2024年度 卒業論文

パラメータを動的に調整する Fast Adversarial Trainingに関する研究

指導教員　史 又華　　教授

早稲田大学 基幹理工学部
電子物理システム学科
1 W212275-2
内藤 舜介
Shunsuke Naito

2025年2月

目 次

第1章

序論

1.1　本論文の背景と目的

近年，Artificial Intelligence（AI）の分野では多くの技術革新が起きており，その成果として AI 技術を応用した製品やサービスが社会のさまざまな場面で利用されるようになっている．例えば，音声認識を用いた翻訳や文字起こし，画像認識を用いた顔認証や自動運転，さらには医療診断などの多岐にわたる応用が現実化し，私たちの日常生活に深く浸透している．このような AI の基盤技術の一つとして注目されているのが機械学習である．機械学習とは，コンピュータに膨大なデータを読み込ませることで，データの中に潜むパターンや法則を見つけ出し，それをもとに分析や判断を行えるように学習させる技術である．この学習プロセスにより，膨大なデータからルールや関係性を自律的に学習することで，従来は人間が行っていたデータ分析や意思決定の一部をコンピュータが自律的に行えるようになる．さらに，機械学習を発展させた手法として注目されているのが Deep Learning である．Deep Learning は，人間や動物の神経細胞（ニューロン）の働きを模倣した Neural Network を基盤として学習させる技術である．この Neural Network は，モデルに学習データを取り込む入力層，データを処理して特徴を抽出する中間層，処理結果を外部に出力する出力層の三層構造をしており，特に中間層を多層化したものが Deep Neural Network（DNN）と呼ばれる．DNN では，多層化した中間層のニューロン同士を結びつけることで，信号の伝達や処理を複雑に行い，より複雑な特徴を抽出することができる．この高度な構造が，従来の機械学習手法では困難だった画像認識や自然言語処理，音声認識といった非構造的データの処理を可能にしている．

画像認識の分野で，Deep Learning が現在のように注目を集めるきっかけとなったのは，2012 年に開催された ILSVRC（ImageNet Large Scale Visual Recognition Challenge）という画像認識コンテストである．このコンテストは，2010 年から開催されている画像認識技術の競技会であり，ImageNet[1] という大規模な画像データセットに対する分類精度の高さを競うものである．2012 年の ILSVRC では，トロント大学の Hinton 率いる Super Vision チームが，AlexNet[2] と呼ばれる Convolutional Neural Network（CNN）を用いた Deep Learning モデルを提出し，Support Vector Machine（SVM）などの従来の機械学習アルゴリズムを大幅

に上回る性能で優勝した．CNN とは，入力データから局所的な特徴を抽出する畳み込み層，特徴を圧縮し位置の変動にロバストにするプーリング層，抽出された特徴から最終的な出力を生成する全結合層を，Neural Network に取り入れた Deep Learning モデルの一種である．この CNN は画像認識や物体検出において非常に有効であり，人間を超える精度を実現可能になった．これ以降，CNN ベースの VGG[3] や ResNet[4]，自然言語処理で用いられる Transformer ベースの Vision Transformer（ViT）[5] などの登場により，画像認識の性能は大きく向上し，その応用範囲を大きく拡大させた．

DNN は多くの問題やタスクにおいて最先端の性能を達成している一方，しばしば人間には知覚できないような敵対的摂動を加えた Adversarial Examples（AEs）に対して脆弱である [6]．例えば画像分類の分野では，オリジナルの画像に敵対的摂動を加えた AEs を誤ったクラスに誤分類してしまう．この脆弱性は，DNN を用いたセキュリティが重要な応用分野において大きなリスクとなる．この問題に対処するために，多くの防御手法が開発されてきた．中でも多段階の最適化を行う Projected Gradient Descent （PGD）[7] で生成された AEs を使用して訓練を行う標準的な Adversarial Training（AT）[7] は，AEs に対する DNN のロバスト性を高めるための最も効果的な方法の 1 つである．しかしながらこの手法は多段階であるがゆえに計算コストが非常に高く，標準的な AT の実用性を制限する要因となっている．一方，単一段階の Fast Gradient Sign Method（FGSM）[8] で生成された AEs を使用して訓練を行う Fast Adversarial Training（FAT）[8, 9] は計算コストが小さい手法である．この手法により計算コストが大きく削減される一方，訓練の途中にモデルのロバスト性が突然失われる，壊滅的過学習（catastrophic overfitting）[9] という現象が発生してしまう．この問題に対処するために，さまざまな FAT のバリエーションが提案されてきた．これらの手法は，摂動初期化や正則化といった手法を用いることで，壊滅的過学習を軽減し，最先端のモデルロバスト性を達成している．

本論文では，既存の動的な摂動初期化と正則化を組み合わせて，AEs の品質，及びモデルの分類能力に応じて動的に学習強度を調整する FAT 手法を提案する．さらに，CIFAR10[10]，CIFAR100[10] の 2 つのデータセットに対して提案手法の防御性能の評価と，比較及び検討を行うことを目標とする．

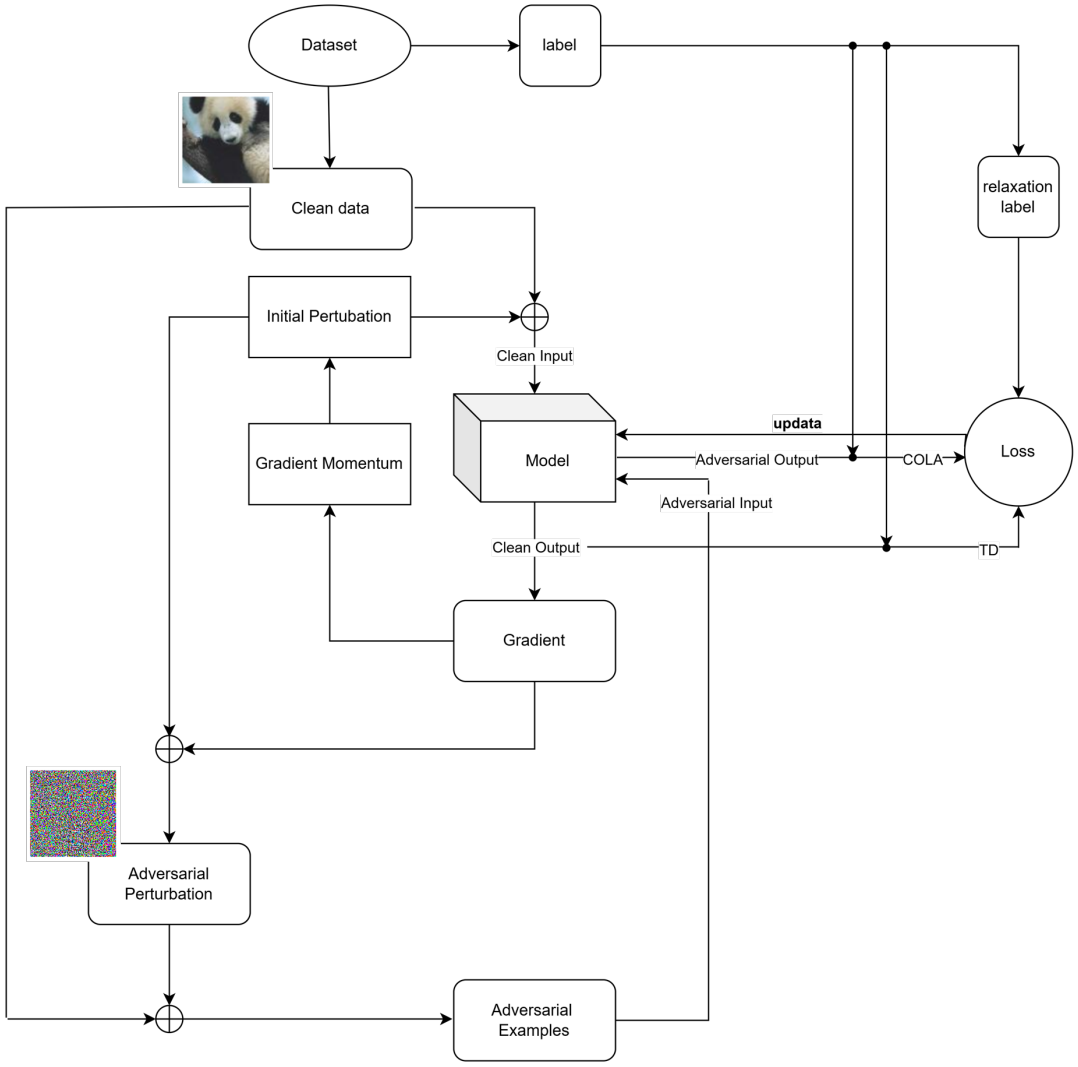

図 1.1: 提案手法 WMEP-ETA の概略図.

1.2　本論文の構成

本論文では，Adversarial Examples 及び Adversarial Training の先行研究について述べた後に，既存の動的なパラメータの調整を用いた摂動初期化と正則化を組み合わせた新たな Fast Adversarial Training を提案し，CIFAR10，CIFAR100 の 2 つのデータセットに対して行った実験結果及び考察を述べる．本論文は 4 章で構成される．以下に各章の概要を述べる．

第 2 章**「Adversarial Examples と Fast Adversarial Training」**では，いくつかの AEs 生成手法について述べた後，既存の FAT 手法についていくつか述べる．

第 3 章**「パラメータの動的な調整を用いた Fast Adversarial Training の提案」**では，本論文の提案手法について述べ，実験条件やデータセット，及び結果と考察について述べる．

第 4 章**「結論」**では，本論文を総括し，今後の展望について述べる．

第2章

Adversarial Examplesと Fast Adversarial Training

2.1　本章の概要

本章では，いくつかの Adversarial Examples の生成手法と既存の Fast Adversarial Training の手法について述べる．

2.2 節**「Adversarial Examples」**では，AT で用いる AEs の生成手法についていくつか述べる．

2.3 節**「Fast Adversarial Training」**では，既存の FAT の壊滅的過学習を抑制しロバスト性を向上させる手法の内，摂動初期化，重み平均化，ラベル緩和，分類駆動型損失関数，収束制約の 5 つについて述べる．

2.4 節**「既存研究の考察」**では，特に FAT に関する既存研究から，どのような手法を用いることで FAT のロバスト性が向上するかを考察する．

2.5 節**「本章のまとめ」**では，本章のまとめについて述べる．

2.2 Adversarial Examples

DNN は，しばしば人間には知覚できない微小な敵対的摂動を加えることで生成される Adversarial Examples に対して脆弱であることが知られており，これに対処するために AEs の性能を向上させるための一連の研究が提案されてきた．具体的には，Goodfellow らは，単一段階で損失関数の勾配の方向に敵対的摂動を生成する FGSM[8] という手法を提案した．Carlini と Wagner は，摂動の大きさと誤分類確率のトレードオフを見つける目的関数を最小化して敵対的摂動を生成する CW Attack（CW）[11] という強力な手法を提案した．Madry らは，多段階の最適化手法を用いた攻撃手法である PGD[7] という手法を提案した．さらに，Croce らは，PGD 攻撃を改良したパラメータ不要の 2 つの手法，APGD-CE と APGD-DLR[12] を提案した．この 2 つの手法は，他の 2 つの既存の攻撃手法（FAB Attack[13] と Square Attack[14]）と組み合わせて，AutoAttack[12] と呼ばれる，より強力で幅広い敵対的ロバスト性評価を可能にする攻撃手法を形成した．本節では，これらの AEs を生成する攻撃手法について述べる．

2.2.1 FGSM

FGSM[8] とは Goodfellow らによって提案された，単一段階で損失関数 $\mathcal{L}$ の勾配の方向に敵対的摂動 δ を生成する攻撃手法である．x を入力データ，y をそのラベル，w をモデルの重みパラメータ，$f_w(x)$ をモデルの出力，ϵ を最大許容摂動強度として，式 (2.1) のように表すことができる．

$$\delta = \epsilon \cdot \mathrm{sign}(\nabla_x \mathcal{L}(f_w(x), y)) \tag{2.1}$$

ただし，sign(·) は符号関数を表し，勾配の各成分について正なら+1，負なら-1 を返し，勾配の大きさには依存せずに摂動の方向を決定する．∇_x は入力 x に対する勾配を表し，損失関数 $\mathcal{L}$ を入力 x に対して微分した結果を返し，符号関数と合わさることで損失が増加する方向を示す．このように損失が増加する方向に摂動を加えることで誤分類を誘発する．図 2.1 はパンダのオリジナル画像に FGSM で生成した人間には知覚できないほどの摂動を加えて誤分類させた例である．ここでの J は損失関数，θ は重みパラメータを表している．オリジナル画像では 57.7%

でパンダであると分類されていたが，敵対的摂動が加えられた AEs では 99.3% でテナガザルであると誤分類してしまうようになっている．

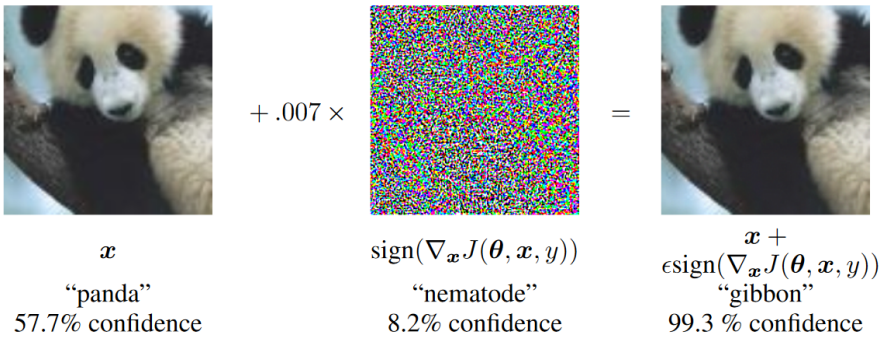

図 2.1: FGSM による Adversarial Examples の例 [8].

