Part-Prototypeによる解釈可能な深層学習の研究動向

鵜飼 祐生*[†a)]　佐野 景介*[††,†††b)]　平川 翼[†††c)]　山下 隆義[†††d)]
藤吉 弘亘[†††e)]

Prototypical Part Interpretable Deep Learning: A Survey

Yuki UKAI*[†a)], Keisuke SANO*[††,†††b)], Tsubasa HIRAKAWA[†††c)],
Takayoshi YAMASHITA[†††d)], and Hironobu FUJIYOSHI[†††e)]

あらまし　深層学習はその識別性能の高さから様々な認識タスクへの応用が進められている．一方，深層学習モデルの推論過程は非常に複雑なため，人間に理解のできないブラックボックスとなっている．そのため，推論結果が重大な影響を及ぼすハイリスクドメインへ深層学習を適用することは難しい課題がある．この課題に対し深層学習モデルの解釈可能性を改善するための研究が活発に行われている．中でもPart-Prototypeを用いた事例ベースの推論による‘this looks like that’フレームワークが解釈可能な深層学習モデルを実現するアプローチとして幅広い注目を集めている．理論・応用の観点から数多くの‘this looks like that’フレームワークに基づき解釈可能性を実現する手法が発表されている．一方，これら手法を包括的かつ体系的にまとめたサーベイは存在しておらず，既存研究の成果は明確となっていない．そこで本論文ではこれら手法のサーベイを行い，技術的な進展や応用の観点から各手法を分類する．また従来研究では解決しきれていない未解決の問題を説明し，今後の展望について議論する．

キーワード　深層学習，説明可能性，解釈可能性，Prototypical Part Interpretability，‘This looks like that’フレームワーク

1. まえがき

深層学習は多くの認識タスクにおいて高い精度を達成しており，様々なアプリケーションへの応用が進められている．深層学習の推論過程は人間に理解できないほど複雑であり，事実上のブラックボックスとなっている．そのため，推論結果が重大な影響を及ぼすハイリスクドメインでは推論過程の透明性も求められるため，深層学習技術を適用することが難しい．例えば医療分野では病名の診断だけではなく，どのような医学的根拠からその病名と診断されたかを説明することがシステム要件として求められる．また2021年欧州連合（EU）に提出（2024年3月可決）されたAI規制法案[1]では，ハイリスクドメインにおけるAIシステムの透明性が要件に課されている．これらの社会的要請に加え，推論根拠の説明可能性はAIシステムの信頼性に大きな影響を与えることが報告されており[2]，深層学習モデルの解釈可能性は信頼できるAIシステムの社会実装に不可欠となっている．このような背景から，2018年以降深層学習モデルの推論過程を解釈・説明することを目的とした，深層学習の解釈可能性に関する研究が非常に活発に行われている．

解釈可能性に関する研究は，一般的に二つのアプローチに分類される．第一のアプローチはpost-hocなアプローチであり，学習済みの深層学習モデルの推論過程を解釈しようとするアプローチである．post-hocなアプローチでは，伝統的にモデルが入力データ内のどの部分を重視したかを出力することでモデルの推論根拠を説明するSaliency-basedな手法が多く採用され

† グローリー株式会社，姫路市
GLORY LTD., 1–3–1 Shimoteno, Himeji-shi, 670–8567 Japan
†† 株式会社デンソー，刈谷市
DENSO CORPORATION, 1–1 Showa-cho, Kariya-shi, 448–8661 Japan
††† 中部大学，春日井市
Faculty of Engineering, Chubu University, 1200 Matsumoto-cho, Kasugai-shi, 487–0027 Japan
a) E-mail: y.ukai@mail.glory.co.jp
b) E-mail: keisuke.sano.j5s@jp.denso.com
c) E-mail: hirakawa@mprg.cs.chubu.ac.jp
d) E-mail: takayoshi@isc.chubu.ac.jp
e) E-mail: fujiyoshi@isc.chubu.ac.jp
DOI:10.14923/transinfj.2024JDR0001
* equal contribution

る [3]〜[6]．Saliency-based な手法ではヒートマップという直感的に理解しやすい形式で説明を行うことができる利点がある．またより詳細にモデルの推論過程を理解するため，モデルがどのような特徴を捉えたかを解析しようとする Concept-based な手法も提案されている [7]〜[9]．post-hoc なアプローチは既存のモデルを再学習する必要なく，モデルの推論根拠を説明できる利点がある．一方で post-hoc なアプローチにより生成される説明は，モデルの推論過程と何ら関係がない場合があり，忠実性に課題を抱えていることが提起されている [10]．

第二のアプローチは Interpretable-by-design (ante-hoc) なアプローチである．ante-hoc なアプローチは解釈可能性の実現のため，解釈可能なモデルを構築・学習することを目的とする．解釈可能なモデルでは推論根拠の説明がモデルの推論過程と密接に関連するため，忠実性が保証される利点がある．このような解釈可能なモデルを実現する一手法として，‘this looks like that’ フレームワーク [11] が挙げられる．‘this looks like that’ フレームワークは事例ベースの推論に基づく解釈可能性を実現する．ここで事例ベースの推論とは，学習データ等既知のデータ群内の入力データと類似するデータを用いて推論を行う手法であり，事例ベースの推論では “B は A と似ており，A は Y の原因であるから B も同様に Y を引き起こす” という，画像や文章などで有効となるアナロジーに基づく説明が可能となる [12]．事例ベースの推論では提示されるデータの意味を人間が理解できれば有効な説明を提供できる利点がある．また全てのデータを用いる代わりにプロトタイプと呼ばれる代表的なデータインスタンスを用いて推論を行うことで，全てのデータと入力データの類似度の計算を行う必要なく効率的に推論することが可能となる．‘this looks like that’ フレームワークはこれらのプロトタイプを用いて推論を行う手法の内，学習データ内のあるインスタンス全体ではなくデータ内のある部分領域をプロトタイプとして用いて事例ベースの推論を行う手法である．‘this looks like that’ フレームワークにおいて用いられるプロトタイプは特に Part-Prototype と呼称され，データインスタンス全体のプロトタイプとは区別される．以降本論文で用いられるプロトタイプは全て Part-Prototype を指すことに注意されたい．‘this looks like that’ フレームワークではデータ内の部分領域を用いることから，入力データと似ているデータを提示する以上の詳細なモデル推論根拠の説明が可能となる．より詳細な ‘this looks like that’ フレームワークにより実現される推論根拠の説明とその解釈方法は **2. 3** で説明する．

事例ベースの推論と同様に，‘this looks like that’ フレームワークによる解釈可能性の実現には説明において提示されるプロトタイプ，すなわちデータ内の部分領域の意味が人間に理解できることのみが必要となる．そのため，‘this looks like that’ フレームワークは様々なドメインのデータや認識タスクに適用可能な汎用性の高いフレームワークとなる．加えて ‘this looks like that’ フレームワークにはクラスラベル以上の教師ラベルを必要としない利点がある．このような背景から ‘this looks like that’ フレームワークは幅広い注目を集めており，図 1 に示すように様々な手法が多く提案されている．しかし，これら手法を包括的にまとめたサーベイは筆者の知る限り存在しておらず，既存研究の成果や未解決の問題は明確となっていない．そこで本論文では，‘this looks like that’ フレームワークによる解釈可能性を実現する研究の成果と未解決の課題を把握することを目的に，これら手法の整理と要約を試みる．

本論文の構成は以下のとおりである．まず **2.** では，後続の章の準備として ‘this looks like that’ フレームワークを初めて実現した手法である Prototypical Part Network (ProtoPNet) のモデル構造，学習方法及び ProtoPNet による推論根拠説明の解釈方法について説明する．次に **3.** では ProtoPNet に存在する課題をまとめ，それらを改良した後続研究（本論文では Prototypical Part Models (ProtoPModels) と呼称する）について説明する．その後 **4.** では ProtoPModels の応用研究を適用先ドメイン及び適用タスクの二つの観点から整理し説明する．また Transformer を始めとする基盤モデルを活用した研究について述べる．**5.** では ProtoPModels により学習されるプロトタイプの解釈性に関する議論や評価指標について説明する．最後に **6.** で本論文のまとめを行い，今後の展望を述べる．

1. 1 関連するサーベイと本論文の意義

「説明」は説明相手に応じて理想的な説明の仕方が変わる．そのため，解釈可能性に関する研究は，対象となるドメインに加え，どのような説明を行うかという点においても非常に多岐にわたる．このような背景から，包括的な観点 [13] に加え，あるドメインのデータ [14] や説明方法 [15] に重点をおいて解釈可能性の研究を分類したサーベイが多く発表されてい

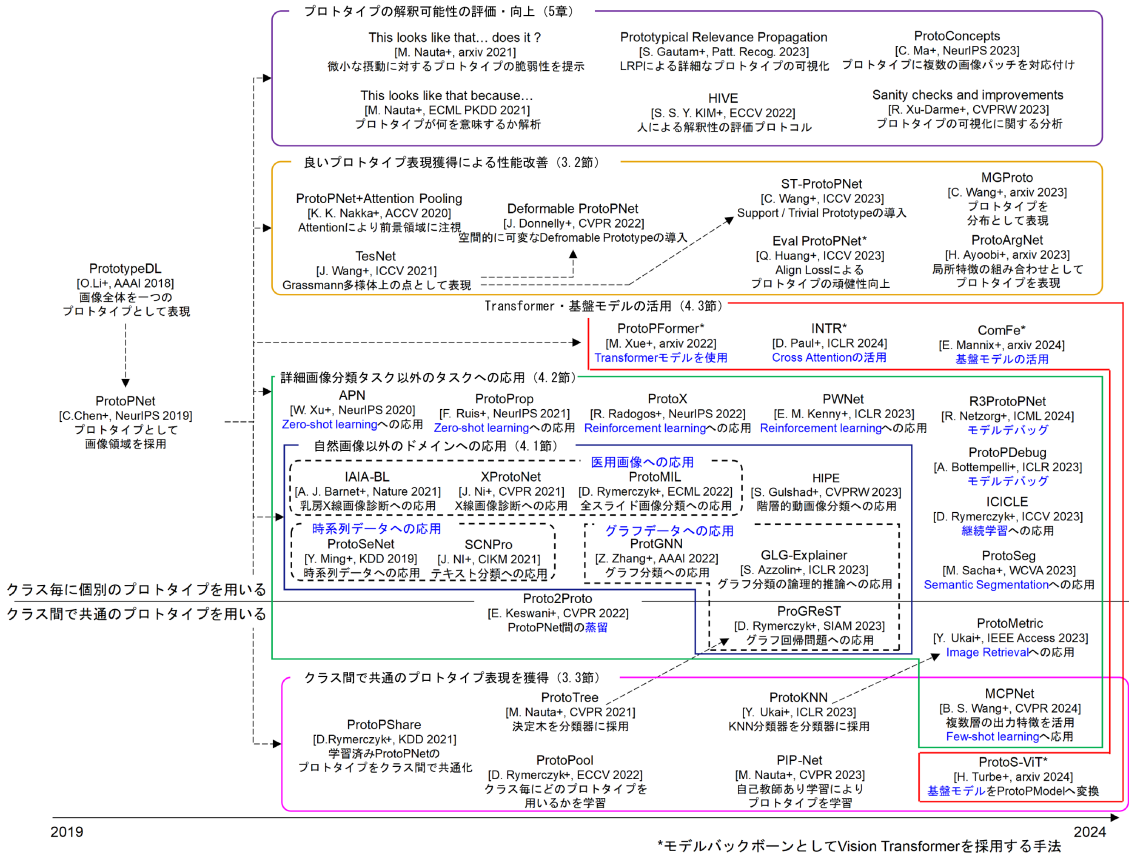

図 1　Prototypical Part Models の相関図

る [16]〜[19]．これらサーベイのうち本論文と最も関連するサーベイとして，事例ベースの推論により解釈可能性を実現する手法を体系的にまとめた Poché らによるサーベイ [15] が挙げられる．しかし，彼らのサーベイはあるデータ全体を推論根拠として用いる手法を主眼にしているため，データの部分領域を推論根拠として用いる ProtoPModels に関しては高々数本の関連研究が簡単に触れられる程度となっている．

以上のように，従来のサーベイでは ProtoPModels に関する調査は十分行われているとは言えない．一方で，ProtoPModels による ‘this looks like that’ フレームワークは解釈可能なモデルを実現するための主流なアプローチの一つとなっており，数多くの手法が提案されている重要度の高いトピックと言える．そのため，ProtoPModels の既存研究を整理し，その範囲を明確にすることは，後続の新たな研究を触発する意味でも意義深いと考えられる．そこで本論文では，従来十分整理されてこなかった ProtoPModels の研究について整理を行い，今後の ProtoPModels に関する研究について議論する．

2. Prototypical Part Network

本章では ‘this looks like that’ フレームワークを初めて実現した手法である Prototypical Part Network (ProtoPNet) [11] について説明する．本章ではまず．ProtoPNet のモデル構造とその学習方法について説明する．その後 ProtoPNet により実現される推論根拠の解釈方法について説明し，他の解釈手法に対するプロトタイプを用いる説明手法の利点をまとめる．

2.1 ProtoPNet のモデル構造

図 2 に ProtoPNet のモデル構造を示す．ProtoPNet は Feature Extractor，Prototype Layer 及び線形層（Fully Connected (FC) Layer）の三つの層から構成される．ただし図 2 において Prototype Layer の色付きのブロックは各ブロックが異なるクラスに属するプロトタイプであることを示している．Prototype Layer の右側に例

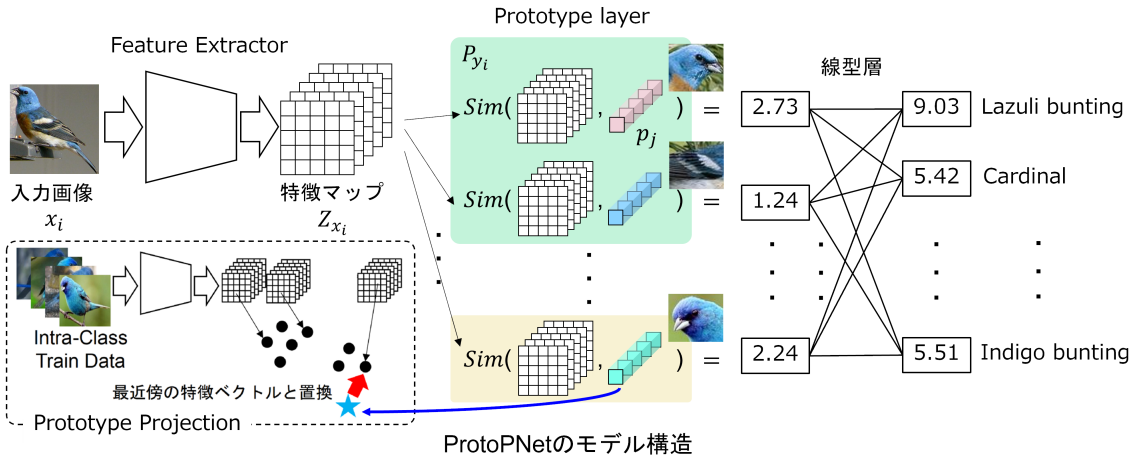

図 2　ProtoPNet のモデル構造

表 1　詳細画像分類における ProtoPModels の分類

分類	手法名	発表年	プロトタイプと特徴マップの類似度指標	分類器
クラスごとに個別のプロトタイプを学習	ProtoPNet [11]	2019	Log Inverse	線形分類
	ProtoPNet + Attention Pooling [24]	2020	Log Inverse	線形分類
	Sparrow [25]	2021	Log Inverse	線形分類
	Tesnet [26]	2021	Projection Metric	線形分類
	Deformable ProtoPNet [27]	2022	Cosine Similarity	線形分類
	Eval ProtoPNet [21]	2023	Projection Metric	線形分類
	Support-Trivial ProtoPNet [28]	2023	Projection Metric / Cosine Similarity (注1)	線形分類
	MGProto [29]	2023	Gaussian Kernel	生成的分類
クラス間で共通のプロトタイプを学習	ProtoPShare [30]	2021	Log Inverse	線形分類
	ProtoTree [31]	2021	Negative Exponential	決定木
	ProtoPool [32]	2022	Log Inverse	線形分類
	ProtoKNN [33]	2023	Cosine Similarity	K 近傍分類
	ProtoMetric [34]	2023	Cosine Similarity	最近傍分類
	PIPNet [22]	2023	Softmax	線形分類
	MCPNet [35]	2024	Cosine Similarity	最近傍分類

