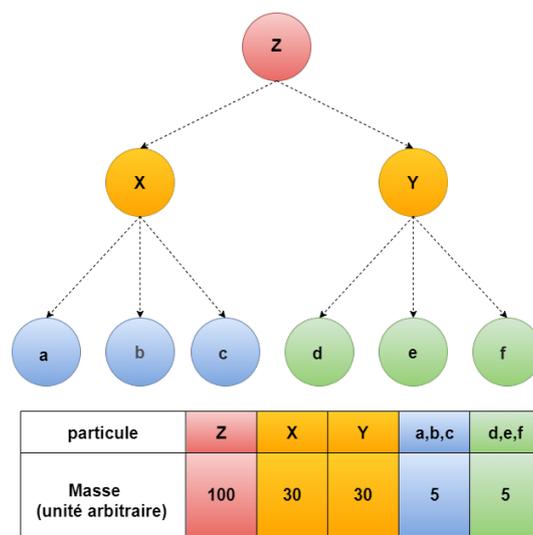


FIGURE 4.4 – En haut : arbre de désintégration générique. En bas : tableau répertoriant les différentes masses des particules génériques.

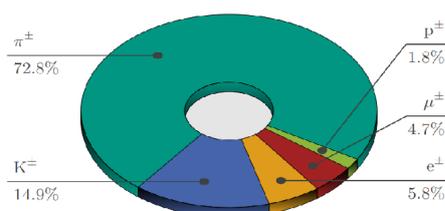


FIGURE 4.5 – Proportion des particules de l'état final dans une collision $e^+e^- \rightarrow \Upsilon(4S)$.

4.5 Optimisation : hyperparamètres et métrique

Nous avons utilisé l'entropie croisée [17] de la bibliothèque PyTorch comme fonction de perte. L'entropie croisée est une mesure de la différence entre la distribution de la variable prédite et la distribution réelle de cette variable. Nous l'avons choisi car elle permet de réaliser une classification. L'optimiseur utilisé est Adam [18], également de la bibliothèque PyTorch. C'est une méthode de descente de gradient stochastique. Le taux d'apprentissage est de 0.001, le nombre d'époques est usuellement de 100 et les données sont traitées par l'algorithme par paquets de 20 pour les masses génériques, puis 60 pour les masses réelles. En ce qui concerne la métrique, c'est-à-dire la manière dont nous avons traité les résultats, nous avons déterminé la proportion de matrices des relations correctement prédites. Nous les avons ensuite séparées en plusieurs catégories :

- les matrices des relations parfaitement prédites
- les matrices des relations prédites correctement de 90 à 99%
- les matrices des relations prédites correctement de 80 à 90%

- les matrices des relations prédites correctement de 70 à 80%
- les matrices des relations prédites correctement de 50 à 70%
- les matrices des relations prédites correctement de 0 à 50%

4.6 Outil de comparaison : méthode Thrust

L'objectif de notre projet est d'attribuer chaque particule de l'état final à la particule de laquelle elle est issue, à l'aide d'un réseau de neurones à graphes. Cependant, nous pouvons remettre en question la nécessité de l'usage de l'intelligence artificielle. En effet, selon la nature des données, et des motifs que nous pouvons en déduire, il est tout à fait possible qu'un algorithme moins complexe puisse être utilisé pour classifier les particules de l'état final. Nous avons choisi de faire la comparaison des performances obtenues avec le réseau de neurones à graphes avec les performances obtenues avec la méthode Thrust car cette méthode a été utilisée préalablement pour l'expérience Belle [5].

4.6.1 Principe de la méthode

Chaque particule de l'état final possède quatre composantes dans sa quadri-impulsion : la composante selon x , selon y , selon z et l'énergie E :

$$\vec{p} = (p_x, p_y, p_z, E) \quad (4.1)$$

Ces quatre composantes nous donnent le vecteur quadri-impulsion de la particule.

Dans la méthode Thrust, nous projetons le vecteur de la quadri-impulsion de chaque particule de l'état final sur un axe T . Nous choisissons l'axe T comme étant l'axe selon lequel la somme des valeurs absolues des projections des quadri-impulsions est maximale :

$$T = \text{Max} \left(\sum_{i=0}^{i=N-1} |\vec{T} \cdot \vec{p}_i| \right) \quad (4.2)$$

La figure 4.6 à droite représente l'axe T en rouge qui maximise les valeurs absolues des projections des quadri-impulsions des particules de l'état final. Les flèches noires représentent les particules de l'état final issues d'une même particule, et les flèches vertes représentent les particules de l'état final issues de l'autre particule. La figure de gauche représente la même chose, mais avec un choix d'axe T différent, qui ne maximise pas les valeurs absolues des projections des quadri-impulsions.

L'attribution des particules de l'état final aux particules desquelles elles sont issues se fait alors par calcul du produit scalaire de la quadri-impulsion de la particule avec l'axe T choisi :

- si le produit scalaire est positif, alors la particule de l'état final est issue de la première particule
- si le produit scalaire est négatif, alors la particule de l'état final est issue de la deuxième particule

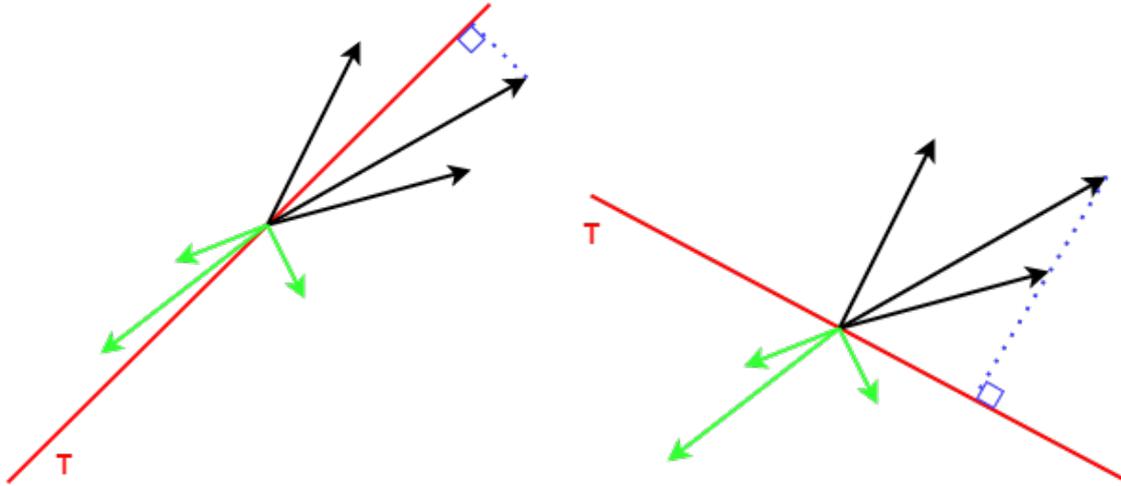


FIGURE 4.6 – Schématisation de la méthode Thrust. A gauche : axe T bien choisi. A droite : axe T mal choisi.

4.6.2 Implémentation

Nous définissons l'axe T selon trois composantes, les composantes selon x , y et z , et deux degrés de liberté θ et ϕ :

$$T_x = \sin(\theta)\cos(\phi) \quad (4.3)$$

$$T_y = \sin(\theta)\sin(\phi) \quad (4.4)$$

$$T_z = \cos(\theta) \quad (4.5)$$

Nous reprenons l'équation 4.2. En la développant, nous obtenons l'équation 4.6

$$T = \text{Max} \left(\sum_{i=0}^{i=N-1} |p_{ix}\sin(\theta)\cos(\phi) + p_{iy}\sin(\theta)\sin(\phi) + p_{iz}\cos(\theta)| \right) \quad (4.6)$$

où p_{ix} , p_{iy} et p_{iz} sont les composantes du vecteur quadri-impulsion de la particule i et N est le nombre de particules de l'état final de la désintégration considérée.

Pour trouver l'axe T selon lequel les valeurs absolues des projections des quadri-impulsions des particules de l'état final sont maximales, nous devons trouver l'angle θ et l'angle ϕ pour lesquels l'équation 4.6 est maximale.

Pour cela, nous allons faire varier l'angle θ entre 0 et π , et l'angle ϕ entre 0 et 2π . Pour maximiser cette fonction, nous utilisons la bibliothèque `scipy.optimize` de Python et nous utilisons la fonction `minimize`. La fonction `minimize` donne pour chaque désintégration l'angle θ et l'angle ϕ qui maximisent l'équation 4.6 (cette équation est donc multipliée par un facteur -1 avant la minimisation pour obtenir la maximisation). A partir de ces deux angles, nous reconstruisons pour chaque désintégration l'axe T à partir de ses trois composantes, puis les vecteurs quadri-impulsions des particules appartenant à cette désintégration. Enfin, nous calculons pour chaque particule le produit scalaire suivant l'équation 4.7.

$$\vec{T} \cdot \vec{p}_i \quad (4.7)$$

Comme dit précédemment, si le produit scalaire est positif, la particule de l'état final appartient à la première particule et si le produit scalaire est négatif, alors la particule de l'état final appartient à la deuxième particule.

La Figure 4.7 donne un exemple de résultat pour une désintégration de type 4o4 dans l'espace tridimensionnel.