2.2.2　PGD

PGD[7] とは Madry らによって提案された，多段階の最適化手法を用いて敵対的摂動を生成する攻撃手法である．t ステップ目の摂動を δ^t，各ステップでのステップサイズを α とした時，δ^{t+1} は式 (2.2) のように表すことができる．

$$\delta^{t+1} = \Pi_{[-\epsilon,\epsilon]} \left(\delta^t + \alpha \cdot \mathrm{sign} \left(\nabla_x \mathcal{L} \left(f_w \left(x + \delta^t \right), y \right) \right) \right) \tag{2.2}$$

ただし，$\Pi_{[-\epsilon,\epsilon]}$ は射影操作を表し，摂動 δ^{t+1} を $[-\epsilon, \epsilon]$ の範囲に制限する．このように多段階の最適化を行うことで，摂動空間全体を探索し，FGSM よりもさらに最適解に近い摂動を生成しやすく強力な攻撃である一方，ステップ数に比例して計算コストが大きくなるというデメリットもある．

2.2.3　CW Attack

CW Attack[11] とは，Carlini と Wagner によって提案された，摂動の大きさと誤分類確率のトレードオフを見つける目的関数を最小化する敵対的摂動を生成する攻撃手法であり，摂動の大きさと誤分類確率のトレードオフを制御する重みパラメータ c を用いて，式 (2.3) のように表すことができる．

$$\min_{\delta} \|\delta\|_q + c \cdot f(x + \delta) \tag{2.3}$$

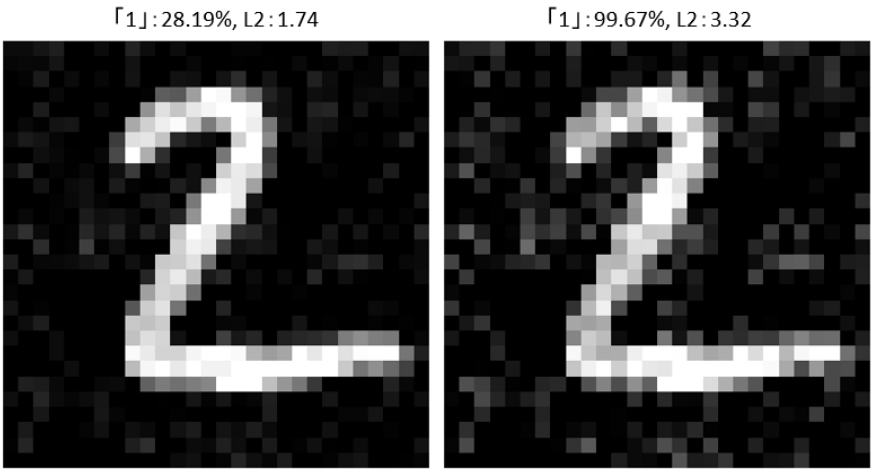

図 2.2: CW による Adversarial Examples の例 [11].

ここで，$\|\delta\|_q$ は L_q ノルム [11] での摂動の大きさを表す．また，$f(x+\delta)$ は非標的型攻撃の場合式 (2.4)，非標的型攻撃の場合式 (2.5) のように表され，モデルが誤分類を起こすように調整された損失関数である．重みパラメータ c は，大きい程損失関数の最小化，つまり誤分類を優先し，小さい程摂動の大きさの最小化，つまり摂動の小ささを優先する．

$$f(x+\delta) = \max\left(z(x+\delta)_y - \max_{i\neq y} z(x+\delta)_i + \kappa,\, 0\right) \tag{2.4}$$

$$f(x+\delta) = \max\left(\max_{i\neq t} z(x+\delta)_i - z(x+\delta)_t + \kappa,\, 0\right) \tag{2.5}$$

$z(x+\delta)_i$ は i 番目のクラスに対する出力ロジットであり，ここでの y は正解クラス，t は標的クラスを表す．また，κ はモデルが誤分類する際の信頼度を調整するパラメータであり，$\kappa > 0$ の場合は高い確率で誤分類されるような設定となり，$\kappa = 0$ の場合は最低限の確率でただ誤分類されれば十分という設定になる．具体的には損失関数 $f(x+\delta) = 0$ を目標とするが，例として標的型攻撃の場合を考え，$\max_{i\neq t} z(x+\delta)_i - z(x+\delta)_t < 0$ の時，標的クラスのロジットが標的クラス以外のクラスの最大ロジットより大きく，AEs が標的クラスに誤分類される．しかしながら信頼度パラメータが大きい場合は，誤分類はされても，損失関数の値を最小化するためには，標的クラスのロジットが標的クラス以外のクラスの最大ロジットよりかなり大きい必要があり，結果として高い確率で誤分類されるようになる．図

2.2 は MNIST データセットの「2」の画像を二つの信頼度パラメータ $\kappa = 0$，$\kappa = 7$ で「1」に誤分類させた例である．信頼度パラメータが小さい $\kappa = 0$ の時,「1」への分類確率が小さいが，L_2 ノルムの値も小さく摂動が目立たないことがわかる．一方で信頼度パラメータが大きい $\kappa = 7$ の時,「1」への分類確率が大きいが，L_2 ノルムの値も大きく摂動がかなり目立つことがわかる．

ここでノルム [11] について簡単に説明する．ノルムとは，ベクトルや関数の大きさや距離を測るための尺度であり，AEs と入力データの類似性を定量化するために用いられる．一般に L_q ノルム $\|v\|_q$ は式 (2.6) のように表すことができる．

$$\|v\|_q = \left(\sum_{i=1}^{n} |v_i|^q \right)^{\frac{1}{q}} \tag{2.6}$$

よく用いられるノルムとしては，L_0 ノルム，L_2 ノルム，L_∞ ノルムがある．まず L_0 ノルム $\|\delta\|_0$ は，入力データと AEs の間で変更された要素の数，特に画像の場合は変更されたピクセルの数を表す．次に，L_2 ノルム $\|\delta\|_2$ は，一般的な距離として用いられるユークリッド距離を表し，画像中の多くのピクセルに小さな変更を加えても L_2 ノルムは小さくなる可能性がある．そして，L_∞ ノルム $\|\delta\|_\infty$ は，最大変更量を表し，画像中の変更されるピクセルの数に制限はない．CW Attack をはじめ多くの攻撃手法では，これらのノルムを用いて摂動の大きさを定量化する．

2.2.4　AutoAttack

AutoAttack[12] とは，Croce らによって提案された，PGD 攻撃を改良したパラメータ不要の Auto-PGD（APGD）の 2 つのバリエーション（APGD-CE[12] と APGD-DLR[12]）と，既存の 2 つの攻撃手法（FAB Attack[13] と Square Attack[14]）をアンサンブルした，モデルのロバスト性を評価するための攻撃フレームワークである．

APGD[12] について説明する．APGD では通常の PGD とはモメンタムが追加されている点とステップサイズ η^k が動的である点が異なり，式 (2.7) のように表すことができる．

$$\begin{aligned} z^{k+1} &= \Pi_{[-\epsilon,\epsilon]} \left(x^k + \eta^k \nabla \mathcal{L} \left(x^k \right) \right), \\ x^{k+1} &= \Pi_{[-\epsilon,\epsilon]} \left(x^k + \alpha \cdot \left(z^{k+1} - x^k \right) + (1-\alpha) \cdot \left(x^k - x^{k-1} \right) \right) \end{aligned} \tag{2.7}$$

ここでの z^{k+1} は $k+1$ ステップでの一時的な AEs となっている．$k+1$ ステップでの最終的な AEs である x^{k+1} は，モメンタムを導入して現在の更新 $(z^{k+1}-x^k)$ と前回の更新 (x^k-x^{k-1}) を組み合わせることで，更新方向の一貫性を確保して安定した探索を実現させる．ステップサイズ η^k（通常 $\eta^0=2\epsilon$）は各チェックポイント w_j において，次の二つの条件のどちらかを満たす場合半減する．

1. $\sum_{i=w_{j-1}}^{w_j-1}\mathbf{1}_{\mathcal{L}(x^{i+1})>\mathcal{L}(x^i)} < \rho\cdot(w_j-w_{j-1})$
2. $\eta^{w_{j-1}}\equiv\eta^{w_j}\quad\text{and}\quad\mathcal{L}_{\max}^{w_{j-1}}\equiv\mathcal{L}_{\max}^{w_j}$

1 は条件を満たす場合 1，満たさない場合 0 を返す関数である．条件 1 は，前回のチェックポイント w_{j-1} から今回のチェックポイント w_j-1 までで損失関数が大きくなった，つまり更新が上手くいったステップが全体の ρ（通常 $\rho=0.75$）未満の場合を表す．条件 2 は，前回のチェックポイント w_{j-1} からステップサイズが更新されず，損失関数の最大値も更新されない場合を表し，アルゴリズムがサイクルに陥る可能性を排除する．これらの条件を満たした時，ステップサイズ η^k を半減するだけでなく，$x^{w_j+1}=x_{\max}$ としてベストな AEs でリスタートする．ここで，チェックポイント w_j の決定は，反復回数を N_{iter} として，式 (2.8) で決定される．

$$\begin{aligned} w_j &= p_j N_{\text{iter}}, \\ p_{j+1} &= p_j + \max\{p_j - p_{j-1} - 0.03,\, 0.06\} \end{aligned} \tag{2.8}$$

ただし，$p_j\in[0,1]$，$p_0=0$，$p_1=0.22$ である．このようにチェックポイントが決定されることにより，初めは大域的な探索が行われて，徐々にチェックポイントの間隔が狭くなることで局所的な最適化を行うように変化する．Auto-PGD の二つのバリエーション Auto-PGD with Cross-Entropy（APGD-CE）と Auto-PGD with Difference of Logits Ratio（APGD-DLR）の違いは，使用する損失関数である．APGD-CE では式 (2.9) で表される Cross-Entropy Loss を使用する．

$$\mathcal{L}_{\text{CE}}(x,y) = -\log(p_y) = -\log\left(\frac{e^{z_y}}{\sum_{i=1}^{C}e^{z_i}}\right) \tag{2.9}$$

p_y は Softmax 関数であり，正解クラスのロジットの指数関数値を全てのクラスのロジットの指数関数値で割ることで確率値を返す関数である．ここで C は総クラス数を表している．Cross-Entropy Loss は正解クラスの Softmax 確率 p_y の対数を

取ったものを-1 倍したものとなる．Cross-Entropy Loss は損失関数として一般的に用いられているが，正解クラスのロジットが極めて大きい時，勾配消失 [12] が起こり，学習が停滞してしまうことがあることも知られている．これは式 (2.10) において，$p_y \simeq 1$ の時，$p_j \simeq 0 \quad (j \neq y)$ であり，$\frac{\partial \mathcal{L}_{\mathrm{CE}}}{\partial z_y} \simeq 0$，$\frac{\partial \mathcal{L}_{\mathrm{CE}}}{\partial z_j} \simeq 0$ となることによる．

$$
\begin{aligned}
\mathcal{L}_{\mathrm{CE}}(x, y) &= -z_y + \log \sum_{i=1}^{C} e^{z_i} \\
\frac{\partial \mathcal{L}_{\mathrm{CE}}}{\partial z_y} &= -1 + \frac{e^{z_y}}{\sum_{i=1}^{C} e^{z_i}} = -1 + p_y \\
\frac{\partial \mathcal{L}_{\mathrm{CE}}}{\partial z_j} &= \frac{e^{z_j}}{\sum_{i=1}^{C} e^{z_i}} = p_j \quad (j \neq y)
\end{aligned}
\tag{2.10}
$$

APGD-DLR では式 (2.11) で表される非標的型 Difference of Logits Ratio Loss 及び，式 (2.12) で表される標的型 Difference of Logits Ratio Loss を使用する．

$$
\mathcal{L}_{\mathrm{DLR}}(x, y) = -\frac{z_y - \max\limits_{j \neq y} z_j}{z_{\pi_1} - z_{\pi_3}} \tag{2.11}
$$

$$
\mathcal{L}_{\text{DLR-t}}(x, y) = -\frac{z_y - z_t}{z_{\pi_1} - (z_{\pi_3} + z_{\pi_4})/2} \tag{2.12}
$$

z_{π_i} は i 番目に大きなロジットを表す．もし入力データが正しく分類されている場合，$\pi_1 = y$ であり，$j = \pi_2$ となる．DLR Loss が大きくなるように摂動を加えることで，分子の値が負（$z_y - \max\limits_{j \neq y} z_j < 0$，または $z_y - z_t < 0$）で，非標的型の場合はモデルが入力データを正解クラス以外に誤分類するように，標的型の場合は標的クラスに誤分類するようになることを目指す．ここで，分母の値（$z_{\pi_1} - z_{\pi_3}$，または $z_{\pi_1} - (z_{\pi_3} + z_{\pi_4})/2$）は正規化の役割を果たしている．DLR Loss ではすべてのロジットに定数を加算しても差分を計算することにより影響はなく，シフト不変性を持つ．また全てのロジットに定数を乗算しても分母のスケーリングに影響はなく，スケール不変性を持つ．さらに誤分類に近い場合，分子の差分の変動は大きくなりやすい．以上の理由により，Cross-Entropy Loss では正解クラスのロジットが極めて大きい場合に勾配消失に陥りやすかったが，DLR Loss ではシフト不変性及びスケール不変性によりロジットの絶対値の影響は受けず，勾配消失に陥りにくい設計となっている．