示した出力の数値は各プロトタイプと特徴マップとの類似度であり，各類似度が線形層（FC Layer）によりクラスロジット（図内線形層の右側の数値）に変換されることを表している．また Prototype Projection は ProtoPNet を学習する際に各プロトタイプを学習データ内の最も類似する画像パッチに接地する操作であり，**2.2** で詳細に説明される．以下では ProtoPNet を構成する三つの層についてそれぞれ説明する．

Feature Extractor は入力画像 x_i を特徴マップ Z_{x_i} に変換する層である．ProtoPNet 及びその後続研究では Feature Extractor として畳み込みニューラルネットワーク（CNN）が用いられることが多い．また 2023 年後半以降の研究では Vision Transformer (ViT) [20] を Feature Extractor として採用した研究 [21]〜[23] が提案されている．

続く Prototype Layer はモデル内の重みとして実装されるプロトタイプと特徴マップの類似度を算出する．ProtoPNet ではプロトタイプ p_j と特徴マップ Z_{x_i} の類似度 $S(\mathrm{Z}_{x_i}, p_j)$ は次式により算出される．

$$
\begin{aligned}
S(\mathrm{Z}_{x_i}, p_j) &= \max_{z \in \mathrm{Z}_{x_i}} \theta(z, p_j), \\
\theta(z, p_j) &= \log\left(1 + \frac{1}{\|z - p_j\|_2^2 + \epsilon}\right)
\end{aligned}
\tag{1}
$$

ただし，z は特徴マップの各ピクセルに含まれる特徴ベクトルであり，ϵ は数値計算の安定化のため加算する微小な定数である．表 1 に示すように，ProtoPModels では上記の Log Inverse [11], [32] のほかに Projection Metric [26], [28] やコサイン類似度 [27], [33], [35] 等の

（注1）：Support Trivial ProtoPNet では画像全体を用いる実験設定と画像内の対象領域を切り出す実験設定で異なる類似度指標を用いている．

類似度指標が用いられる．ProtoPNet ではクラスごとに複数（10 個）定義されたプロトタイプがクラス固有の特徴を代表するよう学習する．特に各プロトタイプは学習データ内のある画像領域と特徴空間上で一致（接地）するように学習される．したがって Prototype Layer より出力される特徴マップとプロトタイプの類似度は，入力された画像がプロトタイプの代表する学習データ内の画像領域と類似する画像特徴をどの程度もっているかに関する尺度として解釈することができる．

最後に線形層ではプロトタイプと特徴マップの類似度を入力とし，クラス分類予測を行う．ここで線形層による分類予測には線形分類モデルが採用される．そのため，ProtoPNet のクラス分類は（1）入力された画像が学習データ内の画像特徴と類似する特徴をどの程度もっており，（2）それら特徴がクラス予測へどのように寄与するかを解釈できる，という意味で解釈可能となる．すなわち ProtoPNet の解釈性は学習データ中のある画像領域を代表するよう学習されたプロトタイプにより実現されていると言える．続く **2.2** ではランダムなモデル重みとして初期化されるプロトタイプが，ProtoPNet においてどのように学習データ内のある画像領域を代表するように学習されるかについて述べる．

2.2 ProtoPNet の学習方法

本節では ProtoPNet の学習方法について説明する．ProtoPNet の学習では Prototype Layer と Feature Extractor を学習した後，線形層をファインチューニングする二段階の学習方式を採用する．以下では Prototype Layer と Feature Extractor の学習について説明した後線形層のファインチューニングについて説明する．

ProtoPNet はクラスごとに複数個のプロトタイプを定義し，各プロトタイプがそれらの所属するクラスに特徴的な画像特徴を代表するよう学習する．この目的のため Prototype Layer と Feature Extractor の学習では Cross Entropy Loss L_{ce} に加え，Cluster Loss L_{clst} 及び Separation Loss L_{sep} と呼ばれる二つの損失関数を最小化する．図 3 に Cluster Loss 及び Separation Loss の模式図を示す．ただし図 3 では，入力画像 x_i を Feature Extractor に入力することで得られる特徴マップ Z_{x_i} 内の各ピクセルに含まれる特徴量（画像パッチ特徴）を黒い点で，入力画像クラスに所属するプロトタイプ $p_j \in \mathrm{P}_{y_i}$ を青い星マーク，入力画像クラスと異なるクラスに所属するプロトタイプ $p_k \notin \mathrm{P}_{y_i}$ を赤い星マーク

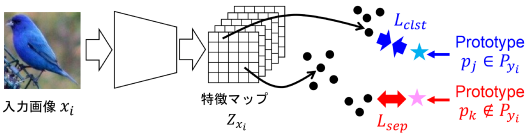

図 3　Cluster Loss L_{clst} 及び Separation Loss L_{sep} の模式図

で表している．図 3 では Cluster Loss は各画像パッチ特徴のうち，入力画像クラスに所属するプロトタイプ $p_j \in \mathrm{P}_{y_i}$ と最も近い画像パッチ特徴をプロトタイプ p_j と近づける損失関数であることを内向きの青い矢印で示している．また Separation Loss は Cluster Loss とは対照的に，各画像パッチ特徴のうち入力画像クラスと異なるクラスに所属するプロトタイプ $p_k \notin \mathrm{P}_{y_i}$ と最も近い画像パッチ特徴をプロトタイプ p_k から遠ざける損失関数であることを外向きの赤い双方向の矢印で表している．より具体的には各損失関数はバッチ内に含まれるデータインデックスの集合を B として次式のように定式化される．

$$L_{ce} = -\frac{1}{|\mathrm{B}|}\sum_{i\in\mathrm{B}} \log p(y_i|x_i) \tag{2}$$

$$L_{clst} = \frac{1}{|\mathrm{B}|}\sum_{i\in\mathrm{B}} \min_{p\in\mathrm{P}_{y_i}} \min_{z\in\mathrm{Z}_{x_i}} \|z-p\|_2^2 \tag{3}$$

$$L_{sep} = -\frac{1}{|\mathrm{B}|}\sum_{i\in\mathrm{B}} \min_{p\notin\mathrm{P}_{y_i}} \min_{z\in\mathrm{Z}_{x_i}} \|z-p\|_2^2 \tag{4}$$

特に，Prototype Layer と Feature Extractor の学習時には線形層の重み W はプロトタイプの所属するクラスに対しては 1.0，その他のクラスに対しては −0.5 で固定される．このように初期化された線形重みを用いて ProtoPNet を学習することで，各プロトタイプがそれらの所属するクラスを認識するために重要な特徴を代表するよう学習できる．更に ProtoPNet では Feature Extractor と Prototype Layer の学習の終了時に次式で表される Prototype Projection と呼ばれる操作が実行される．

$$p_j \leftarrow \underset{z\in\mathrm{Z}_{x_i}, i\in\mathrm{D}}{\arg\max}\ \theta(z, p_j) \tag{5}$$

ただし，D は学習データセット内に含まれる全てのデータインデックスの集合である．式 (5) より Prototype Projection は各プロトタイプを学習データ内の最も類似する画像パッチと置き換える操作と言える．そこでプロトタイプは学習データ内のある画像パッチに接地

される．そのためプロトタイプと特徴マップの類似度は入力画像が学習データ内のある画像領域と似た特徴をどの程度有しているかに関する指標として解釈することができるようになる．

続く線形層のファインチューニングでは，Cross Entropy 損失を最小化するよう線形層を学習する．ただし，異なるクラスに属するプロトタイプが推論に寄与しないように L_{ce} に加え，L1 損失 $L_{l1} = \sum_{y \in \mathrm{Y}, j \notin \mathrm{P}_y} |\mathrm{W}_{y,j}|$ が課される．

2.3 ProtoPNet による推論根拠の解釈

本節では具体例を交えながら ProtoPNet で提示される推論根拠の解釈方法について説明する．図 4 に ProtoPNet による推論根拠の説明例を示す．ただし図 4 内の各行における Training Image は各プロトタイプを Prototype Projection する際に選択された学習データ内の画像である．ここで Training Image 内の矩形は Prototype Projection 後のプロトタイプと画像間の類似度の 95 パーセンタイルを含む最小の矩形であり，矩形で囲まれた学習データ内の画像部分領域がプロトタイプとして提示される．また図 4 内における Activation Map は入力画像と各プロトタイプの類似度をヒートマップとして可視化した結果であり，各行左端の Original Image における矩形は入力画像と各プロトタイプの類似度の 95 パーセンタイルを含む最小の矩形を表している．そこで Original Image 内の矩形により，入力画像内のどの領域が各プロトタイプと似ているかが可視化される．

図 4 は ProtoPNet が入力画像である図左端の画像を “red-bellied woodpecker” と分類した推論根拠の説明となっており，クラススコアに寄与するプロトタイプが降順に並べられている．そこで図 4 より，入力画像を “red-bellied woodpecker” と分類する最大の推論根拠はプロトタイプとして提示される赤い頭であり，プロトタイプと入力画像の類似度（図内 Similarity Score）$S(\mathrm{Z}_{x_i}, p_j)$ と各プロトタイプのクラスに対する重み（図内 Class Connection）$\mathrm{W}_{y_i,j}$ との積 7.669 だけクラススコアに寄与していることがわかる．同様に各行について繰り返せば，“red-bellied woodpecker” に対するクラススコアを入力画像と各プロトタイプの類似度の重みづけ和として分解することができる．そこで ProtoPNet により，Original Image 内の矩形により示される「この (this)」部分領域が各プロトタイプにより代表される学習データ内の「その (that)」部分領域と「似ている (looks like)」ためにあるクラスに分類される，という説明が推論根拠として提示されると確認できる．

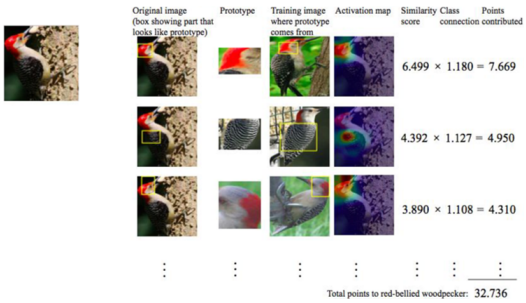

図 4　ProtoPNet によるクラス分類予測の説明例．図は [11] より抜粋，一部改変

2.4 プロトタイプを用いる説明の利点

本節では他の解釈手法に対する ProtoPNet によるプロトタイプを用いた説明の優位性について説明する．深層学習における解釈可能性に関する手法では，入力（画像）データ内のどの部分領域が重要かを説明として出力する，Saliency-based な手法が post-hoc，ante-hoc なアプローチによらず多く採用される [4], [6], [36]．これらの手法では画像内のどの領域が重要かを直感的に示すことができる利点がある一方で，画像内のどのような特徴が重要なのかを説明できない課題がある．一方で ProtoPNet ではプロトタイプを通して画像内のどの部分領域が重要であるかに加えて，どのような画像特徴が推論にとって重要なのかまで踏み込んで説明することができる．そのため ProtoPNet には，説明の詳細性という点で Saliency-based な手法に対し優位性があると言える．

またプロトタイプを用いる手法とは別に，どのような画像特徴が含まれるかがアノテーションされたデータセット（プローブデータセット）を用いてニューロンの発火パターンと言語化された画像特徴（コンセプト）を紐づけて説明を構成する，Concept-based な手法が多く提案されている [8], [37]．Concept-based な手法では言語的にどのような特徴が重要であるかを明確に説明できる利点がある．一方で Concept-based な手法では，プローブデータセット次第でニューロンの意味が異なったものになる [38] 等，アノテーションコストの大きいプローブデータセットを「適切に」用意する必要がある課題がある．また，正確に言語的に表現することが難しい画像特徴は扱うことができないことに加え，コンセプトを回帰するモデルを学習することは

表 2　ProtoPNet と他の解釈可能性手法の比較

手法名	説明の忠実性が保証される (Ante-hoc な手法)	推論に重要な画像領域を提示可能	推論に重要な画像特徴を説明可能	追加のデータセットが説明のために必要とならない
CAM [4]		✓		✓
Grad-CAM [5]		✓		✓
LRP [39]		✓		✓
TCAV [8]			✓	
ACE [9]			✓	✓
ABN [36]	✓	✓		✓
CBM [37]	✓		✓	
SENN [40]	✓		✓	✓
ProtoPNet [11]	✓	✓	✓	✓

クラスラベルを学習するより難しい場合があることが示唆されている [38]．これに対しプロトタイプを用いる手法では，画像領域をプロトタイプとして視覚的に提示することで言語的に表現の難しい画像特徴も扱うことが可能となる．また ProtoPNet は特別なプローブデータセットを用いることなく学習することが可能な点にも注意されたい．

以上まとめると，ProtoPNet による説明の他の解釈手法に対する優位性は表 2 のようにまとめられる．すなわち，ProtoPNet は解釈可能なモデルを構築・学習する ante-hoc なアプローチを採用するため，第一に（1）推論根拠説明の忠実性が保証される利点がある [10]．加えて，ProtoPNet によるプロトタイプを用いる説明手法には，（2）推論に重要な画像領域だけでなく，（3）どのような画像特徴が推論に重要なのかまでを（4）特別なアノテーションのされた追加のデータセットを用いる必要なく説明できる利点がある，とまとめられる．

3. ProtoPModels の進展

2. で説明したように，ProtoPNet は学習データ内の部分領域に基づき事例ベースの推論を行うことで解釈可能性を実現した手法であり，ProtoPModels として多数の後続研究が登場している．図 1 に ProtoPModels の相関図を示す．ProtoPModels はクラスごとに個別のプロトタイプを用いる手法とクラス間で共通のプロトタイプを用いる手法に大別でき，その目的に応じて更に体系分類することができる．体系分類の第一のグループは本章で説明する ProtoPNet 自体の問題を改善した手法群であり，更に良いプロトタイプ表現獲得に向けて改善を行った手法とクラス間で共有のプロトタイプ表現を獲得する手法とに分けられる．第二のグループは ProtoPNet を詳細画像分類以外のタスクやドメインに応用する手法群であり，自然画像以外のドメインへの応用，詳細分類タスク以外のタスクへの応用，Transformer・基盤モデルの活用に分けられる．これらの手法については **4.** で説明する．最後のグループはプロトタイプの解釈可能性の評価，及びプロトタイプの可視化を改善することで解釈可能性を改善した手法群であり，これらの手法は **5.** で説明する．なお，図 1 に掲載の研究は本論文で解説する全ての ProtoPModels ではなく，その中から重要と思われる手法を抽出したものである．

本章では，まず ProtoPNet が抱える課題を四つのカテゴリーに分類し説明した後（**3. 1**），各課題を解決するため提案された手法をそのアプローチごとに体系分類し説明する（**3. 2**，**3. 3**）．

3. 1　ProtoPNet の課題と改善手法

本節では，ProtoPNet が抱える課題をまとめる．表 3 に本章で説明する手法及び解決しようとする課題とその分類，改善のためのアプローチの対応を示す．表 3 の大分類に示す通り，ProtoPNet の課題は次の四つに分類できる：（1）学習されたプロトタイプが解釈性の観点で望ましくない性質をもつ場合がある，（2）学習されたプロトタイプがクラス分類に有効でない場合がある，（3）クラス間でプロトタイプを共有できずメモリ効率が悪い，（4）分類器が線形モデルに限られる．

課題（1）及び（2）は，学習により得られたプロトタイプの性質に焦点を当てた課題となっている．前者は説明のプロトタイプと画像パッチとの類似度の挙動に関する問題であり，説明の解釈性に関する課題となる．より具体的には，クラス内で挙動が類似したプロトタイプが発生し説明が冗長となる，異なる意味の画像パッチそれぞれに対し類似度が高いプロトタイプが存在し説明の解釈が難しくなる，プロトタイプと入力画像の類似度がわずかな画像ノイズに対し脆弱であり説明の頑健性が低い，またそれらの原因として特徴空間上でのデータ分布を考慮できていないといった課題が挙げられる．また後者はクラス分類性能の向上に関