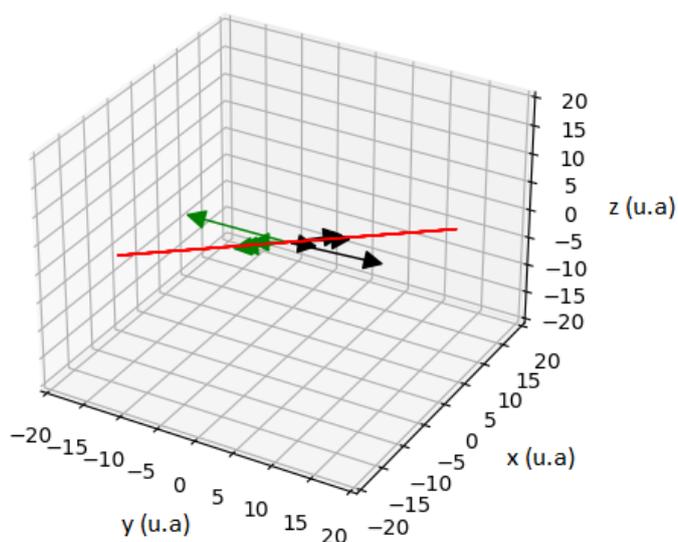


FIGURE 4.7 – Représentation tridimensionnelle de la méthode Thrust pour une désintégration de type 4o4 pour une énergie de transfert égale à 8542 MeV².

5

Résultats et discussions

5.1 Étude des données

Selon la nature des données, nous obtenons des performances très différentes pour le réseau de neurones à graphes, ainsi que pour la méthode Thrust. Dans cette sous-partie, nous mettons en exergue un paramètre qui influence très majoritairement les performances : l'énergie de transfert q^2 . Nous définissons l'énergie de transfert comme la différence entre la masse au carré de la particule racine moins la masse au carré des deux particules intermédiaires :

$$q^2 = \text{masse}(\text{particuleRacine})^2 - \text{masse}(\text{droite})^2 - \text{masse}(\text{gauche})^2 \quad (5.1)$$

Dans cette sous-partie 5.1, toutes les désintégrations considérées possèdent une topologie de type 3o3.

5.1.1 Grande énergie de transfert

La Figure 5.1 représente les quatre composantes de la quadri-impulsion des particules de l'état final, pour un échantillon de dix désintégrations. En abscisse, nous avons représenté chaque désintégration, et chaque point représente la quadri-impulsion d'une des particules de l'état final, pour une certaine composante et pour cette désintégration. La couleur du point indique si la particule de l'état final provient du premier méson B ou du deuxième méson B .

Dans le cas précis de la Figure 5.1, les désintégrations possèdent une grande énergie de transfert, supérieure à 7000 MeV^2 . En analysant les composantes de leurs quadri-impulsions, nous pouvons distinguer assez fréquemment quelle particule de l'état final provient de quel méson B . Par exemple, étudions la désintégration numéro quatre. Il est possible d'attribuer chaque particule de l'état final en fonction du signe de la composante selon x de la quadri-impulsion : si celle-ci est positive, alors la particule provient du premier méson B et si elle est négative, alors elle provient du deuxième méson. Ce phénomène s'explique par le fait que les mésons B possèdent une énergie suffisante pour se propulser loin l'un de l'autre si la désintégration possède une grande énergie de transfert. Nous avons illustré ceci sur la Figure 5.2.

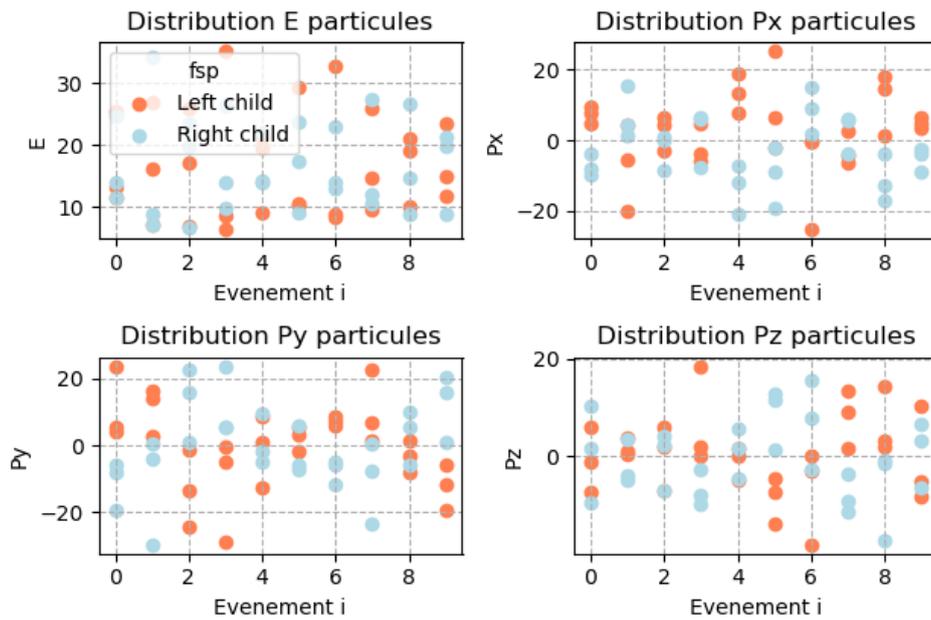


FIGURE 5.1 – Distribution des quadri-impulsions des particules de l'état final pour dix désintégrations possédant une grande énergie de transfert (8200 MeV^2).

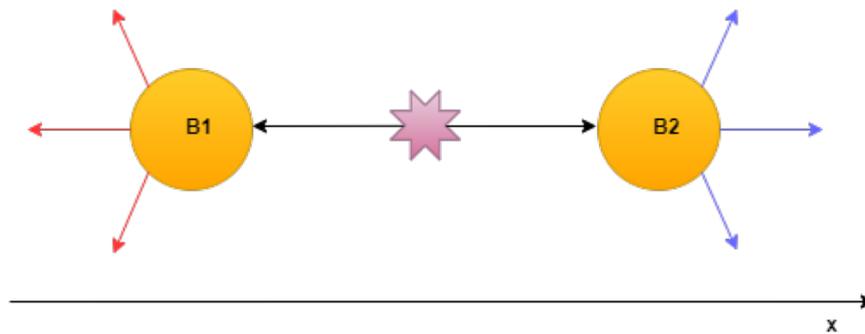


FIGURE 5.2 – Schéma représentant l'influence d'une grande énergie de transfert sur une désintégration.

5.1.2 Faible énergie de transfert

Lorsque l'énergie de transfert de la désintégration est faible, inférieure à 7000 MeV^2 , alors les particules de l'état final issues de la désintégration de $\Upsilon(4S)$ possèdent des quadri-impulsions très semblables.

Comme nous le voyons sur la Figure 5.3, il ne ressort pas de motif particulier. Effectivement, si la désintégration possède une énergie de transfert faible, alors les deux mésons générés ne sont pas propulsés loin l'un de l'autre, et les quadri-impulsions des particules de l'état final des deux

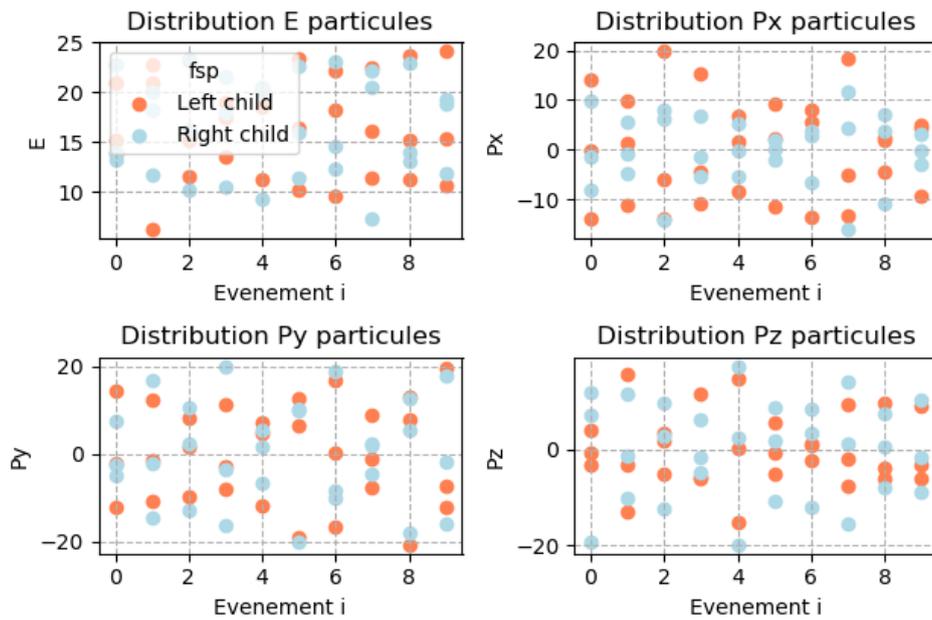


FIGURE 5.3 – Distribution des quadri-impulsions des particules de l'état final pour dix désintégrations possédant une faible énergie de transfert (5950 MeV^2).

mésons B se confondent. Nous avons schématiser cette situation sur la Figure 5.4.

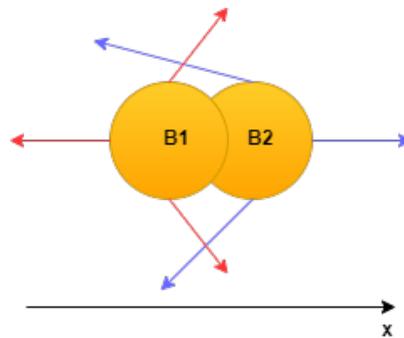


FIGURE 5.4 – Schéma représentant l'influence d'une faible énergie de transfert sur une désintégration.

Ainsi, il n'est plus possible de distinguer de quel méson B vient chaque particule de l'état final à partir d'une simple lecture des composantes des quadri-impulsions.

5.2 Première étude : désintégrations génériques

Nous avons réalisé les premiers entraînements avec des désintégrations génériques générées par la bibliothèque phasepace. Ces premiers tests ont permis de prendre en main le réseau de neurones à graphes et d'optimiser grossièrement les hyperparamètres. Les entraînements se sont divisés en trois grandes phases.

