2025年度
中部大学大学院工学研究科ロボット理工学専攻

博士学位論文

知識転移グラフによる
深層共同学習に関する研究

岡本 直樹

論 文 要 旨

知識蒸留とは，モデルが学習で獲得した知識を別のモデルに転移することで，パラメータ数の削減や性能改善を行う技術である．モデルの出力分布を知識とした最も基本的な知識蒸留では，学習済みモデルの出力分布を模倣するように未学習モデルの学習を行う．one-hot ベクトルで表現された正解ラベルと比べて，モデルの出力分布にはモデルが学習で獲得したクラス間の類似度情報（dark knowledge）が含まれており，この情報がモデルの性能改善に寄与していると考えられている．知識蒸留は，パラメータ数の大きい学習済みモデルの知識をパラメータ数の少ない未学習モデルへ伝える一方向の知識転移によってモデル軽量化を実現した．モデルの軽量化を目的として研究が進められた知識蒸留であったが，パラメータ数が同等の学習済みモデルからの知識転移や未学習のモデル間の双方向の知識転移においても，性能改善が可能であることが示されて以降は，性能向上を目的とした研究も行われている．本研究では，このようなモデルからモデルへ知識を転移する学習方法を共同学習と定義する．共同学習の研究は，知識表現の設計，損失関数の設計，モデルの組み合わせ，知識の転移方向といった観点から様々な改善が行われている．

しかし，これらの重要な要素の組み合わせ方は膨大であり，各要素が複雑に絡み合っていることから，シンプルな組み合わせから複雑な組み合わせまでを手探りで探索し，共同学習の手法設計が行われてきた．また，3 つ以上のモデルを用いた共同学習では，各モデル間で異なる知識表現や損失関数を採用する複雑な共同学習が設計可能であるが，従来法では全てのモデル間で同じ設定を採用するシンプルな設計に留まっている．

本研究では，はじめに共同学習における学習戦略の多様化と効果的な手法の設計に焦点を当て，従来法を含んだ多様な共同学習を表現可能な知識転移グラフを提案した．知識転移グラフでは，モデルをノード，損失計算と知識転移の関係をエッジで表現し，共同学習の従来法に統一的な視点を提供する．また，損失値に重み付けを行うゲート関数を損失関数に導入することで，多様な知識転移戦略を表現可能とする．そして，効果的なグラフ構造を人手により設計するのではなく，ハイパーパラメータ探索によって大規模かつ複雑な共同学習の自動設計を実現した．次に，1 つのモデルの高精度を目指す従来の共同学習における目的を，共同学習を行う複数モデルからなるグループ単位の改善に拡張し，アンサンブル学習のための共同学習を提案した．また，知識転移グラフをアンサンブル学習に拡張し，ハイパーパラメータ探索によって各モデルが異なる注目領域を持つことを促すアンサンブル学習に適した共同学習を獲得した．最後に，ラベルなしデータの活用を目的として，半教師あり学習の代表的な従来法をグラフ表現に落とし込み，半教師あり共同学習を提案した．半教師あり学習の代表的な従来法を統一的なグラフで表現し，一致性正則化と擬似ラベリングの組み合わせ，教師あり学習・半教師あり学習・教師なし学習の連携，共同学習への拡張を含んだ多様な探索空間を設計した．この探索空間における知識転移グラフのハイパーパラメータ探索によって，半教師あり学習における新たな知見を獲得した．また，獲得した知見に基づいた手法の手動設計を行い，知見の有効性を確認した．これらの大きく 3 つの取り組みによって共同学習は，知識転移戦略の多様化，個人レベルからグループレベルの精度改善，異なる学習方法の連携が可能となり，共同学習の枠組みを大きく拡張した．

目 次

図 目 次

表目次

第1章

序論

本章では，本研究の背景及び目的，本論文の構成について述べる．

1.1 研究の背景

画像分野の深層学習モデルは，クラス分類 [10, 11] や物体検出 [12, 13]，セマンティックセグメンテーション [14, 15] などの幅広い問題設定で高い性能を達成している．各問題設定において，モデル構造の発展やデータセットの大規模化に伴って，モデルのパラメータ数は増加傾向である．そのため，性能が高くなると同時に計算コストやメモリコストも増加している．スマートフォンなどのエッジデバイスでは，計算資源が限られていることから，このような大規模かつ高コストなモデルを動かすことは困難であり，社会実装の障壁の一つとなっている．このような背景から知識蒸留をはじめとするモデルの軽量化技術に注目が集まっている．

知識蒸留とは，モデルが学習で獲得した知識を別のモデルに転移することで，パラメータ数の削減や性能改善を行う技術である．モデルの出力分布を知識とした最も基本的な知識蒸留 [16] では，学習済みモデルの出力分布を模倣するように未学習モデルの学習を行う．one-hot ベクトルで表現された正解ラベルと比べて，モデルの出力分布にはモデルが学習で獲得したクラス間の類似度情報（dark knowledge）が含まれており，この情報がモデルの性能改善に寄与していると考えられている [16, 17]. 知識蒸留は，パラメータ数の大きい学習済みモデルの知識をパラメータ数の少ない未学習モデルへ伝える一方向の知識転移によってモデル軽量化を実現した．モデルの軽量化を目的として研究が進められた知識蒸留であったが，パラメータ数が同等の学習済みモデルからの知識転移 [5] や未学習のモデル間における双方向の知識転移 [9] においても，性能改善が可能であることが示されて以降は，性能向上を目的とした研究も行われている．本研究では，このようなモデルからモデルへ知識を転移する学習方法を共同学習と定義する．共同学習の研究は，知識表現の設計，損失関数の設計，モデルの組み合わせ，知識の転移方向といった観点から様々な改善が行われている．

しかし，これらの重要な要素の組み合わせ方は膨大であり，各要素が複雑に絡み合っていることから，従来法では限られた組み合わせの有効性しか明らかになっていない．例えば，モデルの組み合わせと知識の転移方向の観点では，パラメータ数が大きいモデルから小さいモデルへの一方向の知識転移 [16]，パラメータ数が同等の 2 つのモデルにおける一方向の知識転移 [5]，パラメータ数が同等の 2 つのモデルにおける双方向の知識転移 [9]，パラメータ数が異なる 3 つのモデルにおける段階的な一方向の知識転移 [18] といったように，シンプルな関係性から複雑な関係性までを手探りで探索し，設計が行われてきた．この共同学習の手法設計は，学習に使用するモデル数が増えるほど，多様で複雑な設計ができるようになるが，手探りで効果的な設計をすることが困難となる．そのため，3 つ以上のモデルを用いた共同学習では，各モデル間で異なる知識表現や損失関数を採用する複雑な共同学習が設計可能であるが，従来法では全てのモデル間で同じ設定を採用するシンプルな設計に留まっている．

1.2 研究目的

本研究では，以下に示す 3 つの項目について取り組む．

1. 知識転移グラフによる深層共同学習
2. 多様な知識転移によるアンサンブル学習
3. 一致性正則化と擬似ラベリングによる半教師あり共同学習

1 つ目は，共同学習の代表的な従来法をグラフ表現に落とし込み，グラフ構造の組み合わせ問題としてハイパーパラメータ探索を行うことで，複雑な関係性を含む多様な共同学習の中から効果的な学習方法を発見することを目指す．2 つ目は，1 つのモデルの高精度を目指す従来の共同学習における目的を，共同学習を行う複数モデルからなるグループ単位の改善に拡張する．3 つ目は，半教師あり学習の代表的な従来法をグラフ表現に落とし込み，共同学習のグラフ表現と組み合わせることで，半教師あり共同学習に拡張する．半教師あり共同学習に拡張したグラフ構造の探索を通して，教師あり学習・半教師あり学習・教師なし学習の連携を考慮した共同学習の設計を目指す．

以下に各項目における本研究の目的を示す．

知識転移グラフによる深層共同学習

共同学習は，知識表現の設計，損失関数の設計，モデルの組み合わせ，知識の転移方向の 4 つの重要な要素について，それぞれの要素や各要素の組み合わせ方について，人手で改善することで発展してきた．しかし，これらの重要な要素の組み合わせ方は膨大であり，各要素が複雑に絡み合っていることから，従来法では限られた組み合わせの有効性しか明らかになっていない．そこで本研究では，共同学習の代表的な従来法をグラフ表現に落とし込み，従来法を含んだ多様な共同学習を表現可能な知識転移グラフを提案する．知識転移グラフでは，モデルをノード，損失関数を有向エッジ，知識転移の方向を有向エッジの向きで表現する．また，損失値に重み付けを行うゲート関数を損失関数に導入し，知識転移の勾配を制御する．ゲート関数として，目的の異なる 5 つのゲート関数を設計する．このグラフ表現において，ノードで使用するモデルの種類やエッジで使用するゲート関数の種類を任意のものに設定することで，従来法や従来法を組み合わせ手法などの多様な共同学習を容易に設計することができる．最終的なグラフ構造は，人手によって手探りで設計のするのではなく，ノードで使用するモデルとエッジで使用するゲート関数を知識転移グラフのハイパーパラメータとして，ハイパーパラメータ探索によって決定する．これにより，これまで考慮されなかった複雑な関係性を含む多様な共同学習の中から効果的な学習方法を発見することを目指す．

多様な知識転移によるアンサンブル学習

共同学習は，任意の複数モデルを使用し，ある 1 つのモデルの性能改善を目指す．任意のモデルとして，パラメータ数の大きい学習済みモデルを使用する場合はモデルの軽量化，パラメータ数が同等の学習済みモデルや未学習モデルを使用する場合はモデルの高精度化を研究の目的としていることが多い．ある 1 つのモデルの性能に着目して学習を行うが，最終的に複数モデルを獲得できることから，推論時にこれらの複数モデルをアンサンブル学習によって統合することで，更なる性能改善が可能であることが示されている [5, 9]．しかし，アンサンブル学習による性能改善は，複数モデルを獲得することによる副次的な効果であり，共同学習とアンサンブル学習の関係は，明らかになっていない．そこで本研究では，ある 1 つのモデルの性能改善ではなく，共同学習を行う複数モデルからなるグループ単位の改善を目的として，共同学習の設計を行う．まず初めに，共同学習の従来法と

アンサンブル学習の関係を分析する．次に，分析の傾向に基づいてアンサンブル学習のための多様な共同学習について検討する．そして，知識転移グラフをグループ単位の学習に拡張することでアンサンブル学習に適用し，グラフ構造のハイパーパラメータ探索によって多様な共同学習を用いたアンサンブル学習を自動設計する．最後に，自動設計した学習方法により得られた多様性のある複数モデルの知識を 1 つのモデルへ転移することで，アンサンブル学習による高い性能を維持しつつ，アンサンブル学習の推論コストを削減する．

一致性正則化と擬似ラベリングによる半教師あり共同学習

共同学習は，ラベルありデータを用いた教師あり学習の設定において，様々な手法の検討が行われている．ラベルなしデータを活用した学習方法として半教師あり学習がある．半教師あり学習は，ラベルありデータとラベルなしデータの両方を用いてモデルを学習する．半教師あり学習の代表的な手法である FixMatch [19] は，ラベルなしデータに擬似ラベルを付与して学習を行う擬似ラベリングと，異なる摂動を付与した同一データの出力に一貫性を持たせるように学習する一致性正則化を組み合わせた手法である．FixMatch の登場以降，FixMatch を基盤技術とした派生手法が数多く提案されている．しかし，FixMatch における一致性正則化と擬似ラベリングの組み合わせ方は手動で設計されており，組み合わせ方に関する包括的な調査はまだ行われていない．そこで本研究では，半教師あり学習における一致性正則化と擬似ラベリングの多様な組み合わせ方の検討と共同学習におけるラベルなしデータの活用を目的として，半教師あり学習のための知識転移グラフを設計する．半教師あり学習の代表的な従来法を主要な要素に分解し，知識転移グラフで表現することで，グラフ構造の探索を通して一致性正則化と擬似ラベリングの多様な組み合わせ方を検討する．また，グラフ構造の拡張性から，従来の半教師あり学習を半教師あり共同学習へ容易に拡張可能となる．半教師あり学習を組み込んだ知識転移グラフは，教師あり学習，半教師あり学習，教師なし学習の 3 種類の学習方法を表現することができる．これらを内包した多様な探索空間におけるグラフ構造の探索によって，一致性正則化と擬似ラベリングの組み合わせ，3 種類の学習方法の連携，共同学習への拡張を考慮した学習方法の自動設計を行い，自動設計された学習方法を分析することで，新たな知見の獲得を目指す．

1.3 本論文の構成

本論文は，図 1.1 に示すように 6 つの章で構成されている．1 章では，本研究の背景と目的を述べた．2 章では，知識蒸留の概要について述べ，知識蒸留の進展と代表的な枠組みについて述べる．その後，各枠組みにおける代表的な従来法について述べる．3 章では，知識蒸留を共同学習として定義し，知識転移グラフとグラフ構造の探索について述べる．4 章では，関連研究である深層学習におけるアンサンブル学習について述べ，共同学習とアンサンブル学習の分析と知識転移グラフのグループ単位の学習への拡張について述べる．5 章では，関連研究である半教師あり学習について述べ，半教師あり学習のグラフ表現と半教師あり共同学習への拡張について述べる．6 章では，本論文の結論と展望について述べる．

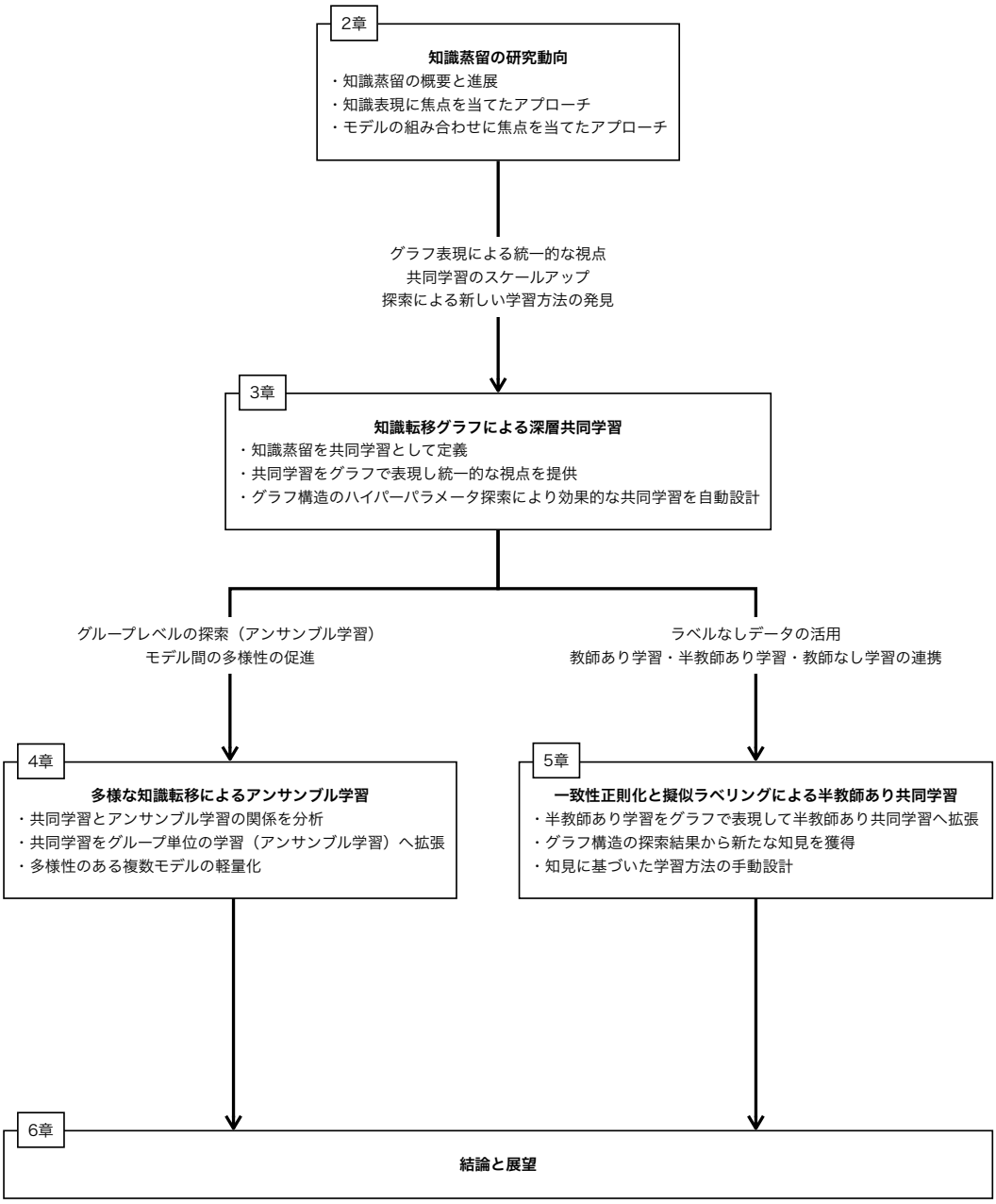

2章
知識蒸留の研究動向
・知識蒸留の概要と進展
・知識表現に焦点を当てたアプローチ
・モデルの組み合わせに焦点を当てたアプローチ
グラフ表現による統一的な視点
共同学習のスケールアップ
探索による新しい学習方法の発見
3章
知識転移グラフによる深層共同学習
・知識蒸留を共同学習として定義
・共同学習をグラフで表現し統一的な視点を提供
・グラフ構造のハイパーパラメータ探索により効果的な共同学習を自動設計
グループレベルの探索（アンサンブル学習）
モデル間の多様性の促進
ラベルなしデータの活用
教師あり学習・半教師あり学習・教師なし学習の連携
4章
多様な知識転移によるアンサンブル学習
・共同学習とアンサンブル学習の関係を分析
・共同学習をグループ単位の学習（アンサンブル学習）へ拡張
・多様性のある複数モデルの軽量化
5章
一致性正則化と擬似ラベリングによる半教師あり共同学習
・半教師あり学習をグラフで表現して半教師あり共同学習へ拡張
・グラフ構造の探索結果から新たな知見を獲得
・知見に基づいた学習方法の手動設計
6章
結論と展望

図 1.1: 本論文の構成

第2章

知識蒸留の研究動向

知識蒸留とは，パラメータ数の多い学習済みモデルの知識をパラメータ数の少ないモデルに転移することで，パラメータ数の多いモデルと同等の性能を発揮するパラメータ数の少ないモデルを作成し，モデルの軽量化を行う技術である．

本章の構成は，以下のとおりである．2.1 節では，知識蒸留の概要と知識蒸留の進展について述べる．2.2 節では，知識表現に焦点を当てたアプローチについて述べる．2.3 節では，モデルの組み合わせに焦点を当てたアプローチについて述べる．

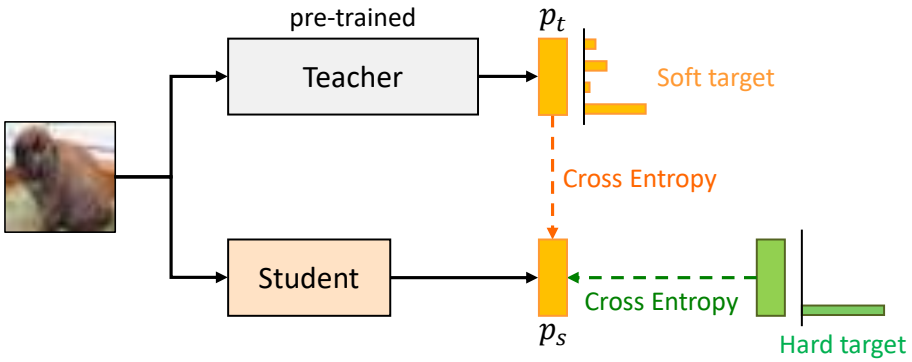


図 2.1: Knowledge distillation における学習方法

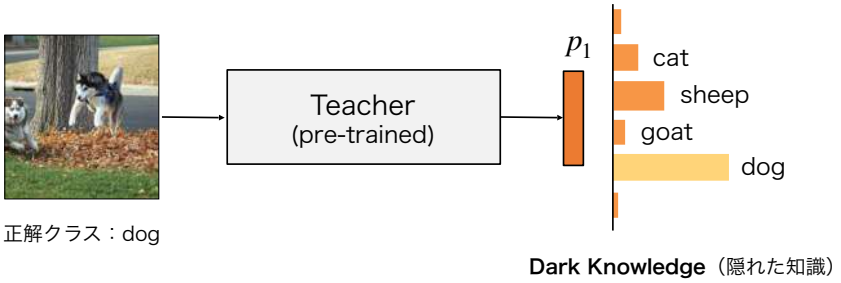


図 2.2: Soft target の効果

2.1 知識蒸留

本節では，はじめに知識蒸留の基盤技術である Knowledge Distillation (KD) [16] について述べ，その後に KD 以降の知識蒸留の進展について述べる．

2.1.1 Knowledge Distillation

Knowledge Distillation (KD) [16] は，クラス分類タスクを学習したモデルの軽量化を目的とした手法であり，パラメータ数の多い学習済みモデルの出力分布を知識とし，人手で作成した正解ラベルと学習済みモデルの知識を用いてパラメータ数の少ない未学習モデルを学習する．KD の概要を図 2.1 に示す．KD では，学習済みモデルを教師モデル (Teacher)，教師モデルの知識を用いて学習するモデルを生徒モデル (Student)，人手で作成した正解ラベルを Hard target，教師モデルの出力分布を Soft target と呼ぶ．KD は，教師モデルと生徒モデルに同一データを入力し，生徒モデルの出力分布を正解ラベルに近づける学習と，生徒モデルの出力分布を教師モデルの出力分布に近づける学習の 2 つを同時に行い，生徒モデルを学習する．このとき，教師モデルはパラメータを固定化し，パラメータの更新を行わない．また，生徒モデルの学習に使用するデータセットは，教師モデルの学習に使用したデータセットを使用する．Soft target の不正解クラスの確率値は，図 2.2 に示すような正解クラスとの相関性を示していることが知られており，この正解クラスとの相関関係 (Dark Knowledge) を

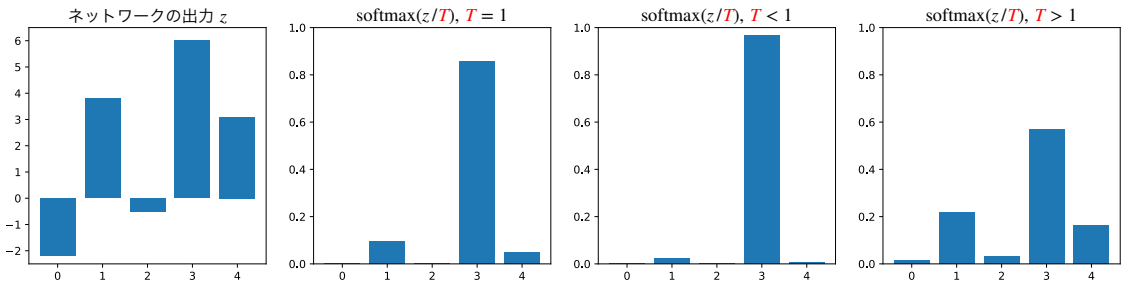


図 2.3: 温度パラメータによる確率分布の変化

学習に利用することで，生徒モデルは Hard target のみを用いて学習した場合と比べて高い精度を発揮すると考えられている [16, 17].

生徒モデルと教師モデルの出力分布は，各モデルの最終出力である logit に対して Softmax 関数を適用することで獲得する．Softmax 関数を式 (2.1) に示す.

$$\mathrm{softmax}(z_i) = \frac{\exp(z_i)}{\sum_j^C \exp(z_j)} \tag{2.1}$$

ここで，z_i は logit におけるクラス i に対するスコア，C はクラス数である．Softmax 関数は，指数関数と総和で割る正規化の性質から logit の中で最も大きいクラススコアを 1 に近い値に変換し，それ以外のクラススコアを 0 に近い値に変換する．そのため，Softmax 関数の出力を Soft target として使用すると，不正解クラスの確率値が非常に小さく抑えられ，教師モデルの知識が生徒モデルに十分伝わらない可能性がある．そこで，KD は Softmax 関数に温度パラメータを導入し，温度パラメータを調整することで，各クラスの確率値の大小関係はそのままに確率値の大きさを調整した．温度付き Softmax 関数を式 (2.2) に示す.

$$\mathrm{softmax}(z_i/T) = \frac{\exp(z_i/T)}{\sum_j^C \exp(z_j/T)} \tag{2.2}$$

ここで，T は温度パラメータである．温度パラメータによる確率分布の変化を図 2.3 に示す．温度パラメータが 1 の確率分布は，通常のソフトマックス関数による確率分布と同様になる．一方で，温度パラメータを 1 未満の値にすることで logit 内の大きなクラススコアをより強調することができ，温度パラメータを 1 を超える値にすることで logit 内の小さなクラススコアを強調することができる．KD では，温度パラメータを 1 を超える値に設定することで，Dark Knowledge を強調し，教師モデルの知識を生徒モデルへ伝達しやすくした.

KD における生徒モデルの損失関数は，Hard target との交差エントロピー損失 L_{label} と Soft target との交差エントロピー損失 L_{kd} の 2 つの項から構成される．生徒モデルの損失関数を式 (2.3) に示す.

$$L_{\mathrm{total}} = (1-\alpha)L_{\mathrm{label}} + \alpha T^2 L_{\mathrm{kd}} \tag{2.3}$$

$$L_{\mathrm{label}} = -\sum_i^C y_i \log \mathrm{softmax}(z_i^{\mathrm{s}}) \tag{2.4}$$

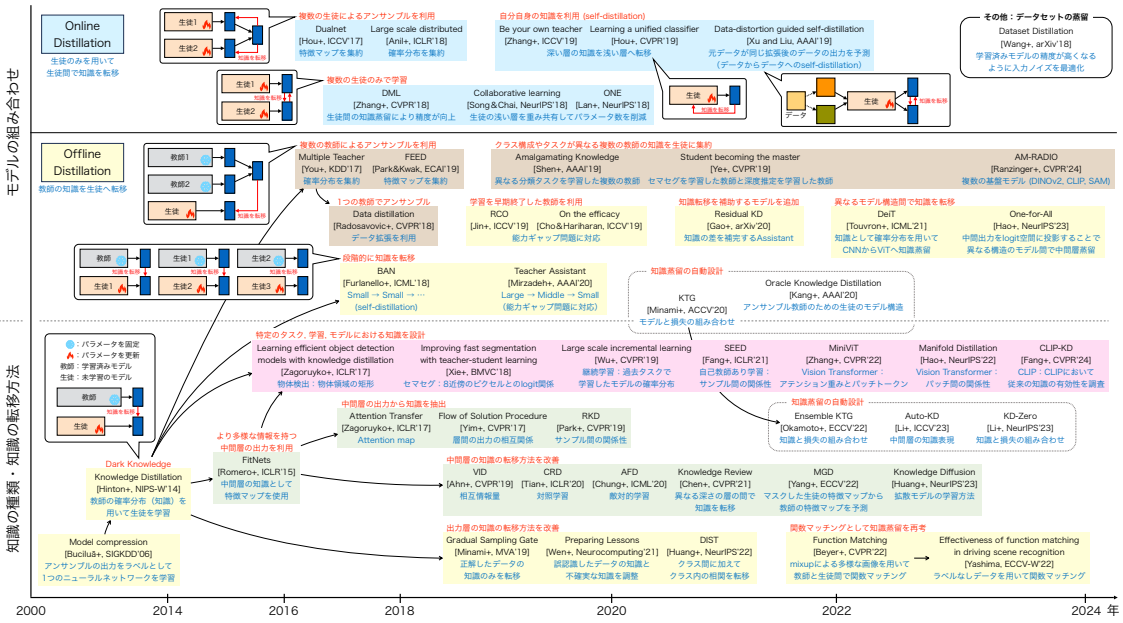


図 2.4: 知識蒸留の進展

$$L_{\mathrm{kd}} = -\sum_{i}^{C} \mathrm{softmax}(z_i^{\mathrm{t}}/T) \log \mathrm{softmax}(z_i^{\mathrm{s}}/T) \tag{2.5}$$

ここで，α は 2 つの損失のバランスを調整する係数，y_i は正解ラベルにおけるクラス i の確率値，z_i^{s} は生徒モデルの logit におけるクラス i のスコア，z_i^{t} は教師モデルの logit におけるクラス i のスコアである．Hard target との交差エントロピー損失では，従来のソフトマックス関数を適用した確率分布が使用され，Soft target との交差エントロピー損失では，温度付きソフトマックス関数を適用した確率分布が使用される．温度付き Softmax 関数による確率分布を用いた交差エントロピー損失の勾配は，温度パラメータ T によって勾配の大きさが $1/T^2$ 倍される．そこで，損失値に対して T^2 を乗算することで，誤差逆伝播時の L_{hard} と L_{soft} の寄与率を等しくしている．

2.1.2 知識蒸留の進展

KD によるモデル軽量化の有効性が示されて以降，図 2.4 に示すように KD を起点として様々な派生手法が提案された．蒸留手法の代表的なアプローチを図 2.5 に示す．代表的なアプローチとして，知識表現に焦点を当てたアプローチとモデルの組み合わせに焦点を当てたアプローチがある．これらのアプローチは，教師モデルが獲得した知識をより効果的に生徒モデルへ転移することを目的としており，生徒モデルの性能を指標としてクラス分類タスクにおいて評価が行われる．これらのアプローチの詳細は，それぞれ 2.2 節と 2.3 節で述べる．一方で，クラス分類タスクでは ResNet [10] や Vision Transformer (ViT) [11]，物体検出タスクでは Faster R-CNN [12] や FCOS [13]，セマンティックセグメンテーションタスクでは U-Net [14] や DeepLabv3+ [15] といったように，問題設定に応じて

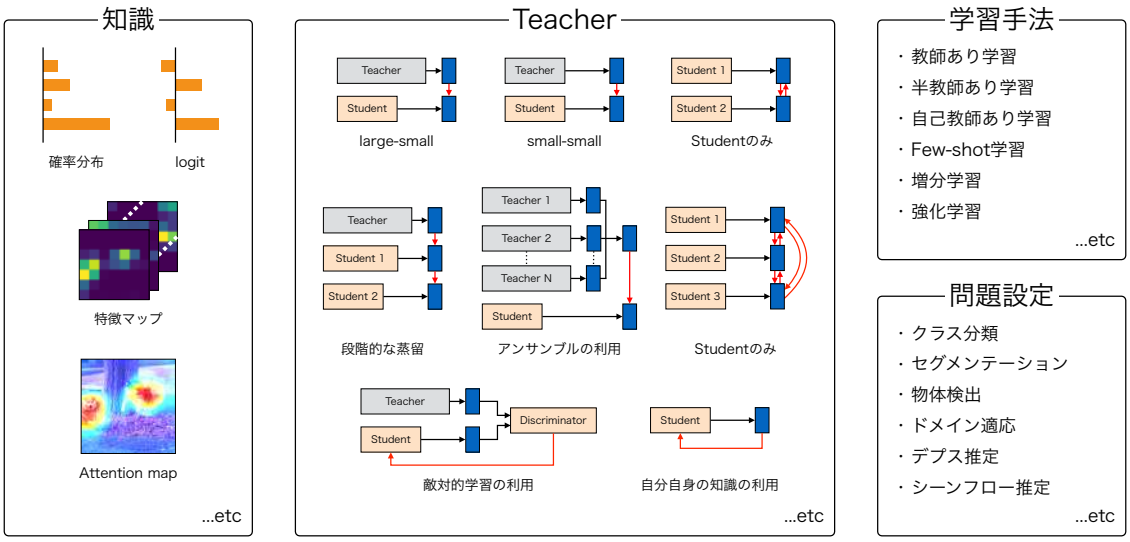


図 2.5: 蒸留手法のアプローチ

モデル構造の設計や発展が行われており，問題設定や学習方法，モデル構造によってモデルが獲得する知識が異なる可能性がある．そこで，特定の設定に特化した知識蒸留の設計が行われている．特定の問題設定に焦点を当てたアプローチとして物体検出タスクに焦点を当てた手法 [20, 21, 22, 23] やセマンティックセグメンテーションタスクに焦点を当てた手法 [24, 25, 26, 27]，特定の学習方法に焦点を当てたアプローチとして半教師あり学習に焦点を当てた手法 [28, 29, 30, 31, 32, 33] や自己教師あり学習に焦点を当てた手法 [34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44]，特定のモデル構造に焦点を当てたアプローチとして ViT に焦点を当てた手法 [45, 46] などが提案されている．また，モデルからモデルへの知識転移という関係をデータからデータへの知識転移という関係に置き換えることで，データセットの軽量化を行うデータセット蒸留 [47] も提案されている．

2.2 知識表現に焦点を当てたアプローチ

KD において知識として使用された確率分布は，モデルの最終出力であり，教師モデルが学習で捉えたクラス間の関係を表現している．しかし，確率分布による知識表現は，教師モデルが獲得した知識の 1 つの側面しか表現しておらず，教師モデルの内部に潜む豊富な情報を十分に活用できないという課題がある．そこで，確率分布以外の知識を活用し，教師モデルの内部表現そのものを生徒モデルに転移させる手法が数多く提案されている．本節では，はじめに出力層における知識表現について述べ，その後に中間層における知識表現について述べる．

2.2.1 出力層における知識

出力層における知識表現は，モデルの最終出力を利用することから，教師モデルが学習した問題設定に応じて設計が行われている．本項では，クラス分類タスク，物体検出タスク，セマンティックセグメンテーションタスクなど多くの問題設定で共通して出力される確率分布を用いた知識表現に焦点を当て，クラス間の関係を表現した知識とクラス内の関係を表現した知識について述べる．

■ クラス間の関係

知識表現 クラス間の関係を表現した知識として，確率分布 [16] と logit [48] が提案されている．2.1.1 項で述べたように確率分布は，logit に Softmax 関数を適用することで獲得するため，Dark Knowledge が非常に小さく抑えられ，教師モデルの知識が生徒モデルに十分伝わらない可能性がある．そこで KD では，温度付き Softmax 関数を導入し，温度パラメータを適切な値に調整することで，Dark Knowledge を強調して効果的な知識転移を可能とした．一方で logit を用いた知識蒸留は，Softmax 関数を適用する前の logit を知識として直接使用することで，Softmax 関数における問題に対応した．

損失関数 確率分布を知識とした蒸留損失は，KD において交差エントロピー損失，DML [9] において Kullback-Leibler (KL) Divergence，Knowledge Distillation from A Stronger Teacher (DIST) [1] においてピアソン相関係数が使用されている．交差エントロピー損失と KL Divergence は，2 つの分布間の相違度を表す指標であり，指標の値を最小化するように学習することで，各クラスのスコアを模倣する．KL Divergence を用いた蒸留損失を式 2.6 に示す．

$$L_{\mathrm{kd}} = \mathrm{KL}(\boldsymbol{p}^{\mathrm{t}} \parallel \boldsymbol{p}^{\mathrm{s}}) \tag{2.6}$$

$$\mathrm{KL}(\boldsymbol{p}^{\mathrm{t}} \parallel \boldsymbol{p}^{\mathrm{s}}) = \sum_{i}^{C} p_i^{\mathrm{t}} \log \frac{p_i^{\mathrm{t}}}{p_i^{\mathrm{s}}} \tag{2.7}$$

ここで，C はクラス数，$\boldsymbol{p}^{\mathrm{t}}$ は教師モデルが出力した確率分布，$\boldsymbol{p}^{\mathrm{s}}$ は生徒モデルが出力した確率分布である．KL Divergence は距離の公理を満たす指標ではないため，$\mathrm{KL}(\boldsymbol{p}^{\mathrm{t}} \parallel \boldsymbol{p}^{\mathrm{s}})$ と $\mathrm{KL}(\boldsymbol{p}^{\mathrm{s}} \parallel \boldsymbol{p}^{\mathrm{t}})$ は通常異なる値となる．一方でピアソン相関係数は，2 つのデータ間の線形関係の強さと方向を -1 から $+1$ の範囲の値で定量化する指標であり，値が大きいほど正の線形関係，値が小さいほど負の線形関係にあることを意味する．ピアソン相関係数を用いた知識蒸留では，ピアソン相関係数を最大化するように学習することで，クラス間の相関関係のみを模倣する．これにより，教師モデルと生徒モデル間の出力分布のスケールやオフセットの差異が学習に及ぼす影響を緩和できる．ピアソン相関係数を用いた蒸留損失を式 2.8 に示す．

$$L_{\mathrm{kd}} = 1 - \rho_p(\boldsymbol{p}^{\mathrm{t}}, \boldsymbol{p}^{\mathrm{s}}) \tag{2.8}$$

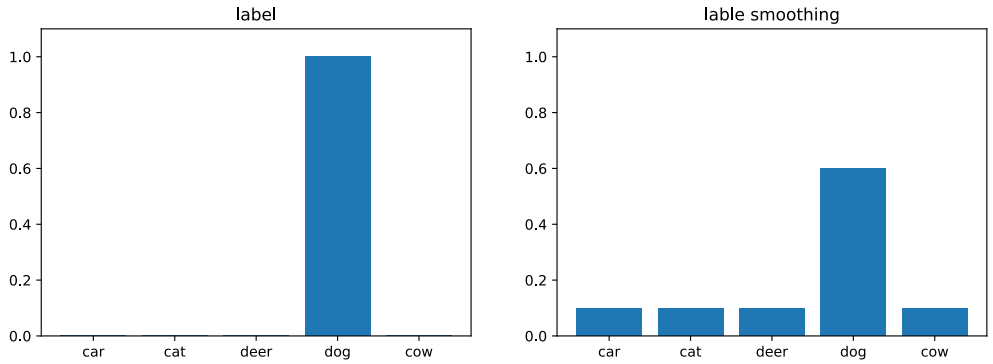


図 2.6: Label Smoothing の適用例

$$\rho_p(\boldsymbol{p}^{\mathrm{t}}, \boldsymbol{p}^{\mathrm{s}}) := \frac{\mathrm{Cov}(\boldsymbol{p}^{\mathrm{t}}, \boldsymbol{p}^{\mathrm{s}})}{\mathrm{Std}(\boldsymbol{p}^{\mathrm{t}})\,\mathrm{Std}(\boldsymbol{p}^{\mathrm{s}})} = \frac{\sum_{i=1}^{C}(p_i^{\mathrm{t}} - \bar{p}^{\mathrm{t}})(p_i^{\mathrm{s}} - \bar{p}^{\mathrm{s}})}{\sqrt{\sum_{i=1}^{C}(p_i^{\mathrm{t}} - \bar{p}^{\mathrm{t}})^2 \sum_{i=1}^{C}(p_i^{\mathrm{s}} - \bar{p}^{\mathrm{s}})^2}} \tag{2.9}$$

ここで，$\mathrm{Cov}(\cdot,\cdot)$ は 2 つの確率分布の共分散，$\bar{p}$ は確率分布の平均値，$\mathrm{Std}(\cdot)$ は確率分布の標準偏差である．

