

2022年度 修士論文

電子陽電子ヒッグスファクトリーのための
ジェット測定技術の研究

九州大学大学院 理学府物理学専攻
粒子物理学分野素粒子実験研究室

尾上 友紀

指導教員 末原大幹 助教 川越清以 教授

2023年2月21日



九州大学
KYUSHU UNIVERSITY

概要

2012 年に発見されたヒッグス粒子は、宇宙の物質の起源の解明につながる粒子であり、その性質は謎に包まれている。そのため、ヒッグスファクトリーによってヒッグス粒子を大量に生成し精密測定することで、暗黒物質などの標準理論を超える物理を切り開くことは、現代の物理学における重要事項となっている。本研究で念頭に置いている国際リニアコライダー (ILC) 計画は次世代の電子陽電子衝突型加速器であり、ヒッグスファクトリーとしての運転を期待されている。ヒッグスファクトリーにおけるヒッグス粒子生成事象は、多数のハドロンの束であるジェットを終状態に複数含むことが多いため、ジェットを高い精度で検出・再構成することは物理解析の性能に直結する。そのため、本論文では ILC のジェット測定技術に関する 2 つの研究を行なった。

ILC で重要な事象にはジェットを含むものが多くあるため、ILC の物理目標を達成するためには高いジェットエネルギー分解能が必要である。そのため、ILC では Particle Flow Algorithm と呼ばれる粒子識別のアルゴリズムによって高い分解能の達成を目指しており、これには非常に高精細な電磁カロリメータが求められる。シリコンタングステン電磁カロリメータは、1 つの読み出しセルの大きさが $5 \text{ mm} \times 5 \text{ mm}$ と非常に細分化されており、国際協力によって技術プロトタイプの開発が行われている。本研究では、15 層の検出層を持つプロトタイプを構築し、欧州原子核研究機構 (CERN) の Super Proton Synchrotron (SPS) 加速器にて最大 150 GeV の高エネルギービームを用いたテストビーム実験を行った。そして実験結果から、読み出しシステムの性能やシリコンセンサーの振る舞いについて調査し、将来の技術プロトタイプに向けた改善策を講じた。特に読み出しにおいてトリガー情報が誤って収集されてしまう Re-triggering 現象について調査を行った。

さらに、ILC のジェット再構成におけるフレーバー識別アルゴリズムの開発を、深層学習技術を用いて行った。フレーバー識別では、ジェットの構成粒子の種類や運動量、崩壊点に関する情報から、ジェットの元となる夸克のフレーバーを識別する。現在は従来の機械学習手法である Boosted Decision Trees が用いられているが、本研究では更なる識別性能の向上を目的に深層学習を用いたアルゴリズムを開発した。本アルゴリズムの最大の特徴は、フレーバー識別のためにグラフ構造のデータを構築し、グラフデータの学習が可能なニューラルネットワークであるグラフニューラルネットワークで学習を行った点である。グラフデータはデータの関係をグラフ形式で表現することで、データの相互関係を考慮することができる。これによって実際の物理現象と比較した際の情報損失を減らし、より高い精度で識別を行うことが出来ると考えた。本アルゴリズムの結果は従来技術と比較して b フレーバーの識別のみの改善となつたが、グラフデータの構築によって従来は別々にプロセスを行なっていた崩壊点検出のアルゴリズムを統合することができた。

目次

| | |
|---|-----------|
| 第1章 序論 | 12 |
| 1.1 素粒子物理学 | 12 |
| 1.1.1 標準模型 | 12 |
| 1.1.2 ヒッグス機構 | 14 |
| 1.2 国際リニアコライダー計画; ILC | 15 |
| 1.3 ILC の物理 | 16 |
| 1.3.1 ヒッグス生成過程と質量精密測定 | 16 |
| 1.3.2 ヒッグス結合定数の精密測定 | 17 |
| 1.3.3 ヒッグス自己結合 | 20 |
| 1.3.4 階層性問題 | 20 |
| 1.3.5 その他の新物理 | 21 |
| 1.4 ILC の検出器 | 21 |
| 1.4.1 ILD で検出器可能な粒子群 | 21 |
| 1.4.2 Particle Flow Algorithm (PFA) | 22 |
| 1.4.3 International Large Detector: ILD | 23 |
| 崩壊点検出器 | 23 |
| 中央飛跡検出器 | 24 |
| カロリメータ | 24 |
| ミューオン検出器 | 25 |
| 1.5 ILC のソフトウェア | 25 |
| 1.5.1 イベントジェネレータと検出器シミュレーション | 25 |
| 1.5.2 事象再構成 | 26 |
| 飛跡再構成 | 26 |
| 1.5.3 PFA | 26 |
| 崩壊点検出 | 27 |
| ジェットクラスタリング | 27 |
| フレーバー識別 | 27 |
| 1.6 本研究の目的 | 28 |

| | | |
|--------------|---------------------------------------|-----------|
| 1.6.1 | 高エネルギービームによる SiW-ECAL の性能評価 | 28 |
| 1.6.2 | 深層学習を用いたフレーバー識別アルゴリズムの開発 | 29 |
| 第 2 章 | シリコンタングステン電磁カロリメータ | 30 |
| 2.1 | 入射粒子と物質の相互作用 | 30 |
| 2.1.1 | 荷電粒子 | 30 |
| | 電離損失 | 30 |
| | 制動放射 | 32 |
| 2.1.2 | 光子 | 33 |
| 2.1.3 | ハドロン | 34 |
| 2.2 | 粒子検出器の動作原理 | 34 |
| 2.2.1 | ガス検出器 | 35 |
| 2.2.2 | 半導体検出器 | 35 |
| 2.2.3 | シンチレーション検出器 | 36 |
| 2.3 | シリコンタングステン電磁カロリメータ SiW-ECAL | 36 |
| 2.3.1 | SiW-ECAL の全体構造 | 36 |
| 2.3.2 | シリコン半導体検出器 | 36 |
| 2.3.3 | 読み出しシステム | 37 |
| 2.3.4 | タングステン吸収層 | 39 |
| 2.3.5 | 技術プロトタイプ | 40 |
| 第 3 章 | ビームテストによる評価試験 | 42 |
| 3.1 | 概要 | 42 |
| 3.1.1 | ビームライン | 42 |
| 3.2 | 実験セットアップ | 43 |
| 3.2.1 | 測定機器のセットアップ | 43 |
| 3.2.2 | 信号読み出し | 45 |
| 3.3 | 実験結果 | 46 |
| 3.3.1 | 検出器応答 | 46 |
| 3.3.2 | ペデスタル・Retriggering | 48 |
| 第 4 章 | 深層学習 | 54 |
| 4.1 | ニューラルネットワーク | 54 |
| 4.1.1 | パーセプトロン（単層ニューラルネットワーク） | 54 |
| 4.1.2 | 多層パーセプトロン（多層ニューラルネットワーク） | 55 |
| | 活性化関数 | 56 |
| | 出力層の設計 | 57 |
| | 損失関数 | 57 |

| | |
|---|-----------|
| 誤差逆伝播法 | 58 |
| 前処理 | 58 |
| ミニバッチ処理 | 59 |
| 最適化アルゴリズム | 59 |
| 過学習 | 61 |
| 4.1.3 ディープニューラルネットワーク | 62 |
| 4.2 グラフニューラルネットワーク | 63 |
| 4.2.1 メッセージパッシング | 63 |
| 4.2.2 Graph Convolution Network (GCN) | 64 |
| Spectral Graph Convolution | 64 |
| Spatial Graph Convolution | 65 |
| 4.2.3 Graph Attention Network (GAT) | 66 |
| 第 5 章 深層学習を用いたジェットフレーバー識別 | 68 |
| 5.1 イベントサンプルと先行研究 | 68 |
| 5.1.1 LCFIPlus におけるフレーバー識別 | 69 |
| 5.2 ディープニューラルネットワークによる実装 | 71 |
| 5.2.1 実装目的 | 71 |
| 5.2.2 入力変数とネットワークアーキテクチャ | 71 |
| 5.2.3 ハイパーパラメータの最適化 | 72 |
| 5.2.4 学習評価 | 74 |
| 5.3 グラフニューラルネットワークによる実装 | 78 |
| 5.3.1 実装目的 | 78 |
| 5.3.2 飛跡によるグラフデータセット | 78 |
| 5.3.3 ネットワークアーキテクチャ | 81 |
| 5.3.4 ハイパーパラメータの最適化 | 83 |
| 5.3.5 学習評価 | 85 |
| 第 6 章 まとめと今後の展望 | 89 |
| 付録 A AppendixA | 93 |
| A.1 飛跡のパラメータ | 93 |
| A.1.1 LDC 座標系 | 93 |
| A.1.2 飛跡パラメータ | 93 |
| xy 平面 | 94 |
| sz 平面 | 95 |
| 参考文献 | 99 |

図目次

| | | |
|------|---|----|
| 1.1 | 素粒子標準模型 | 13 |
| 1.2 | ヒッグスボテンシャル | 14 |
| 1.3 | ILC の概略図 | 15 |
| 1.4 | ILC におけるヒッグス生成過程のファインマンダイアグラム | 17 |
| 1.5 | ILC の重心系エネルギーに対するヒッグス生成断面積 | 17 |
| 1.6 | 標準模型におけるヒッグス粒子の質量と崩壊分岐比の関係 | 18 |
| 1.7 | SUSY, 複合ヒッグスモデルにおけるヒッグス結合定数 | 19 |
| 1.8 | ヒッグスの結合定数精度における ILC と HL-LHC の比較 | 19 |
| 1.9 | ヒッグス自己結合 | 20 |
| 1.10 | 階層性問題のフェルミオンループ | 21 |
| 1.11 | ILD と SiD の全体図 | 22 |
| 1.12 | ILD の断面図 [7] | 23 |
| 1.13 | ILD 崩壊点検出器の構造 [7] | 24 |
| 1.14 | ILC におけるシミュレーションと事象再構成、物理解析の流れとソフトウェア [1] | 26 |
| 1.15 | ジェット崩壊の様子 | 28 |
| 2.1 | 平均エネルギー損失と入射粒子速度の関係 | 32 |
| 2.2 | 光子と物質の相互作用 | 33 |
| 2.3 | 光子と鉛の相互作用における断面積と光子のエネルギー | 35 |
| 2.4 | ILD および ECAL の全体図 | 37 |
| 2.5 | SiW-ECAL の構造 [7] | 37 |
| 2.6 | シリコンパッドセンサー | 37 |
| 2.7 | SKIROC2A のアナログ部の回路図 | 39 |
| 2.8 | short slab の構造 | 41 |
| 2.9 | 多層読み出しのための CORE モジュール | 41 |
| 2.10 | FEV13(BGA), COB | 41 |
| 3.1 | CERN 加速器の全体図 [2] | 43 |

| | | |
|------|--|----|
| 3.2 | セットアップの全体図 | 44 |
| 3.3 | SiW-ECAL のプロトタイプモジュールと電源供給、読み出しシステム | 44 |
| 3.4 | SiW-ECAL のイベントモニター | 46 |
| 3.5 | 各ビーム種類におけるヒットマップ | 47 |
| 3.6 | ペデスタルヒストグラムの例 | 48 |
| 3.7 | 異なるビームエネルギーにおけるペデスタル | 49 |
| 3.8 | 全チャンネルに対するペデスタルのフィッティング | 50 |
| 3.9 | ダブルペデスタルの例 | 50 |
| 3.10 | 各層におけるダブルペデスタルのピーク間距離 | 51 |
| 3.11 | Trigger のマップ | 52 |
| 3.12 | Retriggeirng のマップ | 53 |
| 4.1 | パーセプトロン [24] | 54 |
| 4.2 | 多層パーセプトロン（ニューラルネットワーク） | 55 |
| 4.3 | 活性化関数のグラフ | 57 |
| 4.4 | ディープニューラルネットワーク | 63 |
| 4.5 | メッセージパッシング | 64 |
| 4.6 | GraphSAGE | 66 |
| 4.7 | Graph Attention Network | 67 |
| 5.1 | フレーバー識別のためのディープニューラルネットワークの概略図 | 72 |
| 5.2 | 各試行における評価関数の値 | 73 |
| 5.3 | RAdam の学習に対するパラメータによる評価関数の 2 次元ヒストグラム。 横軸が β_1 の値を、縦軸が β_2 の値を、3 次元方向が評価関数の値を表している。 | 74 |
| 5.4 | 各パラメータの重要度。表 5.4 に挙げたパラメータの重要度をそれぞれ示している。 | 74 |
| 5.5 | 学習経過における損失と精度 | 75 |
| 5.6 | ディープニューラルネットワークの学習における混合行列 | 75 |
| 5.7 | LCFIPlus と DNN の b-tag に関する比較 | 76 |
| 5.8 | LCFIPlus と DNN の c-tag に関する比較 | 77 |
| 5.9 | グラフデータの例 | 79 |
| 5.10 | ノードラベル毎のデータ数 | 80 |
| 5.11 | フレーバー識別のためのグラフデータを用いたネットワークの概略図 | 82 |
| 5.12 | 各試行における評価関数の値 | 84 |
| 5.13 | 評価関数の 2 次元ヒストグラム | 85 |
| 5.14 | 各パラメータの重要度 | 85 |
| 5.15 | 学習経過における損失と精度 | 86 |

| | | |
|------|---|----|
| 5.16 | グラフ学習結果の混合行列 | 86 |
| 5.17 | LCFIPlus と GAT の b-tag に関する比較 | 87 |
| 5.18 | LCFIPlus と GAT の c-tag に関する比較 | 88 |
| A.1 | 飛跡の xy 平面への射影 | 95 |
| A.2 | 飛跡の sz 平面への射影 | 96 |

表目次

| | | |
|------|---|----|
| 1.1 | $m_h = 125 \text{ GeV}$ における SM ヒッグス粒子の崩壊分岐比 | 18 |
| 1.2 | ジェットを占める各粒子と対応する検出器 [3] | 23 |
| 2.1 | シリコンセンサーの仕様 | 37 |
| 2.2 | 物質量の大きい吸収層の候補物質 (λ は相互作用長、 L_R は放射長、 R_M はモリエール半径を示す。) | 40 |
| 3.1 | H2A ビームラインにおけるビームパラメータ | 43 |
| 3.2 | SiW-ECAL のレイヤー構成 (0 層目がビーム上流側) | 45 |
| 3.3 | ヒットマップの作成において使用した run の詳細 | 47 |
| 5.1 | シミュレーションデータのパラメータ | 68 |
| 5.2 | LCFIPlus におけるフレーバー識別の入力変数 | 70 |
| 5.3 | ディープニューラルネットワークにおけるハイパーパラメータ | 72 |
| 5.4 | 最適化を行ったハイパーパラメータの一部とその値の範囲。 | 73 |
| 5.5 | 識別効率 (Tagging Efficiency) が 0.8 の時のジェット誤認率 (Mis-id fraction) | 77 |
| 5.6 | 各ノード (飛跡) が持つ特徴量。詳細については Appendix を参照。 | 79 |
| 5.7 | ノード分類の答えラベル | 80 |
| 5.8 | リンク予測の答えラベル | 80 |
| 5.9 | グラフ分類の答えラベル | 80 |
| 5.10 | グラフニューラルネットワークにおけるハイパーパラメータ | 83 |
| 5.11 | 最適化を行ったハイパーパラメータとその値の範囲 | 84 |
| 5.12 | 識別効率 (Tagging Efficiency) に対するジェット誤認率 (Mis-id fraction) | 88 |

第 1 章

序論

本章では、はじめに 1.1 節で素粒子とそれらに働く相互作用を説明する標準模型 (The Standard Model; SM) について述べる。そして 1.2 節にて将来の電子陽電子ヒッグスファクターである、国際リニアコライダー計画 (International Linear Collider; ILC) の概要に触れたのち、1.3 節で ILC が探索する物理について、特にヒッグス粒子に関係した事項を中心に述べる。ILC の実現において必要な ILC の検出器について 1.4 節で、ソフトウェアについて 1.5 節でまとめたのち、1.6 節で本研究の目的を述べる。

1.1 素粒子物理学

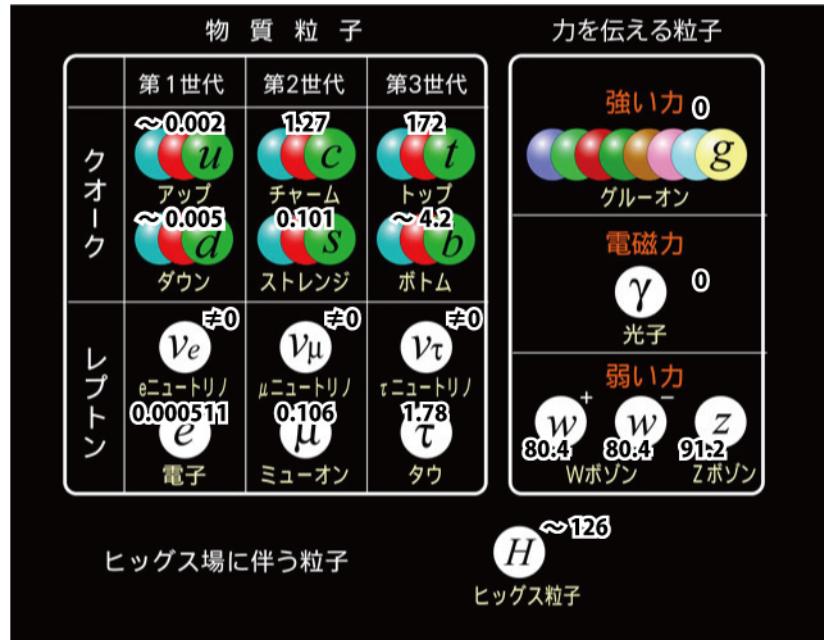
1.1.1 標準模型

素粒子とは、物質を構成している究極要素をさす名称である。そして素粒子物理学は、それら構成要素とその間に働く相互作用の性質を解明する学問である。現代の素粒子物理学では、すべての現象を説明するための基本的な枠組みとして図 1.1 のような標準模型を掲げており、これは現時点の実験データと高い精度で一致することが確認されている。

標準模型は、主に次に挙げる 2 つの基本的な前提に沿って記述されている。1 つ目に、物質の究極要素である素粒子はクォークとレプトンというスピン $1/2$ のフェルミオンである。2 つ目に、素粒子の相互作用はゲージ理論によって記述され、標準模型における相互作用は電磁相互作用・弱い相互作用・強い相互作用の 3 つである。

物質の化学的性質を失わない最小単位は分子であり、分子はさらに原子の組み合わせによって構成されている。そして原子は原子核と電子によって構成されており、原子核は陽子と中性子のような核子からなっている。この核子を構成するものがクォークであり、標準模型においては 6 種類存在する。一方で、電子のような強い相互作用をしないものをレプトンと呼び、同様に 6 種類存在する。クォーク・レプトンとともに 3 つの世代と 2 つの電荷タイプをもっており、世代の大きい粒子ほど重いため弱い相互作用により小さい世代のクォーク・レプトンへと崩壊する。

場の量子論では量子場 ϕ が素粒子と関連して記述されており、ラグランジアンによって相

図 1.1: 素粒子の標準模型 (数値は質量 [GeV/c²])

互作用の性質が記述される。ラグランジアンが量子場の局所的なゲージ対称性のもとで不变であると仮定すると、ゲージ場と呼ばれるベクトル場が現れるが、このゲージ場と量子場の積によって相互作用を表す理論をゲージ理論という。ゲージ粒子はこのゲージ場が粒子として現れたもので、素粒子の相互作用を媒介するとされているスピン 1 のゲージ粒子には、グルーオン・光子・W ボソン・Z ボソンの 4 種類がある。クォークとグルーオンの相互作用である強い相互作用は、量子色力学に基づき $SU(3)$ 対称性をもつ。また荷電粒子と光子の相互作用である電磁相互作用と W・Z ボソンを介する弱い相互作用は、グラショウ=ワインバーグ=サラム理論 (GWS 理論) [4] によって統一され電弱相互作用と呼ばれており、 $SU(2) \times U(1)$ 対称性をもつ。これに加えて重力相互作用が存在するが、他の 3 つの相互作用と比較して非常に弱いため標準模型では扱われない。

本論文のテーマである電子・陽電子はレプトンに分類される素粒子であり、粒子・反粒子の関係にある。反粒子とはある粒子と符号の異なる電荷を持つが、質量やスピン角運動量など電荷以外は同じ特性を持つ粒子を指しており、陽電子は電子の反粒子である。粒子・反粒子の関係にある粒子はその質量エネルギーを放出して対消滅を起こす性質があり、ヒッグスファクターにおいてもこの現象によって別の素粒子を生成する。

また、標準模型は実験データと高い精度で一致しているが、標準模型だけでは説明しきれない物理も数多く存在しており、標準理論を越えた新しい物理 (Beyond the Standard Model; BSM) を検証する実験が世界中で行われている。

1.1.2 ヒッグス機構

GWS理論においてゲージ粒子はゲージ対称性により質量項が禁止されているが、先述の $W \cdot Z$ ボソンはそれぞれ $80.4 \text{ GeV}/c^2$ 、 $91.2 \text{ GeV}/c^2$ の質量を持っている。標準模型ではこれを説明するためにヒッグス機構を導入し、ゲージ対称性が自発的に破れることで質量を獲得している。このヒッグス機構によると、宇宙の膨張によって真空は冷却されてヒッグスが凝縮した状態に相転移が起き、この相転移によって対称性が破れる。そのため真空に複素2次元のスカラー場としてヒッグス場 Φ を導入し、これとゲージ場との相互作用によって質量を持つとしている。またヒッグス場の存在と同時に、対応する粒子としてヒッグス粒子の存在が必要となる。相転移後のヒッグス場のポテンシャル [5] は

$$V(\phi) = \mu^2(\Phi^\dagger\Phi) + \lambda(\Phi^\dagger\Phi)^2 \quad (\text{ただし } \mu^2 < 0) \quad (1.1)$$

と書くことができ、ポテンシャルの概形は図 1.2 のような形状になる。

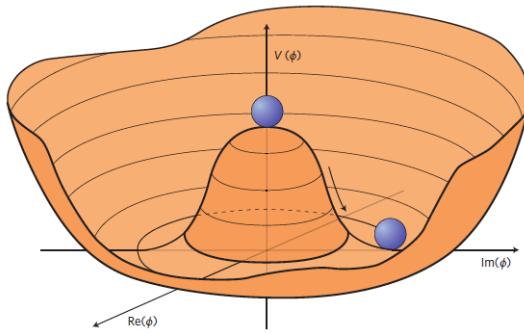


図 1.2: ヒッグスボテンシャル [6]

図 1.2 のようなポテンシャルでは、 $V(\phi)$ 軸周りに真空がリング状に縮退している。また真空のポテンシャルは最小値をとる点で安定するが、ヒッグスボテンシャルにおいては一つの点をとると位相回転対称性が破れてしまう。このときヒッグス場に、

$$\langle \Phi \rangle = \frac{v}{2} = \sqrt{\frac{-\mu^2}{2\lambda}} \quad (1.2)$$

の有限の真空期待値が現れる。これによってゲージ粒子が質量を獲得する。より直感的には、真空中に凝縮されたヒッグス場の中でゲージ粒子を加速しようとした時に、ヒッグス場から抵抗を受ける。この抵抗は、ゲージ場が1個のヒッグス粒子と衝突する頻度を意味する結合定数と、真空のヒッグスの密度に比例することから、質量は加速されにくさを表す量と考えることができる。

このヒッグス粒子は、2012年7月に欧洲原子核研究機構 (Conseil Européen pour la Recherche Nucléaire; CERN) の大型ハドロン衝突型加速器 (Large Hadron Collider; LHC) における ATLAS、CMS 実験によって発見され、理論と実験との一致が確認された。本論文

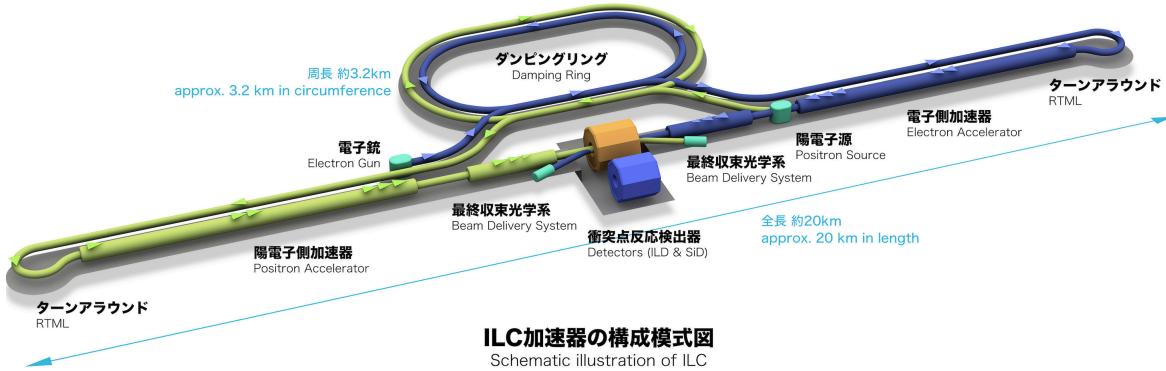


図 1.3: ILC の概略図

のテーマであるヒッグスファクトリーでは、このヒッグス粒子を大量に生成し詳細に研究することを最大の目的としている。

1.2 国際リニアコライダー計画; ILC

国際リニアコライダー (International Linear Collider; ILC) は、東北地方北上山地が有力な建設候補地となっている電子陽電子衝突型線形加速器であり、将来のヒッグスファクトリーとしての稼働を期待されている。全長 20 km の線形加速器を用いて電子と陽電子を加速し、中央の Interaction Point (IP) で衝突させることで様々な粒子を生成し、これを解析することでヒッグス粒子を始めとする新物理を探索することを目的としている（図 1.3）。また ILC は重心系エネルギー $\sqrt{s} = 250 \text{ GeV}$ での運転開始を予定しているが、線形加速部の延長、あるいは技術革新による加速勾配の向上によって 1 TeV を超えるエネルギーまでのアップグレードも可能になっており、各エネルギーにおける物理プログラムのメインターゲットは次のようにになっている。

- $\sqrt{s} = 250 \text{ GeV}$: Zh 随伴生成過程を用いたヒッグス精密測定
- $\sqrt{s} = 350 \text{ GeV}$: $t\bar{t}$ 対生成、 WW 融合過程によるヒッグス生成
- $\sqrt{s} = 550 \text{ GeV}$: ヒッグスの自己結合とトップ湯川結合の測定、高統計によるヒッグス精密測定、新物理の直接探索
- $\sqrt{s} = 1 \text{ TeV}$: ヒッグスの自己結合とトップ湯川結合の精密測定、新物理の直接探索

ヒッグス粒子を発見した LHC と比較して ILC には以下の 3 つの利点が存在する。1 つ目は LHC が複合粒子であるハドロンのコライダーであるのに対して、ILC はレプトンコライダーである点である。ILC では背景事象が少ないクリーンな環境でトリガーレスでの運転が可能であり、ヒッグス粒子を始めとした網羅的な新物理探索が可能になっている。更にバックグラウンドが小さいため崩壊点検出器をビームからおよそ 15 mm と近い距離におくことができ、フ

レーバーの識別において b フレーバのみでなく c フレーバの識別も可能となる。また、LHC では断面積を計算する上で QCD に基づく系統的な不確定性が存在するが、ILC では電弱相互作用のみについて考えることができるために、高精度な理論検証が可能になる。2つ目は加速粒子である電子陽電子が粒子反粒子の関係にある点である。粒子反粒子が対消滅するため、全エネルギーを目的粒子の生成に効率的に用いることが可能である。3つ目はビーム偏極が全エネルギーで可能となっている点である。ビーム偏極とは進行方向に対するスピンを意味しており、これによって測定可能な物理量を増やすことができる。

1.3 ILC の物理

1.3.1 ヒッグス生成過程と質量精密測定

1.1 節で述べた通り、ILC はヒッグスファクトリーとしての役割を期待されている。ヒッグスファクトリーでは、ヒッグス粒子を大量に生成し崩壊過程を精密測定することで、他の粒子との結合定数を測定し標準模型を検証することができる。ILC におけるヒッグス粒子は図 1.4 のようなファインマン図で表される過程によって生成され、その生成断面積は図 1.5 のようになっている。運転開始で予定している $\sqrt{s} = 250 \text{ GeV}$ 付近では、主に $Z h$ 随伴生成過程の断面積が最大となり、この $Z h$ 随伴生成過程では、反跳粒子である Z ボソンを正確に再構成することで、ヒッグス粒子の質量を

$$M_{recoil}^2 = (\sqrt{s} - E_{ff})^2 - |\vec{p}_{ff}|^2 \quad (1.3)$$

によって、高い精度で求めることができる。 $(E_{ff} \text{ は } Z \text{ ボソンが崩壊するフェルミオン対のエネルギーを、} \vec{p}_{ff} \text{ は運動量を表す。})$

LHC のような複合粒子同士の衝突では、始状態の運動量が定まっていないことに加えバックグラウンドとなる事象も多い。しかし ILC の反応過程では、始状態の運動量がわかっているため、反跳粒子を正確に再構成することでヒッグス粒子の崩壊モードに依存せず全断面積の測定が可能となっている。このようヒッグス粒子の崩壊モードに依存しない測定によってヒッグス粒子の invisible 崩壊や、ヒッグス粒子が標準模型を超える粒子に崩壊する exotic 崩壊の存在を検証することが出来る。これにはヒッグス粒子が暗黒物質や後述する超対称性粒子に崩壊するモードなどが含まれており、標準模型を超える新物理の検証を行うことが可能である。

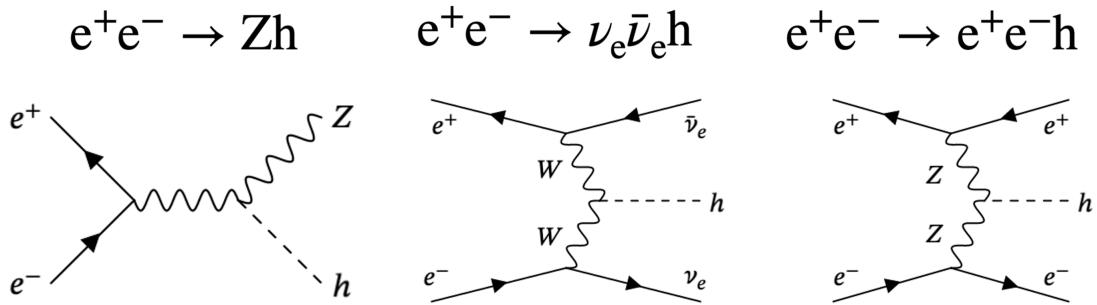


図 1.4: Zh 随伴生成、 WW 融合反応、 ZZ 融合反応におけるファインマンダイアグラム [7]。

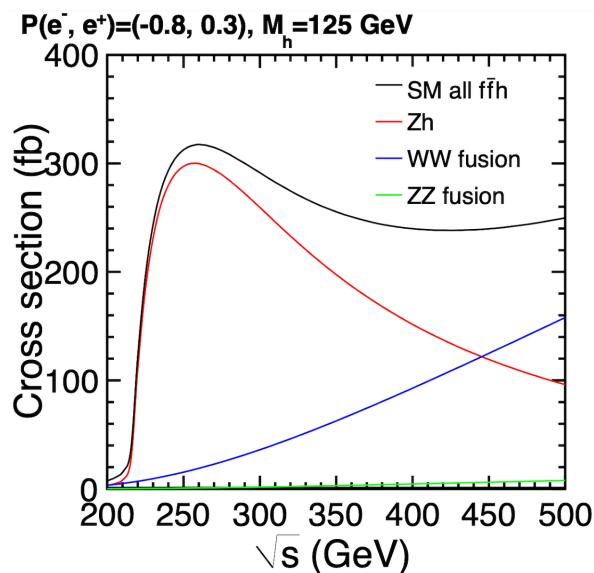


図 1.5: ILC の重心系エネルギーに対するヒッグス生成断面積。ヒッグス粒子の質量が $m_h = 125$ GeV であるとして、 Zh 随伴生成、 WW 融合反応、 ZZ 融合反応をそれぞれ赤、青、緑線で示している [7]。

1.3.2 ヒッグス結合定数の精密測定

電子陽電子衝突によって生成されるヒッグス粒子は不安定であるため、より質量の小さい粒子のペアに崩壊する。標準模型におけるヒッグス粒子の各質量に対する崩壊分岐比の割合を図 1.6 に、 $m_h = 125$ GeV における崩壊分岐比を表 1.1 に示す。ILC ではヒッグス粒子の生成断面積と全崩壊幅を精密に測定することができるため、ヒッグス粒子の崩壊分岐比を精密に決定することができる。この崩壊分岐比はヒッグス粒子の結合定数の 2 乗に比例しており、標準模型を超える新物理のシナリオにおいてヒッグス粒子の結合定数に生じるズレを検証することができる。