表 3　ProtoPNet の課題と改善手法の対応

大別	手法	手法が解決しようとする課題	課題の大分類	解決へのアプローチ
クラスごとに個別のプロトタイプを学習	TesNet [26]	類似プロトタイプの発生	(1) 学習されたプロトタイプが解釈性の観点で望ましくない性質を持つ場合がある	正則化損失による性質改善 (**3.2.1**)
	EvalProtoPNet [21]	異なる意味の画像パッチが同一プロトタイプで高い類似度となる		
		プロトタイプの発火が摂動に脆弱		
	MGProto [29]	特徴空間におけるデータ分布が考慮されない	(2) 学習されたプロトタイプがクラス分類に有効でない場合がある	Prototype Layer の改善 (**3.2.2**)
	Deformable ProtoPNet [27]	画像特徴の空間的な変形に対応できない		
	ST ProtoPNet [28]	クラス境界に近い有効特徴が学習されにくい		
クラス間で共通のプロトタイプを獲得	ProtoPShare [30]	クラス間で類似したプロトタイプが存在	(3) クラス間でプロトタイプを共有できずメモリ効率が悪い	線形層で使用するプロトタイプ削減 (**3.3.1**)
	ProtoPool [32]	プロトタイプとクラスの対応を End-to-End で学習できない		
	PIP-Net [22]	クラス非依存かつ単一の概念を表現するプロトタイプを学習できない		
	ProtoTree [31]	決定木による分類に対応していない	(4) 分類モデルが線形分類モデルに限られる	線形層以外の分類モデル学習 (**3.3.2**)
	ProtoKNN [33]	k 近傍法による分類に対応していない		

する問題であり，データ分布を考慮できないことに加え，入力画像における様々な幾何変形に対応ができない，クラス分類境界に近い有効な特徴を捉えるプロトタイプが学習されにくいといった課題である．表 3 に示すとおり，課題（1）に対しては正則化損失を活用するアプローチ，課題（2）に対しては Prototype Layer の改善により解決を図るアプローチが主に採用されている．本論文ではこれら手法を良いプロトタイプ表現を獲得するための改善手法として **3.2** にまとめ，アプローチ別に解説する．

課題（3）は，プロトタイプのクラス間共有に関する問題である．ProtoPNet は，モデルの構造上クラスごとに個別に定数個のプロトタイプを用意するため，モデルあたりのプロトタイプの総数がクラス数に応じて増加する．そのためクラス間でプロトタイプを共有できずメモリ効率が悪いという課題がある．また，クラス間で共通した特徴が識別に用いられるため，クラス別に固有のプロトタイプを用意すると，獲得されるプロトタイプが冗長になる場合がある．これらの課題に対して，クラス間でプロトタイプを共有し，線型層で使用されるプロトタイプ数を削減することを目的とした手法が提案されている．これらの手法ではプロトタイプとクラスの対応関係の効率的な学習方法やクラス間で共通の特徴を捉えるプロトタイプの獲得方法の確立が課題となる．また課題（4）は，ProtoPNet の分類方式が線形モデルのみに限定されており，線形モデル以外の解釈可能性が高い分類器である決定木や k 近傍法に対応できていない点を課題としている．これらの分類器を ProtoPNet で利用するためには，クラス間で共通したプロトタイプを用いることが前提となる．そのため課題（4）を解決するにあたっては，各分類器のもとでいかにクラス間で共通したプロトタイプを獲得するよう学習するかが課題となる．そこで課題（3）及び（4）に対する改善手法は，クラス間で共通のプロトタイプ表現を獲得する手法としてまとめられる．本論文ではこれら手法を **3.3** にまとめ，各アプローチの手法についてそれぞれ解説する．

3.2　良いプロトタイプ表現獲得による性質改善

3.1 で説明したように，ProtoPNet の学習の結果得られるプロトタイプは，解釈性やクラス分類への有効性の観点から望ましくない性質をもつ場合がある．これらのプロトタイプの性質に関する課題に対しては，正則化損失によりプロトタイプの性質を改善する手法と Prototype Layer の改善によりプロトタイプの性質を改善する手法が提案されている．本節では各手法について述べる．

3.2.1　正則化損失による性質改善

図 5 に示すように，プロトタイプベクトルや Feature Extractor に対して正則化損失を課すことによりプロトタイプの性質を改善した手法として，Tesnet [26] と EvalProtoPNet [21] が挙げられる．ただし図 5 において，灰色領域は ProtoPNet を示し，ProtoPNet 上に示す赤枠は各手法において正則化損失を課す対象を表す．以下ではこれらの手法について述べる．

Tesnet は異なるプロトタイプが学習の結果似た値となり，重複したプロトタイプが学習される課題に対処することを目的とする．そのため Tesnet では，図 5 左のとおり，プロトタイプベクトルに対して式 (6) に示

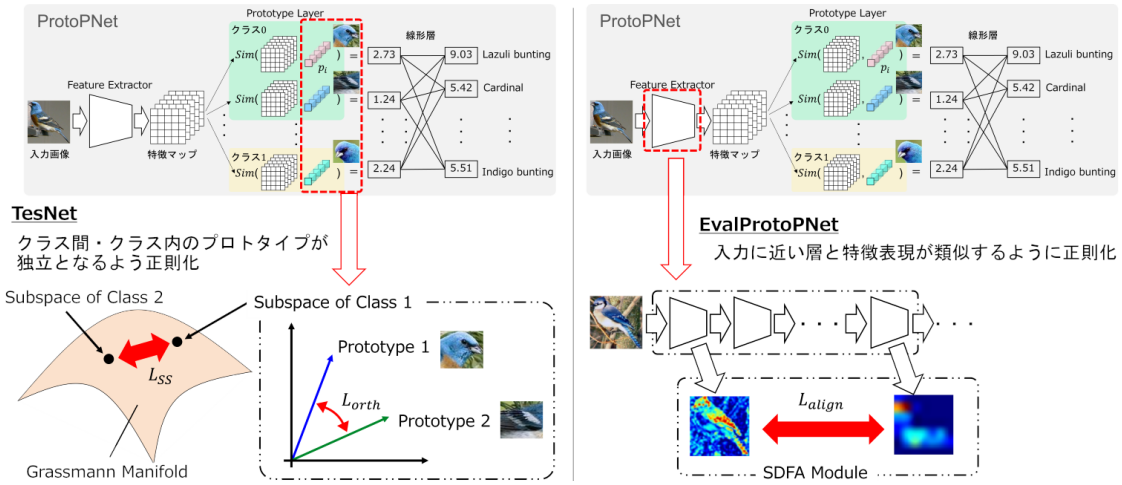

図 5　正則化によりプロトタイプの性質を改善する手法の図解

す正則化損失を課し学習を行う．具体的には，図 5 左内の点線枠内に示すとおりクラス内のプロトタイプが互いに直交・独立するように L_{orth} を課す．また，図 5 左下に示すとおり各クラスごとのプロトタイプがなす部分空間を分離されるように L_{SS} を課す．

$$
\begin{aligned}
L_{orth} &= \sum_{y \in \mathrm{Y}} \|\mathrm{P}_y \mathrm{P}_y^\top - \mathrm{I}\|_F^2 \\
L_{ss} &= \sum_{\substack{y_1, y_2 \in \mathrm{Y} \\ y_1 < y_2}} \frac{-1}{\sqrt{2}} \|\mathrm{P}_{y_1}^\top \mathrm{P}_{y_1} - \mathrm{P}_{y_2}^\top \mathrm{P}_{y_2}\|_F
\end{aligned}
\tag{6}
$$

ここで I 及び $\mathrm{P}_y \in \mathbb{R}^{|\mathrm{P}_y| \times C}$ は単位行列及びクラス y に属するプロトタイプを各行の成分とする行列であり，$\|\cdot\|_F$ は Frobenius ノルムである．Tesnet では各プロトタイプは L2 正規化されるため，L_{orth} によりクラス内の各プロトタイプは互いに独立な基底ベクトルとなるよう学習される．また L_{ss} における $\mathrm{P}_y^\top \mathrm{P}_y \in \mathbb{R}^{C \times C}$ はクラス y に属するプロトタイプで張られる部分空間に含まれる次元が非ゼロの値をとる対称行列となる．すなわち L_{ss} は各クラスに属するプロトタイプで張られる部分空間が重ならない場合に最小となる．そこで L_{orth} 及び L_{ss} の最小化により重複したプロトタイプの学習を抑制することができる．

EvalProtoPNet は（A）同一のプロトタイプに対し，異なる意味をもつ画像パッチが高い類似度をもつ場合がある，（B）プロトタイプの発火領域がわずかな摂動に対し脆弱となる，という二つの課題への対処を目的とする．課題（A）及び課題（B）は深層学習において層を重ねるごとに深刻化する．実際課題（A）について Huang らは深い層と浅い層の出力を比較し，深い層において空間的な情報が失われることを実験的に観察した．また課題（B）に関しては層を重ねるにつれて Lipschitz 定数が大きくなる結果，わずかな摂動に対する特徴空間での違いが大きくなることが一つの原因となる．そこで EvalProtoPNet では，図 5 右内の SDFA Module と記された点線枠内に示すとおり，入力に近い層と出力に近い層で空間的な情報が保たれるように正則化損失を課して学習する．具体的には，次式で表される Alignment Loss L_{align} を正則化損失として課す．

$$
L_{align} = \sum_{i,j} \left[\left| \frac{z_i^{\mathcal{D}} \cdot z_j^{\mathcal{D}}}{\|z_i^{\mathcal{D}}\| \|z_j^{\mathcal{D}}\|} - sg\left(\frac{z_i^{\mathcal{S}} \cdot z_j^{\mathcal{S}}}{\|z_i^{\mathcal{S}}\| \|z_j^{\mathcal{S}}\|} \right) \right| - \gamma \right]_+ \tag{7}
$$

ただし，$z_i^{\mathcal{D}}$，$z_i^{\mathcal{S}}$ はそれぞれ出力に近い層及び入力に近い層から出力される特徴マップ内の i 番目の特徴ベクトルであり，$\gamma(\geq 0)$ は類似度の誤差をどの程度許すかを決定するマージンである．また sg は Stop Gradient を表し，括弧内の変数に対し勾配を計算しない操作を示す．

3.2.2　Prototype Layer の改善

図 6 に示すように，Prototype Layer の構造を改善することでプロトタイプの性質を改善した手法として，Mixture of Gaussian-distributed Prototypes (MGProto) [29]，Deformable ProtoPNet [27]，Support-Trivial (ST) ProtoPNet [28] が挙げられる．ただし図 6 において灰色領

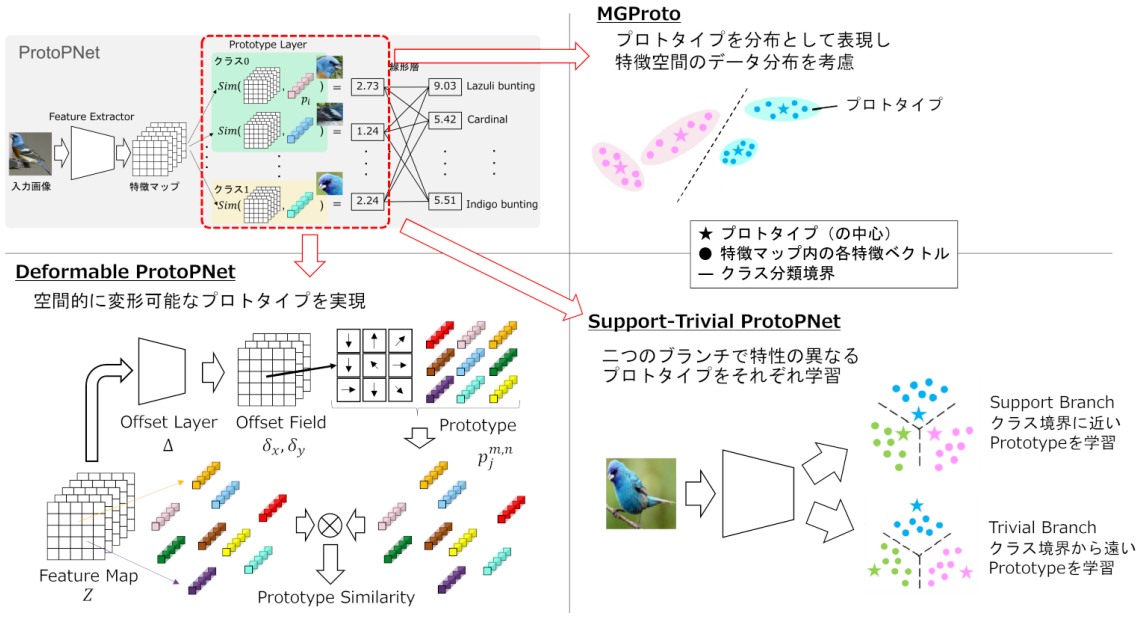

図 6　Prototype Layer の改善によりプロトタイプの性質を改善する手法の図解

域は ProtoPNet を表し，ProtoPNet 上の赤枠は各手法が Prototype Layer を改善した手法であることを図示している．以下では各手法について述べる．

MGProto は図 6 右上のとおりプロトタイプをガウス分布としてモデル化する．ただし図 6 右上において，星マークはガウス分布として表現されるプロトタイプの中心であり，色付きの領域は中心からの距離が分散幅以内の領域を表す．これより MGProto は，ProtoPNet のプロトタイプは特徴空間上の 1 点として表現されるために，特徴空間におけるデータ分布が考慮されていない課題に対応する．MGProto では特徴マップ Z_x と各プロトタイプ p_j との類似度 $S(\mathrm{Z}_x, p_j)$ は次式で定義される．

$$S(\mathrm{Z}_x, p_j) = \max_{z \in \mathrm{Z}_x} \mathcal{N}(z; p_j, \Sigma_j) = \max_{z \in \mathrm{Z}_x} \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma_j|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(z - p_j)^\top \Sigma_j (z - p_j)\right) \tag{8}$$

ただし，D は特徴マップ及びプロトタイプの次元数であり，Σ_j はガウス分布の分散である．MGProto ではガウス分布のパラメータを最適化するステップとその他のモデルパラメータを最適化する二つのステップを交互に行うことで学習が実行される．

Deformable ProtoPNet は，図 6 左下のとおり各プロトタイプに対するオフセットを算出する Offset Layer Δ を利用した Deformable Convolution により，入力画像に応じてプロトタイプを空間的に変形可能とする．これより，固定されたサイズの画像領域を参照するため，画像内の画像特徴の変形に対応できない ProtoPNet の課題を解決する．図 6 左下に示すように，Deformable Prototype p_j は複数のプロトタイプ $p_j^{(m,n)}$ で構成され，各プロトタイプに対する空間的な位置ずれ（オフセット）が Offset Layer により $\delta_k = \Delta_k(Z_i, a, b, m, n)$ として算出される．その後プロトタイプと特徴マップの類似度が，算出されたオフセットと対応する位置の画像パッチ特徴と各プロトタイプの類似度の平均値として算出される．そこでプロトタイプと特徴マップ Z_x との類似度 $S(\mathrm{Z}_x, p_j)$ は

$$\max_{a,b} \sum_{m,n} \frac{1}{r} \frac{p_j^{(m,n)}}{\|p_j^{(m,n)}\|} \cdot \frac{z_i^{(a+m+\delta_x, b+n+\delta_y)}}{\|z_i^{(a+m+\delta_x, b+n+\delta_y)}\|} \tag{9}$$

と表される．ただし r は Deformable Prototype p_j に属するプロトタイプの数である．

ST-ProtoPNet は図 6 右下のとおり二つの異なるブランチをもつ構造を採用することで，クラス境界から離れた Trivial 特徴に加えクラス境界に近い Support 特徴がプロトタイプとして学習されるようモデル構造を改善した手法である．これより，ProtoPNet のプロト

