Hierarchical Cross-Class Part Alignment for Prototypical Part Networks via Hyperbolic Entailment

Keisuke Sano[1,2] **Takayoshi Yamashita**[2] **Hironobu Fujiyoshi**[2]

[1] DENSO CORPORATION [2] Chubu University

Abstract

In fine-grained image recognition, semantic parts (e.g., wings, head) recur across classes while discriminative cues lie in within-part features (e.g., yellow wings, striped wings). The prototypical part network (ProtoPNet) is an interpretable model that provides case-based explanations for image recognition by measuring similarities between local patches in an images and their closest prototypes representing patches in a training set. However, existing ProtoPNet variants do not explicitly model this two-level structure, causing prototypes for the same anatomical part to fragment across classes. We propose a hierarchical prototype learning framework that separates shared prototypes capturing cross-class part concepts from class-specific prototypes capturing intra-part features. To model this hierarchy geometrically, prototypes are embedded in hyperbolic space and an entailment loss constrains each class-specific prototype to lie within the cone of its corresponding shared prototype. To prevent degenerate trivial solutions such as redundant prototypes, we introduce a patch-level contrastive learning using pseudo patch IDs from a vision foundation model, used only at training time, leaving the backbone choice free at inference. Unlike prior hyperbolic prototype methods where prototype hierarchies are uncontrolled, our framework defines explicit shared part to intra-part feature correspondences. Across four ProtoPNet variants and four backbones on CUB-200-2011 and Stanford Cars, our method improves classification accuracy by up to 10 points, raises hierarchical part hierarchy IoU from 11% to 75%, and consistently improves four interpretability metrics on CUB-200-2011.

1 Introduction

While deep neural networks have achieved remarkable accuracy in image recognition, their predictions are often produced by opaque reasoning processes that make it difficult to assess why a model should be trusted. This lack of transparency is a practical barrier to their deployment in high-stakes settings where decisions must be auditable, debuggable, and compliant with accountability requirements, which has motivated extensive research on improving their interpretability [1, 2, 5, 7, 3, 4, 6]. The Prototypical Part Network (ProtoPNet) [7] is a self-explaining model known as the *This Looks Like That* framework. It employs case-based reasoning [8] at the patch level. Given a feature map extracted from an input image, ProtoPNet computes patch-to-prototype similarities and explains its decision by highlighting input regions with the highest similarity to a particular prototypes. Each prototype is visualized as its nearest patch in the training set. ProtoPNet [7] and its variants [9, 10, 11, 12, 13, 14] have primarily been used for fine-grained classification within the same category such as classifying 200 bird species [15] or 196 car models [16]. These tasks involve distinguishing classes by using parts shared across classes (e.g., wings, head, tail) as cues and finding differences in fine-grained features within those parts (e.g., yellow wings, gray wings). However, existing ProtoPNet-based models primarily encourage finding similar local regions (Figure 1-left) and do not explicitly enforce cross-class alignment of prototypes corresponding to the same part, nor do they identify a hierarchical