Fast Adaptive Boundary Attack（FAB Attack）は入力を分類境界まで最短距離で移動させることで，最小の摂動でモデルを誤分類させる AEs を生成する手法で

ある．また，Square Attack はランダム検索を利用して画像中の局所的な正方形領域を変更することで AEs を生成する，効率的なクエリベースのブラックボックス攻撃手法である．ブラックボックス攻撃とは，モデルの内部構造やパラメータにアクセスできないブラックボックス条件下で，モデルの入力と出力のみを利用する攻撃である．今まで紹介してきた Square Attack 以外の攻撃（FGSM，PGD，CW Attack，APGD，FAB Attack）はホワイトボックス攻撃に分類される．ホワイトボックス攻撃とは，モデルの入出力に加えて，内部構造やパラメータ，特に勾配情報にアクセスできるホワイトボックス条件下での攻撃である．ブラックボックス攻撃はモデルの内部情報が不要なため，より現実的なシナリオでの適用が可能となる一方，モデルの内部情報を利用するホワイトボックス攻撃と比較して，一般的に攻撃の精度が悪くなり計算コストも大きくなる．AutoAttack は APGD-CE，APGD-DLR，FAB Attack，Square Attack の 4 つの攻撃をアンサンブルした攻撃フレームワークであり，アンサンブルすることにより攻撃の多様性を確保し，モデルの網羅的なロバスト性を評価する際に役立つ．具体的には，PGD ベースの APGD は損失の勾配を利用するホワイトボックス攻撃，FAB Attack は分類境界との距離を利用するホワイトボックス攻撃，Square Attack はクエリベースのブラックボックス攻撃であり，それぞれ異なる特徴を持った攻撃となっている．

2.3　Fast Adversarial Training

Adversarial Training は，Adversarial Examples をトレーニングセットに追加することで，モデルが敵対的摂動に対してよりロバストになることを目指す手法である．この方法は式 (2.13) のミニマックス最適化問題として定式化される．この最適化問題では，AEs を生成する内側の最大化問題と，モデル重みパラメータを更新する外側の最小化問題を交互に解く．

$$\arg\min_{w} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\delta} \mathcal{L}(f_w(x+\delta), y) \right], \quad \text{s.t. } \|\delta\|_p \leq \epsilon \tag{2.13}$$

内側の最大化問題を解くためには，主に PGD や FGSM のような攻撃手法を用いて AEs を生成する．標準的な AT では多段階の攻撃である PGD を用いる．PGD では多段階の更新により摂動空間全体を効果的に探索することで強力な AEs を生成することができるため，モデルのロバスト性を向上させるのに適している．しかしながら，計算コストが非常に大きく，特に大規模なデータセットやモデルで訓練することが困難であるという課題もある．そこで代替手段として，単一段階の攻撃である FGSM を用いて内側最大化問題を近似した AT である Fast Adversarial Training が注目されている．FGSM は PGD と比較して計算コストが小さいため，標準的な AT に比べて FAT は小さな計算コストで訓練することができる．しかしながら，FGSM により生成される AEs は，摂動空間を単一段階でしか探索しないために品質が PGD に比べて劣り，壊滅的過学習が発生してしまうという課題もある．壊滅的過学習とは訓練の途中にモデルのロバスト性が突然失われる現象である．FGSM のような単一段階の攻撃は，摂動空間内を単一段階でしか探索せず，同じような摂動が繰り返し生成されるため，モデルがその特定の摂動を過学習し，より強力な PGD のような攻撃に対するロバスト性を失ってしまうと考えられる．壊滅的過学習を防ぐためにさまざまな改良手法が提案されている．例えば，摂動の初期化や，ハードラベルを緩和してソフトラベルにするラベル緩和，データの分類結果に基づき損失の大きさを調整する分類駆動型損失，モデルパラメータの更新を安定化させる収束制約などの手法が提案されている．本節では，これらの FAT で用いられる壊滅的過学習を抑制してロバスト性を向上させるための防御手法について述べる．

2.3.1　摂動初期化

Wong らは，FGSM で AEs を生成する際にランダムな初期化を導入する FGSM-Random Start（FGSM-RS）[9] を提案した．

$$\delta = \mathrm{Uniform}(-\epsilon, \epsilon) + \epsilon \cdot \mathrm{sign}(\nabla_x \mathcal{L}(f_w(x), y)) \tag{2.14}$$

ランダムな初期化を行うことにより，より多様な AEs を生成することができるため壊滅的過学習をある程度抑制できる．しかしながら，訓練が進むにつれて AEs の品質が低下し，壊滅的過学習が発生することもある．

そこで Jia らは，過去のエポック全体での勾配情報を，モメンタムを利用して蓄積し，蓄積されたモメンタムから初期摂動を生成する FGSM-Momentum of Epochs Prior（FGSM-MEP）[15] を提案した．式 (2.15) は現在の勾配 G を表す．また，式 (2.16) は現在の勾配を踏まえた k エポック目での勾配モーメンタム G_k を表す．ここで μ はモメンタムの減衰係数である．式 (2.17) は k エポック目での摂動 δ_k を表す．ここで初期摂動 η_k は式 (2.18) のように蓄積された勾配モメンタム G_k を利用して求められる．

$$G = \mathrm{sign}\left(\nabla_x \mathcal{L}(f_w(x + \eta_{k-1}), y)\right), \tag{2.15}$$

$$G_k = \mu \cdot G_{k-1} + G, \tag{2.16}$$

$$\delta_k = \Pi_{[-\epsilon,\epsilon]}\left[\eta_{k-1} + \alpha \cdot G\right], \tag{2.17}$$

$$\eta_k = \Pi_{[-\epsilon,\epsilon]}\left[\eta_{k-1} + \alpha \cdot \mathrm{sign}(G_k)\right]. \tag{2.18}$$

これは 2 段階の PGD を用いた AT では壊滅的過学習が発生しないことから，適切な摂動初期化を行った FAT でも壊滅的過学習が発生しないだろうという考察から提案された手法である．FGSM-MEP では過去の勾配情報のモメンタムから初期摂動を生成するため，FGSM-RS より高品質な AEs を生成するための初期化を行うことができ，さらなる壊滅的過学習の抑制とロバスト性の向上が実現した．さらに Jia らは，勾配モメンタムの計算に工夫を加え，FGSM-WMEP[16] を提案した．式 (2.19) のように AEs の品質を評価して，モメンタムにおける現在の勾配の重みを決定するパラメータ Γ を取り入れた．Γ は式 (2.20) のように求められる．

$$G_k = \mu \cdot G_{k-1} + \Gamma \cdot G \tag{2.19}$$

$$\Gamma = 2 - \frac{\mathrm{Acc}(f_w(x+\delta_k), y)}{\mathrm{Acc}(f_w(x+\eta_k), y)}, \tag{2.20}$$

Γ の値が大きい程，AEs の攻撃成功率が高く品質が良い．一方で Γ の値が小さい程，攻撃成功率が低く品質が悪い．現在の勾配 G に Γ を乗算することで，品質が良い AEs に対する勾配は大きくモメンタムに取り入れ，品質が悪い AEs に対する勾配はモメンタムにあまり取り入れないように動的に調整される．これは [16] において，AEs の攻撃成功率が大きく低下したのに伴ってロバスト性も急激に低下して壊滅的過学習が発生するという観察に基づいて設計された．さらに，FGSM-WMEP では重み平均化という手法も用いられている．次項では重み平均化 [16, 17] について述べる．

2.3.2　重み平均化

重み平均化はほとんど追加の計算コストをかけることなく，モデルの汎化性能を向上させられることが示されている．指数荷重移動平均（Exponential Moving Average: EMA）を用いた重み平均化 [16, 17] では，各イテレーションで平均化モデル重み $\tilde{w}$ はモデル重み w と減衰率 κ を用いて式 (2.21) のように表すことができる．

$$\tilde{w} = \kappa \cdot \tilde{w} + (1-\kappa) \cdot w \tag{2.21}$$

減衰率 κ は一般的に 0.999 とされる．標準的な PGD を用いた AT においても，重み平均化を組み合わせることでロバスト性が向上することが示された [18, 19]．これは重み平均化により平坦で滑らかな損失ランドスケープを形成することで，モデルのロバスト性を向上させられることが示された．損失ランドスケープとは，モデルパラメータに対する損失関数の値を可視化したものである．損失ランドスケープが平坦でなく急峻であるほど，敵対的摂動により損失の値が大きく変動してしまうため，ロバスト性が低いモデルとなる．このように標準的な AT で重み平均化が損失ランドスケープを平坦化することでロバスト性を向上させる一方，FAT と重み平均化を組み合わせると壊滅的過学習を悪化させ，ロバスト性の向上が限定的になることがある [16]．これは FAT の訓練後半では多くの重みが壊滅的過学習に苦しみ，それが最終モデルに蓄積してしまうためであると考えられる．そこでモデルのロバスト性に基づいて重みを更新するかどうかを決定する D-EMA が

提案された [16, 17]．モデルのロバスト性が大きい場合はさらに重みを更新すると過学習を起こす危険があるため，モデル重みの更新をスキップすることで壊滅的過学習を回避するという手法である．モデルのロバスト性は式 (2.22) のように決定される．

$$\Lambda = \frac{\mathrm{Acc}(f_w(x+\delta), y)}{\mathrm{Acc}(f_w(x+\eta_k), y)} \tag{2.22}$$

Λ は大きい程ロバスト性がすでに大きいことを表しており，式 (2.23) のように動的減衰率 $\tilde{\kappa}$ を決定する．

$$\tilde{\kappa} = \begin{cases} \kappa, & \Lambda < \nu, \\ 1, & \Lambda \geq \nu, \end{cases} \tag{2.23}$$

ここで ν は EMA 閾値であり，重みを更新するかどうかを決定するハイパーパラメータである．$\Lambda \geq \nu$ でロバスト性がすでに大きい時は $\tilde{\kappa} = 1$ で $\tilde{w} = \tilde{w}$ となり更新は行わない．

2.3.3　ラベル緩和

ラベル緩和とは，ワンホットのハードラベルを，正解クラスのラベルは 1 から γ に，不正解クラスのラベルは 0 から $\frac{1-\gamma}{C-1}$ に平滑化する技術であり，Szegedy らによって一般的なモデルの訓練での性能向上を目指して提案された [20]．特にクリーンデータに対する一般化性能を向上させるために用いる文脈では，ラベル平滑化とも呼ばれる．ここで γ はラベル緩和係数と呼ばれる．ラベルを平滑化することで，モデルの過学習を抑制し，一般化性能を向上させる手法であり，AT においても壊滅的過学習を抑制し，ロバスト性を向上させるための正則化手法として用いられる．具体的に緩和ラベル y' は式 (2.24) のように計算される．ここでも C は総クラス数を表す．

$$y' = y \cdot \gamma + (1-y) \cdot \frac{1-\gamma}{C-1} \tag{2.24}$$

さらに Gui らは，訓練が進み AEs の品質が次第に悪くなるにつれて，ラベルを緩和していき壊滅的過学習を抑制する動的ラベル緩和 [21, 22] を提案した．具体的には以下の式 (2.26) のようにラベル緩和係数 γ' を計算する．ここで e は現在のエポック，E は総エポック数，β はラベル緩和の強度を制御するハイパーパラメー

タ，$\gamma_{\min}$ は最小緩和係数である．

$$y' = y \cdot \gamma' + (1 - y) \cdot \frac{1 - \gamma'}{C - 1} \tag{2.25}$$

$$\gamma' = \max\left(\beta \cdot \tanh\left(1 - \frac{e}{E}\right), \gamma_{\min}\right) \tag{2.26}$$

このように訓練が進むにつれてラベル緩和係数を動的に小さくして，より強くラベルを緩和することで，品質の悪い AEs への過学習を防ぐ．

2.3.4　分類駆動型損失

Tong らは，AEs を 5 つのカテゴリーに分類して調査を行った [21, 22]．

1. クリーンデータがモデルに正しく分類され，対応する AEs がモデルに誤分類される．

2. クリーンデータがモデルに正しく分類され，対応する AEs もモデルに正しく分類される．

3. i クラスのクリーンデータがモデルに j クラスに誤分類され，対応する AEs もモデルに j クラスに誤分類される．

4. i クラスのクリーンデータがモデルに j クラスに誤分類され，対応する AEs はモデルに i クラスに正しく分類される．

5. i クラスのクリーンデータがモデルに j クラスに誤分類され，対応する AEs はモデルに k クラス（$k \neq i, j$）に誤分類される．

FGSM-AT 及び FGSM-RS での各ケースの訓練の様子を観察した結果，壊滅的過学習が発生した時の攻撃成功率は極めて低く，また「ラベル反転現象」が観察された．これは，多くの誤分類されたクリーンデータが攻撃後に正しいラベルに分類される，ケース 4 が増加する現象である．この状況では AT の内側の最大化問題が効果を失い，AEs が初期のモデルを誤分類させるという目標ではなく，外側の最小化問題を満たしてモデルが正しく分類できるように生成されることによると考えられる．誤分類されるクリーンデータは特にトレーニング初期段階では，全データの内の多くの割合を占めるため，これらを除去することなく活用する手法を考える．そこで誤分類されたデータの影響を考慮した分類駆動型損失（Taxonomy

Driven Loss）[21, 22] が提案された．分類駆動型損失 $\mathcal{L}_{\mathrm{TD}}$ は式 (2.27) のように正解クラスへのソフトマックス確率 p_y に基づいて正則加項の大きさを調整することで計算される．

$$\mathcal{L}_{\mathrm{TD}} = \mathcal{L}_{\mathrm{CE}} + \lambda \cdot \|f_w(x+\delta) - f_w(x+\eta_k)\|_2 \cdot \tanh(1-p_y) \tag{2.27}$$