知識の転移戦略 KD は，ミニバッチ内のデータごとに蒸留損失を計算し，ミニバッチ内の全てのデータに対する知識を生徒モデルへ伝達する．しかし，教師モデルは必ずしも全てのデータに対して正解するとは限らず，教師モデルの持つ誤った知識や不確かな知識が生徒モデルに転移されると，学習が不安定になる可能性がある．こうした問題に対して，知識の選別や調整を行う手法が提案されている．Gradual Sampling Gate [49] は，教師モデルの精度に基づいた損失値への重み付けや正解したデータのみを用いた知識の転移を行い，誤った知識の転移を抑制した．Preparing Lessons [50] は，誤った知識や不確かな知識に直接手を加える 2 つのアプローチにより，生徒に悪影響を与える知識の転移を抑制した．1 つ目は，誤認識したデータに対して，そのデータに対する確率分布の代わりに，Label Smoothing [51] を適用した正解ラベルを知識として使用し，誤った知識の転移を抑制する．Label Smoothing は，図 2.6 に示すように正解クラスの確率値を一定の割合だけ減算し，減算した分の確率値を正解クラス以外に均等に割り当てる．誤った確率分布を one-hot ラベルに置き換える場合と比べ，学習によって小さな確率値を保持することが促されるため，知識転移の安定化が期待できる．2 つ目は，過度に Dark Knowledge が強調された不確実性の高い確率分布に対して，データごとに温度パラメータを動的に変更する戦略を採用することで，不確かな知識の転移を抑制する．これらの手法は，教師モデルの持つ全ての知識をそのまま転移するのではなく，学習に有効だと考えられる知識のみを転移する戦略だと言える．一方で，知識蒸留を教師モデルと生徒モデルの出力空間において，同一データの位置をマッチングして整列させる関数マッチングと捉え，教師モデルの logit 空間全体を転移する蒸留戦略も提案されている．logit を用いた知識蒸留は，1 データごとのマッチングのため，logit 空間全体を転移するには，logit 空間内の様々な場所でマッチングを行う必要があ

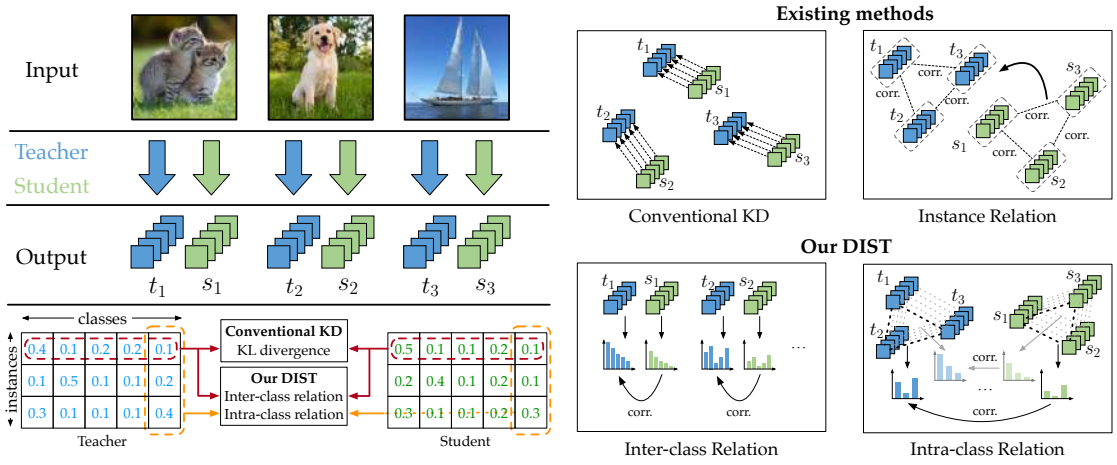


図 2.7: DIST における知識転移（文献 [1] より引用）

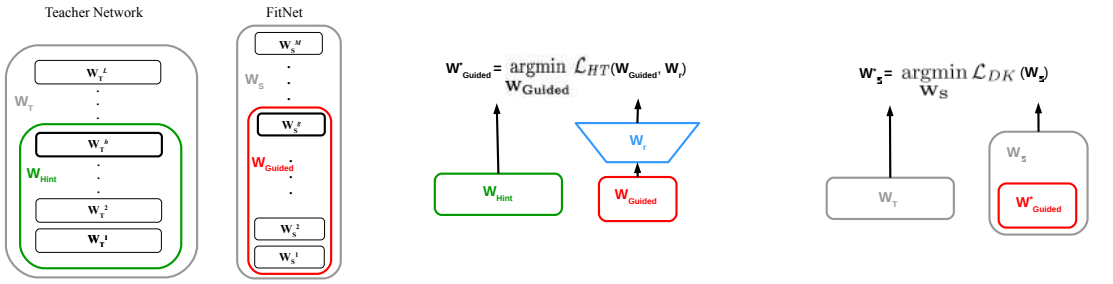
る．そこで，Function Matching [52, 53] は，mixup [54] によるデータ拡張やラベルなしデータの活用によって大量のデータを用意し，関数マッチングを実現した．

■ クラス内の関係

同じクラスに属するデータに着目すると，出力されるクラススコアや確率値にばらつきが存在する．このばらつきは，教師モデルが「どのデータがそのクラスにより適しているか」という知識を持っていることを示している．クラス内の関係を知識とした手法として， DIST が挙げられる．DIST は，クラス間の関係とクラス内の関係を知識として転移する手法である．DIST の概要を図 2.7 に示す．クラス内の関係は，同じクラスに属するデータに対する正解クラスの確率値を，データ数と同じ次元数を持つベクトルの形で集約することで，知識として表現する．DIST では，クラス内の関係を知識とした蒸留損失としてピアソン相関係数を使用する．

2.2.2 中間層における知識

代表的な中間層の知識として，特徴マップ [2] とアテンションマップ [3, 55, 56] が挙げられる．異なるモデルの特徴マップは，異なる特徴空間に写像されているため，直接比較することができない．そこで Romero ら [2] は，生徒モデルの特徴マップを教師モデルの特徴空間に写像する Regressor と呼ばれる層を生徒モデルへ追加することで，特徴マップを用いた知識蒸留を可能とした．アテンションマップは，モデルの注視領域をヒートマップで表現することで，モデルの判断根拠を可視化したものである．アテンションマップを知識として用いることで，Regressor のような写像を行う層が不要となる．また，モデルは複数の中間層を通して推論が行われることから，各層の出力や出力関係を知識として利用する手法 [4, 57, 58, 59, 60] が提案されている．本項では，各知識表現における代表的な手法について述べる．



(a) Teacher and Student Networks (b) Hints Training (c) Knowledge Distillation

図 2.8: FitNets における知識転移（文献 [2] より引用）

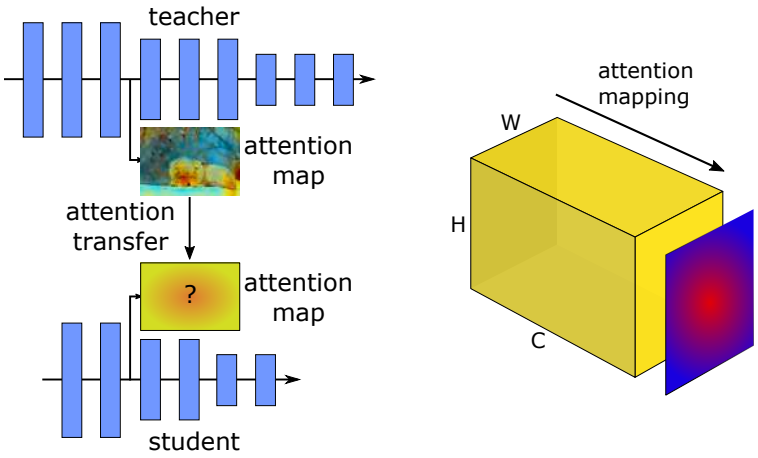


図 2.9: Attention Transfer における知識転移（文献 [3] より引用）

■ 特徴マップ

特徴マップを用いた代表的な手法として，FitNets [2] が挙げられる．FitNets の概要を図 2.8 に示す．FitNets は，特徴マップを転移するステップと確率分布を転移するステップの 2 段階の学習により，生徒モデルを学習する．異なるモデルの特徴マップは，異なる特徴空間に写像されているため，直接比較することができない．また，教師モデルと生徒モデルは，層の数など異なるモデル構造を採用するため，特徴マップのサイズが異なり，直接比較することができない．そこで FitNets では，生徒モデルの特徴マップを教師モデルの特徴マップのサイズに合わせ，教師モデルの特徴空間に写像する Regressor と呼ばれる層を生徒モデルへ追加することで，特徴マップを用いた知識蒸留を可能とした．FitNets における特徴マップを知識とした蒸留損失を式 2.10 に示す．

$$L_{\mathrm{kd}} = \frac{1}{2}\|u_h(\boldsymbol{x}) - r(u_g(\boldsymbol{x}))\|^2 \tag{2.10}$$

ここで，$\boldsymbol{x}$ は入力データ，$u_h(\cdot)$ は教師モデルの特徴マップ，$u_g(\cdot)$ は生徒モデルの特徴マップ，$r(\cdot)$ は Regressor の出力である．FitNets のような特徴マップのアライメントに焦点を当てた手法の他には，蒸留損失を改善することで特徴マップを用いた知識転移を改善する取り組みも行われてい

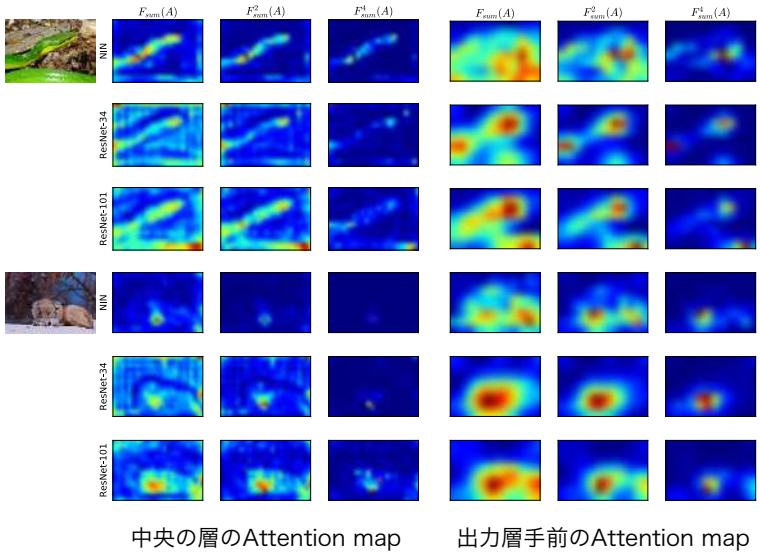


図 2.10: Attention Transfer によるアテンションマップ（文献 [3] より引用）

る [61, 62, 63, 64].

■ アテンションマップ

アテンションマップを用いた代表的な手法として，Attention Transfer [3] が挙げられる．Attention Transfer の概要を図 2.9，Attention Transfer によるアテンションマップの例を図 2.10 に示す．Attention Transfer は，特徴マップをチャネル方向に加算することでアテンションマップを作成する．アテンションマップの作成を式 2.11 に示す.

$$\boldsymbol{Q} = \sum_{i=1}^{C} |\boldsymbol{A}_i|^p \tag{2.11}$$

ここで，C は特徴マップのチャンネル数，$\boldsymbol{A}_i$ は特徴マップにおける i チャンネル目の特徴量，p はハイパーパラメータである．Attention Transfer におけるアテンションマップを知識とした蒸留損失を式 2.12 に示す.

$$L_{\mathrm{kd}} = \| \frac{\boldsymbol{Q}_s}{\|\boldsymbol{Q}_s\|_2} - \frac{\boldsymbol{Q}_t}{\|\boldsymbol{Q}_t\|_2} \|_p \tag{2.12}$$

ここで，$\boldsymbol{Q}_s$ は生徒モデルのアテンションマップ，$\boldsymbol{Q}_t$ は教師モデルのアテンションマップである．

■ 層間の相互関係

モデルの入力を質問，出力を答えとすると，中間層の特徴量は解答プロセスの中間結果と捉えることができる．しかし，教師モデルの中間層の特徴量を直接模倣することは，生徒モデルにとって

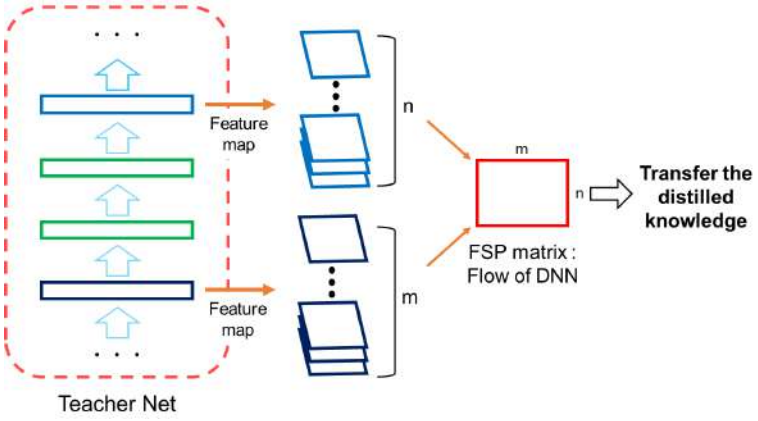


図 2.11: FSP matrix による層間の相互関係（文献 [4] より引用）

強い制約項の働きとなる可能性がある．そこで Yim ら [4] は，中間結果を直接模倣するのではなく，問題解決の流れを表現した FSP 行列を模倣することを提案した．FSP 行列の概要を図 2.11 に示す．FSP 行列は，モデル内の任意の 2 つの層から得られた特徴マップを用いて，各チャネル対の空間平均内積を計算することで，層間の情報の流れを表現する．計算するチャネルの組み合わせは総当たりで行い，その結果を行列の形で表現する．FSP 行列の作成を式 2.13 に示す．

$$G_{i,j} = \sum_{s=1}^{h} \sum_{t=1}^{w} \frac{F^1_{s,t,i}(\boldsymbol{x}) \times F^2_{s,t,j}(\boldsymbol{x})}{h \times w} \tag{2.13}$$

ここで，$\boldsymbol{x}$ は入力画像，i と j は特徴マップのチャンネル番号，h と w は特徴マップの高さと幅，$F^1_{s,t,i}(\cdot)$ は特徴マップにおける i 番目のチャンネル内の 1 画素，$F^2_{s,t,j}(\cdot)$ は特徴マップにおける j 番目のチャンネル内の 1 画素である．FSP 行列を知識とした蒸留損失を式 2.14 に示す．

$$L_{\mathrm{kd}} = \|\boldsymbol{G}^t - \boldsymbol{G}^s\|_2^2 \tag{2.14}$$

ここで，$\boldsymbol{G}^t$ は教師モデルの FSP 行列，$\boldsymbol{G}^s$ は生徒モデルの FSP 行列である．

■ サンプル間の類似度関係

確率分布や特徴マップなどを知識とした蒸留は，1 データごとに知識を転移する．そのため，距離学習のようなデータ間の関係を学習したモデルに対して，学習で獲得した知識を損なうことなく転移することが困難である．そこで，2 データ間の関係を知識とした手法，複数データ間の関係を知識とした手法，データの分布を知識とした手法が提案されている．2 サンプル間の関係を知識とした手法 [57, 65, 66, 67, 68, 69] は，2 つのデータをモデルに入力し，データに対する出力間の類似度や距離を知識として転移する．複数データ間の関係を知識とした手法 [44, 70, 71, 72, 73, 74] は，複数データに対する出力を用いてデータ間の関係をグラフ構造や確率分布で表現し，知識として転移する．サンプルの分布を知識とした手法 [75, 76, 77, 78] は，生徒モデルを Generator，教師モデルを Real データとして敵対的学習 [79] を行うことで，教師モデルが学習したサンプルの分布を知識として転移する．

2.2.3 まとめ

知識表現に焦点を当てたアプローチは，教師モデルが学習で捉えた知識をより良く表現することで，知識を転移した際の生徒モデルの性能を改善する．知識表現は，出力層の出力を知識とするアプローチ，中間層の出力を知識とするアプローチ，中間層の出力から知識を抽出するアプローチに大別できる．出力層の出力を知識とするアプローチでは，確率分布や logit を知識とする．確率分布を知識とする際には，温度付きソフトマックス関数を採用し，温度パラメータを適切な値に調整する必要がある．中間層の出力を知識とするアプローチでは，特徴マップを知識とする．教師モデルと生徒モデル間で特徴マップのサイズや特徴空間が異なることから，Regressor などによるアライメントが必要となる．中間層の出力から知識を抽出するアプローチでは，特徴マップに対して何かしらの処理を適用することで，アテンションマップや層間の情報の流れ，特徴空間上におけるサンプル間の関係などの知識を表現する．一方で知識の抽出は，抽出の処理によって特徴マップのチャンネル方向や空間方向の情報が失われてしまう．また，各アプローチでは知識表現に応じた蒸留損失の設計も行われている．

2.3 モデルの組み合わせに焦点を当てたアプローチ

KD では，パラメータ数の多い学習済みモデルとパラメータ数の少ない未学習モデルの 2 つのモデルを使用し，学習済みモデルから未学習モデルへ知識を転移する．このモデルの組み合わせや知識の転移関係は，様々な組み合わせに拡張することができる．本節では，学習済みモデルと未学習モデルを用いたオフライン蒸留と，未学習モデルのみを用いたオンライン蒸留について述べる．

2.3.1 オフライン蒸留

オフライン蒸留は，KD のように学習済みモデルである教師モデルと未学習モデルである生徒モデルを使用し，教師モデルから生徒モデルへ知識を転移するアプローチである．このアプローチでは，1 つの生徒モデルに対して，様々な教師モデルの組み合わせ方が提案されている．本項では，代表的な 5 つの教師モデルの組み合わせについて述べる．

■ 1 つの教師モデルからの知識蒸留

1 つの教師モデルからの知識蒸留は，最もシンプルな方法であり，知識表現の改善や蒸留損失の改善などモデルの組み合わせ以外に焦点を当てたアプローチにおいて広く採用されている．教師モデルと生徒モデルの組み合わせ方は，パラメータ数の多い教師モデルとパラメータ数の少ない生徒モデル [16] だけではなく，同等のパラメータ数の教師モデルと生徒モデルの組み合わせにおいても，生徒モデルの性能が改善することが知られている [5]．一方で，教師モデルと生徒モデルの間に大きな能力ギャップが存在する場合，知識の転移がうまくいかないことが知られている [6, 18, 80].

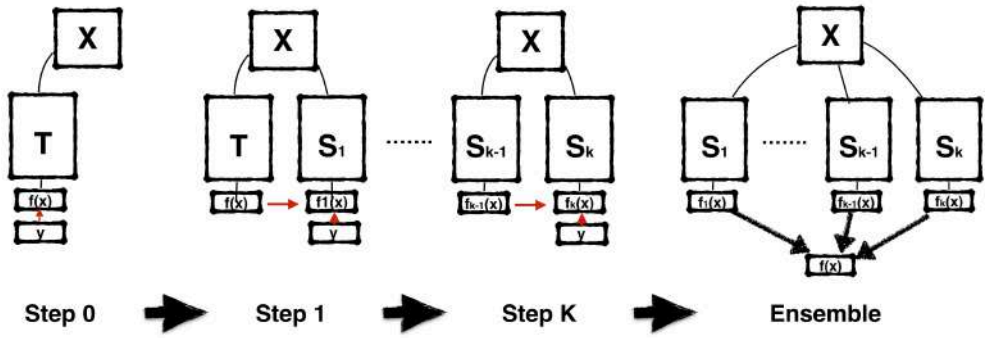


図 2.12: Born-Again Networks における段階的な知識蒸留（文献 [5] より引用）

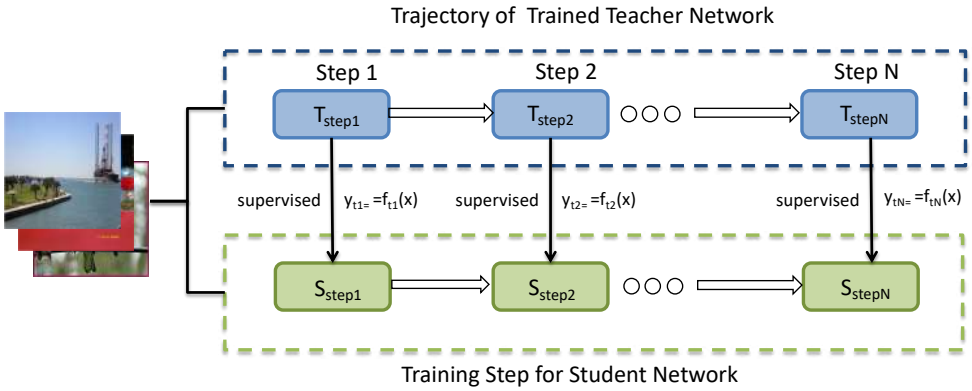


図 2.13: RCO における知識蒸留（文献 [6] より引用）

■ 段階的な知識蒸留

Born-Again Networks [5] をはじめとした手法 [81, 82, 83, 18] は，2 つのモデルを用いた知識蒸留を繰り返し行い，1 つ前の知識蒸留で学習した生徒モデルを教師モデルとして，新たな生徒モデルを学習する．最終的に複数のモデルを獲得するため，推論時に各モデルの出力分布を平均することで各モデルの予測を集約するアンサンブルを行うことで，さらなる性能改善も可能である．Born-Again Networks における段階的な知識蒸留を図 2.12 に示す．Born-Again Networks では，教師モデルと生徒モデルで同じモデル構造を使用する．一方で Teacher Assistant [18] は，能力ギャップ問題の解決を目的として，大規模な教師モデル，中規模な生徒モデル，小規模な生徒モデルを用意し，大規模な教師モデルから中規模な生徒モデルへの蒸留，蒸留後の中規模な生徒モデルから小規模な生徒モデルへの蒸留の 2 段階の蒸留を行なう．

■ 学習を早期終了した教師モデルの利用

知識蒸留における能力ギャップ問題の傾向から，必ずしも性能が良い教師モデルの知識が生徒モデルの学習に役立つとは限らない．そこで，学習を早期終了した教師モデルの利用 [6, 80] が検討されて

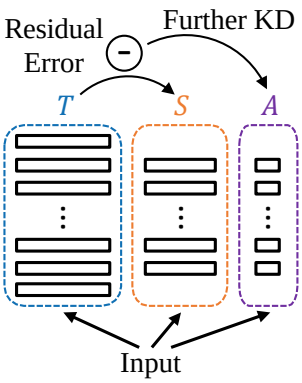


図 2.14: RKD における知識蒸留（文献 [7] より引用）

いる．Cho と Hariharan [80] は，様々なパラメータ数の教師モデルを用意して能力ギャップ問題の分析を行い，最後まで学習を行なった教師モデルと比べて，学習を早期終了した教師モデルを用いた場合に生徒モデルの性能が改善することを実験的に示した．Route Constrained Optimization (RCO) [6] は，学習初期から学習終盤までの各学習段階の教師モデルを使用した．RCO の概要を図 2.13 に示す．RCO では，生徒モデルの学習段階に合わせて蒸留に使用する教師モデルの学習段階を変更することで，能力ギャップ問題へ対応し，教師モデルの学習過程を模倣しながら知識の転移を行う．

■ 知識転移を補助するモデルの利用

知識転移を補助するモデルの利用では，教師モデルと生徒モデルに加えて，知識転移を補助するモデルを導入する．敵対的学習を利用した知識蒸留 [75, 76, 77, 78] は，生徒モデルと教師モデルに加え，生徒モデルと教師モデルの知識を見分ける Discriminator を用いて，敵対的学習 [79] を行うことで知識を転移する．Discriminator は，教師モデルまたは生徒モデルの知識を入力とし，入力が生徒モデルと教師モデルのどちらのものかを予測する学習を行う．一方で生徒モデルは，生徒モデルの知識に対する Discriminator の予測結果を用いて，Discriminator が生徒モデルの知識に対して教師モデルの知識と予測するように学習を行う．生徒モデルは，教師モデルの知識との相違度などを用いた直接的な模倣を行わないが，Discriminator を介して教師モデルの持つ知識の分布を転移することができる．Residual Knowledge Distillation (RKD) [7] は，能力ギャップ問題への対応を目的として，教師モデルと生徒モデル間の知識の差を模倣するアシスタントモデルを導入した．RKD の概要を図 2.14 に示す．生徒モデルは，教師モデルの知識を模倣するように学習し，アシスタントモデルは，教師モデルと生徒モデルの知識差を模倣するように学習する．推論時は，生徒モデルの出力とアシスタントモデルの出力を加算したものを最終出力として利用する．生徒モデルに加えて，アシスタントモデルを用意すると，パラメータの総数が増加してしまう．そこで RKD では，生徒モデルの各層をチャンネル方向に分割して 2 つのモデルを用意することで，パラメータの総数を増やすことなく，生徒モデルとアシスタントモデルを用意した．

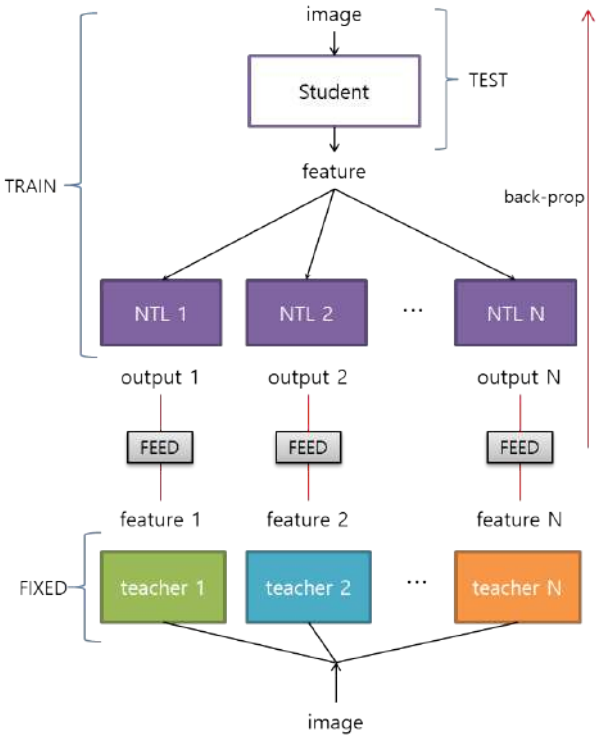


図 2.15: FEED における知識蒸留（文献 [8] より引用）

■ 複数の教師モデルからの知識蒸留

複数の教師モデルを用いた知識蒸留は，各教師モデルから個別に生徒モデルへ知識を転移する戦略 [8] と，各教師モデルの知識を 1 つに集約してから生徒モデルへ転移する戦略 [84, 85] に大別できる．FEature-level Ensemble for knowledge Distillation (FEED) [8] は，特徴マップを知識として，各教師モデルから個別に生徒モデルへ知識を転移した．FEED の概要を図 2.15 に示す．異なるモデルの特徴マップは，特徴マップのサイズや特徴空間が異なるため，生徒モデルの特徴マップを各教師モデルの特徴空間に写像するための Regressor である NTL を用意し，個別に知識を転移する．各教師モデルの知識を生徒モデルに転移することで，最終的に生徒モデルはアンサンブルのような複数の教師モデルの知識を統合したモデルとなることが期待できる．一方で，各教師モデルから個別に生徒モデルへ知識を転移すると，蒸留損失の計算コストや各蒸留損失のバランスを調整するハイパーパラメータの数が教師モデルの数に応じて増加する．そこで，各教師モデルの知識を 1 つに統合することで，1 つの教師モデルを用いたシンプルな蒸留フレームワークで学習することが可能となる [84].

2.3.2 オンライン蒸留

オンライン蒸留は，オフライン蒸留と異なり，未学習モデルである生徒モデルのみを用いて，生徒モデルの学習中に生徒モデルから生徒モデルへ知識を転移するアプローチである．オンライン蒸留

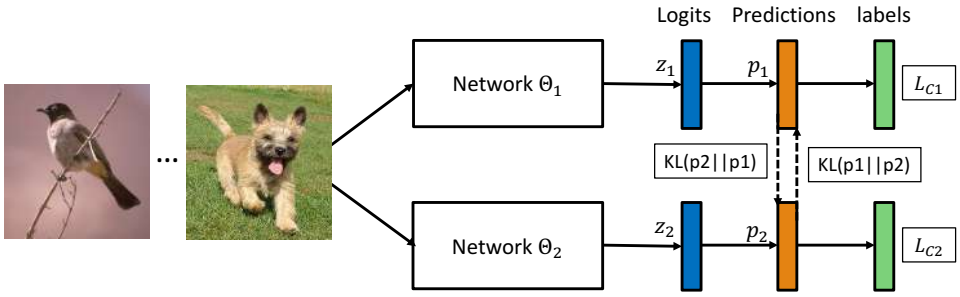


図 2.16: DML における知識蒸留（文献 [9] より引用）

では，複数の生徒モデルを用いた学習と，1 つの生徒モデルのみを用いて自分自身の知識を利用する学習に大別できる．

■ 生徒モデルのみを用いた知識蒸留

生徒モデルのみを用いた知識蒸留は，学習方法や問題設定に応じて様々な手法が提案され，生徒モデル間で知識を転移した場合においても生徒モデルの性能が改善することが知られている [9, 86, 87, 88, 89, 90, 91, 92]．生徒モデルのみを用いた知識蒸留の代表的な手法として，Deep Mutual Learning (DML) [9] が挙げられる．DML は，複数の生徒モデルを同時に学習し，生徒モデル間で双方向に知識を転移し合う手法である．DML の概要を図 2.16 に示す．学習時は，各生徒モデルに同一データを入力し，確率分布を得る．その後，各生徒モデル間で KL divergence によって確率分布の相違度を計算し，相違度が最小化するように各生徒モデルを学習する．DML は，各生徒モデルが同一のモデル構造と異なるモデル構造の両方で有効なことが実験的に示されている．また，DML の学習に使用する生徒モデルの数に応じて各生徒モデルの性能が向上する傾向にある．一方で，生徒モデルの数を増やすことはパラメータ数の増加につながり，学習時に必要な計算機のコストが増加する．そこで，Collaborative Learning [86] や On-the-fly Native Ensemble (ONE) [93] は，生徒モデルの浅い層の重みを生徒モデル間で共有することで，パラメータ数の増加を軽減した．

■ 自分自身の知識を用いた知識蒸留

自分自身の知識を用いた知識蒸留は，深い層の知識を浅い層へ転移することや，データ拡張などによる同一データに対するモデル出力の変動を利用することで，1 つの生徒モデルのみを用いて知識蒸留を実現している．深い層から浅い層への知識蒸留 [58, 59, 60] は，浅い層と比べて深い層の出力が識別に有効な情報を含んでいると仮定し，深い層の知識を浅い層へ転移する．データからデータへの知識蒸留 [94, 95] は，1 つのデータからデータ拡張によって複数のデータを作成し，同一データから作成した各データに対する知識が一致するように学習を行う．

2.3.3 まとめ

モデルの組み合わせに焦点を当てたアプローチは，知識蒸留の補助や複数モデルの知識の集約など手法ごとに異なる目的で，様々なモデルの組み合わせ方が導入された．従来のパラメータ数の大きい教師モデルとパラメータ数の少ない生徒モデルの組み合わせだけではなく，同等のパラメータ数の教師モデルと生徒モデルの組み合わせ，複数の生徒モデルからなる組み合わせ，3 モデル以上の活用，学習初期や学習途中の教師モデルの活用など様々なモデルの関係性において，知識転移の有効性が確認されている．しかし，これらのモデルの組み合わせ方は，人手によって設計されており，網羅的な評価は行われていない．そのため，より効果的なモデルの組み合わせ方が存在する可能性がある．

第3章

知識転移グラフによる深層共同学習

確率分布や中間層の特徴量を用いた複数のモデル間の知識転移は，単純な教師・生徒アプローチから双方向の転移アプローチまで，広範な手動設計により発展してきた．これらの従来法における手法設計の傾向から，効果的な知識転移の設計には，モデルのサイズ，モデルの数，知識の転移方向，損失関数の設計の 4 つの要素が重要であると考える．しかし，これらの重要な要素の組み合わせは膨大であり，複雑に絡み合っているため，従来法では人手で設計された限られた組み合わせしか評価が行われていない．

本章では，従来法に統一的な視点を与えることで多様な知識転移戦略へ拡張可能とし，効果的な知識転移の方法を探索により自動設計することを提案する．本研究では，モデルからモデルへ知識を転移する学習方法を共同学習と定義し，従来の共同学習を内包しつつ，新たな共同学習を表現可能なグラフ表現である知識転移グラフを定義する．知識転移グラフは，従来研究されてきた多くの共同学習に統一的な視点を与え，知識の移転戦略の多様性と柔軟性を大きく拡張する．知識転移グラフでは，ノードはモデルを表し，エッジは知識の転移方向と損失計算を表す．各エッジで異なる損失関数を設定することや各ノードで異なるモデル構造を設定することで，多様な共同学習を表現可能である．そして，生徒モデルの性能が高くなる知識転移グラフの構造（共同学習の方法）を人手で設計するのではなく，グラフ構造をハイパーパラメータ探索することで，人が設計した共同学習よりも効果的な共同学習の発見を目指す.