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

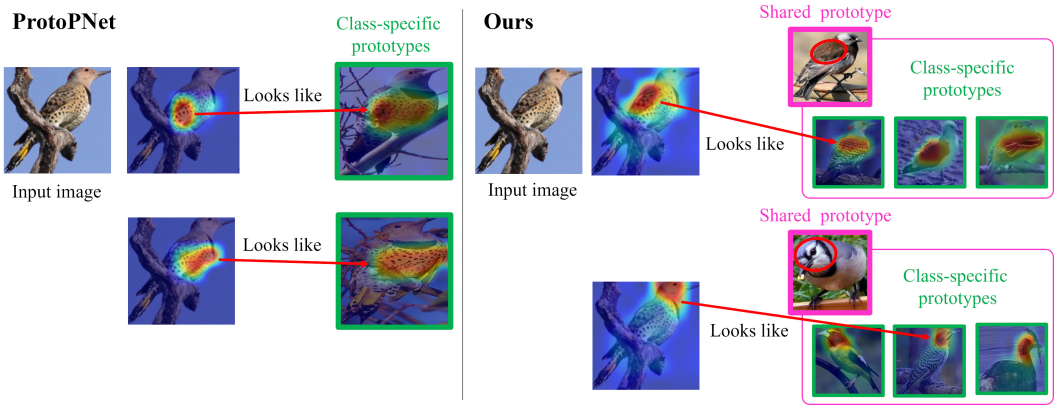


Figure 1: Comparison between ProtoPNet [7] and the proposed framework. ProtoPNet explains reasoning behind its prediction by presenting local regions that contributed to its inference and class-specific prototypes with high similarity to a region. In contrast, the proposed framework identifies local regions that contributed to an inference based on which part group each prototype belong to.

relationship between a shared part and its class-specific intra-part features, which can fragment and duplicate part-derived concepts across classes and hurt generalization.

In this paper, we propose a hierarchical prototype learning framework that improves upon ProtoPNet, as shown in Figure 1-right. The proposed framework introduces shared prototypes as a higher-level concept representing cross-class and part-like semantics and class-specific prototypes as a lower-level concept encoding discriminative intra-part features for each class. To model this hierarchy, we embed prototypes and local patch features in hyperbolic space and optimize an entailment loss that encourages each class-specific prototype to be geometrically contained within its corresponding shared prototype. However, entailment constraints imposed only in feature space can be satisfied even when meaningless solutions such as non-corresponding activated regions between prototypes with a hierarchical pair or learning redundant prototypes between different shared prototype groups. To anchor the hierarchy to consistent input regions, we assign pseudo patch IDs by applying principal component analysis (PCA) and clustering to patches extracted from a vision foundation model (DINOv2) [17], and add a patch-level contrastive loss that encourages corresponding shared and class-specific prototypes to activate on patches with the same pseudo ID. Our proposed framework can be integrated into multiple ProtoPNet variants by replacing Euclidean similarity with hyperbolic similarity, introducing shared prototypes, and adding loss terms without modifying the backbone network or standard prototype-based inference procedure. Experiments were conducted to empirically demonstrate improvements in the classification accuracy, hierarchical feature extraction, and interpretability score of prototypes with the proposed framework.

The contributions of this paper are as follows:

- We propose a hierarchical prototype learning framework for ProtoPNet-based models that separates shared prototypes encoding higher-level cross-class part concepts from class-specific prototypes encoding lower-level intra-part discriminative features. We enforce this hierarchy by using hyperbolic entailment constraints and align activated regions via patch-level contrastive loss using pseudo patch IDs.

- We demonstrate that the proposed framework can be integrated with multiple ProtoPNet variants and backbone networks to improve classification accuracy and prototype interpretability while encouraging cross-class part-feature alignment and hierarchical prototype organization.

2 Related Works

2.1 Prototypical Part Networks

ProtoPNet [7] enables case-based explanations by storing local patches in training images as prototypes and presenting which regions of an input image resemble which prototypes. Since its introduction, various derivatives have been proposed to improve its classification accuracy or interpretability. For example, ProtoConcepts [9] enhances the visual uniqueness of interpretations by associating multiple patches with a single prototype, and EvalProtoPNet [10] improves the consistency and stability of prototype responses by promoting the alignment of feature representations between shallow and deep layers. ProtoPFormer [13] and ProtoViT [14] enable the use of Vision Transformer (ViT) [18] as a backbone. However, while many of these approaches can present similar regions as explanations, they offer limited ability to explicitly learn cross-class correspondences of the same parts (e.g., black head vs red head, brown wing vs striped wing), which are crucial for fine-grained classification of objects within the same category, or to learn the hierarchical structure of higher-level part and lower-level intra-part features. Focusing on this issue, this study aims to construct a part hierarchy by organizing prototypes.