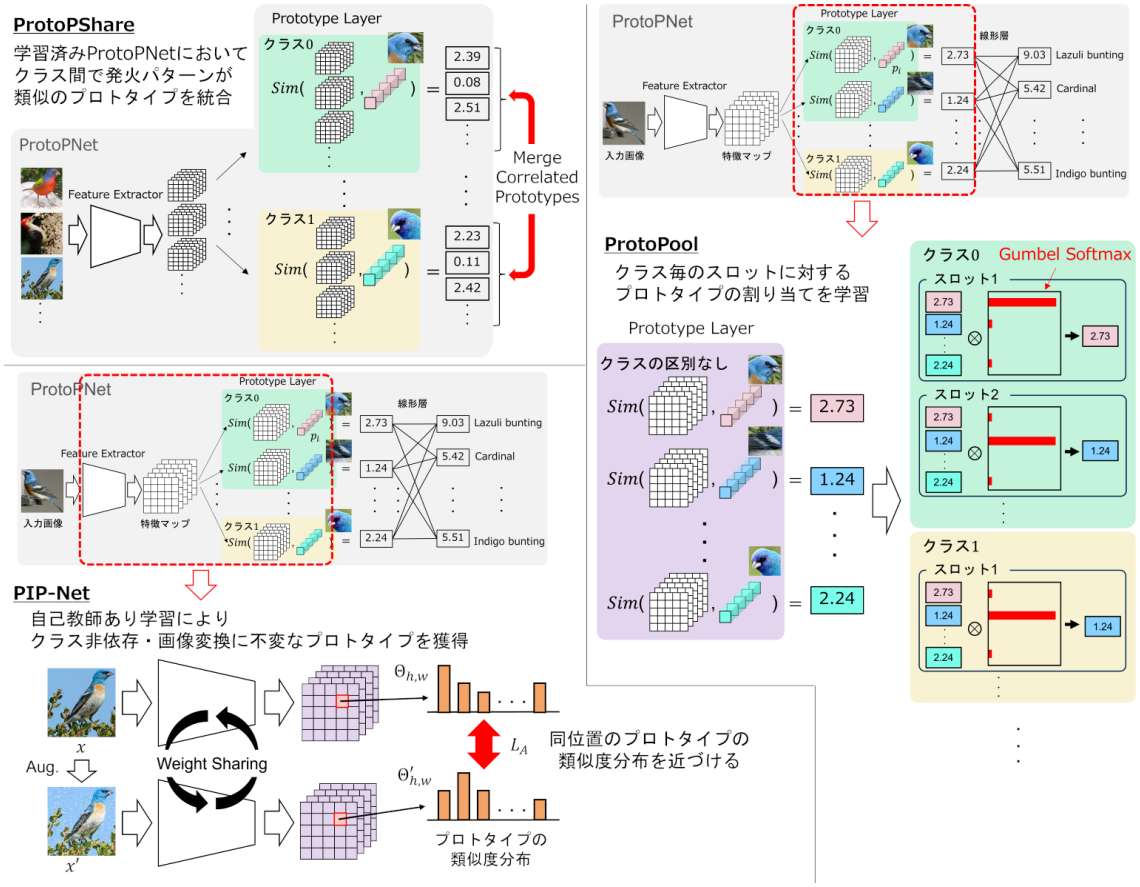

図 7　線形層で使用するプロトタイプを削減する手法の図解

タイプには Trivial 特徴を代表する特性があるため，クラス分類に有効な Support 特徴がプロトタイプとして学習されない課題に対応する．Support 及び Trivial 特徴を学習する目的のため，ST-ProtoPNet では次式で表される損失関数 L_{cls}, L_{dsc} を Support，Trivial ブランチそれぞれに課し学習を行う．

$$
\begin{aligned}
L_{cls} &= -\sum_{\substack{y_1,y_2\in Y \\ y_1<y_2}} \min_{p_j\in \mathrm{P}_{y_1}, p_k\in \mathrm{P}_{y_2}} p_j^{\top}\cdot p_k \\
L_{dsc} &= \sum_{\substack{y_1,y_2\in Y \\ y_1<y_2}} \max_{p_j\in \mathrm{P}_{y_1}, p_k\in \mathrm{P}_{y_2}} p_j^{\top}\cdot p_k
\end{aligned}
\tag{10}
$$

式 (10) より L_{cls} は異なるクラスに属するプロトタイプ間の類似度の最小値を最大化し，L_{dsc} は異なるクラスに属するプロトタイプ間の類似度の最大値を最小化すると言える．そこでこれらの正則化損失を課すことにより，各ブランチにおいてクラス境界に近い（遠い）プロトタイプが獲得される．

3.3　クラス間で共通のプロトタイプ表現の獲得

3.1 で説明したように，ProtoPNet は，クラスごとに個別のプロトタイプを用意するため，クラス間でプロトタイプを共有できずメモリ効率が悪い．このため，クラス間で挙動が類似した冗長なプロトタイプが発生する場合がある．この課題に対しては，クラス間でプロトタイプを共有することで，線形層で使用するプロトタイプを削減する手法が提案されている．またクラス間で共有されたプロトタイプを学習することで，決定木や k 近傍法などの解釈性の高い分類器へ ProtoPNet を拡張する手法が提案されている．本節ではこれらの手法についてそれぞれ説明する．

3.3.1　線形層で使用するプロトタイプの削減手法

図 7 に示すように，学習済み ProtoPNet の解析や Prototype Layer，Feature Extractor の学習方法の改善によりクラス間で共有するプロトタイプを獲得し，線形

層で使用されるプロトタイプの総数を削減する手法として，ProtoPShare [30]，ProtoPool [32]，PIP-Net [22] が挙げられる．

ProtoPShare は，図 7 左上のとおり学習済みの ProtoPNet において発火パターンが類似するプロトタイプを統合することで，プロトタイプの総数を削減する．ProtoPShare はプロトタイプ間の類似度の指標として $d^{DD}(p,\tilde{p})$ を採用する．式 (11) に示すように，$d^{DD}(p,\tilde{p})$ は学習データの全画像 $x \in \mathrm{X}^{Train}$ についてのプロトタイプ p 及び $\tilde{p}$ との類似度の 2 乗誤差を用いる．この計算は，言い換えれば画像内のプロトタイプの発火の高さの相関を計算しているともいえる．

$$d^{DD}(p,\tilde{p}) = \frac{1}{\sum_{x \in \mathrm{X}^{\mathrm{Train}}} (S(\mathrm{Z}_x, p) - S(\mathrm{Z}_x, \tilde{p}))^2} \quad (11)$$

ProtoPShare はクラス間で共通したプロトタイプによるメモリ効率の改善を初めて実現した．しかし，ProtoPShare は事前学習済みの ProtoPNet を再学習する必要があるため，合計の学習時間が増加する欠点がある．

ProtoPool は，図 7 右のとおり赤枠で示す Prototype Layer を改良する．ProtoPool は各プロトタイプのクラスへの対応を事前定義せずに，クラスごとに用意された学習可能な K 個のスロットへのプロトタイプの割り当てを分類タスクの学習と同時に行う．各スロットには，画像と割り当てられたプロトタイプとの類似度が格納され，全結合層に入力される．クラス y に用意された k 番目のスロットについて，プロトタイプの割り当て確率分布を $q_{y,k} \in \mathbb{R}^M$ として，プロトタイプ p_j が割り当てられる確率は式 (12) で定義される．ただし，τ は温度パラメータであり，$\{\eta_m\}_{1,\ldots,M}$ は標準 Gumbel 分布に従う確率変数である．

$$P(p_j|y,k) = \frac{\exp\left(\frac{q_{y,k,j}+\eta_j}{\tau}\right)}{\sum_{m=1}^{M} \exp\left(\frac{q_{y,k,m}+\eta_m}{\tau}\right)} \quad (12)$$

クラス y の k 番目のスロットに格納される値は，式 (13) で計算される類似度の重み付き和となる．

$$g_{y,k}^{select} = \sum_{m=1}^{M} P(p_m|y,k) g_{p_m}^{focal} \quad (13)$$

ただし $g_{p_j}^{focal}$ は，式 (14) で計算される Focal Similarity であり，各パッチ位置における類似度の最大値と平均値の差を用いることで，プロトタイプとの類似度が高くなる領域がより局所的になるように促される．

$$g_{p_j}^{focal} = S(\mathrm{Z}_x, p_j) - \frac{1}{|\mathrm{Z}_x|} \sum_{z \in \mathrm{Z}_x} \theta(z, p_j) \quad (14)$$

また，各クラスへのプロトタイプの割り当てが互いに異なるものとなるように，式 (15) に示す損失を併せて用いる．

$$L_{orth} = \sum_{i<j}^{K} \frac{q_i \cdot q_j}{\|q_i\|_2 \cdot \|q_j\|_2} \quad (15)$$

PIP-Net は，POF 図 7 左下のとおり赤枠で示す Feature Extractor 及び Prototype Layer を改良する．PIP-Net は各プロトタイプのクラスへの対応は事前定義せず，画像パッチ特徴と各プロトタイプの類似度（図 7 左下におけるプロトタイプの類似度分布）が，画像変換に対し不変となるよう自己教師あり学習を行うことでプロトタイプを学習する．これにより，単一の概念をもつパッチとのみ類似度が高くなるクラス間で共通したプロトタイプを獲得できる．PIP-Net で採用される損失関数 L_A を式 (16) に示す．

$$L_A = -\frac{1}{HW} \sum_{(h,w) \in H \times W} \log(\Theta_{h,w} \cdot \Theta'_{h,w}) \quad (16)$$

ただし $\Theta, \Theta' \in \mathbb{R}^{H \times W \times M}$ は，ある入力画像 x 及び x に対して画像変換を付与した画像 x' から得られる各プロトタイプとの類似度マップであり，$\Theta_{h,w,m}$ は位置 (h,w) におけるプロトタイプ p_m との類似度を示す．なお，Θ, Θ' は，式 (17) のように特徴マップ $\mathrm{Z} \in \mathbb{R}^{H \times W \times M}$ についてプロトタイプ方向に softmax 関数を適用した結果であり，$\Theta_{h,w}$ は位置 (h,w) がどのプロトタイプに属するかの確率分布を表す．

$$\Theta_{h,w,m} = \frac{e^{\mathrm{Z}_{x,h,w,m}}}{\sum_{m=1}^{M} e^{\mathrm{Z}_{x,h,w,m}}} \quad (17)$$

3.3.2 線形層以外の分類モデルを用いる手法

ProtoPNet は，プロトタイプと入力画像の類似度を線形分類モデルにより分類することで解釈可能性を実現する．一方で解釈可能なクラス分類手法には線形分類以外にも決定木や k 近傍法などが挙げられ，これら分類器を使用する ProtoPModels が提案されている．図 8 の赤枠に示すように，これら手法では Prototype Layer 及び線形層による分類部分を改良する．本節では，その代表として ProtoTree [31] と ProtoKNN [33] について述べる．

ProtoTree は，図 8 左のとおりプロトタイプとクラ

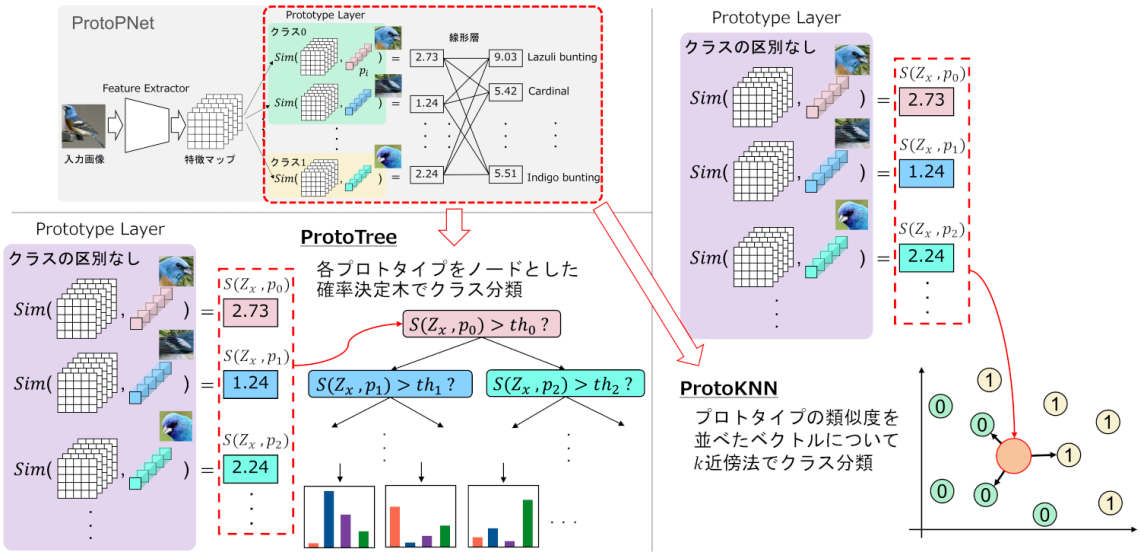

図 8　線形層以外の分類モデルを学習する手法の図解

スの対応を事前定義せずに特徴マップとの類似度を計算し，以降の線形層を二分決定木に置き換える．二分決定木の各ノードには各プロトタイプが割り当てられ，画像と各プロトタイプの類似度を各ノードでの分岐確率として用いる．ただし，プロトタイプ p_m と特徴パッチ $z \in \mathrm{Z}_x$ の類似度は，ユークリッド距離を用いて式 (18) のように計算する．

$$p_e(\mathrm{Z}_x) = \max_{z \in \mathrm{Z}_x} \exp(-\|z - p_m\|_2) \tag{18}$$

このとき，葉ノード l に到達する確率は式 (19) に示すとおり，子ノードへ移動する経路 $\mathcal{P}_l$ を辿る同時確率で表される．

$$\boldsymbol{\pi}_l(\mathrm{Z}_x) = \prod_{e \in \mathcal{P}_l} p_e(\mathrm{Z}_x) \tag{19}$$

なお ProtoTree の学習では，Deep Neural Decision Forests [41] と同様に分類器を学習する．

ProtoKNN は，図 8 右のとおりプロトタイプとクラスの対応を事前定義せずに特徴マップとの類似度を計算し，以降の線形層を k 近傍法によるクラス分類に置き換える．ProtoKNN では，サンプル間の距離は画像とプロトタイプの類似度を並べたベクトル間の距離で定義される．ProtoKNN はプロトタイプの学習のため，下式で定義されるサンプリングでミニバッチ内の各サンプル間でプロトタイプの類似度を比較し，そのサンプルに存在するプロトタイプを推定する．

$$E(x_a, p_j) = \sum_{b \in \mathrm{B}/\{a\}} \Gamma\left(\frac{S(\mathrm{Z}_{x_a}, p_j) - S(\mathrm{Z}_{x_b}, p_j)}{\tau}\right) \tag{20}$$

ただし Γ は Gumbel top-k trick [42] によるサンプリングを示し，τ は温度パラメータである．その後，E をクラス内平均で平準化した $\hat{E}$ を用いて，式 (21) で定義される各プロトタイプのクラスごとの所属度 $P(y, p_j)$ を算出する．

$$P(y, p_j) = \frac{\hat{E}(y, p_j)}{\sum_{i=1}^{M} \hat{E}(y, p_i)} \tag{21}$$

この所属度を基に，式 (22) で定義される Cluster Loss を最小化することで学習を行う．

$$\begin{aligned} L_{clst} &= \sum_{y \in \mathrm{Y}} \sum_{a \in \mathrm{B}, j \in \mathrm{P}} T_{y,a,j} C_{a,j} \\ s.t. &\sum_{j \in \mathrm{B}} T_{y,a,i} = \mathbf{1}(y_a = y), \sum_{a \in \mathrm{B}} T_{y,a,j} = N_y P(y, p_j) \end{aligned} \tag{22}$$

ただし，N_y はミニバッチ内でクラスラベルが y となるサンプル数であり，$C_{a,j}$ は $C_{a,j} = \min_{z \in Z_a} \|z - p_j\|$ で定義される特徴マップとプロトタイプとの距離である．

表 4　自然画像以外のドメインにおける ProtoPModel 応用研究の分類

ドメイン	タスク	手法名	発表年
医用画像 (4.1)	Digital Mammography Classification	IAIA-BL [53]	2021
	X-ray Image Classification	XProtoNet [54]	2021
	Covid-19 Classification	Ps-ProtoPNet [55]	2021
	Covid-19 Classification	NP-ProtoPNet [56]	2021
	Whole-slide Image Classification	ProtoMIL [57]	2022
	Medical Image Classification	M. Nauta [58]	2023
	Aortic Stenosis Classification	ProtoASNet [59]	2023
	Brain Tumor Classification	MProtoPNet [60]	2023
時系列 (4.1.1)	Text / Sequential Data Classification	ProSeNet [44]	2019
	Sequential Data Classification	P2ExNet [43]	2020
	Text Classification	SCNPro [46]	2021
	(Clinical) Text Classification	ProtoPatient [45]	2022
	Seizure prediction	MSProtoPNet [47]	2023
	Text Classification	ProtoryNet [48]	2023
グラフ (4.1.2)	Graph Classification	PxGNN [51]	2022
	Graph Classification	ProtGNN [61]	2022
	Graph Classification	GLGExplainer [49]	2023
	Graph Regression	PRoGreST [52]	2023
その他	Earth system Image Classification	ProtoLNet [62]	2022
	Anomaly Detection	Semi-ProtoPNet [63]	2022
	Autonomous Driving	InAction [64]	2022
	Kidney Stone Image Classification	D. Flores-Araiza [65]	2023
	Hierarchical Video Classification	HIPE [66]	2023