評価実験では，11 種類のクラス分類タスクのデータセットを用いて 6 つの観点から提案手法の評価を行う．1 つ目は，探索により獲得した知識転移グラフのグラフ構造の可視化である．探索により発見した知識転移戦略が従来法とどのように異なるのか評価する．2 つ目は，従来法との精度比較である．クラス分類タスクにおいて，共同学習の従来法と探索により得られた共同学習の精度を比較する．3 つ目は，探索により得られた共同学習の転移性の評価である．探索により得られた共同学習が，様々なデータセットで高い精度を発揮するのか評価する．4 つ目は，探索により得られた共同学習の分析である．探索により得られた共同学習の一部を，従来の共同学習の戦略に置き換えた際の精度変化から，探索により得られた戦略が精度に寄与しているのかを評価する．5 つ目は，グラフ構造の探索方法の分析である．探索時に選択可能な知識転移戦略の多様性が探索結果に与える影響を評価する．最後に，グラフ構造の探索における計算コストについて議論する．

本章の構成は以下の通りである．3.1 節で深層共同学習の定義について述べる．次に，3.2 節で提案手法である知識転移グラフについて述べる．続いて，3.3 節で提案手法の評価実験について述べる．最後に，3.4 節で本章についてまとめる．

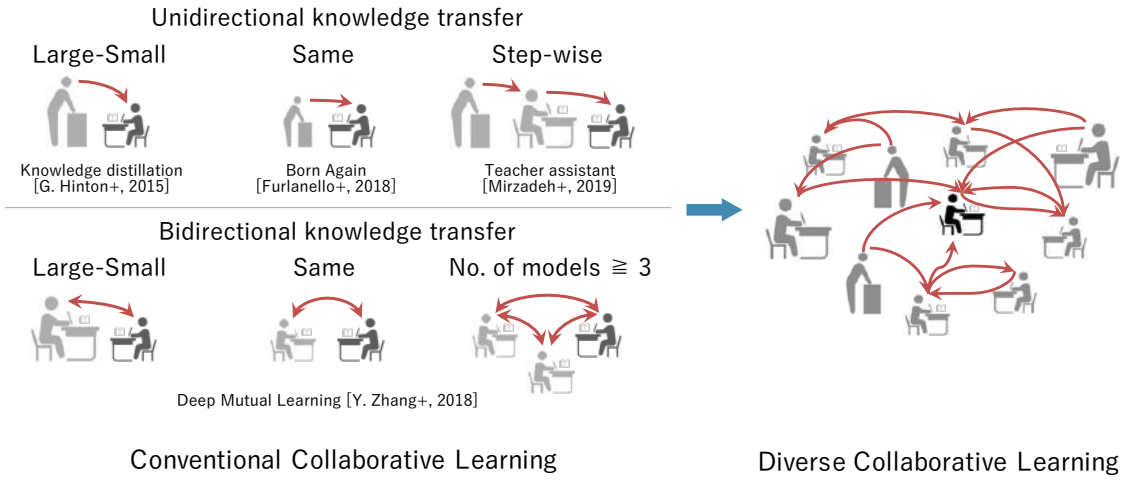


図 3.1: 深層共同学習のコンセプト

3.1 深層共同学習

第 2 章で紹介したように知識蒸留は，モデルのサイズ，モデルの数，知識の転移方向，損失関数の観点から様々な知識転移戦略が提案されている．モデルの軽量化を目的として研究が進められた知識蒸留であったが，パラメータ数が同等の教師モデルからの知識転移 [5] や生徒モデル間の双方向の知識転移 [9] においても，性能改善が可能であることが示されて以降は，性能向上を目的とした研究も行われている．本稿では，モデル軽量化を目的とした従来の知識蒸留と性能改善を目的とした派生手法を，知識の転移関係に着目して共同学習と定義する．共同学習のコンセプトを図 3.1 に示す．これにより，研究目的に応じて固定化されていたモデルの組み合わせ方や各モデルの学習方法などの設定をより柔軟に設計することを可能とする．例として，1 つの教師モデルと複数の生徒モデルを用いて，教師モデルから各生徒モデルへの知識転移，各生徒モデル間での知識転移を採用した，学校におけるクラスルームスケールのような学習が挙げられる．加えて，このような知識転移関係だけではなく，教師あり学習や半教師あり学習といった異なる学習方法の複数モデルの採用，複数モデル全体に焦点を当てたグループレベルの研究目的といった多様な共同学習も含まれる．そのため，共同学習は従来の知識蒸留の枠組みを超え，知識蒸留，相互学習．Co-Training（半教師あり学習）を包含する領域として定義する．共同学習を構成する各要素において可能なアプローチの例を以下に示す．

- 研究目的：モデルの軽量化，1 モデルの改善，複数モデルの改善，異なる学習方法の連携
- モデル　：教師と生徒，生徒のみ，構造が同じモデル，構造の異なるモデル
- 学習方法：一方向の知識転移，双方向の知識転移，モデルごとに異なる損失関数

本稿では，研究目的に焦点を当て，研究目的に応じてモデルや学習方法を設計することで，従来の知識蒸留の枠組みを超えた多様な共同学習の検討を各章で行う．

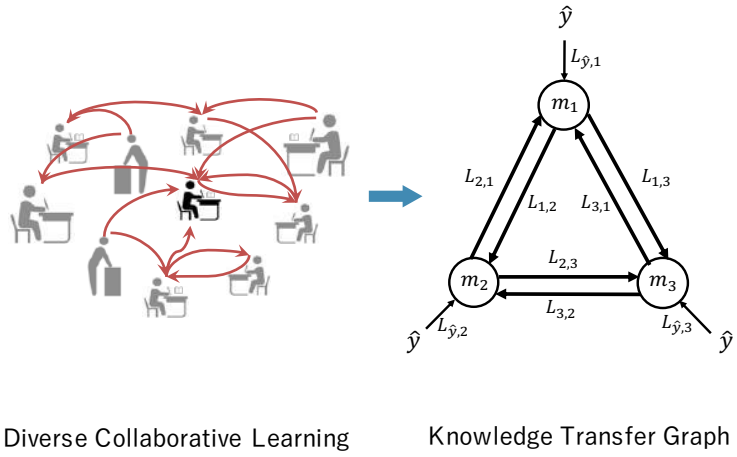


図 3.2: 知識転移グラフのコンセプト

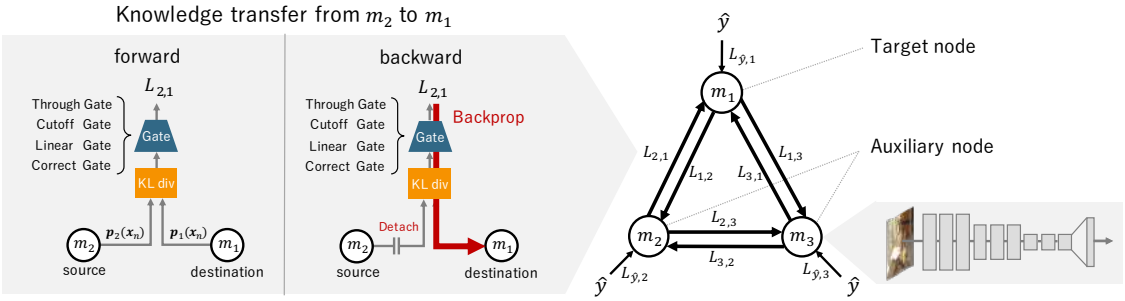


図 3.3: 知識転移グラフにおけるグラフ表現

3.2 提案手法：知識転移グラフ

本節では，従来の共同学習と新しい共同学習が表現可能なグラフ表現である知識転移グラフについて述べる．知識転移グラフのコンセプトを図 3.2 に示す．知識転移グラフでは，従来の共同学習を統一的なフレームワークで表現することで，従来法の派生手法や従来法の組み合わせを含む多様な共同学習を表現可能とする．生徒モデルの性能が高くなる効果的な共同学習は，人手で知識転移グラフの構造を設計するのではなく，ハイパーパラメータ探索によって自動設計することで，より高い性能を発揮する生徒モデルの獲得と共に，新たな知見の獲得を目指す．本節では，3.2.1 項で知識転移グラフにおけるグラフ表現，3.2.2 項で損失関数，3.2.3 項でゲート関数，3.2.4 項でモデルの最適化，3.2.5 項でグラフ構造の探索について述べる．

3.2.1 知識転移グラフにおけるグラフ表現

本項では，知識転移グラフの詳細ついて述べる．ノード数 M を 3 としたときの知識転移グラフを図 3.3 に示す．知識転移グラフは，モデルをノード，損失関数をエッジ，正解ラベルを $\hat{y}$ で表す．学

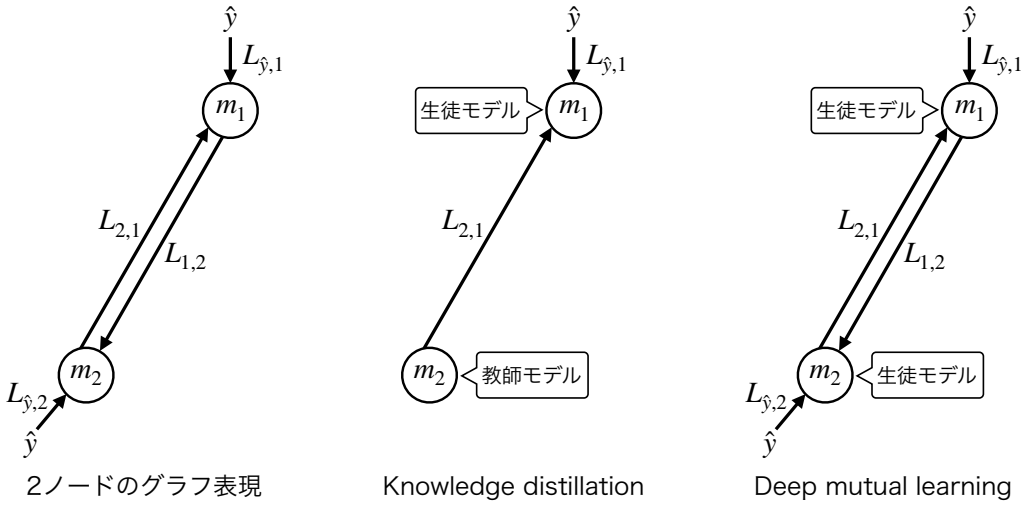


図 3.4: 従来法のグラフ表現

習に使用する i 番目のモデルをノード m_i とし，各ノード間に方向が異なる 2 つのエッジを定義し有向グラフとする．エッジの向きは，知識が伝わる方向を表す．そのため，ノードからノードへのエッジは知識転移の損失を表し，$\hat{y}$ からノードへのエッジは正解ラベルを用いた教師あり損失を表す．知識転移元のノードは source node，知識転移先のノードは destination node と呼ぶ．知識転移の損失に関する誤差逆伝播は，destination node へのみを行い，source node へは行わない．これにより，source node から destination node への知識の転移のみが行われる．

エッジには任意の損失関数，ノードには任意のモデルを設定可能とする．例えば，各エッジで同じ損失関数を設定することで，従来法の知識転移戦略が表現できる．また，各エッジで異なる損失関数を設定することで，各ノードは異なる知識転移戦略により学習することとなり，従来法にはない様々な共同学習が表現できる．2 つのノードを持つ知識転移グラフにおけるグラフ構造の例を図 3.4 に示す．2 つのノードを持つ知識転移グラフに対して，2 つのノードにそれぞれ学習済みモデルと未学習モデルを設定し，一部のエッジを削除することで Knowledge Distillation (KD) [16] が表現できる．また，2 つのノードに未学習モデルを設定し，エッジは削除しないことで，Deep Mutual Learning (DML) [9] が表現できる．

3.2.2 損失関数

n データ目の入力画像 $\boldsymbol{x}_n$ とその正解ラベル $\hat{y}_n$ で構成されるミニバッチを $\mathcal{B} = \{\boldsymbol{x}_n, \hat{y}_n\}_{n=1}^{N}$ で表し，ミニバッチ $\mathcal{B}$ のバッチサイズを $|\mathcal{B}|$ で表す．正解ラベル $\hat{y}_n$ は正解クラスの ID である．知識転移グラフのノードの数を M，知識転移元である source node を m_s，知識転移先である destination node を m_t とする．

ノード間の出力確率の差異を求める際には，Kullback-Leibler (KL) divergence $\mathrm{KL}(\boldsymbol{p}_s(\boldsymbol{x}_n)||\boldsymbol{p}_t(\boldsymbol{x}_n))$ を使用する．ここで，$\boldsymbol{p}_s$ は source node の出力分布，$\boldsymbol{p}_t$ は destination node の出力分布である．各出

力分布は，softmax 関数によって正規化された確率分布である．KL divergence は，複数の生徒モデル間でお互いに知識を転移する DML で使用されている損失関数であり，知識として確率分布を使用する場合の代表的な損失関数の 1 つである．

正解ラベル $\hat{y}_n$ の one-hot ベクトル表現を $\boldsymbol{p}_{\hat{y}_n}$ とすると，$\boldsymbol{p}_{\hat{y}_n}$ と destination node m_t の出力 $\boldsymbol{p}_t(\boldsymbol{x}_n)$ との誤差は，クロスエントロピー関数 $H(\boldsymbol{p}_{\hat{y}_n}, \boldsymbol{p}_t(\boldsymbol{x}_n))$ で算出する．$H(\boldsymbol{p}_{\hat{y}_n}, \boldsymbol{p}_t(\boldsymbol{x}_n))$ は，式 3.1 のように KL divergence と自己エントロピーの和に分解できる．

$$\begin{aligned} H(\boldsymbol{p}_{\hat{y}_n}, \boldsymbol{p}_t(\boldsymbol{x}_n)) &= \mathrm{KL}(\boldsymbol{p}_{\hat{y}_n} || \boldsymbol{p}_t(\boldsymbol{x}_n)) + H(\boldsymbol{p}_{\hat{y}_n}, \boldsymbol{p}_{\hat{y}_n}) \\ &= \mathrm{KL}(\boldsymbol{p}_{\hat{y}_n} || \boldsymbol{p}_t(\boldsymbol{x}_n)) \end{aligned} \tag{3.1}$$

$\boldsymbol{p}_{\hat{y}_n}$ は one-hot ベクトルであるため，その自己エントロピー $H(\boldsymbol{p}_{\hat{y}_n}, \boldsymbol{p}_{\hat{y}_n})$ は 0 になり，$H(\boldsymbol{p}_{\hat{y}_n}, \boldsymbol{p}_t(\boldsymbol{x}_n)) = \mathrm{KL}(\boldsymbol{p}_{\hat{y}_n} || \boldsymbol{p}_t(\boldsymbol{x}_n))$ が成り立つ．よって，正解ラベルと出力間の誤差も，ノード間の出力の誤差と同様に，KL divergence で表現する．以後，$\boldsymbol{p}_{\hat{y}_n}$ を $\boldsymbol{p}_0(\boldsymbol{x}_n) = \boldsymbol{p}_{\hat{y}_n}$ として，正解ラベルを 0 番ノードの出力 $\boldsymbol{p}_0(\boldsymbol{x}_n)$ として表す．

source node m_s から destination node m_t へ知識を伝える際に使用する損失関数を $L_{s,t}$ で表す．損失関数 $L_{s,t}$ を式 3.2 に示す．

$$L_{s,t} = \sum_{n}^{|\mathcal{B}|} G_{s,t}(\mathrm{KL}(\boldsymbol{p}_s(\boldsymbol{x}_n) || \boldsymbol{p}_t(\boldsymbol{x}_n))) \tag{3.2}$$

ここで，$G_{s,t}(\cdot)$ はゲート関数である．ゲート関数は，データごとに計算した KL divergence の値をどのように逆伝播するかを制御する．ゲート関数の詳細は，3.2.3 項で述べる．

最終的に destination node m_t の損失関数は，各ノードとの損失を総和する式となる．最終的な destination node m_t の損失関数を式 3.3 に示す．

$$L_t = \sum_{s=0, s\neq t}^{M} L_{s,t} \tag{3.3}$$

3.2.3 ゲート関数

学習フェーズ全体にわたって，source node の知識を全て destination node へ伝えると，destination node の学習に悪影響を与える可能性がある．そこで，多様な知識転移が可能となるように，知識を伝えるか否かを制御するゲート関数を損失関数へ導入する．ゲート関数の概要を図 3.5 に示す．ゲート関数は，損失関数が出力したデータごとの損失のうち，どの損失値を逆伝播するかを決めるための重み付けを行う．ゲート関数として Through Gate， Cutoff Gate， Linear Gate， Correct Gate, Sine Gate の 5 種類を定義する．

Through Gate は，入力された損失値をそのまま出力とするゲート関数である．これにより，従来の source node の知識を全て destination node へ伝える知識転移が表現できる．Through Gate を式 3.4 に示す．

$$G_{s,t}^{\mathrm{Through}}(a) = a \tag{3.4}$$

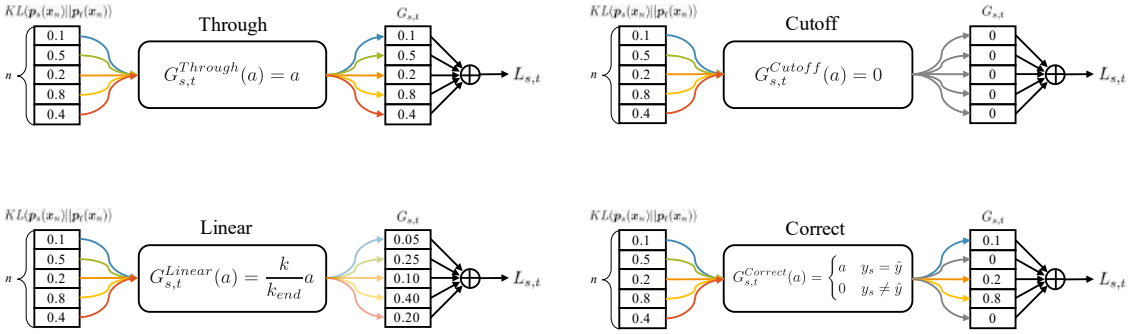


図 3.5: 4 種類のゲートの概要

ここで，a はある 1 データに対する損失の値である．

Cutoff Gate は，入力された損失値によらず 0 を出力するゲート関数である．Cutoff Gate を用いることで，知識転移グラフの任意のエッジを切断することができる．このゲート関数により，KD のようなモデルからモデルへの一方向の知識転移や知識転移を行わない関係が表現できる．Cutoff Gate を式 3.5 に示す．

$$G_{s,t}^{\mathrm{Cutoff}}(a) = 0 \tag{3.5}$$

Linear Gate は，学習時間によって損失の重みを線形に変化させるゲート関数である．学習初期には重みを小さくし，学習が進むにつれて重みが大きくなる．このゲート関数により，学習序盤は知識を積極的には転移せず，学習が進むにつれて知識を徐々に強く転移する動的な知識転移が表現できる．また，source node が事前学習済みモデルではない場合に，学習初期の不確かな知識の転移の抑制が期待できる．Linear Gate を式 3.6 に示す．

$$G_{s,t}^{\mathrm{Linear}}(a) = \frac{k}{k_{\mathrm{end}}} a \tag{3.6}$$

ここで，k は累積更新回数であり，k_{end} は学習終了時の更新回数である．

Correct Gate は，source node が正解したデータの損失のみを通すゲート関数である．このゲート関数により，source node が間違えたデータに対する知識の転移を抑制できる．Correct Gate を式 3.7 に示す．

$$G_{s,t}^{\mathrm{Correct}}(a; \hat{y}, y_s) = \begin{cases} a & y_s = \hat{y} \\ 0 & y_s \neq \hat{y} \end{cases} \tag{3.7}$$

ここで，y_s は source node m_s の Top-1 クラスの ID である．source node が事前学習済みモデルではない場合に Linear Gate と Correct Gate は，似た学習効果が期待できる．しかし，Linear Gate が各データの損失値に同じ重みを乗算するのに対して，Correct Gate はデータごとに乗算する重みを選定するため，学習初期から知識を転移することが可能である．

これらの 4 種類のゲートを用いて共同学習を自動設計すると，各ゲートが明確な目的を持っているため，自動設計した共同学習の解釈が容易となる．一方で，あらかじめゲートのパターンを 4 種

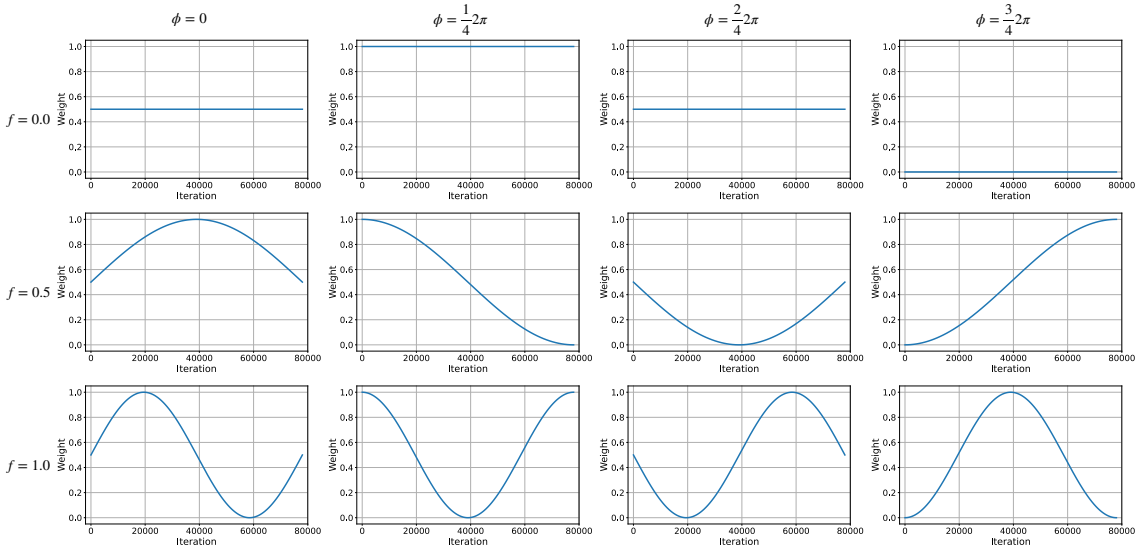


図 3.6: Sine Gate の概要

類に固定しているため，選択できる知識転移の戦略は限定的となる．そこで，Through Gate，Cutoff Gate，Linear Gate を内包しつつ，より多様な重み付けが表現可能な Sine Gate を設計する．Sine Gate は，ハイパーパラメータである f と ϕ に応じて多様な重み付け戦略を表現可能なゲートである．sine Gate を式 3.8 に示す．

$$G_{s,t}^{\mathrm{Sine}}(a) = (0.5 + 0.5\sin(2\pi f_{s,t}\frac{k}{k_{\mathrm{end}}} + \phi_{s,t})) \cdot a \tag{3.8}$$

Sine Gate による重み付け戦略の例を図 3.6 に示す．f と ϕ をそれぞれ 0 と $1/4 \cdot 2\pi$ と設定することで Through Gate，0 と $3/4 \cdot 2\pi$ と設定することで Cutoff Gate，0.5 と $3/4 \cdot 2\pi$ と設定することで Linear Gate が表現できる．このように f と ϕ を調整することで，様々な定数重み付けや学習戦略の切り替えが表現可能である．

3.2.4 モデルの最適化

学習するモデルの更新方法を Algorithm 1 に示す．はじめに，全てのモデルの重みを乱数で初期化する．ただし，ノード m_i を destination node とするゲート関数 $G_{i,t}$ が全て Cutoff Gate の場合，学習フェーズ中に m_i は学習を行わないことになるため，学習済みモデルの重みで初期化する．学習済みモデルは，共同学習で使用するデータセットにおいて，正解ラベルのみを用いて学習を行ったモデルである．学習済みモデルの重みで初期化したノード m_i は，学習中に重みを凍結し，重みの更新は行われない．そのため，このノードは KD における教師モデルに相当する役割となる．

全てのノードには同じデータを入力し，損失を計算する．得られた損失から勾配を計算して，全てのノードを同時に更新する．式 3.3 により計算した損失 L_t の勾配は，ノード m_t にのみ逆伝播され，他のノードには影響を与えない．DML では，1 つ目のノードの重みを更新した後，更新したノードに再びデータを入力して出力分布を得る．そして，更新したノードを含めた各ノードの出力分布か

Algorithm 1 共同学習におけるモデルの更新

Require: ノード数 M, エポック数 E,

Ensure: 全てのモデルの重みを初期化，または，学習済みモデルの重みを読み込む

for _ = 1 to E **do**

各モデル m_n に同一画像を入力して確率分布 $\boldsymbol{p}_n$ を得る

各ノードの出力 $\boldsymbol{p}_1(\boldsymbol{x}_n), \boldsymbol{p}_2(\boldsymbol{x}_n), \cdots, \boldsymbol{p}_M(\boldsymbol{x}_n)$ を得る

式 (3.3) に従って損失 L_n を求める．

L_n の勾配から m_n の重みの更新量を求める．

全てのモデルの重みを更新する．

end for

らノード間の損失を再度計算して，2 つ目のノードに対して勾配降下法を実行する．これを全てのノードが更新されるまで繰り返す．しかし，この更新方法はノード数が増加するにしたがって計算コストが大幅に増加する．そこで，本稿では一度の forward 計算で全てのノードの重みを更新する．

3.2.5 グラフ構造の探索

共同学習により性能を向上させたいノードを target node と呼び，target node の学習をサポートする役割のノードを auxiliary node と呼ぶ．target node で使用するモデルの種類と知識転移グラフのノードの数は，探索を行う前にあらかじめ設定する．知識転移グラフのハイパーパラメータは，auxiliary node として使用するモデルの種類とエッジで使用するゲート関数の種類である．エッジごと，ノードごとに使用するハイパーパラメータの設定を行う．そのため，探索空間の大きさは，ノード数を N, モデルの種類数を M，ゲートの種類数を G とすると，$M^{(n-1)} \cdot G^{N^2}/2$ である．これは，例えば $N = 3, M = 3, G = 4$ とすると，200 万通り以上の非常に膨大な共同学習が表現できる．

ゲート関数の選択肢として 4 種類のゲート，auxiliary node の選択肢として 3 種類のモデル構造を設定する．知識転送グラフの構造は，各エッジのゲート関数と各 auxiliary node のモデル構造を独立してランダムに選択することによって選択する．ゲート関数の選択肢として 4 種類のゲートの代わりに Sine Gate を使用する場合，各エッジにおいて f を 0 から 1 の範囲，ϕ を 0 から $3/4 \cdot 2\pi$ の範囲から値をランダムに選択する．

ハイパーパラメータの最適化手法として Asynchronous Successive Halving Algorithm (ASHA) [96] を使用する．ASHA は，並列分散環境で積極的な早期終了を導入したランダムサーチであり，時間効率と精度を両立することができる．ハイパーパラメータ探索時は，D 個の GPU サーバーを使用し，各サーバーでは独立してランダムに選択されたグラフ構造を評価する．各サーバーにおいてモデルは，Algorithm 1 を使用して，選択されたグラフ構造に基づいた共同学習が行われる．各サーバーでは，$1, 2, 4, \cdots, 2^k$ エポック目で検証用データを用いて target node の精度を評価する．target node の精度が過去に評価された全ての target node の精度のうち下位 50%に入っている場合，そのサーバーでの学習を終了する．最終エポックまたは学習が早期終了した時のエポックにおける target node の精

度が，そのサーバーで使用したグラフ構造の評価スコアとなる．学習を終了したサーバーでは，新しい知識転移グラフを用意して学習を始める．この時，新たに作成した知識転移グラフの各ノードは，3.2.4 項に基づいて初期化を行う．各サーバーで学習を終了した知識転移グラフの総数が T 個になるまで，各サーバーで学習と評価を繰り返す．T 個のグラフ構造を評価した後，最も高い target node の精度を達成したグラフ構造を探索結果の知識転移グラフとする．3.3 節の実験では，$D = 30, T = 1500$ の条件を使用した．

3.3 評価実験

評価実験として，グラフ構造の探索により獲得した知識転移グラフを用いて学習した target node の性能を評価する．3.3.1 項では，評価実験に使用した実験設定について述べる．3.3.2 項では，探索により獲得した知識転移グラフのグラフ構造を可視化する．3.3.3 項では，クラス分類タスクにおいて共同学習の従来法と精度を比較する．3.3.4 項では，クラス分類タスクにおいてモデルの軽量化を目的とした共同学習の従来法と精度を比較する．3.3.5 項では，探索により得られた共同学習を探索時と異なるデータセットで評価する．3.3.6 項では，探索で選択されたゲート関数の有効性を評価する．3.3.7 項では，グラフ構造の探索におけるゲート関数の候補の多様性が探索結果に与える影響を評価する．3.3.8 項では，グラフ構造の探索における計算コストについて議論する．

3.3.1 実験条件

本項では，評価実験に使用したデータセット，モデル，モデルの学習条件と評価条件，グラフ構造の探索条件について述べる．

■ データセット

グラフ構造の探索のためのデータセットとして，CIFAR-10 [97]，CIFAR-100 [97] と Tiny-ImageNet [98] を使用する．性能評価のためのデータセットとして，CIFAR-10, CIFAR-100，Tiny-ImageNet, Caltech-101 [99], FGVC Aircraft [100], Stanford Cars [101]，Food-101 [102]，Oxford-IIIT Pets [103], Oxford 102 Flowers [104]，SUN397 [105]，DTD [106] を使用する．CIFAR-10, CIFAR-100，Tiny-ImageNet，Caltech-101 は，それぞれ 10 クラス，100 クラス，200 クラス，101 クラスの一般物体で構成されたデータセットである．FGVC Aircraft は 102 クラスの飛行機の機種，Stanford Cars は 196 クラスの自動車の車種，Food-101 は 101 クラスの食べ物の種類，Oxford-IIIT Pets は計 37 クラスの犬種と猫種，Oxford 102 Flowers は 102 クラスの花の種類，SUN397 は飛行機の客室やコンピュータルームなどの 397 クラスのシーン，DTD は 47 クラスのテクスチャで構成されたデータセットである．

表 3.1: 正解ラベルのみを用いた学習における各モデルの精度 [%]

Model	CIFAR-10	CIFAR-100	TinyImageNet
ResNet32	92.99 ± 0.28	70.71 ± 0.39	52.89 ± 0.18
ResNet110	94.01 ± 0.28	72.59 ± 0.54	55.49 ± 0.55
WRN28-2	94.40 ± 0.07	74.60 ± 0.38	58.60 ± 0.25

■ モデル

モデルとして，ResNet32 [10]，ResNet110 [10]，Wide ResNet 28-2 (WRN28-2) [107] を使用する．これらのモデルは，CIFAR データセットの画像サイズに合わせて，畳み込み層などの設定が行われている．本項の実験で使用する CIFAR 以外のデータセットは，CIFAR と比べて画像サイズが大きい．そのため，Tiny-ImageNet データセットで学習する場合は，最初の畳み込み層のストライドを 1 に設定する．また，CIFAR と Tiny-ImageNet 以外のデータセットで学習する場合は，1 層目の畳み込み層のカーネルサイズを 7, ストライドを 1, パディングを 3 に設定し，1 層目の後にカーネルサイズが 7, ストライドが 7, パディングが 0 の Maxpooling を導入する．

各モデルを正解ラベルのみを用いて学習した場合の精度を表 3.1 に示す．3 つのモデルの中で，ResNet32 は最も精度の低いモデル，WRN28-2 は最も精度の高いモデルであり，ResNet110 はその中間のモデルである．

■ モデルの学習条件と評価条件

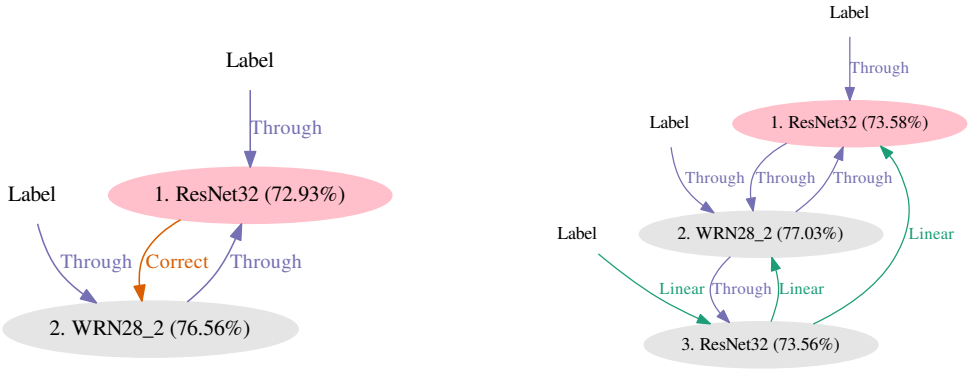
本項における全ての実験と手法において，以下に示す学習条件と評価条件を使用する．最適化アルゴリズムは SGD と Nesterov momentum を使用し，初期学習率は 0.1，momentum は 0.9，バッチサイズは 64 とする．CIFAR で学習する場合は，60 エポックごとに学習率を 10 分の 1 にし，合計で 200 エポックの学習を行う．Tiny-ImageNet で学習する場合は，40，60，70 エポック目に学習率を 10 分の 1 にし，合計で 80 エポックの学習を行う．

CIFAR-10，CIFAR-100，Tiny-ImageNet では，学習データに対して 4 ピクセルのパディング，ランダムクロップ，ランダムフリップをデータ拡張として適用し，テストデータに対してはデータ拡張を行わない．その他のデータセットでは，学習データに対してランダムクロップ，ランダムフリップをデータ拡張として適用し，テストデータの画像に対してリサイズとセンタークロップを適用する．

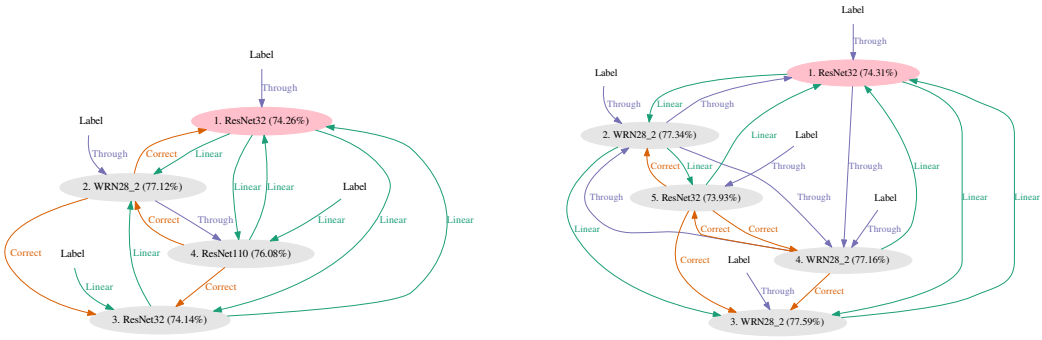
報告する結果は，モデルの重みの初期値を変更して 5 回試行したときの平均値と標準偏差である．

■ グラフ構造の探索条件

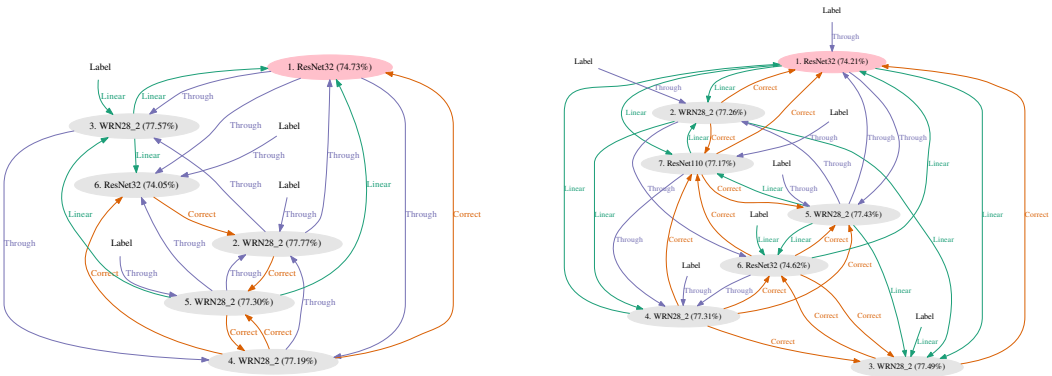
データセットのテストデータへの過剰な適合を防ぐために，グラフ構造の探索時はデータセットの学習データをグラフ探索用の学習データとグラフ探索用の検証データに分割し，データセットに用意されている本来のテストデータは使用しない．グラフ構造の探索時の指標である target node の



(a) 2 ノード (72.93%)　　(b) 3 ノード (73.58%)



(c) 4 ノード (74.26%)　　(d) 5 ノード (74.31%)



(e) 6 ノード (74.73%)　　(f) 7 ノード (74.21%)

図 3.7: 4 種類のゲートと 3 種類のモデル構造を候補とした場合の探索結果

精度は，グラフ探索用の検証データに対する精度を使用する．探索により獲得した知識転移グラフを用いてモデルの学習と評価をする際には，データセットで用意されている元々の学習データとテ

ストデータを使用する．

CIFAR-10 と CIFAR-100 データセットを用いて探索する場合は，50,000 サンプルの学習データをグラフ探索用の学習データとして 40,000 サンプル，グラフ探索用の検証データとして 10,000 サンプルに分割する．Tiny-ImageNet dataset を用いて探索する場合は，100,000 サンプルの学習データをグラフ探索用の学習データとして 90,000 サンプル，グラフ探索用の検証データとして 10,000 サンプルに分割する．

探索時の GPU サーバーとして，90 台の Quadro P5000 サーバを使用する．

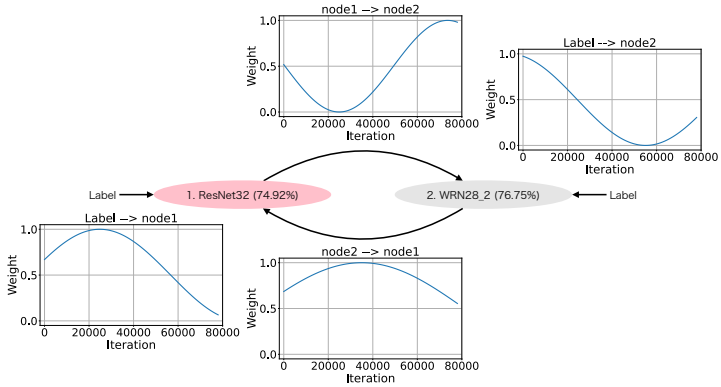
3.3.2 探索後のグラフ構造の可視化

本項では，探索用のデータセットとして CIFAR-100，target node としてノード 1，target node で使用するモデル構造として ResNet-32 を使用した場合の探索結果を示す．ゲート関数の候補として Through Gate, Cutoff Gate, Linear Gate, Correct Gate，auxiliary node のモデル構造の種類として ResNet32, ResNet110, WRN28-2 を使用した場合の探索結果を図 3.7 に示す．ゲート関数として Sine Gate，auxiliary node のモデル構造の種類として ResNet32, ResNet110, WRN28-2 を使用した場合の探索結果を図 3.8 示す．Sine Gate では，f と ϕ の値を探索した．全てのグラフにおいて，target node の精度は表 3.1 で示した ResNet32 の精度と比較して高い結果となった．また，探索時の指標として使用していない auxiliary node の精度も同様に表 3.1 で示した精度と比較して高い結果となった．共同学習の従来法との精度比較は，3.3.3 項で示す．

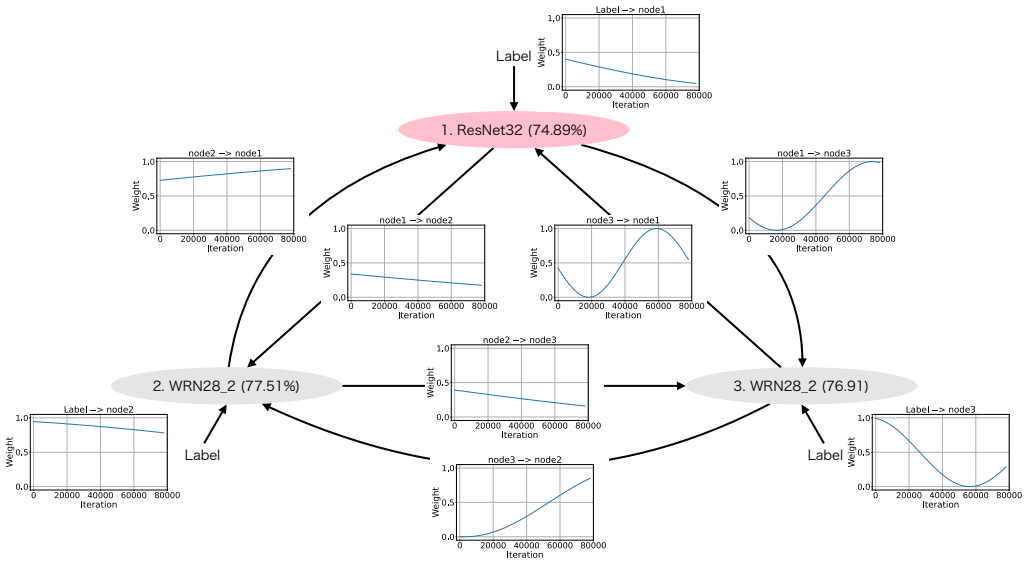
図 3.7a のノード数 2 の知識転移グラフは，双方向に知識を伝え合う共同学習となっている．ResNet32 から WRN28-2 へのエッジでは Correct Gate，それ以外のエッジは Through gate が選択されている．小さなモデルである ResNet32 から大きなモデルである WRN28-2 への知識の転移において，ResNet32 が間違えたサンプルの知識を伝えないことで，WRN28-2 の学習へ悪影響を与えない効果があると考えられる．

図 3.7b のノード数 3 の知識転移グラフは，一方向の知識転移と双方向の知識転移を組み合わせた共同学習となっている．学習初期は，Linear Gate が選択されたエッジの損失が 0 に近い値となるため，ノード 1 とノード 2 では双方向の知識転移，ノード 2 からノード 3 では一方向の知識転移が行われる．学習が進むにつれて，Linear Gate が選択されたエッジにおける知識転移が徐々に有効になるため，ノード 2 とノード 3 の間の学習が一方向の知識転移から双方向の知識転移へ変化し，ノード 1 とノード 3 において一方向の知識転移が新たに行われる．そのため，一方向の知識転移と双方向の知識転移を組み合わせ，学習中に知識の転移戦略が動的に変化する新しい共同学習が獲得できたと言える．

図 3.7c のノード数 4 の知識転移グラフは，合計 6 つの知識転移が設定可能な中で，1 つの一方向の知識転移，5 つの双方向の知識転移を行う共同学習となっている．エッジに着目すると，多くのエッジで Linear Gate が選択されている．そのため，学習初期は性能が高いモデルが選択されたノード 2 やノード 4 の知識を用いた学習が行われる．具体的には，ノード 1 は性能が高いノード 4 の知識を用いて学習，ノード 3 はノード 2 とノード 4 の知識を用いて学習を行う．また，性能が高いノード 2



(a) 2 ノード (74.92%)



(b) 3 ノード (74.89%)

図 3.8: Sine Gate と 3 種類のモデル構造を候補とした場合の探索結果

とノード 4 の間では，双方向の知識転移が行われ，良い知識を用いてお互いの性能を高め合っていると言える．その後，学習が進むにつれて様々なノード間で双方向の知識転移が行われる．このような共同学習は，意図的に人が設計するのが困難であると言える．

図 3.7d のノード数 5，図 3.7e のノード数 6，図 3.7f のノード数 7 の知識転移グラフは，人が解釈することが難しい複雑な関係の共同学習となっている．

Sine Gate を用いた図 3.8a のノード数 2 と図 3.8b のノード数 3 の知識転移グラフは，複雑な重み付け戦略となっている．選択可能な重み付けの戦略を固定しないことで，より多様な重み付け戦略が選択され，図 3.7 のグラフと比べて少ないノード数で高い精度を達成した．しかし，非常に複雑な重み付け戦略のため，これらの共同学習の戦略を人が解釈することが困難である．