Dans la première phase, nous avons créé des groupes de données possédant six particules de l'état final avec une topologie de type 3o3.

Pour chacun des groupes, la masse de la particule « racine », c'est-à-dire à la tête de la désintégration, est de 100 ; les masses de toutes les particules de l'état final sont de 5. Puis, pour chacun des groupes, la masse des deux particules intermédiaires varie entre 30 et 50. Toutes ces masses sont données en unités arbitraires. L'énergie de transfert varie de 5000 à 8200 unités arbitraires. Le groupe de données d'entraînement et de validation possède chacun 5000 événements, c'est-à-dire 5000 désintégrations.

Nous avons ensuite entraîné le réseau de neurones à graphes et relevé sa performance. Nous définissons ici sa performance comme étant la proportion moyenne de matrices des relations correctement prédites sur les 25 dernières époques de l'entraînement. Nous pouvons trouver la Table B.1 répertoriant les données avec leurs performances dans l'annexe B.

De la même manière, nous avons réalisé deux autres phases de tests :

- une phase avec huit particules de l'état final, avec une topologie de type 4o4. La Table C.1 répertoriant les données avec leurs performances est présente dans l'annexe C
- une phase avec dix particules de l'état final, avec une topologie de type 5o5. La Table D.1 répertoriant les données avec leurs performances est présente dans l'annexe D

La Figure 5.5 nous montre les performances de validation obtenues pour les trois grandes phases. En abscisse, nous avons représenté l'énergie de transfert divisée par la masse au carré de la particule au sommet des désintégrations. Cela est justifié par la volonté de représenter sur les mêmes graphiques l'énergie de transfert de la désintégration $\Upsilon(4S)$ qui nous intéresse, en trait pointé rouge. Les résultats obtenus pour le réseau de neurones à graphes sont représentés en ronds bleus et les résultats pour la méthode Trust sont représentés en croix orange.

L'énergie de transfert varie en moyenne de 5000 à 8200 MeV². Nous constatons que pour certaines plages de valeurs, les performances sont très mauvaises. Nous pouvons en particulier remarquer une plage de valeur minimales pour un rapport compris entre 0.52 et 0.60 pour la topologie 4o4 et pour un rapport compris entre 0.5 et 0.6 pour la topologie 5o5.

Par ailleurs, l'énergie de transfert de la désintégration de la résonance $\Upsilon(4S)$ vaut 5616 MeV². Elle est indiquée par le trait en pointillé rouge sur la Figure 5.5. Elle se trouve donc dans les plages de performances les plus faibles, notamment pour les topologies 4o4 et 5o5. C'est pour cela qu'il est difficile d'obtenir de bonnes performances pour ces topologies.

Cette différence de performance provient de la manière dont notre réseau de neurones à graphes apprend à partir des données.

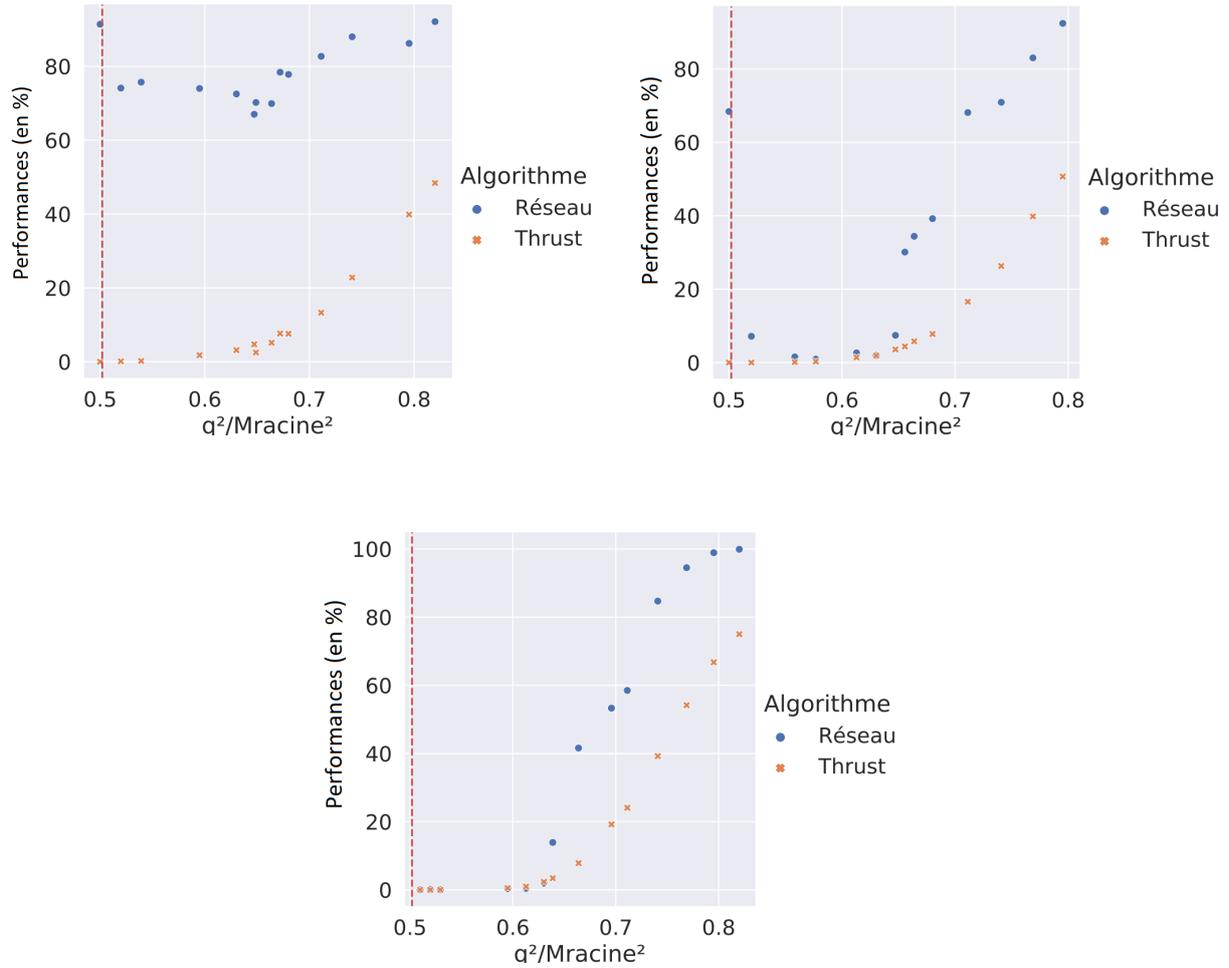


FIGURE 5.5 – Performances du réseau de neurones pour les topologies 3o3 (en haut à gauche), 4o4 (en haut à droite) et 5o5 (en bas au milieu) en fonction de q^2/M_{racine}^2 . Chaque groupe de données possède 5000 évènements pour l’entraînement et 5000 évènements pour la validation.

En effet, lorsque l’énergie de transfert de la désintégration est importante, au-delà de 7000 MeV^2 , alors les deux mésons B issus de la désintégration de $\Upsilon(4S)$ possèdent des quadri-impulsions très différentes. Nous attribuons alors chaque particule de l’état final au méson B duquel elle est issue grâce aux composantes des quadri-impulsions. C’est ce que fait le réseau de neurones. C’est une méthode très peu complexe, qui pourrait être réalisée avec un autre type d’algorithme, comme par exemple la méthode Thrust. Par ailleurs, comme nous nous y attendions, les performances sont bonnes pour la méthode Thrust pour des désintégrations possédant de grandes énergies de transfert, supérieure à 7000 MeV^2 ou à un rapport q^2/M_{racine}^2 supérieur à 0.6. Plus l’énergie de transfert est grande, plus la méthode Thrust devient performante. Cependant, il est important de noter que peu importe l’énergie de transfert considérée, les performances seront toujours supérieures avec le réseau de neurones qu’avec la méthode Thrust. Ainsi, nous pouvons en conclure que le réseau de neurones à graphes n’apprend pas uniquement grâce

à une simple séparation des quadri-impulsions, il prend en considération d'autres motifs plus complexes présents dans les données.

En revanche, si l'énergie de transfert de la désintégration est faible, inférieure à 7000MeV^2 , alors il n'est plus possible de distinguer de quel méson B vient chaque particule à partir des composantes des quadri-impulsions.

Le réseau de neurones attribue alors chaque particule de l'état final au méson B duquel elle est issue en combinant les différentes masses entre elles jusqu'à trouver la meilleure combinaison possible. Cette combinaison sera alors la désintégration la plus probable et donc celle retenue par le réseau de neurones à graphes. Cette méthode est beaucoup plus complexe et nécessite des temps d'entraînement plus longs. Sur ces plages de valeurs, la méthode Trust n'est pas efficace : aucun motif n'est déductible des composantes des quadri-impulsions.

Les plages de valeurs pour lesquelles les performances de validation que nous voyons sur la Figure 5.5 sont minimales correspondent donc au moment où le réseau de neurones décide de changer de méthode d'apprentissage. Elles correspondent à un rapport $q^2/\text{Masse}^2(\text{racine})$ de 0.65 pour la topologie 3o3, un rapport $q^2/\text{Masse}^2(\text{racine})$ de 0.58 pour une topologie 4o4 et un rapport $q^2/\text{Masse}^2(\text{racine})$ compris entre 0.5 et 0.6 pour la topologie 5o5.

Comme nous l'avons précisé précédemment, la désintégration $\Upsilon(4S)$ possède une énergie de transfert de 5616 MeV^2 , correspondant à un rapport $q^2/\text{Masse}^2(\text{racine})$ environ égal à 0.5. Ainsi, la méthode Trust est inefficace pour mener à bien notre projet. C'est pour cela que pour la suite de notre étude, nous utiliserons uniquement le réseau de neurones à graphes.