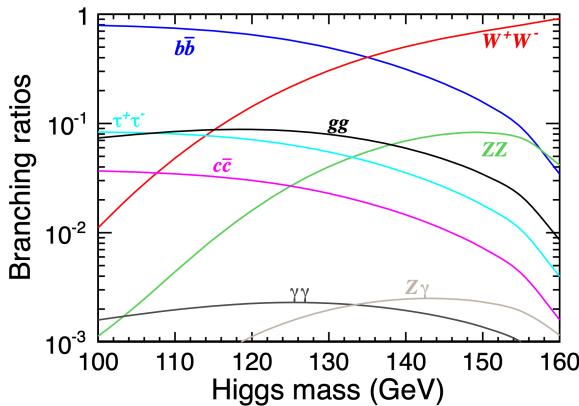


図 1.6: 標準模型におけるヒッグス粒子の質量と崩壊分岐比の関係

| 崩壊モード | 崩壊分岐比 |
|----------------|--------|
| $b\bar{b}$ | 58.1 % |
| WW | 21.5 % |
| gg | 8.2 % |
| $\tau^+\tau^-$ | 6.3 % |
| $c\bar{c}$ | 2.9 % |
| ZZ | 2.6 % |
| $\gamma\gamma$ | 0.2 % |

表 1.1: $m_h = 125 \text{ GeV}$ における SM ヒッグス粒子の崩壊分岐比

以下では、結合定数の精密測定によって検証可能な新物理について述べる。図 1.7 は新物理の各シナリオにおける、標準模型ヒッグス粒子との結合定数のズレを表す。

- 超対称性理論

超対称性理論 (Supersymmetry; SUSY) は、フェルミオンとボソンを交換する変換に対する不变性（超対称性）を定義する理論である。この理論においては、標準模型におけるすべての粒子に対してスピンが $1/2$ 異なる超対称性パートナーが導入される。超対称性が完全である場合、標準模型粒子と同じ相互作用を持ち、後述の階層性問題などを解決するには TeV スケールで対称性が破れ、標準模型のヒッグス粒子と異なる質量を持つことが期待されているが、現段階では SUSY 粒子は発見に至っていない。SUSY では標準模型以外のヒッグス粒子が存在し、その影響によって図 1.7 のように主にヒッグス粒子する b クォークと τ レプトンとの結合が大きくなる。

- 複合ヒッグス模型

ヒッグス粒子は TeV スケールにおいて複合粒子のように振る舞い、その内部により基本的な粒子を持っているとするモデルが複合ヒッグス模型である。この模型では Compositeness scale によって抑制される高次の作用素により、標準模型と比べて結合定数が大きくずれてしまう。Compositeness scale を f とすると、ヒッグス粒子 h のゲージボソンやフェルミオン xx への結合 g_{hxx} は、

$$\frac{g_{hxx}}{g_{h_{SM}xx}} \simeq 1 \pm \mathcal{O}(v^2/f^2) \quad (1.4)$$

のオーダーで表される。図 1.7 において複合ヒッグス模型では、標準模型に比べて粒子との結合が小さくなっている。

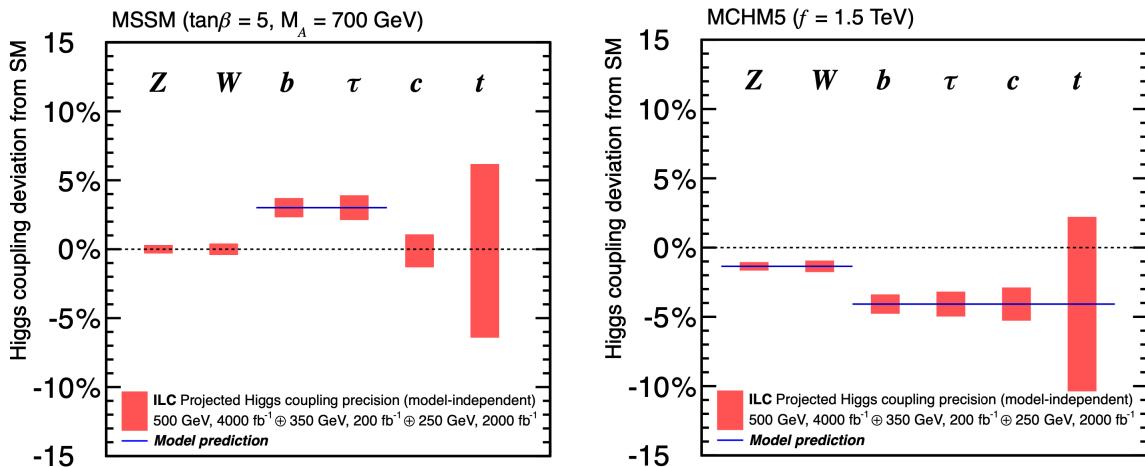


図 1.7: 2 つの新物理のモデルにおける、標準模型のヒッグス粒子結合定数とのずれ。 (左) 超対称性 (SUSY) モデル (右) 複合ヒッグスモデル。誤差棒は ILC (250 ~ 500 GeV) での測定精度 (1σ) の範囲を表している。 [8]

ヒッグスの崩壊分岐比の測定精度は信号事象数を S 、背景事象数を N とすると $S/\sqrt{S+N}$ となり、背景事象の影響を十分低減させることができた場合には不定性を 1% 以下まで下げることができる。図 1.8 では LHC と ILC におけるヒッグス粒子の結合定数の測定精度を表しており、ILC では LHC に比べておよそ 1 桁優れた精度で測定することができる。これを達成するために、高い検出器性能と精度の高い事象再構成・解析手法が求められている。

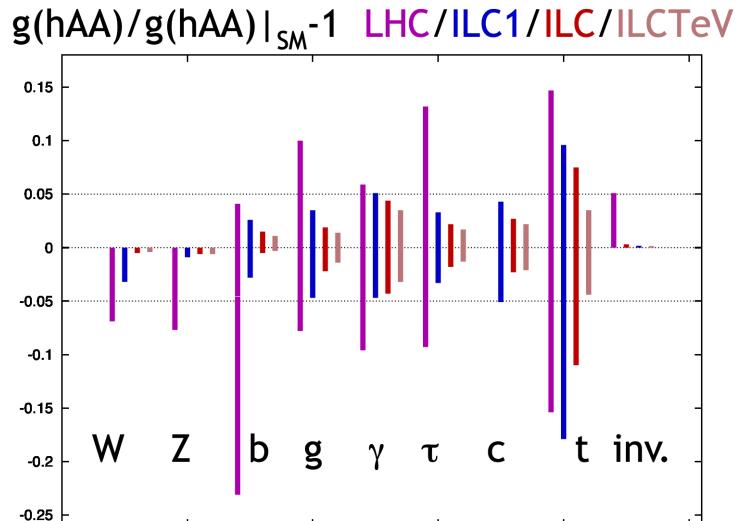


図 1.8: ILC と HL-LHC におけるヒッグス粒子の各粒子に対する結合定数の測定精度予測。誤差棒は 1σ の範囲を表している。(紫 : LHC (300 fb^{-1}) , 青 : ILC1 (250 GeV) , 薄橙 : ILCTeV (1 TeV)) [8]

1.3.3 ヒッグス自己結合

500 GeV 以上の ILC では図 1.9 のような、 $e^+e^- \rightarrow Zhh$ 反応の断面積を測定することで、自己結合について探索することができる。ゲージ不变性によりヒッグスの三点結合は、四点結合における真空凝縮としてのみ起こることができるため、三点結合を確認することでヒッグス場の真空凝縮について検証することができる。ILC500 GeV における図 1.9 の断面積は 0.2 fb と小さく高統計を得ることが難しいため、崩壊 $e^+e^- \rightarrow Zhh \rightarrow q\bar{q}bbb\bar{b}$ における b フレーバーの識別精度が非常に重要になる。

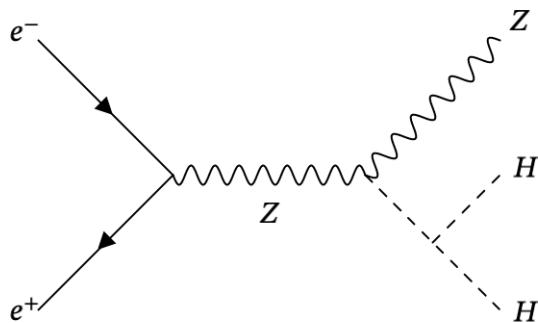


図 1.9: ヒッグス自己結合 $e^+e^- \rightarrow Zhh$

1.3.4 階層性問題

ヒッグス粒子の質量は LHC によって $125 \text{ GeV}/c^2$ と測定されている。しかしヒッグス粒子の質量は、図 1.10 のような高次ダイアグラムからプランクスケール (10^{19} GeV) 程度の質量補正を受けることが分かっている。そのため標準模型を超える新物理による質量の量子補正をキャンセルする解決策がなければ、量子補正を受ける前の質量がプランクスケール程度の質量であり、それに対して偶然プランクスケール程度の負の量子補正がかかって $125 \text{ GeV}/c^2$ の質量を再現しているということになってしまう（ファインチューニング）。これを回避するために、以下に挙げるような TeV スケールの超対称性理論や余剰次元理論など新物理によるシナリオが提案されている。これらのシナリオにおけるヒッグス粒子との結合定数は標準模型における予測からズレることとなるため、ILC においてヒッグス粒子の精密測定を行うことで崩壊分岐比を決定することの意義は大きいと言える。

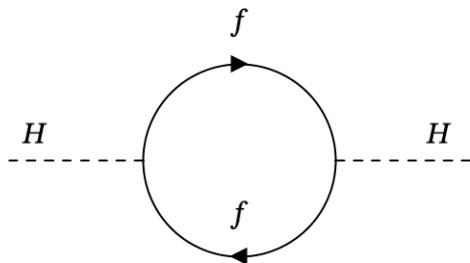


図 1.10: ヒッグス粒子の質量補正となるフェルミオンループ

- 超対称性理論

階層性問題においては、超対称性によりヒッグス粒子の質量補正に関する 2 次の発散を超対称性パートナーの寄与で打ち消す。これによって、対数による発散に落とすことができる。

- 余剰次元理論

余剰次元理論とは、四次元時空以外にも次元があるとする理論である。この理論では空間の次元数を増やすことで、増えた次元のゲージ場にヒッグス場の起源を求める。この場合にはゲージ不变性により、繰り込みの発散が現れないため階層性問題に対応できる。

1.3.5 その他の新物理

上にあげた階層性問題に関する物理に加え、トップクォークの質量の精密測定、電弱相互作用の精密検証が可能であり、ILC の実現やそのアップグレードを通して宇宙の謎に迫る大発見を期待できる。

1.4 ILC の検出器

ILC の検出器(図 1.11)には、日欧が中心となって開発が進められている International Large Detector (ILD) と、米国が中心となって開発が進められている Silicon Detector (SiD) の二つのコンセプトが提案されており、ILC ではこれら 2 つの検出器が IP を共有できるように push-pull 方式を採用する予定である。また ILD、SiD ともに Particle Flow Algorithm (PFA) という事象再構成アルゴリズムに沿って最適化されている。

1.4.1 ILD で検出器可能な粒子群

ILC の電子陽電子衝突で生じる粒子は、ヒッグス粒子などの重いボソンを介した後に、クォークやグルーオン、荷電レプトン、光子に崩壊する。この中でもクォークとグルーオンは

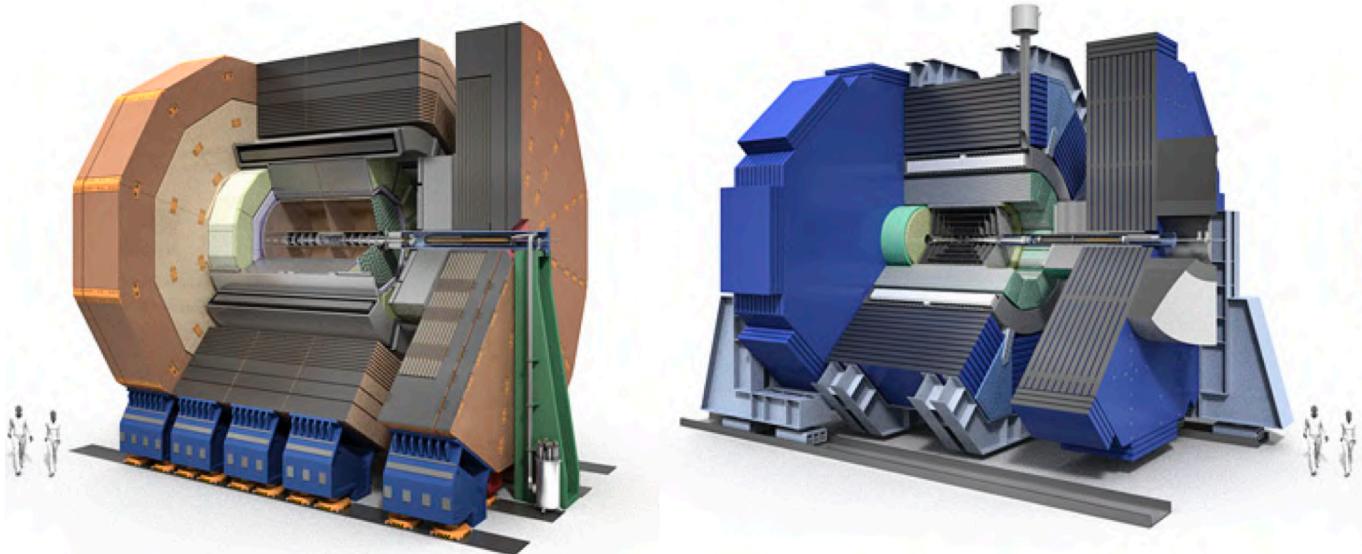


図 1.11: (左) ILD (右) SiD の全体図 [7]

QCD における閉じ込めにより単体で存在することができず、多数のハドロン（強い相互作用で結合した複合粒子）を発生させる。また、レプトンのうちニュートリノは物質とほとんど相互作用を起こさず、測定器を通り抜けてしまうため、直接検出することは出来ない。そのため実際に測定器で検出することができるのは、終状態としてハドロンが束になったハドロンジェット、荷電レプトン、光子となる。

1.4.2 Particle Flow Algorithm (PFA)

前節の通り ILC で生成されるクォークやグルーオンは、ジェットの終状態として検出される。そのため ILC の物理を探求する上で必要となる粒子識別や事象再構成において、ジェットのエネルギーは非常に重要な情報であり、一般的にジェットのエネルギー分解能は

$$\frac{\sigma_E}{E} = \frac{a}{\sqrt{E}} \oplus \frac{b}{E} \oplus c \quad (1.5)$$

として表される。ここで、 a, b, c を係数に E はエネルギーを、 \oplus は 2 乗和を表しており、第一項はカロリメータに由来する統計項を、第二項は補正項、第三項は定数項となっている。従来の素粒子実験におけるエネルギー測定では、およそ 7 割に相当する粒子がハドロンカロリメータでエネルギーを測定されるが、ハドロンカロリメータはそれ以外の検出器と比較してエネルギー分解能が低く、およそ $\sigma_E/E = 55\%/\sqrt{E(\text{GeV})}$ の分解能となっている。ILC ではジェットエネルギー分解能 $\sigma_E/E = 30\%/\sqrt{E(\text{GeV})}$ を目指しており、これを達成するため導入されているアルゴリズムが Particle Flow Algorithm [9] である。PFA はジェット内の粒子をその種類ごとに最適な検出器でエネルギー測定を行うことでジェットエネルギー分解能を向上させる再構成手法であり、過去の実験結果から、ジェット中に含まれる主な粒子の種類の典型的な割合 ($Z \rightarrow q\bar{q}$ の場合) とそれに対応する検出器について以下のように分かってい

る（表 1.2）。

| 粒子のタイプ | 検出器 | ジェット中のエネルギー割合 |
|--------|-------|---------------|
| 荷電粒子 | 飛跡検出器 | 62% |
| 光子 | ECAL | 27% |
| 中性ハドロン | HCAL | 10% |
| ニュートリノ | - | 1.5% |

表 1.2: ジェットを占める各粒子と対応する検出器 [3]

1.4.3 International Large Detector: ILD

ILD [7] は内側から順に崩壊点検出器、飛跡検出器、電磁カロリメータ、ハドロンカロリメータ、ミューオン検出器で構成されている。カロリメータとミューオン検出器の間には 3.5 テスラのソレノイドコイルが設置されている。図 1.12 に断面図を示す。

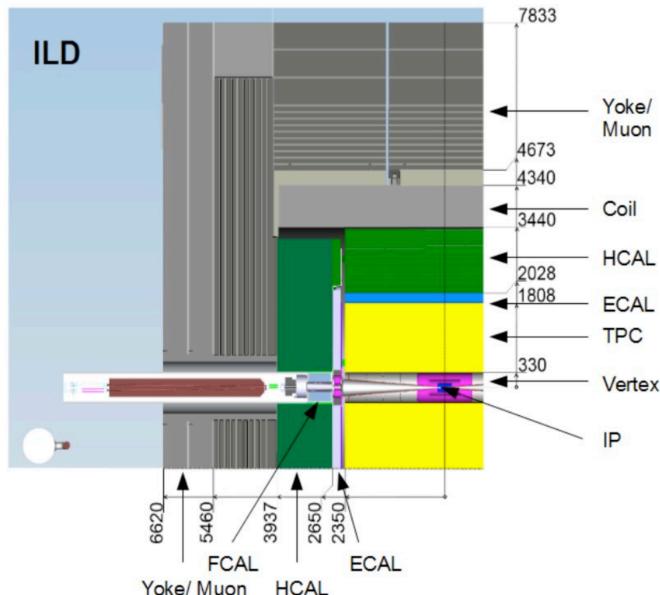


図 1.12: ILD の断面図 [7]

崩壊点検出器

崩壊点検出器は IP に最も近い場所に置かれる検出器であり、図 1.13 のようにシリコンピクセルセンサーが両面に貼られた層を ILD の半径方向に 3 層重ねた構造になっている。このシリコンピクセルセンサーで飛跡を高い位置分解能のもと検出することで、荷電粒子の生成点を高い精度で決定することができる。これによって短寿命粒子の崩壊点を高精度に再構成することができ、再構成された二次崩壊点はジェットのクォークフレーバーを識別する上で非常に重

重要な情報となる。ILC では飛跡検出における位置分解能 σ (式 1.6) を目標としており、そのために CMOS センサー、DEPFET、Fine Pixel CCD、SOI など様々な技術候補 [7] が研究されている。

$$\sigma[\mu\text{m}] = 5 \oplus \frac{10}{p[\text{GeV}/c] \sin^{3/2} \theta} \quad (1.6)$$

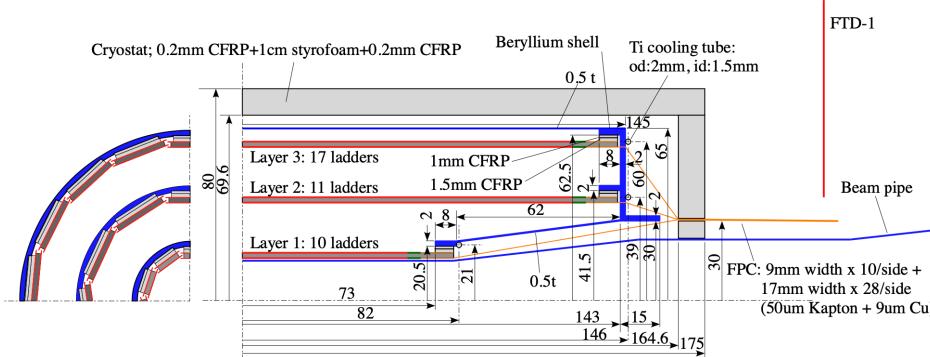


図 1.13: ILD 崩壊点検出器の構造 [7]

中央飛跡検出器

中央飛跡検出器は崩壊点検出器の外側に位置しており、Time Projection Chamber (TPC) とその周囲に設置されるシリコン検出器のハイブリッドで構成されている。TPC は大型のガスチェンバーであり、荷電粒子の通過でガス内に生じる電離電子を電極間にかけられた電場によってドリフトし、ドリフト時間などの情報をもとに飛跡を 3 次元的に再構成する検出器である。荷電粒子の飛跡を再構成することで運動量の測定が可能であり、粒子の飛跡を連続的に検出できることから高い運動量分解能を持つ。さらに信号の大きさからエネルギー損失も測定することが可能であり、これは粒子識別において重要な役割を果たす。また TPC はドリフト時間が長いため時間分解能を上げるために、検出器近くにシリコン飛跡検出器を設置してビームバンチの特定の補助を行っている。

カロリメータ

カロリメータは入射粒子のエネルギーを測定するための検出器で、ILD では内側から電磁カロリメータ (ECAL)、ハドロンカロリメータ (HCAL) によって構成されており、またビーム軸方向に対して前方カロリメータ (FCAL) が設置される。これら ILD のカロリメータにはサンプリング型カロリメータが提案されており、シャワーを起こすための吸収層と生成されたシャワー内の粒子のエネルギーを測定する検出層が交互に組み合わさった構造となっている。

ECAL は主に電磁シャワー内の光子のエネルギーを測定するために利用される。ILD では PFA に最適化するためジェット内の粒子を分離できる高精細なカロリメータが必要とされて

おり、吸収層には物質量が大きいため放射長が短く、モリエール半径の小さいタングステンが検討されている。また、検出層には読み出しセルが高精細なシリコン検出器を用いるシリコン電磁カロリメータ (SiECAL) やシンチレータストリップを用いるシンチレータカロリメータ (ScECAL) が提案されている。

HCAL は荷電ハドロンと中性ハドロンのエネルギー損失を分離し、中性ハドロンのエネルギーを測定するための検出器である。HCAL ではハドロンとの相互作用を起こすための吸収層に鉄が用いられ、検出層には 3 cm 角のシンチレータタイルを用いてシンチレーション光を検出するアナログカロリメータ (AHCAL) と、ガラス抵抗板チェンバー (RPC) を用いて、1 cm 角 2bit の信号を読み出すセミデジタルカロリメータ (SDHCAL) の 2 つが提案されている。

ミューオン検出器

ミューオン検出器はその名の通りミューオンを検出する検出器である。ミューオンは物質中で電磁シャワーやハドロンシャワーをほとんど生成しないため貫通力が大きく、IP から遠い検出器の最も外側に設置されており、RPC と SiPM シンチレータストリップの両方が検討されている。

1.5 ILC のソフトウェア

1.5.1 イベントジェネレータと検出器シミュレーション

ILC をはじめとする線型加速器には「iLCSOFT[10]」というソフトウェアフレームワークが開発されており、検出器シミュレーションから事象再構成までを実行することができる。iLCSOFT 内では専用の LCIO フォーマットを使用し、C++ アプリケーションフレームワークである Marlin[11] によって運用され、検出器のジオメトリなどの記述には DD4hep[12] というツールキットを使用するという点で統一されている。

ILC は将来実験計画であるため、現在実験データは存在しないがシミュレーションによって検出器応答や新物理探索など研究が可能であり、本論文における研究で扱うデータもモンテカルロ (Monte Carlo; MC) 法に基づくソフトウェア群を用いて生成したシミュレーションデータである。図 1.14 に ILC におけるソフトウェアの流れを示す。シミュレーションではまず、Whizard[13] というイベントジェネレータを用いて、標準模型や様々な理論を背景とした物理事象を生成する。Whizard では終状態で最大 8 粒子までの事象を生成し、Pythia[14] によって粒子の崩壊過程のシミュレーションを行う。更に生成されたイベントに対して、DDSim という Geant4[15] をベースとした検出器シミュレーションを実行し、粒子から検出器ヒットデータが生成される。

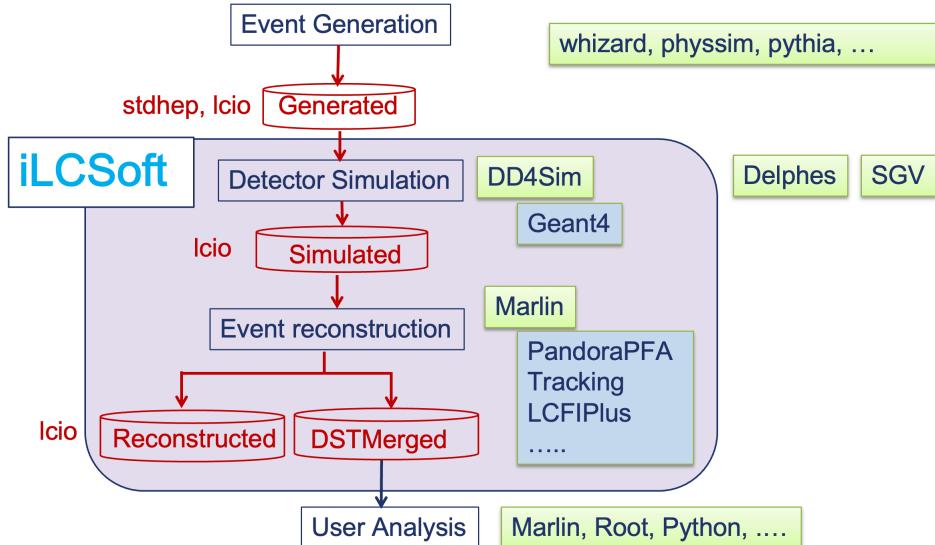


図 1.14: ILC におけるシミュレーションと事象再構成、物理解析の流れとソフトウェア [1]

1.5.2 事象再構成

前節までで生成された検出器ヒットをもとに、終状態粒子のエネルギーや飛跡を推定する事象再構成が行われる。iLCSoft では、Marlin によって測定器出力のデジタル化や飛跡再構成、PFA を行ったのちにジェットの再構成を実行し、物理解析へと繋げていく。iLCSoftにおいて、ジェットの再構成にあたる崩壊点検出からフレーバー識別のプロセスは、LCFIPlus [16] というフレームワークで実行することができる。

飛跡再構成

多数の飛跡を含むジェットは各検出器を通過するため、検出器ヒットをもとにフィッティングを行うことで飛跡は再構成することができる。特に崩壊点検出器では主に飛跡の方向情報を、中央飛跡検出器では運動量や時間情報を取得し、様々なパターン認識アルゴリズムを有する MarlinTrk によって再構成される。

1.5.3 PFA

先述した PFA による再構成は PandoraPFA によって行われる。ここでカロリメータクラスタリングおよび再構成はカロリメータにおけるヒットの情報や TPC における飛跡情報などを元にして行われ、荷電粒子だけでなく中性粒子の情報も用いて以降のプロセスでジェットの再構成を行う。

崩壊点検出

ジェットは IP で生成された粒子が崩壊を繰り返し、そのうち荷電粒子は多くの飛跡を残すことで再構成される。この粒子が崩壊する点を崩壊点 (Vertex) と呼び、特に IP を primary vertex、そこで生成された粒子の二次崩壊点を secondary vertex と呼ぶ。LCFIPplus における崩壊点検出では、主に 2 本以上の飛跡の組み合わせから飛跡の発生源となる点を、 χ^2 値が最小値をとるフィッティングによって求める Vertex Fitter プロセスを用いる。この情報をもとに、primary vertex の再構成を tear-down アルゴリズムを用いて行う。tear-down アルゴリズムでは、全ての飛跡対に対して IP を崩壊点とするフィッティングを Vertex Fitter によって行い、一定の閾値に到達するまで χ^2 値の大きい飛跡から排除していく。そして primary vertex に使用されていない全ての飛跡に対して再びフィッティングを行い、 χ^2 値や不変質量、運動量などを用いたカットや、飛跡同士の関係性の評価、他の飛跡の追加を行い、secondary vertex を再構成している。

ジェットクラスタリング

ILC の終状態の多くは 4 ジェット以上のマルチジェットであり、これらを幾つかのグループに分類（クラスタリング）し、ジェットの元となった粒子のフレーバー識別を行う。そのためジェットクラスタリングのプロセスでは、再構成された崩壊点やレプトンの情報をジェットのコアとし、開き角が 0.2rad にある粒子をジェットのコアに結合する。そして残った粒子に対しては Durham [17] のクラスタリングアルゴリズム

$$Y(i, j) = \frac{2\min(E_i, E_j)^2(1 - \cos\theta_{ij})}{Q^2} + \alpha \quad (1.7)$$

を用いてクラスタリングを行い、値が小さい順にジェットのコアに結合していく。ここで、 E_i, E_j はジェットのエネルギーを、 θ_{ij} はジェットの間の角度を、 Q^2 は重心系エネルギーを、 α はジェットのコアが重なってしまうことを防ぐために用いる定数を表している。このように崩壊点やレプトンの特徴的な物理量や飛跡同士の開き角とエネルギーを用いて、全粒子に対してクラスタリングを行う。

フレーバー識別

崩壊点の情報や飛跡の情報など 20 程度の物理量をもとに、多変量解析によってジェットの親粒子のフレーバー識別が行われる。LCFIPplus では ROOT [18] の TMVA [19] パッケージを使用して、従来の機械学習手法である Boosted Decision Trees (BDTs) を用いた識別が行われている。LCFIPplus では b フレーバーのジェット、 c フレーバーのジェット、 $u \cdot d \cdot s$ フレーバーのジェットの 3 つを分類している。分類においては、 $b \cdot c$ フレーバーのハドロンは衝突点から離れたところで崩壊するという特徴がある。(図 1.15) ハドロンの寿命 τ は光速 c を用いて、 b ハドロンでおよそ $c\tau = 400 \sim 500 \mu\text{m}$ 、 c ハドロンでおよそ $c\tau = 20 \sim 300 \mu\text{m}$ である。上記に加えて b ハドロンから派生した c ハドロンも崩壊するため、1 つのジェット中に崩

壞点を2つもつ。これらの情報やジェットの物理量を用いてフレーバーを識別することができる。そのため崩壊点検出のプロセスにおいて、高精度に崩壊点を求めることが重要となる。

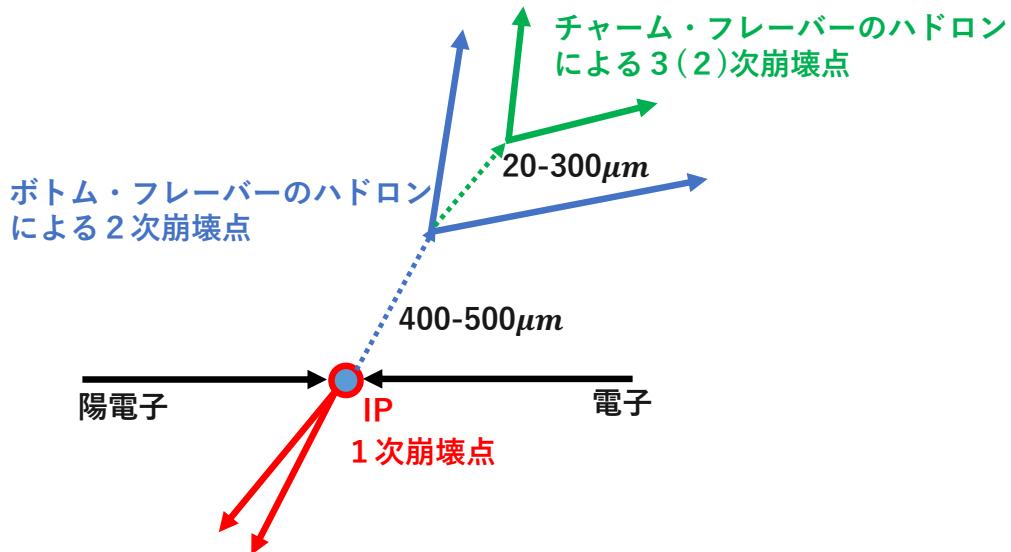


図 1.15: ジェットの崩壊の様子。赤線が1次崩壊点、青線が2次崩壊点、緑線が3次崩壊点を由来とする飛跡を表す。

1.6 本研究の目的

本研究の目的は、電子陽電子ヒッグスファクトリーである ILC におけるジェット測定技術の開発である。ヒッグス粒子や 350 GeV 以上の ILC で探索可能なトップクォークなど、ILC の物理において重要な事象の多くはジェットを含んでおり、ジェットを高い精度で測定することが物理解析の性能に直結する。そのジェット測定技術において、シリコンタングステン電磁カロリメータの性能評価、フレーバー識別アルゴリズムの開発の 2 つのテーマで研究を行った。