MCPNet [19] organizes channel-segment concepts into a multi-level hierarchy along the network-layer axis and performs classification using their concept distributions. In contrast, our method introduces a cross-class hierarchy over spatial part prototypes through shared and class-specific prototypes with entailment constraints. Thus, MCPNet and our method address fundamentally different hierarchical structures, namely a hierarchy across network layers and a hierarchy across classes, respectively.

With the development of large-scale foundation models [20, 21, 17], Zhu et al. [22] proposed leveraging vision foundation models such as DINO [21] and DINOv2 [17] in ProtoPNet. While they achieved high classification accuracy by utilizing a powerful pre-trained backbone, their approach was limited by needing to use the same backbone during inference. In contrast, the proposed framework differs in that it does not require a foundation model as the backbone; instead, it uses pseudo patch IDs obtained by separately pre-clustering the foundation model's local features as a label signal to induce patch-level alignment during the training of ProtoPNet-based models with an arbitrary backbone.

Several methods for learning class-agnostic prototype sets and class assignments [23, 24, 25, 26] have been proposed to improve the memory efficiency of ProtoPNet. However, these methods focus primarily on learning the correspondence between prototypes and classes and require specialized model architectures or post-training. Thus, their purpose differs from that of our study, which introduces shared prototypes as higher-level concepts of shared body parts (not discriminative feature for classification themselves), designed to encompass corresponding class-specific prototypes. In this study, we apply our framework to base models that utilize class-specific prototypes.

2.2 Hyperbolic Embeddings for Learning Hierarchy

Hyperbolic embeddings have been widely used to represent hierarchical structures [27, 28, 29, 30, 31]. Since the volume of hyperbolic space grows exponentially with radius, tree-like hierarchies can be embedded with low distortion by placing more general concepts near the origin and more specific concepts farther from it. Ganea et al. [28] introduced hyperbolic entailment cones, whose aperture narrows with distance from the origin, and trained embeddings so that each lower-level concept lies inside the cone of its higher-level concept.

However, in most existing approaches, the hierarchical relations are defined over whole images, captions, or class labels. Pal et al. [30] additionally considers object boxes, but these are image crops extracted by an external grounding model and encoded as separate inputs, rather than local patch features within the network. Prototype-based methods such as those of Fonio et al. [32] and Berg et al. [33] also define prototypes at the class level. Few studies have addressed hierarchies among local part features. Among the derivatives of ProtoPNet, HAPPI [34] proposed a framework that introduces generic prototypes for each class and trains each class-specific prototype to be entailed in at least one generic prototype in each class in hyperbolic space. HiProtoNet [35] applies a similar hyperbolic part-prototype framework as HAPPI to medical image classification. Nevertheless, HAPPI does not explicitly specify which generic prototype entails each specific prototype, nor does it establish

part-level correspondence across classes. In contrast, our approach fixes the hierarchical pairing of each part (shared) with its corresponding intra-part attributes (class-specific) for the entailment loss and enforces spatial part alignment across classes via patch-level learning using pseudo patch IDs.

3 Method

In this section, we describe the proposed framework shown in Figure 2, which is added on top of a baseline. We first recap ProtoPNet [7] training as a baseline in Section 3.1. Next, section 3.2 introduces the projection from Euclidean space to hyperbolic space and similarity computation between prototypes and feature vectors in our approach. Then, Section 3.3 presents how we train hierarchical relationships in hyperbolic space between shared prototypes and class-specific prototypes via entailment loss. Finally, section 3.4 describes patch-level contrastive learning using pseudo patch IDs.

3.1 Preliminary

ProtoPNet [7] comprises three basic components: a backbone, prototype layer, and fully connected (FC) layer. The backbone maps an input image x_i to a feature map $\mathbf{Z}_i \in \mathbb{R}^{H\times W\times D}$ with resolution of $(H \times W)$ and feature dimension D. The subsequent prototype layer computes similarities between a set of class-specific prototypes $\mathbf{P} \in \mathbb{R}^{C\times K\times D}$ and $\mathbf{Z}_i$, where C is the number of classes and K is the number of prototypes per class. Here, $\mathbf{P}_k^c \in \mathbb{R}^D$ denotes the k-th prototype of class c. The similarity $\theta\left(z, p\right)$ between a feature vector $z \in \mathbb{R}^D$ in $\mathbf{Z}_i$ and a prototype p is computed as follows:

$$\theta\left(z, p\right) = \log\left(1 + \frac{1}{d(z, p) + \epsilon}\right). \tag{1}$$

where, $d(z, p) = \|z - p\|_2^2$ and ϵ is a small constant added for numerical stability. Finally, the FC layer with the weights matrix $\mathbf{W} \in \mathbb{R}^{C\times(C\times K)}$ computes the logit for each class. The logit $\hat{y}_i^c$ for class c is expressed as follows using the similarity between the prototypes and feature map:

$$\hat{y}_i^c = \sum_{c'=1}^{C}\sum_{k=1}^{K} \mathbf{W}_{c,(c'-1)\times K+k} \cdot g_{i,k}^{c'}, \text{ where } g_{i,k}^{c'} = \max_{\substack{1\le h\le H\\ 1\le w\le W}} \theta\left(\mathbf{Z}_{i,h,w}, \mathbf{P}_k^{c'}\right). \tag{2}$$

ProtoPNet trains each class-specific prototype to identify features characteristic of the associated class, which involves minimizing not only the cross-entropy loss L_{ce} , but also the cluster loss L_{clst} and separation loss L_{sep}. The cluster loss encourages patch features in the feature map to be close to prototypes of the input image class. In contrast, the separation loss encourages patch features to be far from prototypes that do not belong to that class. Let B denote the set of data indices in a mini-batch. Then, these losses can be formulated as follows:

$$L_{\text{clst}} = \frac{1}{|\text{B}|}\sum_{i\in\text{B}} \min_{\substack{1\le k\le K\\ 1\le h\le H\\ 1\le w\le W}} d(\mathbf{Z}_{i,h,w}, \mathbf{P}_k^{y_i}), \quad L_{\text{sep}} = -\frac{1}{|\text{B}|}\sum_{i\in\text{B}} \min_{c\neq y_i} \min_{\substack{1\le k\le K\\ 1\le h\le H\\ 1\le w\le W}} d(\mathbf{Z}_{i,h,w}, \mathbf{P}_k^{c}). \tag{3}$$

3.2 Hyperbolic-based Similarity

Projection to Hyperbolic Space

The proposed framework computes the similarity between the feature map and prototypes in hyperbolic space to learn a hierarchical relationship between corresponding shared and class-specific prototypes. Following the Lorentz-model formulations of Desai et al. [29], we project patch features and prototypes from Euclidean space into hyperbolic space and compute their hyperbolic distances. Let $V_{\text{euc}} \in \mathbb{R}^D$ denote a vector in the Euclidean space, and let $V_{\text{hyp}} \in \mathbb{R}^{D+1}$ denote its mapped vector in the hyperbolic space. V_{hyp} comprises a spatial component $V_{\text{space}} \in \mathbb{R}^D$ and temporal component $V_{\text{time}} \in \mathbb{R}$, which are respectively computed by the following transformations:

$$V_{\text{space}} = \frac{\sinh\left(\sqrt{\kappa}\|V_{\text{euc}\|}\right)}{\sqrt{\kappa}\|V_{\text{euc}\|}} V_{\text{euc}}, \tag{4}$$