5.3 Deuxième étude : désintégrations réelles de la résonance $\Upsilon(4S)$

Au-delà des hyperparamètres donnés au réseau de neurones à graphes et de l'énergie de transfert q^2 , les performances sont influencées par la structure des données. Plusieurs paramètres sont à prendre en compte. Nous allons nous concentrer principalement sur les paramètres de structure. Parmi les paramètres de structure, nous allons nous intéresser au nombre de particules présentes dans l'état final et à l'écart du nombre de particules de l'état final qui existe entre le nombre de particules de l'état final du premier méson et du deuxième, que nous appellerons "écart topologie". Nous nous intéresserons également au minimum de la structure de désintégration, par exemple "2" pour une topologie de type "2o7". Nous étudions également l'influence de l'entraînement sur la validation. Pour cela, nous avons comparé les performances de validation obtenues lorsque nous réalisons l'entraînement avec toutes les désintégrations possibles (entraînement global) ou seulement avec la désintégration d'intérêt (entraînement individuel).

5.3.1 Désintégrations entraînées individuellement et validées individuellement

Nous avons produit des groupes de données, en remplaçant les valeurs génériques par les valeurs réelles des particules de la désintégration $\Upsilon(4S)$. Ainsi, la particule racine, qui est la résonance $\Upsilon(4S)$, possède une masse de $10\,579\text{ MeV}$, les deux mésons B possèdent chacun

une masse de 5279.8 Mev. Ensuite, nous avons donné à chaque particule de l'état final sa masse réelle. Chaque particule de l'état final apparaît dans les événements selon sa proportion que nous retrouvons dans les données de Belle II [16]. Nous avons réalisé les entraînements en augmentant progressivement le nombre de particules de l'état final, en partant de quatre jusqu'à dix. Comme précédemment, tous les groupes de données possèdent 5000 événements pour l'entraînement et 5000 événements pour la validation. L'annexe F, Table F.1 répertorie les performances obtenues pour l'entraînement et la validation. Nous avons représenté les performances de validation de la Table F.1 dans les Figures 5.6, 5.7 et 5.8 en fonction des paramètres suivants : le nombre de particules de l'état final, l'écart topologie, le nombre de particules minimal de la topologie appelé min (par exemple min est égal à 2 pour une topologie 2o3) et le nombre de particules maximal de la topologie appelé max (par exemple max est égal à 3 pour une topologie 2o3).

Influence du nombre de particules de l'état final

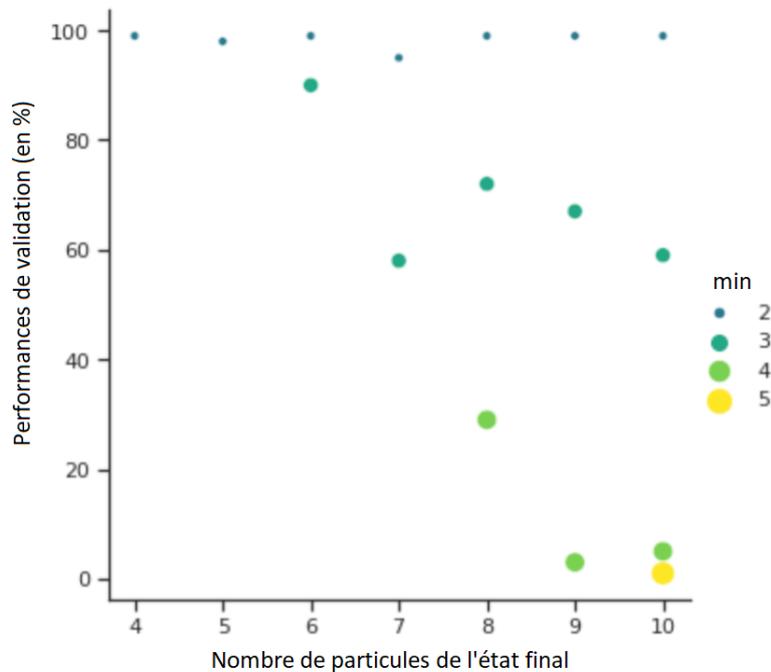


FIGURE 5.6 – Influence du nombre de particules de l'état final et du minimum sur la performance de la validation.

La structure de chaque topologie influence les performances de l'entraînement et de la validation. En effet le réseau de neurones à graphes déduit les liens qui existent entre toutes les particules de l'état final. Pour cela, il teste toutes les combinaisons de particules de l'état final possibles pour chaque méson B , jusqu'à trouver la meilleure. Il est donc naturel que la complexité augmente si le nombre de particules de l'état final augmente. Ainsi, la performance de validation pour une topologie 2o2, ayant quatre particules de l'état final, est très bonne, 99.1%,

tandis que la performance de validation de la topologie 4o5, ayant neuf particules de l'état final, est faible : 3.47%. Cependant, comme nous pouvons le voir dans la Table F.1 de l'annexe F, il existe de nombreuses exceptions, comme la topologie 2o8 (dix particules de l'état final) qui possède une des meilleures performances de validation. Nous pouvons trouver l'influence du nombre de particules de l'état final sur la performance de validation dans la Figure 5.6.

Influence de l'écart topologie

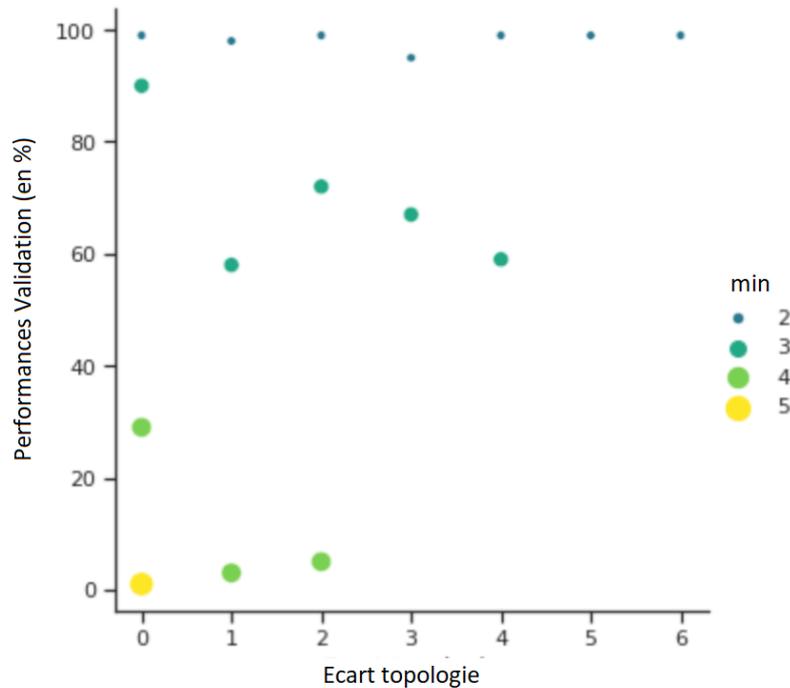


FIGURE 5.7 – Influence de l'écart topologie et du minimum sur la performance de la validation.

Les performances de l'entraînement et de la validation sont également influencées par l'écart topologie. Par exemple, les topologies 8o2 et 3o7 ont le même nombre de particules de l'état final, dix, mais ils n'ont pas la même différence : six pour le 2o8 (huit moins deux) et quatre pour le 3o7 (sept moins trois). Comme dit précédemment, la topologie 2o8 possède une excellente performance de validation, ce qui n'est pas le cas de la topologie 3o7. Cela vient du fait que la topologie 2o8 possède une désintégration triviale à 2 corps. Ce type de désintégration est très facile à reconnaître pour le réseau de neurones à graphes, et donc l'attribution de chaque particule à son méson B d'origine n'en est que plus facile. Nous pouvons trouver l'influence de l'écart topologie sur la performance de validation dans la Figure 5.7.

Influence du minimum

Comme dit précédemment, et comme nous pouvons voir dans la Table F.1 de l'annexe F que chaque topologie possédant cette désintégration triviale (2o2, 2o3, 2o4, 2o5, 2o6, 2o7 et 2o8)

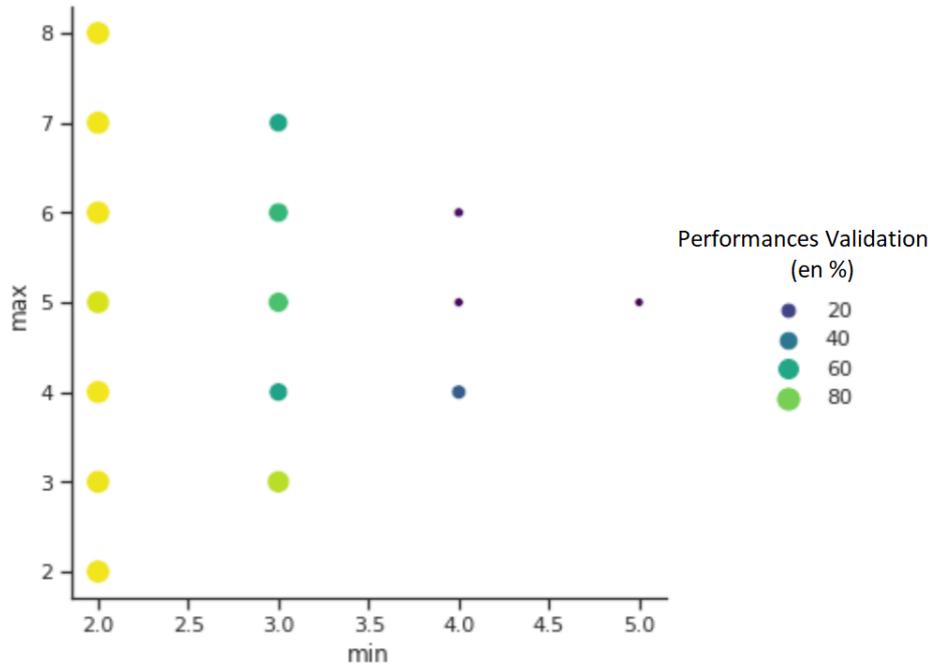


FIGURE 5.8 – Influence du maximum et du minimum sur la performance de la validation.

possède une excellente performance. Nous pouvons aller plus loin dans cette analyse : le minimum influe nettement la performance de la validation, indépendamment des autres paramètres, comme nous pouvons le constater dans les Figures 5.6, 5.7 et 5.8. Les meilleurs résultats sont obtenus lorsque ce minimum est le plus faible possible. Nous n’avons pas réalisé des entraînements pour un minimum inférieur à 2 car cela signifierait que le méson B ne se désintègre pas.