ここで，正則化項 $\|f(x+\delta) - f(x+\eta_k)\|_2$ は入力データと AEs で同じような出力を生成するように機能し，モデルの決定境界を滑らかにしてロバスト性を向上させる．正解ラベルのソフトマックス確率に基づくスケール $\tanh(1-p_y)$ に比例して正則化の強度を調整することで，誤分類確率が高いクリーンデータに対しては正則化は強めて，集中的にロバスト性を向上させるように学習するという手法である．

さらに Gui らは，AEs を異なる損失範囲ごとに分けて FGSM-RS や FGSM-MEP でトレーニングを行った [22]．すると，大きな損失または小さな損失を持つトレーニング用の AEs の一部を除外して損失を集中化することで，クリーン精度とロバスト精度の両方が向上することが分かった．大きな損失を持つ AEs は，モデルがこれらのデータを過剰に学習してしまうことにより精度の低下を招くと考えられる．しかしながら高損失な AEs を過剰に除外すると，トレーニングデータの不足により精度が低下してしまう．また，小さな損失を持つデータは敵対的特徴が不足しており，ロバストな特徴を効率的に学習できないため，全体的な精度の低下につながると考えられる．そこで低損失な AEs を徐々に除外した結果，壊滅的過学習は発生せずにクリーン精度とロバスト精度の両方が向上することが分かった．以上の観察から，低損失なデータ，つまり正しく分類されているデータの損失を軽減することでそれらのデータの影響を緩和し，誤分類されているデータを重点的に学習してロバスト性の向上を目指す Catastrophic Overfitting Aware Loss Adaption（COLA）[22] という手法を提案した．バッチ内の入力データ x_i の AEs に対する損失を l_i として，式 (2.28) のように損失ベクトル Ψ を計算する．

$$\Psi = \mathcal{L}(f_w(x+\delta), y) = [l_1, l_2, \ldots, l_n]^T \tag{2.28}$$

さらに，損失セレクター Ω は式 (2.29) のように，正しく分類されたサンプルには損失調整 η（$\eta \in (0,1)$），誤分類されたサンプルには 1 を返し，誤分類されたデー

タを重点的に学習するよう選択する．

$$\Omega = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_n \end{bmatrix}, \quad \omega_i = \begin{cases} \eta, & \text{if } f_w(x_i) = y_i, \\ 1, & \text{otherwise.} \end{cases} \tag{2.29}$$

最終的な損失 $\mathcal{L}_{\mathrm{COLA}}$ は式 (2.30) のようになる．

$$\mathcal{L}_{\mathrm{COLA}} = \Psi^T \Omega \tag{2.30}$$

2.3.5　収束制約

Zhao らは，訓練エポック間での損失値の差を制限し，収束の過程を滑らかにすることで，モデルパラメータの不安定な更新を避け，壊滅的過学習を抑制する収束制約 [23] という手法を提案した．前回のエポック $t-1$ での入力データに初期摂動を加えたものに対する Cross-Entropy Loss の平均値 u_{t-1} を式 (2.31) のように求め，エポック間での損失の平均値の差を d_t，収束制約 $\mathcal{L}_{\mathrm{CS}}$ を式 (2.33) のように，現在のエポック t での入力データに初期摂動を加えたものに対する Cross-Entropy Loss と前回のエポックでの Cross-Entropy Loss の平均値 u_{t-1} との差とする．

$$u_{t-1} = \mathbb{E}_{(x,y)\sim D}\, \mathcal{L}_{\mathrm{CE}} \left(f_w(x + \eta_{k-1}) \right) \tag{2.31}$$

$$d_{t-1} = |u_{t-1} - u_{t-2}| \tag{2.32}$$

$$\mathcal{L}_{\mathrm{CS}} = |\mathcal{L}_{\mathrm{CE}} \left(f_w(x + \eta_k) \right) - u_{t-1}| \tag{2.33}$$

もし $\mathcal{L}_{\mathrm{CS}} \geq \min\left(\max\left(d_{t-1}, \gamma_{\min}\right), \gamma_{\max}\right)$ を満たす場合，つまり現在の損失と一つ前の損失との差が閾値以上のデータに対して，エポック間での損失差が大きいペナルティとして $\mathcal{L}_{\mathrm{CS}}$ を加え，エポック間での損失値の差を制限する手法である．ここで閾値 $\min\left(\max\left(d_{t-1}, \gamma_{\min}\right), \gamma_{\max}\right)$ は，訓練初期段階ではエポック間での損失差が大きく，次第に小さくなるという特性を踏まえて，最大閾値 $\gamma_{\max}$ から，エポック間での損失差 d_{t-1}，最小閾値 $\gamma_{\min}$ へと動的に変化する．この制約はもともと最大許容摂動強度 $\epsilon = 16$ などの大きい摂動に対して，学習が不安定になるために用いる手法である．本提案手法では，より一般的な $\epsilon = 8$ での学習性能の向上を目

標とするため，$\epsilon = 16$ のように学習が不安定になることは少なく，最終的に手法として採用はしなかった．しかしながら，パラメータを動的にすることで学習が不安定になることが考えられるため，そのような場合にも応用は可能であると考える．

2.4　既存研究の考察

本節では，これまでに述べた既存研究を踏まえた Adversarial Training への考察を述べる．Fast Adversarial Training における対処すべき課題は壊滅的過学習を防ぐことであると考える．この課題に対処するために大きく二つの手法が考えられる．一つ目は，攻撃性能が高く且つ多様な AEs を FGSM ベースの攻撃で生成する手法である．これは AEs の生成を工夫する方法である．多様な AEs は FGSM-RS[9] でも生成できる．しかしながらランダムな初期化点に摂動を加える手法は攻撃性能が低いため，訓練が進むと過学習が発生してしまうと考えられる．一方，FGSM-MEP[15] 及び FGSM-WMEP[16] は，過去のエポック全体での勾配情報から初期摂動を生成するため，多様且つ攻撃性能が高い AEs を生成でき，より優れた手法と考えられる．二つ目は，モデル性能に応じて学習の強度を調整する手法である．これはモデルの学習のためのパラメータ，つまり損失関数や正則化，ラベルやモデル重みの更新に工夫をする手法である．攻撃性能が低くモデルがすでに学習ができている AEs に対しては，ラベルを緩和したり，正則化の強度を弱めたりすることで過学習を引き起こし得る攻撃性能の低い AEs の影響を小さく抑える．これらの考察から，提案手法として摂動初期化には過去のエポック全体での勾配情報から初期摂動を生成する FGSM-WMEP を用いて，データの分類確率や分類結果に応じた分類駆動型損失や，バッチ単位での D-EMA，エポック単位での動的ラベル緩和により，モデル性能に応じて学習強度を動的に調整する手法を考える．

2.5　本章のまとめ

本章では，いくつかの Adversarial Examples の生成手法と既存の Fast Adversarial Training の手法について述べた．

2.2 節**「Adversarial Examples」**では，Adversarial Training の訓練及び評価で用いるいくつかの代表的な Adversarial Examples の生成手法について述べた．

2.3 節**「Fast Adversarial Training」**では，既存の Fast Adversarial Training の壊滅的過学習を抑制しロバスト性を向上させる手法の内，摂動初期化，重み平均化，ラベル緩和，分類駆動型損失関数，収束制約の 5 つについて説明した．

2.4 節**「既存研究の考察」**では，Fast Adversarial Training に関する既存研究から，どのような手法が FAT のロバスト性の向上に効果的であるか考察した．

第3章

パラメータの動的な調整を用いたFast Adversarial Trainingの提案

3.1　本章の概要

本章では，提案手法について述べた後，本論文で行った実験手法について述べる．そして実験を行った結果について示し，既存のベースラインとなる手法との比較及び評価を行う．

3.2 節**「提案手法」**では，既存研究への考察を踏まえて提案する，新たな Fast Adversarial Training 手法の概要について述べる．

3.3 節**「実験準備」**では，実験を行う環境や，実験で利用するデータセット，訓練時のパラメータ設定，評価に用いる攻撃手法やベースラインとする既存手法について述べる．

3.4 節**「実験結果と評価」**では，CIFAR10 及び CIFAR100 に対して行った実験結果を示し，提案手法とベースラインとの比較，評価を行う．

3.5 節**「本章のまとめ」**では，本章のまとめについて述べる．

3.2　提案手法

本節では，まず提案手法のベースとなる手法について述べる．続いて，提案手法についてベースとなる手法との違いを含めて述べる．

3.2.1　ベース手法

提案手法では大きく二つの手法をベースとする．一つ目は FGSM-WMEP[16] である．これは前章で説明した摂動初期化と動的重み平均化を取り入れた手法であり，アルゴリズム 1 のように表される．WMEP による摂動初期化は，epoch-reset ごとにリセットされ，ランダム初期化される．このことにより，特定の摂動パターンに偏った学習を回避する．また WMEP ではデータ拡張が行われる．データ拡張として，CIFAR10 及び CIFAR100 の時，ランダムクロップとランダムな左右反転が行われる．ランダムクロップとは，入力画像のランダムな領域を切り取る手法であり，ここでは入力としてパディングを 4 として 40×40 の CIFAR 画像を受け取り，ランダムな位置で 32×32 に切り取る．ランダムな左右反転とは，画像を反転させるかさせないかをランダムに決定し，反転させる場合のみ左右を反転させる．ラベル緩和係数 γ は訓練を通して一貫して同じ値であり，それに応じて緩和ラベル y' が生成される．現在の勾配 G は，クリーンデータに初期摂動を加えた $x + \eta_k$ の出力と緩和ラベルの Cross-Entropy 損失の x に関する勾配の符号として計算される．この現在の勾配をステップサイズ α の分だけ初期摂動に加えて，クリッピングしたものを k エポックでの敵対的摂動 δ_k として計算する．クリッピングとは，摂動を最大摂動許容強度 ϵ の摂動空間内に収める射影操作である．ここで現在のモデルの敵対的摂動に対するロバスト性を評価する指標 Λ 及び Γ を計算するために，クリーンデータ精度に対する AEs 精度の比を計算する．理想的にはモデルのクリーンデータの出力は $f_w(x)$ であるが，この場合新たに $f_w(x)$ の計算が必要になり追加の計算コストがかかる．そこで，近似的にクリーンデータに初期摂動を加えたもの $x + \eta_k$ の出力をクリーンデータの出力とする．このようにして計算した指標 Γ を用いて，現在の勾配をどの程度モメンタムに取り入れるか決定する．具体的には Γ が小さい程 AEs 精度が大きく，敵対的攻撃が成功していなく AEs の品質が低いため，現在の勾配はモメンタムにあまり取り入れない．反対

Algorithm 1 FGSM-WMEP

Require: クリーンデータ x，ワンホットラベル y，クラス数 C，学習率 μ，モメンタム初期摂動 η_k，モメンタム係数 μ_m，ラベル緩和係数 γ，緩和ラベル y'，スケーリング係数 β，EMA 閾値 ν，重みモメンタム係数 κ.

$k \leftarrow 1$
for epoch in **EPOCHS do**
  if epoch % epoch-reset == 0 **then**
    $\eta_k = U(-\epsilon, \epsilon)$
  end if
  for batch in **BATCHS do**
    $y' \leftarrow y \cdot \gamma + (y-1) \cdot \frac{\gamma - 1}{C-1}$
    $G = \mathrm{sign}(\nabla_x \mathcal{L}_{\mathrm{CE}}(f_w(x+\eta_k), y'))$
    $\delta_k = \Pi_{[-\epsilon,\epsilon]}[\eta_k + \alpha \cdot G]$
    $\Lambda = \frac{\mathrm{Acc}(f_w(x+\delta_k), y)}{\mathrm{Acc}(f_w(x+\eta_k), y_i)}$
    $\Gamma = 2 - \frac{\mathrm{Acc}(f_w(x+\delta_k), y)}{\mathrm{Acc}(f_w(x+\eta_k), y)}$
    $G_k = \mu_m \cdot G_{k-1} + \Gamma \cdot G$
    $\eta_{k+1} = \Pi_{[-\epsilon,\epsilon]}[\eta_k + \alpha \cdot \mathrm{sign}(G_k)]$
    $\mathcal{L} \leftarrow \mathcal{L}_{\mathrm{CE}}(f_w(x+\delta_k), y') + \lambda \cdot \mathcal{L}_{\mathrm{MSE}}(f_w(x+\eta_k), f_w(x+\delta_k))$
    $g_w \leftarrow \nabla_w \mathcal{L}$
    $w = w - \mu g_w$
    $\tilde{\kappa} = \begin{cases} \kappa & \text{if } \Lambda < \nu \\ 1 & \text{else} \end{cases}$
    $\tilde{w} = \tilde{\kappa} \cdot \tilde{w} + (1 - \tilde{\kappa}) \cdot w$
  end for
  $k \leftarrow k + 1$
end for
return parameter w'

に Γ が大きい程 AEs の品質が高いため，現在の勾配を多めにモメンタムに取り入れる．計算された k エポックでの勾配モメンタムから次のエポックでの初期摂動を計算する．学習のための損失 $\mathcal{L}$ は，分類損失と正則化項の和で計算される．ここでは正則化項として Mean Squared Error(MSE) が用いられており，バッチサイズを N として式 (3.1) のように計算される．

$$\mathcal{L}_{\mathrm{MSE}} = \frac{1}{N}\sum_{i=1}^{N}\left(f_w(x+\eta_k)_i - f_w(x+\delta_k)_i\right)^2 \tag{3.1}$$