$$V_{\text{time}} = \sqrt{\frac{1}{\kappa} + \|V_{\text{space}}\|^2}. \tag{5}$$

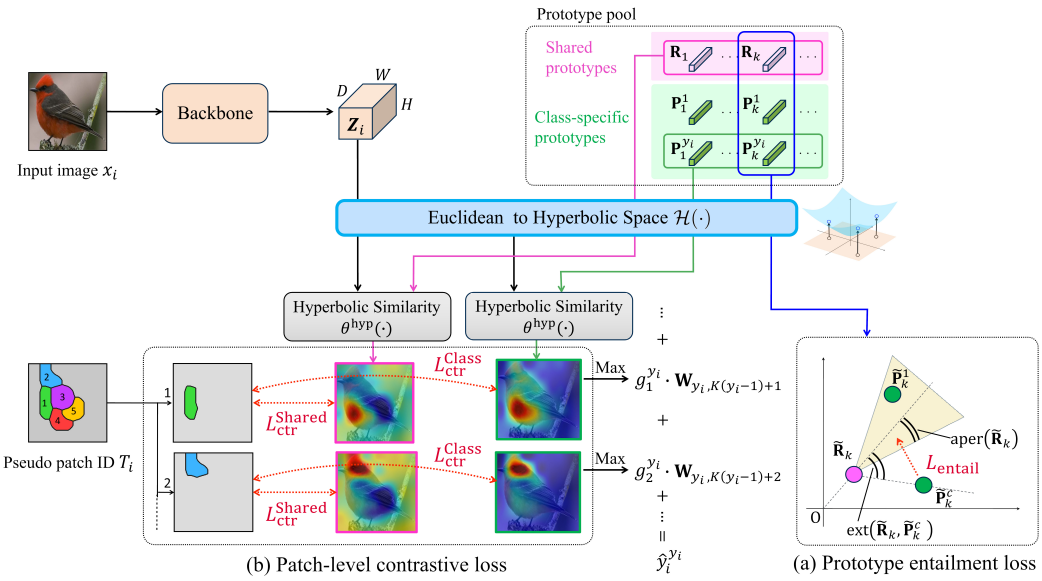


Figure 2: Overview of the proposed framework. The feature map $\mathbf{Z}_i$ obtained from the input image x_i is projected into hyperbolic space via $\mathcal{H}(\cdot)$ along with the shared prototypes $\mathbf{R}$ and class-specific prototypes $\mathbf{P}$. (a) Prototypical entailment loss L_{entail} encourages each shared prototype to entail its corresponding class-specific prototype. (b) Patch-level contrastive losses $L_{\text{ctr}}^{\text{Shared}}$ and $L_{\text{ctr}}^{\text{Class}}$ spatially align the activated regions calculated with $\theta^{\text{hyp}}(\cdot)$ across the corresponding shared prototypes and class-specific prototypes using pseudo patch IDs.

Here, κ denotes the curvature. Let $\mathcal{H}(\cdot)$ denote the above mapping from the Euclidean space to the hyperbolic space. Then, a feature vector z in the feature map $\mathbf{Z}$ and a prototype p are projected into the hyperbolic space as $\tilde{z} = \mathcal{H}(z), \tilde{p} = \mathcal{H}(p)$.

Similarity in Hyperbolic Space

The distance $d_L(a, b)$ between two vectors in hyperbolic space $a = [a_{\text{time}}, a_{\text{space}}]$ and $b = [b_{\text{time}}, b_{\text{space}}]$ is computed as follows:

$$d_L(a, b) = \frac{1}{\sqrt{\kappa}} \cosh^{-1}\left(-\kappa \langle a, b \rangle_L\right). \tag{6}$$

Here, $\langle a, b \rangle_L$ denotes the Lorentz inner product:

$$\langle a, b \rangle_L = \langle a_{\text{space}}, b_{\text{space}} \rangle - a_{\text{time}} b_{\text{time}}, \tag{7}$$

where $\langle a_{\text{space}}, b_{\text{space}} \rangle$ is the standard inner product.

Therefore, the similarity between a feature vector $\tilde{z}$ and prototype $\tilde{p}$ in hyperbolic space is computed by replacing the distance $d(z, p)$ in Equation 1 with $d_L(\tilde{z}, \tilde{p})$:

$$\theta^{\text{hyp}}(\tilde{z}, \tilde{p}) = \log\left(1 + \frac{1}{d_L(\tilde{z}, \tilde{p}) + \epsilon}\right). \tag{8}$$

Accordingly, the loss terms L_{clst} and L_{sep} in Equation 3 use $d_L(\tilde{z}, \tilde{p})$ as well.