表 3.2: CIFAR-100 における従来法との精度比較

Method	Accuracy (Node 1)	Node 1	Node 2	Node 3	Node 4
Vanilla	70.71 ± 0.39	ResNet32	–	–	–
DML [9]	72.00 ± 0.44	ResNet32	ResNet32	–	–
KD ($T = 2$) [16]	71.88 ± 0.78	ResNet32	WRN28-2*	–	–
DML [9]	72.71 ± 0.18	ResNet32	WRN28-2	–	–
Ours (Four types of gates)	**72.88** ± 0.41	ResNet32	WRN28-2	–	–
Ours (Sine Gate)	**74.10** ± 0.48	ResNet32	WRN28-2	–	–
DML [9]	72.09 ± 0.43	ResNet32	ResNet32	ResNet32	–
DML [9]	72.89 ± 0.21	ResNet32	WRN28-2	ResNet32	–
Ours (Four types of gates)	**73.46** ± 0.28	ResNet32	WRN28-2	ResNet32	–
DML [9]	73.03 ± 0.27	ResNet32	WRN28-2	WRN28-2	–
Ours (Sine Gate)	**74.36** ± 0.38	ResNet32	WRN28-2	WRN28-2	–
DML [9]	72.76 ± 0.35	ResNet32	ResNet32	ResNet32	ResNet32
Song [86] †	73.68 ± 0.26	(4×ResNet32 with shared intermediate layers)			
ONE [93] †	73.42 ± N/A	(4×ResNet32 with shared intermediate layers)			
DML [9]	72.87 ± 0.49	ResNet32	WRN28-2	ResNet32	ResNet110
Ours (Four types of gates)	**74.06** ± 0.34	ResNet32	WRN28-2	ResNet32	ResNet110
DML [9]	73.38 ± 0.20	ResNet32	WRN28-2	WRN28-2	WRN28-2
Ours (Sine Gate)	**73.77** ± 0.29	ResNet32	WRN28-2	WRN28-2	WRN28-2

3.3.3 従来法との精度比較

CIFAR-100 を用いて探索したグラフ構造に基づいた共同学習を CIFAR-100 における学習で評価を行う．具体的には，代表的な従来法，3 モデル以上を用いた共同学習が可能な従来法と精度を比較する．ResNet32 のテストデータに対する精度の比較結果を表 3.2 に示す．各ノード数における Ours は，それぞれ CIFAR-100 を使用して探索したノード数 2，3，4 のグラフ（図 3.7a, 図 3.7b, 図 3.7c）に基づいて学習を行った target node の精度である．表内の “†” は対応する手法の論文から引用した精度であることを意味し，“*” はそのノードに事前学習済みモデルが採用されたことを意味する．KD [16] は，教師モデルと生徒モデルの 2 つのモデルを用いた一方向の知識転移手法である．学習済みモデルである教師モデルは，ノード数 2 の Ours で選択された WRN28-2 を使用し，温度パラメータを $T = 2$ として学習した．DML [9] は，複数の生徒モデルを用いた双方向の知識転移手法である．DML では，全てのモデルを ResNet32 とする場合と，各ノード数の Ours で選択されたモデルを使用する場合の 2 パターンで学習を行った．Song *et al.* [86] と ONE [93] は，各ノードのモデル間で浅い層を共有するモデル構造を採用する双方向の知識転移手法である．Song *et al.* [86] と ONE [93] では，各ノードの出力分布をアンサンブルにより統合した確率分布を知識とし，双方向の知識転移を行う．

表 3.3: CIFAR-100 におけるモデルの軽量化を目的とした従来法との精度比較

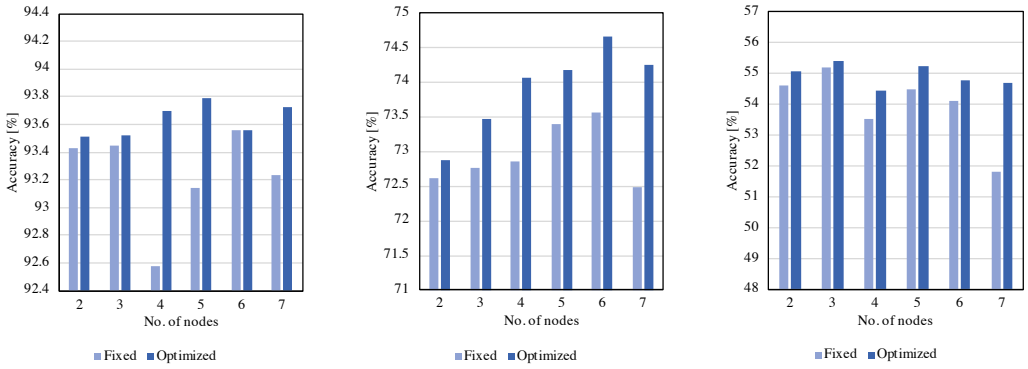
Method	Trained Teacher	Knowledge type			Accuracy [%]
		Feature	Relation	Logit	
FitNets [2]	✓	✓			70.94 ± 0.22
AT [3]	✓	✓			71.88 ± 0.17
SVD [34]	✓	✓			71.39 ± 0.34
VID-I [61]	✓	✓			71.22 ± 0.54
CRD [62]	✓	✓			73.76 ± 0.27
KR [108]	✓	✓			72.38 ± 0.30
SRD [109]	✓	✓			74.69 ± 0.19
PKT [110]	✓		✓		72.45 ± 0.48
RKD [65]	✓		✓		70.87 ± 0.23
CCKD [68]	✓		✓		74.71 ± 0.07
SP [67]	✓		✓		74.66 ± 0.11
ICKD [111]	✓		✓		74.68 ± 0.14
KD (T=2) [16]	✓			✓	71.88 ± 0.78
DKD [112]	✓			✓	72.49 ± 0.34
KD + DOT [113]	✓			✓	71.41 ± 0.33
KD + logit standardization [114]	✓			✓	74.65 ± 0.12
Ours				✓	74.10 ± 0.48

探索により獲得した知識転移グラフを使用して学習した Ours は，共同学習の従来法を上回る精度を達成した．Ours で選択されたモデルを使用した DML は，図 3.7 の各グラフのゲート関数を全て Through Gate に置き換えた学習と言える．そのため，Ours で選択されたモデルを使用した DML と比較して，Ours の精度が向上していることから，グラフ構造の探索によって効果的な知識転移戦略が選択されたと言える．

Ours (Sine gate) は，ノード数が 2 と 3 の場合において Ours (Four types of gates) を超える精度を発揮した．特に，2 ノードの Ours (Sine gate) は，ノード数が 4 の Ours (Four types of gates) と同等の精度であり，多様な重み付け戦略が設計可能なことで，より効果的な知識転移戦略が獲得できたと言える．一方で，ノード数が 4 の Ours (Sine gate) は，ノード数が 2 と 3 の Ours (Sine gate) より低い精度となった．Sine Gate は多様な重み付け戦略が表現可能な一方で，ノード数に応じて表現可能な共同学習の数が増加する．そのため，今回使用した 1,500 種類のグラフ構造をランダムに選択して評価する探索方法は，Sine Gate を用いた探索において探索不足の可能性が考えられる．

表 3.4: 探索時と異なるデータセットにおける精度評価

Method	Number of nodes in graph				
	Two nodes	Three nodes	Four nodes	Five nodes	Six nodes
	Caltech-101				
DML [9]	48.59	40.49	38.84	11.44	12.02
Ours (Four types of gates)	58.65	60.60	**61.33**	59.24	**60.18**
Ours (Sine Gate)	**60.42**	**60.94**	57.94	**59.29**	58.00
	FGVC Aircraft				
DML [9]	23.88	11.10	4.11	7.29	2.70
Ours (Four types of gates)	**38.70**	38.76	**36.84**	41.55	**43.38**
Ours (Sine Gate)	34.23	**40.32**	35.88	**43.05**	42.93
	Stanford Cars				
DML [9]	43.10	27.84	10.83	8.32	14.61
Ours (Four types of gates)	**59.63**	**59.57**	**59.22**	**60.08**	**57.49**
Ours (Sine Gate)	58.74	57.79	55.34	59.06	55.35
	Food-101				
DML [9]	53.70	54.91	57.22	56.65	52.62
Ours (Four types of gates)	55.01	56.61	57.52	**58.63**	61.15
Ours (Sine Gate)	**58.48**	**59.85**	**58.01**	56.36	**61.43**
	Oxford-IIIT Pets				
DML [9]	14.47	13.19	10.28	9.59	8.07
Ours (Four types of gates)	**36.39**	31.94	**31.97**	**33.93**	**40.83**
Ours (Sine Gate)	30.66	**36.20**	31.04	31.72	31.32
	Oxford 102 Flowers				
DML [9]	22.67	5.27	8.49	2.41	3.71
Ours (Four types of gates)	**33.70**	**37.38**	35.26	39.29	37.47
Ours (Sine Gate)	29.31	37.32	**38.04**	**40.25**	**39.55**
	SUN397				
DML [9]	23.58	25.75	26.37	28.30	29.04
Ours (Four types of gates)	26.28	**28.35**	27.79	29.20	**30.27**
Ours (Sine Gate)	**26.60**	27.85	**28.37**	**30.04**	30.24
	DTD				
DML [9]	11.17	8.24	4.26	6.97	6.33
Ours (Four types of gates)	**21.97**	**20.53**	**18.83**	21.91	**25.32**
Ours (Sine Gate)	17.71	18.40	18.56	**22.66**	17.66



(a) CIFAR-10 (b) CIFAR-100 (c) Tiny-ImageNet

図 3.9: 探索により選択されたゲートの評価

3.3.4 モデル軽量化を目的とした従来法との精度比較

共同学習における最先端の手法の多くは，モデルの軽量化を目的として，教師モデルから生徒モデルへの一方向の知識転移を行う．そこで，図 3.7a のノード数が 2 のグラフにおいて選択された WRN28-2 を教師モデル，ResNet32 を生徒モデルとして従来法を学習し，精度比較を行う．CIFAR-100 における ResNet32 のテストデータに対する精度の比較結果を表 3.3 を示す．Ours は，最先端の手法と比べて約 0.5 ポイント低い精度となった．最先端の手法は，損失関数の設計，勾配や温度パラメータに焦点を当てたアプローチにより高い精度を達成している．一方で，Ours はモデル間の知識の転移関係や転移タイミングに焦点を当てていることから，異なるアプローチにより最先端の手法に匹敵する精度を達成したと言える．

3.3.5 探索により獲得したグラフ構造の転移性

探索を行った知識転移グラフを，探索時と異なるデータセットにおいて評価を行う．具体的には，CIFAR-100 データセットで探索したノード数が 2 から 6 のグラフ構造を，問題設定が大きく異なる 8 種類のデータセットで学習と評価を行った．各データセットにおけるノード 1 (ResNet32) のテストデータに対する精度を表 3.4 に示す．DML [9] は，全てのノードで ResNet32 を使用した．全てのデータセットにおいて，Ours の精度が DML [9] を上回っている．この結果は，獲得したグラフ構造が探索時とは異なるデータセットへ転用可能であることを示している．そのため，探索時とは異なる新しいデータセットを適用した場合に，新しくグラフ構造を探索する過程を省略できるため，大幅な計算コストの削減が可能である．

表 3.5: ゲート候補の多様性が探索に与える影響

Gate types used as candidates in search					Accuracy [%]
Through	Cutoff	Correct	Linear	Sine	
✓					73.22 ± 0.40
✓	✓				73.29 ± 0.20
✓	✓	✓			73.42 ± 0.26
✓	✓	✓	✓		73.46 ± 0.28
				✓	74.36 ± 0.38

3.3.6 探索により選択されたゲートの評価

探索によって選択されたゲート関数が target node の精度へ与える影響を調査する．target node として ResNet32 を使用し，ノード数が 2 から 7 までのグラフ構造をそれぞれ探索する．ゲート関数の候補として Through Gate，Cutoff Gate，Linear Gate，Correct Gate を使用し，auxiliary node で使用するモデル構造の候補として ResNet32，ResNet110，WRN28-2 を使用する．また，探索用のデータセットとして CIFAR-10，CIFAR-100，Tiny-ImageNet を使用し，それぞれのデータセットでグラフ構造を探索する．

各データセットにおける target node の精度を図 3.9 に示す．Optimized は，獲得した知識転移グラフを用いて学習を行った target node の精度である．Fixed は，獲得した知識転移グラフの全ての gate 関数を Through Gate に変更した知識転移グラフにより学習を行った target node の精度である．そのため Fixed は，モデルの組み合わせ方を工夫した DML であると言える．Optimized は，全ての条件において Fixed よりも高い精度を達成した．このことから，適切なゲートを使用して知識転移を制御することの重要性を示していると言える．

3.3.7 ゲート候補の多様性が探索に与える影響の評価

グラフ構造の探索におけるゲート関数の候補の多様性が探索結果に与える影響を評価する．具体的には，ノード数が 2 つのグラフの探索において，ゲート関数の候補の数を変更する．各探索では，auxiliary node で使用するモデル構造の候補として ResNet32，ResNet110，WRN28-2 を使用する．

評価結果を表 3.5 に示す．✓ は，グラフ構造の探索時にゲート関数の候補として使用したことを意味する．ゲート関数の候補の数を増やすと，精度が向上する傾向にある．この結果は，各ゲート関数がそれぞれ精度に寄与していることを示しており，ゲート関数の候補の数を増やすことでより多様な学習方法が表現可能となり，結果として効果的な知識転移の戦略が獲得できたと考える．

表 3.6: Sine Gate を用いた探索における探索時間 (Wall-clock time)

Number of nodes	Two	Three	Four	Five	Six
Time (h)	9.43	11.87	16.03	15.95	24.90

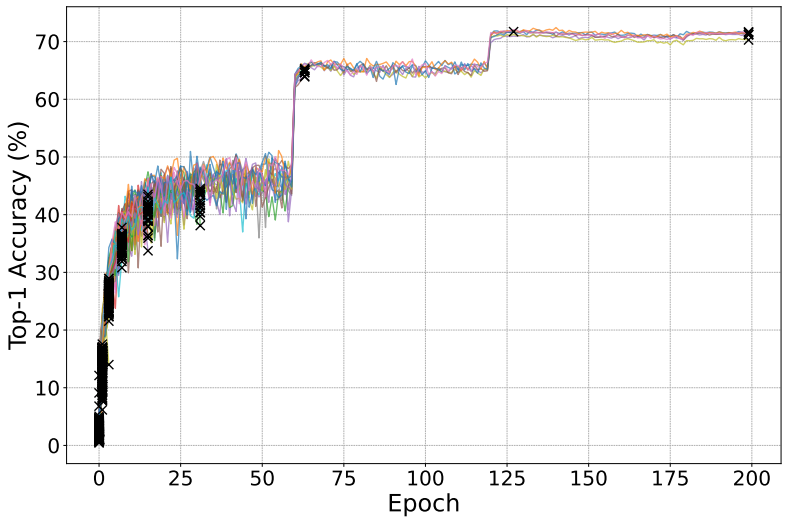


図 3.10: 探索のための学習における各グラフ構造の精度変化

3.3.8 グラフ構造の探索コスト

グラフ構造の探索コストは，グラフ構造のノード数，評価するグラフ構造の数，使用するハイパーパラメータ探索の手法に依存する．本節の評価実験では，ハイパーパラメータ探索の手法として，ランダムサーチと ASHA を使用した．ASHA は，並列探索が可能な枝刈りアルゴリズムのため，探索の時間的コストは並列可能な環境数に依存する．

1,500 のグラフ構造を評価するグラフ構造の探索に必要とした wall-clock time を表 3.6 示す．グラフのノード数に応じて，探索が完了するまでに必要な時間が増加していることがわかる．これは，ノード数が増えることで共同学習を行う関係の数が増え，モデルの forward 処理や損失計算の回数が増えたことが原因である．

探索のための学習における各グラフ構造の精度変化を図 3.10 に示す．ばつ印は，最終エポックの到達または ASHA による学習の早期終了によって，学習が終了したタイミングを表している．200 エポック目ではないタイミングでばつ印がある場合，ASHA によって学習の早期終了が行われたことを意味する．1,500 のグラフ構造のうち，6 種類の構造のみが最終エポックに到達した．このことから，ASHA によって精度が高くなる見込みがないグラフ構造を早期に打ち切ることで，効率的な探索を行うことができたと言える．

一方で，ゲート関数の中には，学習の更新回数によって損失に対する重み付けの値が変化するゲート関数があることから，ASHA によって学習終盤に高い精度となるような学習方法を早期終了してしまっている可能性も存在する．また，膨大な探索空間の中から 1,500 のグラフ構造のみを評価して

いるため，効果的なグラフ構造の獲得は可能であるが，必ずしも最も良いグラフ構造が得られている保証はない．知識転移グラフのグラフ構造に適した効率的な探索方法の設計は，今後の課題である．

3.4 まとめ

本章では，複数のモデル間で知識を転移する手法を共同学習と定義し，共同学習の従来法を表現可能な知識転移グラフを提案した．知識転移グラフでは，モデルをノード，損失計算と知識転移の関係をエッジで表現し，共同学習の従来法に統一的な視点を与えた．また，損失関数に損失値に重み付けを行うゲート関数を導入することで，多様な知識転移戦略を表現可能とした．そして，効果的なグラフ構造を人手により設計するのではなく，探索によって自動設計することで，従来法では探索しきれなかった大規模かつ複雑な共同学習の設計を可能にした．11 種類のデータセットを用いた評価実験では，探索により従来法と異なる知識転移の戦略を獲得し，探索により獲得した共同学習が従来法を上回る精度を達成することを確認した．また，探索で選択されたモデル構造やゲート関数が精度に寄与していることを確認し，探索時におけるグラフ構造の候補の多様性が大きいほど，効果的なグラフ構造を発見しやすいことを確認した．今後の課題として，探索のコスト削減や中間層の知識の導入，グループ単位の性能改善への拡張などが挙げられる．

第4章

多様な知識転移によるアンサンブル学習

DML [9] をはじめとする双方向の共同学習は，複数の生徒モデルの学習時に生徒モデル間で互いに知識を転移することで，各生徒モデルの精度を向上させる．さらに，推論時に複数モデルの出力を平均化するアンサンブル学習を用いることで，精度をさらに改善することが可能である．しかし，共同学習したモデルのアンサンブル学習は，共同学習を用いない一般的なアンサンブル学習と比較して，精度の改善幅が低くなる．この傾向から，アンサンブル学習による精度向上に不可欠なモデルの多様性と共同学習におけるモデルの知識との間には関係性があると考える．

そこで本章では，アンサンブル学習と共同学習の関係を分析し，多様な知識の獲得を促す共同学習によってアンサンブル学習の高精度化を行う．分析では，正解ラベルのみを用いて学習した複数のモデルに対して，モデル間の知識の相違度とアンサンブル学習による精度向上の関係を調査する．その後，分析の傾向に基づいて，モデル間の知識に多様性を促すアンサンブル学習のための共同学習について検討する．アンサンブル学習のための共同学習として，モデル間の知識を離す学習を設計する．しかし，出力分布を知識とする共同学習において，モデル間の知識を離す学習を行った場合，出力分布の多様化が期待できるが，モデルの精度低下が懸念される．そこで，精度低下を引き起こすことなく多様性を得るためにアテンションマップに着目し，出力分布とアテンションマップを知識とした共同学習を行う．これにより，アンサンブル学習に適した多様な共同学習を実現できる．また，共同学習はモデル間ごとに異なる知識転移の設計をすることが可能だが，モデル数に応じて組み合わせ方が膨大になるため，アンサンブル学習に適した共同学習を人手で意図的に設計することは困難である．そこで，知識転移グラフをグループ単位の学習に拡張することでアンサンブル学習に適用し，アンサンブル精度を指標としてグラフ構造の探索を行うことでアンサンブル学習に適した共同学習を自動設計する．最後に，各モデルの個性を考慮した軽量化のための知識転移手法を設計し，自動設計した共同学習により獲得した多様性のある複数モデルの知識を単一のモデルに転移することで，アンサンブル学習の推論コストの問題を解決する．

評価実験では，5 種類のクラス分類タスクのデータセットを用いて 5 つの観点から提案手法の評価を行う．1 つ目は，探索により獲得した知識転移グラフのグラフ構造の可視化である．探索により発見した知識転移戦略が従来法とどのように異なるのか評価する．2 つ目は，従来法との精度比較である．クラス分類タスクにおいて，探索により得られた共同学習によるアンサンブル精度を共同学習の従来法やアンサンブル学習と比較する．3 つ目は，探索により得られた共同学習法の転移性の評価である．探索により得られた共同学習法が，様々なデータセットで高い精度を発揮するのか評価す

る．4 つ目は，学習後のモデルとアンサンブル精度の関係の分析である．探索により得られた共同学習法により異なる知識を獲得した各モデルがアンサンブル精度に寄与しているのか調査する．最後に，多様な知識を持つ複数のモデルに対して，各モデルの個性を考慮した軽量化のための知識転移手法の評価を行う．

本章の構成は以下の通りである．4.1 節で本章の提案手法に関連する深層学習におけるアンサンブル学習について述べる．4.2 節で共同学習とアンサンブル学習の分析について述べる．4.3 節で提案手法である多様な知識転移によるアンサンブル学習と各モデルの個性を考慮した軽量化のための知識転移について述べる．4.4 節で提案手法の評価実験について述べる．最後に，4.5 節で本章についてまとめる．

4.1 関連研究：深層学習におけるアンサンブル学習

アンサンブル学習は，複数の機械学習モデルを組み合わせて使用する機械学習アルゴリズムである．複数の弱識別器を作成し，各弱識別器の出力を統合することで，１つの弱識別器による推論と比べて認識精度が向上する．代表的な手法としてバギング [115] やブースティング [116] が挙げられる．バギングは，データセットからデータをランダムサンプリングすることでサブセットを作成し，各弱識別器は異なる構成のサブセットから学習する．ブースティングは，弱識別器を１つずつ学習し，データと弱識別器にそれぞれ重み付けを行う．学習後の弱識別器の結果から分類が難しいデータを分類できるように，データに対して重み付けを行い，次の弱識別器を学習する．各弱識別器の出力を統合する際は，弱識別器のエラー率から計算した信頼度に基づいて出力を統合する．

深層学習におけるアンサンブル学習は，重みの初期値が異なる複数のモデルを独立して学習し，複数のモデルを利用した推論を行う．従来の機械学習におけるアンサンブル学習と異なり，各モデルは重みの初期値のみが異なり，データセットの構成や学習条件は同一である．複数のモデルを用いた推論は，入力データに対して各モデルが出力した確率分布または logit を平均した結果から推論を行う．複数のモデルを用いた推論を式 (4.1) に示す．

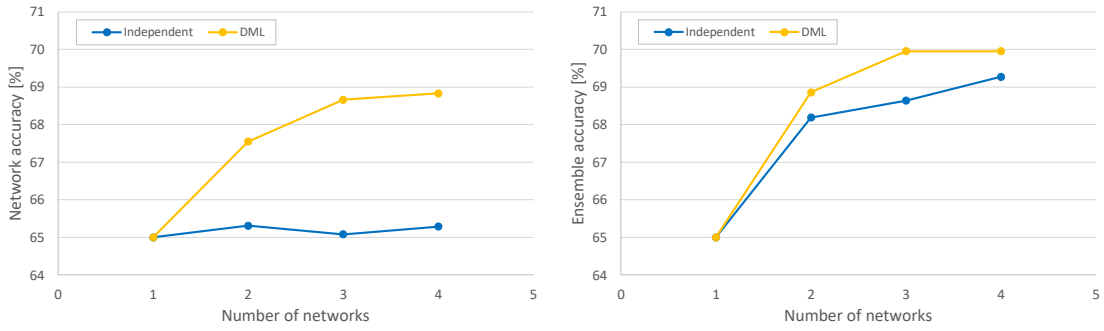
$$y = \frac{1}{M}\sum_{m=0}^{M} p_m(x) \tag{4.1}$$

ここで，M はモデル数，$p_m(\cdot)$ はモデルの出力，x は入力データを表す．複数のモデルを用いて推論を行うことで，１つのモデルによる推論と比べて認識精度が向上する．また，推論に用いるモデル数に応じて認識精度が向上し，あるモデル数を超えた段階で認識精度が向上しなくなることが知られている．本章では，深層学習における推論のみを行うアンサンブル学習をアンサンブル推論，アンサンブル推論による精度をアンサンブル精度，アンサンブル精度を改善するための学習アプローチをアンサンブル学習と呼ぶ．

深層学習におけるアンサンブル学習の手法は，大きく学習時や推論時のコストの削減，様々な問題設定への適用，アンサンブル推論の分析という観点から研究が行われている．

コストの削減 アンサンブル学習は，複数のモデルを利用する性質からモデル数に応じて重みパラメータ数が増えるため，学習時と推論時の計算コストとメモリコストが増加する．Batch ensemble [117] と Hyperparameter ensemble [118] は，モデル間で重みパラメータの一部を共有することでパラメータ数の増加を防いだ．Snapshot ensembles [119] は，１つのモデルの学習時に学習過程のモデルを保存し，推論時に利用することで１つのモデルの学習のみでアンサンブル推論を可能とした．Knowledge distillation [16] では，アンサンブル推論による確率分布を知識として 1 つの生徒モデルを学習することで，アンサンブル推論と同程度の性能を発揮する１つのモデルを獲得することを可能とした．

様々な問題設定への適用 アンサンブル推論は，教師あり学習によるクラス分類問題だけではなく，半教師あり学習 [120]，Adversarial attack [121]，Out-of-distribution 検知 [122] など様々な問題設定に対しても効果を発揮することが確認されている．



(a) モデルの精度　　(b) アンサンブル精度

図 4.1: モデル数による精度変化

アンサンブル推論の分析　深層学習モデルにおけるアンサンブル推論の効果は，重みパラメータの初期値やミニバッチのデータ選択など学習時のランダムに決定される要素によって発生した，個々のモデルの独立した誤差によるものだと考えられている．Allen-Zhu ら [123] は，モデルの重みパラメータの初期値により各モデルが異なる畳み込みフィルタを獲得することが，アンサンブル推論に寄与しているのではないかと理論的，実験的に示している．CNN は，畳み込み層によって特徴を抽出し，その特徴に対して重み付き和を繰り返すことで，最終的に確率分布を出力する．そのため，抽出される特徴が異なることでモデルは異なる確率分布を出力し，アンサンブル推論は各モデルが抽出した特徴を集約して推論を行っていると考えられている．

4.2　アンサンブル学習と共同学習の分析

双方向の共同学習として代表的な DML は，複数の生徒モデルを用いて相互的に知識を転移することで，正解ラベルのみを用いた学習と比べてモデルの精度が向上する．本節では，まず双方向の共同学習を行ったモデルの精度が，正解ラベルのみを用いた学習したモデルと比べて大きく向上したにも関わらず，双方向の共同学習を行ったモデルのアンサンブル精度が，正解ラベルのみを用いた学習したモデルのアンサンブル精度から大きく向上しないことを示す．この傾向は，共同学習における知識とアンサンブル推論におけるモデルの個性に関係性があるためだと考える．そこで，モデル間の知識の相違度とアンサンブル推論による精度向上の関係を調査し，知識を近づける共同学習と知識を離す共同学習がアンサンブル推論へ与える影響を調査する．本節では，4.2.1 項でアンサンブル学習と共同学習の関係，4.2.2 項でモデルの知識と個性の分析，4.2.3 項でアンサンブル学習のための多様な共同学習の検討について述べる．

4.2.1 アンサンブル学習と共同学習の関係

DML におけるモデルの精度とアンサンブル精度の関係を図 4.1 に示す．モデルは ResNet18 [10] をベースモデルとした Attention branch network [124]，データセットは Stanford Dogs [125] を使用した．Independent は，重みパラメータの初期値が異なる複数の生徒モデルを，正解ラベルのみを用いて独立して教師あり学習した際の精度である．

モデルの精度に着目すると DML は，Independent と比べて精度が向上している．DML によるモデルの精度向上は，共同学習に使用するモデル数が多いほど高くなり，モデル数が 4 の場合は約 4 ポイント向上している．Independent は，モデル間で知識を転移せず，独立して学習を行うため,，モデル数によらず同程度の精度を発揮している．

アンサンブル推論に着目すると，DML はモデルの精度向上に対して，アンサンブル精度は微小な向上となっている．モデル数が 4 の場合は，モデルの精度が 4 ポイント向上しているにも関わらず，アンサンブル精度は 1 ポイント未満の向上にとどまっている．この傾向は，共同学習における知識とアンサンブル学習におけるモデルの個性に関係性があるためだと考える．

4.2.2 モデルの知識と個性の分析

4.2.1 項の傾向から，モデル間の知識の相違度と個性の相違度の関係を調査する．本項では，2 個のモデルを用いたアンサンブル推論に焦点を当てて分析を行う．モデルの知識として確率分布を使用し，知識の相違度として DML の損失として使用される KL divergence を使用する．個性の相違度は，モデル間の個性に違いがあるほどアンサンブル推論による精度向上の効果が高くなると考え，アンサンブル推論による精度向上を指標とする．

■ 知識の相違度と個性の相違度

知識の相違度 知識の相違度は，確率分布間の相違度を表す KL divergence により計算する．しかし，KL divergence は距離の公理を満たさないため，対称性がない．そこで分析では，知識の相違度を KL divergence を用いて次のように定義する．

$$\mathrm{Mutual\,KL} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}(\mathrm{KL}(\boldsymbol{p}_1(\boldsymbol{x}_n)\|\boldsymbol{p}_2(\boldsymbol{x}_n)) + \mathrm{KL}(\boldsymbol{p}_2(\boldsymbol{x}_n)\|\boldsymbol{p}_1(\boldsymbol{x}_n)) \tag{4.2}$$

$$\mathrm{KL}(\boldsymbol{p}_1(\boldsymbol{x}_n)\|\boldsymbol{p}_2(\boldsymbol{x}_n)) = \sum_{c=1}^{C} p_1^c(\boldsymbol{x}_n)\log\frac{p_1^c(\boldsymbol{x}_n)}{p_2^c(\boldsymbol{x}_n)} \tag{4.3}$$

ここで，N はデータ数，$\boldsymbol{x}_n$ は入力データ，$\boldsymbol{p}_1(\cdot)$ と $\boldsymbol{p}_2(\cdot)$ はモデル 1 の確率分布とモデル 2 の確率分布，$p_1^c(\cdot)$ と $p_2^c(\cdot)$ はクラス c に対するモデル 1 の確率値とモデル 2 の確率値，C はクラス数である．

個性の相違度 個性の相違度は，アンサンブル推論に使用するモデルの平均精度とアンサンブル推論の精度の差から評価する．2 個のモデルを用いたアンサンブル推論による精度向上は次のように定義

表 4.1: 100 個の ResNet における精度傾向 [%]

	最大値	最小値	中央値	平均値
Stanford Dogs	66.28	63.79	65.02	65.02
Stanford Cars	85.47	82.38	83.91	83.89
CUB-200-2011	59.16	55.52	57.79	57.79
CIFAR-10	92.50	91.31	92.05	92.04
CIFAR-100	67.78	66.00	67.13	67.10

する．

$$\Delta\mathrm{Accuracy} = \mathrm{ACC}_{\mathrm{ens}} - \frac{1}{2}(\mathrm{ACC}_1 + \mathrm{ACC}_2) \tag{4.4}$$

ここで，$\mathrm{ACC}_{\mathrm{ens}}$ はアンサンブル精度，ACC_1 はモデル 1 の精度，ACC_1 はモデル 2 の精度である．

■ 分析条件

データセットごとに 100 個のモデルを用意して分析を行う．データセットは，Stanford Dogs [125]，Stanford Cars [101]，CaltechUSCD Birds (CUB-200-2011) [126]，CIFAR-10 [97]，CIFAR-100 [97] を使用する．モデルは ResNet [10] を使用し，データセットが CIFAR-10 と CIFAR-100 のとき ResNet20 を使用し，Stanford Dogs，Stanford Cars，CUB-200-2011 のとき ResNet18 を使用する．モデルの学習条件は全てのデータセットにおいて，最適化方法を Momentum SGD，初期学習率を 0.1，Momentum の係数を 0.9，weight decay を 0.0001，ミニバッチサイズを 16，学習回数を 300 epoch とし，学習率は 150 epoch と 225 epoch のタイミングで 0.1 倍する．100 個のモデルは，重みパラメータの初期値が異なる 100 個のモデルを用意し，共同学習などは行わずに独立して教師あり学習を行う．アンサンブル推論は，100 個のモデルから 2 個のモデルを選択して行い，全ての組み合わせ (4,950 組) において知識の相違度と個性の相違度の評価を行う．

■ 100 個のモデルの傾向

モデルの精度 各データセットにおける 100 個の ResNet の精度を表 4.1 に示す．データセットによらず，同一の条件で学習した場合においても精度に小さなばらつきがあることがわかる．

アンサンブル精度 各データセットにおける 4,950 組のアンサンブル精度を表 4.2，アンサンブル推論による精度向上を表 4.3 に示す．このとき，アンサンブル推論による精度向上は式 (4.4) により算出した．データセットによらず，アンサンブル推論に使用するモデルによって，アンサンブル精度とアンサンブル推論による精度向上にばらつきがあることがわかる．

表 4.2: 4,950 組におけるアンサンブル精度の傾向 [%]

	最大値	最小値	中央値	平均値
Stanford Dogs	69.24	66.88	68.02	68.02
Stanford Cars	87.34	84.65	85.92	85.94
CUB-200-2011	62.08	58.54	60.42	60.42
CIFAR-10	94.06	92.91	93.50	93.50
CIFAR-100	73.15	71.05	72.09	72.09

表 4.3: 4,950 組におけるアンサンブル推論による精度向上の傾向 [%]

	最大値	最小値	中央値	平均値
Stanford Dogs	3.71	2.22	3.00	2.99
Stanford Cars	2.81	1.46	2.05	2.05
CUB-200-2011	3.89	1.77	2.69	2.70
CIFAR-10	1.92	0.96	1.46	1.45
CIFAR-100	5.88	4.09	4.99	4.99

■ 分析結果

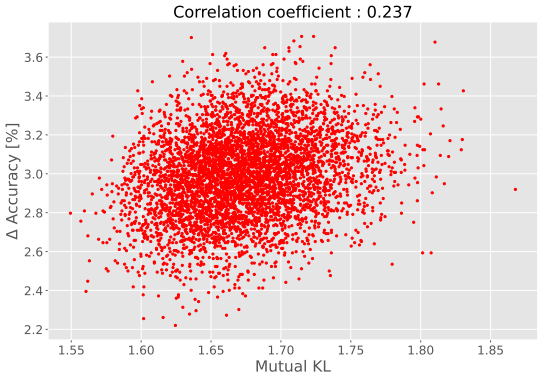
各データセットにおける 2 つの指標の関係性と相関係数を図 4.2 に示す．散布図における各点は，ある 1 組のアンサンブル推論における評価結果を表す．各データセットにおいて，確率分布の相違度とアンサンブル推論による精度向上の間に弱い正の相関があることがわかる．このことから，モデルの出力分布間に多様性を持たせるように学習することで，アンサンブル推論による精度向上の改善が見込めると考えられる．

4.2.3 アンサンブル学習のための多様な共同学習の検討

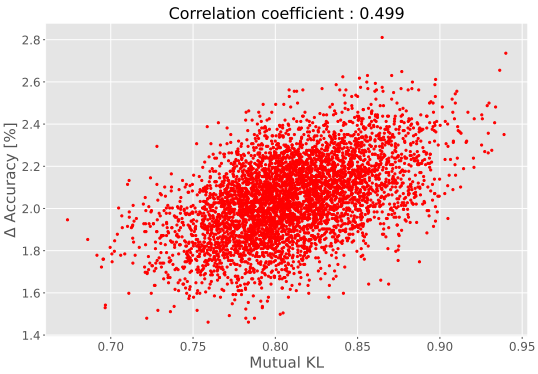
4.2.2 項の分析の傾向からアンサンブル推論による精度向上の改善を目的として，知識を離す共同学習を検討する．そこで，知識を近づける共同学習と知識を離す共同学習を行い，アンサンブル精度への影響を調査する．

■ モデル間の多様性を促す損失設計

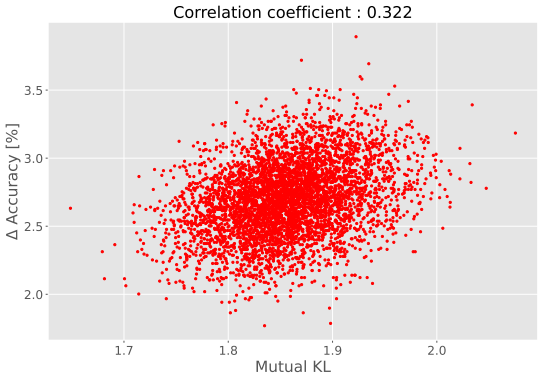
4.2.2 項の分析では，確率分布の相違度とアンサンブル推論による精度向上に正の相関があることを確認した．この傾向に基づいて，モデル間の確率分布を直接離すように学習した場合，モデルの精度が低下する可能性がある．そこで知識として，モデルの最終的な出力である確率分布と中間層の情報を表すアテンションマップの 2 つを利用する．



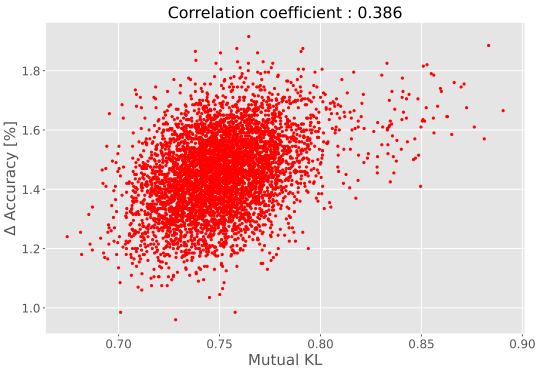
(a) Stanford Dogs



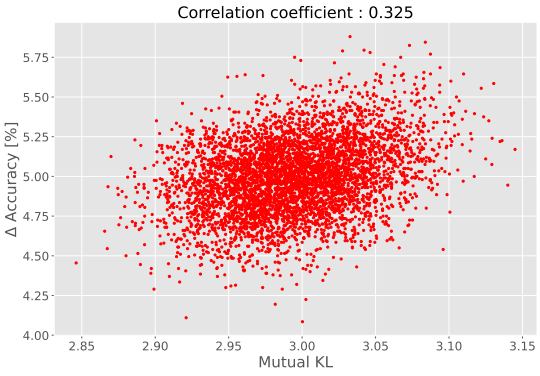
(b) Stanford Cars



(c) CUB-200-2011



(d) CIFAR-10



(e) CIFAR-100

図 4.2: 知識の相違度と個性の相違度の関係

確率分布に関する損失設計は，確率分布を近づける場合に KL divergence，確率分布を離す場合にコサイン類似度を利用する．アテンションマップに関する損失設計は，アテンションマップを近づける場合に平均二乗誤差，アテンションマップを離す場合にコサイン類似度を利用する．最終的なモデルの損失関数は，次のように定義する．

$$L_t = L_{\mathrm{CE}} + \sum_{s=1, s\neq t}^{M} L_{s,t} \tag{4.5}$$

ここで，L_{CE} は正解ラベルとモデル t の出力分布の交差エントロピー損失，$L_{s,t}$ はモデル s からモデル t への知識転移の損失，M は共同学習に使用するモデル数を表す．$L_{s,t}$ は，確率分布を近づける損失，確率分布を離す損失，アテンションマップを近づける損失，アテンションマップを離す損失の中から 1 つを選択し，全てのモデル間で同じ知識転移の損失設計を使用する．そのため，この損失関数による共同学習は，DML のような双方向の共同学習の派生と捉えることができる．