MSE は二つの入力の類似度を測定するものであり，ここでは近似されたクリーンデータに対するモデル出力と AEs に対するモデル出力を入力として類似度を測定しており，類似している程 $\mathcal{L}_{\mathrm{MSE}}$ の出力は小さくなる．モデルは損失が小さくなるように学習を行うため，クリーンデータに対する出力と AEs に対する出力が類似するように学習を行い，摂動に対するロバスト性の向上を目的としている学習率スケジューラーとして MultiStepLR が採用されている．これはある特定のエポック（マイルストーン）ごとに学習率を段階的に減少させる手法である．学習初期段階では高めの学習率で安定的に大域的な探索を行い，訓練が進むにつれて学習が停滞してきた段階で学習率を減少させて局所的な調整を行う．具体的には，0 から 99 エポック目までは学習率 0.1，100 から 104 エポック目までは学習率 0.01，105 から 109 エポック目までは学習率 0.001 として学習を行う．ここで先程計算した，近似的なクリーンデータ精度に対する AEs 精度の比 Λ が EMA 閾値 ν より小さい，つまり十分 AEs の品質が良かった場合は EMA 重み w' の更新を行い，その他の場合は更新を行わない．以上の処理を繰り返し行うことで FGSM-WMEP はロバスト性を獲得していく．

もう一つの手法は ETA[22] である．これは，前章で説明した動的ラベル緩和と分類駆動型損失を取り入れた手法であり，アルゴリズム 2 のように表される．各エポックごとに現在のトレーニングの進行状況に応じてラベル緩和係数を決定する．トレーニングが進むほど攻撃成功率が下がり AEs の品質が低下するためラベルを緩和して過学習を抑制する．各バッチのサンプルごとに，正解クラス c の Softmax 確率を計算する．現在の勾配 G は，クリーンデータにバッチモメンタム初期摂動を加えた $x+\delta_{m-1}$ の出力と計算された緩和ラベル y' の Cross-Entropy 損失の x に関する勾配の符号として計算され，損失の勾配を用いて敵対的摂動 δ

Algorithm 2 ETA(Example Taxonomy Aware Fast Adversarial Training)

Require: クリーンデータ x，ワンホットラベル y，クラス数 C，学習率 μ，バッチモメンタム初期摂動 δ_{m-1}，バッチモメンタム係数 μ_b，ラベル緩和係数 γ'，緩和ラベル y'，スケーリング係数 β，損失調整 η.

$\delta_0 = U(-\epsilon, \epsilon)$
for epoch in **EPOCHs do**
  $\gamma' \leftarrow \max(\beta \cdot \tanh(1 - \frac{\text{epoch}}{\text{EPOCHs}}), \gamma_{\min})$
  $m \leftarrow 1$
  for batch in **BATCHes do**
    $p_y \leftarrow \text{Softmax}(f_w(x + \delta_{m-1}))\,[c]$, where $y[c] = 1$
    $y' \leftarrow y \cdot \gamma' + (y - 1) \cdot \frac{\gamma' - 1}{C - 1}$
    $G \leftarrow \text{sign}(\nabla_x \mathcal{L}_{\text{CE}}(f_w(x + \delta_{m-1}), y'))$
    $\delta \leftarrow \Pi_{[-\epsilon,\epsilon]}[\delta_{m-1} + \alpha \cdot G]$
    $\mathcal{L}_{\text{TD}} \leftarrow \mathcal{L}_{\text{CE}}(f_w(x+\delta), y') + \lambda_{\text{TD}} \cdot \|f_w(x+\delta) - f_w(x+\delta_{m-1})\|_2 \cdot \tanh(1-p_y)$
    $\Omega \leftarrow [\eta \quad \text{if } f_w(x + \delta_{m-1}) = y \quad \text{else } 1]$
    $\Psi \leftarrow [l_1, l_2, \ldots, l_i, \ldots, l_{\text{N}}]$
    $g_w \leftarrow \nabla_w \mathcal{L}_{\text{COLA}}$
    $w \leftarrow w - \mu g_w$
    $\delta_m \leftarrow \mu_b \cdot \delta_{m-1} + (1 - \mu_b) \cdot \delta$
    $m \leftarrow m + 1$
  end for
end for
return parameter w

も計算される．学習のための損失として分類駆動型損失を採用する．式 (2.27) のように，分類損失としてここでも Cross-Entropy 損失を採用し，正則化項として $\|f_w(x+\delta)-f_w(x+\delta_{m-1})\|_2 \cdot \tanh(1-p_y)$ を採用する．$\|f_w(x+\delta)-f_w(x+\delta_{m-1})\|_2$ 部分は MSE と類似しており目的は同じで，クリーンデータに対する出力と AEs に対する出力が類似するようにするための正則化項である．$\tanh(1-p_y)$ は各クリーンデータの正解ラベルへのソフトマックス確率 p_y に応じて正則化強度を調整するスケーリング項であり，誤分類確率が高いクリーンデータに対しては集中的にロバスト性を向上するよう学習させる．さらにこの損失 $\mathcal{L}_{\mathrm{TD}}$ を損失セレクター Ω と内積を取ることで最終的な分類駆動型損失 $\mathcal{L}_{\mathrm{COLA}}$ を計算する．ここで Ω は近似的クリーンデータが正しく分類された場合は η，誤分類された場合は 1 を返すことで，誤分類された損失を集中的に学習するように調整する．このようにして得られた分類駆動型損失から学習を行う．バッチ処理の最後にバッチモメンタム係数 μ_b を用いて，バッチモメンタムを計算する．以上の処理を繰り返し行うことで ETA はロバスト性を獲得していく．

3.2.2　提案手法

本項では，前の項で述べたベースとなる手法から新たに WMEP-ETA という手法を提案する．これは，摂動初期化，動的重み平均化，動的ラベル緩和，分類駆動型損失を組み合わせた手法であり，アルゴリズム 3 のように表される．本論文のテーマであるパラメータの動的な調整とは，モデルのロバスト性に基づいて，初期摂動生成時に現在の勾配をモメンタムにどの程度取り入れるかを決定したり，EMA による重み更新の有無を決定したりする他にも，トレーニングの進行状況に応じてラベル緩和係数を決定したり，データの分類結果に応じて正則化強度を決定したりすることを指す．このようにパラメータを動的に調整する既存手法を組み合わせた提案手法がロバスト性の向上につながることを示していく．まず各エポックごとに $\gamma' = \max(\beta \cdot \tanh(1-\frac{\mathrm{epoch}}{\mathrm{EPOCHs}}), \gamma_{\min})$ で動的ラベル緩和係数を計算する．各バッチで WMEP 初期摂動による敵対的摂動 δ_k を生成する．近似的クリーンデータ $x+\eta_k$ の正解クラスへのソフトマックス確率に基づき，損失 $\mathcal{L}_{\mathrm{TD}}$ を計算し，この損失 $\mathcal{L}_{\mathrm{TD}}$ を損失セレクター Ω と内積を取ることで最終的な分類駆動型損失 $\mathcal{L}_{\mathrm{COLA}}$ を計算する．ここで Ω は先ほどの手法とは異なり，近似的クリーン

データである $x+\eta_k$ の分類ではなく，AEs である $x+\delta_k$ の分類に基づいて，正しく分類された場合は η，誤分類された場合は 1 を返すように変更する．これはもともと敵対的特徴の少ない低損失な AEs の影響を小さくすることで，ロバスト性の向上を目指す手法であり，クリーンデータの分類ではなく AEs の分類に基づいて影響を緩和するかどうか決定した方が良いと考えられるためこのような変更をした．またこの際，あらかじめ $f_w(x+\delta_k)$ は計算されているため追加の計算コストはない．このようにして計算された最終的な分類駆動型損失から学習を行い，モデルのロバスト性の指標 Λ が EMA 閾値 ν に満たなかった場合は，AEs の品質も良いため EMA モデルパラメータの更新を行い，そうでない場合は，壊滅的過学習を引き起こさないように EMA モデルパラメータの更新を行わない．また本提案手法においては，WMEP で行ったものと同じデータ拡張を行う．

Algorithm 3（提案）WMEP-ETA

Require: クリーンデータ x，ワンホットラベル y，クラス数 C，学習率 μ，モメンタム初期摂動 η_k，モメンタム係数 μ_m，ラベル緩和係数 γ'，緩和ラベル y'，スケーリング係数 β，損失調整 η，EMA 閾値 ν，重みモメンタム係数 κ.

$k \leftarrow 1$
for epoch in **EPOCHs do**
  if epoch % epoch-reset == 0 **then**
    $\eta_k = U(-\epsilon, \epsilon)$
  end if
  $\gamma' \leftarrow \max(\beta \cdot \tanh(1 - \frac{\text{epoch}}{\text{EPOCHs}}), \gamma_{\min})$
  for batch in **BATCHes do**
    $p_y \leftarrow \text{Softmax}(f_w(x + \eta_k))\,[c]$, where $y[c] = 1$
    $y' \leftarrow y \cdot \gamma' + (y - 1) \cdot \frac{\gamma' - 1}{C - 1}$
    $G = \text{sign}(\nabla_x \mathcal{L}_{\text{CE}}(f_w(x + \eta_k), y'))$
    $\delta_k = \Pi_{[-\epsilon,\epsilon]}[\eta_k + \alpha \cdot G]$
    $\Lambda = \frac{\text{Acc}(f_w(x+\delta_k), y)}{\text{Acc}(f_w(x+\eta_k), y_i)}$
    $\Gamma = 2 - \frac{\text{Acc}(f_w(x+\delta_k), y)}{\text{Acc}(f_w(x+\eta_k), y)}$
    $G_k = \mu_m \cdot G_{k-1} + \Gamma \cdot G$
    $\eta_{k+1} = \Pi_{[-\epsilon,\epsilon]}[\eta_k + \alpha \cdot \text{sign}(G_k)]$
    $\mathcal{L}_{\text{TD}} \leftarrow \mathcal{L}_{\text{CE}}(f_w(x+\delta_k), y') + \lambda_{\text{TD}} \cdot \|f_w(x+\delta_k) - f_w(x+\eta_k)\|_2 \cdot \tanh(1 - p_y)$
    $\Omega \leftarrow [\eta \quad \text{if } f_w(x + \delta_k) = y \quad \text{else } 1]$
    $\Psi \leftarrow [l_1, l_2, \ldots, l_i, \ldots, l_{\text{N}}]$
    $g_w \leftarrow \nabla_w \mathcal{L}_{\text{COLA}}$
    $w \leftarrow w - \mu g_w$
    $\tilde{\kappa} = \begin{cases} \kappa & \text{if } \Lambda < \nu \\ 1 & \text{else} \end{cases}$
    $\tilde{w} = \tilde{\kappa} \cdot \tilde{w} + (1 - \tilde{\kappa}) \cdot w$
  end for
  $k \leftarrow k + 1$
end for
return parameter w'

3.3　実験準備

本節では，実装環境について述べた後，使用する 2 つのデータセットごとのパラメータ設定について述べる．続いて防御手法の評価のために用いる攻撃手法や，評価基準について述べる．最後に提案手法の比較のためのベースラインとする手法について述べる．

3.3.1　実装環境

実験を行う実装環境を表 3.1 にまとめた．実験は Google Colaboratory で Python 3.11.11，PyTorch 2.5.1 を主に使用して行い，GPU としては高性能な A100GPU を用いて行った．

表 3.1: 実装環境.

OS	Ubuntu 22.04.3 LTS (Jammy Jellyfish)
CPU	Intel(R) Xeon(R) CPU @ 2.20GHz
CPU メモリ (RAM)	50.0 GiB
GPU	NVIDIA A100-SXM4-40GB
GPU メモリ	40.0 GB
Python	3.11.11
PyTorch	2.5.1
CUDA	12.1
cuDNN	8.9.6
Platform	Google Colaboratory PRO+

3.3.2　データセットと訓練設定

各データセットごとに，表 3.2 のようにパラメータを設定して実験を行った．これらのパラメータは先行研究 [15, 16, 21, 22] に則って決定した．固定ラベル緩和係数 γ は動的ラベル緩和を行わない時に用い，動的ラベル緩和を行うときは動的ラベル最小緩和係数 $\gamma_{\min}$ 及びラベルスケーリング β を用いて動的ラベル緩和係数 γ' を計算した．正則化強度は $\mathcal{L}$ と $\mathcal{L}_{\mathrm{TD}}$ で正則化項が異なるため用いる損失に応じて強度を決定する．また，EMA 閾値 ν 及びモーメンタム係数 μ_m は WMEP で初

表 3.2: パラメータ設定.