5.3.2 Désintégrations entraînées globalement et validées individuellement

L’annexe G, Table G.1 répertorie les performances obtenues pour l’entraînement et la validation. Dans cette partie, chaque processus a été réalisé avec 2500 évènements pour l’entraînement et 2500 évènements pour la validation. Nous sommes passés de 5000 à 2500 évènements car nous avons constaté un effet de seuil dans l’apprentissage du réseau de neurones à graphes : les performances sont très similaires pour des groupes de données possédant 5000 évènements et 2500 évènements. En revanche, l’utilisation de 5000 évènements requiert un temps d’apprentissage assez long, variant de quelques heures à quelques jours. Ainsi, et pour la suite de ce projet, nous avons utilisés des groupes de données ayant 2500 évènements pour l’entraînement et pour la validation.

Influence du nombre de particules de l’état final

En entraînant globalement les données, et en les validant individuellement, nous obtenons la même relation entre performance de validation et nombre de particules de l’état final. En effet, si

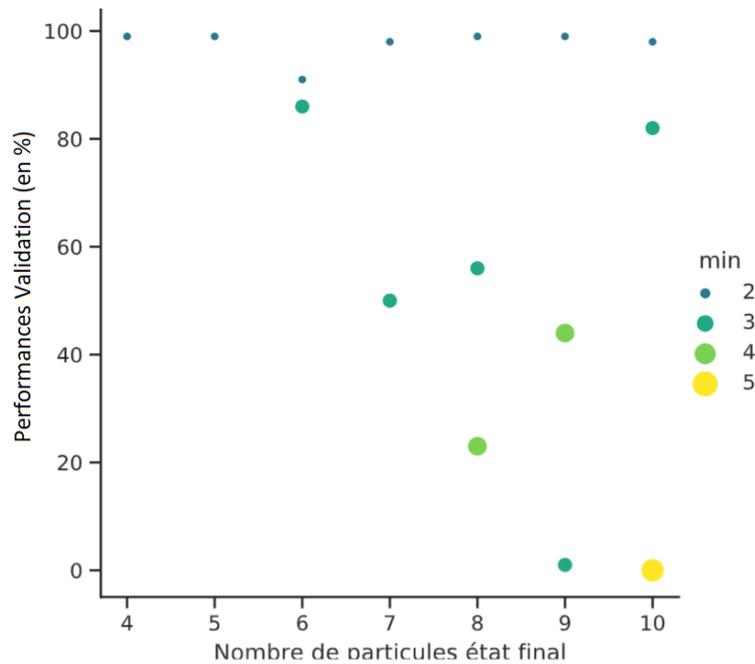


FIGURE 5.9 – Influence du nombre de particules de l'état final et du minimum sur la performance de la validation.

le nombre de particules de l'état final augmente, la performance de validation diminue. Mais ce n'est pas le paramètre majeur influençant la performance de validation. Nous avons représenté ce résultat sur la Figure 5.9.

Influence de l'écart topologie

En entraînement globalement les données, et en les validant individuellement, nous obtenons la même relation entre performance de validation et écart topologie. En effet, plus l'écart topologie est petit, plus la performance de validation est élevée. Mais ce n'est pas le paramètre majeur influençant la performance de validation. Nous avons représenté ce résultat sur la Figure 5.10

Influence du minimum

En entraînement globalement les données, et en les validant individuellement, nous obtenons la même relation entre performance de validation et minimum. En effet, plus le minimum est petit, plus la performance de validation est élevée. C'est le paramètre majeur influençant la performance de validation. Nous avons représenté ce résultat sur la Figure 5.10

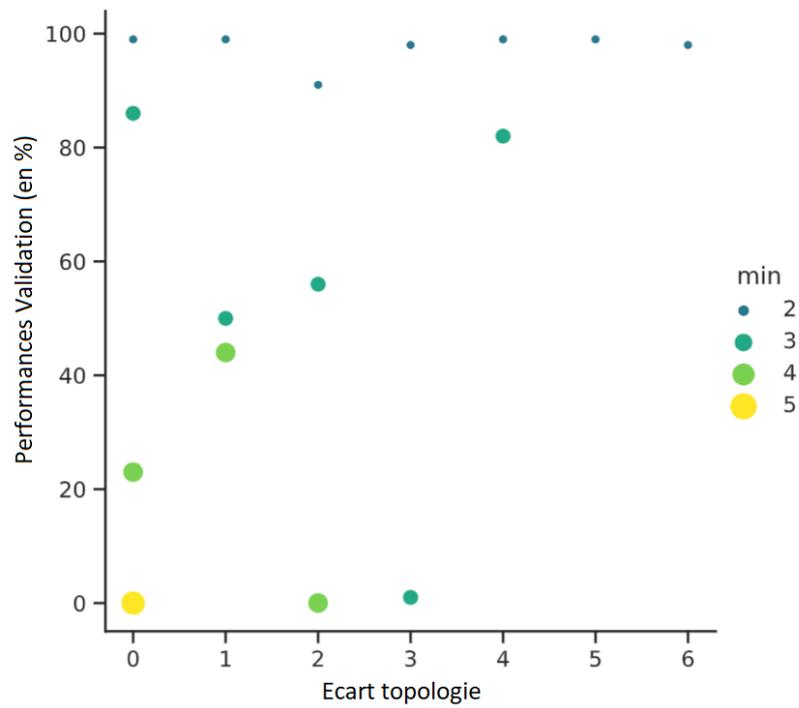


FIGURE 5.10 – Influence de l'écart topologie et du minimum sur la performance de la validation.

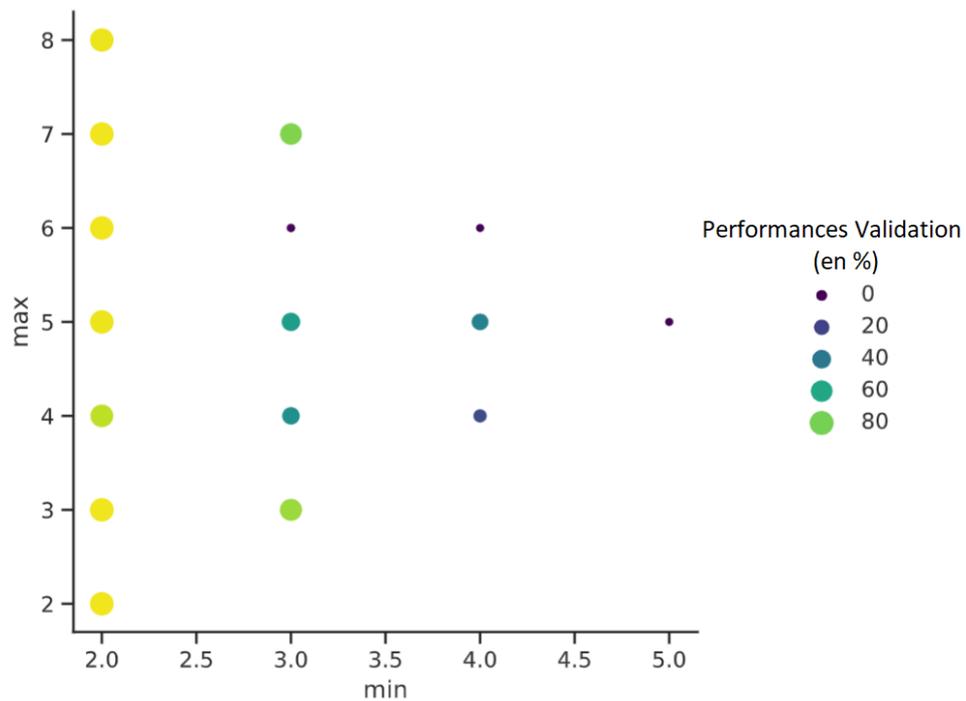


FIGURE 5.11 – Influence du maximum et du minimum sur la performance de la validation.

5.3.3 Comparaison entre désintégrations entraînées individuellement/globalement et validées individuellement

La Figure 5.12 représente les performances de validation individuelles obtenues pour chaque topologies. Les ronds représentent les performances de validation obtenues après un entraînement global des données et les croix représentent les performances de validation obtenues après entraînement individuel des données.

Comme précédemment, nous constatons que le minimum dans la structure de désintégration joue un rôle primordial dans la performance de validation, peu importe le nombre de particules de l'état final, l'écart topologie, ou un entraînement individuel ou global. Par exemple, les performances de validation pour les topologies 2o3 ou 2o8 sont très élevées, malgré leurs différences dans leurs structures ou d'entraînement. A contrario, la topologie 5o5 obtient une très mauvaise performance de validation, autant pour un entraînement global que individuel. Autrement dit, même si le réseau de neurones connaît cette topologie, et a été entraîné pour réaliser la classification des particules de l'état final pour cette topologie spécifiquement grâce à un entraînement individuel, il n'est pas capable de mieux assigner chaque particule de l'état final au méson B duquel elle est issue que lorsque le réseau a été entraîné sur un grand nombre de topologies. Le minimum a donc un très fort impact sur les performances de validation, peu importe les conditions d'apprentissage.