1.6.1 高エネルギービームによる SiW-ECAL の性能評価

ILC ではパートンレベルでの再構成が求められ、そのためにはジェットのエネルギーや方向の分解能を高い精度で得ることが重要となる。高いジェットエネルギー分解能を達成するために提案されている PFA では、ジェット中の粒子を分離しクラスタリングが可能になるほ

どに高精細なカロリメータが求められており、日本やフランスのグループによって開発されたシリコンタングステン電磁カロリメータはその有望な候補である。本研究では、その技術プロトタイプに高エネルギービームを照射するビームテスト実験を行い、実験結果から更なる改善に向けたフィードバックを得た。本論文では、2章で背景となる物理や SiW-ECAL の仕組みについて述べ、3章でビームテスト実験について報告し、6章でまとめと今後の方針について論じる。

1.6.2 深層学習を用いたフレーバー識別アルゴリズムの開発

再構成において重要なプロセスとなるフレーバー識別アルゴリズムの開発を、深層学習技術を用いて行った。本研究では、はじめに最もシンプルなネットワークである全結合型ニューラルネットワークによる実装を行った後、フレーバー識別により適したネットワークとして、特にグラフ構造のデータを扱うグラフニューラルネットワークを実装し、従来技術である LCFIPlus と比較した識別精度の向上を目指した。グラフニューラルネットワークでは、対象の特徴量に加え構造のトポロジー情報をデータに含んだ学習を行うことができる。また同時に、全ての事象再構成アルゴリズムを深層学習に置き換えることを最終目標に、フレーバー識別アルゴリズムと崩壊点検出アルゴリズムの統合を試みた。本論文では、4章で深層学習に関する基本的事項について述べ、5章で実際に考案したアルゴリズムについて述べた上で、6章でまとめと今後の方針について論じる。

第2章

シリコンタングステン電磁カロリメータ

本章では、ILD のシリコンタングステン電磁カロリメータについて説明する。まず 2.1 節で検出器を理解する上で必要な粒子と物質の相互作用について述べたのち、2.2 節でカロリメータの検出原理やシリコン検出器の検出原理について述べる。そして 2.3 節で現在の ILD におけるシリコンタングステン電磁カロリメータの読み出し方法、また ASIC の設計性能や読み出し方法、現在の技術プロトタイプについて説明する。

2.1 入射粒子と物質の相互作用

素粒子実験で捉えたい素粒子やハドロンは、粒子と物質との相互作用によって捉えることができる。よって本節では入射粒子の種類ごとに物質との相互作用について述べる。

2.1.1 荷電粒子

荷電粒子のエネルギー損失の要因には、主に電離損失と制動放射が挙げられ [20]、特にエネルギーの低いところでは電離損失の割合が高くなる。（電子ではエネルギーの高いところでは制動放射の割合が高くなる。）以下ではそれについて述べる。

電離損失

荷電粒子は物質を通過することで、物質中の原子を電離あるいは励起させ電離エネルギー損失を生じる。この電離エネルギー損失は原子中の電子によるクーロン散乱によるものが支配的であり、Bethe-Bloch の式に従う。

$$-\frac{dE}{dx} = 4\pi N_A r_e^2 m_e c^2 z^2 \frac{Z}{A} \frac{1}{\beta^2} \left[\frac{1}{2} \ln\left(\frac{2m_e c^2 \beta^2 \gamma^2 W_{\max}}{I^2}\right) - \beta^2 - \frac{\delta(\beta\gamma)}{2} \right] \quad (2.1)$$

| 変数 | 値または単位 |
|---|--|
| N_A : アボガドロ定数 | $6.022 \times 10^{23} \text{ mol}^{-1}$ |
| r_e : 古典電子半径 | 2.817 fm |
| m_e : 荷電粒子の質量 | 0.511 MeV |
| c : 光速 | $2.998 \times 10^8 \text{ m/s}$ |
| z : 入射粒子の電荷 | - |
| v : 入射粒子の速度 | m/s |
| Z : 物質の原子番号 | - |
| A : 物質の相対原子質量 | g mol^{-1} |
| β : 入射粒子の v/c | - |
| γ : $1/\sqrt{1 - \beta^2}$ | - |
| W_{max} : 1回の衝突で物質に与える最大エネルギー | MeV |
| I : 物質の平均イオン化ポテンシャル | eV |
| $\delta(\beta\gamma)$: 密度効果による電離エネルギー損失の補正 | $\sqrt{\rho \langle Z/A \rangle} \times 28.816 \text{ eV}$ |

Bethe-Bloch の式より、電離エネルギー損失 $-dE/dx$ は荷電粒子の入射速度に依存する。様々な物質に対する電離エネルギー損失と入射速度の関係を図 2.1 に示す。入射速度の小さいとき電離エネルギー損失は $1/\beta^2$ に比例しており、 $\beta\gamma \approx 3 \sim 4$ で電離エネルギー損失は最小値に達する。これを最小電離損失といい、この領域にある粒子を MIP (Minimum Ionization Particle) と呼ぶ。

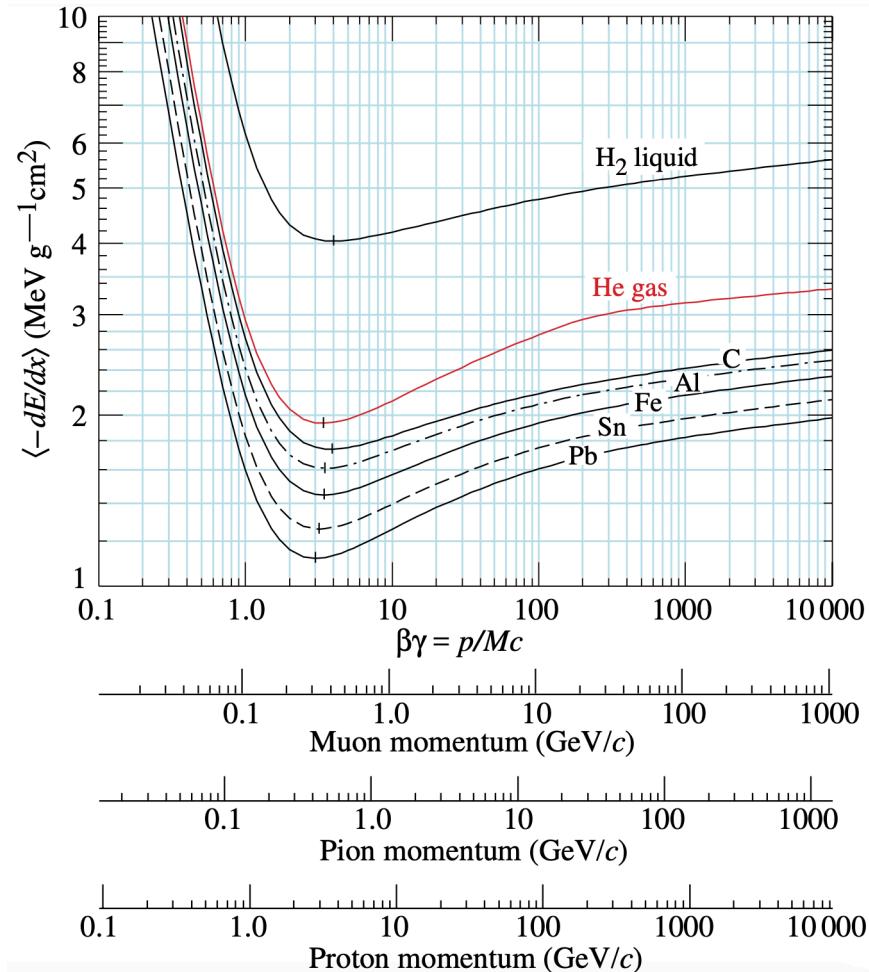


図 2.1: 水素(液体)、ヘリウム(気体)、炭素、アルミニウム、鉄、スズ、および鉛における平均エネルギー損失と、入射粒子(ミューオン、パイ中間子、陽子)の速度の関係。[20]

制動放射

荷電粒子が物質を通過する際には電離の他に、原子核との衝突によって電磁波を放射しエネルギーを失うこともある。物質を構成する原子核はそれぞれ電場を持っており、電場によって Rutherford 散乱を受けた荷電粒子は加速、減速をされ、光子を放射しエネルギーを失う。これを制動放射と呼ぶ。制動放射によって荷電粒子が失うエネルギー損失率は、

$$-\frac{dE}{dx} = \frac{E}{L_R} \quad (2.2)$$

L_R は放射長と呼ばれしており、平均エネルギーが $1/e$ になる平均の長さを指す。 L_R は近似的に以下のように与えられる。

$$\frac{1}{L_R} = 4 \left(\frac{\hbar}{mc} \right)^2 Z(Z+1) \alpha^3 n_\alpha \ln \left(\frac{183}{Z^{1/3}} \right) \quad (2.3)$$

式(2.2)を積分することで、初期エネルギー E_0 を持った荷電粒子が物質を x だけ進むときのエネルギーは以下のようになる。

$$E = E_0 \exp(-x/L_R) \quad (2.4)$$

電離エネルギーと制動放射によって失うエネルギーの大きさが同じになる入射電子のエネルギーを臨界エネルギー E_c と呼び、この値よりもエネルギーが小さい場合はエネルギー損失が Bethe-Bloch の式に従い、大きい場合は制動放射によって主にエネルギーを失う。(臨界エネルギーは物質によって異なるがおおよそ $E_c \simeq 500 \text{ MeV}/Z$ と表される。) 荷電粒子の制動放射によって失われるエネルギーは質量の二乗に反比例するが、電離エネルギー損失は質量に強く依らないため、ほとんどの粒子に対しては制動放射よりも電離損失が支配的となる。一方で電子においては制動放射によるエネルギー損失が支配的であり、 E_c は電磁カロリメータなどの設計において重要なパラメータとなる。

2.1.2 光子

電荷を持たない光子は物質中で電離は起こさず、主に図 2.2 に示す光電効果、コンプトン散乱、電子陽電子対生成の 3 つの過程で相互作用する。

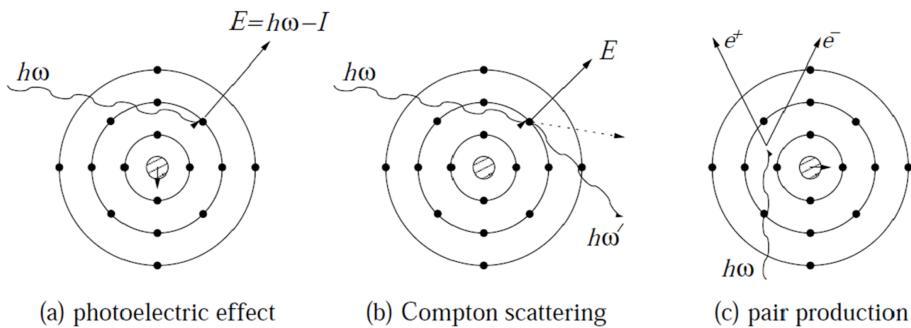


図 2.2: 光子と物質の相互作用

- 光電効果 (Photoelectric effect): 入射光子が物質に当たることで光子の持っていたエネルギー $h\nu$ が物質の電子に与えられる。これによって励起された電子が $h\nu - I$ の運動エネルギーで飛び出す現象。 $(I$ はイオン化エネルギー)
- コンプトン散乱 (Compton scattering): 入射光子と原子核に束縛されている 1 つの電子との弾性散乱。光電効果よりも光子のエネルギーが大きく、電子陽電子対生成反応よりも小さい時に支配的な反応である。
- 電子陽電子対生成 (Pair production): 入射光子が原子核のつくるクーロン場において消滅し、電子陽電子の対を生成する反応。この反応では、光子のエネルギーが電子陽電子の静止質量の和（およそ 1.02 MeV）よりも大きい必要がある。

光子のエネルギーによってこれらの反応確率は異なり、図2.3に光子のエネルギーに対する各反応の確率を示す。中でも数 MeV 以上の光子においては電子陽電子生成反応が主要なプロセスであり、ILC のような高エネルギーにおいては電子陽電子対生成が重要である。物質に入射した光子は電子陽電子を生成し、さらに制動放射によって光子を放出する。これを繰り返すことで電子陽電子と光子の数が指数関数的に増加していく、この現象が電磁シャワーと呼ばれている。この時、電子陽電子対生成過程の断面積は $E_\gamma \gg mc^2/\alpha Z^{1/3}$ において

$$\sigma_{pair} = \frac{7}{9} \frac{1}{n_a L_R} \quad (2.5)$$

と近似することができ、光子の飛程はおよそ $9/7 L_R$ となる。電磁シャワーは発展するにつれて光子の平均エネルギーが下がり、臨界エネルギー（ILC の検出器ではおよそ 10 MeV）に到達すると電子陽電子生成過程が起こらなくなり終息する。粒子のエネルギーを測定する場合には、シャワー内の荷電粒子を MIP とみなしそれらの粒子数が初めの光子のエネルギーに比例すると考えることで、検出器のセンサーに残したエネルギー損失の和をとることで測定する。

また電磁シャワーは進行方向だけでなく横方向にも広がり、モリエール半径 R_M によって広がりが特徴づけられる。モリエール半径とは、エネルギーの 90% が入るシャワーの半径を指し、以下の式で表される。

$$R_M \sim \frac{21(\text{MeV})L_R}{\text{臨界エネルギー (MeV)}} (\text{g/cm}^2) \quad (2.6)$$

2.1.3 ハドロン

π 中間子や K 中間子などのハドロンは物質を構成する原子核と衝突し、非弾性散乱を繰り返すことでハドロンシャワーを生成する。ハドロンの相互作用長は典型的に放射長よりも大きく、ハドロンをカロリメータで測定する場合には非常に多くの物質を必要とする。相互作用長は以下のように表される。

$$\lambda = \frac{A}{N_a \rho} \sigma_{total} \quad (2.7)$$

ここで、 ρ は物質の密度を、 σ_{total} は反応断面積の総和を表す。ハドロンシャワーには $\pi^0 \rightarrow \gamma\gamma$ 崩壊によって発生する電磁シャワーが混ざってしまっており、検出器のエネルギー応答が異なることから、ハドロンシャワーのエネルギー分解能は電磁シャワーと比較して非常に悪くなってしまう。

2.2 粒子検出器の動作原理

素粒子実験では、前節の相互作用を用いて粒子を検出する。粒子の検出には、事象を区別するため十分な時間分解能と位置分解能を持つ必要があり、また各粒子を識別するために、エ

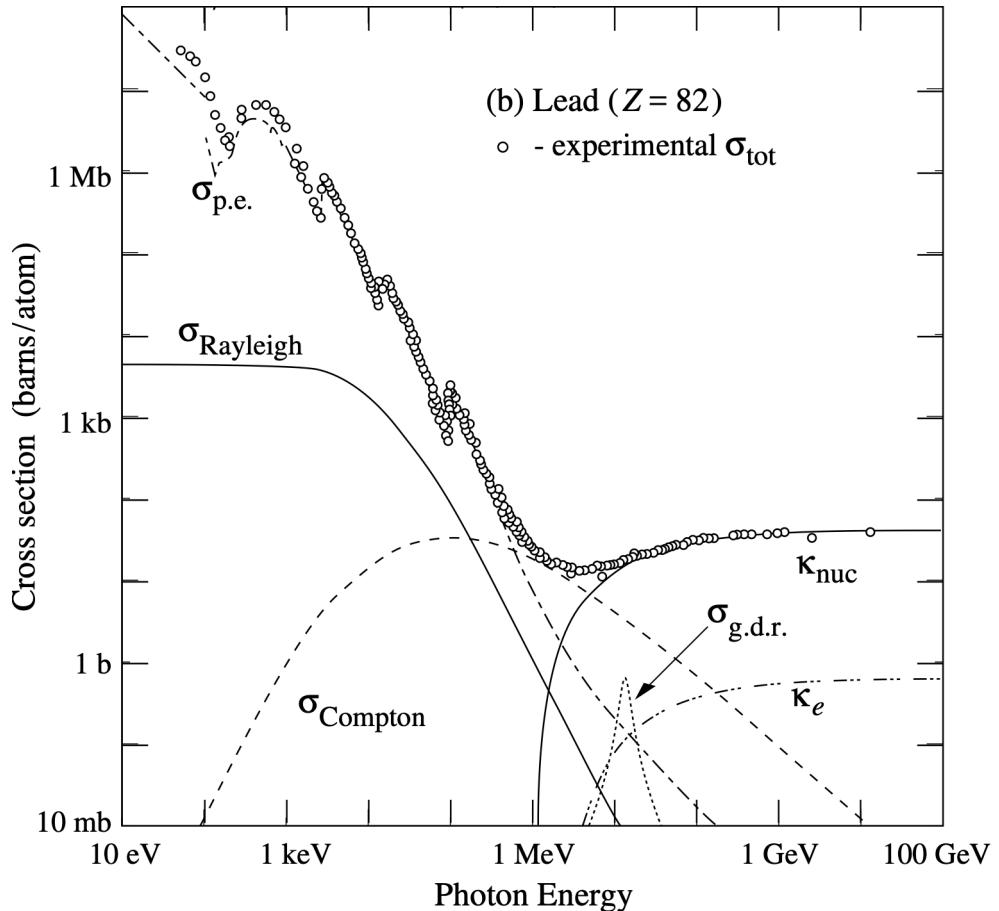


図 2.3: 光子と鉛の相互作用における断面積と光子のエネルギーの関係。 $\sigma_{p.e.}$ は光電効果を、 $\sigma_{Compton}$ はコンプトン散乱を、 κ_e, κ_{nuc} は電子陽電子対生成を指す。[20]

エネルギーと運動量を十分な精度で測定する必要がある。以下では、測定器を構成する検出器のうち特に重要なものを取り上げる。

2.2.1 ガス検出器

ガス検出器は、主にアルゴンのような活性の低いガスを検出器内に充填した検出器である。荷電粒子がガス中を通過することで電離反応を起こし、生成される電子と陽イオンを電極に集める、あるいは電離の軌跡を可視化することで荷電粒子を検出することができる。主な検出器としては、電極への印加電圧が小さい領域では電離箱が、大きい領域ではワイヤーチェンバーや RPC などが挙げられる。

2.2.2 半導体検出器

半導体検出器とは半導体材料を使用した検出器を指し、光や電子をはじめとして様々な粒子を検出するものがある。半導体材料には主にシリコンやゲルマニウムが用いられ、接合ダイ

オードの原理を使用して製造される。以下では、シリコン検出器の構造と動作原理について説明する。まず基本構造としては、検出器の一方に正孔の多い p 型半導体、もう一方の面に自由電子が多い n 型半導体の p-n 接合が作られている。それぞれに対して逆バイアス電圧（p 型に負、n 型に正）を印加することで、p 型と n 型との間で正孔と自由電子の結合が進み、空乏層と呼ばれる安定化した領域が検出器の接合面を中心に広がる。この空乏層を通過した荷電粒子は、検出器内のシリコン原子を励起し、電子正孔対を生成する。飛跡に沿って生成された電子正孔対は、空乏層内の電場によって両印加極板までドリフトされ、パルス電流として測定される。また、シリコンなど半導体のバンドギャップは 1eV 程度であり、電子正孔対を生成するために必要なエネルギーはおよそ 3,4eV となり、近傍（あるいは半導体内部）の增幅回路によって信号の増幅を行う。半導体検出器は、電極が平面構造のピクセル検出器や帯状のストリップ検出器、検出器上に增幅回路や信号処理回路を形成するモノリシック検出器など、電極や増幅回路の実装によって様々な構造が存在する。

2.2.3 シンチレーション検出器

励起エネルギーの一部が、より低いエネルギー準位へ遷移する際に可視光として表れる物質をシンチレータという。シンチレータ検出器は、荷電粒子の通過によって発生した蛍光（シンチレーション光）を光検出器によって測定することで動作する検出器である。シンチレーション光は非常に弱い光信号であるため、光電子増倍管や半導体光センサーなどを用いて信号を検出する。

2.3 シリコンタングステン電磁カロリメータ SiW-ECAL

2.3.1 SiW-ECAL の全体構造

ILD の SiW-ECAL (図 2.4 青色部分) は、図 2.5 のようにタングステンの吸収層とシリコンパッドセンサーの検出層が、30 層サンドウィッヂ状に交互に重なったサンプリング型カロリメータである。1 つのモジュールが 10 ほどのサブモジュールに分かれており、サブモジュールにはそれぞれ 4 枚のシリコン半導体センサーが貼り付けられる。

2.3.2 シリコン半導体検出器

SiW-ECAL では検出層にシリコン半導体検出器を用いる。センサーの大きさは図 2.6 のような 1 枚あたり $9 \times 9 \text{ cm}^2$ の大きさで、1 枚に $5.5 \times 5.5 \text{ mm}^2$ のピクセルが 16×16 個並んでおり、1 つのサブモジュールあたり 1024 チャンネル読み出しが可能となっている。またシリコンセンサーには、電極としてアルミニウム (Al)、絶縁層に二酸化ケイ素 (SiO_2) が使用される。厚さ $320\mu\text{m}$ のセンサーでは、1MIP あたり 86.8 keV のエネルギー損失が起こり、臨界エネルギーに達するまでに生成される電子正孔対はおよそ 24,000、電荷にして 4 fC となる。

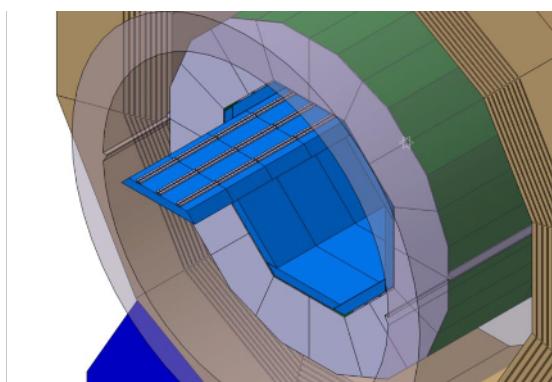


図 2.4: ILD および ECAL の全体図 [7]

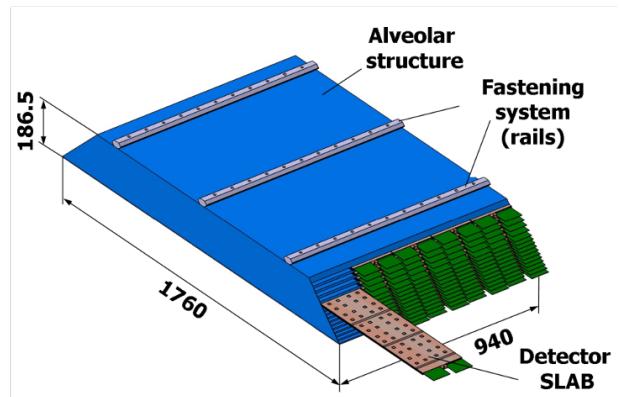


図 2.5: SiW-ECAL の構造 [7]

シリコンセンサーは、常温硬化型導電性接着剤によって回路基板（PCB）と接着されており、PCB を通して信号の読み出しが行われる。以下にセンサーの仕様と 1 枚のシリコンセンサー パッドを示す。

表 2.1: シリコンセンサーの仕様

| | |
|---------|---------------------------------|
| 制作会社 | 浜松ホトニクス株式会社 |
| サイズ | $89.7 \times 89.7 \text{ mm}^2$ |
| セルサイズ | $5.5 \times 5.5 \text{ mm}^2$ |
| セル数 | $16 \times 16 = 256$ |
| 厚さ | $320/500/650 \mu\text{m}$ |
| 完全空乏化電圧 | $40/70/110 \text{ V}$ |

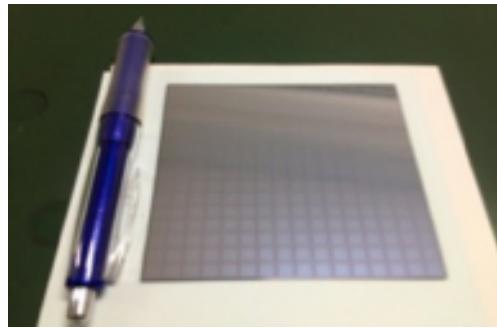


図 2.6: シリコンパッドセンサー [21]

2.3.3 読み出しシステム

SiW-ECAL は高精細であることから読み出しチャンネル数が非常に多くなっており、30 層の ECAL 全体ではおよそ 1 億にもおよぶ。そのため、シリコンセンサーからの信号読み出しをコンパクトにする必要があり、読み出し専用の Application Specific Integrated Circuit (ASIC) が開発された。現在の技術プロトタイプに実装されている ASIC には、フランスの Omega グループが開発した Silicon Kalorimeter Integrated ReadOut Chip (SKIROC) シリーズの第二バージョンである SKIROC2A を用いている。

まず、読み出しに用いる ASIC に求められる性能には次に挙げる項目が求められる。

- 自動トリガー：
信号に対して ASIC 自身でトリガーをかける。
- 完全デジタル出力：

シリコン半導体検出器からのアナログ情報をすべてデジタル変換し、データ量を圧縮して Data Acquisition (DAQ) へ送信する。

- 発熱量の抑制：
電力消費によって発生するジュール熱を、1 チャンネルあたり $25\mu\text{W}$ 以下に抑える。
- 1 MIP 相当の信号を識別できる高い Signal to Noise 比 (S/N 比)：
PFAにおいてジェットエネルギー分解能を向上させるために、高い精度でノイズからシグナルを分離する必要がある。

続いて、SKIROC2A の基本的な仕様を以下に示し、図 2.7 に SKIROC2A のアナログ部の回路図を示す。

- Austria Micro Systems 社製 $0.35\mu\text{m}$ SiGe
- $7.5 \times 8.5\text{ mm}^2/1$ チップ
- 1 チップあたり 64 チャンネル読み出し可能
- 2 種類のダイナミックレンジをもつ Analog to Digital Converter (ADC) mode：
電磁シャワー内の粒子が 1 つのチャンネルに大量に入射したときに全エネルギーを測定出来るよう、幅広いゲインのレンジを持つ。
 - High gain … $0.5 \sim 150$ MIP 相当の信号に対応
 - Low gain … $150 \sim 2500$ MIP 相当の信号に対応
- TDC (Time to Digital Converter) mode : 1 ns 程の時間分解能で時間情報を保存
- 1 チャンネルあたり 15 イベント保持可能な Analog memory cell
1 つの Memory cell では、High gain ADC、Low gain ADC を、時間情報である Bunch crossing ID (BCID) と紐づけて保存。
- 数珠つなぎ型読み出し：順番に読み出すことで読み出していないチップの電源を必要とせず、消費電力を減らす
- 0.5 MIP での自動トリガー
- 全 64 チャンネルの閾値を個別で同時設定可能な 10 bit Digital Analog Converter (DAC) threshold
- Power pulsing mode

データ収集の手順は、以下のとおりである。まずシリコンセンサーからのアナログ信号が各チャンネルに入力され、前置増幅器によって前段増幅を行う。ここでの増幅率は、前置増幅器の feedback capacitance によって変更可能となっており、増幅率が最大となる 0 pF から 6.0 pF まで 0.4 pF 刻みで決定することができる。前段増幅を経た信号は、Fast shaper と 2 つの Slow shaper (Low/High gain) の 3 つに分割される。Fast shaper に入った信号は、CRRC shaper によって信号をさらに増幅したのち、discriminator にて閾値を超えた場合のみトリガー信号を出し、同時に入ってきた Slow shaper の信号電圧を Memory cell に保持する。ここで、CRRC shaper とは微分回路と積分回路を組み合わせた回路であり、信号増幅率と立ち

上がり時間を調整している。Memory cell に保持された電荷は、読み出し時間が経過、あるいは Memory cell が満たされたタイミングでマルチプレクサー (MUX) に送られる。一方、Slow shaper においても信号が CRRC shaper によって増幅率 1、10 倍に増幅され、トリガー信号によって MUX に送られる。MUX では、Slow shaper (Low or High) と TDC の組み合わせを決定し、Wilkinson 型 12 bit ADC によってアナログ信号をデジタル化し、メモリに保存されたのち外部へ転送される。

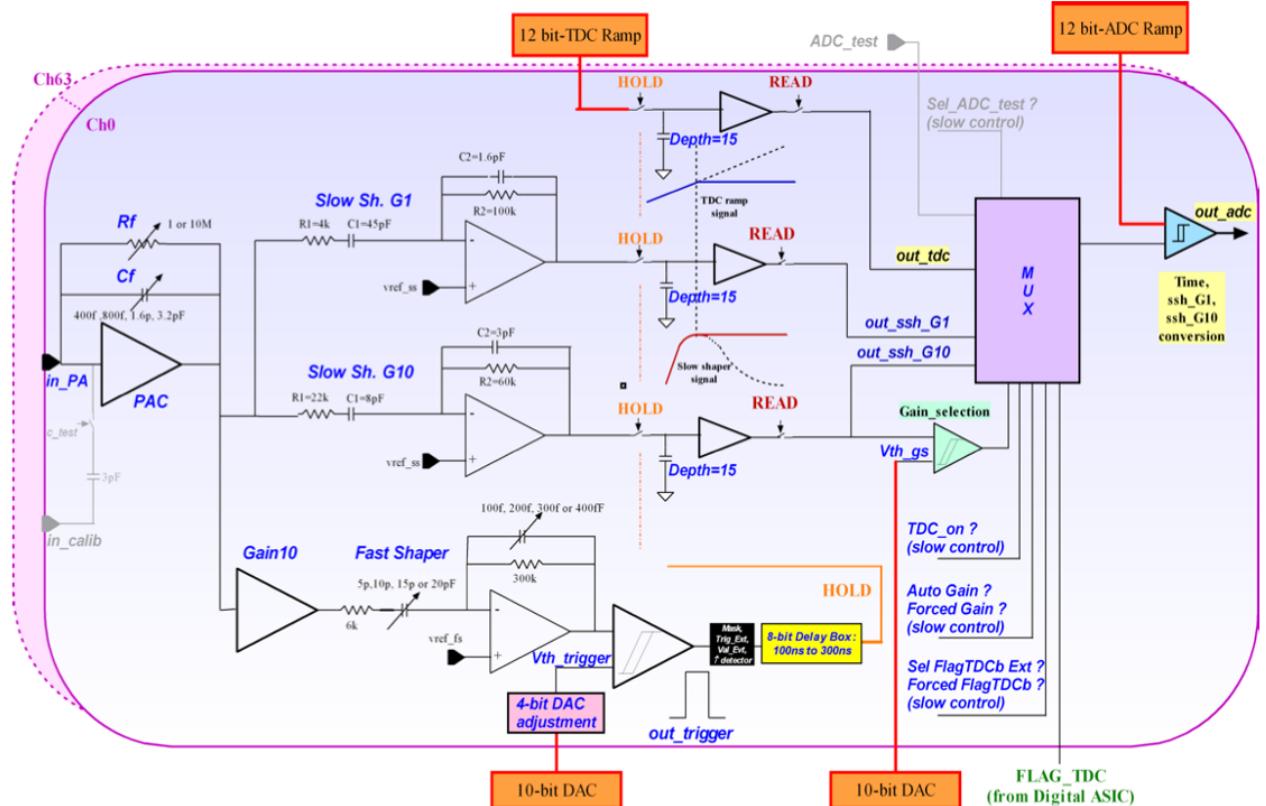


図 2.7: SKIROC2A のアナログ部の回路図

2.3.4 タングステン吸収層

ILD の SiW-ECAL の吸収層では PFA の要件を満たすために、相互作用長が長く放射長の短い物質でシャワーの広がりを小さく抑える物質を採用する必要がある。そのため、モリエール半径が小さく、放射長に比べ相互作用長が大きいタングステンが採用されている。表 2.2 に物質量が大きく吸収層の候補となる物質の性質について示す。

| 物質 | λ/cm | L_R/cm | R_M/cm |
|--------|---------------------|-----------------|-----------------|
| 鉄 | 16.8 | 1.76 | 1.69 |
| 銅 | 15.1 | 1.43 | 1.52 |
| タングステン | 9.6 | 0.35 | 0.93 |
| 鉛 | 17.1 | 0.56 | 1.00 |

表 2.2: 物質量の大きい吸収層の候補物質 (λ は相互作用長、 L_R は放射長、 R_M はモリエール半径を示す。)