パラメータ名	**CIFAR10**	**CIFAR100**
データセット	CIFAR10	CIFAR100
モデル名	ResNet18	ResNet18
バッチサイズ N	128	128
総エポック数 EPOCHs	110	110
エポックリセット epoch-reset	10	40
学習率スケジューラー	multistep	multistep
マイルストーン 1	100	100
マイルストーン 2	105	105
初期重み学習率 μ_0	0.1	0.1
重み減衰	5×10^{-4}	5×10^{-4}
モーメンタム	0.9	0.9
固定ラベル緩和係数 γ	0.5	0.8
動的ラベル最小緩和係数 $\gamma_{\min}$	0.15	0.05
ラベルスケーリング β	0.6	0.6
正則化強度 λ	5.6	1
TD 正則化強度 λ_{TD}	0.65	0.05
EMA 閾値 ν	0.82	0.82
モーメンタム係数 μ_m	0.3	0.1
バッチモーメンタム係数 μ_b	0.75	0.75
損失調整 η	0.75	0.75
最大許容摂動強度 ϵ	8	8
摂動ステップサイズ α	8	8
摂動初期化	random	random

期摂動を生成する時に使用し，WMEP を用いずにバッチモメンタムを用いるときはバッチモーメンタム係数 μ_b を用いる．また損失調整 η は COLA で誤分類させられない AEs の影響を軽減する時のみに用いる．

3.3.3　評価手法

評価手法として，まず用いる攻撃は FGSM，PGD10，PGD20，PGD50，CW-PGD20，AutoAttack の 6 つの攻撃を用いる．PGDn 攻撃は n 回反復を行う PGD 攻撃を表す．CW-PGD20 攻撃は PGD20 により摂動を生成し CW Attack で最適化を行う攻撃を表す．また，AutoAttack における攻撃の成功はアンサンブルされる 4 つの攻撃のいずれかが成功すればよい．したがって攻撃成功率が高く計算コストの低いホワイトボックス攻撃である Auto-PGD から攻撃を行い，攻撃が成功したデータは除外して次の攻撃を行うことで計算効率を高める．各防御手法に対して，クリーンデータ及び 6 つの攻撃で生成した AEs の分類正解率を求め，分類正解率が高いモデル程優れていると評価する．さらに，先行研究 [15, 16, 21, 22] に則って，各防御手法から二つの「best」モデルと「last」モデルを提出する.「best」モデルはトレーニング中に PGD10 に対して最もロバストであった時，つまり PGD10 で生成した AEs の分類正解率が全体の中で最も高かった時のモデルである.「last」モデルはトレーニングがすべて完了した時のモデルである．この二つのモデルを比較することで，ピーク性能と最終性能のギャップを多面的に評価できる．

3.3.4　ベースライン

提案手法の性能を評価するためのベースラインとして，FGSM-WMEP[16]，TDAT[21]，ETA[22] の 3 つの FAT 手法を用いる．提案手法 WMEP-ETA の評価に際して，動的ラベル緩和（DL），分類駆動型損失の $\mathcal{L}_{\mathrm{TD}}$（TD）及び $\mathcal{L}_{\mathrm{COLA}}$（COLA）の 3 つの手法がそれぞれどのような役割を持つかを評価するために，CIFAR10 データセットに対しては，WMEP-DL，WMEP-TD，WMEP-COLA，WMEP-DL+TD（WMEP-TDAT），WMEP-TD+COLA，WMEP-DL+COLA，WMEP-DL+TD+COLA（WMEP-ETA）に分けて提案手法として評価を行う．ここで TDAT は ETA において COLA を行わない手法で，ETA の基になった手法である．TDAT はアルゴリズム 4 のように表される．

Algorithm 4 TDAT(Taxonomy Driven Fast Adversarial Training)

Require: クリーンデータ x，ワンホットラベル y，クラス数 C，学習率 μ，バッチモメンタム初期摂動 δ_{m-1}，バッチモメンタム係数 μ_b，ラベル緩和係数 γ'，緩和ラベル y'，スケーリング係数 β.

$\delta_0 = U(-\epsilon, \epsilon)$
for epoch in **EPOCHs do**
  $\gamma' \leftarrow \max(\beta \cdot \tanh(1 - \frac{\text{epoch}}{\text{EPOCHs}}), \gamma_{\min})$
  $m \leftarrow 1$
  for batch in **BATCHes do**
    $p_y \leftarrow \text{Softmax}(f_w(x + \delta_{m-1}))\,[c]$, where $y[c] = 1$
    $y' \leftarrow y \cdot \gamma' + (y - 1) \cdot \frac{\gamma' - 1}{C - 1}$
    $G \leftarrow \text{sign}(\nabla_x \mathcal{L}_{\text{CE}}(f_w(x + \delta_{m-1}), y'))$
    $\delta \leftarrow \Pi_{[-\epsilon,\epsilon]}[\delta_{m-1} + \alpha \cdot G]$
    $\mathcal{L}_{\text{TD}} \leftarrow \mathcal{L}_{\text{CE}}(f_w(x+\delta), y') + \lambda_{\text{TD}} \cdot \|f_w(x+\delta) - f_w(x+\delta_{m-1})\|_2 \cdot \tanh(1-p_y)$
    $g_w \leftarrow \nabla_w \mathcal{L}_{\text{TD}}$
    $w \leftarrow w - \mu g_w$
    $\delta_m \leftarrow \mu_b \cdot \delta_{m-1} + (1 - \mu_b) \cdot \delta$
    $m \leftarrow m + 1$
  end for
end for
return parameter w

3.4　実験結果と評価

本節では 2 つのデータセット CIFAR10 及び CIFAR100 を用いて行った実験結果を示し，提案手法の評価を行う．CIFAR10 データセットの画像の例を図 3.1 に示した．今回はこのような 32×32 の解像度の小さいデータセットでしか実験できなかったため，今後は ImageNet などの解像度の大きいデータセットに対しても実験を行いたい．

図 3.1: CIFAR10 画像の例 [10].

3.4.1　CIFAR10

CIFAR10 データセットを用いて，表 3.2 で示したパラメータ設定で提案手法及びベースラインとなる手法で FAT を行い，クリーン精度及び 6 つの攻撃に対するロバスト精度を測定した結果を表 3.3 に示した．また，各提案手法を WMEP での精度と比較した時の精度の増減率（%）を表 3.4 に示した．

表 3.3: CIFAR10 でのクリーン精度及びロバスト精度の比較.

	Clean		FGSM		PGD10		PGD20		PGD50		CW-PGD20		AutoAttack		Time
Method	Best	Last	Best	Last	Best	Last	Best	Last	Best	Last	Best	Last	Best	Last	(m)
WMEP	0.8064	0.8099	0.6443	0.6460	0.5607	0.5597	0.5542	0.5528	0.5517	0.5520	**0.5114**	**0.5119**	**0.4951**	**0.4947**	42
TDAT (DL+TD)	0.8219	0.8219	0.6652	0.6662	0.5694	0.5697	0.5632	0.5632	0.5615	0.5611	0.5035	0.5033	0.4855	0.4859	60
ETA (DL+TD+COLA)	**0.8226**	**0.8236**	**0.6692**	**0.6690**	0.5732	0.5709	0.5642	0.5638	0.5612	0.5598	0.5036	0.5021	0.4755	0.4718	60
（提案）*ETA (DL+TD+*COLA)	**0.8258**	**0.8299**	**0.6704**	**0.6691**	0.5808	0.5780	0.5744	0.5715	0.5727	0.5700	0.4984	0.5013	0.4804	0.4814	64
（提案）WMEP+DL	0.7848	0.7848	0.6442	0.6428	0.5750	0.5750	0.5706	0.5703	0.5694	0.5697	**0.5128**	**0.5129**	**0.4987**	**0.4986**	41
（提案）WMEP+TD	0.8050	0.8050	0.6442	0.6430	0.5543	0.5547	0.5500	0.5502	0.5481	0.5483	0.5094	0.5091	0.4924	0.4922	42
（提案）WMEP+*COLA	0.8009	0.8009	0.6462	0.6458	0.5726	0.5735	0.5682	0.5676	0.5670	0.5662	0.5057	0.5055	0.4921	0.4920	42
（提案）WMEP+TDAT	0.8164	0.8181	0.6603	0.6613	0.5769	0.5767	0.5712	0.5707	0.5701	0.5688	**0.5119**	**0.5118**	0.4953	**0.4940**	45
（提案）WMEP+DL+*COLA	0.7994	0.7994	0.6552	0.6556	**0.5872**	**0.5872**	**0.5828**	**0.5829**	**0.5811**	**0.5828**	0.5076	0.5075	0.4933	0.4934	40
（提案）WMEP+TD+*COLA	0.8130	0.8161	0.6559	0.6562	0.5701	0.5711	0.5652	0.5643	0.5630	0.5630	0.5069	0.5070	0.4900	0.4904	41
（提案）WMEP+*ETA	0.8146	0.8146	0.6632	0.6631	**0.5832**	**0.5832**	**0.5785**	**0.5783**	**0.5764**	**0.5767**	0.5067	0.5068	0.4896	0.4896	41

WMEP では損失関数 $\mathcal{L} = \mathcal{L}_{\mathrm{CE}}(f_w(x+\delta_k), y') + \lambda \cdot \mathcal{L}_{\mathrm{MSE}}(f_w(x+\delta_k), f_w(x+\eta_k))$ を用いていたが，提案手法では $\mathcal{L} = \mathcal{L}_{\mathrm{CE}}(f_w(x+\delta_k), y') + \lambda_{\mathrm{TD}} \cdot \| f_w(x+\delta_k) - f_w(x+\eta_k) \|_2$

表 3.4: 各提案手法の WMEP と比較した精度の増減率（%）.

	Clean		FGSM		PGD10		PGD20		PGD50		CW-PGD20		AutoAttack	
Method	Best	Last	Best	Last	Best	Last	Best	Last	Best	Last	Best	Last	Best	Last
（提案）WMEP+DL	-2.679	-3.099	-0.326	-0.495	+2.550	+2.734	+2.959	+3.166	+3.208	+3.207	+0.274	+0.195	+0.727	+0.788
（提案）WMEP+TD	-0.174	-0.605	-0.016	-0.464	-1.141	-0.893	-0.758	-0.470	-0.653	-0.670	-0.391	-0.547	-0.545	-0.505
（提案）WMEP+*COLA	-0.682	-1.111	+0.295	-0.031	+2.122	+2.466	+2.526	+2.677	+2.773	+2.572	-1.115	-1.250	-0.606	-0.546
（提案）WMEP+TDAT	+1.240	+1.012	+2.483	+2.368	+2.889	+3.037	+3.067	+3.238	+3.335	+3.043	+0.098	-0.020	+0.040	-0.141
（提案）WMEP+DL+*COLA	-0.868	-1.296	+1.692	+1.486	+4.726	+4.913	+5.161	+5.445	+5.329	+5.580	-0.743	-0.860	-0.364	-0.263
（提案）WMEP+TD+* COLA	+0.818	+0.766	+1.800	+1.579	+1.676	+2.037	+1.985	+2.080	+2.048	+1.993	-0.880	-0.957	-1.030	-0.869
（提案）WMEP+*ETA	+1.017	+0.580	+2.933	+2.647	+4.013	+4.199	+4.385	+4.613	+4.477	+4.475	-0.919	-0.996	-1.111	-1.031

を用いる．もし分類駆動型損失にする場合は $\mathcal{L}_{\mathrm{TD}} = \mathcal{L} \cdot \tanh(1 - p_y)$ とする．また，従来の COLA ではクリーンデータの分類結果に応じて損失セレクター Ω を決定していたが，提案手法で用いる COLA では敵対的摂動を加えてた AEs である $x + \delta_k$ の分類結果に応じて損失セレクター Ω を決定する．明示的に提案手法の COLA は*COLA と表記する．

まず，WMEP に動的ラベル緩和を単体で導入した提案手法 WMEP+DL を評価する．通常の WMEP では固定ラベル緩和係数 $\gamma = 0.5$ が用いられていたが，ここでは動的ラベル緩和係数 $\gamma' = \max(\beta \cdot \tanh(1 - \frac{\mathrm{epoch}}{\mathrm{EPOCHs}}), \gamma_{\min})$ を用いた．この時 $\beta = 0.6$ 及び $\gamma_{\min} = 0.15$ である．訓練エポックが進むにつれてラベル緩和を強めて特にトレーニング後半での過学習を抑制する手法である．動的ラベル緩和を導入するとクリーン精度が大きく低下する一方，PGD 精度及び CW Attack, AutoAttack 精度が上昇した．これはラベル緩和により正解ラベルへの学習が緩和されるためクリーン精度は少し低下してしまう代わりに，不確実性を残した形で学習を行うため壊滅的過学習を抑制し，ノイズが加わっても出力が大きく変わることがなくなりロバスト性を確保するようになったと考えられる．

次に，WMEP の損失関数 $\mathcal{L}$ を $\mathcal{L}_{\mathrm{TD}}$ に置き換えた提案手法 WMEP+TD を評価する．もともとの損失関数 $\mathcal{L} = \mathcal{L}_{\mathrm{CE}}(f_w(x+\delta_k), y') + \lambda \cdot \mathcal{L}_{\mathrm{MSE}}(f_w(x+\delta_k), f_w(x+\eta_k))$ を $\mathcal{L}_{\mathrm{TD}} = \mathcal{L}_{\mathrm{CE}}(f_w(x+\delta_k), y') + \lambda_{\mathrm{TD}} \cdot \|f_w(x+\delta_k) - f_w(x+\eta_k)\|_2 \cdot \tanh(1 - p_y)$ に置き換える．ここでは動的ラベル緩和ではなく固定のラベル緩和が行われている．新たな損失関数では，クリーンデータの正解ラベルのソフトマックス確率に応じて正則化強度を調整するため，トレーニングに悪影響を与える可能性のある，誤分類されやすいクリーンデータに対してはペナルティを課し，重点的な学習を行

うという手法である．しかし，この手法はクリーン精度及びロバスト精度ともに低下した．原因究明にはさらなる実験が必要であるが，これはそもそも誤分類されやすい p の低いデータに対してペナルティを課し，誤分類されやすいデータの学習を強化すること自体がトレーニング全体で見たら精度向上にはつながらない可能性がある．さらに言うとトレーニング前半では誤分類されやすいデータにペナルティを課すことが学習の補助になるとしても，トレーニング後半では誤分類されやすいデータは異常があるデータの可能性が高まり，その学習を強化することはトレーニングに悪影響を及ぼす可能性もある．今後の研究では，まずスケーリング関数のスケーリングの大きさが適切かどうかを吟味したり，訓練段階に応じた適用の影響を調査したり，誤分類されやすいクリーンデータをどのように扱うことが適切であるかをさらに模索していく必要があると考える．