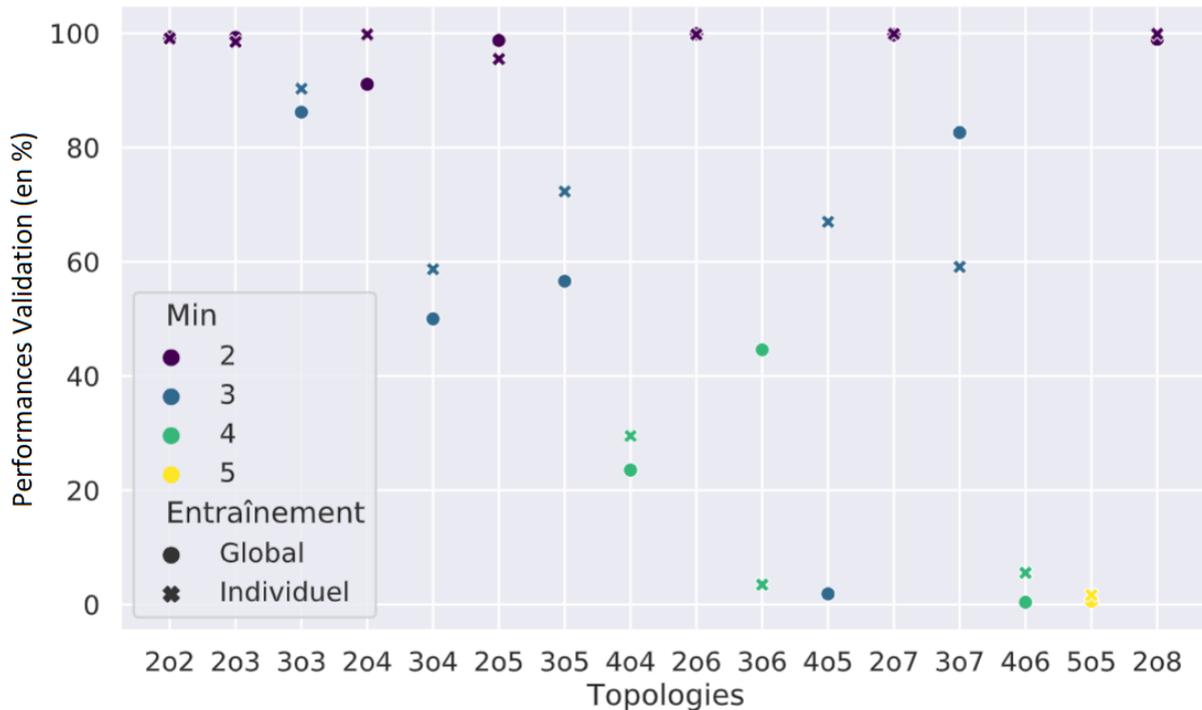


FIGURE 5.12 – Performances de validation individuelles obtenues pour toutes les topologies, entraînement global (rond) ou individuel (croix).

La Figure 5.13 représente la différence entre les performances de validation obtenues pour

des entraînements globaux et individuels, pour chaque topologie. Nous ne remarquons pas de corrélation entre cette différence et le minimum. En revanche, nous remarquons que le réseau de neurones obtient de meilleures performances de validation lorsque l'entraînement est individuel. Autrement dit, lorsque le réseau de neurones apprend à partir d'une seule topologie, il est plus à même d'assigner chaque particule de l'état final au méson B duquel elle est issue.

Il faut cependant nuancer ces résultats. En effet, comme nous l'avons expliqué, lors de l'initialisation du réseau de neurones, les poids synaptiques et les biais possèdent des valeurs aléatoires. Ainsi, les performances de validation sont soumises à des variations statistiques que nous ne pouvons pas estimer avec précisions à l'heure actuelle.

Pour obtenir plus de précision sur les résultats énoncés précédemment, notamment sur la différence entre entraînement global ou individuel, il faudrait réaliser plusieurs fois la séquence entraînement-validation pour chaque topologie, puis étudier la moyenne des performances de validation obtenues. Avec cette méthode, nous pourrions très certainement réduire la différence qui existe dans la Figure 5.13 pour les topologies 6o3 et 4o5 que nous ne pouvons pas expliquer à ce jour.

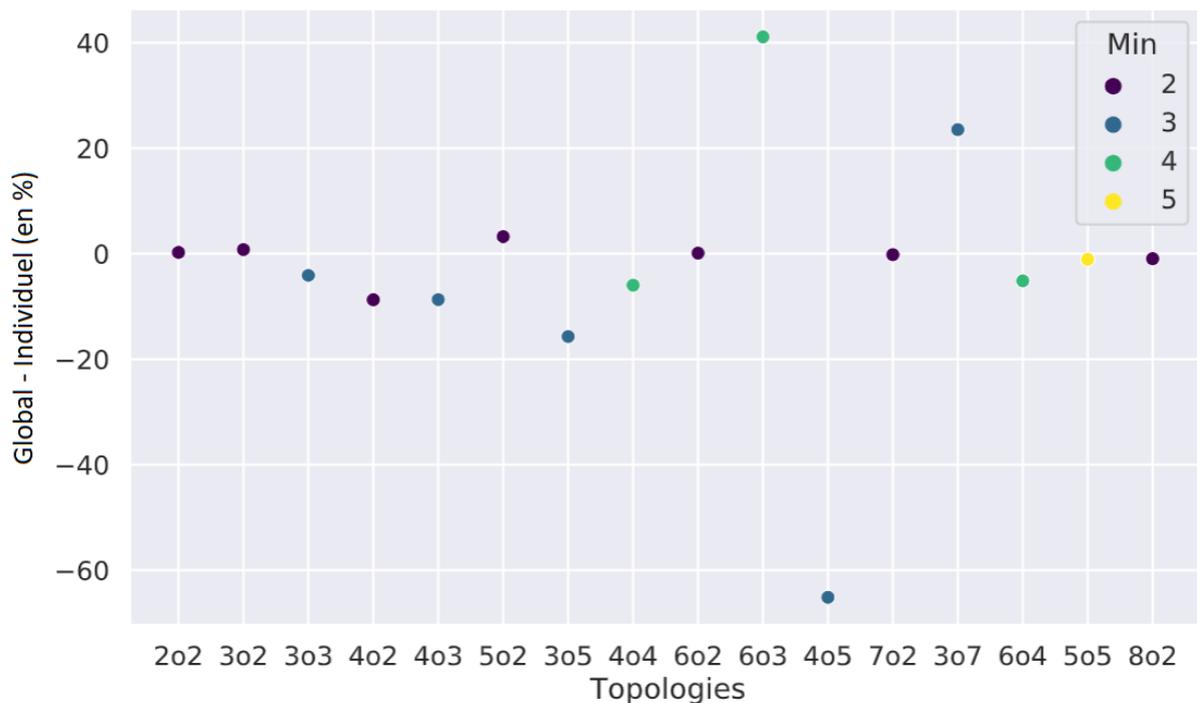


FIGURE 5.13 – Différence des performances de validation individuelles obtenues entre l'entraînement global et l'entraînement individuel, pour chaque topologie.

5.3.4 Désintégrations entraînées globalement et validées globalement

Nous avons ensuite progressivement augmenté la complexité de l'entraînement en réunissant plusieurs topologies. Les topologies sélectionnées sont celles présentes dans les groupes de données

entraînés précédemment. Chaque topologie possède 2500 évènements pour l'entraînement et 2500 évènements pour la validation. La Table 5.1 répertorie les performances obtenues pour l'entraînement et la validation.

Nombre FSP	Topologies	Performance Training	Performance Validation
4, 5 et 6	2o2, 3o2, 3o3, 4o2	89.2%	86.7%
4, 5, 6 et 7	2o2, 3o2, 3o3, 4o2, 3o4, 5o2	91.1%	73.1%
4, 5, 6, 7 et 8	2o2, 3o2, 3o3, 4o2, 3o4, 5o2, 3o5, 4o4, 6o2	89.4%	77.0%
4, 5, 6, 7, 8 et 9	2o2, 3o2, 3o3, 4o2, 3o4, 5o2, 3o5, 4o4, 6o2, 3o6, 4o5, 7o2	75.7%	70.4%
4, 5, 6, 7, 8, 9 et 10	2o2, 3o2, 3o3, 4o2, 3o4, 5o2, 3o5, 4o4, 6o2, 3o6, 4o5, 7o2, 3o7, 6o4, 5o5, 8o2	72.1%	65.9%

TABLE 5.1 – Nombre FSP : nombre de particules de l'état final. Performances d'entraînement et de validation pour des ensembles de topologies. Chaque topologie possède 2500 évènements pour l'entraînement et 2500 évènements pour la validation.

Comme nous pouvions nous y attendre grâce aux résultats précédents, les performances de validation diminuent progressivement au fur et à mesure que l'on augmente le nombre de topologies dans la phase d'entraînement et de validation, et également en fonction du nombre de particules de l'état final (abrégé FSP dans le tableau). Les résultats pour un entraînement global et une validation globale demeurent cependant satisfaisants. Ainsi, le réseau de neurones pourra être utilisé sur les données Monte Carlo de l'expérience Belle II.

Conclusion et perspectives

Au cours de ce projet, nous avons montré qu'il était possible d'attribuer une particule de l'état final à son méson B d'origine, à partir de la matrice des relations, des masses des particules et des quadri-impulsions. Nous avons optimisé le réseau de neurones à graphes pour répondre à ce problème en augmentant progressivement la complexité des données : premièrement avec des données génériques simulées par la bibliothèque phasespace, puis avec des données simulées par la bibliothèque phasespace en reprenant les masses et proportions réelles des particules de l'état final. Enfin, nous avons entraîné le réseau de neurones avec plusieurs topologies différentes en même temps.