2.3.5 技術プロトタイプ

SiW-ECAL の技術プロトタイプとして、図 2.8 のような構造が考えられている。図 2.8 は、1 層の検出器 (Slab) の一部として作製された Short slab である。図の上から構造体として炭素繊維強化プラスチック (CFRP) の板があり、その下に読み出し基板 (FEV; フロントエンドボード) や ASIC の制御基板があり、基板裏にはセンサーを導電性接着剤を用いて接着しており、その下にフレキシブル基板を挟み、再び CFRP の板で挟まれている。1 層あたり、FEV には PCB 上に SKIROC2A が 16 チップ実装されており、裏面にはシリコンパッドセンサー 4 枚が導電性接着剤で接着されている。また、ASIC は FPGA を通して制御されており、さらに図 2.9 に示すモジュールを通して、複数の Slab からの信号を同時に読み出している。

読み出し基板には、図 2.10 のように ASIC の実装手法によって大きく分けて BGA (Ball Grid Array) と COB (Chip On Board) の 2 つの案が挙がっている。(以下では FEV と表記するものは BGA による実装を指す。) FEV は PCB の裏面に 4 枚のシリコンセンサーを導電性接着剤で接着しており、その内側にフレキシブル基板を接着することでシリコンセンサーへの電圧供給を行う。また PCB 表面には ASIC が 16chip 実装されており、センサーからの信号をデジタル変換している。FEV には FEV7 から 13 までのバージョンがあり、内部配線や外部接続の面でアップデートが行われている。一方で COB では、ASIC が壅みにワイヤーボンディングされており、これによって読み出し基板が FEV よりも薄いコンパクトな構造になっている点が特徴である。しかし同時に、FEV と比較してノイズの影響を受けやすいなど欠点も抱えている。

これら技術プロトタイプの開発は、フランスと日本を中心に CALICE グループによって国際協力で行われている。CALICE (Calorimeter for Linear Collider Experiment) は ILC に向か、カロリメータの共同開発を目的として結成された国際コラボレーションであり、九州大学のグループもこの一員である。また、モジュールの生産・組み立ては国内においても可能くなっている。

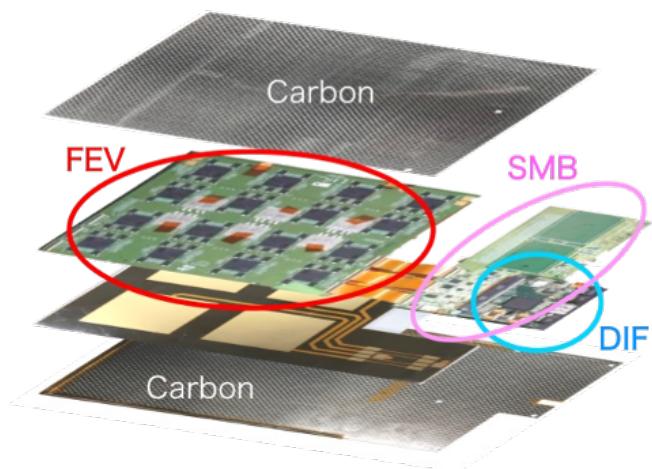


図 2.8: short slab の構造



図 2.9: 多層読み出しのための CORE モジュール



図 2.10: (左) FEV13 (右) COB

第3章

ビームテストによる評価試験

本章では、CERN SPS 加速器で行ったビームテストについて述べる。初めに 3.1 節でビームテストの概要について述べる。続いて 3.2 節でビームセットアップについて述べ、3.3 節で実験結果についてまとめるとする。

3.1 概要

これまでに作成された SiW-ECAL の技術プロトタイプ (FEV および COB) の性能評価実験を、2023 年 6 月 7 日から 2023 年 6 月 22 日の期間に CERN SPS 加速器のビームラインにて行った。本実験の主な目的は、電磁カロリメータとハドロンカロリメータの技術プロトタイプを同じビーム軸上に設置し、同時に運転を行いデータを取得すること、またこれまでの評価実験の中でも最高エネルギーのハドロンビームを用いて 15 層の SiW-ECAL の評価を行うことの 2 点であった。また本実験におけるハドロンカロリメータは、同じく CALICE グループにおいてドイツやチェコが中心となって開発を進めている AHCAL を用いた。以下では実験の詳細と、結果について述べる。

3.1.1 ビームライン

ビームテストは、フランスとスイスの国境付近に位置する CERN (図 3.1) の SPS 加速器のビームラインを用いて行った。SPS は、現在 LHC の前段加速器として利用されており、陽子シンクロトロンから来た 26 GeV の陽子を、周長 7km の加速器によって 400 ~ 450 GeV まで加速している。SPS のビームラインでは、陽子ビームを固定ターゲットに入射することで、電子、ミューオン/パイ中間子の 2 次ビームを運動量 10 ~ 400 GeV で得ることができる。本実験ではこれらのビームを用いて、North エリアにある H2A ビームラインで実験を行った。実験に用いたビームパラメータを表 3.1 に示す。

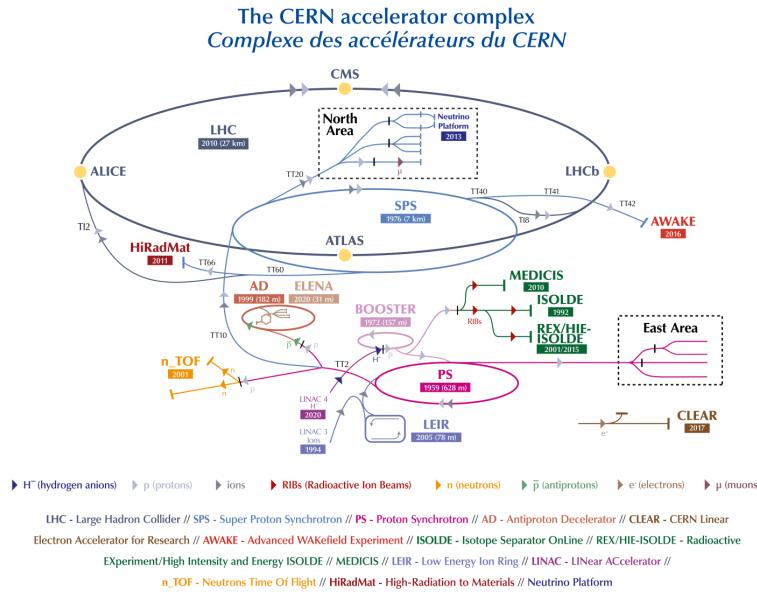


図 3.1: CERN 加速器の全体図 [2]

| | |
|------------------|---------------|
| 運動量 | 10-200[GeV/c] |
| 電子の purity | 10-99.5% |
| Max $\delta p/p$ | 2% |
| ビームの高さ | 2460[mm] |

表 3.1: H2A ビームラインにおけるビームパラメータ

3.2 実験セットアップ

3.2.1 測定機器のセットアップ

本実験でのセットアップの概観を図 3.2 に示す。図中右手からビームが照射され、ILD の構成と同様に上流側に SiW-ECAL が、下流側に AHCAL を設置した。SiW-ECAL のプロトタイプは、FEV が 13 層（うち FEV10 が 1 層、FEV11 が 3 層、FEV12 が 2 層、FEV13 が 7 層）COB が 2 層を組み合わせた計 15 層からなる検出層とタンゲステン板 15 層の吸収層からなるモジュールを組み立てた。今回のビームテストの検出層は、最も良い性能が期待できる FEV13 を前方に設置し、性能比較のために COB を加えた構成となった。15 層の検出層において、センサーと読み出しボードに供給する電源は、15 層に対して図 3.3 左のように並列で印加した。また、信号はカプトンケーブルを通して 15 層分の信号を一括して CORE モジュールに送り、PC で読み出しを行った。また、AHCAL もサンプリング型ハドロンカロリメータであり、鉄の吸収層とシンチレータ検出層によって構成されていて、読み出しはシンチレータ

に統合されたエレクトロニクスによって行われる。

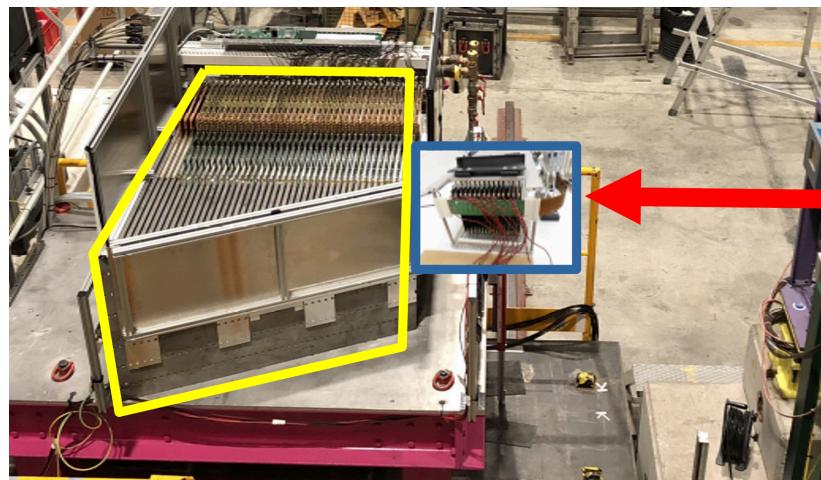


図 3.2: セットアップの全体図。黄枠 : AHCAL、青枠 : SiW-ECAL、赤 : ビーム位置

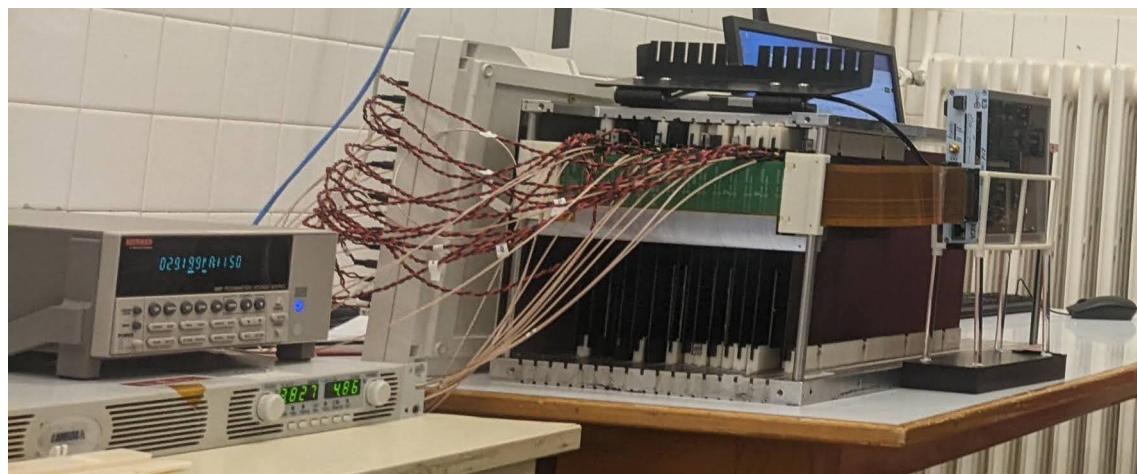


図 3.3: SiW-ECAL のプロトタイプモジュールと電源供給、読み出しシステム

| 層数 | ボードの種類 | ウェハー厚み [μm] | タンクステン厚み [mm] |
|----|--------|--------------------------|---------------|
| 0 | FEV13 | 650 | 4.2 |
| 1 | FEV13 | 650 | 4.2 |
| 2 | FEV13 | 650 | 4.2 |
| 3 | FEV13 | 650 | 4.2 |
| 4 | FEV13 | 500 | 4.2 |
| 5 | FEV13 | 500 | 4.2 |
| 6 | COB | 500 | 4.2 |
| 7 | FEV12 | 500 | 4.2 |
| 8 | COB | 500 | 5.6 |
| 9 | FEV12 | 500 | 5.6 |
| 10 | FEV11 | 320 | 5.6 |
| 11 | FEV11 | 320 | 5.6 |
| 12 | FEV10 | 320 | 5.6 |
| 13 | FEV13 | 320 | 5.6 |
| 14 | FEV11 | 320 | 5.6 |

表 3.2: SiW-ECAL のレイヤー構成 (0 層目がビーム上流側)

3.2.2 信号読み出し

SiWECAL と AHCAL は独立に DAQ を行うが、本実験では EUDAQ という読み出しフレームワークによって AHCAL と同期した読み出しを行った。EUDAQ では、SiWECAL と AHCAL の間で Clock and Control Card (CCC) を同期させており、CCC ではクロック、スタート、ストップ信号をバックエンドへ送っているため、PC 上で時間情報である Bunch Crossing ID (BCID) が同じイベントを収集することができる。

また、SiWECAL では専用の DAQ ソフトウェア [22] が稼働しており、DAQ のみでなくチャネル単位での閾値の設定やイベントのモニター (図 3.4) が可能となっている。データは raw 形式で保存され、解析用のソフトウェアを通してツリー形式にデータを保存した root ファイルへ変換し解析に利用することができる。

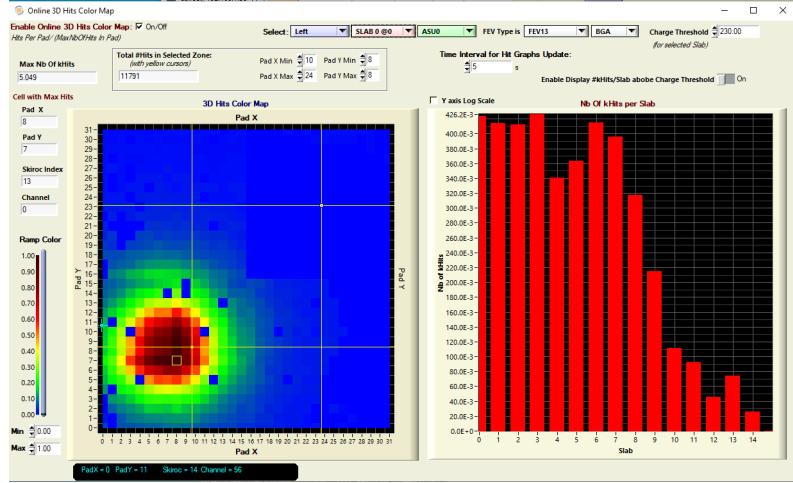


図 3.4: SiW-ECAL 読み出しソフトウェアにおけるイベントモニター (左) ヒットマップ (右) 各層あたりのヒット数

3.3 実験結果

3.3.1 検出器応答

ビームテストを通して、 10GeV から 150GeV まで電子・ミューオン・パイオンビームを検出器に照射し、SiW-ECAL と AHCAL の BCID を同期したイベントの収集を行った。一度の測定 (run) はおよそ 30 分間でビーム位置を固定して照射した。初めに、各 run に対して層ごとのヒットマップを作成し、各層の応答を確認した。作成したヒットマップを図 3.5 に、run の詳細を表 3.3 に示す。ビームはヒットマップ左下のセンサーを中心に照射を行った。

電子ビームは相互作用を起こし電磁シャワーを形成するため、ヒット領域が大きくなっていることが確認できる。またミューオンビームは、基本的に単体では電磁シャワーを形成しないものの、ヒットマップではビームサイズが大きくなっている。これは、ビームライン中においてパイオンの崩壊などによってビーム内の位置分布が広がることによる影響であると考えられる。また白く抜けている箇所はノイズが多いためマスクしている、あるいは信号がないチャンネルであるが、ビームの種類によらず全体を通して 4 つほど四角く抜けている部分が確認できる。これは 1 枚のセンサーの場所と対応しており、導電性接着剤の剥離、高電圧の接続不良、センサーの損傷などによるものである。

さらにビームのエネルギーを上げていくと、 80GeV 以上のエネルギーで図 3.5 (b) 第 6 層のような、1 センサー全体にヒットが集中してしまう現象が確認された。エネルギーが大きくなった場合にはセンサーに入ってくる信号が多くなることから、センサーや信号処理回路に大きな電流が流れたことによるノイズの増大や電位の変動などが原因として考えられるが、現在調査を行なっている。

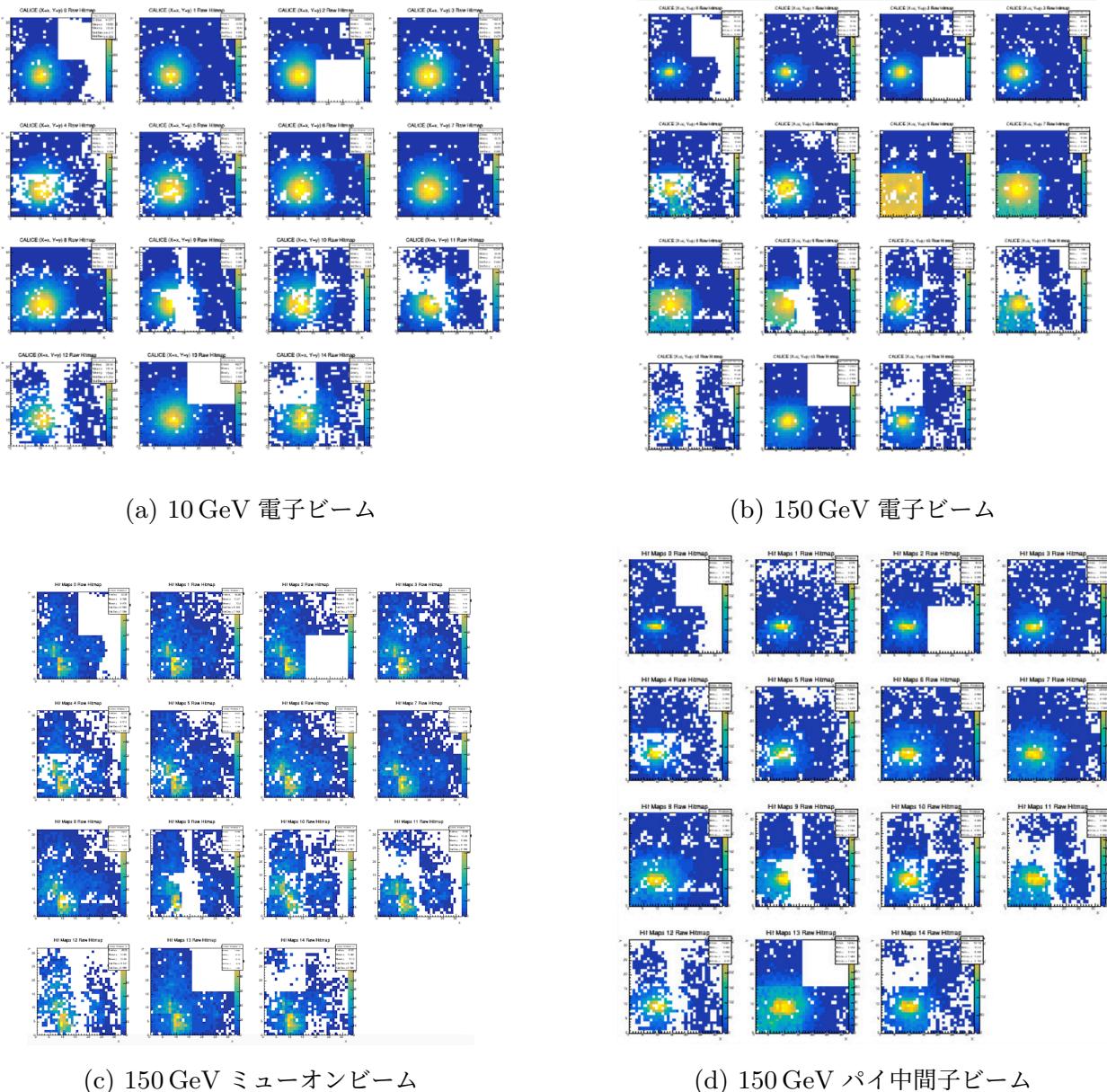


図 3.5: 各ビーム種類におけるヒットマップ。左上から 0 層右上が 3 層、右下が 14 層と順に並んでいる。

| 図番号 | ビーム種類 | エネルギー [GeV] | トリガー数 | データ取得時間 [分] | run 番号 |
|-----|-------|-------------|--------|-------------|--------|
| (a) | 電子 | 10 | 23790 | 30 | 90289 |
| (b) | 電子 | 150 | 40542 | 30 | 90344 |
| (c) | ミューオン | 150 | 179078 | 30 | 90484 |
| (d) | パイオニ | 200 | 64208 | 30 | 90341 |

表 3.3: ヒットマップの作成において使用した run の詳細

3.3.2 ペデスタル・Retriggering

続いて、ペデスタルの解析を行った。ペデスタルとは信号が全くきていない時（トリガーが入っていない時）の信号の大きさで、エレクトロニクス由来のノイズ等によってその値の幅は変化する。信号のゲインなどは、ペデスタルからどれだけ離れているかで定義するため、それぞれのチャンネルでペデスタルがどのようにになっているか調べることは重要な情報となる。一般的にペデスタルは図3.6のようにガウス関数でフィッティングすることができ、その中央値はペデスタル信号の大きさを、標準偏差はその揺らぎの大きさ（ノイズ）に相当する。図3.7に各エネルギーの電子ビームのrunにおける同じチャンネルのペデスタルを示した。この図より、ビームのエネルギーに依らずペデスタルの中央値は一定の値を取ることが確認された。また、runの長さは同じであるものの、エネルギーが大きいビームでは、ペデスタルの信号数が多くなった。さらにペデスタルの波形に注目すると、低エネルギーになるにつれてピークが2つに分かれてきていていることも確認できる。これはこの節の後半に示すダブルペデスタルであると考えられる。

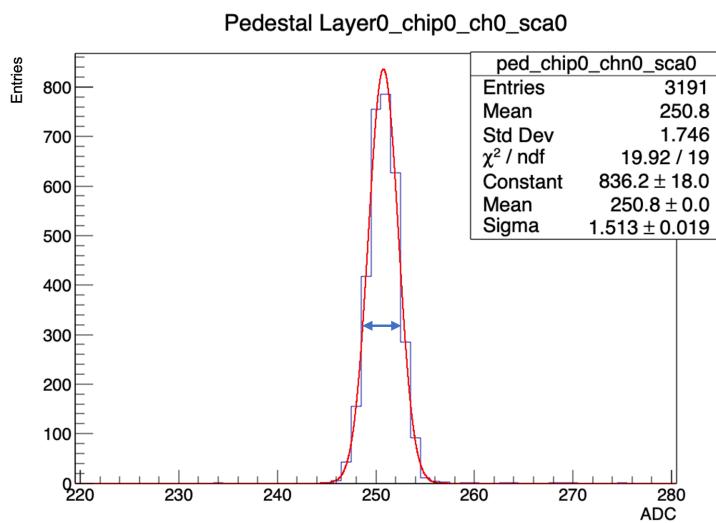


図3.6: ペデスタルヒストグラムの例

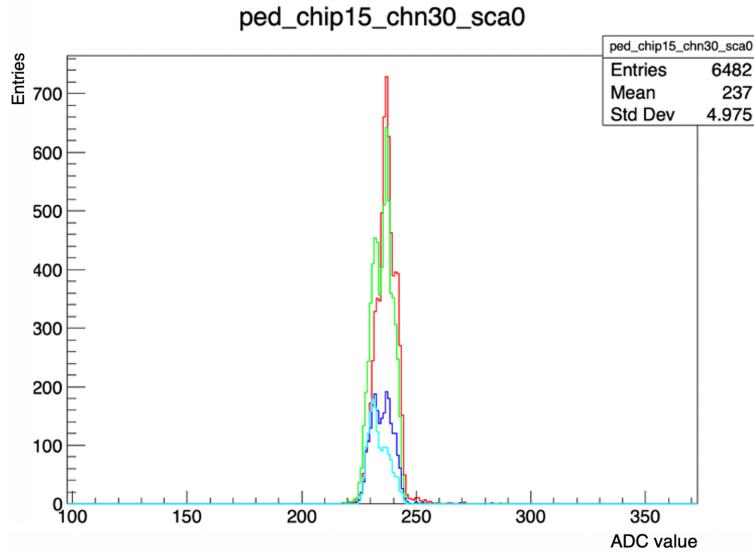


図 3.7: 水色 : 40 GeV / 紺 : 60 GeV / 緑 : 100 GeV / 赤 : 150 GeV の電子ビームにおけるペデスタル

また、各チャンネルにおけるペデスタルの中央値の大きさと標準偏差を図 3.8 に示す。図は横軸がチップのインデックスに対応し、縦軸がチャンネルのインデックスに対応している。また、黒枠は一つの層を表している。この図より、ペデスタルの中央値が層、チャンネルによって異なることが確認された。そのためペデスタルの分布を確認したところ、チャンネルによって図 3.9 に示すような 2 つのピークを持つペデスタルが確認された。このペデスタルピークが 2 つ見える現象をダブルペデスタルと呼んで調査・解析を行なった。また、今回の実験によって取得されたデータには、トリガーが入っているもののヒットのイベントが記録されていないデータがあり、この現象を Retriggering と呼んでいる。Retriggering はあるチャンネルに信号が入った時に、時間情報である BCID が近いところで多数のチャンネルにヒットが残る現象であり、実際のヒットではビームの周波数からこのような短時間でのヒットの集中は起こらない。Retriggering が繰り返し発生してしまうと、memory cell が偽物のヒットで満たされてしまうため、本来のデータが収集できないという問題が発生してしまい、DAQ において非常に重要な課題となっている。そのためダブルペデスタルの状況と Retriggering の発生について詳細に解析を行った。

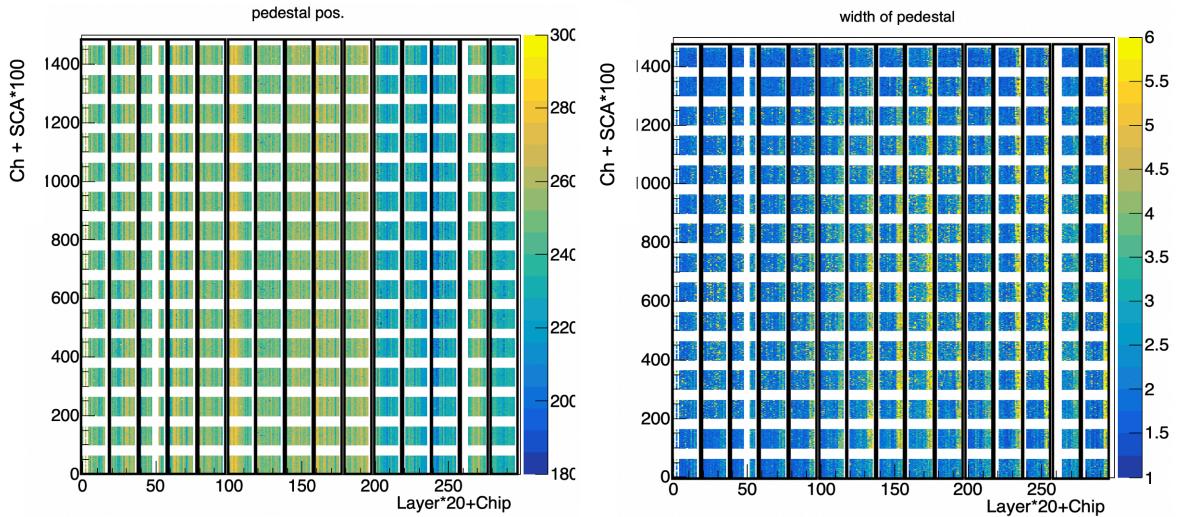


図 3.8: ペデスタルをガウス関数フィットした際のパラメータ。横軸がチップを、縦軸がチャンネル番号を表しており、それぞれ全層、全 SCA での可視化のため、層数や SCA をかけた値となっている。(左) ガウス関数の中央値 (右) ガウス関数の幅の大きさ

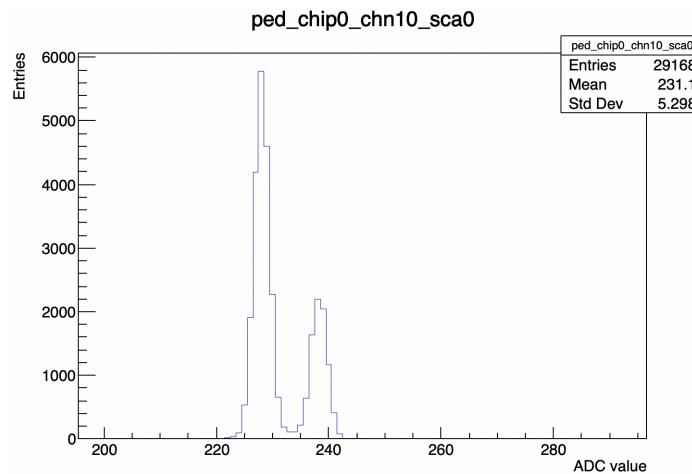


図 3.9: ダブルペデスタルの例

ペデスタルはトリガーが入っていない時の信号であるため、解析にはトリガーが少ないことが想定されるミューオン 150 GeV の run を用いた。はじめに各チャンネルにおける ADC の情報から、トリガーが入っていない条件に加え、Retriggering である可能性が高い場合には時間情報である BCID がヒットよりも短い期間に連続して記録されるため、連続した BCID (+2 BCID まで) をカットする条件をかけることで、ペデスタルのヒストグラムを作成し、ピークの数を調査した。続いて、ピークが 2 つ以上見られるチャンネルに対して、各ピークに対してガウス関数でフィッティングを行い、中央値からペデスタルの信号の大きさを特定した。ピーク同士の信号の大きさはチャンネルによって異なり、図 3.10 に各チャンネルにおいてダブルペデスタルが発生した時の、2 つのペデスタル信号の大きさの差を示す。1 つのマップは 1 つ

の層に対応しており、左上から 0 層右上が 3 層、右下が 14 層と順に並んでいる。また、ダブルペデスタルが発生したチャンネルには、2 つのペデスタル信号の大きさの差をカラーバーに示している。このマップから、10、11、12、14 層でダブルペデスタルが多く発生していることが確認できる。これは 10、11、12、14 層がそれぞれ、FEV11、FEV11、FEV10、FEV11 と以前のバージョンの FEV を使用しており、過去の配線のアップデートによってダブルペデスタルの原因が大きく解消されていることが推測される。また、全層を通してダブルペデスタルはボードの左下部分を中心に発生していることが確認できた。

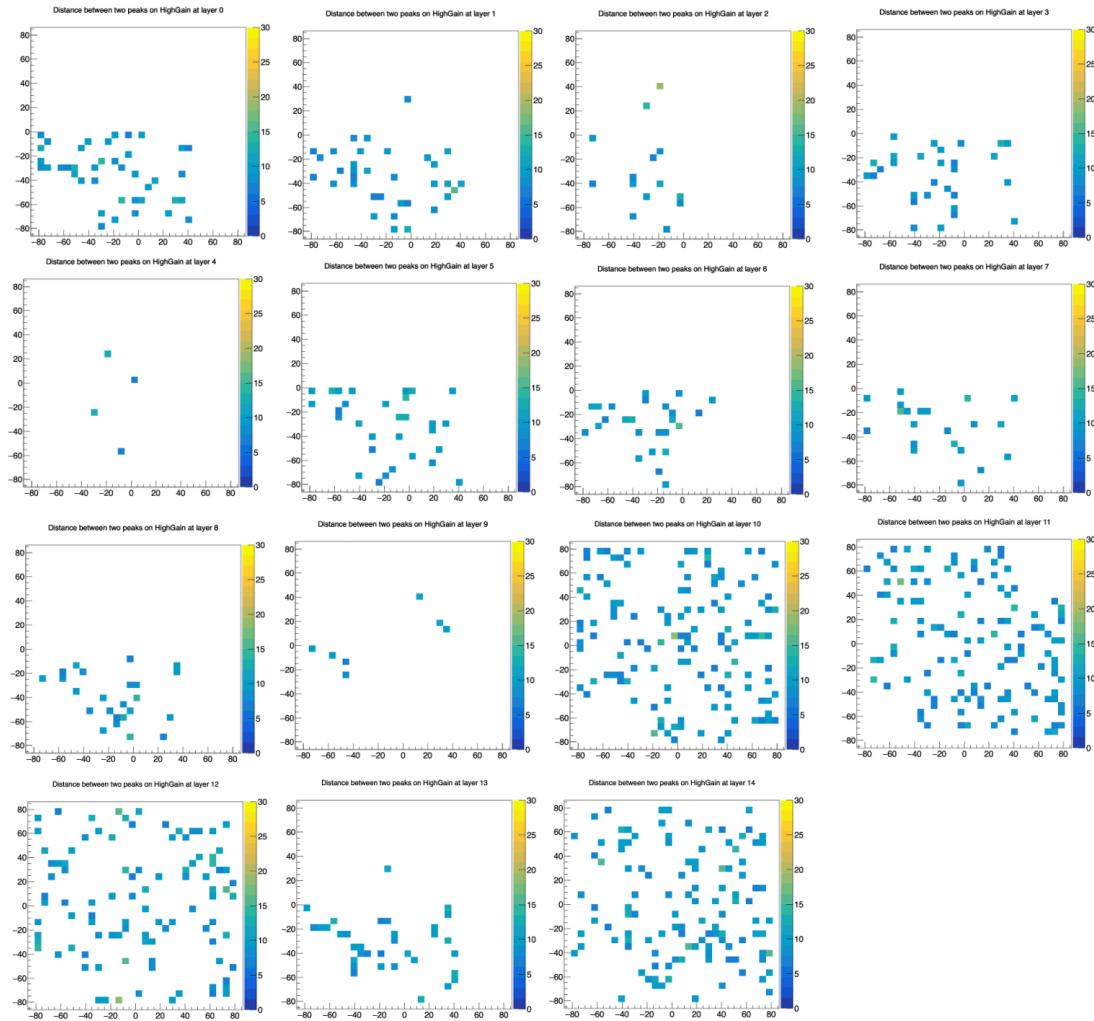


図 3.10: 各層におけるダブルペデスタルのピーク間距離。左上から 0 層右上が 3 層、右下が 14 層と順に並んでおり、 x, y 軸の値はセンサー中心を原点とした、各チャンネルの座標を表す。