3.3 Learning Part Hierarchy Using Entailment Loss

In the proposed framework, the model learns the hierarchical relationship between the shared prototypes $\mathbf{R} \in \mathbb{R}^{K \times D}$ and class-specific prototypes $\mathbf{P} \in \mathbb{R}^{C \times K \times D}$. Each class-specific prototype captures a class-specific feature, and class-specific prototypes that capture the same part concept are hierarchically entailed by the corresponding shared prototype. This hierarchical entailment can be geometrically formulated in hyperbolic space. A pair of two vectors a and b has a hierarchical

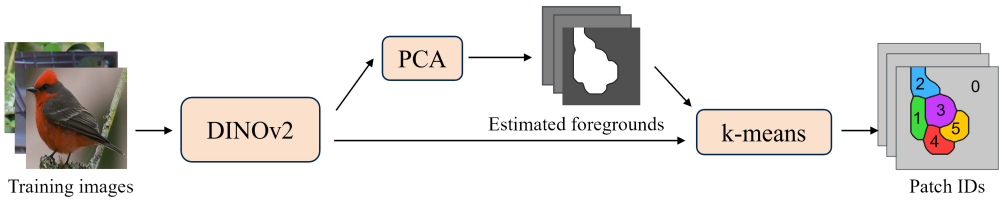


Figure 3: Assignment of pseudo patch IDs. The training images are fed into the DINOv2 backbone to obtain feature vectors. PCA is performed to estimate foreground patches. k-means clustering is applied to the patches.

relation when b lies inside a cone with a top a that expands in the direction away from the origin O. The cone with its half-aperture at a is defined as follows:

$$\mathrm{aper}(a) = \sin^{-1}\left(\frac{2Q}{\sqrt{\kappa}\|a_{\mathrm{space}}\|}\right), \tag{9}$$

where Q is a hyperparameter that controls the half-aperture angle, which we set to $Q = 0.1$ in this paper. Then, the exterior angle of $\angle Oab$ can be defined as follows:

$$\mathrm{ext}(a,b) = \cos^{-1}\left(\frac{b_{\mathrm{time}} + a_{\mathrm{time}} \cdot \kappa\langle a,b\rangle_L}{\|a_{\mathrm{space}}\|\sqrt{\left(\kappa\,\langle a,b\rangle_L\right)^2 - 1}}\right). \tag{10}$$

When $\mathrm{ext}(a,b) < \mathrm{aper}(a)$, then b is contained in a. Accordingly, we use the difference between $\mathrm{aper}(a)$ and $\mathrm{ext}(a,b)$ as a penalty.

To encourage $\mathbf{R}_k$ to entail $\mathbf{P}_k^c$ in the hyperbolic space, we define the prototype entailment loss:

$$L_{\mathrm{entail}} = \frac{1}{KC}\sum_{k=1}^{K}\sum_{c=1}^{C}\max\left(0, \mathrm{ext}(\tilde{\mathbf{R}}_k, \tilde{\mathbf{P}}_k^c) - \mathrm{aper}(\tilde{\mathbf{R}}_k)\right), \tag{11}$$

where $\tilde{\mathbf{R}}_k = \mathcal{H}(\mathbf{R}_k)$ and $\tilde{\mathbf{P}}_k^c = \mathcal{H}(\mathbf{P}_k^c)$. With this loss, the k-th class-specific prototype for each class is trained to be entailed in the k-th shared prototype.

3.4 Part Alignment Using Patch-level Contrastive Losses

A shared prototype $\mathbf{R}_k$ and its corresponding class-specific prototypes $\mathbf{P}_k^c$ are required to activate the same part region in the same input image x_i. Moreover, each shared prototype $\mathbf{R}_k$ is required to capture a different part of the target object from the other shared prototypes. The proposed framework leverages a vision foundation model to assign pseudo patch IDs to each training image and introduces patch-level contrastive learning based on these IDs.