Au cours de ces entraînements, nous avons mis en évidence l'influence des paramètres de structures des données, ainsi que de l'énergie de transfert de la désintégration. Nous avons également montré que les performances de validation étaient influencées par l'entraînement pouvant être global ou individuel. Nous avons comparé nos résultats à la méthode Trust, utilisée par l'expérience Belle II, et avons montré l'importance de l'utilisation de l'apprentissage automatique pour ce type de problème.

Ces résultats seront réutilisés pour entraîner notre réseau de neurones à graphes sur des données Monte Carlo de l'expérience Belle II. Ces données contiendront plusieurs topologies différentes.

Notre réseau de neurones à graphes pourra être amélioré sur deux points. Nous devons affiner les paramètres du réseau pour avoir de bonnes performances même lorsque les données possèdent du bruit. Nous devons également traiter le cas des particules manquantes. En effet, toutes les particules d'une désintégration ne sont pas forcément détectées, c'est par exemple le cas des neutrinos qui sont présents dans la désintégration de la résonance $\Upsilon(4S)$ mais qui ne sont pas détectés par le détecteur Belle II.

Bibliographie

- [1] Institut Pluridisciplinaire Hubert-Curien, <http://www.iphc.cnrs.fr/>.
- [2] Institut National de la Physique Nucléaire et de la Physique des Particules, <https://in2p3.cnrs.fr/fr/institut-national-de-physique-nucleaire-et-de-physique-des-particules-0>.
- [3] Physique des particules : la France entre dans la collaboration internationale BELLE-II, <https://in2p3.cnrs.fr/fr/cnrsinfo/physique-des-particules-la-france-entre-dans-la-collaboration-internationale-belle-ii>.
- [4] W. Altmannshofer et al. The Belle II Physics Book. *PTEP*, 2019(12) :123C01, 2019. [Erratum : *PTEP* 2020, 029201 (2020)].
- [5] A. J. Bevan et al. The Physics of the B Factories. *Eur. Phys. J. C*, 74 :3026, 2014.
- [6] Yuki Yoshi Ohnishi et al. Accelerator design at SuperKEKB. *PTEP*, 2013 :03A011, 2013.
- [7] Kazunori Akai, Kazuro Furukawa, and Haruyo Koiso. SuperKEKB Collider. *Nucl. Instrum. Meth. A*, 907 :188–199, 2018.
- [8] Mark Thomson. *Modern Particle Physics*. Cambridge University Press, 2013.
- [9] Jayashree Kalpathy-Cramer Michael F. Chiang Rene Y. Choi, Aaron S. Coyner and J. Peter Campbell. Introduction to Machine Learning, Neural Networks, and Deep Learning. *Transl Vis Sci Technol.*, 2020 Feb 27.
- [10] I.H Sarker. Machine Learning : Algorithms, Real-World Applications and Research Directions. *SN Computer Science volume*, 22 March 2021.
- [11] Jonathan Shlomi, Peter Battaglia, and Jean-Roch Vlimant. Graph Neural Networks in Particle Physics. 7 2020.
- [12] Peter W. Battaglia et al. Relational inductive biases, deep learning, and graph networks. *CoRR*, abs/1806.01261, 2018.
- [13] PyTorch, <https://pytorch.org>.
- [14] Documentation du module Geometric de Pytorch, <https://pytorch-geometric.readthedocs.io>.
- [15] Documentation de phasespace, <https://pypi.org/project/phasespace/>.
- [16] Valerio Bertacchi et al. Track finding at Belle II. *Comput. Phys. Commun.*, 259 :107610, 2021.
- [17] Cross Entropy, <https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html>.
- [18] Adam, <https://pytorch.org/docs/stable/optim.html>.

A

Pseudo-code du réseau de neurones à graphes

Fonction RéseauDeNeuroneAGraphes(E,V) :

Pour $k \in \{1 \dots N^e\}$ **faire**

$$e'_k \leftarrow \phi(e_k, v_{rk}, v_{sk})$$

1. Calcul de la mise à jour des attributs des liens

Fin de Pour

Pour $i \in \{1 \dots N^n\}$ **faire**

$$\text{Définir } E'_i = \left\{ \left(e'_k, r_k, s_k \right) \right\}_{r_k=i, k=1:N^e}$$

$$\bar{e}'_i \leftarrow \rho^{e \rightarrow v} \left(E'_i \right)$$

2. Agrégation des attributs des liens sur chaque noeud

$$v'_i \leftarrow \phi^v(\bar{e}'_i, v_i)$$

3. Calcul des attributs des noeuds

Fin de Pour

$$\text{Définir } V' = \left\{ v'_i \right\}_{i=1:N^v}$$

Ensemble des noeuds

$$\text{Définir } E' = \left\{ e'_k, r_k, s_k \right\}_{k=1:N^e}$$

Ensemble des liens

$$\bar{e}' \leftarrow \rho(E')$$

4. Agrégation globale des attributs des liens

$$\bar{v}' \leftarrow \rho(V')$$

5. Agrégation globale des attributs des noeuds

Retourner (E', V')

Fin de Fonction

Fonctions ϕ : réseaux de neurones MLP

Fonctions ρ : fonction qui calculent la moyenne des arguments d'entrée

e_k : attribut du lien k entre deux noeuds

v_{rk} : attribut du noeud vers lequel arrive le lien e_k

v_{sk} : attribut du noeud duquel part le lien e_k

s_k : noeud d'où part l'information du lien k

r_k : v_s : noeud où arrive l'information du lien k

\bar{e}' : nouvel attribut pour le lien

\bar{v}' : nouvel attribut pour le noeud

N^e : nombre total de liens

N^v : nombre total de noeuds

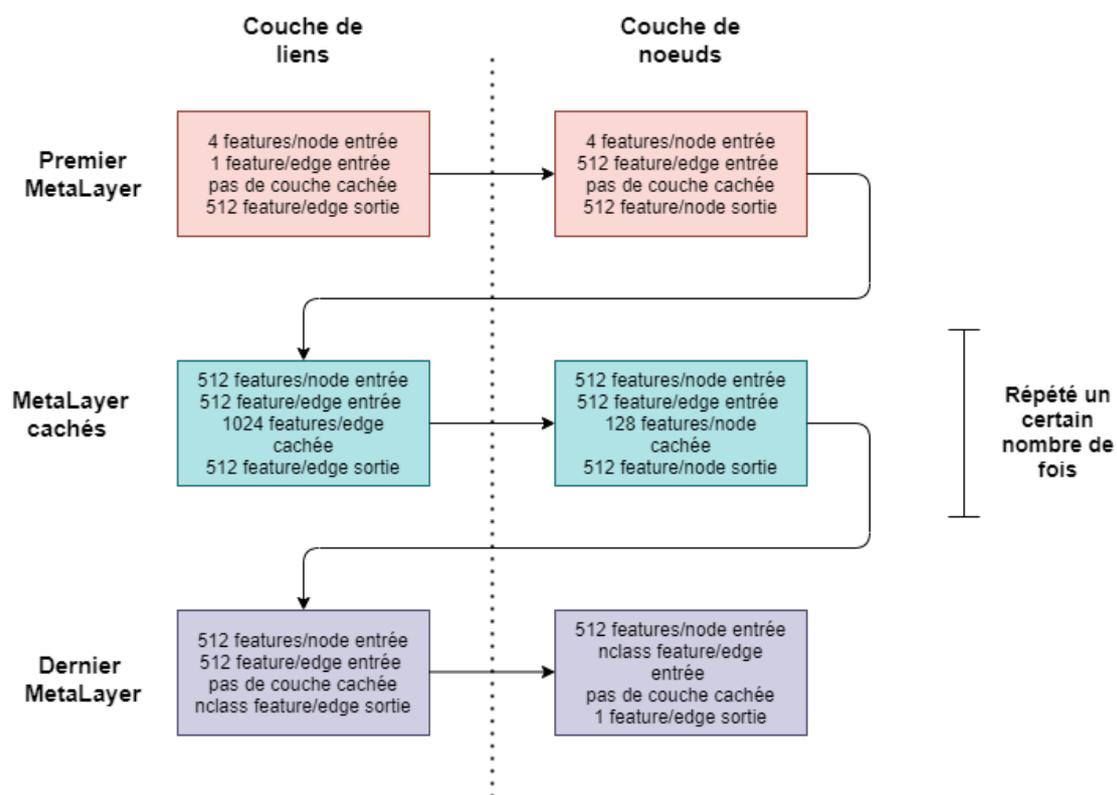


FIGURE A.1 – Structure du réseau de neurones à graphes.

B

Performance 3o3

Masse racine	Masse droite	Masse gauche	q^2	Perf. Validation	Perf. Thrust
100	30	30	8200	92.1%	48.4%
100	31	31	8078	89.0%	-
100	32	32	7952	86.2%	39.86%
100	36	36	7408	88.0%	22.80%
100	38	38	7112	82.7%	13.28%
100	40	40	6800	77.8%	7.85%
100	40.5	40.5	6719.5	78.4%	7.64%
100	40.8	40.8	6670.72	78.0%	-%
100	41	41	6638	69.9%	5.14%
100	41.2	41.2	6605.12	72.0%	-
100	41.5	41.5	6555.5	69.6%	-
100	41.9	41.9	6488.78	70.2%	2.5%
100	42	42	6472	67.0%	4.72%
100	42.5	42.5	6387.5	62.6%	-
100	43	43	6302	72.5%	3.14%
100	45	45	5950	74.0%	1.76%
100	48	48	5392	75.7%	0.20%
100	49	49	5198	74.1%	0.12%
100	50	50	5000	91.4%	0.00%

TABLE B.1 – Performances obtenues pour la topologie 3o3 pour le réseau de neurones (Perf. Validation) et pour la méthode Trust (Perf. Trust), pour différentes valeurs de q^2 . Chaque topologie possède 5000 évènements pour l’entraînement et 5000 évènements pour la validation. FSP : six particules de l’état final.