次に，WMEP の損失関数 $\mathcal{L}$ を COLA により調整する提案手法 WMEP+*COLA を評価する．これは，$\mathcal{L} = \mathcal{L}_{\mathrm{CE}}(f_w(x+\delta), y') + \lambda_{\mathrm{TD}} \cdot \|f_w(x+\delta_k) - f_w(x+\eta_k)\|_2$ に対して，$\mathcal{L}_{\mathrm{COLA}} = \mathcal{L}^T\Omega$ で，損失セレクター Ω により，正しく分類できている AEs の学習より，正しく分類できていない AEs の学習に集中する手法である．この手法は PGD に対してはロバスト精度を上昇させたが CW Attack や AutoAttack に対してはロバスト精度を低下させた．これは COLA により損失範囲の集中が阻害され，特に強力な AEs に対しての性能が低下したと考えられる．COLA では正しく分類された AEs の損失を軽減するが，正しく分類された AEs はもともと小さい値であり，誤分類された AEs は反対に大きい値である傾向にある．[22] では損失範囲の上限と下限を適切に設定することがロバスト性の向上につながることが示されており，COLA では損失範囲の集中化に反した調整を行う．損失範囲が集中化していないことにより，重点データの摂動を強く学習して PGD 精度は上昇するが，非重点データにより生じるモデルの穴を突かれることで強力な CW Attack や AutoAttack への精度は低下すると考えられる．

次に，WMEP に動的ラベル緩和と TD 損失を導入した提案手法 WMEP+TDAT（DL+TD）を評価する．TD 損失は先ほど述べた通り，単体では全く性能向上に寄与しなかったが，動的ラベル緩和と組み合わさることで動的ラベル緩和単体よりも優れた性能を示すようになる．動的ラベル緩和では訓練が進み AEs の品質が悪くなるにつれてラベルを緩和していき，TD 損失では誤分類されやすいクリーン

データの学習を強化する．TD により性能が向上しなかった理由として，トレーニング後半でも誤分類されやすい何かしらの異常のあるデータに対する学習を強化したことであると仮定した時，動的ラベル緩和によりトレーニング後半ではラベルを緩和することにより過学習の抑制がなされ，これらが補完的に作用し，DL と TD を組み合わせることでそれぞれ単体の手法より性能が上昇したと考えられる．

次に，WMEP に動的ラベル緩和と COLA 損失を導入した提案手法 WMEP+DL+*COLA を評価する．COLA では正しく分類できていない AEs の学習に集中することで，PGD 攻撃に対するロバスト性の向上がみられた．ここでも訓練後半でも正しく分類できていない AEs はトレーニングに悪影響を及ぼす異常値である可能性もあり，それらに対する学習も強化してしまう一方で，動的ラベル緩和によりトレーニング後半では強くラベルを緩和することで異常値への過学習を防ぎ性能の向上につながると考えられる．特に PGD に対するロバスト性の向上が顕著であり，提案手法の中で PGD 攻撃に対するロバスト性は最高性能である．これは，動的ラベル緩和により正しく分類されるデータと誤分類されるデータで分散していた損失がその中間値に集中するようになり，COLA により正しく分類できていない AEs の学習に集中する過程で分散してしまう損失範囲を調整してくれるためであるとも考えられる．しかし COLA は CW Attack や AutoAttack にはやや脆弱であり，動的ラベル緩和によりその脆弱性も少し解消されるが，WMEP よりは低いロバスト性となっている．

次に，WMEP に TD 損失と COLA 損失を導入した提案手法 WMEP+TD+*COLA を評価する．WMEP+*COLA の結果と比較すると，TD 損失を導入したことによりクリーン精度と FGSM ロバスト精度は上昇したものの，PGD 精度及び AutoAttack 精度は低下した．これは，TD 損失は誤分類されやすいクリーンデータを集中的に学習するようにし，COLA でも誤分類された AEs の学習に集中するようにするため，高損失な難易度の高いデータを過学習してしまうため，COLA 単体の時よりも精度が低下し，クリーン精度及び FGSM ロバスト精度は上昇するという結果になったと考えられる．

最後に，WMEP に動的ラベル緩和及び TD 損失と COLA 損失を導入した提案手法 WMEP+*ETA（DL+TD+*COLA）を評価する．これまでの提案手法で最も優れたロバスト精度を誇る WMEP+DL+*COLA と比較すると，この手法では TD

損失を導入することにより誤分類されやすいクリーンデータを集中的に学習することで比較的に過学習気味になり，クリーン精度は約 1.8%，FGSM ロバスト精度は約 1.2%上昇するが，PGD 精度は約 0.7%ほど減少してしまう．どちらの手法が優れているのかは優劣つけがたく，クリーン精度を優先する時は WMEP+*ETA，ロバスト精度を優先する時は WMEP+DL+*COLA のが優れていると考えられる．

図 3.2 にベースライン手法（WMEP，TDAT）と提案手法の（*ETA，WMEP-DL+*COLA，WMEP-*ETA）トレーニング中の精度の推移を示した．TDAT と *ETA は各エポックごとに精度の上下が大きく不安定な学習が続き，multistepLR により学習率が小さくなる 100 エポック目で急激に精度が良くなる．一方 WMEP 及び提案手法 WMEP-DL+*COLA，WMEP-*ETA は安定した学習曲線を示す．TDAT と*ETA ではバッチモメンタムによる初期摂動を用いて AEs を生成しており，これまでのバッチで生成した摂動が今回のバッチで適切な初期摂動とは限らず，初期摂動のランダム性が強い．このことにより決定境界の揺らぎが大きいが，学習率が減少することで決定境界が整い精度が向上すると考えられる．一方 WMEP 及び提案手法 WMEP-DL+*COLA，WMEP-*ETA ではサンプルごとに過去の全エポックの摂動のモメンタムによる初期摂動を用いて AEs を生成しており，初期摂動のランダム性が解消され，より一貫した方向に初期摂動を生成する．より適切な初期摂動による一貫性のある AEs を学習でき，安定した学習ができると考えられる．

3.4.2　CIFAR100

CIFAR100 データセットを用いて，表 3.2 で示したパラメータ設定で提案手法及びベースラインとなる手法で FAT を行い，クリーン精度及び 6 つの攻撃に対するロバスト精度を測定した結果を表 3.5 に示した．

表 3.5: CIFAR100 でのクリーン精度及びロバスト精度の比較.

	Clean		FGSM		PGD10		PGD20		PGD50		CW-PGD20		AutoAttack		Time
Method	Best	Last	Best	Last	Best	Last	Best	Last	Best	Last	Best	Last	Best	Last	(m)
WMEP	**0.5866**	**0.6275**	0.3889	0.3887	0.3111	0.2881	0.3052	0.2806	0.3037	0.2793	0.2770	0.2580	0.2536	0.2351	43
TDAT (DL+TD)	0.5772	0.5723	0.4121	0.4071	0.3417	0.3377	0.3369	0.3330	0.3347	0.3315	0.2869	0.2874	0.2633	0.2630	43
（提案）*ETA (DL+TD+*COLA)	0.5826	0.5775	0.4116	0.4086	0.3410	0.3398	0.3374	0.3349	0.3367	0.3346	0.2852	0.2844	0.2655	0.2651	51
（提案）WMEP+DL+*COLA	0.5745	0.5766	**0.4149**	**0.4136**	**0.3490**	**0.3473**	**0.3457**	**0.3436**	**0.3441**	**0.3425**	**0.2892**	**0.2889**	**0.2692**	**0.2681**	43
（提案）WMEP+*ETA	**0.5862**	**0.5892**	**0.4174**	**0.4178**	**0.3479**	**0.3477**	**0.3438**	**0.3438**	**0.3421**	**0.3414**	**0.2892**	**0.2900**	**0.2704**	**0.2704**	44

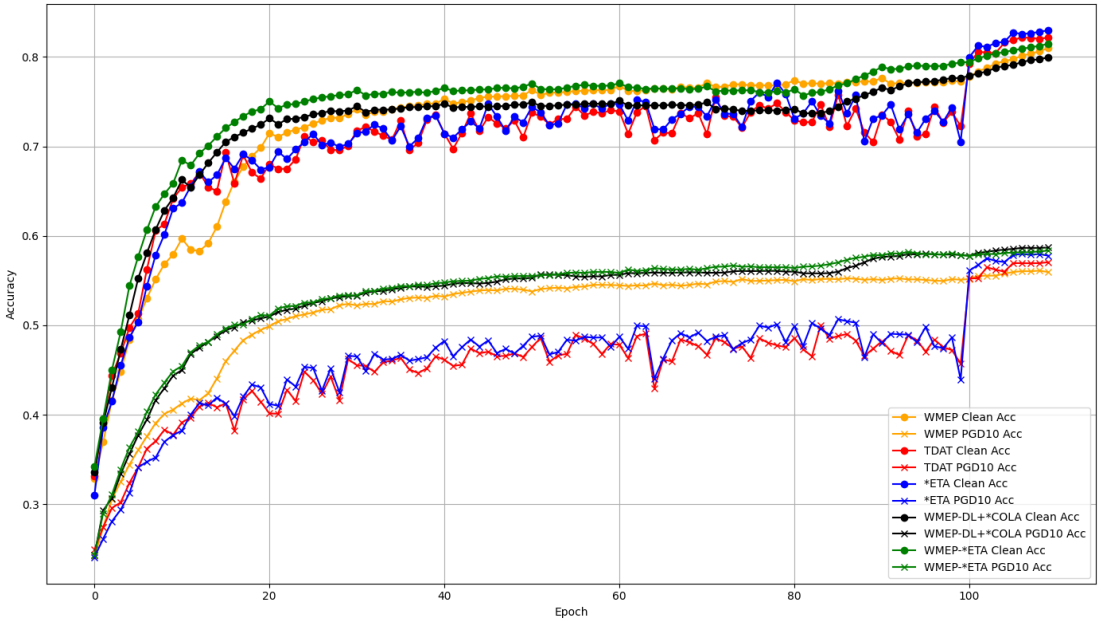

図 3.2: CIFAR10 に対する各防御手法の精度の推移.

CIFAR10 での結果と同じで，WMEP+DL+*COLA が多少クリーン精度は低下してしまうが，最も高いロバスト性を示している．また，WMEP+*ETA は WMEP+DL+*COLA と比較して高いクリーン精度であるが，特に PGD ロバスト精度の部分で多少劣ってしまうという性能を示している．ここでも多少のクリーン精度を犠牲にしてでもロバスト性の向上を目指したい場合は WMEP+DL+*COLA，あまりクリーン精度を犠牲にすることなく高いロバスト性を目指したい場合には WMEP+*ETA が優れていると言える．

図 3.3 に CIFAR100 データセットでの，ベースライン手法（WMEP，TDAT）と提案手法（*ETA，WMEP-DL+*COLA，WMEP-*ETA）のトレーニング中の精度の推移を示した．CIFAR10 での結果と同じで，TDAT と*ETA は不安定な学習曲線を示した一方，提案手法 WMEP 及び WMEP-DL+*COLA，WMEP-*ETA はより安定した学習曲線を示した．WMEP では 100 エポック目に学習率が小さくなるとクリーン精度は大きく上昇した一方，ロバスト精度は低下した．これは学習率が低下したことで，トレーニングデータへの過学習が起こってしまったと考えられる．その他の手法では過学習は見られず，より良い正則化ができていると考えられる．

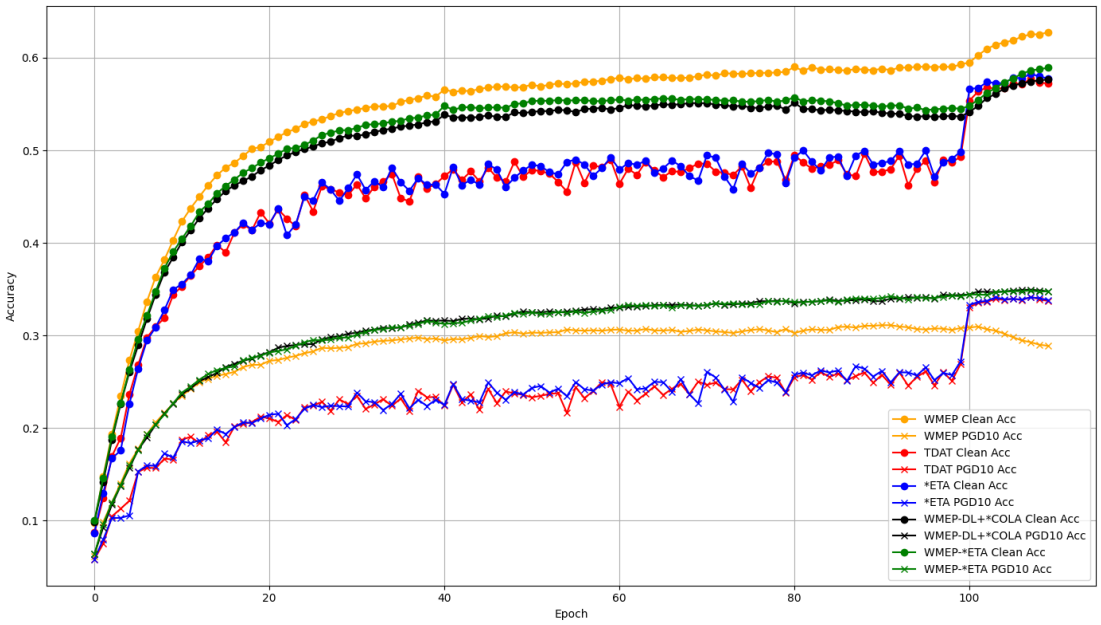

図 3.3: CIFAR100 に対する各防御手法の精度の推移.