また、同一のデータを用いて Trigger と Retriggering のマップを作成した。Retriggering は、先行研究から連続して空のイベントが大量に入ることがわかっているため、BCID の条件にカットをかけることで、セレクションを行った。図 3.11 に Trigger のマップを、図 3.12 に Retrigger のマップを示す。図 3.11 より、各層において右下部分にビームスポットがあり

Trigger が発生していることがわかる。また、Retriggering のマップからは、13 層を除くと 10、11、12、14 層で発生数が多く、ダブルペデスタルが多く発生していた層と一致した。さらに 10、11、12、14 層では顕著だが、全体を通してビームスポットあたりに Retriggering が発生していることが確認できる。

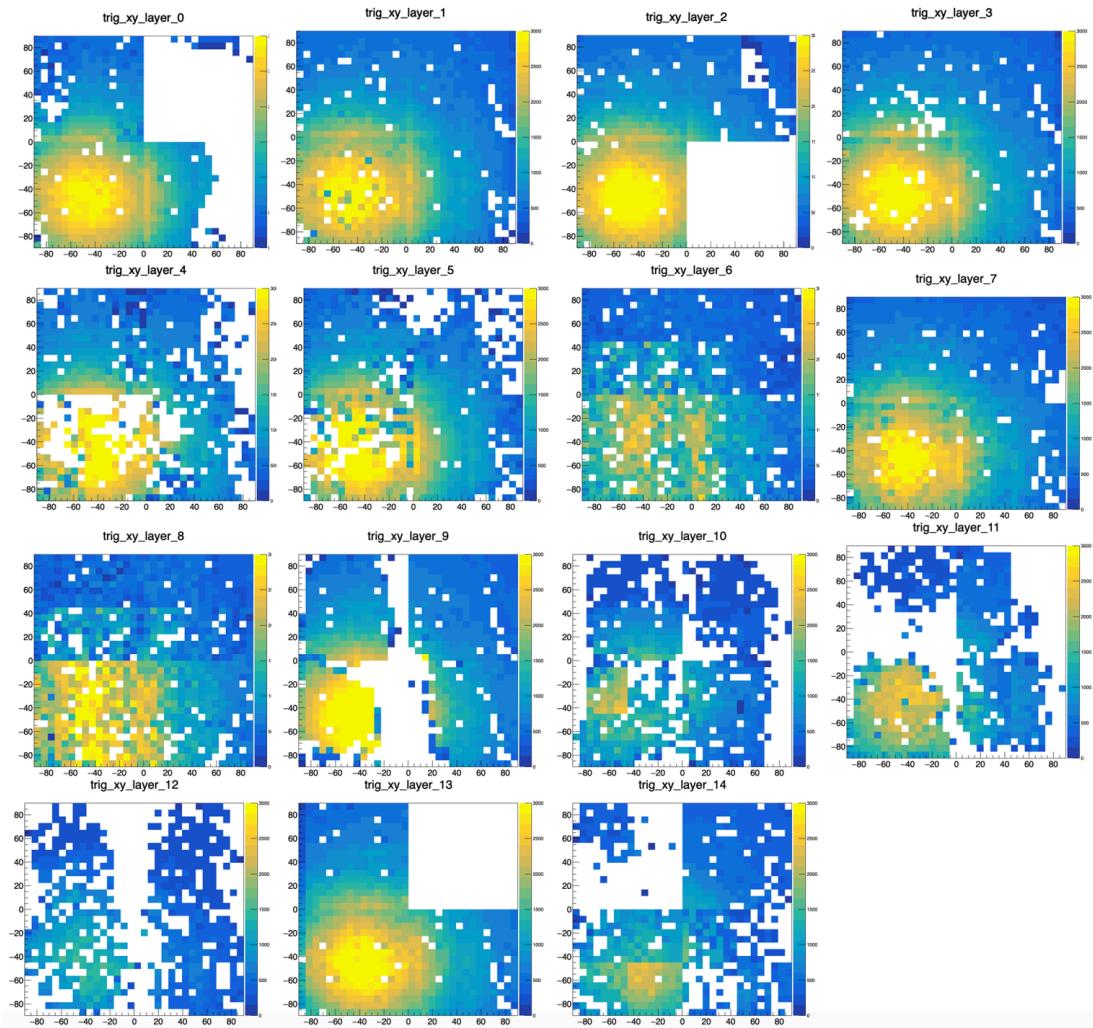


図 3.11: 各層における Trigger の発生数

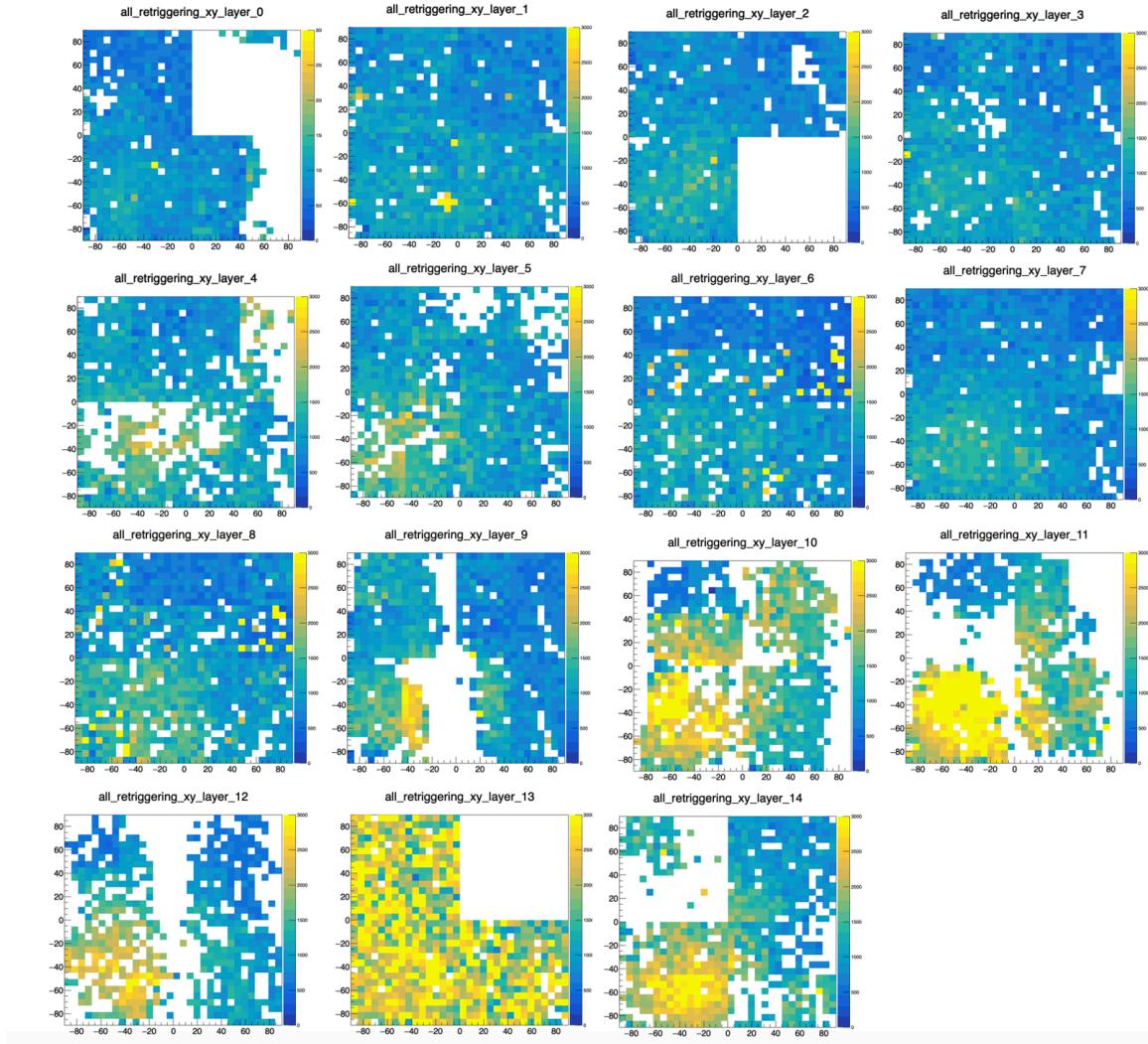


図 3.12: 各層における Retriggering の発生数

上記の結果より、Retriggering は Trigger 数の多い箇所で発生数が多く、またそこでは同時にダブルペデスタルが発生している可能性が高いことを推測することができる。配線を改善したFEVのアップデートでは、ダブルペデスタルだけでなく Retriggering の発生数も抑えることができていることが確認できた。また、Retriggering 現象は先行研究から、ヒットによる基板上の增幅回路の電源の搅乱によって電位が揺れることが原因として考えられており、今回の結果もヒットの多いところで発生が集中していた。そのため Retriggering、ダブルペデスタルを抑制するために、今後のプロトタイプでは回路内でグラウンドを強化するなどの対策が必要である。

第 4 章

深層学習

本章では、本研究で提案する手法である深層学習の理論を述べる。初めに 4.1 節で深層学習の基礎技術であるパーセプトロンについて説明し、パーセプトロンを多層にしたニューラルネットの構造と計算技術について説明する。そして 4.2 節で深層学習のネットワークについて、特にグラフ構造のデータを扱うグラフニューラルネットワークについて紹介する。

4.1 ニューラルネットワーク

4.1.1 パーセプトロン（単層ニューラルネットワーク）

ニューラルネットワークの基礎となるパーセプトロンは、ローゼンブラットにより 1957 年に考案された [23]。パーセプトロンの基本構造は、信号を入力として受け取り論理回路を通して出力信号を出すものである。図 4.1 に最も基本的なパーセプトロンの例 [24] を示す。

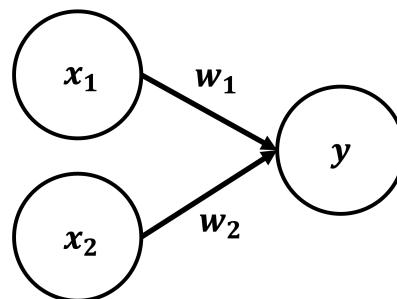


図 4.1: パーセプトロン [24]

x_1, x_2 は入力信号、 y が出力信号であり、 w_1, w_2 がそれぞれの入力信号にかかる重みを表す。また、図中における \circ はノードと呼ぶ。入力信号はノードに送られる前に重みが掛けられ、出力ノードにてそれらの総和をとる。出力ノードでの演算（活性化関数）をステップ関数（階段関数）とすると、その総和が閾値 θ を超えている場合のみ出力信号は 1 を出力すること

になる。数式で示すと以下のようなになる。

$$y = \begin{cases} 0 & (w_1x_1 + w_2x_2) \leq \theta \\ 1 & (w_1x_1 + w_2x_2) > \theta \end{cases} \quad (4.1)$$

パーセプトロンにおいて重要なのは入力信号に対する固有の重みであり、重みは各信号の重要性を操作する要素として働く。すなわち重みが大きいほど、対応する信号の全体における重要性が高くなる。この重みを更新する操作を学習と呼び、ニューラルネットワークでは学習を繰り返すことによって重みパラメータを理想とする値に近づけていく。

また、入力信号が3つ以上の場合についても考えることができ、以下のような式で表される。入力信号 $x = \{x_1, x_2, \dots, x_n\}$ 、重みパラメータ $w = \{w_1, w_2, \dots, w_n\}$ 、活性化関数（ここではステップ関数）を $h(x)$ とすると、出力ベクトル y は以下のようになる。

$$y = h(w^T x) = \begin{cases} 0 & (w_1x_1 + w_2x_2 + \dots + w_nx_n) \leq \theta \\ 1 & (w_1x_1 + w_2x_2 + \dots + w_nx_n) > \theta \end{cases} \quad (4.2)$$

4.1.2 多層パーセプトロン（多層ニューラルネットワーク）

パーセプトロンの演算では線形領域のみしか表現できず、非線形領域においても扱えるよう入力層と出力層の間に中間層（隠れ層）を加えるニューラルネットワークに改良された。このような中間層を複数重ねたパーセプトロンを多層パーセプトロン（Multi Layer Perceptron; MLP）と呼ぶ。多層パーセプトロンの簡単な例を図4.2に示す。

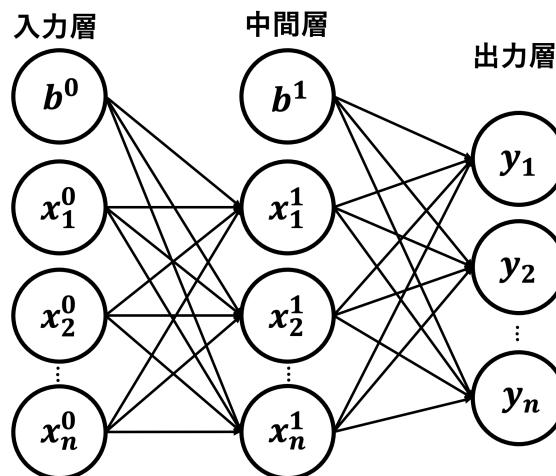


図4.2: 多層パーセプトロン（ニューラルネットワーク）

最も左のノード列を入力層、真ん中のノード列を中間層、一番右のノード列を出力層とすると、以下のような数式で表される。入力信号 $x^0 = \{x_1^0, x_2^0, \dots, x_n^0\}$ 、中間層の各ノードに入ってくる信号 $x^1 = \{x_1^1, x_2^1, \dots, x_n^1\}$ 、入力層と中間層の信号にかかる重みパラメータがそれぞ

れ $w^0 = \{w_1^0, w_2^0, \dots, w_n^0\}$, $w^1 = \{w_1^1, w_2^1, \dots, w_n^1\}$ 、活性化関数を $h(x)$ とすると、出力ベクトル y_n は

$$x_1^1 = h(w_1^0 x_1^0 + w_2^0 x_2^0 + w_3^0 x_3^0 + \dots + w_n^0 x_n^0 + b^0) \quad (4.3)$$

$$y_n = h(w_1^1 x_1^1 + w_2^1 x_2^1 + w_3^1 x_3^1 + \dots + w_n^1 x_n^1 + b^1) \quad (4.4)$$

となる。ここで、 b^n としてより学習にパラメータを加えるため、各層に実数値のバイアスを導入した。重みと信号の積の和を a として、上式に行列を用いると簡略に表現できる。

$$\mathbf{y} = h(\mathbf{a}) \quad (4.5)$$

$$\mathbf{a} = \mathbf{W}\mathbf{x} + \mathbf{b} \quad (4.6)$$

以下ではニューラルネットワークの学習における、学習の仕組みや重要な技術について取り上げる。

活性化関数

活性化関数はニューラルネットワークにおける入力の重み線形和から、出力を決定するための関数である。活性化関数には、非線形演算によって多層構造での表現力を高めるために非線形関数が用いられ、以下に主なものについて示しそれぞれの関数のグラフを図 4.3 に示す。中でも ReLU 関数は、勾配の最大値が 1 であることから勾配消失を起こしにくく、計算の安定性の理由から多くのモデルにおいて用いられている。(sigmoid 関数の場合 $x = 0$ で最大となり、それ以外で急速に小さくなってしまう)

- ステップ（階段）関数

$$h(a) = \begin{cases} 0 & (a \leq \theta) \\ 1 & (a > \theta) \end{cases} \quad (4.7)$$

- sigmoid 関数

$$h(a) = \frac{1}{1 + \exp(-a)} \quad (4.8)$$

- tanh 関数

$$h(a) = \tanh(a) \quad (4.9)$$

- ReLU 関数（ランプ関数）

$$h(a) = \begin{cases} 0 & (a \leq \theta) \\ a & (a > \theta) \end{cases} \quad (4.10)$$

- LeakyReLU 関数 ($s=0.01$ が多い)

$$h(a) = \begin{cases} sa & (a \leq \theta) \\ a & (a > \theta) \end{cases} \quad (4.11)$$

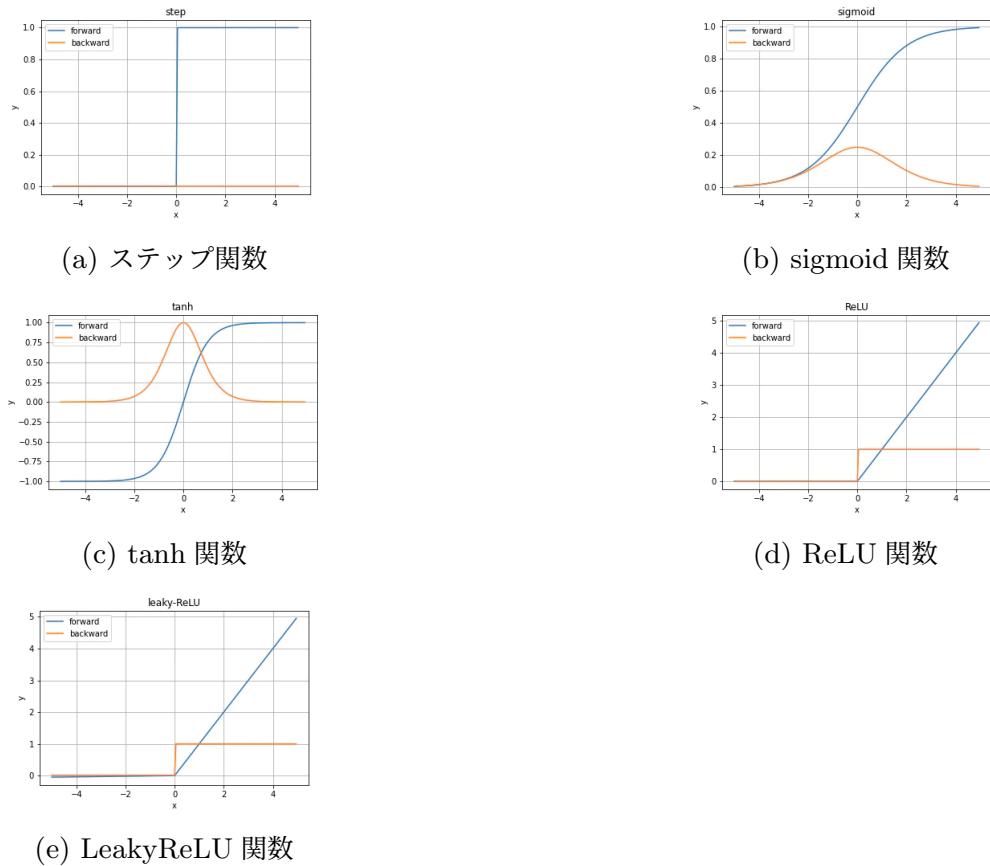


図 4.3: 活性化関数のグラフ。Forward は順方向、backward は逆方向（後に誤差逆伝播法で説明）の際の演算を表す。

出力層の設計

ニューラルネットワークで扱える問題は、主に回帰問題と分類問題に分けられる。それぞれの問題によって出力層の設計が異なり、回帰問題においては恒等関数が、分類問題においては主にソフトマックス関数が用いられる。恒等関数では入力された値をそのまま出力する。ソフトマックス関数は以下の式 4.12 のように、0 から 1 までの値を出力する関数であり、それぞれのカテゴリに分類される確率を表す。ここで y_k はニューラルネットワークの出力、 x_k は出力層へ入ってくる信号を表す。また、出力層のノード数は問題に合わせて適宜調整する必要があり、分類問題であればカテゴリ数だけノードを設計する必要がある。

$$y_k = \frac{\exp(x_k)}{\sum_{i=1}^n \exp(x_i)} \quad (4.12)$$

損失関数

先述の通り、ニューラルネットワークでは学習によって重みを更新するが、その際に学習結果を正しい答えと照らし合わせて評価し、その評価値を最小にするように重みを更新する。この評価関数を損失関数 (loss function) と呼ぶ。損失関数には主に以下の 2 つが用いられる。

- 二乗和誤差関数 (Mean Squared Error)

二乗和誤差関数は、以下の式 4.13 で定義される関数である。ここで、 y_k はニューラルネットワークの出力、 t_k は正解ラベルを表し、 k はデータの次元数を表す。二乗和誤差関数の微分値は y の一次関数となっていることから、出力・正解ラベルが共に連続値であり恒等関数を出力層を持つ回帰問題で採用される。

$$L = \frac{1}{2} \sum_k (y_k - t_k)^2 \quad (4.13)$$

- 交差エントロピー誤差 (Cross Entropy Error)

交差エントロピー誤差は、以下の式 4.14 で定義される関数である。ここで、 y_k はニューラルネットワークの出力で、 t_k は 0 か 1 の one-hot 表現の正解ラベルを表す。交差エントロピー誤差は主にソフトマックス関数を出力層に用いる分類問題において採用される。

$$L = - \sum_k t_k \log(y_k) \quad (4.14)$$

誤差逆伝播法

これまでではニューラルネットワークの順方向の伝播 (forward propagation) について見てきたが、出力層において学習結果と正解ラベルを比較し、逆方向に信号を伝播させ重みを更新するアルゴリズムを誤差逆伝播法 (back propagation) という。誤差逆伝播法では、次のような処理を行う。

1. ニューラルネットワークにおいて順方向に学習を行い、出力層で損失関数によって正解ラベルとの誤差を求める。
2. 誤差から各出力層ノードについて期待される出力と重要度、誤差を計算する。(局所誤差)
3. 特に重要度の高い前層の入力が、局所誤差に影響を及ぼしているとして重みを調整する。
4. 合成関数の微分によって、さらに前層へと処理を繰り返す。

これによって重みを修正していく学習が可能になっており、また勾配の計算には活性化関数の偏微分が積をとって含まれているため、ネットワークの演算は全体を通して微分可能となっている。

前処理

ニューラルネットワークでは、前処理を行うことで識別性能の向上や学習の高速化を見込むことができる。前処理の手法には以下にあげるようなものがあり、データ全体の分布を考慮して適応する必要がある。

- 正規化：最小値を 0、最大値を 1 とするスケーリング

$$x'_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad (4.15)$$

- 標準化：平均を 0、分散を 1 とするスケーリング

$$x'_i = \frac{x_i - \bar{x}}{\sigma} \quad (4.16)$$

- 対数変換：外れ値による分散を小さくし、0 付近の値を区別しやすくする。

$$x'_i = \ln(x_i) \quad (4.17)$$

ミニバッチ処理

ニューラルネットワークを学習させるにあたって、データを 1 つ 1 つ学習させるわけではない。実際にはミニバッチと呼ばれる、学習データをいくつかまとめて束としたものを一度に学習させる。また、このミニバッチのサイズ、つまりいくつのデータをまとめて束にするかという値のことをバッチサイズという。数値計算を扱うライブラリの多くは、大きな配列の計算を効率よく処理できるよう最適化がなされており、ミニバッチによる学習を行うことで、処理時間を短縮することができる。一方で、ミニバッチのデータは誤差逆伝播においてバッチ内で損失関数の和を用いているため、バッチサイズによって学習結果が異なり、サイズが大きい場合には学習精度が悪くなってしまう可能性もある。

最適化アルゴリズム

ニューラルネットワークの学習では、損失関数の値が最小となるような最適なパラメータを探索する。しかし損失関数のパラメータ空間は非常に複雑であることから、最適化は難しい。以下では、勾配降下法をはじめとする最適化手法について述べる。また、一度の学習で更新するパラメータの度合いを学習率 (learning rate) と呼んでおり、ネットワークの重みなどのパラメータとは異なり、学習率のような人の手で設定する必要のあるパラメータをハイパーパラメータと呼ぶ。

- 勾配降下法

現在のネットワークのパラメータの微分 (勾配) を計算し、その微分の値を手がかりにパラメータの値を徐々に更新する方法を、勾配降下法 (gradient descent method) という。勾配降下法は以下の式 4.18 のように表される。ここで、 \mathbf{W} は更新する重みパラメータを、 L は損失関数を、 η は学習率を表す。

$$\mathbf{W} \leftarrow \mathbf{W} - \eta \frac{\partial L}{\partial \mathbf{W}} \quad (4.18)$$

また、ミニバッチ学習を用いた勾配降下法は特に、確率的勾配降下法 (stochastic gradient descent; SGD) と呼ばれており、現在のニューラルネットワークの最適化法

は主に SGD に基づいて設計されている。しかし、SDG には関数の形状が等方的でない場合、勾配の方向が最終的な最小値と異なるため探索が非効率になるという欠点があり、単純に勾配方向へ進む以外の方法としてさまざまな最適化手法が考案されている。

• モーメンタム

モーメンタム (Momentum) は、それまでの学習における損失関数上で更新ステップの動きを考慮することで SGD の振動を抑えるアルゴリズムである。モーメンタムにおける更新方法は、物理学の速度にあたる変数 v を加え、以下の式のように表される。

$$\mathbf{v} \leftarrow \alpha \mathbf{v} - \eta \frac{\partial L}{\partial \mathbf{W}} \quad (4.19)$$

$$\mathbf{W} \leftarrow \mathbf{W} + \mathbf{v} \quad (4.20)$$

上式における αv が、U の字の斜傾を転がるボールが徐々に減速する運動のような役割を果たし、振動を抑えている。

• AdaGrad

AdaGrad では、モーメンタムと同様に SGD の振動を抑えるが、学習率を減衰させることによってこれを達成するアルゴリズムである。AdaGrad の更新方法は次のような式で表される。

$$\mathbf{h} \leftarrow \mathbf{h} + \frac{\partial L}{\partial \mathbf{W}} \odot \frac{\partial L}{\partial \mathbf{W}} \quad (4.21)$$

$$\mathbf{W} \leftarrow \mathbf{W} - \eta \frac{1}{\sqrt{\mathbf{h}}} \frac{\partial L}{\partial \mathbf{W}} \quad (4.22)$$

ここで、 \odot は要素ごとの積を行うアダマール積を表し、 \mathbf{h} はこれまでの勾配の値の二乗和として保持する。AdaGrad では $\eta \frac{1}{\sqrt{\mathbf{h}}}$ によって学習率のスケールを調整することができる。これによって動いた大きさに合わせてパラメータ毎に学習率の減衰を行うことができる。

• Adam

Adam [25] はモーメンタムの考え方と AdaGrad の考え方を融合させた手法であり、モーメンタムの変数 2つと前のステップまでの学習係数を表す変数 1つの計 3つをハイパー・パラメータにもつ。具体的には、勾配の 1 次、2 次のモーメントを

$$\mathbf{m} \leftarrow \beta_1 \cdot \mathbf{m} + (1 - \beta_1) \frac{\partial L}{\partial \mathbf{W}} \quad (4.23)$$

$$\mathbf{v} \leftarrow \beta_2 \cdot \mathbf{v} + (1 - \beta_2) \left(\frac{\partial L}{\partial \mathbf{W}} \right)^2 \quad (4.24)$$

$$(4.25)$$

と表す。ここで β_1 、 β_2 は、モーメント推定の指數関数的減衰率を表しており、 \mathbf{m} はモーメンタムに用いられる速度を、 \mathbf{v} は AdaGrad に用いられているこれまでの勾配の二乗和に関する値となっている。その上で、さらに学習率を調整するパラメータ ϵ を加

えて

$$\mathbf{W} \leftarrow \mathbf{W} - \eta \frac{\sqrt{1 - \beta_2}}{1 - \beta_1} \frac{\mathbf{m}}{\sqrt{\mathbf{v}} + \epsilon} \quad (4.26)$$

によって重みを更新する。これによって勾配の小さい方向にもスムーズに更新が可能になっており、効率的にパラメータ空間を探索することができる。

- **RAdam**

RAdam [26] は Adam に Warmup による改良を行った最適化手法である。これまで挙げた手法では、学習の初期段階ではサンプル数が少ないために、適応学習率の分散が極めて大きくなり粗悪な局所的最適解に陥ってしまうという課題があった。それに対して Warmup 手法によって学習初段階の学習率を下げ、学習全体の学習率の分散を抑えるようなパラメータ空間の探索が可能となっている。

過学習

一般的に深層学習では重みを更新する学習 (training) と、重みは更新せず学習後のモデルを評価するための推論 (validation) の 2 つが行われ、それぞれに別でデータを用意する必要がある。そして学習においては、ネットワークモデルが学習用データに過度に適合し過ぎてしまい、新しいデータに対する性能が低下してしまう、過学習 (Overfitting) という状態が発生する場合がある。これはデータがパラメータを大量に持ち、表現力が極めて高いモデルである場合や、学習用データの数が少ない場合に多く見られ、モデルがトレーニングデータに含まれるノイズ、または特異的な特徴に過度に適合するために起きてしまうものとされている。そのため、深層学習の実装においては過学習を防止するいくつかの方法を用いることが多く、以下に代表的なものを挙げる。

- **ドロップアウト (Dropout)**

ニューラルネットワークでは全てのノード同士が繋がっており演算を行っていたが、学習時のみニューロンをランダムに消去することで表現力が高すぎることによる過学習を抑制するというアプローチがあり、ドロップアウトという。これは機械学習におけるアンサンブル学習に近く、各エポック (学習の回数) の学習でそれぞれ違うモデルを学習させていると解釈することができる。

- **荷重減衰 (Weight Decay)**

適合しすぎる学習では、重みパラメータが極端に大きい値をとってしまっている場合が多く存在する。そのため重みに制限をかけることで抑制する荷重減衰 (Weight Decay) という手法が存在する。具体例として、重みの二乗ノルム (L2) を損失関数 L に加算する場合には、以下のようないくつかの方法を用いることができる。

$$L = E + \frac{1}{2} \lambda \sum_k (w_k)^2 \quad (4.27)$$

ここで、 E は通常の誤差関数（損失関数）、 λ は正則化の強さを表す正則化パラメータ、 w が重みパラメータを表す。また L2 正則化の他にも、重みの絶対値の和をとる L1 正則化や絶対値最大の成分の絶対値に λ をかける L_∞ などがある。これらによって重みパラメータに強いペナルティが課せられた上で損失関数の値を最小にする学習を行うことができる。

- 重み初期化

荷重減衰と同様に重みを大きくしないための手法として、重みの初期値を定める手法も存在する。重みは小さければ良いというものではなく、0 の場合には誤差逆伝播法において全ての重みの値が同じように更新されてしまうため、正しくは重みを対称的な構造を持たないものにすることが重要となる。その手法にはランダムな重みを振る上で様々な分布に従うものがあり、中でも Xavier Glorot による手法 [27] が一般的に用いられる。この手法では、ニューラルネットにおいて入力層と出力層の重みが、ノード数を n とした時 $\frac{1}{\sqrt{n}}$ の標準偏差を持つガウス関数になるという初期化を行う。

- 正規化 (Batch Normalization)

ミニバッチ毎の入力特微量をスケール 1、平均 0 の分布に正規化を行う手法が存在する。これによって学習の高速化と各層で活性化関数の値が正規化されることで、活性化関数の値のスケールが統一され、勾配がほぼ 0 になってしまい学習が上手くいかなくなる勾配消失問題を抑止することができる。

4.1.3 ディープニューラルネットワーク

多層ニューラルネットワークにおいて、特に図 4.4 のように層の数を多数持つモデルに対しては、構造が深いことからディープニューラルネットワーク（Deep Neural Network; DNN; 深層学習）と呼ぶ。ディープニューラルネットワークの演算には過去に勾配消失のような技術的課題が存在していたが、計算機性能の向上に加え、ReLU 関数を活性化関数に用いるなど計算技術の工夫などによって学習が可能となった。