■ 学習条件

モデルは ResNet18 [10]，データセットは Stanford Dogs [125] を使用する．アテンションマップは，Attention Transfer [3] を用いて ResNet18 の ResBlock4 の出力から作成する．最適化方法は Momentum SGD，初期学習率は 0.1，Momentum の係数は 0.9，weight decay は 0.0001，ミニバッチサイズは 16，学習回数は 300 epoch とし，150 epoch と 225 epoch のタイミングで学習率を 0.1 倍する．モデル数は 1 から 4 とし，4 種類の共同学習において，各損失設計のアンサンブル精度への影響を調査する．このとき，モデル数が 1 の場合は $L_{s,t}$ を使用しない，正解ラベルのみを用いた教師あり学習を行う．

■ 評価結果

知識転移の各損失設計によるアンサンブル精度の変化を図 4.3 に示す．ここで，Independent は共同学習を行わない学習，Prob(KL-div) は確率分布を近づける共同学習，Prob(cosine sim) は確率分布を離す共同学習，Attention(MSE) はアテンションマップを近づける共同学習，Attention(cosine sim) はアテンションマップを離す共同学習を行ったモデルのアンサンブル精度である．学習結果から，モデル数が少ない場合に確率分布を近づける共同学習，モデル数が多い場合にアテンションマップを離す共同学習がアンサンブル推論に有効であると言える．また，モデル数が 4 の場合はアテンションマップを近づける以外の共同学習が Independet のアンサンブル精度を超えている．このことから，近づける学習と離す学習を組み合わせることで，より高いアンサンブル推論の効果が期待できると考える．

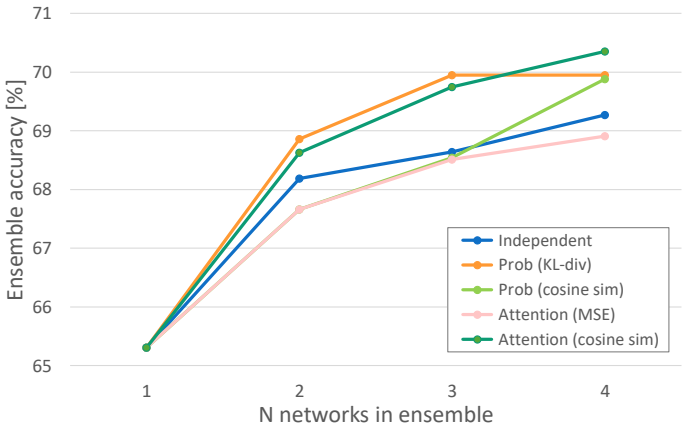


図 4.3: 多様な共同学習によるアンサンブル精度の変化

表 4.4: 確率分布を知識とした損失における指標によるアンサンブル精度の変化 [%]

Loss design	Knowledge	Ensemble accuracy
KL	+	69.95
$-\cos$	+	70.00
$1/\cos$	+	69.12
cos	−	69.88
$-$KL	−	57.87
1/KL	−	67.88

表 4.5: アテンションマップを知識とした損失における指標によるアンサンブル精度の変化 [%]

Loss design	Knowledge	Ensemble accuracy
MSE	+	68.91
$-\cos$	+	68.93
$1/\cos$	+	68.98
cos	−	70.35
$-$MSE	−	64.73
1/MSE	−	68.50

■ 損失計算に使用する指標による精度変化

知識転移の損失に使用する指標がアンサンブル精度へ与える影響を調査する．図 4.3 において，モデル数 4 の場合にアテンションマップを近づける以外の共同学習が Independet のアンサンブル精度を超えていることから，モデル数 4 の場合において各指標の評価を行う．

確率分布を知識とした損失における結果を表 4.4，アテンションマップを知識とした損失における結果を表 4.5 に示す．Loss design は損失計算に使用した指標を表し，Knowledge の + は知識を近づける共同学習，− は知識を離す共同学習であることを表す．このとき，一部の指標において損失値に対する重み付けを行った．重み付けの値として，確率分布を知識とした損失の 1/cos は 10e−4，アテンションマップを知識とした損失の 1/cos と 1/MSE はそれぞれ 10e−1 と 10e−7 を使用した．

2 つの評価結果から，確率分布とアテンションマップの両方で，似た傾向となっていることがわかる．知識を近づける場合は，損失設計によらず同程度のアンサンブル精度となった．一方で知識を離す場合は，負の相違度を使用した場合にアンサンブル精度が低下した．このことから，損失関数を設計する場合，最小化問題となるように設計する必要があると言える．

4.3　提案手法：多様な知識転移によるアンサンブル学習

4.2 節では，双方向の共同学習とアンサンブル推論の関係を分析し，4 種類の多様な共同学習がアンサンブル精度へ与える影響を調査した．アンサンブル精度が高くなる共同学習は，アンサンブル推論に用いるモデル数によって異なる傾向であった．これらの傾向は，4 種類の共同学習の中から 1 つを選択し，単純な共同学習の設定において評価を行った．そのため，4 種類の共同学習を組み合わせることで，更なるアンサンブル精度の改善を達成できる可能性がある．しかし，各モデル数において高いアンサンブル精度を達成できる共同学習を人手で試行錯誤して設計することは，組み合わせ数の観点から困難となる．また，アンサンブル推論は複数のモデルを使用するため，推論時の計算コストが単一のモデルと比べて高くなる．アンサンブル推論の計算コストの問題を解決するために，アンサンブル推論による確率分布を知識として 1 つのモデルを学習する手法が提案されている．これにより，アンサンブル推論と同等の精度を発揮する 1 つのモデルを獲得できる．しかし，これらの手法はモデルの個性を考慮しておらず，異なる知識を持つ複数モデルから知識を転移する際には，知識の転移が困難になる可能性がある．

本節では，多様な知識転移を用いたアンサンブル推論のための共同学習を提案する．1 つの生徒モデルの精度を高くするために，モデルの組み合わせや知識転移戦略の設計を行なってきた従来法とは異なり，アンサンブル精度を高くするためにグループ単位の学習設計を行う．まず，確率分布とアテンションマップを知識として使用し，知識を近づける損失と知識を離す損失を設計する．加えて，損失値に重み付けを行うゲート関数を導入し，知識の種類，知識の転移戦略，ゲート関数を組み合わせることで，多様な共同学習戦略を表現可能とする．ただし，アンサンブル学習における知識転移戦略の最適な組み合わせは，データセットやモデル数によって異なる可能性がある．そこで，知識転移グラフをグループ単位の学習に拡張することでアンサンブル学習に適用し，アンサンブル精

度を指標としてグラフ構造を探索することで，共同学習を用いたアンサンブル学習を自動設計する．最後に，各モデルの個性を考慮した軽量化のための知識転移手法を設計し，自動設計したアンサンブル学習により獲得した多様性のある複数モデルの知識を単一のモデルに転移することで，アンサンブル推論の計算コストの問題を解決する．本節では，4.3.1 項でアンサンブル学習のための多様な知識転移，4.3.2 項でアンサンブル学習のためのグラフ表現，4.3.3 項で多様な知識を持つアンサンブルの軽量化について述べる．

4.3.1 アンサンブル学習のための多様な知識転移

アンサンブル推論のためにモデル間の知識に多様性を促進する知識転移損失を設計する．具体的には，知識として確率分布とアテンションマップを利用し，知識を近づける損失と離す損失を設計する．このとき，知識の転移先をターゲットモデル t，知識の元をソースモデル s とする．4.2.3 項における表 4.4 と表 4.5 の傾向に基づき，知識転移を最小化問題として学習を行うために，知識を近づける損失と知識を離す損失で異なる距離指標を使用する．

■ 確率分布を知識とした知識転移損失

確率分布を近づける場合は KL divergence，離す場合はコサイン類似度を使用する．KL divergence を用いた損失関数は次のように定義する．

$$L_p(\boldsymbol{x}) = \sum_{c=1}^{C} p_s^c(\boldsymbol{x}) \log \frac{p_s^c(\boldsymbol{x})}{p_t^c(\boldsymbol{x})} \tag{4.6}$$

ここで，$\boldsymbol{x}$ は入力データ，C はクラス数，$p_s^c(\cdot)$ はソースモデルのクラス c に対する確率値，$p_t^c(\cdot)$ はターゲットモデルのクラス c に対する確率値である．コサイン類似度を用いた損失関数は次のように定義する．

$$L_p(\boldsymbol{x}) = \frac{\boldsymbol{p}_s(\boldsymbol{x})}{\parallel \boldsymbol{p}_s(\boldsymbol{x}) \parallel_2} \cdot \frac{\boldsymbol{p}_t(\boldsymbol{x})}{\parallel \boldsymbol{p}_t(\boldsymbol{x}) \parallel_2} \tag{4.7}$$

ここで，$\boldsymbol{p}_s(\cdot)$ はソースモデルの確率分布，$\boldsymbol{p}_t(\cdot)$ はターゲットモデルの確率分布である．

■ アテンションマップを知識とした知識転移損失

アテンションマップは，入力データ内で学習に有効な領域に強く反応する．対象物体の大きさはデータごとに異なるため，対象物体の異なる箇所に強く反応しているのに関わらず，類似度が高くなる場合がある．そこで，アテンションマップのクロップを行う．ソースモデルのアテンションマップにおいて最も値の高い位置を中心に正方形のクロップを行い，ターゲットモデルはソースモデルと同じ位置をクロップする．複数のサイズでクロップを行い，それぞれのサイズにおける損失値の

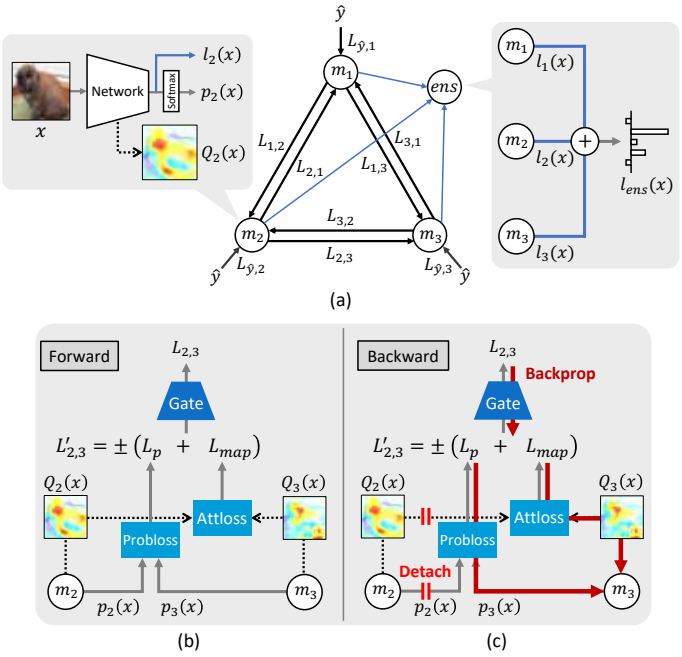


図 4.4: グラフ表現を導入した多様な共同学習を用いたアンサンブル学習

平均を最終的な損失値とする．アテンションマップを近づける場合は平均二乗誤差，離す場合はコサイン類似度を利用する．平均二乗誤差を用いた損失関数は次のように定義する．

$$L_{\mathrm{map}}(\boldsymbol{x}) = \frac{1}{K}\sum_{k=1}^{K}\left(\frac{Q_s^k(\boldsymbol{x})}{\| Q_s^k(\boldsymbol{x}) \|_2} - \frac{Q_t^k(\boldsymbol{x})}{\| Q_t^k(\boldsymbol{x}) \|_2}\right)^2 \tag{4.8}$$

ここで，K はクロップ数，$Q_s^k(\cdot)$ はクロップしたソースモデルのアテンションマップ，$Q_t^k(\cdot)$ はクロップしたターゲットモデルのアテンションマップである．コサイン類似度を用いた損失関数は次のように定義する．

$$L_{\mathrm{map}}(\boldsymbol{x}) = \frac{1}{K}\sum_{k=1}^{K}\frac{Q_s^k(\boldsymbol{x})}{\| Q_s^k(\boldsymbol{x}) \|_2} \cdot \frac{Q_t^k(\boldsymbol{x})}{\| Q_t^k(\boldsymbol{x}) \|_2} \tag{4.9}$$

4.3.2 アンサンブル学習のためのグラフ表現

知識転移グラフをグループ単位の学習に拡張することでアンサンブル学習に適用する．知識転移グラフにおけるアンサンブル学習のグラフ表現を図 4.4a に示す．グラフはノードとエッジで構成される．ノードは，モデルを表すモデルノードとアンサンブルを行うアンサンブルノードを定義する．エッジは，損失計算を表し，モデルノード間のエッジは共同学習，モデルノードとラベル $\hat{y}$ 間のエッジは教師ラベルを用いた交差エントロピー損失を表す．

■ モデルノードとアンサンブルノード

図 4.4a にモデルノードとアンサンブルノードの処理プロセスを示す．モデルノードでは，ノードに割り当てたモデル m_t にデータ x を入力し，確率分布 $p_t(x)$，ロジット $l_t(x)$，アテンションマップ $Q_t(x)$ を取得する．アンサンブルノードでは，すべてのモデルノードのロジットを用いてロジットの平均を計算する．アンサンブルノードにおける処理は次のように定義する．

$$l_{\text{ens}} = \frac{1}{M} \sum_{m=1}^{M} l_m(\boldsymbol{x}) \tag{4.10}$$

ここで M はモデルノードの数，$l_m(\cdot)$ はモデルノードの logit，x は入力データである．

■ ノード間の共同学習

モデルノード間のエッジはノード間の共同学習を行う．図 4.4b と図 4.4c にノード m_2 からノード m_3 へのエッジにおける損失計算の処理過程を示す．エッジではまず図 4.4b のように確率分布とアテンションマップの共同学習の損失計算を行う．共同学習の損失計算は次のように定義する．

$$L_{s,t}^{'}(\boldsymbol{x}) = L_p(\boldsymbol{x}) + L_{map}(\boldsymbol{x}) \tag{4.11}$$

その後，ゲート関数に適用することで，最終的な共同学習の損失となる．ゲート関数に適用した共同学習の損失は次のように定義する．

$$L_{s,t} = \frac{1}{N} \sum_{n=1}^{N} G_{s,t}(L_{s,t}^{'}(\boldsymbol{x}_n)) \tag{4.12}$$

ここで，N はミニバッチのデータ数，$G_{s,t}(\cdot)$ はゲート関数である．共同学習の損失から計算された勾配は，エッジの向きによって勾配を伝えるモデルが変化する．ノード m_2 からノード m_3 へのエッジにおける勾配の流れを図 4.4c に示す．この場合では，ターゲットモデルであるノード m_3 にのみ勾配が伝わるように，ソースモデルであるノード m_2 の計算グラフをカットして勾配を伝えることで，ノード m_2 からノード m_3 への知識転移が行われる．

■ モデルノードの最終的な損失

モデルノードの最終的な損失は，モデルノード間のエッジとモデルノードとラベル間のエッジで計算された損失の総和となる．モデルノードとラベル間のエッジでは，教師ラベルを用いた交差エントロピー損失が計算され，次のように定義する．

$$L_{\text{CE}}(\hat{y}, p_t(\boldsymbol{x})) = -\sum_{c}^{C} \hat{y}^c \log p_t^c(\boldsymbol{x}) \tag{4.13}$$

$$L_{\text{hard},t} = \frac{1}{N} \sum_{n=1}^{N} G_{\text{hard},t}(L_{\text{CE}}(\hat{y}_n, p_t(\boldsymbol{x}_n))) \tag{4.14}$$

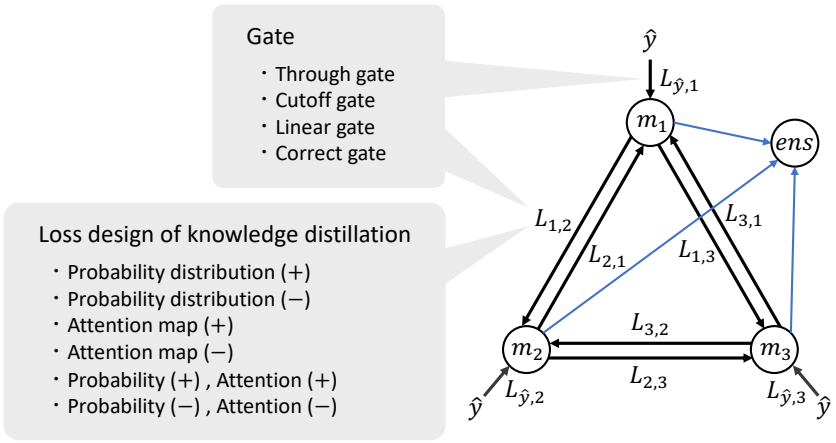


図 4.5: アンサンブル学習のためのグラフ構造におけるハイパーパラメータ

ここで，$\hat{y}$ は教師ラベル，$p_t(\cdot)$ はモデルノードの出力分布，$\hat{y}^c$ は教師ラベルにおけるクラス c の値，p_t^c はモデルノードにおけるクラス c の確率値，$G_{hard,t}$ はゲート関数である．最終的なモデルノードの損失は次のように定義する．

$$L_t = L_{\text{hard},t} + \sum_{s=1,s\neq t}^{M} L_{s,t} \tag{4.15}$$

■ グラフ構造の探索

図 4.5 にグラフ構造のハイパーパラメータを示す．グラフのハイパーパラメータは，モデルノード間のエッジの損失設計，各エッジの損失値に適用されるゲート関数である．損失設計は，確率分布を近づける (式 4.6)，確率分布を離す (式 4.7)，アテンションマップを近づける (式 4.8)，アテンションマップを離す (式 4.9)，確率分布とアテンションマップを同時に近づける (式 4.6+ 式 4.8)，確率分布とアテンションマップを同時に離す (式 4.7+ 式 4.9) の計 6 種類とする．ゲート関数は，Through Gate，Cutoff Gate，Correct Gate，Linear Gate の計 4 種類とする．この時，モデルノードのモデルは，事前に決めた 1 種類に固定する．

グラフ構造の探索方法として，ランダムサーチと Asynchronous successive halving algorithm (ASHA) [96] を使用する．まず初めに，知識転移グラフのハイパーパラメータの組み合わせをランダムに選択し，各モデルの重みパラメータを初期化する．選択したグラフ構造に基づいた共同学習でモデルを学習している際に，$1, 2, 4, 8 \cdots 2^k$ エポックのタイミングでテストデータに対するアンサンブル精度を計算する．もしアンサンブル精度が同じエポックタイミングで以前に評価したグラフ構造のアンサンブル精度の中央値よりも高い場合，学習は次の $2k$ エポックまで継続され，アンサンブル精度を再度評価する．対照的にアンサンブル精度が中央値より低い場合，以前に評価されたグラフ構造よりも高いアンサンブル精度を達成する見込みがないと仮定し，そのグラフ構造に基づいたモデルの学習を終了する．学習が完了または終了した後，新しいグラフ構造をランダムに選択し，同様の

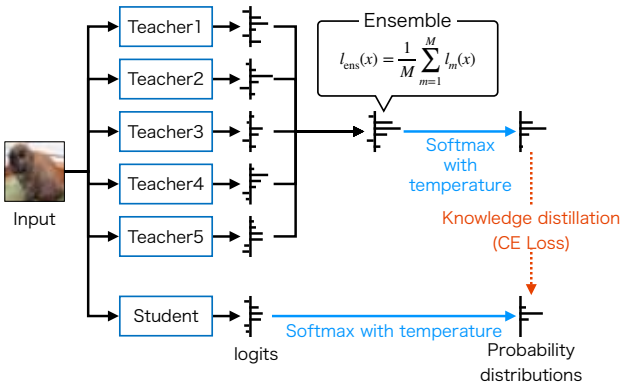


図 4.6: アンサンブル推論を対象とした知識転移による軽量化の従来法

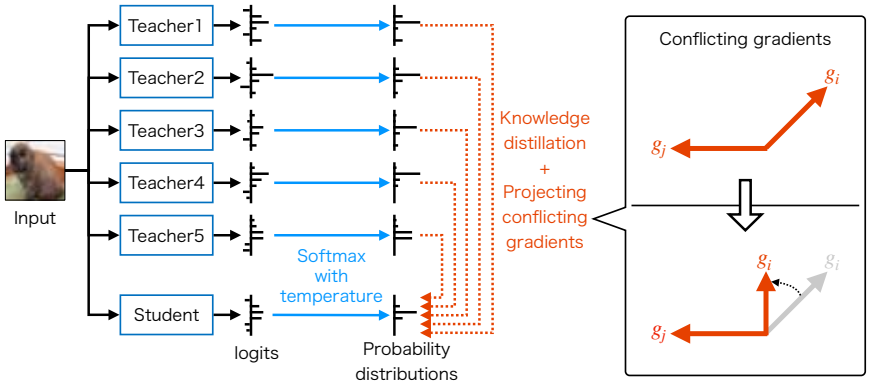


図 4.7: モデル間の多様性を考慮した複数モデルからの知識転移

プロセスを行う．評価済みのグラフ構造の数が事前に設定した数に達した場合，グラフの探索を終了する．そして，評価した中でアンサンブル精度が最も高いグラフ構造を探索結果とする．この探索プロセスは，アンサンブル精度と評価済みの組み合わせの数をサーバー間で共有することで，並列探索が可能である．

4.3.3 多様な知識を持つアンサンブルの軽量化

従来のアンサンブルを用いた知識転移によるモデルの軽量化は，図 4.6 のようにアンサンブルした確率分布を知識として 1 つの生徒モデルを学習する．しかし，この方法は教師モデル間の多様性を考慮しておらず，アンサンブルによって 1 つの確率分布に集約することで各教師モデル特有の知識が消失する可能性がある．そこで，多様性のある教師モデルを用いたモデル圧縮として，図 4.7 のように各教師モデルから個別に知識転移を行う．一方で，各教師モデルから個別に知識転移を行う場合，各教師モデルが異なる領域に注目しているなどの理由から勾配の衝突が起こり，学習が停滞する可能性がある．そこで，各知識転移の損失から計算した勾配に対して勾配集約 [127] を適用する．勾配集約は，2 つの勾配ベクトル間で内積を計算し，内積が負の場合に勾配衝突が起こっていると判

定する．そして，一方の勾配ベクトルからもう片方の勾配ベクトルと衝突している要素を削除することで，勾配衝突を解消する．これにより，複数の教師モデルからの知識転移において，各モデルの多様な知識を有効に活用することを可能とする．

モデル軽量化のための知識転移では，知識として確率分布とアテンションマップを使用する．確率分布の知識転移損失は，クロスエントロピー損失を使用する．知識として使用する確率分布には温度付きソフトマックス関数 [16] を適用する．温度パラメータは，1，2，5，10 の 4 つの中からハイパーパラメータサーチにより決定する．アテンションマップの知識転移損失は，Mean squared error を使用する．

4.4 評価実験

評価実験として，グラフ構造の探索により獲得した知識転移グラフを用いて学習した複数モデルのアンサンブル精度を評価する．4.4.1 項では，評価実験に使用した実験設定について述べる．4.4.2 項では，探索により獲得した知識転移グラフのグラフ構造を示す．4.4.3 項では，クラス分類タスクにおいて従来法と精度を比較する．4.4.4 項では，探索時と異なるデータセットへのグラフ構造の転移性を評価する．4.4.5 項では，学習後のモデルとアンサンブル精度の関係を調査する．4.4.6 項では，複数の教師モデルが持つ知識に多様性がある場合における知識転移によるアンサンブル推論の軽量化について評価を行う．

4.4.1 実験条件

本項では．評価実験に使用したデータセット，モデル，モデルの学習条件，グラフ構造の探索条件について述べる．

■ データセット

データセットは，Stanford Dogs [125]，Stanford Cars [101]，Caltech-UCSD Birds-200-2011 (CUB-200-2011) [126]，CIFAR-10 [97]，CIFAR-100 [97] を使用する．Stanford Dogs，Stanford Cars，CUB-200-2011 は，それぞれ 120 クラスの犬種，196 クラスの自動車の車種，200 クラスの鳥類で構成される．CIFAR-10 と CIFAR-100 は，それぞれ 10 クラスの一般物体，100 クラスの一般物体で構成される．

■ モデル

モデルは，ResNet [10] と ResNet をベースモデルとした Attention branch network (ABN) [124] を使用する．データセットが CIFAR-10，CIFAR-100 のとき ResNet20 を使用し，Stanford Dogs，Stanford Cars，CUB-200-2011 のとき ResNet18 を使用する．ResNet のアテンションマップは，ResNet におけ

る最後の ResBlock の出力から Attention Transfer [3] によって作成する．ABN は，モデル内に Class Activation Map [128] に基づいたアテンションマップの作成とアテンションマップを Attention 機構に応用する仕組みを持ったモデルモデルである．そのため，アテンションマップを用いた共同学習は，ABN が作成したアテンションマップを使用する．また，Attention 機構によってアテンションマップに基づいて特徴マップに対して重み付けが行われるため，アテンションマップの変化がより出力に伝わると考えられる．本研究では，アテンションマップの影響をより特徴マップに伝えるために，ABN の Attention 機構においてバイパス処理は行わない．

■ モデルの学習条件と評価条件

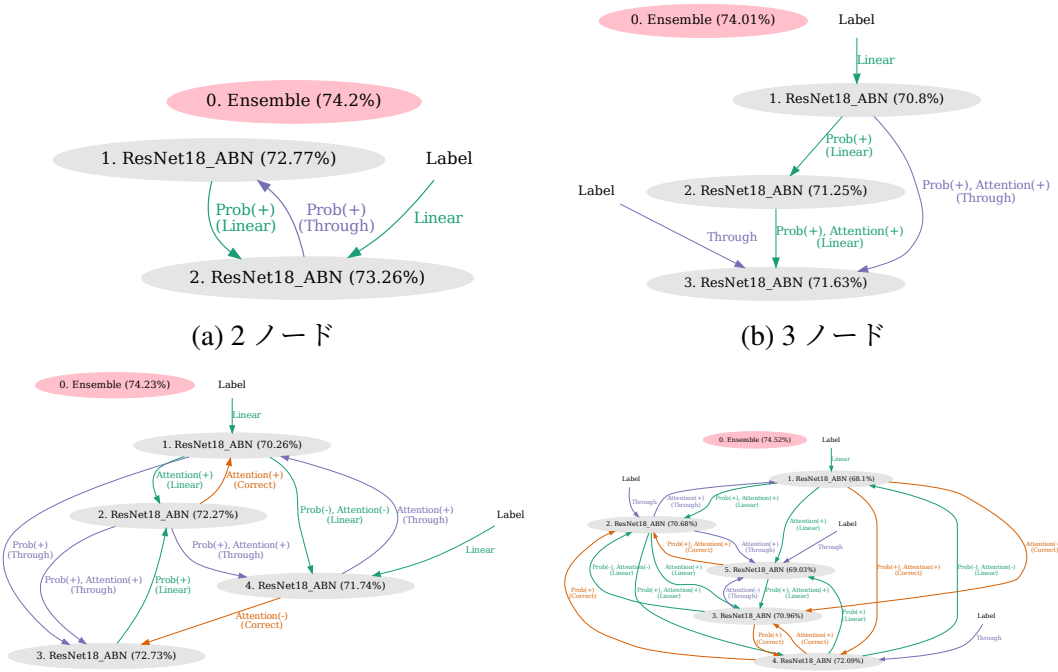
最適化方法は Momentum SGD，初期学習率は 0.1，Momentum の係数は 0.9，weight decay は 0.0001，ミニバッチサイズは 16，学習回数は 300 epoch とし，150 epoch と 225 epoch のタイミングで学習率を 0.1 倍する．データ増幅は，データセットによって異なる設定を使用する．CIFAR-10，CIFAR-100 のとき，学習用データに対してランダムな左右反転と平行移動を行う．平行移動は，画像周辺に 4 ピクセルのパディング（reflection padding）を行い，ランダムに 32×32 の画像サイズでクロップすることで実行する．また，評価用データに対してはデータ増幅を行わない．Stanford Dog，Stanford Cars，CUB-200-2011 のとき，学習用データに対してランダムなクロップ，224×224 の画像サイズへのリサイズ，ランダムな左右反転を行い，評価用データに対してアスペクト比を固定して画像の短辺が 256 ピクセルとなる画像サイズへのリサイズ，画像中央に対して 224×224 の画像サイズのクロップを行う．アテンションマップのクロップは，ResNet と ABN で異なる設定を使用する．ResNet の場合は，3×3，5×5，7×7 の 3 つのサイズでクロップする．ABN の場合は，3×3，7×7，11×11 の 3 つのサイズでクロップする．各結果における精度は，重みパラメータの初期値を変えて学習した計 5 回の平均値を報告する．

■ グラフの探索条件

データセットのテストデータへの過剰な適合を防ぐために，グラフ構造の探索時はデータセットの学習データをグラフ探索用の学習データとグラフ探索用の検証データに分割し，データセットに用意されている本来のテストデータは使用しない．グラフ構造の探索時の指標であるアンサンブルノードの精度は，グラフ探索用の検証データに対する精度を使用する．探索により獲得した知識転移グラフを用いてモデルの学習と評価をする際には，データセットで用意されている元々の学習データとテストデータを使用する．

CIFAR-10 と CIFAR-100 データセットを用いて探索する場合は，50,000 サンプルの学習データをグラフ探索用の学習データとして 40000 サンプル，グラフ探索用の検証データとして 10,000 サンプルに分割する．その他のデータセットを用いて探索する場合は，学習用データの半分をグラフ探索用の学習データ，残り半分のデータをグラフ探索用の検証データとして使用する．

探索時の GPU サーバーとして，90 台の Quadro P5000 サーバを使用する．



(a) 2 ノード　(b) 3 ノード

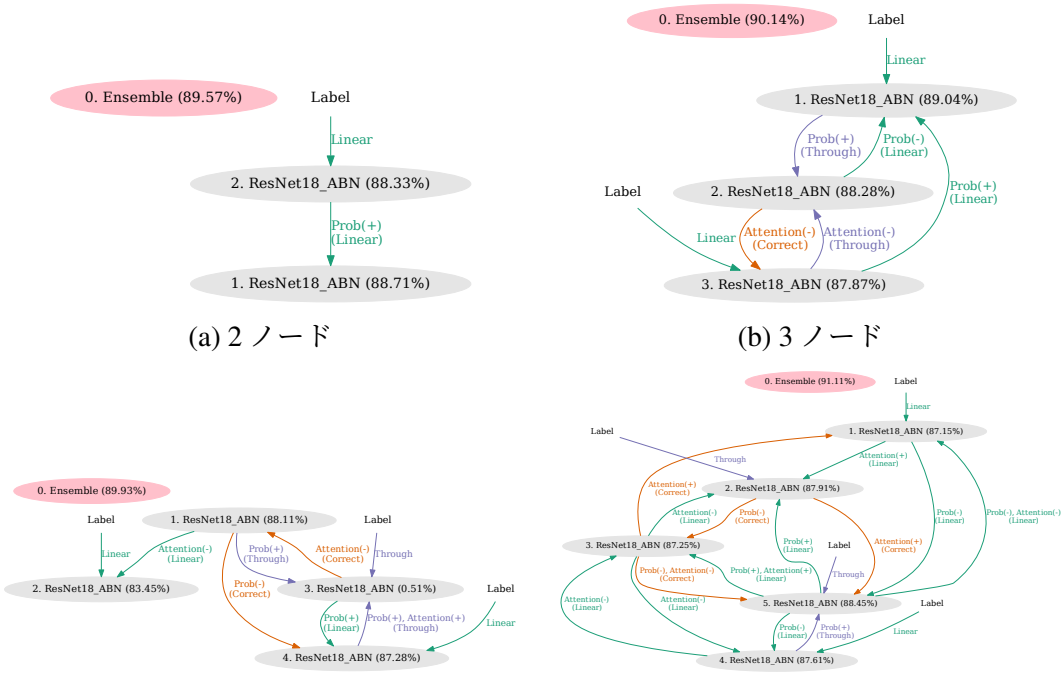
(c) 4 ノード　(d) 5 ノード

図 4.8: StanfordDogs データセットにおける探索結果

4.4.2　探索後のグラフ構造の可視化

Stanford Dogs で探索したグラフを図 4.8，Stanford Cars で探索したグラフを図 4.9，CUB-200-2011 で探索したグラフを図 4.10，CIFAR で探索したグラフを図 4.11 に示す．赤色のノードはアンサンブルノード，灰色のノードはモデルノードを表す．各エッジには選択された損失設計と Gate の種類を示し，エッジの色は選択された Gate の種類を表している．このとき，Cutoff gate は損失計算を行わない Gate のため，Cutoff gate が選択されたエッジはグラフから削除してグラフを可視化した．各ノードにおける括弧内の数値は，探索に用いたデータセットで行ったモデルの重みパラメータの初期値を変えて 5 回行った評価のうちの 1 回の精度である．

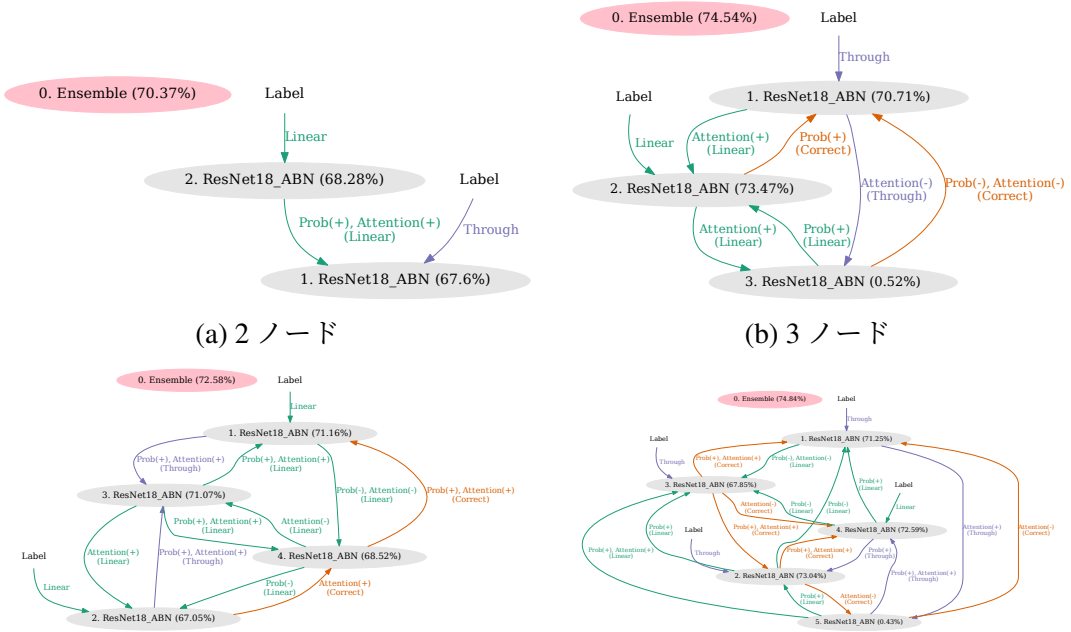
各データセットにおけるグラフ構造を見ると，モデルノードの数が 2 つのグラフでは知識を近づける設計のみが選択され，モデルノードの数が 3 つ以上のグラフでは知識を近づける設計と離す設計を組み合わせた構造を獲得した．Stanford Dogs を用いて探索したグラフに着目すると，モデルノードの数が 2 つのグラフは，双方向の共同学習である DML [9] を拡張したような学習方法を獲得した．モデルノードの数が 3 つのグラフは，ノード 1 とノード 3 の間では一方向の共同学習である KD [16]，ノード 1 からノード 2 とノード 2 からノード 3 では段階的な共同学習である Teacher Assistant [18] を行う複数の共同学習手法を組み合わせた学習方法を獲得した．モデルノードの数が 4 つのグラフと 5 つのグラフは，知識を近づける学習と離す学習の両方が選択され，人手で意図的に設計することが困難な複雑な学習方法を獲得した．