4. ProtoPModels の広がり

3. で説明したように，ProtoPModels は学習データ内のある部分領域を提示することで解釈可能性を実現する．したがって，'this looks like that' フレームワークによる解釈可能性の実現は（学習データ自体が解釈可能である限り成立する）汎用性の高いものと言える．そのため ProtoPModels は画像以外の様々なドメインや認識タスクへの応用が進められている．この点において ProtoPModels は，（自然）画像におけるクラス分類のみを対象とする，他の Ante-hoc な解釈可能性に関する研究 [37], [40] と一線を画している．本章ではこれら ProtoPModels の応用研究を説明する．

4.1 自然画像以外のドメインへの応用

本章では ProtoPModels の自然画像以外のドメインにおける研究を概説する．表 4 に示すように，ProtoPModels は医用画像に加え，時系列データやグラフデータ，また動画像への応用が進められている．これらのドメインは人間の生活に大きな影響を与える分野のため，特にシステムの解釈可能性が重要となる．実際，医用画像及び心拍信号や診断書等の時系列データの分類システムは，人間の生命に大きな影響を与える医療診断に活用される．また，グラフデータを対象とする Graph Neural Network は創薬開発への応用が著しい．以下では特に画像以外のドメインである，時系列データ及びグラフデータにおける ProtoPModels の応用について説明する．

4.1.1 時系列データにおける応用

本節では時系列データにおける ProtoPModels の応用について概説する．時系列データは 1 次元の畳み込み層により処理することが可能である．そのため，画像と同様に畳み込み層へ入力した特徴ベクトルの系列とプロトタイプの類似度を算出し，それらの類似度に基づき推論を行うことで ProtoPModels を構築することができる [43]．また時系列全体を一つのプロトタイプとして捉え，時系列間の類似度に基づいて事例ベースの推論を行うことで解釈可能性を実現した手法も提案されている [44], [45]．しかし，上記の方式では異なる時間スケールの信号特徴や時系列の順序関係を捉えた説明を行うことが難しい．これらの課題に対しては複数の異なる窓サイズをもつ（したがって局所的な順序関係を考慮する）プロトタイプを用いる手法が提案されている [46], [47]．また長期的な順序関係を考慮するため文単位で最も類似するプロトタイプとの類似度を算出し，それらを RNN 等により時系列処理する手法が提案されている [48]．

4.1.2 グラフデータにおける応用

グラフデータではグラフ内のノード単体ではなく各ノードがどのようにつながっているかの関係性が重要となる．そのためグラフデータでは，グラフ全体やグ

表 5　一般クラス分類以外のタスクにおける ProtoPModel 応用研究の分類

大分類	タスク	手法名	発表年
Closed-set 認識タスク (4.2.1)	Hierarchical Classification	HPNet [68]	2019
	Few-shot Learning	RCN [69]	2020
	(Class Incremental) Continual Learning	ICICLE [70]	2023
	Semantic Segmentation	ProtoSeg [71]	2023
	Few-shot Learning	MCPNet [35]	2024
Open-set 認識タスク (4.2.2)	Zero-shot Learning	APN [72]	2020
	Out-of-Distribution Detection	IPRA [73]	2021
	(Compositional) Zero-shot Learning	ProtoProp [74]	2021
	Image Retrieval	ProtoMetric [34]	2023
	Out-of-Distribution Detection	PIP-Net [22]	2023
	Out-of-Distribution Detection	MGProto [29]	2023
認識以外のタスク (4.2.3)	Knowledge Distillation	Proto2Proto [75]	2022
	(Offline) Reinforcement Learning	ProtoX [76]	2022
	(Offline) Reinforcement Learning	PW-Net [77]	2023
	Model Debug	ProtoPDebug [78]	2023
	Model Debug	R3-ProtoPNet [79]	2023

ラフ内の重要なサブグラフ（motif）を用いて説明を構築する必要がある．そこで，特に motif をプロトタイプとして採用する場合には，(1) プロトタイプに対応する motif をどのように探索し，(2) 入力グラフ内のどの motif が重要か特定するかが課題となる．

これらの課題に対し, GLGExplainer [49] は, Post-hoc な説明手法 [50] により分解された部分グラフを用いて学習・推論する戦略を採用した．一方で ProtGNN [61] は探索アルゴリズムの活用により学習済みモデルを前提としない手法を提案した．より具体的には (1) の課題に対し，Monte Carlo Tree Search (MCTS) による探索を採用した．ただし MCTS は計算コストが大きい課題があるため，(2) の課題に対しては MCTS の代わりに Conditional Sub-graph Sampling Module (CSSM) による motif の抽出を提案した．ここで CSSM は全結合層として実装され，各プロトタイプ p_k に対しノード i 及び j を結ぶエッジ e_{ij}^k がサブグラフに含まれるかを予測する．また ProGreST [52] は ProtGNN を踏襲することで (1) 及び (2) の課題に対応する．ただし ProGreST では学習最後のプロトタイプに対応する motif の探索においてのみ MCTS を採用し，その他の場合にはノード特徴を代用することで更なる学習コストの削減を行った．

4.2　クラス分類以外のタスクへの応用

表 5 に示すように，ProtoPModels は画像分類タスクとは異なる認識タスクへの応用が進んでいる．本章では各タスクに関する ProtoPModels の応用について述べる．以下では，まず学習データとテストデータで共通するクラスラベルを扱う Closed-set な問題設定における応用（**4.2.1**）について説明する．その後，学習データに含まれないクラスラベルを扱う Open-set な問題設定における応用（**4.2.2**）について述べ，最後にモデルデバッグや強化学習等（画像）認識以外のタスクにおける応用（**4.2.3**）について説明する．

4.2.1　Closed-set な問題設定における応用

表 5 に示すように Closed-set な問題設定では階層的な画像分類のほかに，Few-shot Learning や Semantic Segmentation，また継続学習への ProtoPModels の応用が進められている．これらのタスクは与えられるサンプルが非常に少ない，分類が画像単位ではなくピクセル単位で行われる，あるいは過去に学習したクラスに対する知識を保持しつつ新しいクラスを分類できるよう学習する等, 非常に特殊な状況における分類タスクを扱っている．以下では各タスクにおける ProtoPModels の応用について詳細に説明する．

a）　Few-shot Learning への応用

Few-shot Learning (FSL) は一クラスあたりのサンプル数 K が非常に少ない状況におけるクラス分類タスクである．FSL ではまず，テストデータに含まれないクラスサンプルにより構成される学習データで事前学習を行う．その後，テストデータより N クラスごとに K 枚与えられるラベル付き画像（Support）を用いて，ラベルのないテスト画像（Query）のクラスラベルを予測する．ただし FSL の研究では N 及び K の値として，Vinyals らの実験設定 [67] である $N = 5, K = 1, 5$ が用いられることが多い．FSL におけるアプローチは大きく二つのアプローチに分類することができる．第一のアプローチは評価時に与えられる Support サンプルに対し，効率的にモデルを適応できるよう事前学習するメタ学習ベースのアプローチである．また第二の

アプローチは事前学習で獲得した特徴表現によりサンプル間の距離を推論し，その距離に基づきクラス分類を行う距離学習ベースのアプローチである．

FSL を対象として ProtoPModels を構築した研究として Region Comparison Network (RCN) [69] 及び MCPNet [35] が挙げられる．RCN は Support サンプルより抽出される特徴マップ内の特徴ベクトルをプロトタイプとして選択し，各プロトタイプにどの程度の重みを与えるべきかをメタ学習するアプローチを採用する．RCN では Support サンプルと Query サンプルが与えられた際，Support サンプル内のどの画像領域に大きな重みを与えるべきかをメタ学習により学習する．一方で MCPNet は距離をもとに推論を行う距離学習ベースのアプローチを採用する．サンプル間の距離を解釈可能とするため，MCPNet では（ProtoKNN と同様に）プロトタイプ分布間の距離によりサンプル間距離を定義する．ただし MCPNet ではプロトタイプ分布間の Jensen-Shannon divergence によりサンプル間の距離が定義される．

b） Semantic Segmentation への応用

Semantic Segmentation は入力画像内の各ピクセルが，各々どのクラスに属するかを推論する分類タスクである．Semantic Segmentation を対象として ProtoPModels を構築した研究として ProtoSeg [71] が挙げられる．ProtoSeg では，まず画像内の各ピクセルに対する特徴を Atrous Spatial Pyramid Pooling (ASPP) 層により算出する．次に各ピクセルに対する特徴とプロトタイプとの類似度を算出し，その類似度に基づき各ピクセルに対するクラスを予測する．また ProtoSeg では，次式で表される Prototype Diversity Loss L_J を課すことにより，各クラスを代表する多様な特徴をプロトタイプとして獲得するように学習する．

$$L_J = \frac{1}{C}\sum_{c\in C} V_J(\{S(\mathrm{Z},p)\}_{p\in \mathrm{P}_c}) \tag{23}$$

ただし，$S(\mathrm{Z},p)$ は特徴マップ Z におけるプロトタイプ p の類似度分布，$V_J(\{U_i\}_{i\in I})$ は分布の集合 $\{U_i\}_{i\in I}$ 間の類似度であり，それぞれ次式で定義される．

$$S(\mathrm{Z},p) = \mathrm{softmax}(\|z_{ij}-p\|^2 | z_{ij}\in \mathrm{Z}, \mathrm{Y}_{ij}=c) \tag{24}$$

$$V_J(\{U\}_{i\in I}) = \sum_{i,j\in I, i<j} \exp(-D_J(U_i,U_j)) \tag{25}$$

ここで $D_J(U_i,U_j)$ は次式で定義される分布 U_i, U_j 間の Jeffrey 距離である．

$$D_J(U_i,U_j) = \frac{1}{2}\left(KL(U_i\|U_j) + KL(U_j\|U_i)\right) \tag{26}$$

c） Continual Learning への応用

Continual Learning（継続学習）は現実世界に現れるタスクの変化に対し，過去のタスクに対する性能を保持しながら新しいタスクに対応する問題設定である．特に Class Incremental Continual Learning (CICL) ではタスクの変化は分類すべきクラスの変化として定義される．すなわち CICL では古いクラスに対する分類性能を保ちつつ新しいクラスを学習することを目的とする．このような問題設定では新しいタスクの学習を行うごとに過去タスクの知識が失われる破滅的忘却が発生することが知られている [80]．特に ProtoPModels では過去タスクの性能だけでなく過去タスクで学習されたプロトタイプの意味が変質することが課題となる [70]．この課題に対し，Interpretable Class Incremental Continual Learning (ICICLE) [70] は次式で示す正則化損失により，プロトタイプの変化を抑制しながら学習することを提案した．

$$L_{IR} = \sum_{z_{ij}^t \in \mathrm{Z}_x^t} |\theta(z_{ij}^t, p^{t-1}) - \theta(z_{ij}^t, p^t)| \cdot M_{ij} \tag{27}$$

ただし $\theta(z,p)$ は特徴ベクトル z とプロトタイプ p 間の類似度であり，p^{t-1}, p^t はタスク $t-1, t$ におけるプロトタイプである．また M_{ij} はプロトタイプと特徴ベクトルの類似度の高い top-γ% の領域が 1，その他の領域が 0 となるバイナリマスクである．

4.2.2 Open-set な問題設定における応用

本節では学習データに含まれないクラスを分類・認識することを目的とする open-set な問題設定への応用について述べる．この問題設定では Zero-shot Learning 及び画像検索タスクにおいて ProtoPModels を実現する手法が提案されている．以下ではこれらタスクにおいて各手法がどのように ProtoPModels を実現しているかについて述べる．

a） Zero-shot Learning への応用

Zero-shot Learning (ZSL) では学習データのクラスラベルに加え，赤い頭など各クラスが保有するクラス属性情報が与えられる．その後テストデータ内のクラス属性情報から，テストデータに対する分類器を構築する．テストデータ内のクラスに属する画像データなしにテストデータに対する分類器を構築することが ZSL の由来となっている．ここで，本章で対象とする ZSL は視覚的属性に基づき未知のクラスを分類する ZSL で

あり，CLIP [81] の画像特徴やテキスト特徴を用いる埋め込みベクトルに基づく ZSL とは異なることに注意されたい．

ZSL ではクラス属性情報が与えられるため，プロトタイプを介して各属性特徴が特徴マップ上に表現されるように学習を行う．APN [72] では無関係な特徴を抑制しつつ各特徴が特徴マップ上のある領域に集中するよう学習するために次式で示す L_{AD} 及び L_{CPT} を課す．

$$\begin{aligned} L_{AD} &= \sum_i \sum_l \|\mathrm{P}_i^{\mathrm{G}_l}\| \\ L_{CPT} &= \sum_{k,h,w} z_{h,w} \cdot p_k \left((h-\hat{h})^2 + (w-\hat{w})^2 \right) \end{aligned} \tag{28}$$

ただし，$\mathrm{P}_i^{\mathrm{G}_l}$ はグループ G_l に属するプロトタイプの第 i 成分を結合することで得られるベクトルであり，$(\hat{h},\hat{w}) = \arg\max_{h,w} z_{h,w} \cdot p_k$ である．これより，APN では単純に属性情報を回帰する手法と比較し，各属性の特徴が特徴マップ上の対応する領域に表現されることが確認されている．

更に ZSL を拡張した手法として Compositional Zero-shot Learning (CZSL) が挙げられる．CZSL では，例えば縞模様をもったトラの頭と白い馬の胴体を含む学習データからシマウマを分類するモデルを構築する等，クラス情報を物体と属性に分離して扱う．そのため CZSL では更にクラス情報の解きほぐしが課題となる．CZSL へ ProtoPModels を応用した研究である ProtoProp [74] では，この課題に対し物体の形状及び属性を代表するプロトタイプをそれぞれ用意し学習する．特に ProtoProp では属性及び形状の特徴がそれぞれ独立となる（解きほぐされる）よう，次式で示す損失関数を課す．

$$L_{HSIC} = \frac{HSIC(z^a, y^o) + HSIC(z^o, y^a)}{2} \tag{29}$$

ただし，y^o 及び y^a は物体の形状及び属性の one-hot ラベル（の集合）であり，$z^k (k \in \{o,a\})$ は

$$z^k = \frac{1}{HW} \sum_{z \in \mathrm{Z}} z \cdot \mathrm{softmax}(z \cdot p_j | p_j \in \mathrm{P}^k) \tag{30}$$

で定義される，各特徴マップを softmax プーリングすることで得られる特徴ベクトル（の集合）である．また，$HSIC$ は Hilbelt-Schmidt Information Criterion (HSIC) であり，与えられた二つの集合が独立であるとき 0，完全に相関する場合 1 をとる．ProtoProp では各物体の形状及び属性に対するプロトタイプを組み合わせることで，最終的な各クラスに対するプロトタイプを算出する．この目的のため ProtoProp は 2 層の Graph Convolutional Network (GCN) を採用する．ここで GCN へ入力されるグラフのエッジは事前知識として与えられるクラス情報により決定される．

b） Image Retrieval への応用

Image Retrieval（画像検索）は，入力画像（クエリ画像）と意味的に類似する画像をギャラリ画像群から検索する画像認識タスクである．画像検索タスクでは，まずテストデータに含まれないクラスサンプルデータを用いて意味的に類似する画像を特徴空間上の近い位置に埋め込む写像を学習する．その後，テストデータ内のある画像と同一クラスの画像をテストデータ内から引き当てられるかを評価指標として評価する．

画像検索タスクへ ProtoPModels を応用した研究として ProtoMetric [34] が挙げられる．ProtoMetric は ProtoKNN の学習戦略によりプロトタイプを学習する．ただし ProtoMetric では画像検索のためのサンプル間の距離指標として，プロトタイプ分布を線形変換することで得られる特徴ベクトル間のコサイン類似度を採用する．すなわち，ProtoMetric におけるサンプル x_i, x_j 間の類似度 $\rho(x_i, x_j)$ は次式のように表される．

$$\rho(x_i, x_j) = \frac{p_i^\top \mathrm{W}_c^\top \mathrm{W}_c p_j}{\|f_i\| \|f_j\|} \propto \sum_{a,b} p_{i,a} K_{a,b} p_{j,b} \tag{31}$$