C

Performance 4o4

Masse racine	Masse droite	Masse gauche	q^2	Perf. Validation	Perf. Thrust
100	27	27	8542	99.9%	0.0%
100	32	32	7952	92.4%	50.68%
100	34	34	7688	83.0%	39.82%
100	36	36	7408	70.9%	26.30%
100	38	38	7112	68.1%	16.58%
100	40	40	6800	39.2%	7.78%
100	40.5	40.5	6719.5	41.5%	-
100	41	41	6638	34.4%	5.80%
100	41.5	41.5	6555.5	30.1%	4.40%
100	42	42	6472	7.43%	3.58%
100	43	43	6302	1.95%	1.90%
100	44	44	6128	2.64%	1.40%
100	45	45	5950	4.47%	-
100	46	46	5768	0.94%	0.28%
100	47	47	5582	1.55%	0.14%
100	48	48	5392	1.74%	-
100	49	49	5198	7.17%	0.0%
100	49.5	49.5	5099.5	46.5%	-
100	50	50	5000	68.4%	0.0%

TABLE C.1 – Performances obtenues pour la topologie 4o4 pour le réseau de neurones (Perf. Validation) et pour la méthode Trust (Perf. Trust), pour différentes valeurs de q^2 . Chaque topologie possède 5000 événements pour l’entraînement et 5000 événements pour la validation. FSP : huit particules de l’état final.

D

Performance 5o5

Masse racine	Masse droite	Masse gauche	q^2	Perf. Validation	Perf. Thrust
100	30	30	8200	99.9%	75.00%
100	32	32	7952	98.9%	66.76%
100	34	34	7688	94.5%	54.18%
100	36	36	7408	84.7%	39.22%
100	38	38	7112	58.5%	24.08%
100	39	39	6958	53.3%	19.2%
100	40	40	6800	53.6%	-
100	41	41	6638	41.6%	7.84%
100	42	42	6472	19.7%	1.-
100	42.5	42.5	6387.5	13.9%	3.42%
100	43	43	6302	1.97%	2.36%
100	44	44	6128	0.43%	0.96%
100	45	45	5950	0.31%	0.52%
100	48	48	5392	0.01%	-
100	48.5	48.5	5295.5	0.06%	0.0%
100	49	49	5198	0.11%	0.0%
100	49.5	49.5	5099.5	0%	0.0%

TABLE D.1 – Performances obtenues pour la topologie 5o5 pour le réseau de neurones (Perf. Validation) et pour la méthode Trust (Perf. Trust), pour différentes valeurs de q^2 . Chaque topologie possède 5000 évènements pour l’entraînement et 5000 évènements pour la validation. FSP : dix particules de l’état final.

E

Performances de validation : exemples

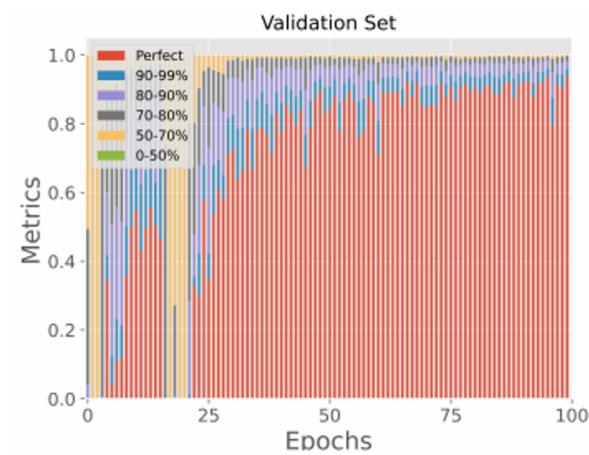


FIGURE E.1 – Performance de validation d'un groupe de données de topologie 3o3. FSP : six particules de l'état final.

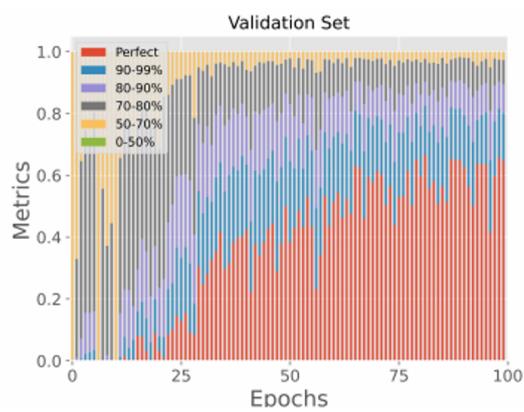


FIGURE E.2 – Performance de validation d'un groupe de données de topologie 4o3. FSP : sept particules de l'état final.

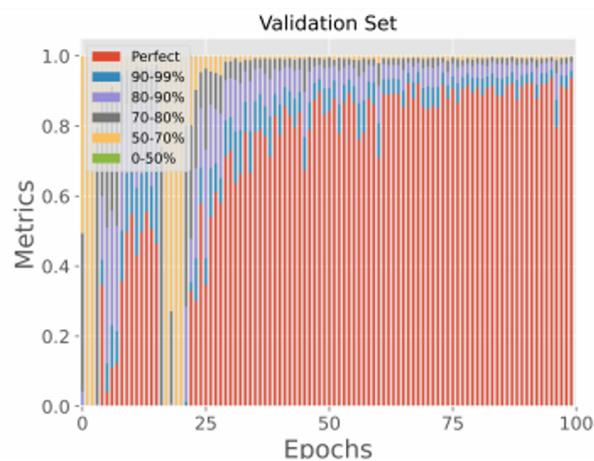


FIGURE E.3 – Performance de validation d’un groupe de données de topologie 4o4. FSP : huit particules de l’état final.

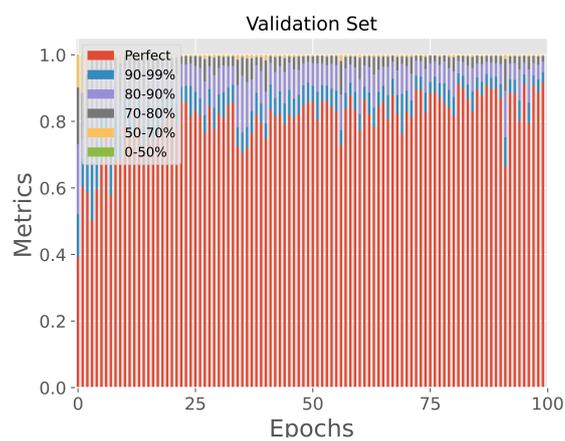


FIGURE E.4 – Performance de validation d’un groupe de données. FSP : entre quatre et six. 2500 évènements par topologie.

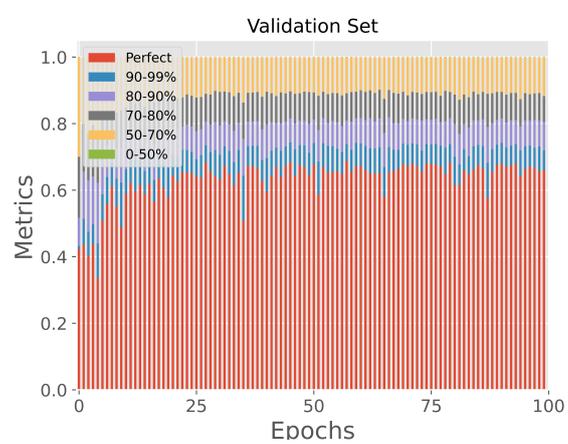


FIGURE E.5 – Performance de validation d'un groupe de données. FSP : entre quatre et dix. 2500 évènements par topologie.

F

Performances pour différentes topologies. Entraînement individuel et validation individuelle

Nombre FSP	Topologie	Performance Training	Performance Validation
4	2o2	99.1%	99.1%
5	2o3	98.6%	98.5%
6	3o3	91.8%	90.3%
6	2o4	99.8%	99.8%
7	3o4	74.7%	58.7%
7	2o5	95.5%	95.5%
8	3o5	84.3%	72.3%
8	4o4	34.9%	29.5%
8	2o6	100%	99.8%
9	4o5	4.04%	3.47%
9	3o6	67.8%	67.0%
9	2o7	100%	99.9%
10	3o7	63.9%	59.1%
10	4o6	6.38%	5.54%
10	5o5	1.70%	1.65%
10	2o8	100%	99.9%

TABLE F.1 – Nombre FSP : nombre de particules de l'état final. Performances d'entraînement et de validation pour différentes topologies. Chaque topologie possède 5000 évènements pour l'entraînement et 5000 évènements pour la validation.

G

Performances pour différentes topologies. Entraînement global et validation individuelle

Nombre FSP	Topologie	Performance Entraînement	Performance Validation
4	2o2	68.67%	99.34%
5	2o3	69.22%	99.28%
6	3o3	72.08%	86.19%
6	2o4	76.48%	91.07%
7	3o4	58.1%	50.3%
7	2o5	67.08%	98.74%
8	3o5	64.71%	56.60%
8	4o4	70.86%	23.53%
8	2o6	72.62%	99.91%
9	4o5	65.29%	44.58%
9	3o6	69.87%	1.85%
9	2o7	69.86%	99.72%
10	3o7	78.66%	82.62%
10	4o6	72.48%	0.39%
10	5o5	72.87%	0.57%
10	2o8	68.66%	98.95%

TABLE G.1 – Nombre FSP : nombre de particules de l'état final. Performances d'entraînement et de validation pour différentes topologies. Chaque topologie possède 5000 évènements pour l'entraînement et 5000 évènements pour la validation.