(a) 2 ノード　(b) 3 ノード

(c) 4 ノード　(d) 5 ノード

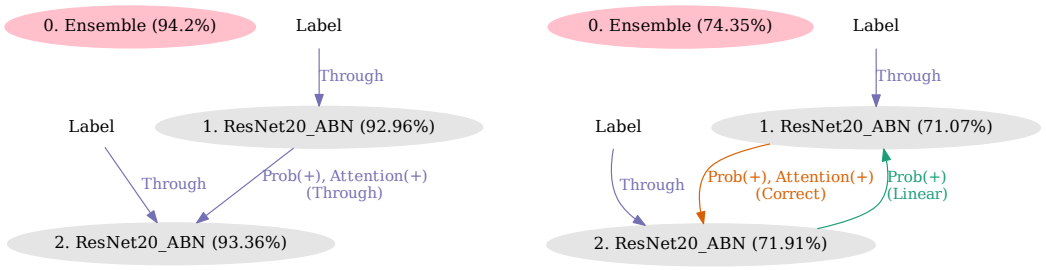
図 4.9: Stanford Cars データセットにおける探索結果



(a) 2 ノード　(b) 3 ノード

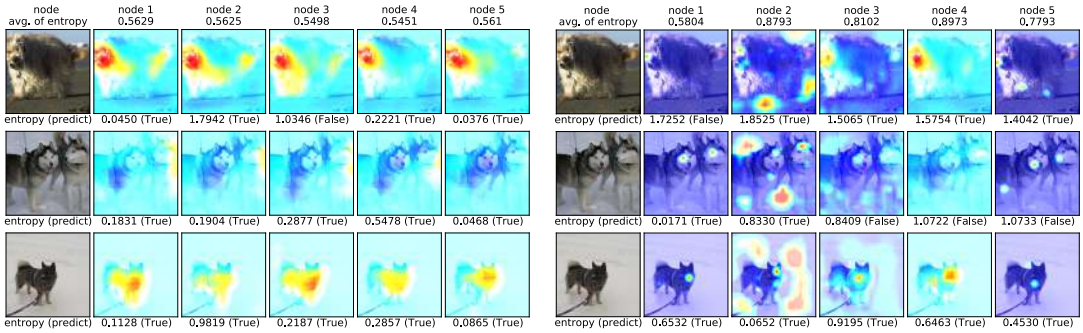
(c) 4 ノード　(d) 5 ノード

図 4.10: CUB-200-2011 データセットにおける探索結果



(a) CIFAR-10 (b) CIFAR-100

図 4.11: CIFAR データセットにおける 2 ノードの探索結果



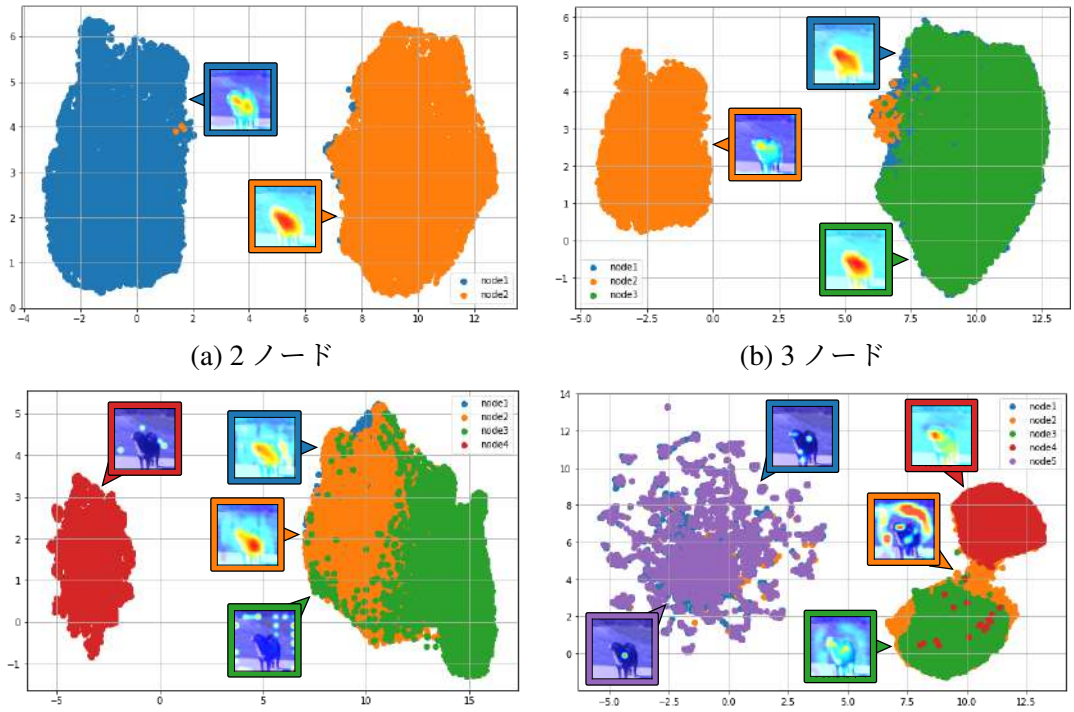
(a) Independent (b) 探索したグラフ

図 4.12: 5 つのモデルを用いた学習におけるアテンションマップの可視化

図 4.12b に図 3.7d のグラフで学習した ABN のアテンションマップを示す．各ノードは異なる注目領域を持っている．平均エントロピーに着目すると，犬の頭の一点に焦点を当てているノード 1 と 5 はエントロピーが低い．画像全体や背景に焦点を当てているノード 2，3，および 4 は，ノード 1 および 5 よりも高いエントロピーを持っている．これは，異なる場所の重要性に基づいて推論が行われ，アテンションマップの状態が確率分布に影響を与えることを意味している．

図 4.12a に個別に学習されたモデルを使用するアンサンブルにおけるアテンションマップを示す．最適化されたグラフと比較して，個別に学習されたモデルを使用するアンサンブルの平均エントロピーは低い．これは，個別に学習されているにもかかわらず，モデル間の注目領域がほぼ同じであるためである．

図 4.13 に図 4.8 の各グラフで学習した ABN のアテンションマップを示す．各モデルの Stanford Dogs の全てのテストデータに対するアテンションマップを取得し，UAMP [129] により 2 次元に圧縮して可視化した．各グラフにおけるアテンションマップの傾向は一貫している．そのため，各グラフは探索により異なる学習戦略が選択されたが，アンサンブルを構成する各モデルは異なるアテンションマップの獲得が促されたと言える．



(a) 2 ノード　(b) 3 ノード

(c) 4 ノード　(d) 5 ノード

図 4.13: UMAP によるアテンションマップの可視化

4.4.3　従来法との精度比較

Stanford Dogs データセットにおけノードの平均精度，アンサンブル精度，アテンションマップの相違度を表 4.6 に示す．アテンションマップの相違度は，データごとに各モデルのアテンションマップ間で差の総和を計算し，各データのアテンションマップの相違度を平均することで計算した．“Ours” は最適化されたグラフの結果であり，“Independent” は個別に学習されたモデルの結果，“DML” は DML [9] を使用したモデルの結果，“ONE” は ONE [93] を使用したマルチブランチモデルの結果である．“ONE(B × 2)” は 2 ブランチモデルの結果である．また，“ONE” はマルチブランチによるシングルモデルのため，アテンションマップの相違度を計算しなかった．“Ours” のアンサンブル精度は，“Independent”，“DML”，“ONE” と比べて高い．“Independent” と “DML” を比較すると，ノードの精度向上に比べてアンサンブル精度の向上は小さい．“Ours” では，“DML” と比較してモデルの精度が向上するとともにアンサンブル精度も向上した．アテンションマップの相違度において，“Ours” は “Independent” と “DML” と比較して，高い相違度となった．したがって，“Ours” は学習によって多様性を獲得することで高いアンサンブル精度を発揮するグラフを得たと言える．

表 4.6: Stanford Dogs における従来法との比較 [%]

Method	Node	Accuracy [%]		Absolute Error of Attention Map
		Node	Ensemble	
Independent	ResNet18 × 2	65.31 ± 0.16	68.19 ± 0.20	**14.32 ± 0.30**
DML [9]	ResNet18 × 2	67.55 ± 0.27	68.86 ± 0.25	9.37 ± 0.22
ONE(B×2) [93]	ResNet18 × 1	67.96 ± 0.41	68.53 ± 0.39	-
Ours	ResNet18 × 2	**71.38 ± 0.08**	**72.41 ± 0.20**	5.82 ± 0.08
Independent	ResNet18 × 3	65.08 ± 0.23	68.64 ± 0.38	**14.14 ± 0.23**
DML [9]	ResNet18 × 3	68.66 ± 0.34	69.95 ± 0.39	7.90 ± 0.02
ONE(B×3) [93]	ResNet18 × 1	68.49 ± 0.60	68.94 ± 0.56	-
Ours	ResNet18 × 3	**69.58 ± 0.15**	**71.87 ± 0.33**	11.48 ± 0.29
Independent	ResNet18 × 4	65.29 ± 0.35	69.27 ± 0.49	14.35 ± 0.25
DML [9]	ResNet18 × 4	68.83 ± 0.44	69.95 ± 0.58	7.13 ± 0.12
ONE(B×4) [93]	ResNet18 × 1	68.48 ± 0.32	68.85 ± 0.37	-
Ours	ResNet18 × 4	**70.34 ± 0.12**	**72.71 ± 0.13**	**41.99 ± 4.17**
Independent	ResNet18 × 5	65.00 ± 0.24	69.47 ± 0.13	14.23 ± 0.24
DML [9]	ResNet18 × 5	**68.77 ± 0.17**	69.94 ± 0.20	6.44 ± 0.10
ONE(B×5) [93]	ResNet18 × 1	68.51 ± 0.18	68.95 ± 0.24	-
Ours	ResNet18 × 5	52.28 ± 0.87	**71.35 ± 0.48**	**59.99 ± 11.12**
Independent	ABN × 2	68.13 ± 0.16	70.90 ± 0.19	11.46 ± 0.54
DML [9]	ABN × 2	69.91 ± 0.46	71.45 ± 0.52	9.49 ± 0.21
ONE(B×2) [93]	ABN × 1	69.38 ± 0.38	69.81 ± 0.33	-
Ours	ABN × 2	**72.77 ± 0.23**	**73.86 ± 0.26**	**30.56 ± 1.30**
Independent	ABN × 3	68.04 ± 0.28	71.41 ± 0.34	10.82 ± 0.86
DML [9]	ABN × 3	70.50 ± 0.26	72.08 ± 0.42	8.73 ± 0.37
ONE(B×3) [93]	ABN × 1	69.96 ± 0.47	70.44 ± 0.44	-
Ours	ABN × 3	**70.95 ± 0.16**	**73.41 ± 0.30**	**26.94 ± 0.90**
Independent	ABN × 4	68.30 ± 0.27	72.06 ± 0.53	10.83 ± 0.25
DML [9]	ABN × 4	**71.50 ± 0.31**	72.87 ± 0.29	8.53 ± 0.23
ONE(B×4) [93]	ABN × 1	70.16 ± 0.47	70.54 ± 0.54	-
Ours	ABN × 4	71.46 ± 0.22	**74.16 ± 0.22**	**41.25 ± 1.28**
Independent	ABN × 5	68.24 ± 0.26	72.32 ± 0.18	11.01 ± 0.24
DML [9]	ABN × 5	**71.15 ± 0.28**	72.50 ± 0.16	8.22 ± 0.20
ONE(B×5) [93]	ABN × 1	70.59 ± 0.28	70.89 ± 0.14	-
Ours	ABN × 5	70.23 ± 0.33	**74.14 ± 0.50**	**40.09 ± 5.97**

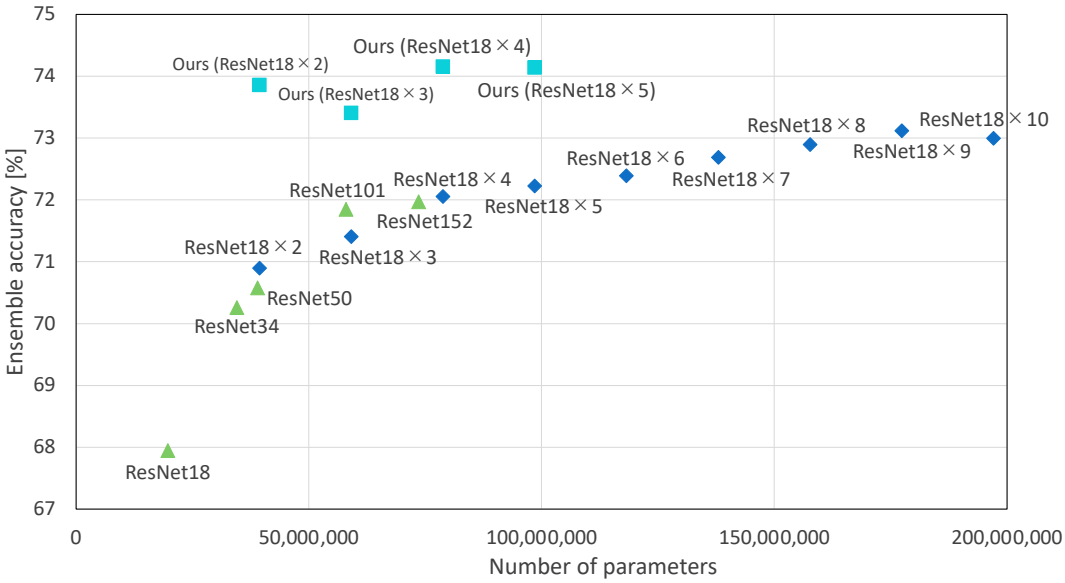


図 4.14: パラメータ数による精度比較

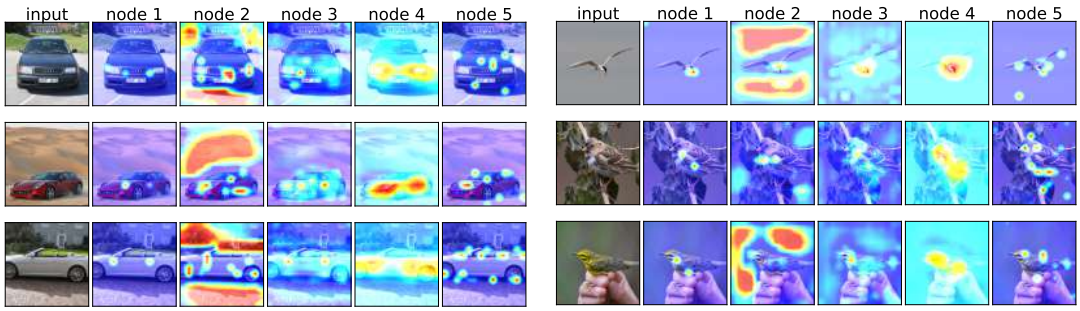
パラメータ数による精度比較を図 4.14 に示す．各点は，モデル数やベースモデルの異なる ABN による精度を表している．青色の四角形は，ベースモデルを ResNet18，モデル数を 2 から 10 とし，通常のアンサンブル学習によるアンサンブル精度を表す．緑色の三角形は，モデル数を 1 とし，ベースモデルを ResNet18 から ResNet152 へと層数を増やした場合のモデルの精度を表す．水色の四角形は，ベースモデルを ResNet18，モデル数を 2 から 4 とし，Stanford Dogs を用いて最適化したグラフによるアンサンブル精度を表す．青色の四角形に着目すると，モデル数を増やすことでアンサンブル精度が向上し，モデル数が 8 以降は精度が向上していない．緑色の三角形に着目すると，層数を増やすことでモデルの精度が向上し，パラメータ数の観点からはアンサンブル学習におけるモデル数を増やした場合と同程度の精度向上にあると言える．提案手法である水色の四角に着目すると，通常のアンサンブル学習の頭打ちを超えるアンサンブル精度を達成し，単一のモデルや通常のアンサンブル学習と比べて，パラメータ効率の良いアンサンブル学習方法を獲得できたと言える．

4.4.4 グラフ構造の汎化性

各データセットにおいてモデルノードの数が 2 つのグラフについて探索を行い，獲得したグラフをさまざまなデータセットで評価する．探索により得られた各グラフのアンサンブル精度を表 4.7 に示す．“Hyperparameter Search” は，グラフの探索に用いたデータセットを表す．“Training & Evaluation” は，モデルの学習と評価に用いたデータセットを表す．各データセット名における (F) は詳細画像識別のデータセット，(G) は一般物体認識のデータセットであることを示している．詳細画像識別のデータセットである Stanford Dogs，CUB-200-2011，Stanford Cars では，詳細画像識別のデータセットで最適化したグラフが従来法と比べて高い精度を達成した．一方で，一般物体認識のデータセット

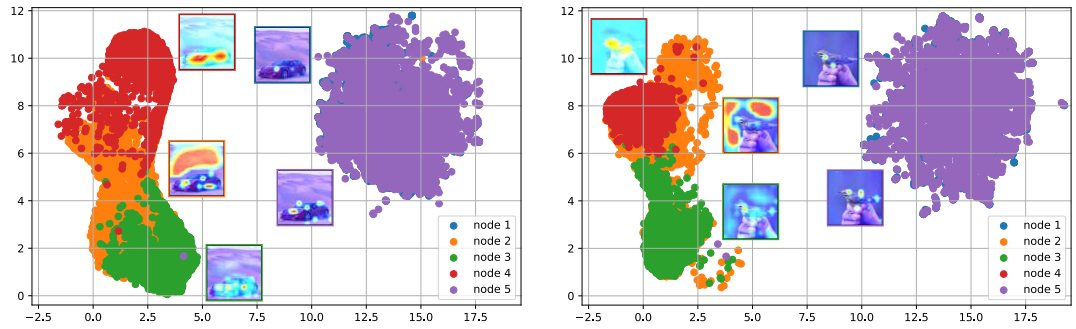
表 4.7: 探索時と異なるデータセットに対するグラフの汎化性 [%]

Method	Hyperparameter Search	Training & Evaluation	Ensemble Acc. [%]
Independent	-	Stanford Dogs (F)	70.90
DML [9]	-	Stanford Dogs (F)	71.45
Ours (Fig. 3.7a)	Stanford Dogs (F)	Stanford Dogs (F)	**73.86**
Ours	CUB-200-2011 (F)	Stanford Dogs (F)	**72.43**
Ours	Stanford Cars (F)	Stanford Dogs (F)	**72.79**
Ours	CIFAR-100 (G)	Stanford Dogs (F)	**71.53**
Ours	CIFAR-10 (G)	Stanford Dogs (F)	70.08
Independent	-	CUB-200-2011 (F)	65.26
DML [9]	-	CUB-200-2011 (F)	66.90
Ours (Fig. 3.7a)	Stanford Dogs (F)	CUB-200-2011 (F)	**72.06**
Ours	CUB-200-2011 (F)	CUB-200-2011 (F)	**69.81**
Ours	Stanford Cars (F)	CUB-200-2011 (F)	**71.27**
Ours	CIFAR-100 (G)	CUB-200-2011 (F)	66.43
Ours	CIFAR-10 (G)	CUB-200-2011 (F)	66.42
Independent	-	Stanford Cars (F)	88.49
DML [9]	-	Stanford Cars (F)	88.89
Ours (Fig. 3.7a)	Stanford Dogs (F)	Stanford Cars (F)	**89.76**
Ours	CUB-200-2011 (F)	Stanford Cars (F)	**89.50**
Ours	Stanford Cars (F)	Stanford Cars (F)	**89.44**
Ours	CIFAR-100 (G)	Stanford Cars (F)	**88.90**
Ours	CIFAR-10 (G)	Stanford Cars (F)	88.63
Independent	-	CIFAR-100 (G)	73.16
DML [9]	-	CIFAR-100 (G)	73.61
Ours (Fig. 3.7a)	Stanford Dogs (F)	CIFAR-100 (G)	72.19
Ours	CUB-200-2011 (F)	CIFAR-100 (G)	**73.66**
Ours	Stanford Cars (F)	CIFAR-100 (G)	72.53
Ours	CIFAR-100 (G)	CIFAR-100 (G)	**74.18**
Ours	CIFAR-10 (G)	CIFAR-100 (G)	73.37
Independent	-	CIFAR-10 (G)	93.99
DML [9]	-	CIFAR-10 (G)	93.97
Ours (Fig. 3.7a)	Stanford Dogs (F)	CIFAR-10 (G)	93.87
Ours	CUB-200-2011 (F)	CIFAR-10 (G)	**94.09**
Ours	Stanford Cars (F)	CIFAR-10 (G)	93.93
Ours	CIFAR-100 (G)	CIFAR-10 (G)	**94.37**
Ours	CIFAR-10 (G)	CIFAR-10 (G)	**94.15**



(a) Stanford Cars (b) CUB-200-2011

図 4.15: 探索時と異なるデータセットで学習した際のアテンションマップ



(a) Stanford Cars (b) CUB200-2011

図 4.16: 索時と異なるデータセットで学習した際の UMAP によるアテンションマップの可視化

である CIFAR-10 と CIFARF-100 では，一般物体認識のデータセットで最適化したグラフが従来法と比べて高い精度を達成する傾向であった．これは，最適化によって問題設定に対応したグラフ構造が得られたことを示しており，問題設定が同じ場合においてグラフ構造には汎用性があると言える．

Stanford Dogs で探索したモデルノードの数が 5 つのグラフ（図 4.8d）を用いて，探索時と異なるデータセットを学習した際の ABN のアテンションマップの可視化を行う．Stanford Cars を学習した際のアテンションマップと CUB-200-2011 を学習した際のアテンションマップを図 4.15 に示す．図 4.15 のアテンションマップと図 4.12 のアテンションマップを比較すると，ノード番号が同じ場合に似たアテンションマップの傾向となっていることがわかる．Stanford Cars と CUB-200-2011 データセットにおける全てのテストデータに対するアテンションマップを UMAP [129] により 2 次元に圧縮して可視化した結果を図 4.16 に示す．図 4.16 の傾向と図 4.13d の傾向を比較すると，ノード番号が同じ場合に似た傾向となっていることがわかる．これらのことから，探索で得られたグラフ構造によって各ノードの注視方法の傾向が促されていると言える．

表 4.8: 使用するノードによるアンサンブル精度の変化 [%]

Used for ensemble					Ensemble
node 1	node 2	node 3	node 4	node 5	
✓	✓	✓	✓	✓	**74.50**
✓	✓	✓	✓		74.30
✓	✓	✓		✓	74.06
✓	✓		✓	✓	74.31
✓		✓	✓	✓	74.23
	✓	✓	✓	✓	**74.44**
✓	✓	✓			73.76
✓	✓		✓		73.80
✓	✓			✓	73.71
✓		✓	✓		73.87
✓		✓		✓	73.28
✓			✓	✓	73.65
	✓	✓	✓		73.51
	✓	✓		✓	**73.88**
	✓		✓	✓	73.40
		✓	✓	✓	73.01
✓	✓				72.54
✓		✓			72.69
✓			✓		**72.74**
✓				✓	72.28
	✓	✓			72.62
	✓		✓		72.69
	✓			✓	72.18
		✓	✓		72.56
		✓		✓	71.52
			✓	✓	72.44

4.4.5　学習後のモデルとアンサンブル精度の関係

Stanford Dogs で探索したモデルノードの数が 5 つのグラフ（図 4.8d）を用いて学習すると，図 4.12b のアテンションマップのように一部のノードが背景に強く注視する．そのため，一部のノードがアンサンブル推論に影響を与えない，または悪影響を与えている可能性がある．そこで，図 4.8d のグラフを用いて各モデルノードの学習を行い，評価時に一部のノードのみを用いてアンサンブル推論を行う．データセットとして Stanford Dogs，アンサンブル推論に使用するモデルノード数を 2 から 5 としたときのアンサンブル精度を表 4.8 に示す．✓ はアンサンブル推論に使用したノードであることを表し，太字のアンサンブル精度は各モデルノード数において最も高いアンサンブル精度であることを表す．各モデルノード数における最も高いアンサンブル精度に着目すると，全てのモデルノードを使用した場合と比べて，アンサンブル精度が低下した．このことから，図 4.8d のグラフは全てのモデルノードが協力することで，高いアンサンブル精度を達成する学習方法であると言える．

4.4.6　アンサンブル推論の軽量化

探索後のグラフで学習した複数モデルの知識を 1 つのモデルへ転移することで，アンサンブル推論の計算コストの削減を行う．データセットとして Stanford Dogs，生徒モデルとして ResNet-18 に基づいた ABN を使用し，図 3.7d のグラフで学習した 5 つの教師モデルによるアンサンブルを教師モデルとして使用する．知識として確率分布とアテンションマップを使用し，アンサンブルした確率分布を用いた知識転移，各教師モデルからの知識転移を比較する．また，知識転移損失の勾配計算時に勾配集約を導入することによる学習効果を評価する．

教師モデルと各方法で学習した生徒モデルの精度を表 4.9 に示す．アンサンブルした確率分布を知識とした場合，生徒モデルの精度は教師モデルのアンサンブル精度と比べて低い．これは，アンサンブルにより各教師モデル特有の知識が消失したことを意味している．一方で，各教師モデルの確率分布を知識とした場合，生徒モデルの精度はアンサンブルした確率分布を知識とした知識転移と比べて高い精度となった．各教師モデルの確率分布を知識とした知識転移に勾配集約を導入した場合，導入しなかった場合と同程度の精度となった．知識としてアテンションマップを導入すると，導入しなかった場合と比べて精度が低下した．知識としてアテンションマップを用いた場合に勾配集約を導入すると，アテンションを有効活用することができ，最も高い精度となった．

各教師モデルの確率分布を知識とした場合の各損失の勾配衝突の回数を表 4.10，各教師モデルのアテンションマップを知識とした場合の各損失の勾配衝突の回数を表 4.11 に示す．確率分布を知識とした場合，勾配衝突が起こっていない．そのため，各教師モデルの確率分布を知識とした知識転移に勾配集約を導入しても精度に変化がなかった．アテンションマップを知識とした場合，損失間で勾配衝突が発生しており，特に教師 1 のアテンションマップと教師 2 のアテンションマップ，教師 2 のアテンションマップと教師 5 のアテンションマップの間で勾配衝突の頻度が高い．教師 1 のアテンションマップは図 4.12b のノード 1，教師 2 のアテンションマップは図 4.12b のノード 2，教師 5 のアテンションマップは図 4.12b のノード 5 である．前景への注目を促す知識転移と背景への注目を

表 4.9: 多様な知識を持つ複数モデルから 1 モデルへの知識転移

Ensemble probability	Network probability	Attention map	Gradient aggregation	Network Acc. [%]
Ensemble accuracy of teacher networks: 74.52 [%]				
✓				69.45 ± 0.30
	✓			73.80 ± 0.15
	✓		✓	73.86 ± 0.22
	✓	✓		71.95 ± 1.89
	✓	✓	✓	**74.00 ± 0.14**

表 4.10: 確率分布を知識とした知識転移における勾配衝突の回数

	Teacher1	Teacher2	Teacher3	Teacher4	Teacher5
Teacher1	-	0	0	0	0
Teacher2	0	-	0	0	0
Teacher3	0	0	-	0	0
Teacher4	0	0	0	-	0
Teacher5	0	0	0	0	-

表 4.11: アテンションマップを知識とした知識転移における勾配衝突の回数

	Teacher1	Teacher2	Teacher3	Teacher4	Teacher5
Teacher1	-	173	25	33	64
Teacher2	173	-	26	26	124
Teacher3	25	26	-	0	26
Teacher4	33	26	0	-	34
Teacher5	64	124	26	34	-

促す知識転移を同時に学習することで，勾配衝突が発生していると考えられる．そのため，勾配衝突を用いなかった場合に学習が不安定となり，精度が低下したと考えられる．

4.5 まとめ

本章では，アンサンブル学習におけるモデルの多様性を促進するための知識転移戦略を提案し，知識転移グラフの探索によってアンサンブル推論に効果的な共同学習を自動設計した．さらに，モデルの多様性から生じる知識の衝突を考慮した，多様な知識を持つ複数の教師モデルを用いたモデル軽量化手法を設計し，アンサンブル推論の計算コストの問題に対応した．5 つのデータセットにおける評価実験から，自動設計された手法は従来法よりも高いアンサンブル精度を達成した．さらに，複数の教師モデルから単一の生徒モデルへのモデル軽量化では，各教師モデルが異なるアテンションマップを持つ場合に勾配の衝突が発生し，知識転移が失敗することを確認した．また，提案したモデル軽量化手法が勾配の衝突を効果的に解消し，多様なアテンションマップを活用したモデル軽量化が可能であることを示した．一方でグラフ構造の探索では，ランダムサーチと ASHA を用いて 6,000 通りの構造を評価したことから，探索コストが高いと言える．また，表現可能なグラフ構造の数が膨大なため，必ずしも探索によって最適な手法が自動設計された保証はない．今後の課題として，知識転移グラフに特化した低コストな探索手法の設計と探索結果を用いた再探索によるグラフ構造探索の改善が挙げられる．具体的には，実験結果から探索したグラフ構造が探索に使用したデータセットと似た問題設定のデータセットにおいて高いアンサンブル精度を発揮したことから，目的の問題設定または似た問題設定の小規模データを用いてグラフ構造の探索を行い，その探索結果を用いて目的の問題設定において大規模データを用いて学習するメタラーニングベースの探索などが考えられる．

第5章

一致性正則化と擬似ラベリングによる半教師あり共同学習

教師あり学習は，モデルが出力する確率分布を人間が作成した正解ラベルと一致させるように学習する．教師あり学習で学習したモデルは，高い精度を達成できる一方で，正解ラベルの作成には人的・時間的コストがかかる．半教師あり学習は，ラベルありデータとラベルなしデータを用いた学習方法である．ラベルなしデータを直接使用することから，データ収集後に追加のコストなしで精度の改善が期待できる．FixMatch [19] は，一致性正則化と擬似ラベリングを組み合わせた半教師あり学習手法であり，最先端の半教師あり学習手法における基盤技術となっている．一致性正則化 [130, 28, 131, 132, 133, 134, 30] は，ラベルなしデータに摂動を加え，それらの摂動に対してモデルが頑健になるように学習する．一方，擬似ラベリング [135, 136, 137, 138] は，ラベルなしデータに対してモデルの出力分布に基づいた擬似ラベルを付与する．擬似ラベルを付与したラベルなしデータは，ラベルありデータと同様に使用できる．FixMatch は，これら 2 つのアプローチを組み合わせ，弱い摂動を与えたデータに対する出力分布から作成した擬似ラベルと，強い摂動を与えたデータに対する出力分布を一致させるようにモデルを学習する．FixMatch 以降の手法は FixMatch を基盤技術として，新しいしきい値戦略の導入 [139, 140, 141, 142]，異なるデータ間の関係の活用 [143, 144, 145, 146]，自己教師あり学習を用いた事前学習の導入 [147, 148, 149, 150, 151] などの様々な改良が行われた．しかし，FixMatch における一致性正則化と擬似ラベリングの組み合わせ方は手動で設計されており，組み合わせ方に関する包括的な調査はまだ行われていない．

そこで本章では，半教師あり学習の代表的な従来法を主要な要素に分解し，知識転移グラフで表現することで，グラフ構造の探索を通して一致性正則化と擬似ラベリングの多様な組み合わせ方を検討する．また，グラフ構造の拡張性から，従来の半教師あり学習を半教師あり共同学習へ容易に拡張可能となる．半教師あり学習を組み込んだ知識転移グラフは，教師あり学習，半教師あり学習，教師なし学習の 3 種類の学習方法を表現可能となる．これらの要素を内包した多様な探索空間におけるグラフ構造の探索によって，一致性正則化と擬似ラベリングの組み合わせ，3 種類の学習方法の連携，共同学習への拡張を考慮した学習方法の自動設計を行い，自動設計された学習方法を分析することで，新たな知見の獲得を目指す．

評価実験では，CIFAR データセットを用いて 4 つの観点から提案手法の評価を行う．1 つ目は，FixMatch に基づいた探索空間において知識転移グラフの探索を行い，探索結果を最先端の半教師あり学習手法と精度比較を行う．半教師あり学習の設定において，一致性正則化と擬似ラベリングの組み合わせ方を精度の観点から評価する．2 つ目は，多様な探索空間における探索空間において知識

転移グラフの探索を行い，探索結果を半教師あり学習の代表的な従来法と精度比較を行う．半教師あり共同学習の設定において，従来法の精度傾向からラベルありデータ数と学習戦略の関係を評価する．3 つ目は，探索により獲得した知識転移グラフのグラフ構造の可視化である．探索により発見した知識転移戦略が従来法とどのように異なるのかを評価する．最後に，探索結果のグラフ構造から得られた知見に基づいてグラフ構造の手動設計を行う．探索により発見した知見が精度に寄与するのかを評価する．

本章の構成は以下の通りである．5.1 節で本章の提案手法に関連する半教師あり学習について述べる．5.2 節で提案手法である一致性正則化と擬似ラベリングによる半教師あり共同学習について述べる．5.3 節で提案手法の評価実験について述べる．最後に，5.4 節で本章についてまとめる．

5.1 関連研究：半教師あり学習

本節では，代表的な半教師あり学習のアプローチである一致性正則化，擬似ラベリング，FixMatch について述べる．

一致性正則化 一致性正則化は，ラベルなしデータを用いた学習により摂動に対する頑健性を獲得する．代表的な手法として Π-model [130] がある．Π-model は，異なる摂動を加えた同じデータの確率分布を一致させるようにモデルを学習する．Π-model を基に，さまざまな一致性正則化手法が提案されている．Temporal Ensembling [130] は，一致性正則化を計算する確率分布の 1 つに，過去の確率分布を用いたアンサンブルを利用することで，計算コストを削減しつつ精度を向上させる．Mean Teacher [28] は，重みパラメータのアンサンブルを行う指数移動平均モデルの出力を利用することで，Temporal Ensembling のような確率分布を用意する．MixMatch [131] は，摂動を加えた K 個のデータの確率分布に対して，アンサンブルと先鋭化処理を適用することで，より高い精度を実現する．Virtual Adversarial Training [132] は，敵対的攻撃（Adversarial attack）を摂動として利用する．Adversarial Transformation [133] は，敵対的摂動をデータ拡張の一部として適用する．Unsupervised Data Augmentation [134] は，RandAugment [152] などの強力なデータ拡張を摂動として利用する．Dual Student [30] は，Π-model で学習された 2 つのモデルに Knowledge Distillation [16] を導入することで，半教師あり学習における知識転移の有効性を示した．Multiple Student [30] は，半教師あり共同学習におけるモデル数を増やす有効性を示した．これらの一致性正則化手法は，正解ラベルを用いた教師あり損失とラベルなしデータを用いた一致性損失の 2 つの損失から構成される損失関数を用いてモデルを学習する．

一方で，一致性損失のような摂動を利用し，ラベルなしデータのみを用いてモデルを事前学習するアプローチとして，自己教師あり学習の対照学習 [153, 154, 155, 156] がある．自己教師あり学習は，ラベルなしデータのみを用いてモデルの事前学習を行う学習方法である．対照学習では，同一データから摂動を付与して作成された 2 つの画像ペアを正例ペア，異なるデータから摂動を付与して作成された 2 つの画像ペアを負例ペアと定義し，正例ペアにおける特徴量の類似度を大きく，負例ペアにおける特徴量の類似度を小さくするように学習する．このとき，摂動として色変換とランダムクロップが使用される．対照学習は，負例ペアの数に応じて損失の計算コストが増加することから，正例ペアのみを用いた手法が提案されている [157, 158, 159]．この正例ペアのみを用いた自己教師あり学習は，一致性正則化と似た学習方法となっており，摂動の種類と損失計算に使用する出力空間のみが異なる．

擬似ラベリング 擬似ラベリングは，ラベルなしデータに対して擬似ラベルを作成し，ラベルありデータと擬似ラベルありデータを用いた教師あり学習によってモデルを学習する．擬似ラベリング手法には，one-hot ラベリング [135, 136] とソフトラベリング [137, 138] の 2 種類がある．one-hot ラベリングは，1 つのクラスが 1，それ以外のクラスが 0 の確率値を持ったラベルを作成する．一方でソフトラベリングは，各クラスが確率値を持ったラベルを作成する．Pseudo-Label [135] は，one-hot ラベリングにおける代表的な従来法である．Pseudo-Label では，最初にラベルありデータを用いて

モデルを学習する．次に，ラベルありデータで学習したモデルにラベルなしデータを入力し，出力確率の中で最も高いクラスを正解クラスとした one-hot 擬似ラベルを作成する．そして，ラベルありデータと擬似ラベルありデータを用いた教師あり学習を行う．Label Propagation [136] は，ラベルありデータとラベルなしデータの類似性を利用し，ラベルありデータのラベル情報をラベルなしデータに伝播して擬似ラベルを付与する．Arazo ら [137] は，ラベルありデータの分布を考慮して，ソフトな擬似ラベルを作成する．Noisy Student[138] は，ラベルありデータで学習した教師モデルの出力分布をソフトな擬似ラベルとして利用し，ラベルありデータと擬似ラベルありデータを用いて生徒モデルを学習する．生徒モデルの学習時に Dropout [160]，RandAugment [152]，Stochastic Depth [161] などの強い摂動を付与することで，高い精度を実現した．

FixMatch FixMatch [19] は，一致性正則化と擬似ラベリングを組み合わせることで高い精度を達成できることを示した手法である．FixMatch は，弱い摂動を加えたデータから作成した擬似ラベルと，強い摂動を加えたデータの確率を一致させるようにモデルを学習する．弱い摂動として左右反転と水平・垂直方向への平行移動を使用し，強い摂動として RandAugment を使用する．擬似ラベルは，最も高い確率値がしきい値を超えた場合にのみ作成される．FixMatch 以降では，FixMatch を基盤技術として，新しいしきい値戦略の導入 [139, 140, 141, 142]，異なるデータ間の関係の活用 [143, 144, 145, 146]，自己教師あり学習を用いた事前学習の導入 [147, 148, 149, 150, 151] などの様々な改良が行われている．これらの最先端の半教師あり学習手法における一致性正則化と擬似ラベリングの組み合わせ方は，FixMatch の方法を採用している．しかし，FixMatch における一致性正則化と擬似ラベリングの組み合わせ方は，人手によって設計されていることから必ずしも最適とは限らず，このような組み合わせに関する広範な調査は行われていない．

5.2 提案手法：一致性正則化と擬似ラベリングによる半教師あり共同学習

FixMatch [19] は，一致性正則化と擬似ラベリングの組み合わせが精度向上に寄与することを示した．しかし，これらの組み合わせは手動で設計されており，組み合わせに関する広範な調査は行われていない．そこで本章では，知識転移グラフの探索を用いた半教師あり学習の自動設計を提案する．一致性正則化と擬似ラベリングの代表的な従来法を主要な要素に分解し，それらを統一的にグラフで表現する．これらのグラフ表現は，教師あり学習，半教師あり学習，教師なし学習を内包する形となる．そしてグラフ構造の探索を通じて，一致性正則化と擬似ラベリングの組み合わせ，教師あり学習・半教師あり学習・教師なし学習の連携，共同学習への拡張を考慮した学習方法の自動設計を行う．本節では，5.2.1 項で半教師あり学習のグラフ表現，5.2.2 項でゲート関数と損失計算，5.2.3 項でグラフ構造の探索について述べる．

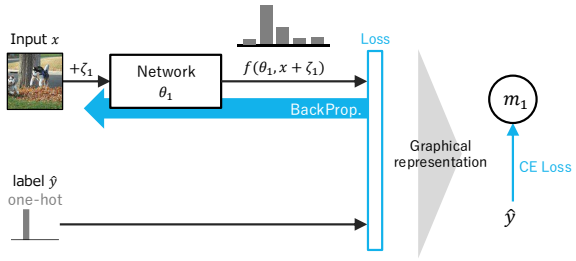
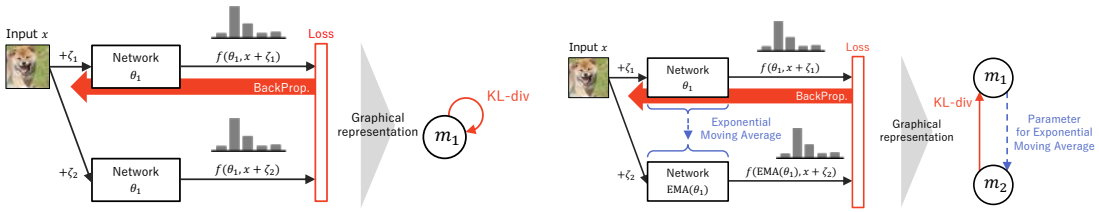
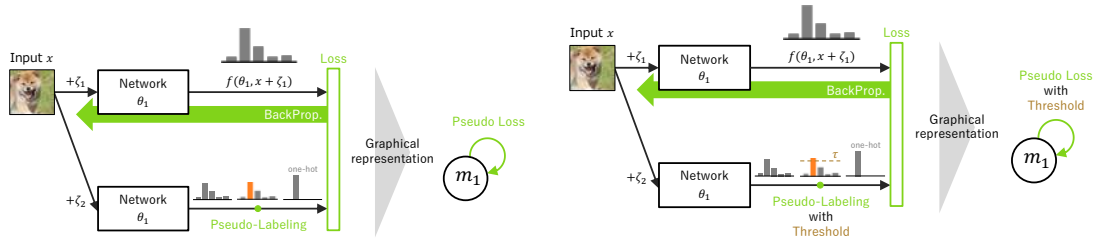