そこで ProtoMetric により出力されるサンプル間の類似度は各サンプルの各プロトタイプとの類似度の線形和で表されることとなり，解釈可能となる．また，ProtoMetric は画像検索タスクでは過学習のため，学習時間が長くなるほど性能が低下する課題に対し Black-box モデルを蒸留する学習戦略 [82] を採用する．これより，ProtoMetric は Black-box モデルと同等の精度を保ちながら，プロトタイプの学習のため十分長くモデルを学習することを可能とした．

4.2.3 （画像）認識タスク以外における応用

深層学習は認識タスクのほかにも強化学習など重要な応用分野をもつ．ProtoPModels はこれらの応用分野に加え，大規模モデルで学習した知識を小規模モデルに転移する知識蒸留やモデル推論の間違いを修正するモデルデバッグに関する手法が提案されている．以下ではこれらの（画像）認識タスク以外のタスクにおける ProtoPModels の応用について説明する．

a） Reinforcement Learning への応用

Reinforcement Learning（強化学習）は与えられた環境に対し，報酬を最大化するような最善の行動を決定するエージェントを学習するタスクである．強化学習はエージェントが実際に環境とやり取りしながら学習を行うオンライン強化学習と，各環境シーンに対する行動記録をもとに学習を行うオフライン強化学習に大別される．ProtoPModels の強化学習への応用研究では，エキスパートや学習済みのエージェントの行動履歴を学習データとして用いるオフラインのアプローチが採用される．これは過去事例をもとに学習・推論を行う ProtoPModels の特性のためである．強化学習へ ProtoPModels を応用した研究として ProtoX [76] と PW-Net [77] が挙げられる．以下ではこれらの研究について説明する．

ProtoX はフレームの系列をプロトタイプとして採用しエキスパートの行動履歴を学習データとして解釈可能なエージェントを学習する．そのため ProtoX はまず，時間的に近く同一の行動が採用されるデータが特徴空間上で時系列的に近いが異なる行動が採用されるデータ及び時間的に離れたデータとの距離が遠くなるようエンコーダ f を学習する．その後 Cross-Entropy 損失，Cluster Loss 及び Separation Loss に加え，Representability Loss L_{rep} 及び Isometry Loss L_{iso} により線形層 W を学習する．ここで L_{rep}，L_{iso} は次式により定義される損失関数である．

$$
\begin{aligned}
L_{rep} &= \sum_{p \in \mathrm{P}} \min_{i} \|\mathrm{W} f(x_i) - p\|_2^2 \\
L_{iso} &= \|\mathrm{W}^\top \mathrm{W} - \mathrm{I}\|
\end{aligned}
\tag{32}
$$

ただし，P は全てのプロトタイプの集合であり，I は単位行列である．式 (32) により，エンコーダ f の特徴表現を保ちつつ，プロトタイプが学習データ内の部分系列に近づくように学習される．

一方 PW-Net は学習済みのエージェントを活用し解釈可能なエージェントを構築する．具体的には学習済みエージェントのエンコーダ f により変換した特徴ベクトルとプロトタイプの類似度に基づき行動を予測する．ここで PW-Net のプロトタイプはデータセット内で行動 a に対する理由として解釈可能な状態をあらかじめ指定することにより定義される．PW-Net の学習では学習済みのエージェントを用いるため Feature Extractor，線形層は学習の必要なく固定され，Addon Layer $\mathcal{A}$ のみが学習される．

b） Knowledge Distillation への応用

Knowledge Distillation（知識蒸留）は教師モデルの出力を教示として生徒モデルを学習することで，教師モデルの知識を生徒モデルへ転移させる手法である．ProtoPModels においても知識蒸留手法を用いることでモデル軽量化や推論の高速化を実現できることが期待される．一方，一般の知識蒸留手法はモデル出力内の暗黙的知識の転移を目的としているため，これら手法ではプロトタイプにより表現される ProtoPModels の明示的な知識が転移されない場合がある．この課題に対し Proto2Proto [75] は ProtoPModels の明示的な知識を転移するため，次式で示される二つの損失関数 L_{ppc}, L_{global} により知識蒸留を行うことを提案した．

$$
\begin{aligned}
L_{ppc} &= \frac{1}{|\mathrm{X}|} \sum_{x \in X} \sum_{m \in \mathrm{M}_x} \|z_{x,m}^{Teacher} - z_{x,m}^{Student}\|_2 \\
L_{global} &= \frac{1}{|\mathrm{P}|} \sum_{i} D\left(p_i^{Teacher}, p_i^{Student}\right)
\end{aligned}
\tag{33}
$$

ただし，X, P は入力データ及びプロトタイプの集合であり，$|\cdot|$ は集合の要素数を表す．また $D(\cdot,\cdot)$ は二つのプロトタイプ間の Euclid 距離であり，M_x はいずれかのプロトタイプと最も Euclid 距離が近く，かつその距離がしきい値 τ 以下となる特徴マップのピクセルインデックスの集合である．式 (33) に表される損失により，プロトタイプ間の特徴及びそれらのプロトタイプと類似する特徴マップ上の特徴ベクトルが一致するように知識蒸留されるとわかる．

c） Model Debug への応用

ProtoPModels はプロトタイプを介してモデルの推論課程を解釈し，モデル推論過程における間違いを特定することができる．一方どのようにすれば人間の事前知識あるいはフィードバックをもとに，モデル推論過程の間違いを正す（Model Debug/モデルデバッグ）ことができるかは自明ではない．この課題に対しプロトタイプの学習に人間のフィードバックを組み込むことでモデルデバッグを行う手法が提案されている．

このような手法のうち，最初の手法として IAIA-BL [53] が挙げられる．IAIA-BL は意図しない画像領域においてプロトタイプが高い類似度をもたないよう正則化損失を与えることで，推論に意味ある特徴のみをプロトタイプが学習するようにした．しかし IAIA-BL は各画像に対し推論に有効な画像領域を指定

する必要があり，モデル推論修正のための作業が膨大となる．この課題に対し，ProtoPDebug [78] はプロトタイプを重要，不適切，その他の三つに分類し，重要なプロトタイプの忘却を防ぎつつ不適切なプロトタイプを排除するよう再学習する手法を提案した．これより ProtoPDebug では，画像ごとにフィードバックを与えることなくモデルデバッグを行うことが可能となった．更に R3-ProtoPNet [79] はユーザのフィードバックをもとに報酬関数を学習し，学習した報酬関数を最大化するようプロトタイプを再学習する．これより R3-ProtoPNet では，少ない修正でより効率的にモデルデバッグを行うことが可能となった．

4.3 Transformer モデルとのつながり

Vision Transformer (ViT) [20] は自然言語分野で発展した Transformer [83] を画像処理分野へ適用した手法であり，CNN に代わる次世代アーキテクチャとして注目されている．また大規模データセットで ViT を事前学習した DINO [84] や CLIP [81], [85] などの基盤モデルが様々な下流タスクに有効であることが知られており，基盤モデルを活用した研究が多く提案されている．そこで ViT や基盤モデルを活用することでより性能の高い ProtoPModels を実現することができると期待される．

しかし，図 9 に示すように Feature Extractor に単純に ViT に置き換えて学習するのみでは，プロトタイプと高い類似度をもつ画像領域が画像全体に分散する．これは，CNN が局所な受容野により画像構造を保つ帰納的バイアスをもつ一方で，ViT の Self-Attention はローパスフィルタとして作用し [86]，各トークンの特徴を均質化するためである．この課題に対し，ProtoPFormer [23] は次式に示す正則化損失を与えることで，プロトタイプ間で互いに異なる局所領域の類似度が高

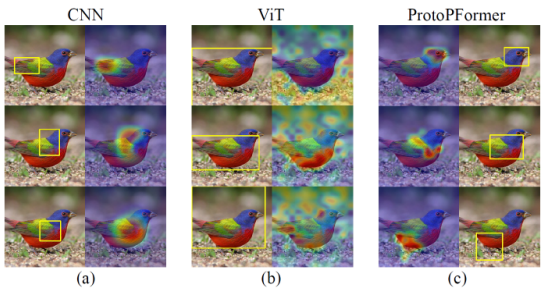

図 9　Feature Extractor が (a)CNN ベースの場合と (b)ViT ベースの場合，及び (c)ProtoPFormer [23] におけるプロトタイプの可視化例．図は [23] より抜粋

くなるように学習する．

$$
\begin{aligned}
L^{\mu}_{PPC} &= \frac{1}{|\mathrm{P}|}\sum_{i\neq j}\max(t_{\mu}-\|\hat{\mu}_i-\hat{\mu}_j\|^2,0)\\
L^{\Sigma}_{PPC} &= \mathrm{tr}\left(\max(0,\hat{\Sigma}-t_{\sigma})\right)
\end{aligned}
\tag{34}
$$

ただし，類似度マップを 2 次元ガウス分布として扱い，推定される平均位置を $\hat{\mu}$，分散を $\hat{\Sigma}$，t_{μ}, t_{Σ} を事前に設定するしきい値とする．

また INTR [87] は，DETR [88] に代表される Encoder-Decoder 構造を活用し，デコーダへの入力をクラス固有のクエリとして学習する．デコーダは Multi-head Cross Attention ブロックで構成されており，各ヘッドで捉えた局所的な画像特徴を次式に示すように合算することでクラススコア y_c を算出する．なお，クラススコアに寄与する特徴の計算は，入力画像にのみ依存する特徴とクラス固有のクエリ $q^{(c)}$ に依存する特徴の積に分解することができる．

$$
\begin{aligned}
\hat{y}_c &= \mathrm{W}^{\top}\mathrm{H}^{(c)}\\
&= \mathrm{W}^{\top}\mathrm{W}^{\mathrm{O}}\left[h_1^{(c)\top},\ldots,h_r^{(c)\top},\ldots,h_R^{(c)\top}\right]^{\top}\\
&\propto \sum_{z\in\mathrm{Z}}(\mathrm{W}^{\top}\mathrm{W}^{\mathrm{V}}z)\times\exp(q^{(c)\top}\mathrm{W}^{\mathrm{K}}z)
\end{aligned}
\tag{35}
$$

ただし W は全結合層の重み，$h_r^{(c)\top}$ はクラス c に対応する各ヘッドの出力，$\mathrm{W}^{\mathrm{O}}, \mathrm{W}^{\mathrm{V}}, \mathrm{W}^{\mathrm{K}}$ は射影行列，Z はエンコーダの出力である．特に，第 2 項はパッチ z の Attention を表すため，クラススコアの計算が解釈可能となっている．

一方で ProtoS-ViT [89] は，DINOv2 などの学習済み基盤モデルを固定された特徴抽出器として用いる．これより，大規模データで十分に学習して獲得された良い部位特徴表現をプロトタイプとして取り出すように学習することができる．

5. プロトタイプの解釈性

ProtoPModels は推論結果と判断根拠としての各プロトタイプの類似度の間に明確な因果関係が成立している．そのため，ProtoPModels では忠実性の高い推論根拠の説明が可能となる．一方，プロトタイプと類似度が高いパッチの可視化は，推論とは無関係な post-hoc な手続きとなるため忠実性が保証されない．そのため，説明として提示されるプロトタイプの解釈可能性は議論の余地がある．これに対し，Nauta ら [90], [91]

表 6　Nauta ら [90] が提唱する説明可能 AI を評価する 12 の観点

大分類	観点	概要
内容	Correctness (5.1)	説明がモデルの予測に対してどの程度忠実であるか
	Completeness	モデルの挙動をどの範囲まで説明できているか
	Consistency	同じ入力や同条件の学習で説明がどの程度一貫するか
	Continuity (5.2)	類似の入力に対してどの程度説明が類似するか
	Contrastivity	異なる予測を出力する入力における説明がどの程度異なるか
	Covariate complexity (5.1)	説明の中で提示される特徴の理解のしやすさ
形式	Compactness	説明がどの程度簡潔か
	Composition	説明の提示方法，まとまり方
	Confidence	説明を信頼度や不確実性についてのスコアを伴って提示できているか
ユーザ	Context (5.3)	実際のユースケースやエンドユーザに有効な説明をどの程度提供しているか
	Coherence	人間の知見や信念と比較してどの程度筋の通った説明を提供できているか
	Controllability	説明の提供がどの程度対話的か，またはどの程度説明を制御できるか

表 7　プロトタイプの解釈可能性に焦点を当てた手法と解釈可能性評価観点の対応

大分類	手法・論文名	Correctness	Continuity	Covariate complexity	Context
プロトタイプの可視化方法改善 (**5.1**)	PRP [92]	✓			
	Sanity Checks for Patch Visualization [93]	✓			
可視化されたプロトタイプの意味理解促進 (**5.1**)	This Looks Like That, Because... [94]			✓	
	ProtoConcepts [95]			✓	
入力空間と特徴空間の類似度のギャップ (**5.2**)	This Looks Like That, Does it ? [96]		✓		
	EvalProtoPNet [21]		✓		
	PIP-Net [22]		✓		
Human Study(**5.3**)	HIVE [98]				✓
	‘Guess Who ?’ Game [100]				✓

は，表 6 に示す 12 の観点を以て，ProtoPModels におけるプロトタイプの解釈可能性についてそれぞれ議論した．本章では，このうち特に Continuity，Context，Correctness，Covariate complexity の 4 観点に焦点を当てて，プロトタイプの解釈可能性とその評価方法及び改善手法を議論した研究を説明する．表 7 に本章で取り扱う手法と観点の対応を示す．

5.1　プロトタイプの可視化と意味理解の改善

本節では，プロトタイプの可視化手法の忠実性 (Correctnes) 及び可視化されたプロトタイプの意味の理解のしやすさ (Covariate complexity) を改善した研究を説明する．

ProtoPModels におけるパッチ特徴の可視化では，一般的に特徴マップの解像度から入力画像の解像度への Upsampling が用いられる．ただし，特徴マップは入力画像に比べて解像度が荒いため，活性度が高い領域をヒートマップとして可視化してもどのような特徴を捉えたのか理解しづらい場合がある．この課題に対し，Gautam ら [92] は，LRP [39] の考え方を ProtoPNet に導入した Prototypical Relevance Propagation (PRP) により，プロトタイプと類似度が高い領域の可視化の高解像度化を実現した．また，Xu-Darme ら [93] は，通常の Upsampling, SmoothGrad [99], PRP [92] の三つの可視化方法の比較分析を行った．これにより，Upsampling による可視化では忠実性が低い一方，Smooth Grad や PRP を用いた場合は忠実性が高い可視化が実現できることが定量的に示された．

また，プロトタイプが実際のどの画像領域を表すか正しく可視化された場合でも単一の画像パッチの提示だけではどのような特徴を表現しているのか一意な解釈が難しい場合がある．この課題に対し，Nauta ら [94] は，画像パッチと低次特徴の寄与度をセットで提示することでプロトタイプの意味を解釈する手法を提案した．この手法では，入力画像に 5 種類の画像変換のいずれかを付与し，変換付与前後での各プロトタイプの類似度の変化量を用いた重み付き和 $\phi_{\mathcal{T},j}$ により各変換の寄与度を求める．ただし

$$\phi_{\mathcal{T},j} = \frac{\sum_{x \in X^{Train}} s^{diff}_{\mathcal{T},x,j} \cdot S(\mathrm{Z}_x, p_j)}{\sum_{x \in X^{Train}} S(\mathrm{Z}_x, p_j)}, \quad \text{where } s^{diff}_{\mathcal{T},x,j} = S(\mathrm{Z}_x, p_j) - S(\mathrm{Z}_{x^{\mathcal{T}}}, p_j) \tag{36}$$

である．ここで $S(\mathrm{Z}_{x^{\mathcal{T}}}, p_j)$ は，学習データセット X^{Train} 中の画像 x に対して変換 $\mathcal{T} \in$ {Contrast, Saturation, Hue, Shape, Texture} を付与して得られた画像 $x^{\mathcal{T}}$ における特徴マップとプロトタイプ p_j の類似度である．また ProtoConcepts [95] では，複数パッチを提示することで各パッチに共通した意味として理解しやすいプロトタイプの提示方法を提案した．このために，プロトタイプをベクトルではなく中心 $\boldsymbol{c}_j$，半径 r_j の超球で表現し，超球面に含まれる複数の画像パッチをプロトタイプの代表として扱った．