As illustrated in Figure 3, pseudo patch IDs are assigned to each patch in $\mathbf{Z}_i$ on the target backbone by clustering the feature vectors, which are extracted by feeding the training images into DINOv2 [17] into $K+1$ clusters. There are several approaches to unsupervised part segmentation in the literature [36, 37], and our method for assigning part IDs to each patch is like of Amir et al. [36]. The proposed framework assigns pseudo patch IDs to all training images using PCA and k-means clustering: first, images are fed into DINOv2 [17] and resize each resulting feature map to the spatial resolution of the target backbone with bilinear interpolation; second, the foreground and background are separated by thresholding the first principal component obtained by PCA; finally, feature patches in the foreground region are clustered by k-means clustering. In this paper, we utilize the first five principal components for clustering. The resulting ID map $T_i \in \mathbb{Z}^{H\times W}$ takes an integer value $T_{i,h,w}$ at the position (h,w) where $k = 0, 1, ..., K$. Here, $k = 0$ denotes the background region, and $k = 1, 2, \ldots, K$ denote different parts. These IDs must be consistent across images; in other words, patches assigned the same ID should correspond to the same part in any image. To satisfy this requirement, the proposed framework first samples a subset of the training images, performs PCA and k-means clustering, and then reuses the PCA projection matrix and k-means clustering coordinates for all images.

Table 1: Classification accuracy with/without the proposed framework for different baseline methods and backbones applied to the fine-grained image datasets CUB200-2011 [15] and Stanford Cars [16].

Baseline Method	Backbone	Ours	CUB200-2011[15]	Stanford Cars [16]
ProtoPNet [7]	ResNet50 [38]		74.39 ± 0.19	84.42 ± 0.17
		✓	$\mathbf{78.67 \pm 0.78}$	$\mathbf{89.20 \pm 0.38}$
	DenseNet161 [39]		73.85 ± 0.09	88.16 ± 0.12
		✓	$\mathbf{84.15 \pm 0.17}$	$\mathbf{91.16 \pm 0.16}$
ProtoConcepts [9]	ResNet50 [38]		77.54 ± 0.70	87.05 ± 0.45
		✓	$\mathbf{82.91 \pm 1.06}$	$\mathbf{90.03 \pm 0.15}$
	DenseNet161 [39]		79.57 ± 0.72	89.39 ± 0.21
		✓	$\mathbf{84.22 \pm 0.10}$	$\mathbf{91.09 \pm 0.28}$
ProtoPFormer [13]	DeiT-Small [40]		80.84 ± 0.47	81.53 ± 0.60
		✓	$\mathbf{83.03 \pm 0.29}$	$\mathbf{85.78 \pm 0.09}$
	CaiT-XXS24 [41]		82.68 ± 0.40	83.82 ± 0.33
		✓	$\mathbf{82.84 \pm 0.24}$	$\mathbf{87.54 \pm 0.72}$
HAPPI [34]	ResNet50 [38]		72.92 ± 0.84	84.07 ± 0.29
		✓	$\mathbf{78.12 \pm 0.28}$	$\mathbf{88.37 \pm 0.10}$
	DenseNet161 [39]		77.27 ± 0.46	86.03 ± 1.16
		✓	$\mathbf{83.51 \pm 0.13}$	$\mathbf{90.17 \pm 1.24}$
	DeiT-Small [40]		81.01 ± 0.27	81.37 ± 2.16
		✓	$\mathbf{82.12 \pm 0.41}$	$\mathbf{83.33 \pm 1.21}$
	CaiT-XXS24 [41]		80.66 ± 1.84	84.98 ± 3.71
		✓	$\mathbf{82.19 \pm 0.53}$	$\mathbf{86.44 \pm 1.41}$