図 5.1: 教師あり学習のグラフ表現



(a) Π-model　　(b) Mean Teacher



(c) Pseudo-label　　(d) FixMatch

図 5.2: 半教師あり学習に代表的な従来法のグラフ表現

5.2.1　半教師あり学習のグラフ表現

半教師あり学習における代表的な従来法のグラフ表現を図 5.1 と図 5.2 に示す．$\hat{y}$ は正解ラベル，ノードはモデル，エッジは損失計算を表し，エッジの方向は勾配が伝達される方向を表す．有向エッジがノード m_s からノード m_t に向かう場合，そのエッジ上の損失を $\mathcal{L}_{s,t}$ と定義する．図 5.1 に示すように，ラベルありデータを用いた一般的な教師あり学習は，1 つのノード，1 つのエッジ，$\hat{y}$ で構成されるグラフとして表現できる．ここで，$\hat{y}$ からノードへの有向エッジは，教師あり損失を意味する．以降では，一致性正則化，擬似ラベリング，FixMatch におけるラベルなしデータに対する処理のグラフ表現について述べる．

■ 一致性正則化

一致性正則化の代表的な従来法として，Π-model [130] と Mean Teacher [28] をグラフで表現する．Π-model のグラフ表現を図 5.2a に示す．$f(\boldsymbol{\theta}, \boldsymbol{x})$ は，パラメータ $\boldsymbol{\theta}$ を持つモデルの入力データ $\boldsymbol{x}$ に

対する出力分布を表す．Π-model は，同一データに異なる摂動を加えることで作成した 2 つのデータ $\boldsymbol{x}+\boldsymbol{\zeta}_1$，$\boldsymbol{x}+\boldsymbol{\zeta}_2$ をモデルに入力し，各データに対する確率分布 $f(\boldsymbol{\theta}_1, \boldsymbol{x}+\boldsymbol{\zeta}_1)$，$f(\boldsymbol{\theta}_1, \boldsymbol{x}+\boldsymbol{\zeta}_2)$ が一致するように，一致性損失によってモデルを学習する．一致性損失は次のように定義する．

$$\mathcal{L}_{s,t}^{\mathrm{con}} = \mathcal{R}(f(\boldsymbol{\theta}_t, \boldsymbol{x}+\boldsymbol{\zeta}_1), f(\boldsymbol{\theta}_s, \boldsymbol{x}+\boldsymbol{\zeta}_2)), \tag{5.1}$$

ここで，$\mathcal{R}(\cdot,\cdot)$ は距離関数である．一致性正則化では，距離関数として平均二乗誤差や KL divergence が使用される．本章では，距離関数として KL divergence を使用する．Π-model のグラフ表現は，ノード m_1 からノード m_1 へ接続されたエッジを持つグラフとして表現し，エッジには一致性損失 $\mathcal{L}_{1,1}^{\mathrm{con}}$ を適用する．エッジの始点と終点が同じノードであるため，最終的に Π-model のグラフ表現は，1 つのノードと 1 つのエッジを持つグラフとして表現する．

Mean Teacher のグラフ表現を図 5.2b に示す．Mean Teacher は，Π-model に指数移動平均モデルを導入し，Π-model を 2 つのモデルを用いた学習へ拡張する．Mean Teacher の一致性損失は，パラメータ $\boldsymbol{\theta}_t$ を持つモデルと，EMA パラメータ $\mathrm{EMA}(\boldsymbol{\theta}_t)$ を持つモデルを用いて，式 5.1 により計算する．ここで，EMA パラメータは Backpropagation によって更新するのではなく，他のモデルのパラメータを現在の EMA パラメータに直接加算することで更新する．EMA パラメータの更新は次のように定義する．

$$\mathrm{EMA}(\boldsymbol{\theta}_{t,k}) = \alpha \mathrm{EMA}(\boldsymbol{\theta}_{t,k-1}) + (1-\alpha)\boldsymbol{\theta}_{t,k} \tag{5.2}$$

ここで，α はハイパーパラメータ，$\boldsymbol{\theta}_{t,k}$ は k 回のパラメータ更新を行ったモデル m_t のパラメータである．本章では，α として 0.999 を使用する．Mean Teacher のグラフ表現は，一致性損失によって学習するノード m_1，EMA パラメータを持つノード m_2 の 2 つのノードを持ち，ノード m_2 からノード m_1 へのエッジ $\mathcal{L}_{2,1}^{\mathrm{con}}$ を持つグラフとして表現する．加えて，EMA モデルのパラメータ更新に関する関係も，損失と同様に有向エッジで表現する．最終的に Mean Teacher のグラフ表現は，2 つのノードと 2 つのエッジを持つグラフとして表現する．ノード m_1 は，ノード m_2 の確率分布を用いて学習することから，この学習方法を Knowledge Distillation [16] と捉えることができる．

■ 擬似ラベリング

擬似ラベリングの代表的な従来法として，Pseudo-Label [135] をグラフで表現する．図 5.2c に Pseudo-Label のグラフ表現を示す．Pseudo-Label は，確率分布 $f(\boldsymbol{\theta}_1, \boldsymbol{x}+\boldsymbol{\zeta}2)$ において最も高い確率値を持つクラスを正解クラスとした one-hot 擬似ラベル $\boldsymbol{y}'$ を作成する．擬似ラベルの作成は次のように定義する．

$$y'_{i'} = \begin{cases} 1 & \text{if} \quad i' = \arg\max_i f_i(\boldsymbol{\theta}_s, \boldsymbol{x}+\boldsymbol{\zeta}_2) \\ 0 & \text{otherwise} \end{cases} \tag{5.3}$$

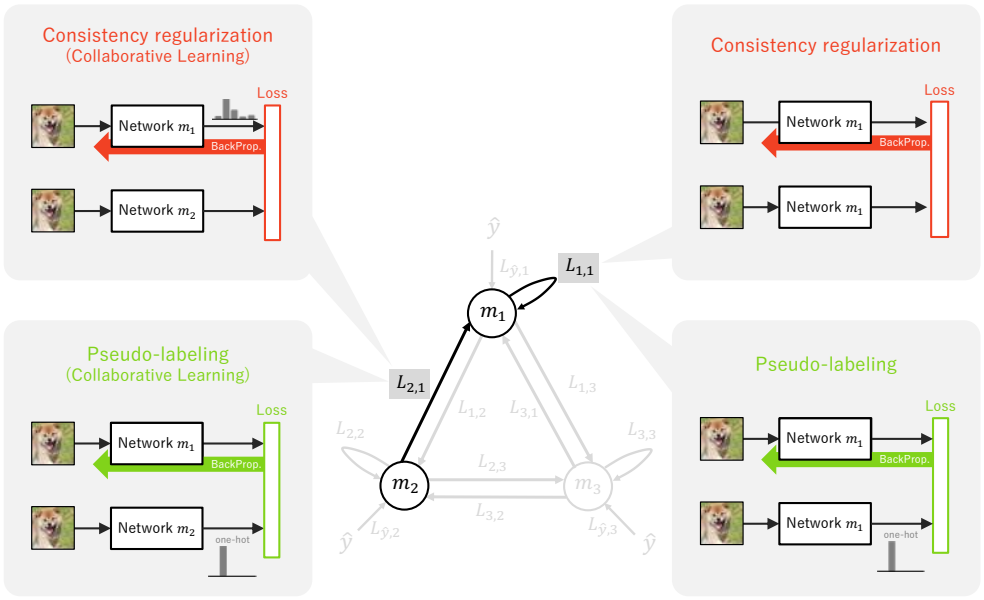


図 5.3: グラフ構造の変更による派生手法と共同学習への拡張

ここで，i' は擬似ラベルのクラス ID，i は確率分布のクラス ID である．そして擬似ラベルの作成後に，擬似ラベル $\boldsymbol{y}'$ と $f(\boldsymbol{\theta}_1, \boldsymbol{x} + \boldsymbol{\zeta}1)$ との交差エントロピー損失を用いてモデルを学習する．擬似ラベルを用いた損失は次のように定義する．

$$\mathcal{L}_{s,t}^{\mathrm{pse}} = -\sum_{i}^{C} y_i' \log f_i(\boldsymbol{\theta}_t, \boldsymbol{x} + \boldsymbol{\zeta}_1) \tag{5.4}$$

したがって Pseudo-Label のグラフ表現は，ノード m_1 からノード m_1 へと接続されたエッジ $\mathcal{L}_{1,1}^{\mathrm{pse}}$ を持つグラフとして表現する．

■ FixMatch

図 5.2d に FixMatch のグラフ表現を示す．FixMatch は，しきい値処理を導入した擬似ラベリングと，擬似ラベルを用いた学習で構成される．したがって FixMatch のグラフ表現は，Pseudo-Label のグラフ表現にしきい値処理を導入することで表現する．しきい値処理を導入した擬似ラベル損失は次のように定義する．

$$\mathcal{L}_{s,t}^{\mathrm{tp}} = \mathbb{1}_{[\max_i f_i(\boldsymbol{\theta}_s, \boldsymbol{x} + \boldsymbol{\zeta}_2) > \tau]} \cdot \mathcal{L}_{s,t}^{\mathrm{pse}} \tag{5.5}$$

ここで，τ はしきい値，$\mathbb{1}[\max_i f_i(\boldsymbol{\theta}_s, \boldsymbol{x} + \boldsymbol{\zeta}_2) > \tau] \in 0, 1$ は指示関数である．指示関数は，$\max_i f_i(\boldsymbol{\theta}_s, \boldsymbol{x} + \boldsymbol{\zeta}2) > \tau$ の場合に 1，それ以外の場合に 0 となる．本章では，τ として 0.95 を使用する．

■ 従来法の派生手法や共同学習への拡張

グラフ構造の変更による従来法の派生手法や共同学習への拡張を図 5.3 に示す．従来法のグラフ表現を組み合わせることで，従来法の派生手法や共同学習に容易に拡張することができる．

5.2.2 ゲート関数と損失計算

より多様な半教師あり学習を表現可能とするために，損失計算にゲート関数を導入する．

■ ゲート関数

エッジの接続を表現する Through Gate，エッジの切断を表現する Cutoff Gate に加えて，Positive Linear Gate と Negative Linear Gate を導入する．Positive Linear Gate は，学習の進行に応じて 0 から 1 へ線形に変化する重みを損失値に適用する．Positive Linear Gate は次のように定義する．

$$G_{s,t}^{\text{positive}}(\mathcal{L}_{s,t}) = \frac{k}{k_{\text{end}}} \cdot \mathcal{L}_{s,t}, \tag{5.6}$$

ここで，k は現在のイテレーション数，k_{end} は学習終了時の総イテレーション数である．Negative Linear Gate は，学習の進行に応じて 1 から 0 へ線形に変化する重みを損失値に適用する．Negative Linear Gate は次のように定義する．

$$G_{s,t}^{\text{negative}}(\mathcal{L}_{s,t}) = \frac{k_{\text{end}} - k}{k_{\text{end}}} \cdot \mathcal{L}_{s,t}. \tag{5.7}$$

これらの 2 種類の Linear Gate を用いることで，学習中に学習戦略を切り替える半教師あり学習を表現することが可能となる．また，しきい値処理をゲート関数として表現する．しきい値処理を行うゲート関数は次のように定義する．

$$G_{s,t}^{\text{threshold}}(\mathcal{L}_{s,t}) = \mathbb{1}_{[\max_i f_i(\boldsymbol{\theta}_s, \boldsymbol{x}+\boldsymbol{\zeta}_2)>\tau]} \cdot \mathcal{L}_{s,t}. \tag{5.8}$$

しきい値処理を表現したゲート関数と式 5.5 のしきい値処理を内包した損失関数を目的に応じて使い分けることで，グラフ構造の探索における探索空間の設計に柔軟性が生まれる．

■ 損失計算

損失計算は各エッジごとに行い，最終的なノード m_t の損失関数は次のように定義する．

$$\mathcal{L}_t = G_{\hat{y},t}(\mathcal{L}_{\hat{y},t}) + G_{t,t}(\mathcal{L}_{t,t}) + \sum_{s}^{N} G_{s,t}(\mathcal{L}_{s,t}), \tag{5.9}$$

ここで，N はグラフ内のノードの数を表す．エッジごとに異なるゲート関数を選択できるようにすることで，より多様な学習戦略を表現することができる．

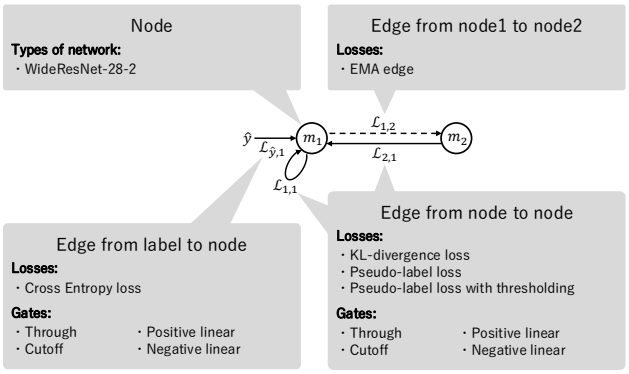


図 5.4: FixMatch に基づいたグラフ構造のハイパーパラメータ

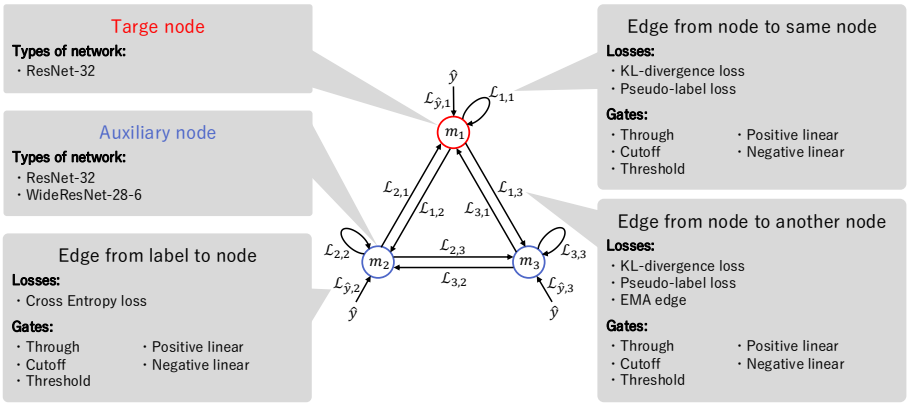


図 5.5: 共同学習を含む多様な探索空間におけるハイパーパラメータ

5.2.3 グラフ構造の探索

グラフ構造のハイパーパラメータ探索を通じて，半教師あり学習手法を自動設計する．グラフ構造のハイパーパラメータを図 5.4 と図 5.5 に示す．ハイパーパラメータ探索では，エッジに割り当てる損失式とゲート関数をエッジごとに選択する．損失式として EMA エッジが選択された場合，EMA エッジの終点ノードを EMA モデルとして設定し，Backpropagetion ではパラメータ更新を行わない．EMA エッジが選択されず，Cutoff Gate によって一切学習を行わないノードが発生した場合，そのノードはラベルありデータで教師あり学習によって事前学習されたモデルを使用する．事前学習されたモデルは，グラフ構造に基づいた学習中にパラメータを凍結し，パラメータの更新は行わない．

探索方法としてランダムサーチと Asynchronous Successive Halving Algorithm (ASHA) [96] を使用する．グラフ構造の評価スコアとして，ノードの精度を使用する．グラフ構造におけるノードの数と精度を最大化したいノード，評価を行うグラフ構造の数は，グラフ構造の探索前にあらかじめ人手で設定する．精度を最大化したいノードは，target node と呼ぶ．評価対象となるグラフ構造はランダムに決定される．ハイパーパラメータ探索中は，一定のイテレーション間隔で，target node の精度をテストデータを用いて評価する．$1, 2, 4, 8, \ldots, 2^k$ 回目の評価時に ASHA によって学習の早期終了の

表 5.1: FixMatch に基づいた探索空間における学習条件

	CIFAR-100
Training Iterations	1,048,576 ($= 2^{20}$)
Labeled Batch size	64
Unlabeled Batch size	448
Learning Rate	0.03
Weight Decay	0.001
SGD Momentum	0.9
SGD Nesterov	True
Perturbation ζ_1	RandomHorizontalFlip, RandomCrop, RandAugment [152]
Perturbation ζ_2	RandomHorizontalFlip, RandomCrop

判定を行う．評価回数が同じタイミングの過去に評価した精度と比較して，中央値の精度を下回っている場合は学習を打ち切る．その後，新しいグラフ構造をランダムに選択し，そのグラフ構造を用いた新しい学習を開始する．これらの処理は，評価済みのグラフ構造の数が事前に設定した評価したいグラフ構造の数に達するまで繰り返す．最終的に，評価した各グラフ構造の中で最も高い精度を達成したグラフ構造を探索結果とする．

5.3 評価実験

評価実験として，グラフ構造の探索により獲得した知識転移グラフを用いて学習した target node の性能を評価する．5.3.1 項では，評価実験に使用した実験条件について述べる．5.3.2 項では，FixMatch に基づいた探索空間における探索結果の精度評価を行う．5.3.3 項では，多様な探索空間における探索結果の精度評価を行う．5.3.4 項では，探索結果のグラフ構造の可視化を行う．5.3.5 項では，探索結果のグラフ構造から得られた知見に基づいてグラフ構造の手動設計を行う．

5.3.1 実験条件

本項では，評価実験に使用したデータセット，モデル，モデルの学習条件と評価条件，グラフ構造の探索条件について述べる．

表 5.2: 多様な探索空間における学習条件

	CIFAR-100	CIFAR-10
Training Iterations	82,400 ∼ 98,800	160,000 ∼ 192,000
Labeled Batch Size	31	50
Unlabeled Batch Size	97	50
Learning Rate	0.2	0.1
Weight Decay	0.0002	0.0001
SGD Momentum	0.9	0.9
SGD Nesterov	True	True
Perturbation ζ_1	RandomHorizontalFlip, RandomCrop	RandomHorizontalFlip, RandomCrop
Perturbation ζ_2	RandomHorizontalFlip, RandomCrop	RandomHorizontalFlip, RandomCrop

■ データセット

CIFAR-100 と CIFAR-10 [97] を使用する．各データセットの 50,000 サンプルの学習データには，人手で作成された正解ラベルが付与されている．半教師あり学習の実験では，学習データをラベルありデータとラベルなしデータに割り当て，ラベルなしデータに割り当てたデータの正解ラベルは使用しない．例として，ラベル付きデータとして 400 サンプルを割り当てた場合，ラベルなしデータは 49,600 サンプルとなる．

■ モデル

図 5.4 の FixMatch に基づいた探索空間では，WideResNet-28-2 [162] を使用する．図 5.5 の多様な探索空間では，ResNet-32 [10] と WideResNet-28-6 [162] を使用する．

■ モデルの学習条件と評価条件

FixMatch に基づいた手法は，学習条件によって精度が大きく変化することが知られている．そこで図 5.4 の FixMatch に基づいた探索空間では，FixMatch に基づいた従来法と公平な比較を行うために，FixMatch [19] で使用された学習条件を使用する．FixMatch に基づいた探索空間における学習条件を表 5.1 に示す．一方で図 5.5 の多様な探索空間では，半教師あり共同学習の従来法である Dual Student [30] で使用された学習条件を使用する．多様な探索空間における学習条件を表 5.2 に示す．学習イテレーションは，ラベルなしデータに基づいて 200 エポックに設定する．そのため，学習イテレーションの設定は，ラベルなしデータの数に応じて異なる．表 5.2 の学習条件は，FixMatch に適していない可能性がある点に注意が必要である．

表 5.3: CIFAR-100 における精度評価（FixMatch に基づいた探索空間）

Method	Venue	400 labels	2,500 labels	10,000 labels
Supervised†	-	10.13 ± 0.47	38.75 ± 0.54	61.01 ± 0.14
Pseudo-Label† [135]	ICML ’13	12.85 ± 0.87	40.91 ± 0.61	61.14 ± 0.09
Meanteacher† [28]	NeurIPS ’17	9.66 ± 0.65	38.87 ± 0.57	60.95 ± 0.12
Π-model† [130]	ICLR ’17	12.87 ± 1.25	39.92 ± 0.61	60.90 ± 0.16
VAT† [132]	TPAMI ’19	16.89 ± 0.27	46.83 ± 0.57	63.42 ± 0.21
MixMatch† [131]	NeurIPS ’19	20.05 ± 0.29	50.42 ± 0.62	67.90 ± 0.13
ReMixMatch† [163]	ICLR ’20	42.90 ± 0.00	65.23 ± 0.32	73.82 ± 0.23
FixMatch† [19]	NeurIPS ’20	46.63 ± 2.15	65.71 ± 0.43	71.72 ± 0.18
UDA† [134]	NeurIPS ’21	46.56 ± 2.06	65.63 ± 0.28	72.48 ± 0.10
FlexMatch† [140]	NeurIPS ’21	49.85 ± 1.51	66.65 ± 0.33	72.88 ± 0.04
Dash† [139]	ICML ’21	46.02 ± 2.31	65.53 ± 0.12	72.28 ± 0.13
CRMatch† [145]	GCPR ’21	50.61 ± 1.16	68.65 ± 0.27	73.76 ± 0.26
CoMatch† [149]	ICCV ’21	39.02 ± 0.77	62.76 ± 0.24	71.85 ± 0.16
AdaMatch† [164]	ICLR ’22	52.18 ± 1.48	66.74 ± 0.55	72.47 ± 0.12
SimMatch† [165]	CVPR ’22	51.18 ± 1.07	67.46 ± 0.27	73.58 ± 0.18
FreeMatch† [142]	ICLR ’23	50.76 ± 2.16	67.21 ± 0.21	72.83 ± 0.11
SoftMatch† [141]	ICLR ’23	50.36 ± 1.46	66.95 ± 0.05	72.74 ± 0.03
Ours	-	46.99 ± 0.90	66.79 ± 0.43	72.12 ± 0.40

■ グラフ構造の探索条件

グラフ構造の探索時は，CIFAR データセットの学習データ 50,000 サンプルを探索の学習データとして 40,000 サンプル，探索のテストデータとして 10,000 サンプルに分割して使用する．したがって，探索中は元々のテストデータを使用しない．ハイパーパラメータ探索後は，従来のデータセットで定義された学習データとテストデータで学習と評価を行い，従来法と比較する．

グラフ構造の探索では，FixMatch に基づいた探索空間と共同学習を含むより多様な探索空間の 2 種類の探索を行う．図 5.4 の探索空間では,2 つノードの精度のうち，高い方の精度をグラフ構造の評価値として使用し，60 のグラフ構造を評価する．図 5.5 の探索空間において, ノード 1 の精度をグラフ構造の評価値として使用し，1,500 種類のグラフ構造を評価する．

5.3.2 従来法との精度比較：FixMatch に基づいた探索空間

図 5.4 の探索空間において自動設計した提案手法の精度比較を表 5.3 に示す．提案手法は，全てのラベル付きデータ数の条件下において，FixMatch を超える精度を達成した．一方で，最先端

表 5.4: CIFAR-100 における精度評価（多様な探索空間）

Dataset	CIFAR-100					
Label Amount	2,000	4,000	6,000	8,000	10,000	all
Supervised	20.85	32.35	41.56	50.38	53.61	69.93
Pseudo-Label [135]	29.01	40.59	47.79	54.51	56.67	-
Π-model [130]	29.54	41.98	50.69	53.44	55.98	-
Mean Teacher [28]	31.28	43.10	48.85	49.98	54.52	-
FixMatch [19]	29.31	41.87	47.33	51.28	54.03	-
Ours (2nodes)	**42.04**	**51.92**	**55.94**	**60.65**	**63.08**	-
Ours (3nodes)	**42.55**	**54.60**	**57.12**	**62.49**	**63.81**	-

表 5.5: CIFAR-10 における精度評価（多様な探索空間）

Dataset	CIFAR-10		
Label Amount	1,000	4,000	all
Supervised	55.42	75.94	91.82
Pseudo-Label [135]	63.73	82.58	-
Π-model [130]	54.41	77.54	-
Mean Teacher [28]	53.45	82.50	-
FixMatch [19]	52.05	84.46	-
Ours (2nodes)	**63.78**	**85.08**	-
Ours (3nodes)	**82.45**	**87.03**	-

の FixMatch ベースの手法と比較すると，提案手法の精度は同程度または低い結果となった．ここで，CRMatch [145] は新しい一致性正則化の戦略に焦点を当てた手法，FlexMatch [140]，Dash [139]，AdaMatch [164]，SimMatch [165]，FreeMatch [142]，SoftMatch [141] は新しいしきい値戦略に焦点を当てた手法，CoMatch [149] は擬似ラベルと特徴空間の相互作用に焦点を当てた手法である．これらの FixMatch ベースの手法とは異なり，提案手法は一致性正則化と擬似ラベリングの組み合わせに焦点を当てている．そのため，提案手法は最先端の手法と異なる方法によって，同等の改善効果を達成したと言える．また，自動設計により獲得した学習方法と最先端の手法におけるアプローチを統合することで，さらなる精度向上が期待される．

5.3.3 従来法との精度比較：多様な探索空間

図 5.5 の探索空間において自動設計した提案手法の精度比較を表 5.4 に示す．Supervised は，ラベルありデータのみを用いた教師あり学習の結果を表す．提案手法は，すべてのラベルありデータ数の

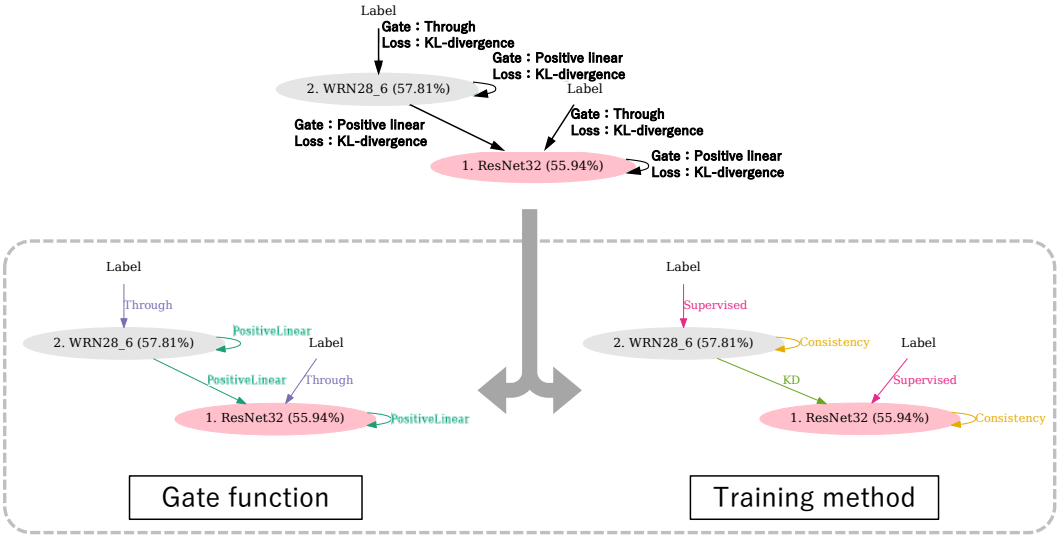


図 5.6: 自動設計されたグラフ構造の可視化方法

条件において，従来法よりも高い精度を達成した．また，提案手法はノード数を 2 から 3 に増やすことで，精度がさらに向上した．これは，ノード数が増加することで表現可能な半教師あり共同学習が増加し，各ラベルありデータ数の条件に適した学習方法が得られやすくなったためだと考える．

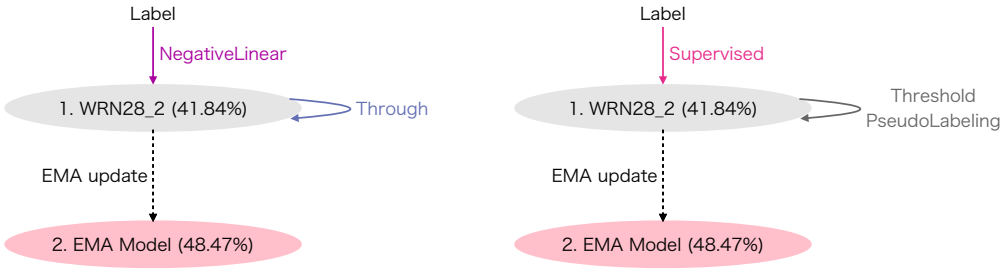
FixMatch の精度に着目すると，Π-model や Mean Teacher といった一致性正則化のみを用いた手法や Pseudo-Label といった擬似ラベリングのみを用いた手法と比べて，同等または低い精度となっている．これは，FixMatch の文献内で示されているように，学習条件が FixMatch に適していないことが原因であると考えられる．一方で提案手法は，探索によって学習条件やラベルありデータ数を考慮した自動設計が行われるため，表 5.3 や表 5.4 といった異なる学習条件下においても高い精度を発揮できたと言える．

CIFAR-100 における一致性正則化と擬似ラベリングに着目すると，一致性正則化手法である Π-model や Mean Teacher は，ラベル付きデータ数が 2,000〜6,000 のように少ない場合に高い精度を示した．一方，擬似ラベリング手法である Pseudo-Label は，ラベル付きデータ数が 8,000〜10,000 のように多い場合に高い精度を示した．したがって，適切な学習手法はラベル付きデータの数によって異なることが考えられる．

5.3.4 探索後のグラフ構造の可視化

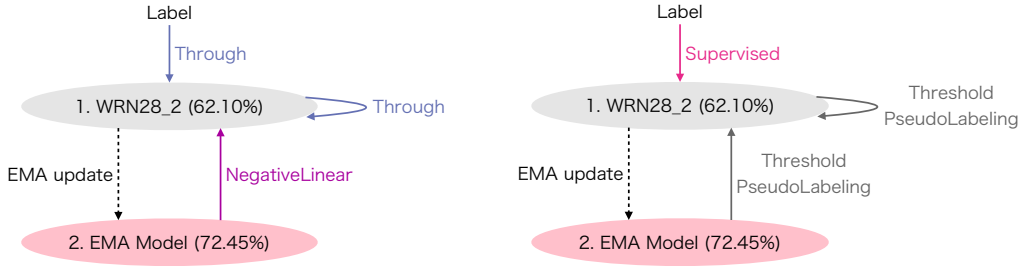
本項では，2 つの探索空間における探索結果を可視化する．エッジは，ゲート関数と損失関数の 2 つのハイパーパラメータを持つため，1 つのグラフ上で可視化すると，複雑な表現となる．そこで，図 5.6 のように各グラフ構造をゲート関数と損失間数の 2 つの観点に分けて可視化を行う．

FixMatch に基づいた探索空間 ラベルありデータが 400，2,500，10,000 の条件において探索した結果をそれぞれ図 5.7，図 5.8，図 5.9 に示す．



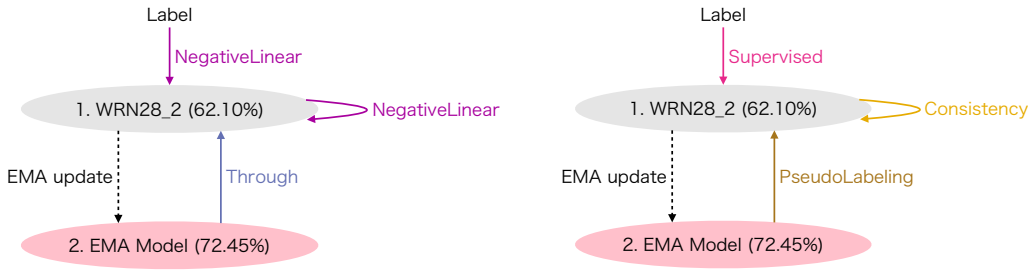
(a) Gate function (b) Training method

図 5.7: CIFAR-100 を用いた FixMatch に基づいた探索空間における探索結果（ラベルありデータ数：400）



(a) Gate function (b) Training method

図 5.8: CIFAR-100 を用いた FixMatch に基づいた探索空間における探索結果（ラベルありデータ数：2,500）



(a) Gate function (b) Training method

図 5.9: CIFAR-100 を用いた FixMatch に基づいた探索空間における探索結果（ラベルありデータ数：10,000）

図 5.7 は，FixMatch におけるラベルあり損失へ Negative Linear Gate による学習の切り替え戦略を導入した学習方法となっている．そのため，学習前半は従来の FixMatch を行い，学習後半はしきい値付きの擬似ラベリングのみで学習を行う．

図 5.8 は，FixMatch に EMA モデルからのしきい値付き擬似ラベリングを導入した学習方法となっている．このとき，EMA モデルからのしきい値付き擬似ラベリングに対して NegativeLinear が選択

されていることから，ノード 1 は学習前半に FixMatch 本来の擬似ラベルと EMA モデルによる擬似ラベルの 2 つを用いて学習を行い，学習後半に従来の FixMatch で学習を行う．

図 5.9 は，図 5.7 や図 5.8 と異なり，FixMatch と大きく異なる学習方法となっている．学習前半はラベルあり損失，一致性正則化，EMA モデルからの擬似ラベルの 3 つの学習を行い，学習後半は EMA モデルからの擬似ラベルのみで学習を行う．

多様な探索空間 ラベルありデータ数が 6,000 の場合における 2 ノードの探索結果を図 5.10a，3 ノードの探索結果を図 5.10c に示す．ラベルありデータ数が 10,000 の場合における 2 ノードの探索結果を図 5.10b，3 ノードの探索結果を図 5.10d に示す．赤色のノードはターゲットノード，青色と灰色のノードは補助ノードであることを表す．また，青色のノードは教師あり学習による事前学習モデルであることを表し，グラフ構造に基づいた学習中に更新は行わない．

図 5.10a のラベルありデータ数が 6,000 の場合における 2 ノードの学習戦略に着目すると，3 つのエッジにおいて Positive Linear Gate が選択されている．Positive Linear Gate は，学習前半ではエッジが切断されるため，各ノードは教師あり学習による学習を行う．一方で学習後半では Positive Linear Gate が選択されたエッジが接続されるため，各ノードは Π-model による学習と，ノード 2 からノード 1 への知識転移による学習を行う．学習後半の戦略は，Π-model と Mean Teacher を組み合わせた学習と言える．表 5.4 では，ラベルありデータが少ない場合に Π-model と Mean Teacher の精度が高い傾向にある．したがって，探索によってラベルありデータ数に応じた，高い精度を達成可能な半教師あり学習手法を選択していると言える．

図 5.10b のラベルありデータ数が 10,000 の場合における 2 ノードの学習戦略に着目すると，ノード間のエッジにおいて Positive Linear Gate，ノード 1 からノード 1 へのエッジにおいて Negative Linear Gate が選択され，ノード 2 は事前学習モデルとなっている．target node であるノード 1 は，学習前半に Π-model の学習を行い，学習後半に事前学習モデルが作成した擬似ラベルを用いて学習を行う．この一致性正則化と擬似ラベリングの組み合わせ方は，従来法にはない組み合わせ方であり，探索によって当たらな組み合わせ方が見出されたと言える．表 5.4 では，ラベルありデータ数が多い場合に Pseudo-Label の精度が高い傾向にある．したがって，ラベルありデータ数が 6,000 の場合と同様に，探索によってラベルありデータ数に応じた，高い精度を達成可能な半教師あり学習手法を選択していると言える．

3 ノードのグラフでは，2 ノードのグラフと同様の傾向が観察された．図 5.10c のラベルありデータ数が 6,000 の場合では学習前半に教師あり学習が採用され，図 5.10d のラベルありデータ数が 10,000 の場合では学習後半に擬似ラベリングが採用されている．

2 ノードと 3 ノードにおけるグラフ構造の傾向から，ラベルありデータ数が少ない場合では，学習前半に教師あり学習，学習後半に一致性正則化を行うことが効果的であると言える．一方でラベルありデータ数が多い場合では，学習前半に一致性正則化，学習後半に擬似ラベリングを行うことが効果的であると言える．また，探索により得られたグラフ構造を分析することで，新たな知見を獲得することができたと言える．

Gate function　　Training method

(a) 2 ノード（ラベルありデータ：6,000）

Gate function　　Training method

(b) 2 ノード（ラベルありデータ：10,000）

Gate function　　Training method

(c) 3 ノード（ラベルありデータ：6,000）

Gate function　　Training method

(d) 3 ノード（ラベルありデータ：10,000）

図 5.10: CIFAR-100 を用いた多様な探索空間における探索結果

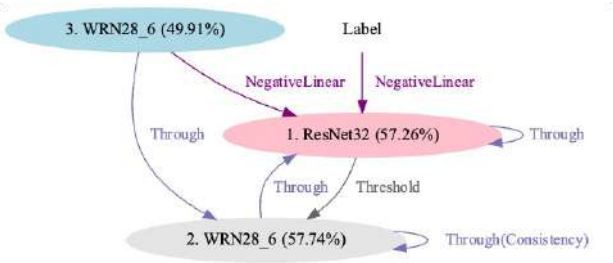


図 5.11: 探索したグラフ構造を手動で再設計したグラフ

5.3.5　探索により得られた知見に基づいたグラフ構造の再設計

グラフ構造の探索では，表現可能なグラフ構造の数が膨大なため，必ずしも最適な構造が得られているとは限らない．例えば，図 5.10c のグラフは，5.3.4 項で得られた知見の一部を反映していない．そこで，図 5.10c のグラフを得られた知見に基づいて人手で再設計し，探索から得られた知見の導入による精度変化を調査する．

手動で再設計したグラフを図 5.11 に示す．ラベルありデータ数が少ない場合に一致性正則化が高い精度を発揮する傾向に基づいて，ノード 2 からノード 2 へのエッジを Through Gate を用いた一致性損失に変更した．target node であるノード 1 の精度は，図 5.10c の変更前のグラフでは 57.12%だったのに対して，図 5.11 の変更後のグラフでは 57.26%となり，得られた知見の導入により精度が向上した．このことから，得られた知見を活用して学習方法を再設計することで，精度向上の可能性があると考えられる．一方で，半教師あり学習におけるグラフ構造の探索に適した探索方法の設計は今後の課題である．

5.4　まとめ

本章では，一致性正則化と擬似ラベリングを統一的なグラフで表現し，一致性正則化と擬似ラベリングの組み合わせ，教師あり学習・半教師あり学習・教師なし学習の連携，共同学習への拡張を考慮可能な多様な探索空間においてグラフ構造の探索を行った．探索により得られたグラフ構造を分析することで，半教師あり学習における新たな知見を獲得した．また，自動設計された手法から得られた知見に基づいて，手動によるグラフ構造の再設計を行い，精度が向上することを確認した．一方で，グラフ構造の探索における探索コストの問題は以前として未解決であり，探索空間の多様性に応じて探索コストが増加する．特に半教師あり学習では，ラベルありデータ数の条件によって適切な学習戦略が異なる傾向にあったため，ラベルありデータ数の条件ごとに探索することが好ましい．今後の課題として，半教師あり学習におけるグラフ構造の探索に適した探索方法の設計が挙げられる．