3.5　本章のまとめ

本章では，提案手法について述べた後，各データセットごとに行った実験の結果について示し，既存のベースラインとなる手法との比較及び評価を行った．

3.2 節**「提案手法」**では，ベースとなる手法を紹介した後，既存研究への考察を踏まえて提案する新たな Fast Adversarial Training 手法の概要について述べた．

3.3 節**「実験準備」**では，実験を行う環境や，実験で利用するデータセット，訓練時のパラメータ設定，評価に用いる攻撃手法やベースラインとする既存手法について述べた．

3.4 節**「実験結果と評価」**では，CIFAR10 及び CIFAR100 データセットでの実験結果を示し，提案手法とベースラインとの比較，評価を行った．

第4章

結論

本論文では，Deep Learning の概要や応用分野，さらに画像認識の分野で Deep Learning が発展を遂げて現在のように注目されるようになった経緯について述べた．また Deep Learning の性能が上がり，その応用が現実化する中で，セキュリティ面で本論文のテーマである Adversarial Examples に対する防御手法である Fast Adversarial Training を研究することの重要性を示した．次に本論文の提案手法の訓練及び評価を行うために用いる，AEs の生成手法について述べた．さらに FAT の既存手法について述べ，FAT の課題である壊滅的過学習をどのように抑制するか考察を行った．そして，これらの考察を踏まえて，新たに既存手法を組み合わせた FAT を提案し，CIFAR10 及び CIFAR100 の二つのデータセットを用いて，本論文で行った実験の概要と結果を示し，ベースラインとなる手法との比較及び評価を行った．

第 2 章「**Adversarial Examples と Fast Adversarial Training**」では，提案手法の訓練及び評価を行うために用いる主要な AEs 生成手法について概要を述べた後，壊滅的過学習を抑制するための既存の FAT 手法についていくつか述べた．Adversarial Examples とは，人間には知覚しづらい微小な摂動を加えることで分類損失を大きくして，Deep Neural Network の誤分類を誘発するデータであった．この生成手法として，単一段階で損失関数の勾配の符号の方向に摂動を加えることで損失を大きくする FGSM という手法，多段階の最適化により，さらに効果的に損失関数の勾配の符号の方向に摂動を加えることで損失を大きくする PGD という手法，誤分類確率と摂動の大きさのトレードオフを表す目的関数を最適化して敵対的摂動を生成する CW Attack という手法，PGD を改良したパラメーターフリーの APGD-CE や APGD-DLR に，分類境界までの最短距離を狙う FAB Attack，クエリベースで摂動を生成する Square Attack を組み合わせたフレームワークであり，多様な攻撃の総合的評価が可能である AutoAttack という手法について述べた．Adversarial Training とは AEs をトレーニングデータに追加することでモデルが AEs に対してロバストになることを目指す手法であった．標準的な Adversarial Training は AEs の生成に多段階の PGD を用いるが，Fast Adversarial Training は AEs の生成に単一段階の FGSM を用いるため，計算コストを小さく抑えられる．しかしながら，単一段階の FGSM を用いると同じような摂動を生成してしまうため，壊滅的過学習が起こりやすいという課題がある．壊滅的過学習を抑制する手

法としては，摂動初期化，重み平均化，ラベル緩和，分類駆動型損失，収束制約に分けて述べた．まず摂動初期化手法としては，ランダム初期化 FGSM-RS や勾配モメンタムを利用する FGSM-MEP，さらに AEs の品質に応じて動的に勾配モメンタムの重みづけを行う FGSM-WMEP という手法について述べた．また，重み平均化手法として，モデルのロバスト性に応じて重みを更新するかしないかを制御し，FAT で壊滅的過学習を引き起こさないようにしながら、損失ランドスケープを平滑化する動的重み平均化について述べた．次に正解ラベルを緩和するラベル緩和手法として，訓練が進むにつれて次第に強くラベルを緩和していき，トレーニング後半での品質の悪い AEs への過学習を抑制し，ロバスト性を向上させる手法について述べた．さらに分類駆動型損失として，誤分類されているクリーンデータを重点的に学習させる手法や，誤分類された AEs を重点的に学習させる手法など，分類に応じて損失の大きさを調整する手法について述べた．他にもエポック間の損失差を制限することで安定したトレーニングを目指す収束制約という手法についても述べた．これらの既存研究を踏まえて，FAT で壊滅的過学習を抑制しロバスト性を向上させるための手法として，多様で攻撃性能が高い AEs を生成し，モデル性能に応じた学習強度の調整を行うことがモデルのロバスト性の向上につながると考えた．多様で攻撃性能の高い AEs を生成するためには，FGSM を用いるため PGD のように何度も損失関数が大きくなる方向にデータを更新することができないため，適切な摂動初期化が重要であり，FGSM-WMEP が有効であると考えた．またモデル性能に応じた学習強度の調整として，現在のモデル性能に応じてラベル緩和や正則化強度，重み平均化によるモデルパラメータの更新の有無を動的に決定する手法が有効であると考えた．

第 3 章**「パラメータの動的な調整による Fast Adversarial Training の改善」**では，既存手法から得た考察を踏まえて，過去のエポックの勾配モメンタムから初期摂動を生成し，モデルの分類性能に応じて学習強度を調整する本論文の提案手法について述べ，実験条件やデータセット，及び結果と考察について述べた．本論文の提案手法では，FGSM-WMEP と ETA の 2 つの既存手法を組み合わせた．まず既存の FGSM-WMEP は，過去の全エポックからの勾配モメンタムをモデルのロバスト性に応じて重み付けして計算し，そこから初期摂動を生成した．さらに定期的に初期摂動をリセットしてランダム摂動にすることで，特定の摂動パターンへの過学

習も回避する．さらに損失関数には正則化項として MSE を加えることで，クリーンデータと AEs の出力を近づけモデルのロバスト性を高めた．重みの更新は，AEs の品質が高い場合にのみ指数移動平均により行い，AEs の品質が低く壊滅的過学習を引き起こす重みは取り入れないようにした．一方 ETA では，動的ラベル緩和により，トレーニング前半では弱めのラベル緩和で学習速度を速めて安定化させ，トレーニング後半では強めのラベル緩和で品質が低下した AEs への過学習を抑制させた．また分類駆動型損失を導入して，誤分類されているクリーンデータへの学習に集中するように損失を調整した．以上を踏まえて新たに WMEP-ETA という FGSM-WMEP の摂動初期化と動的重み平均化に，ETA の動的ラベル緩和及び分類駆動型損失を統合した FAT 手法を提案した．具体的には各エポックごとにラベル緩和係数を動的に計算し，損失関数には分類損失である Cross-Entropy 損失に加えてクリーンデータと AEs の出力を近づける正則化項を導入し，クリーンデータや AEs の分類結果に応じて損失の大きさを調整する．また，AEs の品質を評価し，品質が良いと判断された場合のみ EMA 重みを更新することで，壊滅的過学習を回避しつつロバスト性の向上を目指す．この提案手法の有効性を評価するために行った実験の概要について説明した．実験を行った実装環境や，使用したデータセットごとに設定したパラメータについてまとめ，再現性を担保した．また比較及び評価を行うために用いる既存の FAT 手法と AEs 生成手法についても述べ，CIFAR10 と CIFAR100 でのデータセットごとに実験の結果を示した．どちらのデータセットでも提案手法である WMEP+DL+*COLA 及び WMEP+*ETA は追加の計算コストなく特に PGD で最も優れたロバスト性を示した．WMEP+DL+*COLA と WMEP+*ETA を比較した場合，WMEP+DL+*COLA の方が AEs に対するロバスト精度は高い一方，WMEP+*ETA の方が高いクリーン精度を持つため，どちらを重要視するかで優劣は変わる．本提案手法では，WMEP により多様で攻撃性能の高い AEs を生成でき，動的ラベル緩和（DL）と分類駆動型損失である TD 損失（TD）または COLA 損失（COLA）を組み合わせることで，動的ラベル緩和で特に訓練後半での学習に悪影響を及ぼす異常値への過学習を防ぎつつ，誤分類されるデータに集中して学習を行うため性能が向上したと考えられた．

一連の結果を踏まえた今後の展望として，まずは解像度の高い ImageNet などのデータセットでの提案手法の有効性の検証があげられる．さらに，CNN では

なく ViT などの Transformer ベースのモデルに対しても提案手法でのトレーニングの有効性を検証したい．また，ラベル緩和ではどのような曲線を描いてラベルを緩和していくのがベストであるのか，さらにはモデルのロバスト性も踏まえてラベル緩和係数を決定する手法は有効であるかも検討したい．モデルのロバスト性を評価する指標として現状クリーン精度に対する AEs 精度の比率を用いていたが，この手法はクリーン精度が上昇して AEs 精度が減少する過学習を検知するには有効であるとは考える一方，WMEP において現在の勾配をどの程度モメンタムに取り入れるかを決定するパラメータとして適切であるのかも検討したい．誤分類されるクリーンデータなどの分類ごとのデータの取り扱い方法についても検討し，誤分類されるクリーンデータを無理に学習しようとする TD は本当に有効であるのかも含めて検討したい．損失範囲の集中化がロバスト性の向上につながるという先行研究から，訓練全体での損失ランドスケープを観察しつつ，損失範囲を集中化する手法も検討したい．さらにラベル緩和や重み平均化などの，もともとは通常のトレーニングや通常の PGD を用いた Adversarial Training で用いられた手法が FAT においても有効であることから，視野を広げてこれらのトレーニングで用いられる手法を FAT に実践した有効性の検討もしていきたい．解像度の高い ImageNet などのデータセットやモデルパラメータの大きい ViT を用いた AT を行うには膨大な時間がかかってしまうが，計算コストの小さい FAT でも十分なロバスト性を確保できることができれば，実用性を高めることができる可能性がある．

謝辞

本論文を執筆するにあたり，貴重な御指導，御助言を賜りました，本学電子物理システム学科，史又華教授に深く感謝申し上げます.

参考文献

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.

[6] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *CoRR*, vol. abs/1312.6199, 2013.

[7] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations*, 2018.

[8] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *CoRR*, vol. abs/1412.6572, 2014.

[9] E. Wong, L. Rice, and J. Z. Kolter, "Fast is better than free: Revisiting adversarial training," in *International Conference on Learning Representations*, 2020.

[10] A. Krizhevsky, "Learning multiple layers of features from tiny images," 2009.

[11] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*, 2017, pp. 39–57.

[12] F. Croce and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," in *International Conference on Machine Learning*, 2020.

[13] F. Croce and M. Hein, "Minimally distorted adversarial examples with a fast adaptive boundary attack," *ArXiv*, vol. abs/1907.02044, 2019.

[14] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein, "Square attack: A query-efficient black-box adversarial attack via random search," in *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII*. Berlin, Heidelberg: Springer-Verlag, 2020, pp. 484–501.

[15] X. Jia, Y. Zhang, X. Wei, B. Wu, K. Ma, J. Wang, and X. Cao, "Prior-guided adversarial initialization for fast adversarial training," *ArXiv*, vol. abs/2207.08859, 2022.

[16] X. Jia, Y. Zhang, X. Wei, B. Wu, K. Ma, J. Wang, and X. Cao, "Improving fast adversarial training with prior-guided knowledge," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 9, pp. 6367–6383, 2024.

[17] X. Jia, Y. Chen, X. Mao, R. Duan, J. Gu, R. Zhang, H. Xue, Y. Liu, and X. Cao, "Revisiting and exploring efficient fast adversarial training via law: Lipschitz regularization and auto weight averaging," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 8125–8139, 2024.

[18] T. Chen, Z. Zhang, S. Liu, S. Chang, and Z. Wang, "Robust overfitting may be mitigated by properly learned smoothening," in *International Conference on Learning Representations*, 2021.

[19] H. Wang and Y. Wang, "Self-ensemble adversarial training for improved robustness," in *International Conference on Learning Representations*, 2022.

[20] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.

[21] K. Tong, C. Jiang, J. Gui, and Y. Cao, "Taxonomy driven fast adversarial training," in *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence*, ser. AAAI'24/IAAI'24/EAAI'24. AAAI Press, 2024.

[22] J. Gui *et al.*, "Improving fast adversarial training paradigm: An example taxonomy perspective," *arXiv:2408.03944*, 2024.

[23] M. Zhao, L. Zhang, Y. Kong, and B. Yin, "Fast adversarial training with smooth convergence," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 4697–4706.

2024年度 卒業論文

パラメータを動的に調整する Fast Adversarial Trainingに関する研究

指導教員　史 又華　教授

早稲田大学 基幹理工学部
電子物理システム学科

1 W212275-2

内藤 舜介
Shunsuke Naito

二〇二四年度　卒業論文　パラメータを動的に調整するFast Adversarial Trainingに関する研究　内藤 舜介

二〇二四年度　卒業論文　パラメータを動的に調整するFast Adversarial Training に関する研究　内藤 舜介

二〇二四年度　卒業論文　パラメータを動的に調整するFast Adversarial Training に関する研究　内藤 舜介

二〇二四年度　卒業論文　パラメータを動的に調整するFast Adversarial Training に関する研究　内藤 舜介

二〇二四年度　卒業論文　パラメータを動的に調整するFast Adversarial Training に関する研究　内藤 舜介

二〇二四年度　卒業論文　パラメータを動的に調整するFast Adversarial Training に関する研究　内藤 舜介

二〇二四年度　卒業論文　パラメータを動的に調整するFast Adversarial Training に関する研究　内藤 舜介