5.2 プロトタイプの安定性・一貫性評価

本節では，プロトタイプと画像特徴の類似度の連続性 (Continuity) について，安定性と一貫性の視点から提案された評価手法を説明する．安定性は摂動に対する頑健性を示し，一貫性はプロトタイプと類似度が高くなる部位が画像間でどの程度意味的に一致するかを示す．

安定性について，Hoffman ら [96] は，入力画像に人間に知覚困難な摂動を与えることでプロトタイプと類似度が高い領域が変化することを実験により示し，ProtoPModels の学習では特徴空間での類似性と画像空間での類似性に意味的なギャップが生じる課題があると主張した．この課題を定量評価するベンチマークとして，Huang ら [21] は Stability score を提案している．Stability score は入力画像とプロトタイプの類似度が画像摂動に対しどの程度頑健かを測定する指標であり，次式によって定義される．

$$\Psi^{sta} = \frac{1}{M} \sum_{j=1}^{M} \frac{\sum_{x \in \mathcal{I}_{c(j)}} \mathbb{1}\{o_{p_j}(x) = o_{p_j}(x+\xi)\}}{\|\mathcal{I}_{c(j)}\|} \tag{37}$$

ただし $o_{p_j}(x)$ は入力画像 x についてプロトタイプ p_j との高類似箇所が部位 l であった場合に $o_{p_j}(x)_l = 1$，それ以外は $o_{p_j}(x)_l = 0$ となる指示ベクトルである．

一方，一貫性について Huang [21] 及び Nauta ら [22] は，一つのプロトタイプの類似パッチを集めた際にそれらの意味的な画像特徴が一貫しない課題に言及した．特に Huang ら [21] は画像間でプロトタイプと類似度が高くなる部位がどの程度一致するかの評価指標として，Consistency score を提案している．ただし，Consistency score Ψ^{con} は

$$\Psi^{con} = \frac{1}{M} \sum_{j=1}^{M} \mathbb{1}\{\max a_{p_j} \geq \mu\}, \quad \text{where } a_{p_j} = \frac{\sum_{x \in \mathcal{I}_{c(j)}} o_{p_j}(x)}{\|\mathcal{I}_{c(j)}\|} \tag{38}$$

によって定義される．また，Nauta ら [22] は Consistency score と類似の指標として Purity を提案した．

Stability score，Consistency score，Purity の算出には，画像ごとに部位の正解位置を表すキーポイントラベルが用いられる．しかしこれらキーポイントのアノテーションは高コストであることに加え，キーポイントラベルのみでは各パーツの正確な形状や領域を考慮できない課題がある．この課題に対しては CG データセットを用いた評価が提案されている [97].

5.3 人のための評価

生成された説明がエンドユーザである人間に有益な情報を提示しているか (Context) は実応用を目指す上で重要である．前節では，機械的な指標によりプロトタイプの解釈性を評価する手法について説明した．機械的な指標は客観的にかつ高速に評価をできる利点がある一方で，プロトタイプによる説明が実際のユースケースで有用であるかまで十分に考慮できているとは言えない．そのため，実際の利用形態に近い形で人間のユーザからフィードバックを得て評価を行う Human Study が求められる．

一般的な解釈可能性手法のための Human Study では，説明が予測の正否の判別にどのように役立つかを人間のフィードバックにより調査する．例えば，どちらの説明が合理的・信用できるかを質問したり，説明を見せた場合と見せなかった場合それぞれで，モデルの出力を予測またはその出力が正しいか否かを判断させて比較する等である．ただし，このような Human Study では，説明を与えられるとモデルの予測を（仮に誤っていても）正しいと信じやすくなる確証バイアスや，被験者がもつ事前知識に起因する回答へのバイアスの影響が懸念される．これに対し Human Interpretability of Visual Explanations (HIVE) [98] では，複数の説明と予測の提示により確証バイアスの低減を図るなど，評価に悪影響を及ぼす人間の認知の傾向を考慮した調査方法が採用されている．また Sinhamahapatra ら [100] は，同じクラスに所属するプロトタイプを見た被験

者がそのクラスを推測できるかを調査することで，ProtoPModels が人間にとって有効な説明を提供できるか評価した．ただし，対象タスクに関する専門知識によるバイアスを回避するために，ImageNet [101] のような一般自然画像で構成されるデータセットを利用した．

6. むすび

本論文では，解釈可能性に関する研究の中で幅広い注目を集めている ‘this looks like that’ フレームワークについて体系的に説明した．

まず **3.** では，‘this looks like that’ フレームワークの代表手法である ProtoPNet のモデル構造と学習方法について説明し，ProtoPNet を改善した後続の研究を二つのグループに分けて説明した．一つ目のグループはクラスごとに共通のプロトタイプを用いる手法であり，正則化の付与や Prototype Layer の改善によって良質なプロトタイプを獲得することが可能となった．また二つ目のグループは分類器や学習戦略の工夫によりクラス間で共通したプロトタイプを獲得する手法であり，これら手法によりメモリ効率の改善が達成された．

次に **4.** では，‘this looks like that’ フレームワークがその汎用性の高さから多様なドメインや詳細画像分類以外のタスクへ応用されていることを述べた．特に入力データドメインの拡張では創薬や医療診断等ハイリスクな応用分野への研究が積極的になされていることをみた．また，詳細画像分類とは異なる問題設定への応用については Closed-set な問題設定，Open-set な問題設定，また認識タスク以外の問題への応用の三つの観点から説明した．これらの研究により，従来の他の解釈可能性に関する研究では十分研究されてこなかった問題設定においても ‘this looks like that’ フレームワークによる解釈可能性が実現できることが示された．一方で，これらの応用タスクの範囲は深層学習手法の応用タスク全体と比べると限定的となっている．例えば姿勢推定や物体検出等の回帰を含む密なタスクや Novel Category Discovery などのラベルなしサンプルを含む Open-set タスクへの応用研究はいまだなされていない．しかし，ProtoPModels は，入力データ（の部分領域）を解釈できれば成立する汎用的なフレームワークであるため，将来的にはこれらタスクへの応用も可能となると考えられる．また，モデルデバッグの観点ではプロトタイプを活用することで効率的にモデルの間違いを修正する手法が実現された．しかしこれら手法においても，一度学習したモデルのプロトタイプが画像のどの領域に類似しているかを，少なくとも 500 枚程度目視確認する作業が必要となり [79]，十分効率化されているとは言えない．クラスの属性情報など知識ベースの観点からモデルデバッグを行う等，更なる効率化が期待される．また再学習によらないモデルの編集によるデバッグ [102], [103] も将来的に有益と考えられる．

最後に **5.** では，プロトタイプの解釈可能性の評価及び改善についての研究を説明した．プロトタイプの解釈可能性の評価については，一貫性や安定性といった自動的に計測できる定量的な機械による評価手法と，人間の被験者からのフィードバックを利用する Human Study による評価手法が提案されている．ただし，解釈可能性の評価には今後も議論を続けるべき点が残っている．まず ProtoPModels に限らず，AI 利用においては，手法やタスク，適用先，エンドユーザ等の想定される利用条件により，提供されるべき説明の観点や形式，量は変動する点に注意が必要である．したがって，ある同一の指標を多様なユースケースで機械的に流用して解釈可能性を比較評価することは必ずしも適切とはいえない．ユースケース別に適した観点から解釈可能性を評価する手法を選択し，実用的な運用を考える必要がある．どのような観点や手法を利用すべきかについては，解釈可能性に関する将来の研究の発展に伴い徐々に洗練されていくと期待される．また可視化されるプロトタイプの高解像化や意味理解を促進する補助情報により，プロトタイプが何を表すか解釈しやすいよう提示する手法が提案されている．しかし，モデルがどのような特徴からプロトタイプを認識しているかについては未だブラックボックスのままであり，プロトタイプの解釈可能性が十分に高いとはいえない．実際プロトタイプの可視化では視覚的な情報しか提供されない，あるいは事前に定義された特性に関する情報のみしか与えられないため，人間にとって意味を一意に定めることが難しい場合がある．この課題については視覚言語モデルを活用し，プロトタイプを自然言語により表現する手法の登場が期待される．また，自然言語の利用とは異なるアプローチとして，低次特徴の組み合わせとしてプロトタイプの意味を解釈しやすいように表現する方向性も考えられる．本論文が ‘this looks like that’ フレームワークの活用や，新たな ProtoPModels の研究活動に踏み出す上での参考になれば幸いである．

文　献

[1] European Commission, “Proposal for a regulation of the european parliament and of the council laying down harmonised rules on artificial intelligence (artificial intelligence act),” European Union Law, COM/2021/206 final, 2021.

[2] S. Nagulendra and J. Vassileva, “Providing awareness, explanation and control of personalized filtering in a social networking site,” Information Systems Frontiers, vol.18, pp.148–158, 2016. DOI: 10.1007/s10796-015-9577-y.

[3] M.T. Ribeiro, S. Singh, and C. Guestrin, “Why should I trust you?: Explaining the predictions of any classifier,” Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.1135–1144, 2016.

[4] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning deep features for discriminative localization,” Computer Vision and Pattern Recognition, pp.2921–2929, 2016.

[5] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-cam: Visual explanations from deep networks via gradient-based localization,” Int. Conf. Computer Vision, pp.618–626, 2017.

[6] S.M. Lundberg and S. Lee, “A unified approach to interpreting model predictions,” Advances in Neural Information Processing Systems, pp.4765–4774, 2017.

[7] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, “Network dissection: Quantifying interpretability of deep visual representations,” Computer Vision and Pattern Recognition, pp.6541–6549, 2017.

[8] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, and R. Sayres, “Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav),” Int. Conf. Machine Learning, pp.2673–2682, 2018.

[9] A. Ghorbani, J. Wexler, J. Zou, and B. Kim, “Towards automatic concept-based explanations,” Advances in Neural Information Processing Systems, pp.9273–9282, 2019.

[10] C. Rudin, “Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead,” Nature Machine Intelligence, vol.1, no.5, pp.206–215, 2019.

[11] C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, and J.K. Su, “This looks like that: Deep learning for interpretable image recognition,” Advances in Neural Information Processing Systems, pp.8928–8939, 2019.

[12] C. Molnar. “Interpretable machine learning: A guide for making black box models explainable,” Lulu.com, 2020.

[13] Y. Zhang, P. Tiňo, A. Leonardis, and K. Tang, “A survey on neural network interpretability,” IEEE Trans. Emerging Topics in Compututational Intelligence, vol.5, no.5, pp.726–742, 2021, DOI: 10.1109/TETCI.2021.3100641.

[14] H. Yuan, H. Yu, S. Gui, and S. Ji, “Explainability in graph neural networks: A taxonomic survey,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol.45, no.5, pp.5782–5799, 2023, DOI: 10.1109/TPAMI.2022.3204236.

[15] A. Poché, L. Hervier, and MC. Bakkay, “Natural example-based explainability: A survey,” Explainable Artificial Intelligence, pp.24–47, 2023.

[16] G. Schwalbe and B. Finzel, “A comprehensive taxonomy for explainable artificial intelligence: A systematic survey of surveys on methods and concepts,” Data Mining and Knowledge Discovery, pp.1–59, 2023.

[17] G. Alicioglu and B. Sun, “A survey of explainable AI in deep visual modeling: Methods and metrics,” arXiv preprint arXiv:2301.13445, 2023.

[18] M. Nauta, J. Trienes, S. Pathak, E. Nguyen, M. Peters, Y. Schmitt, J. Schlötterer, M.V. Keulen, and C. Seifert, “From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable ai,” Comput. Surv., vol.55, pp.1–42, 2023.

[19] T. Räuker, A. Ho, S. Casper, and D. Hadfield-Menell, “Toward transparent AI: A survey on interpreting the inner structures of deep neural networks,” Secure and Trustworthy Machine Learning, pp.464–483, 2023. DOI: 10.1109/SaTML54575.2023.00039

[20] 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,” Int. Conf. Learning Representations, 2021.

[21] Q. Huang, M. Xue, W. Huang, H. Zhang, J. Song, Y. Jing, and M. Song, “Evaluation and improvement of interpretability for self-explainable part-prototype networks,” Int. Conf. Computer Vision, pp.2011–2020, 2023.

[22] M. Nauta, J. Schlötterer, M. Keulen, and C. Seifert, “Pip-net: Patch-based intuitive prototypes for interpretable image classification,” Computer Vision and Pattern Recognition, pp.2744–2753, 2023.

[23] M. Xue, Q. Huang, H. Zhang, L. Cheng, J. Song, M. Wu, and M. Song, “Protopformer: Concentrating on prototypical parts in vision transformers for interpretable image recognition,” arXiv preprint arXiv:2208.10431, 2022.

[24] K.K. Nakka and M. Salzmann, “Towards robust fine-grained recognition by maximal separation of discriminative features,” Asian Conference on Computer Vision, 2020.

[25] S. Kraft, K. Broelemann, A. Theissler, and G. Kasneci. “Sparrow: Semantically coherent prototypes for image classification,” British Machine Vision Conference, pp.1–12, 2021.

[26] J. Wang, H. Liu, X. Wang, and L. Jing, “Interpretable image recognition by constructing transparent embedding space,” Int. Conf. Computer Vision, pp.895–904, 2021.

[27] J. Donnelly, A.J. Barnett, and C. Chen, “Deformable protopnet: An interpretable image classifier using deformable prototypes,” Computer Vision and Pattern Recognition, pp.10265–10275, 2022.

[28] C. Wang, Y. Liu, Y. Chen, F. Liu, Y. Tian, D.J. McCarthy, H. Frazer, and G. Carneiro, “Learning support and trivial prototypes for interpretable image classification,” Int. Conf. Computer Vision, pp.2062–2072, 2023.

[29] C. Wang, Y. Chen, F. Liu, D.J. McCarthy, H. Frazer, and G. Carneiro, “Mixture of gaussian-distributed prototypes with generative modelling for interpretable image classification,” arXiv preprint arXiv:2312.00092, 2023.

[30] D. Rymarczyk, Ł. Struski, J. Tabor, and B. Zieliński, “Protopshare: Prototypical parts sharing for similarity discovery in interpretable image classification,” Proc. 27nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, pp.1420–1430, 2021.

[31] M. Nauta, R. van Bree, and C. Seifert, “Neural prototype trees for interpretable fine-grained image recognition,” Computer Vision and Pattern Recognition, pp.14933–14943, 2021.

[32] D. Rymarczyk, Ł. Struski, M. Górszczak, K. Lewandowska, J. Tabor, and B. Zieliński, “Protopool: Interpretable image classification with differentiable prototypes assignment,” European Conference on Computer Vision, pp.351–368, 2022.

[33] Y. Ukai, T. Hirakawa, T. Yamashita, and H. Fujiyoshi, “This looks like it rather than that: Protoknn for similarity-based classifier,” Int. Conf. Learning Representations, 2023.

[34] Y. Ukai, T. Hirakawa, T. Yamashita, and H. Fujiyoshi, “Toward prototypical part interpretable similarity learning with protometric,” IEEE Access, vol.11, pp.62986–62997, 2023. DOI: 10.1109/ACCESS.2023.3287638.

[35] B.S. Wang, C. Wang, and W. Chiu, “Mcpnet: An interpretable classifier via multi-level concept prototypes,” Computer Vision and Pattern Recognition, pp.10885–10894, 2024.

[36] H. Fukui, T. Hirakawa, T. Yamashita, and H. Fujiyoshi. “Attention branch network: Learning of attention mechanism for visual explanation,” Computer Vision and Pattern Recognition, pp.10705–10714, 2019.

[37] P.W. Koh, T. Nguyen, Y.S. Tang, S. Mussmann, E. Pierson, B. Kim, and P. Liang, “Concept bottleneck models,” Int. Conf. Machine Learning, pp.5338–5348, 2020.

[38] V.V. Ramaswamy, S.S.Y. Kim, R. Fong, and O. Russakovsky. “Overlooked factors in concept-based explanations: Dataset choice, concept learnability, and human capability,” Computer Vision and Pattern Recognition, pp.10932–10941, 2023.

[39] S. Bach, A. Binder, G. Montavon, F. Klauschen, K. Müller, and W. Samek, “On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation,” PLOS ONE, vol.10, no.7, 2015.

[40] D. Alvarez-Melis and T.S. Jaakkola, “Towards robust interpretability with self-explaining neural networks,” Advances in Neural Information Processing Systems, pp.7786–7795, 2018.

[41] P. Kontschieder, M. Fiterau, A. Criminisi, and S.R. Bulo, “Deep neural decision forests,” Int. Conf. Computer Vision, pp.1467–1475, 2015.