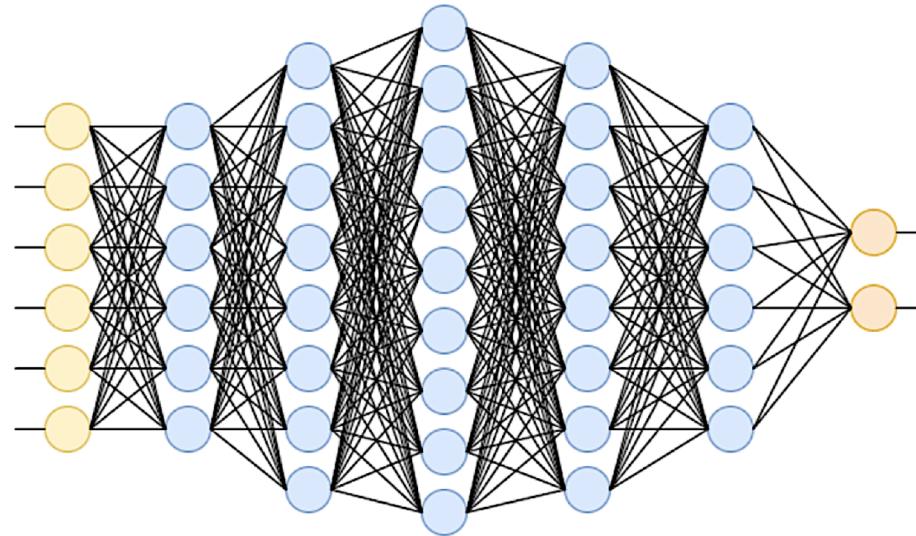


図 4.4: ディープニューラルネットワーク

4.2 グラフニューラルネットワーク

深層学習では数値データのみでなく、画像認識や自然言語処理など様々なデータに対して目覚ましい成果を挙げている。その中で特に近年、グラフで表される構造データに対する研究が非常に盛んになっており、本研究において取り上げるグラフニューラルネットワーク（Graph Neural Network; GNN）[28] もその一つである。グラフ構造データとは、オブジェクトの集合（ノード）を関係（エッジ）で結んだデータのことと、ノードのみが特徴量を持つ場合と、ノードとエッジの両方が特徴量を持つ場合がある。代表的には友人関係や論文の引用関係、化合物などをグラフデータとして構築することは有用であるとされており、データ間の相互関係を用いたモデリング・学習が可能であることを強みとしている。グラフの種類は大きく分けてノードの次元が同じ同種グラフと、異なる次元のデータを扱う異種グラフの 2 種類に分けられる。更に、エッジが方向性を持っている有向グラフと無向グラフ、データが時系列で変化する動的グラフと変化しない静的グラフに分けられる。そしてこのようなグラフデータを扱うニューラルネットをグラフニューラルネットワークという。グラフ構造や演算技術、目的とする課題によって GNN のネットワークモデルの種類は多岐にわたっており、以下では基本的なグラフでの演算や本論文に関連した幾つかの種類のモデルを取り上げる。

4.2.1 メッセージパッシング

GNN のアーキテクチャにおいて、グラフは集合 $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ で構成されており、 \mathcal{V} はノードの集合、 \mathcal{E} はエッジの集合を表す。ニューラルネットワークの場合と同様にノードは特徴量を持っており、エッジも特徴量を持つことができる。図 4.5 に GNN における一般的な演算処理の流れ（メッセージパッシング）を示す。各ノードを h_i とすると、隣接しているノードおよ

びその間のエッジの特徴量を集約し、ノードを更新 \mathbf{h}'_1 する。これによってエッジによって関連づけられたノード間で特徴量を更新していくことができる。エッジが特徴量を持つ場合は、エッジについても更新を行う。これらを数式にまとめると次のようになる。

$$\mathbf{h}_{(i,j)} = f_{\text{edge}}(\mathbf{h}_i, \mathbf{h}_j, x_{(i,j)}) \quad (4.28)$$

$$\mathbf{h}'_i = f_{\text{node}}(\mathbf{h}_i, \sum_{j \in \mathcal{N}_i} \mathbf{h}_{(j,i)}, x_i) \quad (4.29)$$

ここで、 $\mathbf{h}_{(i,j)}$ はエッジを、 x_i はノードの特徴量を、 \mathcal{N}_i は隣接しているノードの集合を表す。

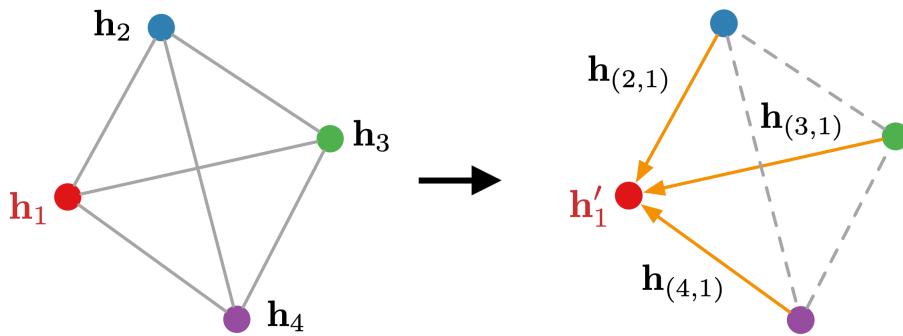


図 4.5: (左) 4 つのノードを持つ全結合グラフニューラルネットワーク。 h_i はノード表現を表す。 (右) メッセージパッシングの処理。 [28]

4.2.2 Graph Convolution Network (GCN)

GCN とは、メッセージパッシングにおいて畳み込み (Convolution) を用いる手法である。一般的に機械学習における畳み込みは、あるフィルターを用いて対象と掛け合わせたものの和をとることで、周辺の情報を含ませることのできる処理であり、画像処理などにおいて広く用いられている。グラフの畳み込みにおいてはスペクトルによるアプローチと空間的なアプローチの 2 つがあり、以下ではそれぞれについて説明する。

Spectral Graph Convolution

スペクトルによる畳み込み [29] は信号処理の考えに基づいたアプローチである。音声などの信号処理においては、関数をフーリエ変換によって周波数成分に変換し、ノイズ除去をして逆変換する。これをグラフデータに置き換えると、グラフラプラシアンの固有ベクトルが張る空間への変換・逆変換となる。グラフ信号を \mathbf{x} とすると、グラフフーリエ変換 $\mathcal{F}(\mathbf{x})$ 、逆グラフフーリエ変換 $\mathcal{F}^{-1}(\mathbf{x})$ は次のように表される。

$$\mathcal{F}(\mathbf{x}) = \mathbf{U}^T \mathbf{x} \quad (4.30)$$

$$\mathcal{F}^{-1}(\mathbf{x}) = \mathbf{U} \mathbf{x} \quad (4.31)$$

ここで、 \mathbf{U} は正規化グラフラプラシアン \mathbf{L}' の固有ベクトル行列を表す。グラフラプラシアン \mathbf{L} は、グラフの隣接を表す隣接行列 \mathbf{A} とグラフの各ノードに接続したノード数を対角成分にもつ次数行列 \mathbf{D} を用いて、 $\mathbf{L} = \mathbf{D} - \mathbf{A}$ で求められる実対称正方行列であり、正規化グラフラプラシアンは $\mathbf{L}' = \mathbf{I} - \mathbf{D}^{\frac{1}{2}} \mathbf{A} \mathbf{D}^{\frac{1}{2}}$ とかける。

これらを用いて、グラフにおける Spectral な畳み込み演算は次のように定義される。

$$\mathbf{g} \star \mathbf{x} = \mathcal{F}^{-1}(\mathcal{F}(\mathbf{g}) \odot \mathcal{F}(\mathbf{x})) \quad (4.32)$$

$$= \mathbf{U}(\mathbf{U}^T \mathbf{g} \odot \mathbf{U}^T \mathbf{x}) \quad (4.33)$$

$\mathbf{g} \star \mathbf{x}$ は畳み込み演算を、 $\mathbf{U}^T \mathbf{g}$ はスペクトルにおける畳み込みのフィルターを表す。学習により更新される集合である \mathbf{g} に焦点を当ててより単純化すると、畳み込みは次のように書くことができる。

$$\mathbf{g} \star \mathbf{x} = \mathbf{U} \mathbf{g} \mathbf{U}^T \mathbf{x} \quad (4.34)$$

上記のように、スペクトルによる畳み込みは一度の更新の中でグラフ構造の一部が全体に影響を与えるような演算であるという特徴を持っている。さらに、行列の次元が固定されてしまうことから異なる構造のグラフ間でパラメータを共有できないことや、行列の固有値分解など計算量が多くなってしまうという問題点がある。

Spatial Graph Convolution

グラフデータにおける畳み込みは空間的なメッセージパッシングの考えに基づいており、グラフ内の 1 つのノードが持っている特徴量に、隣接関係にあるノードの特徴量に重みをかけたものを加えていく。これによりノード自体の特徴量に加え、隣接関係や隣接ノードの特徴量の情報を含んだ演算を行うことができる。メッセージパッシングにおける工夫に応じて様々な畳み込みのモデルが提案されているが、以下では最も一般的なモデルである GraphSAGE (Graph SAmple and aggreGate) [30] について述べる。

GraphSAGE では、隣接から特徴量をサンプリングして集約することで自身のノードを更新する。GraphSAGE における畳み込み演算は式 4.29 と同様に、次の式のように表される。

$$\mathbf{h}_{\mathcal{N}(v)}^k \leftarrow \text{AGGREGATE}_k(\{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\}) \quad (4.35)$$

$$\mathbf{h}_v^k \leftarrow \sigma(\mathbf{W}^k \cdot \text{CONCAT}(\mathbf{h}_v^{k-1}, \mathbf{h}_{\mathcal{N}(v)}^k)) \quad (4.36)$$

$$(\forall k \in \{1, \dots, K\}, \forall v \in \mathcal{V})$$

ここで、 K は集約における隣接の深さを、 \mathbf{W}^k は重みパラメータ行列を、 σ は非線形関数を表しており、1 式目 AGGREGATE で集約を、2 式目 CONCAT で更新を行なっている。これらの学習の処理を図 4.6 に示す。GraphSAGE はあらかじめサンプリングによって、隣接するノード数が異なる場合やグラフ構造が変わった場合にも畳み込み演算を行うことができ、さらに大規模なグラフ演算に対しては計算コストを改善することができる。

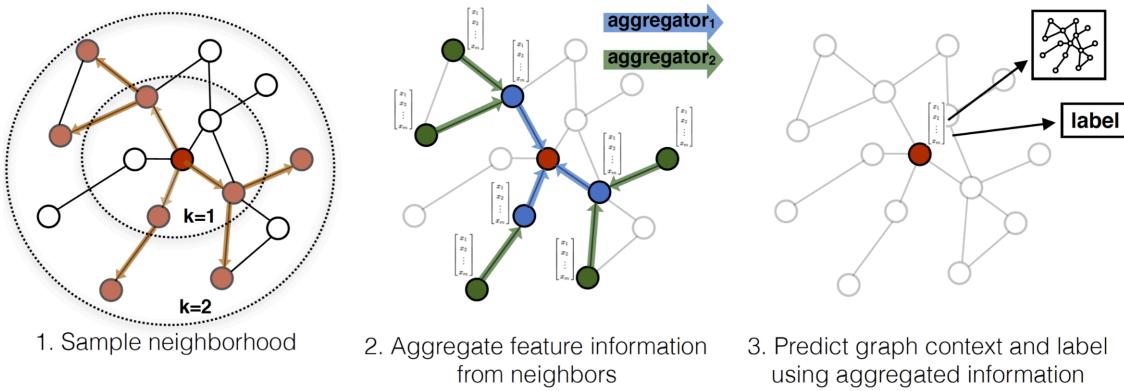


図 4.6: GraphSAGE における処理 (1. サンプリング, 2. 隣接からの集約, 3. 学習結果による推論) [30]

4.2.3 Graph Attention Network (GAT)

深層学習では Attention という、データの中でも特に有益な場所に重み付けを行う手法があり、自然言語処理の分野などにおいてよく用いられている。Attention 機構では各ノードに対して重要度を表す重みを導入し、それらを掛けた和をとることで必要な場所に注目 (attention) を向けることができる。Attention をグラフ学習に適用したものが、Graph Attention Networks (GAT) [31] である。GAT では、重みパラメータ \mathbf{W} に加えて隣接したノードの重要度を表す attention 係数 α_{ij} を導入して、ノードの特徴量の更新を以下のように行う。

$$\mathbf{h} = \sigma \left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W} \mathbf{h}_i \right) \quad (4.37)$$

ここで、 α_{ij} は attention 処理 \mathbf{a} を用いて、次のように表される。

$$\alpha_{ij} = \mathbf{a}(\mathbf{W} \mathbf{h}_i, \mathbf{W} \mathbf{h}_j) \quad (4.38)$$

$$= \frac{e^{\text{LeakyReLU}(\mathbf{a}^T [\mathbf{W} \mathbf{h}_i \| \mathbf{W} \mathbf{h}_j])}}{\sum_{j \in \mathcal{N}_i} e^{\text{LeakyReLU}(\mathbf{a}^T [\mathbf{W} \mathbf{h}_i \| \mathbf{W} \mathbf{h}_j])}} \quad (4.39)$$

GAT では attention 処理に 1 層のニューラルネットワークを用いており、 \mathbf{a} として学習の重みベクトルを用いている。式 4.39においては全ノード間で正規化して確率値を出力するための softmax 関数の適用と、活性化関数 (Leaky ReLU) の適用を行っている。 $(\|)$ はテンソルの concatenate を表す。これらの演算処理のイメージを図 4.7 に示す。GAT は、隣接に任意の重みを割り当てるから次数の異なるノードにも適応可能であり、未知のグラフ構造にも一般化することができる。

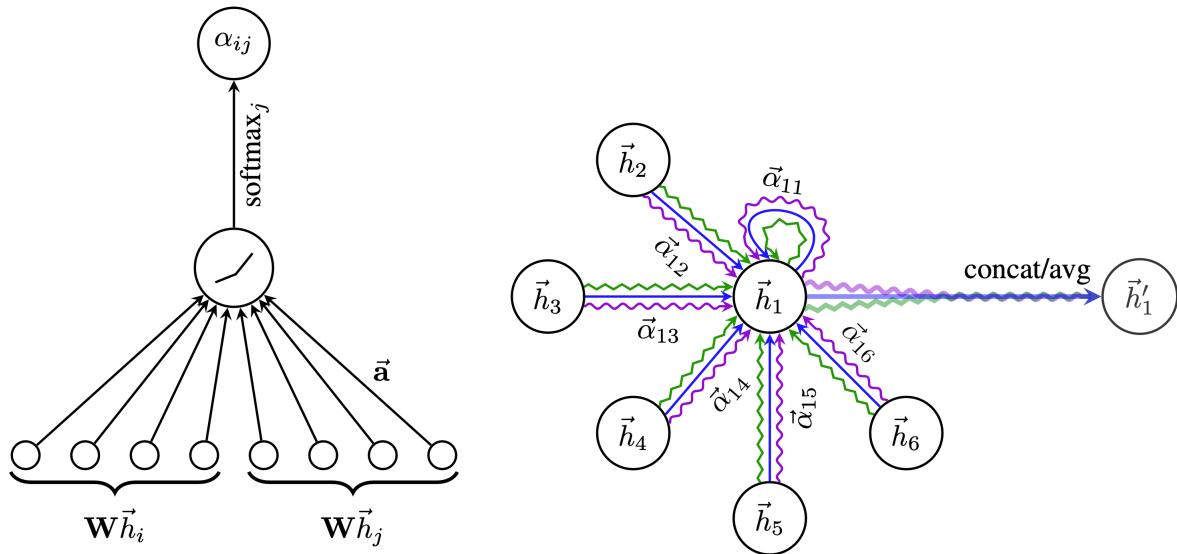


図 4.7: (左) 重みベクトル \mathbf{a} を用いた Attention 処理。 (右) 1 つのノード h_1 に対する隣接ノード $h_{\neq 1}$ の Attention と、ノード特徴量の更新 h'_1 [31]

第 5 章

深層学習を用いたジェットフレーバー識別

本章では、深層学習を用いて開発したジェットフレーバー識別アルゴリズムについて述べる。まず 5.1 節では、フレーバー識別に関する事象の詳細についてや、現在ジェットフレーバー識別に用いられている LCFIPlus について述べる。次に 3.2 節で、今回のアルゴリズムの評価において使用した MC シミュレーションデータについて述べる。またシミュレーションデータには深層学習における精度を上げるために前処理を行なっており、これについても述べる。そして 5.3 節、5.4 節それぞれでディープニューラルネットワーク、グラフニューラルネットワークによる実装について述べ、5.5 節で LCFIPlus と比較を行って学習の性能について議論する。

5.1 イベントサンプルと先行研究

本研究では、MC イベントジェネレーターである whizard による事象生成を行い、ILD 検出器のフルシミュレーションを使用したデータを用いてフレーバー識別を行った。シミュレーションデータについての詳細を表 5.1 に示す。

| | |
|-------------------------|--|
| イベントジェネレーター | whizard |
| シミュレーション | ILD Full simulation |
| 検出器モデル | ILD_15_250GeV_v02-02 |
| 反応過程 | $e^+ e^- \rightarrow \nu \bar{\nu} h \rightarrow \nu \bar{\nu} b\bar{b}/c\bar{c}/q\bar{q}$ ($q = u, d, s$) |
| 重心系エネルギー | 250 GeV |
| Initial State Radiation | あり |
| ビーム偏極 | なし |

表 5.1: シミュレーションデータのパラメータ

5.1.1 LCFIPlus におけるフレーバー識別

LCFIPlus のフレームワークでは、フレーバー識別は崩壊点検出、ジェットクラスタリングの後に行われる。フレーバー識別は、飛跡や崩壊点の特徴量をもとに ROOT の TMVA パッケージにおける Boosted Decision Trees (BDTs) を用いて識別が行われる。この Boosted Decision Trees は、複数のモデルを組み合わせて全体的な性能を向上させるアンサンブル学習 (Boosting) と、入力された特徴量に基づいてデータをクラスに分割していく決定木 (Decision Tree) の手法を組み合わせたものであり、複数の弱い決定木モデルを組み合わせて決定木の精度を向上させる機械学習手法である。データは、初めに再構成された崩壊点の数で場合分けが行われ、次ページに示す、表 5.2 にある入力変数をもとに、最終的に各ジェットを $b/c/uds$ の 3 つに分類する学習を行う。

| 変数名 | 説明 |
|--------------|---|
| trk1d0sig | d_0 の significance が最も高い飛跡の d_0 (d_0 : xy 平面射影における IP と飛跡の距離, Appendix を参照) |
| trk2d0sig | d_0 の significance が 2 番目に高い飛跡の d_0 |
| trk1z0sig | z_0 の significance が最も高い飛跡の d_0 (z_0 : sz 平面射影における IP と飛跡の距離, Appendix を参照) |
| trk2z0sig | z_0 の significance が 2 番目に高い飛跡の d_0 |
| trk1pt | d_0 が最も大きい飛跡の横方向運動量 |
| trk2pt | d_0 が 2 番目に大きい飛跡の横方向運動量 |
| jprobr | 全飛跡を用いた $r\phi$ 平面での結合確率 |
| jprobr5sigma | 5σ 以上のパラメータを持つ全飛跡を用いた $r\phi$ 平面での結合確率 |
| jprobz | 全軌跡を用いた z 軸射影での結合確率 |
| jprobz5sigma | 5σ 以上のパラメータを持つ全飛跡を用いた z 軸射影での結合確率 |
| d0bprob | 全飛跡に対して b, c, uds フレーバーの d_0 分布を用いた d_0 の b -クォーク確率の積 |
| d0cprob | 全飛跡に対して b, c, uds フレーバーの d_0 分布を用いた d_0 の c -クォーク確率の積 |
| d0qprob | 全飛跡に対して b, c, uds フレーバーの d_0 分布を用いた d_0 の uds -クォーク確率の積 |
| z0bprob | 全飛跡に対して b, c, uds フレーバーの z_0 分布を用いた d_0 の b -クォーク確率の積 |
| z0cprob | 全飛跡に対して b, c, uds フレーバーの z_0 分布を用いた d_0 の c -クォーク確率の積 |
| z0qprob | 全飛跡に対して b, c, uds フレーバーの z_0 分布を用いた d_0 の uds -クォーク確率の積 |
| nmuon | ミューオンの数 |
| nelectron | 電子の数 |
| trkmass | d_0/z_0 が 5σ を超える全飛跡の質量 |
| vtxprob | 崩壊点に関連した全飛跡を結合した時の崩壊点確率 |
| vtxlen1 | ジェット内の 2 番目の崩壊点の崩壊長 |
| vtxlen2 | ジェット内の 3 番目の崩壊点の崩壊長 |
| vtxlen12 | ジェット内の 2 番目の崩壊点と 3 番目の崩壊点の距離 |
| vtxsig1 | vtxlen1 の significance |
| vtxsig2 | vtxlen2 の significance |
| vtxsig12 | vtxlen12 を 2 番目の崩壊点と 3 番目の崩壊点の共分散行列の和の誤差で割った値 |
| vtxdirang1 | 運動量と 2 番目の崩壊点の変位との間の開き角 |
| vtxdirang2 | 運動量と 3 番目の崩壊点の変位との間の開き角 |
| vtxmult1 | 2 番目の崩壊点に含まれる飛跡数 |
| vtxmult2 | 3 番目の崩壊点に含まれる飛跡数 |
| vtxmult | secondary vertex を構成するために用いられる飛跡の数 |
| vtxmom1 | 2 番目の頂点に結合された全飛跡の運動量のベクトル和 |
| vtxmom2 | 3 番目の頂点に結合された全飛跡の運動量のベクトル和 |
| vtxmass1 | 飛跡の四元運動量の和から計算される 2 番目の崩壊点質量 |
| vtxmass2 | 飛跡の四元運動量の和から計算される 3 番目の崩壊点質量 |
| vtxmass | secondary vertex を構成する全 飛跡の四元運動量の和から計算される崩壊点質量 |
| vtxmasspc | primary/secondary vertex の誤差行列で許容可能な pt 補正を行なった崩壊点質量 |
| vtxprob | 崩壊点が形成される確率 |

表 5.2: LCFIPlus におけるフレーバー識別の入力変数

5.2 ディープニューラルネットワークによる実装

本節では、ディープニューラルネットワーク（多層ニューラルネットワーク）によるフレーバー識別の実装について述べる。

5.2.1 実装目的

フレーバー識別では、シグナル効率に対して排除できる背景事象の割合が多くなるほど、ヒッグスなどの物理解析において有利であるため、その性能が非常に重要である。LCFIPplusにおいて実装されている BDTs は、崩壊点やジェットの特徴量の場合分けによって分類を行うため、ツリー構造が視覚化されており解釈が容易である。一方で、BDTs の場合分けでは識別理由が解読可能であり見通しが良いが、ILC のフレーバー識別のようなデータが高次元で複雑なタスクに対して精度を出すには、より複雑な表現学習の方が適している可能性がある。さらに BDTs にはノイズに敏感である点や、ツリー数が多くなると過学習を起こしやすくなる点など問題点もある。そこで本研究では、より最適化したモデルの獲得によるフレーバー識別の更なる性能向上を目的に、ディープニューラルネットワークによる実装を行った。

5.2.2 入力変数とネットワークアーキテクチャ

学習には ILD フルシミュレーションにおける $e^+e^- \rightarrow \nu\bar{\nu}h$ 事象を 100 万イベント用いた。そして入力変数には、LCFIPplus において用いた変数（表 5.2）と同様の変数を使用した。また、学習値に対しては前処理を行い、0 付近の値が多い変数や値幅が大きい変数が存在したため、各変数に対して対数変換や正規化を実行した。また出力も LCFIPplus と同様に b フレーバー、 c フレーバー、 uds フレーバーとした。

ネットワークには図 5.1 のように、レイヤー間で全てのノード同士が結合した全結合層を用いたニューラルネットワークを採用した。さらに、過学習を抑制するためにバッチ正規化を各全結合層の後に行い、活性化関数には勾配消失への対策として全て LeakyReLU 関数を使用した。損失関数には分類問題に適した交差エントロピーを用い、最適化アルゴリズムには RAdam を学習率 0.01 で採用し、学習を調整するパラメータは $\beta_1 = 0.89, \beta_2 = 0.85, \text{eps} = 10^{-7}$ 、weight decay を 0.0001 とした。また学習率は 25 エポックごとに 0.1 倍させ、学習の安定化を目的に学習率を減衰させた。表 5.3 に最適化によって求めた学習におけるハイパーパラメータをまとめる。また、上記のネットワークは深層学習ライブラリである PyTorch を用いて実装を行なった。PyTorch は Python で使用可能なライブラリであり、高速な数値計算を行うための GPU サポートや、動的な計算グラフを導入しているため実行時コンパイル（Define-by-Run）という特徴を持っている。

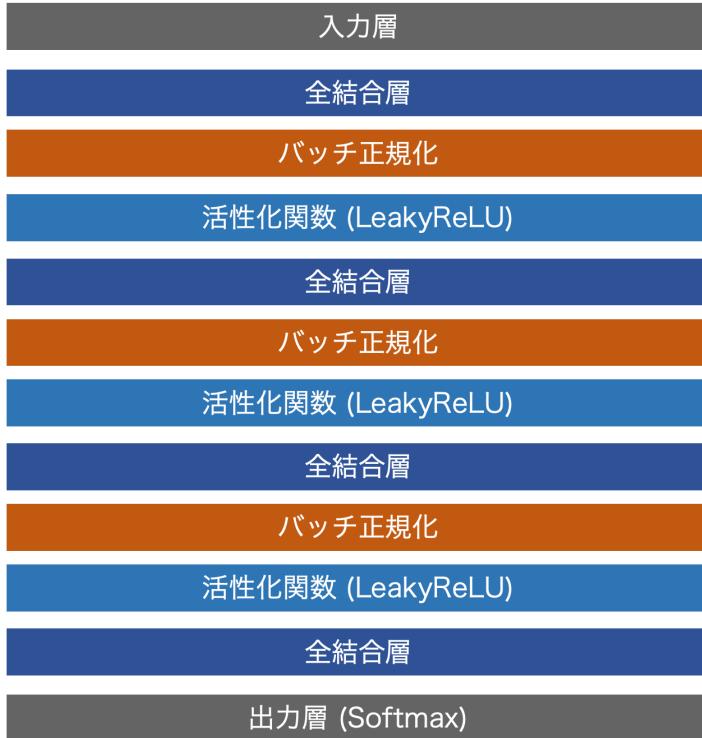


図 5.1: フレーバー識別のためのディープニューラルネットワークの概略図

| | |
|--------------------|---|
| ノード数 | (124, 124, 124) |
| 活性化関数 | LeakyReLU 関数 (0.001) |
| 損失関数 | 交差エントロピー |
| 最適化アルゴリズム | RAdam($\beta_1 = 0.89, \beta_2 = 0.85, \text{eps} = 10^{-7}$) |
| weight decay | 0.0001 |
| 学習率 | 0.01 (25 エポックあたり 0.1 倍) |
| エポック数 (全データでの学習回数) | 100 |
| バッチサイズ | 1024 |

表 5.3: ディープニューラルネットワークにおけるハイパーパラメータ

5.2.3 ハイパーパラメータの最適化

学習におけるハイパーパラメータ (表 5.3) の一部にはチューニングを行った。ハイパーパラメータのチューニングにおいて最適化する対象はブラックボックスであることが多く、ベイズ最適化 [32] のような目的関数を推定しつつ大域的最適解を探索する手法がよく用いられる。ベイズ最適化法は、ガウス過程回帰によって未知の関数をデータから学習し、可能な限り少ない試行数で最適解の推定を行う。ガウス過程回帰とは、入力変数 x と推定精度を表す y との関

係性関数 $y = f(x)$ を推定する手法であり、入力の集合に対して出力の集合が多変量ガウス分布に従うとして推定の幅を算出し、次の候補点を決定している。本アルゴリズムのハイパーパラメータのチューニングでは、Preferred Networks 社のフレームワーク Optuna [33] を使用した。

チューニングを行なったパラメータとその値の範囲を表 5.4 に示す。パラメータの最適化試行回数は 10 回で、図 5.2 に評価関数の値の経過を示す。今回の最適化では損失関数を評価関数に用い、値が最小になるような最適化を行った。図 5.3 は 2 つのパラメータに対する評価関数の 2 次元ヒストограмであり、中間層のノード数が 250 程度で極大値を取っている。また、図 5.4 に各パラメータの評価関数に与える重要度を示す。

| 変数名 | 最適化範囲 |
|--------------|--------------------------------------|
| 中間層のノード数 | [32, 64, 128, 256, 512, 1024] |
| ドロップアウト | 0.1 ~ 0.9 |
| LeakyReLU | [0.00001, 0.0001, 0.001, 0.01, 0.1] |
| 学習率 | [0.00001, 0.0001, 0.001, 0.1] |
| β_1 | 0.5 ~ 0.9 |
| β_2 | 0.5 ~ 0.9 |
| eps | $[10^{-9}, 10^{-8}, 10^{-7}]$ |
| weight decay | [0.000001, 0.0001, 0.001, 0.01, 0.1] |

表 5.4: 最適化を行なったハイパーパラメータの一部とその値の範囲。

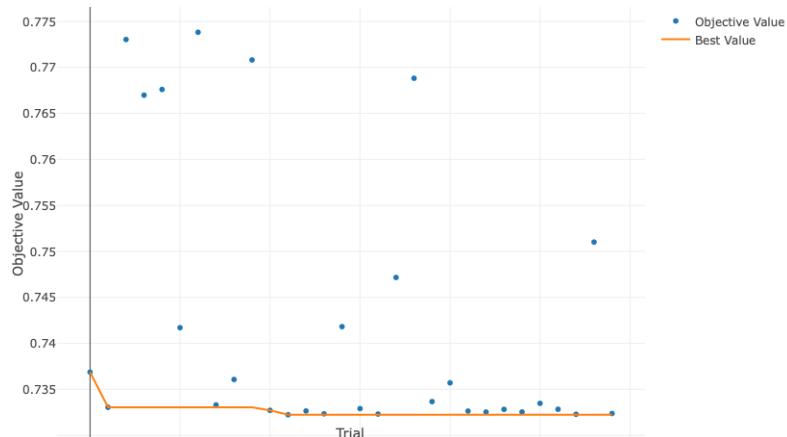


図 5.2: 各試行における評価関数（損失関数）の値。横軸が試行順番、縦軸が評価関数の値を示しており、評価関数の値が最小になるようなアルゴリズムを実行している。

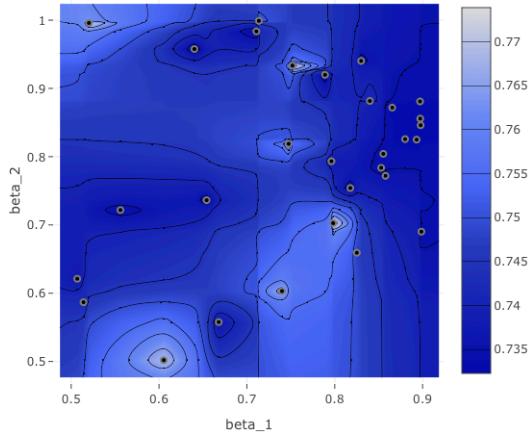


図 5.3: RAdam の学習に対するパラメータによる評価関数の 2 次元ヒストグラム。横軸が β_1 の値を、縦軸が β_2 の値を、3 次元方向が評価関数の値を表している。

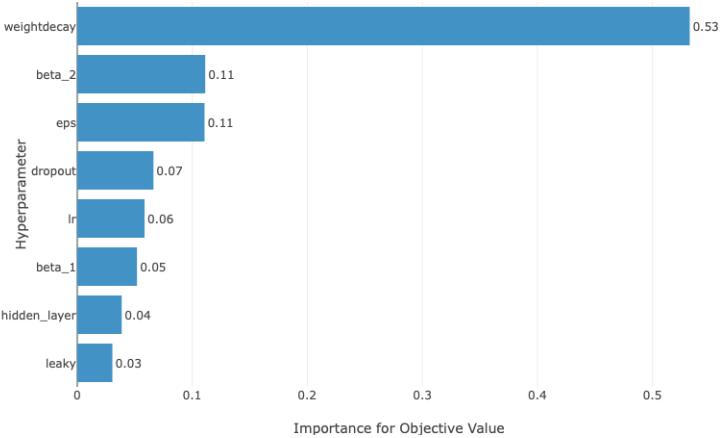


図 5.4: 各パラメータの重要度。表 5.4 に挙げたパラメータの重要度をそれぞれ示している。

5.2.4 学習評価

学習における損失関数の値と識別精度を図 5.5 に示す。100 エポック の学習によって、損失関数の値は降下したのちおおよそ安定しており、学習精度はおよそ 82.5% となっている。また、図 5.6 に混合行列を示す。混合行列とは、分類問題で出力された結果をまとめた表であり、縦軸 (行) が実際の答えを横軸 (列) が学習結果を表していて、実際の答えごと (行) に正規化されている。そのため、左上から右下の対角成分が実際の答えと学習結果が一致している割合を表しており、図 5.5 における学習精度は (学習精度) = (対角成分の和)/(全成分の和) として求められる。図 5.6 より、ディープニューラルネットワークでは c ジェットについては uds ジェットと識別しにくいことがわかる。