第6章

結論と展望

本稿では，モデルからモデルへ知識を転移する学習を共同学習と定義し，共同学習における学習戦略の多様化と効果的な手法の設計に焦点を当てた．共同学習の代表的な従来法をグラフ表現に落とし込み，従来法を含んだ多様な共同学習を表現可能な知識転移グラフを提案し，グラフ構造の組み合わせ問題としてハイパーパラメータ探索を行うことで，効果的な共同学習を自動設計した．次に，1 つのモデルの高精度を目指す従来の共同学習における目的を，共同学習を行う複数モデルからなるグループ単位の改善に拡張し，アンサンブル学習のための共同学習を提案した．また，アンサンブル学習の推論コストの削減を目的として，複数モデルにおける知識の多様性を考慮したモデル軽量化手法を提案した．その後，ラベルなしデータの活用を目的として，半教師あり学習の代表的な従来法をグラフ表現に落とし込み，半教師あり共同学習を提案した．以下に，本論文の結論と今後の展望について述べる．

6.1 結論

各章のまとめは以下の通りである．

2 章では，共同学習を知識表現に焦点を当てたアプローチ，モデルの組み合わせに焦点を当てたアプローチ，特定の設定に特化したアプローチに分類し，各アプローチにおける手法の進展について述べた．その後，本稿の提案手法に関連する知識表現に焦点を当てたアプローチと，モデルの組み合わせに焦点を当てたアプローチについて関連研究をまとめた．

3 章では，共同学習における学習戦略の多様化と効果的な手法の設計に焦点を当て，従来法を含んだ多様な共同学習を表現可能な知識転移グラフを提案した．知識転移グラフでは，モデルをノード，損失計算と知識転移の関係をエッジで表現し，共同学習の従来法に統一的な視点を与えた．また，損失関数に損失値に重み付けを行うゲート関数を導入することで，多様な知識転移戦略を表現可能とした．そして，効果的なグラフ構造を人手により設計するのではなく，探索によって自動設計することで，従来法では探索しきれなかった大規模かつ複雑な共同学習の設計を可能にした．11 種類のデータセットを用いた評価実験では，探索により従来法と異なる知識転移の戦略を獲得し，探索により獲得した共同学習が従来法を上回る精度を達成することを確認した．また，探索で選択されたモデル構造やゲート関数が精度に寄与していることを確認し，探索時におけるグラフ構造の候補の多様性が大きいほど，効果的なグラフ構造を発見しやすいことを確認した．

4 章では，1 つのモデルの高精度を目指す従来の共同学習における目的を，共同学習を行う複数モ

デルからなるグループ単位の改善に拡張し，アンサンブル学習のための共同学習を提案した．アンサンブル推論と共同学習の関係を分析し，モデルの多様性を促進するための知識転移戦略を設計した．その後，知識転移グラフをアンサンブル学習に拡張し，グラフ構造の探索によってアンサンブル推論に効果的な共同学習を自動設計した．さらに，モデルの多様性から生じる知識の衝突を考慮した，多様な知識を持つ複数の教師モデルを用いたモデル軽量化手法を設計し，アンサンブル推論の計算コストの問題に対応した．5 種類のデータセットを用いた評価実験では，探索により獲得した共同学習が従来法より高いアンサンブル精度を達成することを確認した．探索により獲得した共同学習は，各モデルが異なる注視領域を獲得することを促し，アンサンブル推論に適した多様性を獲得したことを確認した．一方でモデル軽量化では，この多様性によって各モデルの知識転移の勾配が衝突し，各モデルの知識を 1 つのモデルに転移することが難しいことを確認した．そこで，勾配衝突を解消する手法を導入し，各モデルが持つ多様な注視領域を活用したモデル軽量化が可能であることを示した．

5 章では，ラベルなしデータの活用を目的として，半教師あり学習の代表的な従来法をグラフ表現に落とし込み，半教師あり共同学習を提案した．半教師あり学習において代表的な従来法である一致性正則化と擬似ラベリングを統一的なグラフで表現し，一致性正則化と擬似ラベリングの組み合わせ，教師あり学習・半教師あり学習・教師なし学習の連携，共同学習への拡張を考慮可能な多様な探索空間においてグラフ構造の探索を行った．探索により得られたグラフ構造を分析することで，半教師あり学習における新たな知見を獲得した．また，自動設計された手法から得られた知見に基づいて，手動によるグラフ構造の再設計を行い，精度が向上することを確認した．

6.2 展望

本稿では，共同学習における学習戦略の多様化，グループ単位の学習への拡張，ラベルなしデータの活用について取り組み，グラフ構造の探索によって各取り組みにおける効果的な共同学習を自動設計した．探索により得られた共同学習は，モデル数やラベルありデータ数などの条件によって異なる学習戦略が採用された．そのため，モデル数やラベルありデータ数などの条件によって適切な学習戦略が異なると言える．一方で，世の中に存在する問題設定は膨大であり，各問題設定においてグラフ構造の探索を行うことは，現実的ではない．また，表現可能なグラフ構造の数が膨大なため，必ずしも探索によって最適な学習方法が自動設計された保証はない．今後の課題として，知識転移グラフに特化した低コストな探索手法の設計が挙げられる．各章における評価実験では，探索時と異なるデータセットにおいても高い精度を発揮すること，探索結果から得られた知見に基づいてグラフ構造を再設計すると精度が改善することを確認した．そのため，目的の問題設定と似た小規模データを用いてグラフ構造の探索を行い，その探索結果を用いて目的の問題設定の学習を行うメタラーニングベースの探索や，探索結果のグラフを初期値としたグラフ構造の再探索によって，低コストなグラフ構造の探索が実現できる可能性がある．

謝　辞

本研究の遂行にあたり，常日頃ご指導を賜りました中部大学 理工学部 AI ロボティクス学科 藤吉弘亘 教授に深く感謝の意を表します．本論文をまとめるにあたり，有益なご討論，ご助言を賜りました中部大学 理工学部 AI ロボティクス学科 平田豊 教授，中部大学 工学部情報工学科 山下隆義 教授，東京科学大学 情報理工学院 佐藤育郎 教授に謹んで感謝いたします．本研究において，貴重なご意見，ご指導を頂きました中部大学 工学部情報工学科 山下隆義 教授，中部大学 AI 数理データサイエンスセンター 平川翼 講師に心から厚く御礼申し上げます．最後に，本研究にご協力して頂いた藤吉研究室と山下研究室の皆様に感謝致します．

参考文献

[1] T. Huang, S. You, F. Wang, C. Qian, and C. Xu, “Knowledge distillation from a stronger teacher”, Advances in Neural Information Processing Systems (NeurIPS), vol.35, pp.33716–33727, 2022.

[2] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, “Fitnets: Hints for thin deep nets”, International Conference on Learning Representations (ICLR), 2015.

[3] S. Zagoruyko, and N. Komodakis, “Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer”, International Conference on Learning Representations (ICLR), 2017.

[4] J. Yim, D. Joo, J. Bae, and J. Kim, “A gift from knowledge distillation: Fast optimization, network minimization and transfer learning”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.4133–4141, 2017.

[5] T. Furlanello, Z. Lipton, M. Tschannen, L. Itti, and A. Anandkumar, “Born again neural networks”, International Conference on Machine Learning (ICML), vol.80, pp.1607–1616, 2018.

[6] X. Jin, B. Peng, Y. Wu, Y. Liu, J. Liu, D. Liang, J. Yan, and X. Hu, “Knowledge distillation via route constrained optimization”, IEEE/CVF International Conference on Computer Vision (ICCV), 2019.

[7] Q. L. Mengya Gao, Yujun Shen, and C. C. Loy, “Residual knowledge distillation”, arXiv preprint arXiv:2002.09168, 2020.

[8] S. Park, and N. Kwak, “Feature-level ensemble knowledge distillation for aggregating knowledge from multiple networks”, European Conference on Artificial Intelligence (ECAI), vol.325, pp.1411–1418, 2020.

[9] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, “Deep mutual learning”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[10] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition”, IEEE conference on computer vision and pattern recognition (CVPR), pp.770–778, 2016.

[11] 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", International Conference on Learning Representations ICLR, 2021.

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks", Advances in Neural Information Processing Systems ((NeurIPS)), vol.28, 2015.

[13] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection", IEEE/CVF International Conference on Computer Vision (ICCV), pp.9626-9635, 2019.

[14] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation", Medical Image Computing and Computer-Assisted Intervention (MICCAI), pp.234–241, 2015.

[15] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation", Computer Vision – ECCV 2018, pp.833–851, 2018.

[16] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network", Neural Information Processing Systems (NeurIPS) Deep Learning and Representation Learning Workshop, 2015.

[17] A. Korattikara Balan, V. Rathod, K. P. Murphy, and M. Welling, "Bayesian dark knowledge", Advances in Neural Information Processing Systems (NeurIPS), eds. C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, vol.28, Curran Associates, Inc., 2015.

[18] S. I. Mirzadeh, M. Farajtabar, A. Li, and H. Ghasemzadeh, "Improved knowledge distillation via teacher assistant: Bridging the gap between student and teacher", Association for the Advancement of Artificial Intelligence (AAAI), 2020.

[19] A. Kurakin, C. L. Li, C. Raffel, D. Berthelot, E. D. Cubuk, H. Zhang, K. Sohn, N. Carlini, and Z. Zhang, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence", Neural Information Processing Systems, pp.596–608, 2020.

[20] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker, "Learning efficient object detection models with knowledge distillation", Advances in Neural Information Processing Systems (NeurIPS), pp.742–751, 2017.

[21] W. Wang, W. Hong, F. Wang, and J. Yu, "Gan-knowledge distillation for one-stage object detection", IEEE Access, vol.8, pp.60719-60727, 2020.

[22] T. Wang, L. Yuan, X. Zhang, and J. Feng, "Distilling object detectors with fine-grained feature imitation", IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.4928-4937, 2019.

[23] S. Tang, L. Feng, W. Shao, Z. Kuang, W. Zhang, and Z. Lu, "Learning efficient detector with semi-supervised adaptive distillation", British Machine Vision Conference (BMVC), 2019.

[24] J. Xie, B. Shuai, J. F. Hu, J. Lin, and W. S. Zheng, "Improving fast segmentation with teacher-student learning", arXiv preprint arXiv:1810.08476, 2019.

[25] Y. Liu, K. Chen, C. Liu, Z. Qin, Z. Luo, and J. Wang, "Structured knowledge distillation for semantic segmentation", IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[26] T. He, C. Shen, Z. Tian, D. Gong, C. Sun, and Y. Yan, "Knowledge adaptation for efficient semantic segmentation", IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.578-587, 2019.

[27] Y. Shan, "Distilling pixel-wise feature similarities for semantic segmentation", arXiv preprint arXiv:1910.14226, 2019.

[28] A. Tarvainen, and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results", Advances in Neural Information Processing Systems (NeurIPS), pp.1195–1204, 2017.

[29] S. Qiao, W. Shen, Z. Zhang, B. Wang, and A. Yuille, "Deep co-training for semi-supervised image recognition", European Conference on Computer Vision (ECCV), 2018.

[30] Z. Ke, D. Wang, Q. Yan, J. Ren, and R. W. Lau, "Dual student: Breaking the limits of the teacher in semi-supervised learning", IEEE/CVF International Conference on Computer Vision, pp.6727–6735, 2019.

[31] Y. Luo, J. Zhu, M. Li, Y. Ren, and B. Zhang, "Smooth neighbors on teacher graphs for semi-supervised learning", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[32] Q. Xie, E. Hovy, M. T. Luong, and Q. V. Le, "Self-training with noisy student improves imagenet classification", IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[33] N. Papernot, M. Abadi, Ú. Erlingsson, I. J. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data", International Conference on Learning Representations (ICLR), 2017.

[34] S. H. Lee, D. H. Kim, and B. C. Song, "Self-supervised knowledge distillation using singular value decomposition", European Conference on Computer Vision (ECCV), eds. V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, vol.11210, pp.339–354, 2018.

[35] M. Noroozi, A. Vinjimoor, P. Favaro, and H. Pirsiavash, "Boosting self-supervised learning via knowledge transfer", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[36] T. Xu, and C. Liu, "Data-distortion guided self-distillation for deep neural networks", Association for the Advancement of Artificial Intelligence (AAAI), pp.5565–5572, 2019.

[37] H. Lee, S. J. Hwang, and J. Shin, "Self-supervised label augmentation via input transformations", International Conference on Machine Learning (ICML), vol.119, pp.5714–5724, 2020.

[38] G. Xu, Z. Liu, X. Li, and C. C. Loy, "Knowledge distillation meets self-supervision", European Conference on Computer Vision (ECCV), eds. A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, vol.12354, pp.588–604, 2020.

[39] Y. Bai, Z. Wang, J. Xiao, C. Wei, H. Wang, A. L. Yuille, Y. Zhou, and C. Xie, "Masked Autoencoders Enable Efficient Knowledge Distillers", IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.24256–24265, 2023.

[40] S. Ren, F. Wei, Z. Zhang, and H. Hu, "TinyMIM: An Empirical Study of Distilling MIM Pre-Trained Models", IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.3687–3697, 2023.

[41] W. Huang, Z. Peng, L. Dong, F. Wei, J. Jiao, and Q. Ye, "Generic-to-Specific Distillation of Masked Autoencoders", IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.15996–16005, 2023.

[42] Y. Wei, H. Hu, Z. Xie, Z. Liu, Z. Zhang, Y. Cao, J. Bao, D. Chen, and B. Guo, "Improving CLIP Fine-tuning Performance", IEEE International Conference on Computer Vision (ICCV), pp.5439–5449, 2023.

[43] H. Wu, Y. Gao, Y. Zhang, S. Lin, Y. Xie, X. Sun, and K. Li, "Self-supervised Models are Good Teaching Assistants for Vision Transformers", International Conference on Machine Learning (ICML), pp.24031–24042, 2022.

[44] L. W. L. Z. Y. Y. Zhiyuan Fang, Jianfeng Wang, and Z. Liu, "Seed: Self-supervised distillation for visual representation", International Conference on Learning Representations ICLR, 2021.

[45] J. Zhang, H. Peng, K. Wu, M. Liu, B. Xiao, J. Fu, and L. Yuan, "Minivit: Compressing vision transformers with weight multiplexing", 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.12135-12144, 2022.

[46] Z. Hao, J. Guo, D. Jia, K. Han, Y. Tang, C. Zhang, H. Hu, and Y. Wang, “Learning efficient vision transformers via fine-grained manifold distillation”, Advances in Neural Information Processing Systems (NeurIPS), 2022.

[47] T. Wang, J. Y. Zhu, A. Torralba, and A. A. Efros, “Dataset distillation”, arXiv preprint arXiv:1811.10959, 2018.

[48] C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil, “Model compression”, Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pp.535–541ACM, 2006.

[49] S. Minami, T. Yamashita, and H. Fujiyoshi, “Gradual sampling gate for bidirectional knowledge distillation”, International Conference on Machine Vision Applications (MVA), pp.1-6, 2019.

[50] T. Wen, S. Lai, and X. Qian, “Preparing lessons: Improve knowledge distillation with better supervision”, Neurocomputing, vol.454, pp.25-33, 2021.

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

[52] L. Beyer, X. Zhai, A. Royer, L. Markeeva, R. Anil, and A. Kolesnikov, “Knowledge distillation: A good teacher is patient and consistent”, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.10915-10924, 2022.

[53] S. Yashima, “Keffectiveness of function matching in driving scene recognition”, European Conference on Computer Vision (ECCV) Workshop, 2022.

[54] Y. N. D. Hongyi Zhang, Moustapha Cisse, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization”, International Conference on Learning Representations ICLR, 2018.

[55] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, “Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks”, IEEE Winter Conference on Applications of Computer Vision (WACV), 2018.

[56] P. Dhar, R. V. Singh, K. C. Peng, Z. Wu, and R. Chellappa, “Learning without memorizing”, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[57] Y. Liu, J. Cao, B. Li, C. Yuan, W. Hu, Y. Li, and Y. Duan, “Knowledge distillation via instance relationship graph”, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[58] L. Zhang, J. Song, A. Gao, J. Chen, C. Bao, and K. Ma, "Be your own teacher: Improve the performance of convolutional neural networks via self distillation", IEEE/CVF International Conference on Computer Vision (ICCV), 2019.

[59] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, "Learning a unified classifier incrementally via rebalancing", IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[60] Y. Luan, H. Zhao, Z. Yang, and Y. Dai, "MSD: multi-self-distillation learning via multi-classifiers within deep neural networks", arXiv preprint arXiv:1911.09418, 2019.

[61] S. Ahn, S. X. Hu, A. Damianou, N. D. Lawrence, and Z. Dai, "Variational information distillation for knowledge transfer", IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.9163–9171, 2019.

[62] Y. Tian, D. Krishnan, and P. Isola, "Contrastive representation distillation", International Conference on Learning Representations (ICLR), 2020.

[63] Z. Yang, Z. Li, M. Shao, D. Shi, Z. Yuan, and C. Yuan, "Masked generative distillation", European Conference on Computer Vision (ECCV), pp.53–69, 2022.

[64] T. Huang, Y. Zhang, M. Zheng, S. You, F. Wang, C. Qian, and C. Xu, "Knowledge diffusion for distillation", Advances in Neural Information Processing Systems (NeurIPS), 2023.

[65] W. Park, D. Kim, Y. Lu, and M. Cho, "Relational knowledge distillation", IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[66] L. Yu, V. O. Yazici, X. Liu, J. v. d. Weijer, Y. Cheng, and A. Ramisa, "Learning metrics from teachers: Compact networks for image embedding", IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[67] F. Tung, and G. Mori, "Similarity-preserving knowledge distillation", IEEE/CVF International Conference on Computer Vision (ICCV), 2019.

[68] B. Peng, X. Jin, J. Liu, D. Li, Y. Wu, Y. Liu, S. Zhou, and Z. Zhang, "Correlation congruence for knowledge distillation", IEEE/CVF International Conference on Computer Vision (ICCV), 2019.

[69] S. Kim, D. Kim, M. Cho, and S. Kwak, "Embedding transfer with label relaxation for improved metric learning", IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.3967-3976, 2021.

[70] S. Lee, and B. C. Song, "Graph-based knowledge distillation by multi-head attention network", British Machine Vision Conference (BMVC), p.141, 2019.

[71] Y. Liu, J. Cao, B. Li, C. Yuan, W. Hu, Y. Li, and Y. Duan, "Knowledge distillation via instance relationship graph", IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[72] C. Lassance, M. Bontonou, G. B. Hacene, V. Gripon, J. Tang, and A. Ortega, "Deep geometric knowledge distillation with graphs", IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp.8484–8488, 2020.

[73] H. Yao, C. Zhang, Y. Wei, M. Jiang, S. Wang, J. Huang, N. V. Chawla, and Z. Li, "Graph few-shot learning via knowledge transfer", Association for the Advancement of Artificial Intelligence (AAAI), pp.6656–6663, 2020.

[74] J. Wang, X. Wang, B. Jin, J. Yan, W. Zhang, and H. Zha, "Heterogeneous graph-based knowledge transfer for generalized zero-shot learning", International Conference on Pattern Recognition (ICPR), pp.1859–1866, 2020.

[75] I. Chung, S. Park, J. Kim, and N. Kwak, "Feature-map-level online adversarial knowledge distillation", International Conference on Machine Learning (ICML), vol.119, pp.2006–2015, 2020.

[76] Y. Wang, A. Gonzalez-Garcia, D. Berga, L. Herranz, F. S. Khan, and J. v. d. Weijer, "Minegan: Effective knowledge transfer from gans to target domains with few images", IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[77] S. Roheda, B. S. Riggan, H. Krim, and L. Dai, "Cross-modality distillation: A case for conditional generative adversarial networks", IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp.2926–2930, 2018.

[78] M. Zhai, L. Chen, F. Tung, J. He, M. Nawhal, and G. Mori, "Lifelong gan: Continual learning for conditional image generation", IEEE/CVF International Conference on Computer Vision (ICCV), 2019.

[79] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets", Advances in Neural Information Processing Systems (NeurIPS), eds. Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, vol.27, 2014.

[80] J. H. Cho, and B. Hariharan, "On the efficacy of knowledge distillation", IEEE/CVF International Conference on Computer Vision (ICCV), pp.4793-4801, 2019.

[81] S. Hahn, and H. Choi, "Self-knowledge distillation in natural language processing", arXiv preprint arXiv:1908.01851, 2019.

[82] C. Yang, L. Xie, S. Qiao, and A. L. Yuille, “Training deep neural networks in generations: A more tolerant teacher educates better students”, Association for the Advancement of Artificial Intelligence (AAAI), pp.5628–5635, 2019.

[83] K. Clark, M. Luong, U. Khandelwal, C. D. Manning, and Q. V. Le, “Bam! born-again multi-task networks for natural language understanding”, Association for Computational Linguistics (ACL), eds. A. Korhonen, D. R. Traum, and L. Màrquez, pp.5931–5937, 2019.

[84] S. You, C. Xu, C. Xu, and D. Tao, “Learning from multiple teacher networks”, ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, p.1285–1294, 2017.

[85] I. Radosavovic, P. Dollár, R. Girshick, G. Gkioxari, and K. He, “Data distillation: Towards omni-supervised learning”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[86] G. Song, and W. Chai, “Collaborative learning for deep neural networks”, Advances in Neural Information Processing Systems (NeurIPS), pp.1837–1846, 2018.

[87] D. Chen, J. P. Mei, C. Wang, Y. Feng, and C. Chen, “Online knowledge distillation with diverse peers.”, Association for the Advancement of Artificial Intelligence (AAAI), pp.3430–3437, 2020.

[88] R. Anil, G. Pereyra, A. Passos, R. Ormandi, G. E. Dahl, and G. E. Hinton, “Large scale distributed neural network training through online distillation”, arXiv preprint arXiv:1804.03235, 2018.

[89] L. Gao, X. Lan, H. Mi, D. Feng, K. Xu, and Y. Peng, “Multistructure-based collaborative online distillation”, Entropy, vol.21, no.4, 2019.

[90] S. Hou, X. Liu, and Z. Wang, “Dualnet: Learn complementary features for image recognition”, IEEE International Conference on Computer Vision (ICCV), 2017.

[91] R. T. Mullapudi, S. Chen, K. Zhang, D. Ramanan, and K. Fatahalian, “Online model distillation for efficient video inference”, IEEE/CVF International Conference on Computer Vision (ICCV), 2019.

[92] A. Cioppa, A. Deliege, M. Istasse, C. De Vleeschouwer, and M. Van Droogenbroeck, “Arthus: Adaptive real-time human segmentation in sports through online distillation”, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2019.

[93] X. Lan, X. Zhu, and S. Gong, “Knowledge distillation by on-the-fly native ensemble”, Advances in Neural Information Processing Systems (NeurIPS), pp.7527–7537, 2018.

[94] T. B. Xu, and C. L. Liu, “Data-distortion guided self-distillation for deep neural networks”, Association for the Advancement of Artificial Intelligence (AAAI), vol.33, no.01, pp.5565-5572, 2019.

[95] H. Lee, S. J. Hwang, and J. Shin, "Self-supervised label augmentation via input transformations", International Conference on Machine Learning (ICML), vol.119, pp.5714–5724, 2020.

[96] L. Li, K. Jamieson, A. Rostamizadeh, E. Gonina, J. Ben-tzur, M. Hardt, B. Recht, and A. Talwalkar, "A system for massively parallel hyperparameter tuning", Proceedings of Machine Learning and Systems (MLSys), eds. I. S. Dhillon, D. S. Papailiopoulos, and V. Sze, vol.2, pp.230–246, 2020.

[97] A. Krizhevsky, and G. Hinton, "Learning multiple layers of features from tiny images", Technical report, Citeseer, 2009.

[98] "Tiny ImageNet Visual Recognition Challenge", `https://tiny-imagenet.herokuapp.com/`.

[99] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories", IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2004.

[100] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi, "Fine-grained visual classification of aircraft", arXiv, 2013.

[101] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization", 4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13), Sydney, Australia, 2013.

[102] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101 - mining discriminative components with random forests", European Conference on Computer Vision (ECCV), 2014.

[103] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar, "Cats and dogs", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012.

[104] M. E. Nilsback, and A. Zisserman, "Automated flower classification over a large number of classes", Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP), 2008.

[105] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010.

[106] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

[107] S. Zagoruyko, and N. Komodakis, "Wide residual networks", British Machine Vision Conference (BMVC), pp.87.1-87.12, 2016.

[108] P. Chen, S. Liu, H. Zhao, and J. Jia, “Distilling knowledge via knowledge review”, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.5008-5017, 2021.

[109] R. Miles, and K. Mikolajczyk, “Understanding the role of the projector in knowledge distillation”, AAAI Conference on Artificial Intelligence (AAAI), 2023.

[110] N. Passalis, and A. Tefas, “Learning deep representations with probabilistic knowledge transfer”, European Conference on Computer Vision (ECCV), 2018.

[111] L. Liu, Q. Huang, S. Lin, H. Xie, B. Wang, X. Chang, and X. Liang, “Exploring inter-channel correlation for diversity-preserved knowledge distillation”, IEEE/CVF International Conference on Computer Vision (ICCV), pp.8271-8280, 2021.

[112] B. Zhao, Q. Cui, R. Song, Y. Qiu, and J. Liang, “Decoupled knowledge distillation”, IEEE/CVF conference on computer vision and pattern recognition (CVPR), 2022.

[113] B. Zhao, Q. Cui, R. Song, and J. Liang, “Dot: A distillation-oriented trainer”, IEEE/CVF International Conference on Computer Vision (ICCV), pp.6189-6198, 2023.

[114] S. Sun, W. Ren, J. Li, R. Wang, and X. Cao, “Logit standardization in knowledge distillation”, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.15731-15740, 2024.

[115] L. Breiman, “Bagging predictors”, Machine Learning, vol.24, no.2, pp.123-140, 1996.

[116] Y. Freund, and R. E. Schapire, “Experiments with a new boosting algorithm”, International Conference on Machine Learning (ICML), pp.148–146, 1996.

[117] Y. Wen, D. Tran, and J. Ba, “Batchensemble: an alternative approach to efficient ensemble and lifelong learning”, International Conference on Learning Representations (ICLR), 2020.

[118] F. Wenzel, J. Snoek, D. Tran, and R. Jenatton, “Hyperparameter ensembles for robustness and uncertainty quantification”, Advances in Neural Information Processing Systems (NeurIPS), eds. H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, vol.33, pp.6514–6527, Curran Associates, Inc., 2020.

[119] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, and K. Q. Weinberger, “Snapshot ensembles: Train 1, get M for free”, International Conference on Learning Representations (ICLR), 2017.

[120] S. Laine, and T. Aila, “Temporal ensembling for semi-supervised learning”, International Conference on Learning Representations (ICLR), 2017.

[121] A. Dabouei, S. Soleymani, F. Taherkhani, J. Dawson, and N. M. Nasrabadi, "Exploiting joint robustness to adversarial perturbations", IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[122] A. Malinin, B. Mlodozeniec, and M. Gales, "Ensemble distribution distillation", International Conference on Learning Representations (ICLR), 2020.

[123] Z. Allen-Zhu, and Y. Li, "Towards understanding ensemble, knowledge distillation and self-distillation in deep learning", arXiv preprint arXiv:2012.09816, 2021.

[124] H. Fukui, T. Hirakawa, T. Yamashita, and H. Fujiyoshi, "Attention branch network: Learning of attention mechanism for visual explanation", IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[125] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei, "Novel dataset for fine-grained image categorization", First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.

[126] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset", Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[127] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn, "Model fusion via optimal transport", Advances in Neural Information Processing Systems, pp.5824–5836, 2020.

[128] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[129] L. McInnes, J. Healy, N. Saul, and L. Grossberger, "Umap: Uniform manifold approximation and projection", The Journal of Open Source Software, vol.3, no.29, p.861, 2018.

[130] S. Laine, and T. Aila, "Temporal ensembling for semi-supervised learning", International Conference on Learning Representations, pp.1–13, 2017.

[131] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "Mixmatch: A holistic approach to semi-supervised learning", Neural Information Processing Systems, 2019.

[132] T. Miyato, S. I. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: A regularization method for supervised and semi-supervised learning", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.41, no.8, pp.1979–1993, 2019.

[133] T. Suzuki, and I. Sato, "Adversarial transformations for semi-supervised learning", AAAI Conference on Artificial Intelligence, pp.5916–5923, 2020.

[134] A. Mehra, B. Kailkhura, P. Y. Chen, and J. Hamm, "Understanding the limits of unsupervised domain adaptation via data poisoning", Neural Information Processing Systems, pp.17347–17359, 2021.

[135] D. H. Lee, "Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks", International Conference on Machine Learning Workshop, 2013.

[136] A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, "Label propagation for deep semi-supervised learning", IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.5065–5074, 2019.

[137] E. Arazo, D. Ortego, P. Albert, N. E. O'Connor, and K. McGuinness, "Pseudo-labeling and confirmation bias in deep semi-supervised learning", International Joint Conference on Neural Networks, pp.1-8, 2020.

[138] Q. Xie, M. T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification", IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020.

[139] Y. Xu, L. Shang, J. Ye, Q. Qian, Y. F. Li, B. Sun, H. Li, and R. Jin, "Dash: Semi-supervised learning with dynamic thresholding", International Conference on Machine Learning, pp.11525–11536, 2021.

[140] B. Zhang, Y. Wang, W. Hou, H. Wu, J. Wang, M. Okumura, and T. Shinozaki, "Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling", Neural Information Processing Systems, pp.18408–18419, 2021.

[141] H. Chen, R. Tao, Y. Fan, Y. Wang, J. Wang, B. Schiele, X. Xie, B. Raj, and M. Savvides, "Softmatch: Addressing the quantity-quality tradeoff in semi-supervised learning", International Conference on Learning Representations, pp.1–21, 2023.

[142] Y. Wang, H. Chen, Q. Heng, W. Hou, Y. Fan, Z. Wu, J. Wang, M. Savvides, T. Shinozaki, B. Raj, B. Schiele, and X. Xie, "Freematch: Self-adaptive thresholding for semi-supervised learning", International Conference on Learning Representations, pp.1–20, 2023.

[143] Z. Hu, Z. Yang, X. Hu, and R. Nevatia, "Simple: Similar pseudo label exploitation for semi-supervised classification", IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.15099–15108, 2021.

[144] I. Nassar, S. Herath, E. Abbasnejad, W. Buntine, and G. Haffari, "All labels are not created equal: Enhancing semi-supervision via label grouping and co-training", IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.7241–7250, 2021.

[145] Y. Fan, A. Kukleva, and B. Schiele, "Revisiting consistency regularization for semi-supervised learning", DAGM German Conference, pp.63–78, 2021.

[146] J. Kim, Y. Min, D. Kim, G. Lee, J. Seo, K. Ryoo, and S. Kim, "Conmatch: Semi-supervised learning with confidence-guided consistency regularization", European Conference on Computer Vision, pp.674–690, 2022.

[147] B. Kim, J. Choo, Y. D. Kwon, S. Joe, S. Min, and Y. Gwon, "Selfmatch: Combining contrastive self-supervision and consistency for semi-supervised learning", Neural Information Processing Systems Workshop, 2020.

[148] F. Taherkhani, A. Dabouei, S. Soleymani, J. Dawson, and N. M. Nasrabadi, "Self-supervised wasserstein pseudo-labeling for semi-supervised image classification", IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.12267–12277, 2021.

[149] J. Li, C. Xiong, and S. C. Hoi, "Comatch: Semi-supervised learning with contrastive graph regularization", IEEE/CVF International Conference on Computer Vision, pp.9475–9484, 2021.

[150] T. Q. Tran, M. Kang, and D. Kim, "Rerankmatch: Semi-supervised learning with semantics-oriented similarity representation", International Joint Conference on Neural Networks, pp.1-8, 2021.

[151] S. Roy, and A. Etemad, "Scaling up semi-supervised learning with unconstrained unlabelled data", AAAI Conference on Artificial Intelligence, 2024.

[152] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space", IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp.3008–3017, 2020.

[153] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A Simple Framework for Contrastive Learning of Visual Representations", International Conference on Machine Learning, pp.1597–1607, 2020.

[154] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton, "Big Self-Supervised Models are Strong Semi-Supervised Learners", Neural Information Processing Systems, pp.22243–22255, 2020.

[155] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum Contrast for Unsupervised Visual Representation Learning", IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.9726–9735, 2020.

[156] X. Chen, H. Fan, R. Girshick, and K. He, "Improved Baselines with Momentum Contrastive Learning", arXiv preprint arXiv:2003.04297, 2020.

[157] J. B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko, "Bootstrap your own latent a new approach to self-supervised learning", Neural Information Processing Systems, pp.21271–21284, 2020.

[158] X. Chen, and K. He, “Exploring Simple Siamese Representation Learning”, IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.15750–15758, 2021.

[159] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, “Emerging Properties in Self-Supervised Vision Transformers”, IEEE/CVF International Conference on Computer Vision, pp.9630–9640, 2021.

[160] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting”, The Journal of Machine Learning Research, vol.15, no.1, pp.1929–1958, 2014.

[161] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, “Deep networks with stochastic depth”, European Conference on Computer Vision, pp.646–661, 2016.

[162] S. Zagoruyko, and N. Komodakis, “Wide residual networks”, British Machine Vision Conference, 2016.

[163] D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, and C. Raffel, “Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring”, International Conference on Learning Representations, 2020.

[164] D. Berthelot, R. Roelofs, K. Sohn, N. Carlini, and A. Kurakin, “Adamatch: A unified approach to semi-supervised learning and domain adaptation”, International Conference on Learning Representations, 2022.

[165] M. Zheng, S. You, L. Huang, F. Wang, C. Qian, and C. Xu, “Simmatch: Semi-supervised learning with similarity matching”, 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.14451-14461, 2022.

研究業績一覧

学術論文

[1] Soma Minami, Naoki Okamoto, Tsubasa Hirakawa, Takayoshi Yamashita, Hironobu Fujiyoshi, “Optimizing knowledge transfer graph for Deep Collaborative Learning,” IEEE Access, vol.13, pp.103565-103579, 2025.

[2] Naoki Okamoto, Tsubasa Hirakawa, Takayoshi Yamashita, Hironobu Fujiyoshi, “Distilling Diverse Knowledge for Deep Ensemble Learning,” IEEE Access, vol.13, pp.100875-100888, 2025.

国際会議発表論文 (査読あり)

[1] Naoto Hayashi, Naoki Okamoto, Tsubasa Hirakawa, Takayoshi Yamashita, Hironobu Fujiyoshi, “Diverse Data Selection Considering Data Distribution for Unsupervised Continual Learning,” International Conference on Computer Vision Theory and Applications (VISAPP), 2024.

[2] Naoki Okamoto, Tsubasa Hirakawa, Takayoshi Yamashita, Hironobu Fujiyoshi, “Deep ensemble learning by diverse knowledge distillation for fine-grained object classification,” European Conference on Computer Vision (ECCV), 2022.

[3] Shungo Fujii, Naoki Okamoto, Toshiki Seo, Tsubasa Hirakawa, Takayoshi Yamashita, Hironobu Fujiyoshi, “Super-class mixup for adjusting training data,” Asian Conference on Pattern Recognition (ACPR), 2021.

学会口頭発表 (査読あり)

[1] 岩野颯, 岡本直樹, 平川翼, 山下隆義, 藤吉弘亘, “最適輸送を用いた Attention ヘッドの知識蒸留,” 画像の認識・理解シンポジウム (MIRU), 2025.

[2] 西川実希, 岡本直樹, 平川翼, 山下隆義, 藤吉弘亘, “CLIP におけるモーダル間の知識蒸留による構造化枝刈り,” 画像の認識・理解シンポジウム (MIRU), 2025.

[3] 藤吉零, 岡本直樹, 平川翼, 山下隆義, 藤吉弘亘, “最適輸送コストを用いたモデルマージによるアンサンブル知識蒸留,” 画像の認識・理解シンポジウム (MIRU), 2025.

[4] 旅田海聖, 岡本直樹, 平川翼, 山下隆義, 藤吉弘亘, “未知度合いと把持量のばらつきを考慮したサンプリング最適化と把持位置選択による食品把持の高精度化,” 画像センシングシンポジウム (SSII), 2025.

[5] 村本佳隆, 岡本直樹, 平川翼, 山下隆義, 藤吉弘亘, “知識転移グラフによる最適な半教師あり共同学習の探索,” 画像の認識・理解シンポジウム (MIRU), 2023.

[6] 岩垣渚, 岡本直樹, 平川翼, 山下隆義, 藤吉弘亘, “SimSiam におけるデータ拡張の効果と mixup の導入による向上,” 画像の認識・理解シンポジウム (MIRU), 2023.

[7] 藤井駿伍, 岡本直樹, 瀬尾俊貴, 平川翼, 山下隆義, 藤吉弘亘, “スーパークラスを考慮した mixup,” 画像の認識・理解シンポジウム (MIRU), 2021.

[8] 岡本直樹, 南蒼馬, 平川翼, 山下隆義, 藤吉弘亘, “詳細画像識別における知識転移グラフによるアンサンブル学習,” 画像の認識・理解シンポジウム (MIRU), 2021.

[9] 村本佳隆, 岡本直樹, 平川翼, 山下隆義, 藤吉弘亘, “Refined Consistency による知識蒸留を用いた半教師あり学習,” 人工知能学会全国大会 (JSAI), 2021.

[10] 岡本直樹, 南蒼馬, 平川翼, 山下隆義, 藤吉弘亘, “知識転移グラフによるアンサンブル学習,” 画像の認識・理解シンポジウム (MIRU), 2020.

講演

[1] 岡本直樹, 藤吉弘亘, 平川翼, 山下隆義, 菅沼雅徳, “自己教師あり学習による事前学習”, CVIM 研究会, 2024.

著書

[1] 岡本直樹. “ニュウモン 自己教師あり学習による事前学習.” コンピュータビジョン最前線 Summer 2024. 井尻善久, 牛久祥孝, 片岡裕雄, 藤吉弘亘, 延原章平 編. 共立出版, 2024, p.86-130.

[2] 箕浦大晃, 岡本直樹. “フカヨミ DINO.” コンピュータビジョン最前線 Summer 2022. 井尻善久, 牛久祥孝, 片岡裕雄, 藤吉弘亘 編. 共立出版, 2022, p.57-73.

学術表彰

[1] 2025 年　画像の認識・理解シンポジウム　MIRU 論文評価貢献賞

[2] 2025 年　画像の認識・理解シンポジウム　MIRU インタラクティブ発表賞
題目: 最適輸送コストを用いたモデルマージによるアンサンブル知識蒸留

[3] 2021 年　情報処理学会 東海支部　学生論文奨励賞
題目: 詳細画像識別における知識転移グラフによるアンサンブル学習

[4] 2021 年　画像の認識・理解シンポジウム　MIRU 学生奨励賞
題目: 詳細画像識別における知識転移グラフによるアンサンブル学習