[42] W. Kool, H. van Hoof, and M. Welling, “Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement,” Proc. Int. Conf. Mach. Learn., pp.3499–3508, 2019.

[43] D. Mercier, A. Dengel, and S. Ahmed, “P2exnet: Patch-based prototype explanation network,” International Conference on Neural Information Processing, pp.318–330, 2020.

[44] Y. Ming, P. Xu, H. Qu, and L. Ren, “Prosenet: Interpretable and steerable sequence learning via prototypes,” Proc. 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019.

[45] B. Aken, J.M. Papaioannou, M.G. Naik, G. Eleftheriadis, W. Nejd, F.A. Gers, and A. Löser. “This patient looks like that patient: Prototypical networks for interpretable diagnosis prediction from clinical text,” Asia-Pacific Chapter of the Association for Computational Linguistics for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing, pp.172–184, 2022.

[46] J. Ni, Z. Chen, W. Cheng, B. Zong, D. Song, Y. Liu, X. Zhang, and H. Chen, “Interpreting convolutional sequence model by learning local prototypes with adaptation regularization,” Conf. Information and Knowledge Management, pp.1366–1375, 2021.

[47] Y. Gao, A. Liu, L. Wang, R. Qian, and X. Chen, “A self-interpretable deep learning model for seizure prediction using a multi-scale prototypical part network,” IEEE Trans. Neural System and Rehabilitation Engineering, vol.31, pp.1847–1856, 2023. DOI: 0.1109/TNSRE.2023.3260845.

[48] D. Hong, T. Wang, and S. Baek, “Protorynet - interpretable text classification via prototype trajectories,” J. Machine Learning Research, vol.24, pp.12344–12382, 2023.

[49] S. Azzolin, A. Longa, P. Barbiero, P. Liò, and A. Passerini, “Global explainability of gnns via logic combination of learned concepts,” International Conference on Learning Representations, 2023.

[50] D. Luo, W. Cheng, D. Xu, W. Yu, B. Zong, H. Chen, and X. Zhang, “Parameterized explainer for graph neural network,” Advances in Neural Information Processing Systems, pp.19620–19631, 2020.

[51] E. Dai and S. Wang, “Towards prototype-based self-explainable graph neural network,” arXiv preprint arXiv:2210.01974, 2022.

[52] D. Rymarczyk, D. Dobrowolski, and T. Danel, “Progrest: Prototypical graph regression soft trees for molecular property prediction,” SIAM Int. Conf. Data Mining, pp.379–387, 2023.

[53] A.J. Barnett, F.R. Schwartz, C. Tao, C. Chen, Y. Ren, J.Y. Lo, and C. Rudin, “A case-based interpretable deep learning model for classification of mass lesions in digital mammography,” Nature Machine Intelligence, vol.3, pp.1061–1070, 2021.

[54] E. Kim, S. Kim, M. Seo, and S. Yoon, “Xprotonet: Diagnosis in chest radiography with global and local explanations,” Computer Vision and Pattern Recognition, pp.15719–15728, 2021.

[55] G. Singh and K.C. Yow, “Object or background: An interpretable deep learning model for covid-19 detection,” Diagnostics, vol.11, 2021. DOI: 10.3390/diagnostics11091732.

[56] G. Singh and K.C. Yow, “These do not look like those: An interpretable deep learning model for image recognition,” IEEE Access, vol.9, pp.41482–41493, 2021. DOI: 10.1109/ACCESS.2021.3064838.

[57] D. Rymarczyk, A. Pardyl, J. Kraus, A. Kaczyńska, M. Skomorowski, and B. Zieliński, “Protomil: Multiple instance learning with prototypical parts for whole-slide image classification,” European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, pp.421–436, 2022.

[58] M. Nauta, J.H. Hegeman, J. Geerdink, J. Schlötterer, M. Keulen, and C. Seifert, “Interpreting and correcting medical image classification with pip-net,” European Conference on Artificial Intelligence Workshop, pp.198–215, 2023.

[59] H. Vaseli, A.N. Gu, S.N.A. Amiri, M.Y. Tsang, A. Fung, N. Kondori, A. Saadat, P. Abolmaesumi, and T.S.M. Tsang, “Protoasnet: Dynamic prototypes for inherently interpretable and uncertainty-aware aortic stenosis classification in echocardiography,” arXiv preprint arXiv:2307.14433, 2023.

[60] Y. Wei, R. Tam, and X. Tang, “Mprotonet: A case-based interpretable model for brain tumor classification with 3d multi-

parametric magnetic resonance imaging," Medical Image with Deep Learning, 2023.

[61] Z. Zhang, Q. Liu, H. Wang, C. Lu, and C. Lee, "Protgnn: Towards self-explaining graph neural networks," Association for the Advancement of Artificial Intelligence, pp.9127–9135, 2022.

[62] E.A. Barnes, R.J. Barnes, Z.K. Martin, and J.K. Rader, "This looks like that there: Interpretable neural networks for image tasks when location matters," Artificial Intelligence for the Earth Systems, vol.1, e220001, 2022. DOI:10.1175/AIES-D-22-0001.1

[63] S.F. Stefenon, G. Singh, K. Yow, and A. Cimatti, "Semi-protopnet deep neural network for the classification of defective power grid distribution structures," Sensors, vol.22, no.13, 2022. DOI: 10.3390/s22134859.

[64] T. Jing, H. Xia, R. Tian, H. Ding, X. Luo, J. Domeyer, R. Sherony, and Z. Ding, "Inaction: Interpretable action decision making for autonomous driving," European Conference on Computer Vision, pp.370–387, 2022.

[65] D. Flores-Araiza, F. Lopez-Tiro, J. El-Beze, J. Hubert, M. Gonzalez-Mendoza, G. Ochoa-Ruiz, and C. Daul, "Deep prototypical-parts ease morphological kidney stone identification and are competitively robust to photometric perturbations," Computer Vision and Pattern Recognition Workshop, pp.295–304, 2023.

[66] S. Gulshad, T. Long, and N. van Noord, "Hierarchical explanations for video action recognition," Computer Vision and Pattern Recognition Workshop, pp.3703–3708, 2023.

[67] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra. "Matching networks for one shot learning," Advances in Neural Information Processing Systems, 2016.

[68] P. Hase, C. Chen, O. Li, and C. Rudin, "Interpretable image recognition with hierarchical prototypes," Association for the Advancement of Artificial Intelligence Conference on Human Computation and Crowdsourcing, pp.32–40, 2019.

[69] Z. Xue, L. Duan, W. Li, L. Chen, and J. Luo, "Region comparison network for interpretable few-shot image classification," arXiv preprint arXiv:2009.03558, 2020.

[70] D. Rymarczyk, J. van de Weijer, B. Zieliński, and B. Twardowski, "Icicle: Interpretable class incremental continual learning," Int. Conf. Computer Vision, pp.1887–1898, 2023.

[71] M. Sacha, D. Rymarczyk, Ł. Struski, J. Tabor, and B. Zieliński, "Protoseg: Interpretable semantic segmentation with prototypical parts," Winter Conference on Applications of Computer Vision, pp.1481–1492, 2023.

[72] W. Xu, Y. Xian, J. Wang, B. Schiele, and Z. Akata, "Attribute prototype network for zero-shot learning," Advances in Neural Information Processing Systems, pp.21969–21980, 2020.

[73] P. Chong, N. Cheung, Y. Elovici, and A. Binder, "Toward scalable and unified example-based explanation and outlier detection," IEEE Trans. Image Process., vol.31, pp.525–540, 2022, DOI: 10.1109/TIP.2021.3127847.

[74] F. Ruis, G.J. Burghouts, and D. Bucur, "Independent prototype propagation for zero-shot compositionality," Advances in Neural Information Processing Systems, pp.10641–10653, 2021.

[75] M. Keswani, S. Ramakrishnan, N. Reddy, and V.N. Balasubramanian, "Proto2proto: Can you recognize the car, the way I do?," Computer Vision and Pattern Recognition, pp.10233–10243, 2022.

[76] R. Ragodos, T. Wang, Q. Lin, and X. Zhou, "Protox: Explaining a reinforcement learning agent via prototyping," Advances in Neural Information Processing Systems, 2022.

[77] E.M. Kenny, M. Tucker, and J. Shah, "Towards interpretable deep reinforcement learning with human-friendly prototypes," Int. Conf. Learning Representations, 2023.

[78] A. Bontempelli, S. Teso, K. Tentori, F. Giunchiglia, and A. Passerini, "Concept-level debugging of part-prototype networks," Int. Conf. Learning Representations, 2023.

[79] A.J. Li, R. Netzorg, Z. Cheng, Z. Zhang, and Bin Yu, "Improving prototypical visual explanations with reward reweighing, reselection, and retraining," Int. Conf. Machine Learning, 2024.

[80] Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio, "An empirical investigation of catastrophic forgetting in gradient-based neural networks," Int. Conf. Learning Representations, 2014.

[81] A. Radford, J.W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," Int. Conf. Machine Learning, pp.8748–8763, 2021.

[82] W. Park, D. Kim, Y. Lu, and M. Cho, "Relational knowledge distillation," Computer Vision and Pattern Recognition, pp.3967–3976, 2019.

[83] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," Advances in Neural Information Processing Systems, 2017.

[84] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P. Huang, S. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "Dinov2: Learning robust visual features without supervision," arXiv preprint arXiv:2304.07193, 2023.

[85] G. Ilharco, M. Wortsman, R. Wightman, C. Gordon, N. Carlini, R. Taori, A. Dave, V. Shankar, H. Namkoong, J. Miller, H. Hajishirzi, A. Farhadi, and L. Schmidt, "Openclip," [Online]. Available: https://doi.org/10.5281/zenodo.5143773

[86] N. Park and S. Kim, "How do vision transformers work?," Int. Conf. Learning Representations, 2022.

[87] D. Paul, A. Chowdhury, X. Xiong, F. Chang, D. Carlyn, S. Stevens, K. Provost, A. Karpatne, B. Carstens, D. Rubenstein, C. Stewart, T. Berger-Wolf, Y. Su, and W. Chao, "A simple interpretable transformer for fine-grained image classification and analysis," Int. Conf. Learning Representations, 2024.

[88] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," European Conference on Computer Vision, pp.213–229, 2020.

[89] H. Turbé, M. Bjelogrlic, G. Mengaldo, and C. Lovis, "ProtoS-ViT: Visual foundation models for sparse self-explainable classifications," arXiv preprint arXiv:2406.10025, 2024.

[90] M. Nauta, J. Trienes, S. Pathak, E. Nguyen, M. Peters, Y. Schmitt, J. Schlötterer, M. van Keulen, and C. Seifert, "From anecdotal

evidence to quantitative evaluation methods: A systematic review on evaluating explainable AI," ACM Comput. Surv., 2023.
[91] M. Nauta and C. Seifert, "The co-12 recipe for evaluating interpretable partprototype image classifiers," World Conference Explainable Artificial Intelligence, pp.397–420, 2023.
[92] S. Gautam, M.M.C. Höhne, S. Hansen, R. Jenssen, and M. Kampffmeyer, "This looks more like that: Enhancing self-explaining models by prototypical relevance propagation," Pattern Recognition, 136:109172, 2023.
[93] R. Xu-Darme, G. Quénot, Z. Chihani, and M. Rousset., "Sanity checks and improvements for patch visualisation in prototype-based image classification, " Computer Vision and Pattern Recognition, Workshops, pp.3691–3696, 2023.
[94] M. Nauta, A. Jutte, J. Provoost, and C. Seifert, "This looks like that, because... explaining prototypes for interpretable image recognition," European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, pp.441–456, 2021.
[95] C. Ma, B. Zhao, C. Chen, and C. Rudin, "This looks like those: Illuminating prototypical concepts using multiple visualizations," Advances in Neural Information Processing Systems, 2023.
[96] A. Hoffmann, C. Fanconi, R. Rade, and J. Kohler, "This looks like that· · ·does it? shortcomings of latent space prototype interpretability in deep networks," arXiv preprint arXiv:2105.02968, 2021.
[97] R. Hesse, S. Schaub-Meyer, and S. Roth, "Funnybirds: A synthetic vision dataset for a part-based analysis of explainable ai methods," Int. Conf. Computer Vision, pp.3981–3991, 2023.
[98] S.S.Y. Kim, N. Meister, V.V. Ramaswamy, R. Fong, and O. Russakovsky, "Hive: Evaluating the human interpretability of visual explanations," European Conference on Computer Vision, pp.280–298, 2022.
[99] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "Smoothgrad: Removing noise by adding noise," Workshop on Visualization for Deep Learning, Int. Conf. Machine Learning, 2017.
[100] P. Sinhamahapatra, L. Heidemann, M. Monnet, and K. Roscher, "Towards human-interpretable prototypes for visual assessment of image classification models," Int. Joint Conf. Computer Vision, Imaging and Computer Graphics Theory and Applications, vol.5, pp.878–887, 2023.
[101] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," Computer Vision and Pattern Recognition, pp.248–255, 2009.
[102] S. Santurkar, D. Tsipras, M. Elango, D. Bau, A. Torralba, and A. Madry, "Editing a classifier by rewriting its prediction rules," Advances in Neural Information Processing Systems, pp.1–15, 2021.
[103] K. Meng, A.S. Sharma, A.J. Andonian, Y. Belinkov, and D. Bau, "Mass-editing memory in a transformer," Int. Conf. Learning Representations, 2023.

（2024 年 10 月 3 日受付，2025 年 3 月 31 日再受付，
5 月 19 日早期公開）

鵜飼　祐生

2018 京都大学大学院理学研究科博士前期課程了，2018 グローリー株式会社入社，2024 中部大学大学院工学研究科博士後期課程了（社会人ドクター），2024 同社技師長．現在，同社研究開発本部所属．コンピュータビジョンの研究に従事．2024 電子情報通信学会論文賞．博士（工学）．

佐野　景介

2019 名古屋工業大学大学院博士前期課程了．2019 株式会社デンソー入社．以来コンピュータビジョンの研究に従事．現在同社先端技術研究所所属．2024 中部大学博士後期課程入学．

平川　　翼

2017 広島大学大学院博士課程後期課程情報工学専攻了．2017〜2019 中部大学研究員，2019 同大学特任助教，2021 同大学講師．2014 独立行政法人日本学術振興会特別研究員 DC1．2014〜2015 ESIEE Paris 客員研究員．コンピュータビジョン，パターン認識，医用画像処理の研究に従事．2019 画像の認識・理解シンポジウム (MIRU) フロンティア賞．2019，2020 画像の認識・理解シンポジウム (MIRU) 長尾賞．博士 (工学)．

山下　隆義　（正員）

2002 奈良先端科学技術大学院大学博士前期課程了，2002 オムロン株式会社入社，2011 中部大学大学院博士後期課程了（社会人ドクター），2014 中部大学講師，2017 同大学准教授，2021 同大学教授．人の理解に向けた動画像処理，パターン認識・機械学習の研究に従事．2009 画像センシングシンポジウム高木賞．2013 電子情報通信学会情報・システムソサイエティ論文賞．2013 電子情報通信学会 PRMU 研究会研究奨励賞．2016 画像センシングシンポジウム最優秀学術賞．2019 画像の認識・理解シンポジウム (MIRU) フロンティア賞．2019，2020 画像の認識・理解シンポジウム (MIRU) 長尾賞．2020 電子情報通信学会論文賞．博士 (工学)．

藤吉　弘亘　（正員：フェロー）

1997 中部大学大学院博士後期課程了．1997〜2000 米カーネギーメロン大学ロボット工学研究所 Postdoctoral Fellow．2000 中部大学講師，2004 同大准教授を経て 2010 より同大教授．2005〜2006 米カーネギーメロン大学ロボット工学研究所客員研究員．計算機視覚，動画像処理，パターン認識・理解の研究に従事．2005 ロボカップ研究賞．2009 情報処理学会論文誌コンピュータビジョンとイメージメディア優秀論文賞，2009 山下記念研究賞．2010〜2014 画像センシングシンポジウム優秀学術賞．2013 電子情報通信学会情報・システムソサイエティ論文賞．2019 画像の認識・理解シンポジウム (MIRU) フロンティア賞．2019，2020 画像の認識・理解シンポジウム (MIRU) 長尾賞．2024 電子情報通信学会論文賞．博士 (工学)．情報処理学会，IEEE 各会員．