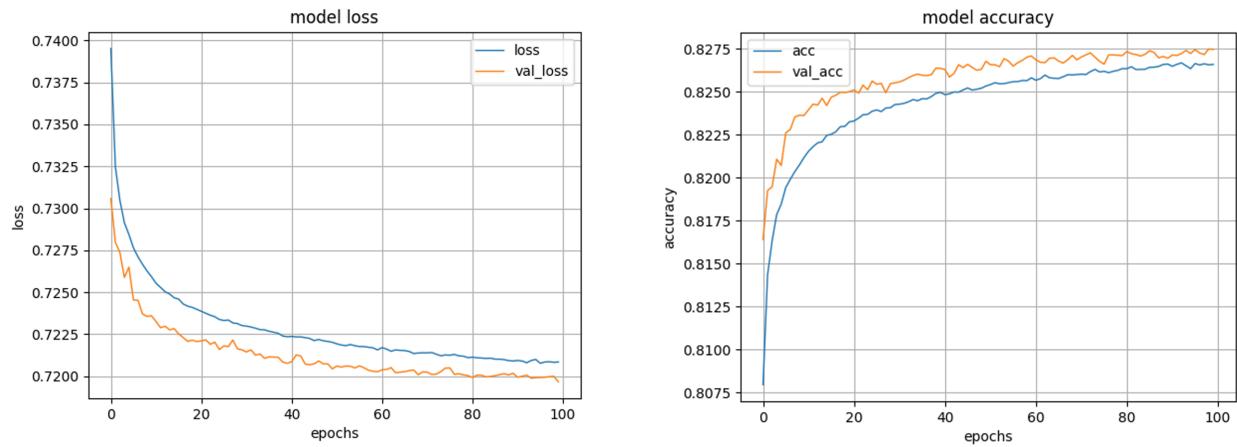


図 5.5: (左) 学習の経過における損失関数。(右) 学習の経過における学習精度。横軸が epoch 数、縦軸がそれぞれの値を示しており、青線が学習データでの値を、橙線が検証用データでの値を表している。

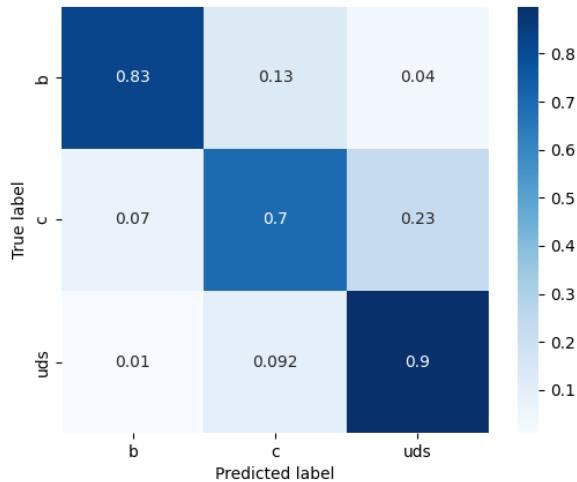


図 5.6: ディープニューラルネットワークの学習における混合行列。縦軸が実際の答えを、横軸が学習結果を表している。

次に、LCFIPlus との比較を示す。本アルゴリズムのモデルの出力値は 1 つのジェットに対してそれぞれのフレーバーである確率値が得られる。そして、物理解析においてはフレーバー識別の効率が重要であり、識別効率 (= フレーバーをある確率値で決定した場合の正しい認識率) に対して背景事象と誤認識してしまう割合が少なくなることが望ましい。このとき識別効率は、正しく識別された目的のジェットの数/識別したい目的のフレーバーのジェット数で定義される。また誤認率は誤って識別された対象のジェットの数/識別したいフレーバーの背景のジェット数として定義される。図 5.7, 5.8 は b/c ジェットの識別効率を示している。b ジェットの識別効率は LCFIPlus と比較して劣っており、特に識別効率 80% における識別割

合は、LCFIPlus が c ジェットの識別割合が 7.3%、 uds ジェットの識別効率が 0.74% であるのに対して、ディープニューラルネットの c ジェットの識別割合は 8.8%、 uds ジェットの識別効率は 2.34% と、 uds ジェットの誤認率は 3 倍近く高くなっている。一方で c ジェットの識別効率は識別効率 80% において、LCFIPlus が b ジェットの識別割合が 22%、 uds ジェットの識別効率が 24% であるのに対して、ディープニューラルネットの b ジェットの識別割合は 13%、 uds ジェットの識別効率は 28% 程度と、 b ジェットの背景割合は半分近くになっている。本来フレーバー識別は b 、 c クォークの寿命が長いという特徴を活かして識別するという発想にあるため、崩壊点に関する変数で線形にモデリングした BDTs では、 c ジェットよりも b ジェットの方が識別効率が高くなっている。一方でディープニューラルネットワークでは、関係性が逆転している。機械学習アルゴリズムの性能はデータの特性やタスクの種類に大きく依存するため、同じデータを用いていてもアルゴリズムが異なる場合、同じタスクに対する学習特性は異なることがある。今回の場合では c ジェットの識別において、ディープニューラルネットワークでデータ内の複雑な関係を処理できたため、このような結果になったと考えることができる。

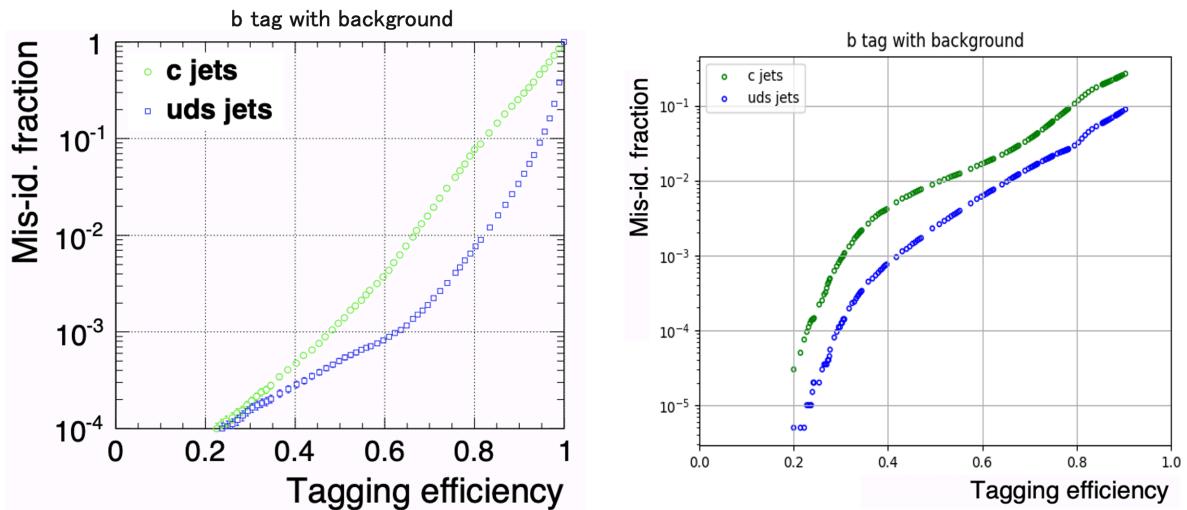


図 5.7: LCFIPlus(左) とディープニューラルネットワーク(右)による b フレーバージェットの識別効率の比較。緑: b ジェットに対する c ジェットの識別効率、青: b ジェットに対する uds ジェットの識別効率

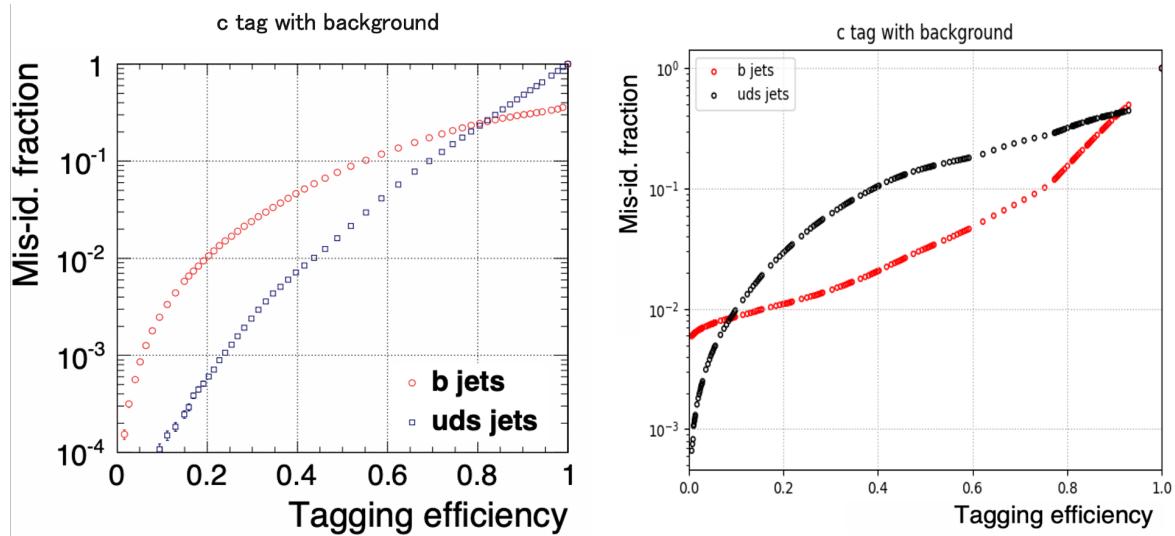


図 5.8: LCFIPlus(左) とディープニューラルネットワーク(右)による c フレーバージェットの識別効率の比較。赤： c ジェットに対する b ジェットの識別効率、黒： c ジェットに対する uds ジェットの識別効率

| 識別効率=0.8 | 背景ジェット | 誤認率 | |
|----------|------------|----------|--------|
| | | LCFIPlus | DNN |
| b ジェット | c ジェット | 0.073 | 0.088 |
| | uds ジェット | 0.0074 | 0.0234 |
| c ジェット | b ジェット | 0.22 | 0.13 |
| | uds ジェット | 0.24 | 0.28 |

表 5.5: 識別効率 (Tagging Efficiency) が 0.8 の時のジェット誤認率 (Mis-id fraction)

5.3 グラフニューラルネットワークによる実装

本節では、グラフニューラルネットワークによるフレーバー識別の実装について述べる。

5.3.1 実装目的

5.2節におけるディープニューラルネットワークの実装では、一部で精度の向上が見られたものの顕著な識別精度の改善はなかった。そこで、ILCのフレーバー識別により最適化した学習を行うために、より高い表現力を持つグラフニューラルネットワークによる実装を考えた。ILCにおけるジェットの物理現象の振る舞いはトポロジカルな構造を持っており、またジェット内の飛跡同士は崩壊点を共有するものが存在している。そのため、グラフデータによってトポロジカルな構造を再現し、データ間の相互関係を活かした学習を行うことで、高い表現力によるアルゴリズムモデルの最適化を目指した。

さらにグラフニューラルネットワークの実装では、データをグラフ構造化する上でジェットの内部構造を理解するようなモデルを構築する必要があり、これを補助学習としたアルゴリズムの統合が可能であると考えた。つまり、これまでのようなプロセスを分離した高度なアルゴリズムではなく、1つのアルゴリズム内で飛跡に関する入力変数のみから、飛跡の分類や崩壊点の予想と同時にジェットフレーバーの識別を行うことを可能とするものである。これによって、情報損失を少なくすることで性能の向上を目指すとともに、アルゴリズム調整の単純化や、飛跡の分類結果など将来的に他のアプリケーションに利用可能な情報の生成までもを期待することができる。

5.3.2 飛跡によるグラフデータセット

ILCにおけるジェットの振る舞いを再現するため、本アルゴリズムでは1つのジェットに対して、飛跡をノードとし全ノードがエッジによって結ばれる1つの同種グラフを構築した(図5.9)。各ノードは表5.5に挙げる特徴量を入力変数として持ち、エッジは特徴量を持たないものとした。ノードの特徴量には、飛跡に関するパラメータ(Appendixを参照)を用いた。入力データの生成のため、まず250 GeVのILDフルシミュレーションにおける $e^+e^- \rightarrow \nu\bar{\nu}h$ 事象を24万イベント使用し、飛跡再構成とジェットクラスタリングを行った。またシミュレーションにおける答えをもとに次の3つの学習に対して次のような答えラベルを準備した。ノードの答えラベルは飛跡がどのフレーバージェットのどのvertexから来たものであるのかを表しており、PVはprimary vertex由来の飛跡を、SVBBはbフレーバージェットのsecondary vertex由来の飛跡を、TVCCはbフレーバージェットのtertiary vertex由来の飛跡を、SVCCはcフレーバージェットのsecondary vertex由来の飛跡を、それ以外がOthersのラベルを持つ。また、エッジの答えラベルは飛跡対が同じ崩壊点を共有するかのバイナリ形式、グラフの答えラベルはb,c,udsのうちジェットがどのフレーバーであるのかを答えとし

て設定した。各学習の答えラベルについて表 5.7~表 5.9 にまとめる。また、入力変数には前処理を行った。さらにデータ数は答えラベル間で偏りがあり、ノードに関するラベルあたりのデータ数 5.10 は Primary Vertex を由来とする飛跡が突出して多かったため、比の逆数を重みとして損失関数で考慮することで対応した。また、本アルゴリズムの実装は PyTorch[34] とグラフニューラルネットワーク向けのグラフ構造を扱うための PyTorch ライブラリである PyTorch Geometric[35] を用いて行なった。

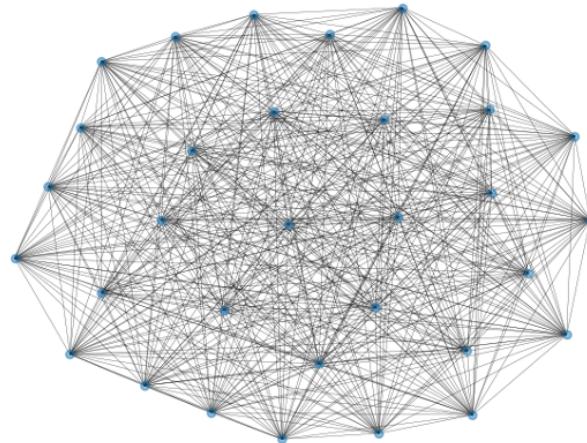


図 5.9: 今回の学習に用いたグラフデータの一例。ノードは飛跡を、グラフ全体が 1 つのジェットに対応する。

| 変数名 | 説明 |
|----------------|-------------------------|
| d_0 | xy 平面射影における IP と飛跡の距離 |
| ϕ | xy 平面射影における飛跡軌道の方位角 |
| ω | xy 平面射影における飛跡の曲率 |
| z_0 | sz 平面射影における IP と飛跡の距離 |
| $\tan \lambda$ | sz 平面射影における dz/ds |
| $\sigma(d_0)$ | d_0 のフィッティングにおける誤差 |
| $\sigma(z_0)$ | z_0 のフィッティングにおける誤差 |

表 5.6: 各ノード (飛跡) が持つ特徴量。詳細については Appendix を参照。

| ラベル | 説明 |
|--------|---------------------------------------|
| PV | primary vertex 由来の飛跡 |
| SVBB | b フレーバージェットの secondary vertex 由来の飛跡 |
| TVCC | b フレーバージェットの tertiary vertex 由来の飛跡 |
| SVCC | c フレーバージェットの secondary vertex 由来の飛跡 |
| Others | 上記の崩壊点を持たない飛跡 |

表 5.7: ノード分類の答えラベル

| ラベル | 説明 |
|---------------|-----------------------|
| Connected | エッジの結ぶ飛跡対同士が崩壊点を構成する |
| Not-Connected | エッジの結ぶ飛跡対同士が崩壊点を構成しない |

表 5.8: リンク予測の答えラベル

| ラベル | 説明 |
|------------|-----------------|
| $b\bar{b}$ | b フレーバージェット |
| $c\bar{c}$ | c フレーバージェット |
| $q\bar{q}$ | uds フレーバージェット |

表 5.9: グラフ分類の答えラベル

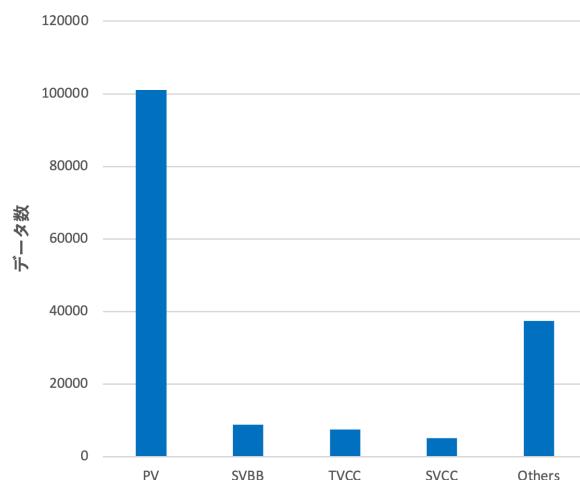


図 5.10: ノードラベル毎のデータ数

5.3.3 ネットワークアーキテクチャ

上述の通り、本アルゴリズムではフレーバー識別に加えて、補助学習を導入する。具体的にはノード分類とリンク予測、グラフ分類の3つのタスクを同時に行うネットワークで、図5.11のようなモデルを構築した。

はじめに、飛跡の変数をより多次元ベクトル化し潜在表現を得るために全結合層を設置する。その後、Graph Attention Network (GAT) を3層設置し、各 GAT 層の後にはバッチ正規化と LeakyReLU 関数の活性化関数を置く。そして学習をノード分類、リンク予測、グラフ分類の3つに分岐させる。ノード分類では、飛跡の由来となる崩壊点の種類によって5つの出力が全結合層によって設計されている。リンク予測では、エッジの隣接行列を取得してノード間に隣接があるかないかを判断する2分類問題を行っており、これはエッジが崩壊点となりうるのかの判断に等しい。グラフ分類では、poolingによってグラフ全体の特徴量を1つの次元に置き換え、各フレーバーらしさを出力する設計になっている。

グラフ層の実装には、空間的な畳み込みを行うことができる GraphSAGE と GAT を検討した。どちらもグラフ分類では近い精度を達成することができたが、GraphSAGE ではエッジの分類に向いた学習を効果的に行うことができない点や、アテンション機構によって注目したい崩壊点を取り上げることができる点を理由に、GAT を採用した。GAT の学習は4.2.3節で述べた通りであり、本アルゴリズムのグラフデータは全結合のグラフニューラルネットワークを構築しており、ノードに対するアテンション α はエッジの隣接関係を用いて式4.39のように学習される。このようなアテンション機構によってグラフ全体、すなわちジェットフレーバーの識別を補助することを目的に、ノードの識別の補助学習を導入した。また、実際に飛跡対が崩壊点を共有している場合、その隣接関係は物理的に重要なものとなるため、その情報を学習に活かすことを目的にリンク予測を補助学習に加えた。また、3つの学習の出力層に置いた全結合層はそれぞれ独立に実行される。

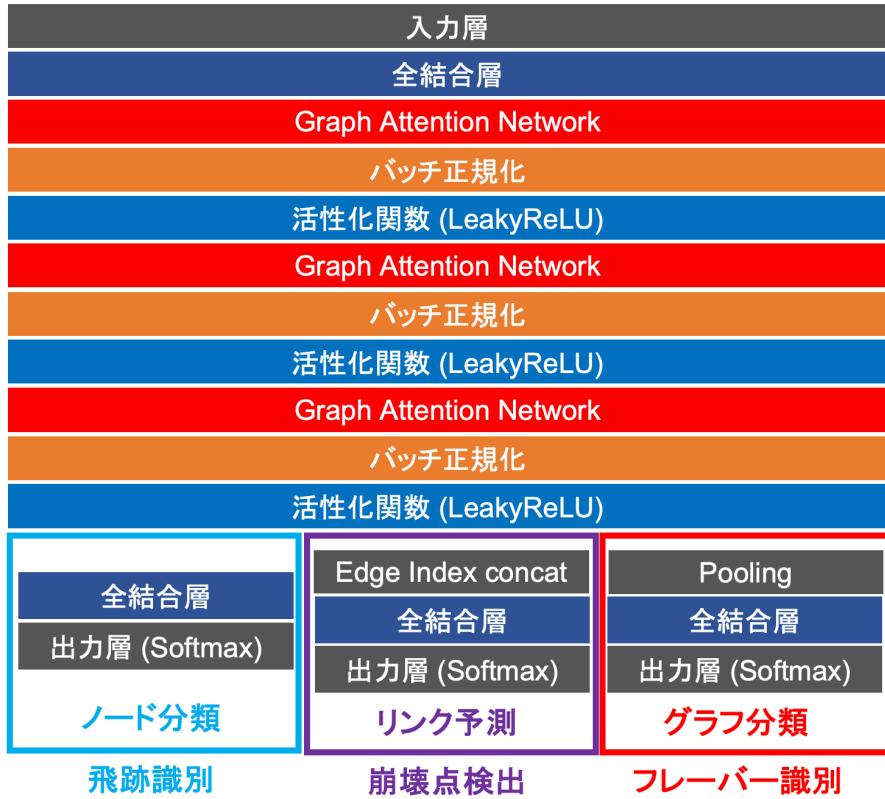


図 5.11: フレーバー識別のためのグラフデータを用いたネットワークの概略図

また本アルゴリズムでは3つの異なる学習を同時に行う必要があり、目的によって相対的な学習のしやすさが異なるため、次のような損失関数をデザインした。

$$L_{\text{total}} = w_{\text{node}} L_{\text{node}} + w_{\text{edge}} L_{\text{edge}} + L_{\text{graph}} \quad (5.1)$$

L はそれぞれの損失関数を、 w_{node} と w_{edge} はそれぞれノード、エッジの損失関数にかける重みを表す。グラフ分類と同程度の損失関数に収束するために、ノード識別とリンク予測における損失関数に重み付けを行った。

また、学習におけるハイパーパラメータにはベイズ最適化を行った上で、表 5.10 に挙げるものを用いた。最適化アルゴリズムには、Adam に徐々に学習率を上げる Warmup 機構を加えた RAdam による学習が最も精度が高くなった。

| | |
|-------------------|--|
| 特徴量の数 | (512, 512, 512) |
| アテンションヘッド数 | 16 |
| 活性化関数 | LeakyReLU 関数 (0.001) |
| 損失関数 | 交差エントロピー |
| 最適化アルゴリズム | RAdam ($\beta_1 = 0.8, \beta_2 = 0.9, \text{eps} = 10^{-8}$) |
| 学習率 | 0.01 |
| Weight decay | 0.0001 |
| w_{node} | 3.0 |
| w_{edge} | 1.0 |
| エポック数 | 100 |
| バッチサイズ | 128 |

表 5.10: グラフニューラルネットワークにおけるハイパーパラメータ

5.3.4 ハイパーパラメータの最適化

ネットワークのハイパーパラメータは、ベイズ最適化法によってチューニングを行った。チューニングしたパラメータは表 5.10 に加え、RAdam における学習率のハイパーパラメータ $\beta_1, \beta_2, \text{eps}$ などについて実行した。チューニングにはディープニューラルネットワークの実装におけるものと同じ Optuna ライブラリを用いたが、評価関数として検証用データの損失関数のうち L_{graph} のみを用いて、最小になるような最適化(図 5.12)を行った。今回の学習において重要な損失関数の割合 w_{node} (図中の α)、 w_{edge} (図中の β)に関する 2 次元ヒストグラムの値は、図 5.13 のように α が 0.5、 β が 3 のあたりで評価関数の値が極小値をとった。また、今回の最適化に使用したパラメータの重要度は RAdam における学習率と学習のパラメータ β_2 が最大となった(図 5.14)。

| 変数名 | 最適化範囲 |
|---------------------------|---|
| 中間層のノード数 | [32, 64, 128, 256, 512, 1024] |
| アテンションヘッドの数 | 1 ~ 20 |
| ドロップアウト | 0.1 ~ 0.9 |
| LeakyReLU | [0.00001, 0.0001, 0.001, 0.01, 0.1] |
| β_1 | 0.5 ~ 0.9 |
| β_2 | 0.5 ~ 0.9 |
| eps | $[10^{-9}, 10^{-8}, 10^{-7}]$ |
| 学習率 | [0.00001, 0.0001, 0.001, 0.01, 0.1] |
| weight decay | [0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1] |
| $w_{\text{node}}(\alpha)$ | 0.0 ~ 5.0 |
| $w_{\text{edge}}(\beta)$ | 0.0 ~ 5.0 |

表 5.11: 最適化を行ったハイパーパラメータとその値の範囲

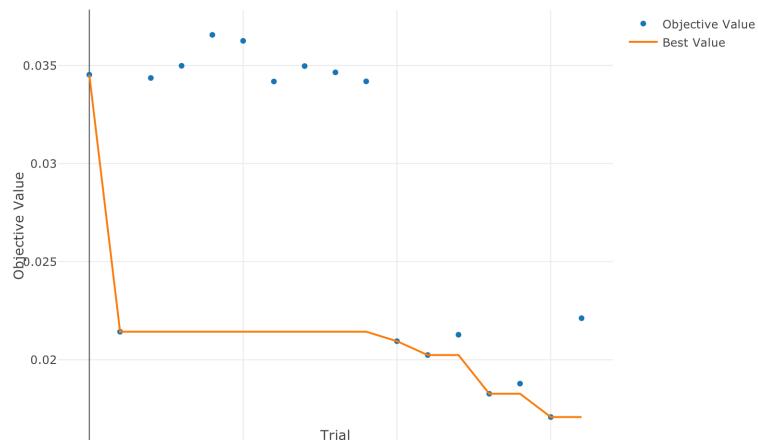


図 5.12: 各試行における評価関数の値。横軸が試行順番、縦軸が評価関数の値を示しており、評価関数の値が最小になるようなアルゴリズムを実行している。

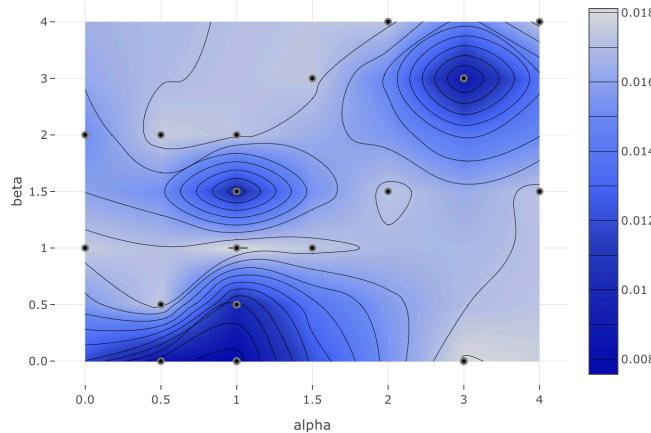


図 5.13: w_{node} (図中の α), w_{edge} (図中の β)に対する評価関数の2次元ヒストグラム。横軸が α の値を、縦軸が β の値を、3次元方向が評価関数の値を表している。

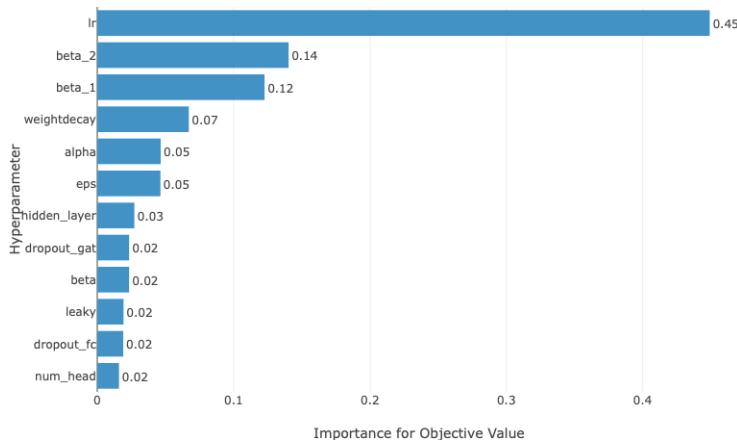


図 5.14: 表 5.11 の各パラメータの重要度。特に lr は RAdam における学習率を、 β_2 は学習の進行度のパラメータを表す。

5.3.5 学習評価

はじめに、学習全体の経過を図 5.15 に示す。学習の経過を通して損失関数は減少しており、特に 80 エポック付近で急激な降下を見せた。しかしそれ以降は過学習が進んでしまい、損失関数の値は安定化させることが難しかった。また、ノード分類、リンク予測、グラフ分類それぞれの精度においても、学習の経過を通して向上が見られた。

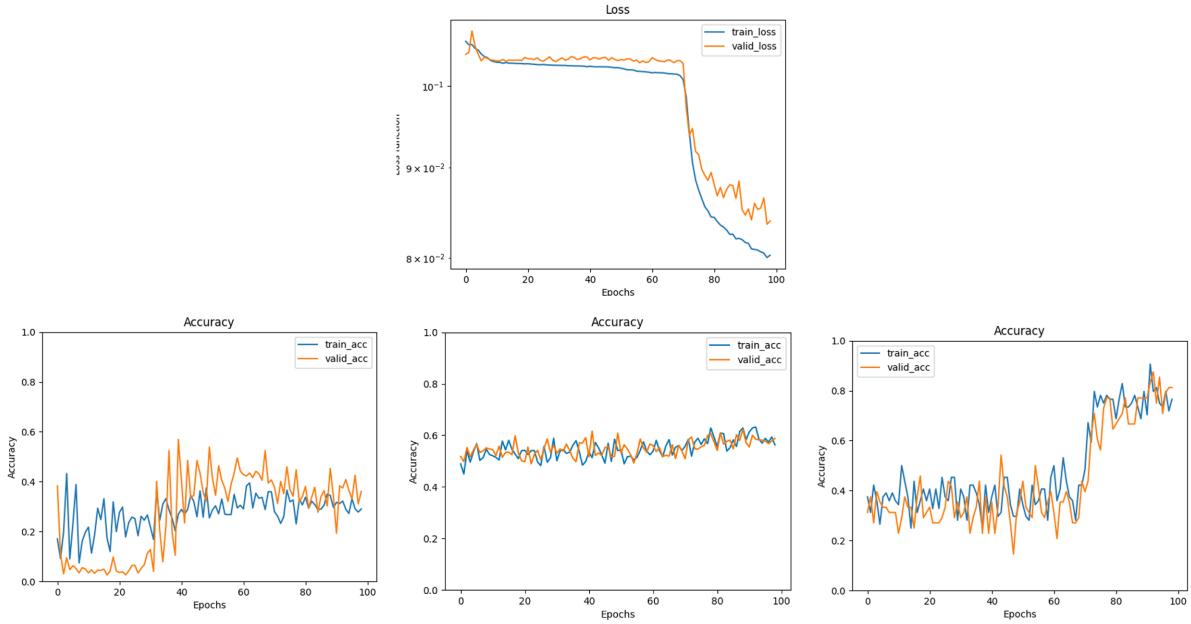


図 5.15: (上) 学習の経過における損失関数。(左下) 学習の経過におけるノードの学習精度、(中央下) リンクの学習精度、(右下) グラフの学習精度。

続いて、ノード、リンク、グラフ分類における混合行列を図 5.16 に示す。ノード、リンクの学習精度は十分良いと言える制度ではなく、一方でグラフの識別においてはディープニューラルネットワークでの実装に匹敵する精度が得られた。ノードの混合行列では PV と SVBB と識別する割合が高く、一方で TVCC、Others について分類が出来ず SVBB に分類されてしまう傾向にあった。

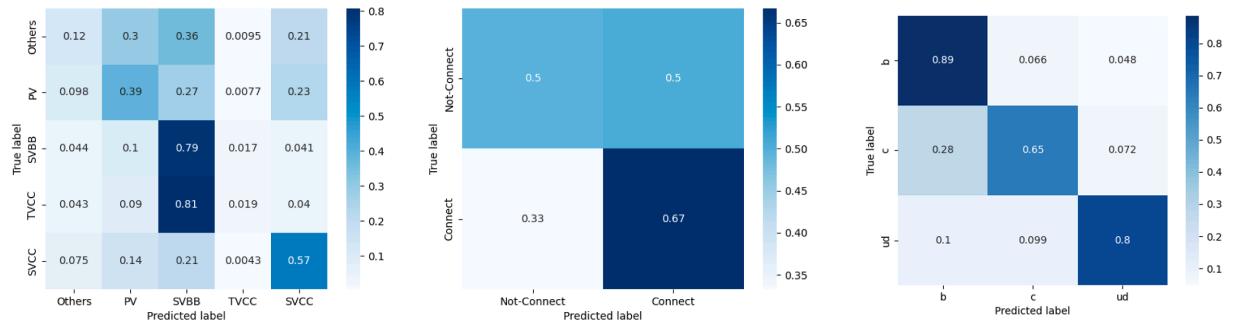


図 5.16: (左) ノード分類の混合行列、(中央) リンク予測の混合行列、(右) グラフ分類の混合行列。

最後に、LCFIPlus との比較を示す。ディープニューラルネットワークの比較と同様に、図 5.17, 5.18 は b/c ジェットの識別効率を示している。 b ジェットの識別効率は c ジェットについて向上しており、識別効率 80% における識別割合は、LCFIPlus が c ジェットの識別割合が 7.3%、 uds ジェットの識別効率が 0.74% であるのに対して、グラフニューラルネットワー

クの c ジェットの識別割合は 2.1%、 uds ジェットの識別効率も 1.49% という結果になり、 c ジェットの誤認率が 7 分の 2 程度になっているものの、 uds ジェットに関しては若干の精度低下となった。また、 c ジェットの識別効率 80% における識別割合は、LCFIPlus が b ジェットの識別割合が 22%、 uds ジェットの識別効率が 24% であるのに対して、グラフニューラルネットワークの b ジェットの識別割合は 40%、 uds ジェットの識別効率は 14% ほどという結果になり、 b ジェットの誤認率が 1.5 倍近く上がってしまったものの、 uds ジェットの誤認率は少々下げる事ができた。ノード分類結果では、PV と SVBB と識別する割合が高いことから、 b ジェットに対する識別率が高くなっていることが考えられる。また、 c ジェットの b ジェットとの分離が悪い点に関しては、主に SVBB, TVCC に対する SVCC の分類精度が課題となっていると推測される。

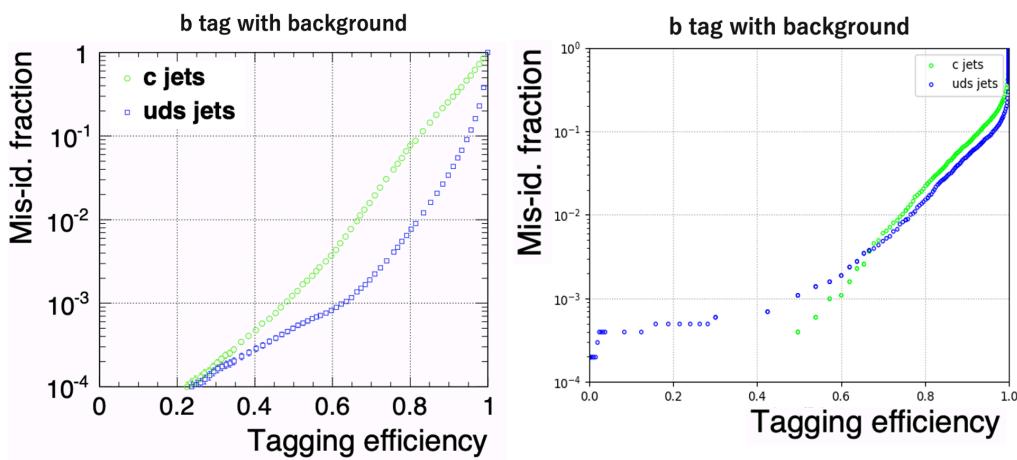


図 5.17: LCFIPlus(左) とグラフニューラルネットワーク(右)による b フレーバージェットの識別効率の比較。緑： b ジェットに対する c ジェットの識別効率、青： b ジェットに対する uds ジェットの識別効率。

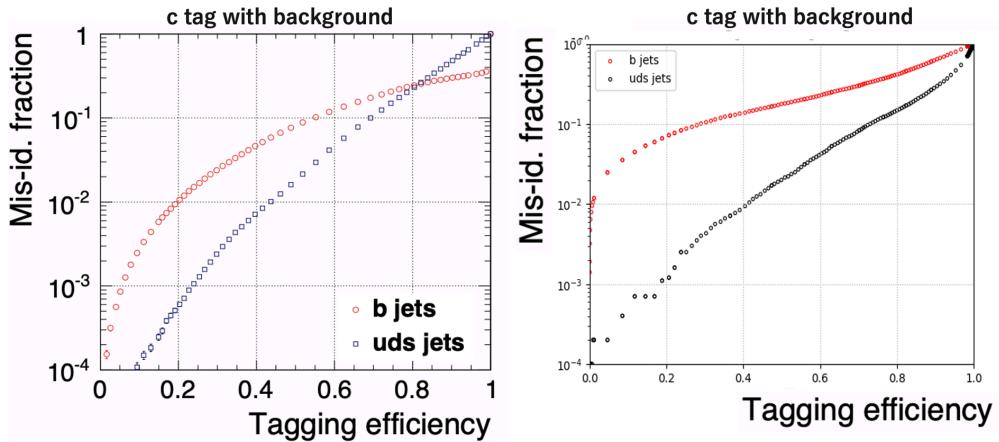


図 5.18: LCFIPlus(左)とグラフニューラルネットワーク(右)による c フレーバージェットの識別効率の比較。赤： c ジェットに対する b ジェットの識別効率、黒： c ジェットに対する uds ジェットの識別効率。

| 識別効率=0.8 | 背景ジェット | 誤認率 | |
|----------|------------|----------|--------|
| | | LCFIPlus | GNN |
| b ジェット | c ジェット | 0.073 | 0.021 |
| | uds ジェット | 0.0074 | 0.0149 |
| c ジェット | b ジェット | 0.22 | 0.40 |
| | uds ジェット | 0.24 | 0.14 |

表 5.12: 識別効率 (Tagging Efficiency) に対するジェット誤認率 (Mis-id fraction)

結果として、グラフニューラルネットワークによる実装では LCFIPlus と比較して顕著な性能向上があったとは言えない結果になった。しかしながらエッジリンク予測の実行によって、これまで別のプロセスで実行していた崩壊点検出を、1つのネットワーク内で内包することができた。これによって同程度の性能ではあるが、低レベルの入力に対して高レベルの出力をを行う、中間層の情報損失の少ない学習を行うことが可能になった。また、LCFIPlus における Vertex Finder では非常に時間的コストを必要としており、今回の実装では計算コストの面でも改善があると言える。

第 6 章

まとめと今後の展望

本論文では、電子陽電子ヒッグスファクトリーのためのジェット測定技術の研究として、特に ILC を念頭に置いた 2 つのテーマの研究を行った。1 つ目のテーマでは、ILC の検出器案である ILD の電磁カロリメータのうち、SiW-ECAL の技術プロトタイプを高エネルギービーム試験によって性能評価した。2 つ目のテーマでは、深層学習を用いたフレーバー識別アルゴリズムの開発を行った。本章ではそれぞれの研究のまとめと今後の展望について述べる。

ILD SiW-ECAL プロトタイプの高エネルギービーム試験による性能評価

本研究では、ILD SiW-ECAL プロトタイプの高エネルギービーム試験による性能評価を行った。ILD は ILC の検出器の候補であり、PFA に最適化された構造となっている。ILD を構成する ECAL には高精細なカロリメータが求められており、その候補として SiW-ECAL がある。SiW-ECAL は $5.5 \text{ mm} \times 5.5 \text{ mm}$ のピクセルシリコンセンサーを読み出しに用いた高精細なカロリメータであり、CALICE コラボレーションによって技術プロトタイプが開発されている。本研究では、CERN SPS 加速器における $10 \sim 200 \text{ GeV}$ 高エネルギービームを用いてプロトタイプの性能評価実験を行った。実験は ECAL と HCAL の同時運転と、これまで最大となる高エネルギービームを用いた SiW-ECAL の試験が目的であった。結果として、ECAL と HCAL の同時運転によるデータ収集を達成し、同時に SiW-ECAL の性能評価試験も問題なく完了した。一方で実験後の解析では、大きく 3 つの課題点が見つかった。1 つ目はセンサーが一部剥がれてしまい、全てのチャンネルでデータの取得ができなかった点である。これは運搬中の衝撃や経年劣化によるボードの変形によってセンサーが剥がれてしまったと予想される。また 2 つ目は 80 GeV 以上の高エネルギービームを SiW-ECAL に照射すると、一部のセンサーではセンサー全体に本来のヒットとは異なるトリガーが収集されてしまった点である。これはエネルギーが高くなった際にセンサーや信号処理回路に大きな電流が流れしたことによる、ノイズの増大や電位の変動などが原因として推察されるが、現在も引き続き調査を行なっている。また、2 つ目は複数の読み出しチャンネルにおいて、ペデスタルのピークが 2 つ（ダブルペデスタルと呼ぶ）確認された点である。これはヒットによる信号を求める上

で大きな問題となる。また、ダブルペデスタル現象は Retriggering とも関係していることが先行研究から分かっており、Retriggering は正しいヒットの情報が保存できなくなってしまう致命的な課題点であり、ダブルペデスタルと Retriggering について調査することは重要であると言える。そのためこの現象についてその発生箇所やピークの振る舞い等の調査を行なった。解析の結果、ダブルペデスタルは内部配線や外部接続をアップデートする以前の FEV10、11 に多く発生していた。そのため、ダブルペデスタルは電源に由来する現象であることが推測することができる。また、Retriggering の発生箇所を調査したところ、ダブルペデスタルが発生している箇所で多く発生しており、なおかつビームスポットに近いところで多いことが確認された。今後の方針として、センサー剥がれについては、センサーとボードの接着の見直しをおこない、また同様の現象が起こらぬよう耐久性の試験や劣化を抑える保存方法の検討が挙げられる。また、ダブルペデスタル現象に対しては、ダブルペデスタルを引き起こしている配線を特定することで次世代のアップデートに向けた施策を練ることや、その発生条件についてより詳細な解析を行い信号の選択によって取り除くことを目指すことが考えられる。また Retriggering についてはヒットによる基板上の增幅回路の電源の搅乱によって電位が揺れることが原因として考えられており、今回の結果もヒットの多いところで発生が集中していた。これに対しては回路内のどこかでグラウンドをとることで安定化を目指す必要がある。

深層学習を用いたフレーバー識別アルゴリズム

本研究では、深層学習を用いたフレーバー識別アルゴリズムの開発を行った。ヒッグスファクトリーの目的であるヒッグス粒子は、クォークやグルーオンなどに崩壊し、多数のハドロンの束であるジェットとして検出器に到達する。そのためジェットの再構成でクォークのフレーバーを識別することは、ヒッグスの物理を知るための解析において非常に重要な情報となる。フレーバー識別はジェットの構成粒子の種類や運動量、崩壊点に関する情報から、ジェットの元となるクォークのフレーバーを識別するプロセスであり、現在ジェットの再構成に用いられている LCFIPlus では従来の機械学習手法である BDTs が用いられている。これに対して深層学習を導入することで、識別性能の向上や崩壊点検出アルゴリズムとの統合などを目指した。

はじめに、最も単純な構造のネットワークモデルであるディープニューラルネットワークによる実装を行った。アルゴリズムには過学習対策や学習の効率化に向けた手法を用いて、全結合層を中心としたディープニューラルネットワークのモデルを、PyTorch を用いて構築した。学習の入力データとして、250 GeV の ILD フルシミュレーションのイベント 400 万イベントを、変数には BDTs の学習に用いた変数と同じものを用いた。学習を重ねるにつれて損失は減少し、学習全体の精度はおよそ 82.5% で得られた。また LCFIPlus との比較にあたっては、識別効率あたりの誤認率によって比較した。結果として、 c ジェットの識別など一部で改善が見られたものの、LCFIPlus と大差ない、あるいはやや劣る結果となった。

続いて、グラフデータを用いたフレーバー識別アルゴリズムの実装を行った。動機として

は、フレーバー識別において重要な IP 付近の物理現象をグラフ構造のデータとして構築することで、物理量を羅列する 1 次元の数値データと比較して表現力が上がり、性能が向上するのではないかという狙いがあった。またグラフデータを構築する際の副産物として、以前は別プロセスであった崩壊点検出を 1 回の識別で同時に処理できるという長所もあった。グラフデータは独自のデータセットクラスを作成し、飛跡をノードとする全結合のグラフを、1 ジェットあたり 1 グラフ構築した。また、このグラフでは全結合のディープニューラルネットワークと異なり、ノードのみ飛跡再構成におけるフィッティングパラメータを特徴量を持つとした。イベントは 250 GeV の ILD フルシミュレーションのイベント 240 万イベントを用いた。ネットワークアルゴリズムは GAT を 3 層使用し、損失関数の工夫やデータ次元変更などの手法を用いてノード識別/リンク予測/グラフ分類の 3 つのタスクを同時に実行し、重みを共有するようなネットワークモデルを PyTorch Geometric を用いて実装した。学習は不安定であったものの損失は減少し、ディープニューラルネットワークに匹敵するグラフ分類の精度が得られた。また識別効率あたりの誤認率のプロットによる LCFIPPlus との比較では、 b ジェット識別において改善が見られたが、 c ジェット識別では非常に悪い結果となった。またノード識別、リンク予測に対しては十分に良い精度を得ることができなかった。しかし、低い精度ながら崩壊点検出アルゴリズムの統合を達成することができた。

今後の展望として、フレーバー識別アルゴリズムには大きく 2 つの方向性で改善が可能であると考えている。

1 つ目は深層学習の理論・技術面での方向性である。GAT に挙げられるネットワークの処理に関してより理解を深め、フレーバー識別により最適なネットワーク構造を提案することや、物理的な性質をより組み込むなどフレーバー識別に最適化された損失関数を設計するなど、演算手法によって更なる改善の見込みがあると考えている。また、グラフデータの設計についても改善ができると考えており、具体例としては今回のデータセットにおいて崩壊点となるエッジに特徴量を持たせることや、グラフデータの構成要素を変更することなどが挙げられる。現在エッジは特徴量を持っておらず、接続したノードの情報のみで更新を行っているが、エッジ自身が特徴量を持つことでデータの学習パラメータが増え、精度が向上すると考えることができる。また、本研究では飛跡をノードとするグラフを構成したが、本来飛跡は点ではなく曲線の形状を取るため、やはり崩壊点のような物理的に点となる量をノードとして、実際の物理現象に近いグラフの構築が期待できると考えている。

2 つ目は、iLCSoft としての実装に向けた方向性である。例えば出力情報の設計や C++ 環境への移行があると考えている。出力設計に関して、今回のグラフデータを用いたフレーバー識別では崩壊点検出を包含するアルゴリズムを開発した。しかし、現段階では解析において崩壊点の情報が必要になった際、物理量として出力することは出来ない。そのため方針としては、中間層で得たネットワークの潜在的なパラメータを回帰問題として設計することで、新しく崩壊点など数値データの出力を得ることなどが考えられる。また別の例としては、開発したフレーバー識別アルゴリズムの C++ 環境への移植である。現在ジェット再構成に用いられている LCFIPPlus は Marlin プロセッサーの 1 部であり、C++ 環境で動作している一方で、本研

究で開発したアルゴリズムは python によって記述されている。そのため libTorch/PyTorch Geometric C++ API を用いる、あるいは C++ によって書き換えを行うことによる iLCSoft への実装は、今後の課題の一つである。

付録 A

Appendix A

A.1 飛跡のパラメータ

A.1.1 LDC 座標系

ILCにおいて検出器を通過する荷電粒子には、飛跡を記述する際に専用の手法が用いられており、それによってシミュレーションツールなどで変換を行うことなく、データを解析することができる。この手法は ILD の母体となった Large Detector Concept (LDC) よって定義された座標系 [36] を基に定義されており、LDC 座標系は次のように定義される。

1. 右手系直交座標。
2. 電子陽電子の相互作用点を原点とする。
3. ビーム方向に沿って z 軸をとる。
4. 垂直上方向に y 軸をとる。

この座標系における任意のベクトル \mathbf{v} は球面座標系において、次のように表される。

$$\mathbf{v} = \begin{pmatrix} |\mathbf{v}| \sin \theta \cos \phi \\ |\mathbf{v}| \sin \theta \sin \phi \\ |\mathbf{v}| \cos \theta \end{pmatrix} \quad (\text{A.1})$$

$$\theta \in [0, \pi]$$

$$\phi \in [-\pi, \pi]$$

A.1.2 飛跡パラメータ

ILC で発生する荷電粒子は一定の磁場の影響を受け、螺旋軌道を描く。磁場は z 軸に並行な方向に一様にかかるものとする。この時、荷電粒子の飛跡は xy 平面への射影における円となり、z 軸方向の変位は弧の長さ s の線形関数となる。

荷電粒子は運動の基準点 $\mathbf{P}^r = (P_x^r, P_y^r, P_z^r)$ と飛跡パラメータ $(\Omega, \phi_0, d_0, z_0, \tan \lambda)$ によっ

て定義される。また、以下の定義において \mathbf{P}^0 は xy 平面上の基準点への最近接点を表す。

xy 平面

xy 平面 (図 A.1) では荷電粒子の移動は、以下のように定義される。[37]

- ϕ_0 は運動量の P_0 における方位角を表す。
- Ω は飛跡の曲率を表す。

$$|\Omega| = \frac{1}{R} \quad (\text{A.2})$$

ここで、 R は飛跡の曲率半径を表す。

- d_0 は xy 平面のインパクトパラメータを表す。 $\mathbf{d} = (d_x, d_y)$ を任意の点 \mathbf{P}^r から \mathbf{P}^0 までのベクトルとする。

$$\mathbf{d} = \mathbf{P}^0 - \mathbf{P}^r \quad (\text{A.3})$$

また、 \mathbf{P}^0 から飛跡への法線ベクトルを \mathbf{n}_{pca} とすると、 d_0 は以下のように表される。

$$d_0 = \mathbf{n}_{pca} \cdot \mathbf{d} = -(P_x^r - P_x^0) \sin \phi_0 + (P_y^r - P_y^0) \cos \phi_0 \quad (\text{A.4})$$

$|\mathbf{n}_{pca}|$ であるため、 $|d_0|$ は xy 平面における \mathbf{P}^r と \mathbf{P}^0 の距離を表す。

xy 平面における円の中心点 \mathbf{P}^c は、通常 \mathbf{P}^r と異なる点を取ることが多く、軌道上の任意の点における運動量を $\mathbf{P} = (P_x, P_y, P_z)$ とすると、 \mathbf{P}^c は以下のように計算される。

$$P_x^c = P_x + \frac{\sin \phi}{\Omega} \quad (\text{A.5})$$

$$P_y^c = P_y + \frac{\cos \phi}{\Omega} \quad (\text{A.6})$$

$$(\text{A.7})$$

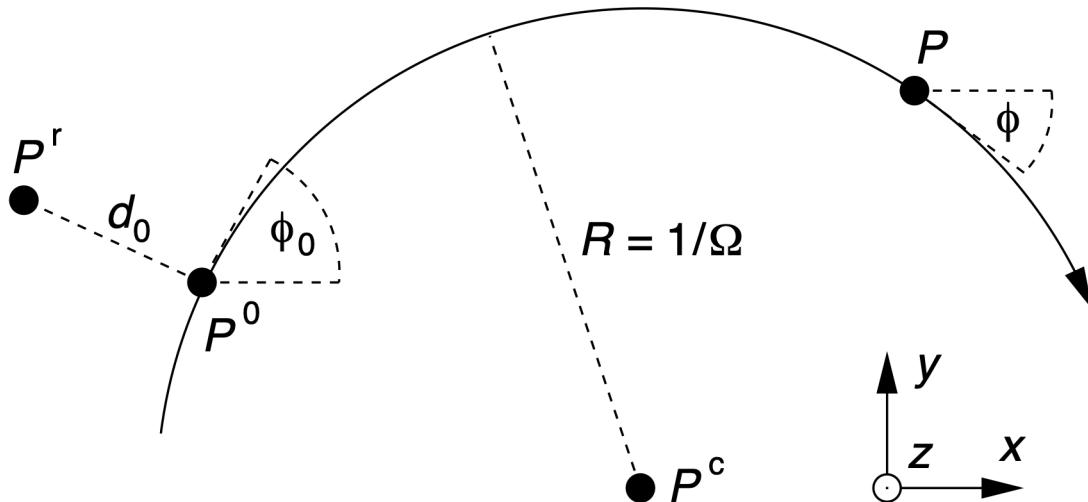


図 A.1: 飛跡の螺旋軌道を xy 平面へ射影した図。軌道は中心点 \mathbf{P}^c 、半径 R をもつ円弧で表される。また、全ての飛跡パラメータは基準点 \mathbf{P}^r を中心に与えられる。

sz 平面

sz 平面（図 A.2）において、飛跡は直線に沿って移動し、2つのパラメータ $(\tan \lambda, z_0)$ によって表される。

- $\tan \lambda$ は sz 平面における直線の傾き dz/ds を表し、運動量ベクトル $\mathbf{p} = (p_x, p_y, p_z)$ に対して以下のように計算される。

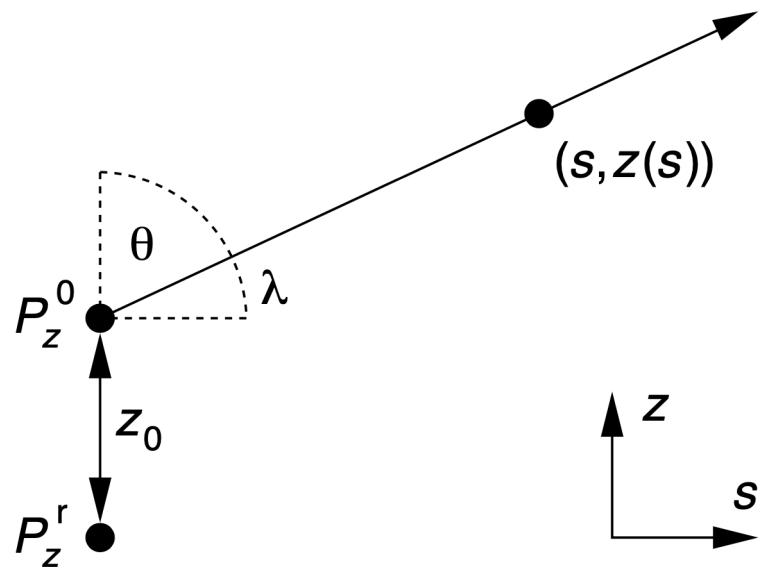
$$\tan \lambda = \frac{p_z}{\sqrt{p_x^2 + p_y^2}} = \cot \theta \quad (\text{A.8})$$

- z_0 は基準点 \mathbf{P}^r から \mathbf{P}^0 までの z 軸上の距離を表す。

$$z_0 = P_z^0 - P_z^r \quad (\text{A.9})$$

これらを踏まえ、飛跡の sz 平面射影の軌跡 z は、 xy 平面における飛跡の経路積分 s を用いて次のように表される。

$$z = (z_0 + P_z^r) + s \cdot \tan \lambda \quad (\text{A.10})$$

図 A.2: 飛跡の sz 平面への射影

謝辞

本研究を進めるにあたり、お世話になりましたすべての方々に感謝申し上げます。

本研究テーマは、指導教員である末原大幹助教に授けて頂きました。末原助教は、日々の研究において私の理解に寄り添い、多くの的確な助言を与えて下さいました。また、国内外での高エネルギー加速器実験への参加や学会・国際会議での発表など、新しいことに挑戦する機会もたくさん与えて頂きました。お陰様で貴重な経験を積み重ね、多くを学ぶことが出来ました。心より感謝致します。川越清以教授には、ゼミナールを通して素粒子物理学の基礎から ILC に至るまで、素粒子実験の基礎を教えていただきました。また、研究の進捗報告や学会の発表練習においても多くの助言を頂き、本論文の執筆においても細かにご指導していただきました。厚くお礼申し上げます。東城順治教授には学部 4 年次の素粒子物理学の講義において、素粒子物理の基礎を教えて頂きました。吉岡瑞樹准教授には、学部 3 年次のプログラミングの講義を通してプログラミングによる数値計算の基礎を教えて頂きました。森津学助教には 3 年生実験の TA 業務や、高校生の体験授業において大変お世話になり、親身になってご相談に乗って頂きました。音野瑛俊助教には、CERN にて最先端の実験施設や研究について教えて頂き、その他の面でもサポート頂きました。山中隆志助教には TA 業務で助言をいただいたほか、g-2 実験の ASIC 検査業務では普段の研究では出来ない経験をさせて頂きました。小林大元特任助教、細川律也学術研究員、小川真治特別研究員、水野貴裕学術研究員にはゼミナールや論文紹介の際に的確なご指摘やご指導を頂いたほか、研究室生活においても色々な話をする機会があり楽しく研究生活を過ごせました。テクニカルスタッフの重松さおり氏には、出張や研究室内の物品管理など研究に関わる事務作業において大変お世話になり、集中して研究に取り組むことが出来ました。また、理学部等事務部や物理学事務室の皆様には、TA や出張の際に事務手続きにおいてサポートいただきました。

ILC グループの先輩である後藤輝一氏には、深層学習の研究にあたって一部修士論文の内容を引き継いだこともあり、様々な面で参考にさせて頂きました。また同じく ILC グループの先輩である久原真美氏には、研究における知識を丁寧に教えて頂いただけでなく、発表会や就職活動に至るまで様々なお話を聞かせて頂き、憧れの先輩でした。ILC グループの同輩にあたる津村周作君とは、互いに研究の進捗を切磋琢磨し、研究で行き詰った時には納得がいくまで議論を交わすことのできる、貴重な友人でありました。ILC グループの後輩である永江航志君には、先輩として有益なアドバイスができる機会が少なかったことが悔やまれますが、研究に向き合う姿勢からは大変刺激を受けました。

研究室の先輩である高田秀佐氏、古賀淳氏、山口尚輝氏、宮崎祐太氏、野口恭平氏、竹内佑甫氏、松崎俊氏、岩下侑太郎氏、姚舜禹氏には、論文紹介や月1ミーティングの研究進捗の報告など様々な場面でアドバイスを頂き、本論文を執筆する上でも多くの助言を頂きました。同期に当たる谷田征輝君、宮本佳門君、樋口義清君、川上真言君とは、ゼミを通して議論を交わしたほか、研究以外にも私のつまらない話に付き合って頂き、楽しく大学院生活を送ることが出来ました。後輩にあたる西原君、星野君、塩谷君、梅林君、山田君、Afiq Azraei君、周君、また特研生の土谷君、花田君、吉川君、今村君、水取君には、研究に取り組む姿に刺激を受けました。

また研究にあたって外部の研究機関の方々にも大変お世話になりました。大阪公立大学の岩崎昌子氏、大阪大学の中島悠太氏、長原一氏、九州工業大学の武村紀子氏には、分野外であった深層学習の研究に際して多くの助言を頂きました。高エネルギー加速器研究機構の Daniel Jeans 氏、Junping Tian 氏、東京学芸大学の日高啓晶氏には、物理解析に際して、また信州大学の竹下徹氏、東京大学の大谷航氏にはカロリメータの研究において多くの助言を頂きました。IJCLab の Roman Pöschl 氏、LLR の Vincent Boundry 氏、IFIC の Adrian Irles 氏、東北大学の奥川悠元氏には、CERN でのビームテスト実験において議論を交わし、カロリメータや解析について助言を頂きました。

その他多くの方にも本当にお世話になりました。皆様の助けあったからこそ研究を進めることができきました。

最後に、今まで 24 年間応援し支えてくれた両親に感謝の意を示し、謝辞の言葉とさせて頂きます。

参考文献

- [1] A. Miyamoto, ilcsoft, ILC 物理解析・ソフトウェア講習会 .
- [2] CERN, Cern's accelerator complex, <https://www.home.cern/science/accelerators/accelerator-complex> .
- [3] J. S. Marshall and M. A. Thomson, Pandora particle flow algorithm, arXiv:1308.4537v1 (2013).
- [4] S. L. Glashow, Partial-symmetries of weak interactions, Nuclear Physics (1960).
- [5] 藤井恵介, 高エネルギー加速器セミナー OHO'14: ILC の物理.
- [6] S. D. Bass, A. D. Roeck, and M. Kado, The higgs boson implications and prospects for future discoveries, Nature Review Physics (2021).
- [7] H. Baer *et al.*, The international linear collider: Technical design report, (2013).
- [8] K. Fujii *et al.*, Physics case for the international linear collider, arXiv:1506.05992 (2015).
- [9] M. A. Thomson, Particle flow calorimetry and the pandora pfa algorithm., Nucl. Instrum. Meth. **A611** (2009).
- [10] Ilc software packages.
- [11] F. Gaede, Software tools for the ilc, nuclear instruments and methods in physics research section a, Proceedings of the X International Workshop on Advanced Computing and Analysis Techniques in Physics Research **559** (2006).
- [12] M. Petric *et al.*, Detector simulations with dd4hep, Journal of Physics Conference Series (2017).
- [13] W. Kilian, T. Ohl, and J. Reuter, Whizard: Simulating multi-particle processes at lhc and ilc, Eur. Phys. J. C 71 1742. arXiv:0708.4233 (2011).
- [14] T. Sjostrand, S. Mrenna, and P. Skands, Pythia 6.4 physics and manual, Journal of High Energy Physics (2006).
- [15] S. A. ae *et al.*, Geant4 - a simulation toolkit, Nuclear Instruments and Methods in Physics Research Section A **506**, 250 (2003).
- [16] T. Suehara and T. Tanabe, Lcfiplus: A framework for jet analysis in linear collider studies, Nucl.Instrum.Meth. A808 , 12 (2016).
- [17] S. Catani, Y. L. Dokshitzer, M. Olsson, G. Turnock, and B. Webber, New clustering

- algorithm for multijet cross sections in e+e – annihilation, Physics Letters B **269**, 432 (1991).
- [18] R. Brun and F. Rademakers, ROOT - An Object Oriented Data Analysis Framework, 1996.
- [19] J. Therhaag, TMVA Toolkit for multivariate data analysis in ROOT, PoS ICHEP2010 (2010).
- [20] ブライアン・R・マーティン and グレアム・ショー, マーティン/ショー 素粒子物理学 原著第4版 (講談社, 2020).
- [21] T. Suehara, Siw-ecal status, ILC-JP seasonal meeting (12 July 2021).
- [22] A. Irles, Siwecal-tb-analysis, <https://github.com/SiWECAL-TestBeam/SiWECAL-TB-analysis> (2019).
- [23] R. Frank, The perceptron: A probabilistic model for information storage and organization in the brain, Psychological Review (1958).
- [24] 斎藤康毅, ゼロから作る DeepLearning 1 (オライリージャパン, 2016).
- [25] D. P. Kingma and J. L. Ba, Adam: A method for stochastic optimization, conference paper at ICLR 2015 (arXiv:1412.6980v9) (2015).
- [26] L. Liu *et al.*, On the variance of the adaptive learning rate and beyond, ICLR 2020 (arXiv:1908.03265) (2020).
- [27] X. Glorot and Y. Bengio, *Understanding the difficulty of training deep feedforward neural networks* (International Conference on Artificial Intelligence and Statistics, 2010).
- [28] J. Zhou *et al.*, Graph neural networks: A review of methods and application, Published at AI Open 2021 (arXiv:1812.08434) (2020).
- [29] 村田剛志, グラフニューラルネットワーク Pytorch による実装 (オーム社, 2022).
- [30] W. L. Hamilton, R. Ying, and J. Leskovec, Inductive representation learning on large graphs, NIPS (arXiv:1706.02216) (2017).
- [31] P. Veličković *et al.*, Graph attention networks, ICLR 2018 (arXiv:1710.10903v3) , 12 (2017).
- [32] P. I. Frazier, A tutorial on bayesian optimization, arXiv:1807.02811 (2018).
- [33] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, Optuna: A next-generation hyperparameter optimization framework, KDD 2019: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery Data Mining (2019).
- [34] Pytorch.
- [35] Pytorch geometric.
- [36] A. Vogel, The coordinate system for ldc detector studies, LC-DET-2005-009 (2005).
- [37] T. Kramer, Track parameters in lcio, LC-DET-2006-004 (2006).