The assigned IDs are used as pseudo supervision to align the patches that the shared and class-specific prototypes should activate. For each patch in the feature map, the proposed framework encourages features to have high similarity to the prototype corresponding to the assigned ID by introducing patch-level contrastive losses $L_{\text{ctr}}^{\text{Class}}$ and $L_{\text{ctr}}^{\text{Shared}}$ defined as follows:

$$L_{\text{ctr}}^{\text{Class}} = -\frac{1}{|\text{B}|} \sum_{i \in \text{B}} \sum_{\substack{1 \le h \le H \\ 1 \le w \le W \\ T_{i,h,w} \neq 0}} \log \frac{\exp\left(\theta^{\text{hyp}}\left(\tilde{\mathbf{Z}}_{i,h,w}, \tilde{\mathbf{P}}^{y_i}_{T_{i,h,w}}\right)/\tau\right)}{\sum_{c=1}^{C}\sum_{k=1}^{K} \exp\left(\theta^{\text{hyp}}\left(\tilde{\mathbf{Z}}_{i,h,w}, \tilde{\mathbf{P}}^{c}_{k}\right)/\tau\right)}, \tag{12}$$

$$L_{\text{ctr}}^{\text{Shared}} = -\frac{1}{|\text{B}|} \sum_{i \in \text{B}} \sum_{\substack{1 \le h \le H \\ 1 \le w \le W \\ T_{i,h,w} \neq 0}} \log \frac{\exp\left(\theta^{\text{hyp}}\left(\tilde{\mathbf{Z}}_{i,h,w}, \tilde{\mathbf{R}}_{T_{i,h,w}}\right)/\tau\right)}{\sum_{k=1}^{K} \exp\left(\theta^{\text{hyp}}\left(\tilde{\mathbf{Z}}_{i,h,w}, \tilde{\mathbf{R}}_{k}\right)/\tau\right)}, \tag{13}$$

where τ is the temperature parameter. In this paper, we set to $\tau = 0.1$. By using the same ID map T_i for both $L_{\text{ctr}}^{\text{Class}}$ and $L_{\text{ctr}}^{\text{Shared}}$, the k-th shared prototype $\mathbf{R}_k$ and k-th class-specific prototype $\mathbf{P}_k^c$ of each class are trained to focus on the same region, which ensures spatially consistent part alignment.

The proposed framework optimizes an augmented loss function that adds L_{entail}, $L_{\text{ctr}}^{\text{Class}}$, and $L_{\text{ctr}}^{\text{Shared}}$ to the standard ProtoPNet loss function $L_{\text{protopnet}}$:

$$L_{\text{total}} = L_{\text{protopnet}} + \lambda_{\text{entail}} L_{\text{entail}} + \lambda_{\text{ctr}}^{\text{Class}} L_{\text{ctr}}^{\text{Class}} + \lambda_{\text{ctr}}^{\text{Shared}} L_{\text{ctr}}^{\text{Shared}}, \tag{14}$$

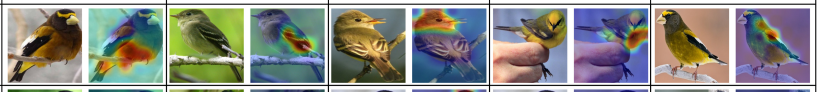
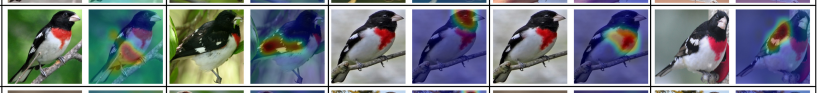
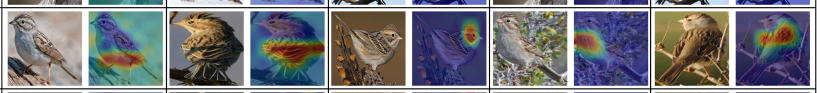
where λ_{entail}, $\lambda_{\text{ctr}}^{\text{Class}}$, and $\lambda_{\text{ctr}}^{\text{Shared}}$ are hyperparameters.

4 Experiments

4.1 Comparison of Classification Accuracy

Experiments were conducted to evaluate the classification accuracy of ProtoPNet [7], ProtoConcepts [9], ProtoPFormer [13], and HAPPI [34] when applied to the fine-grained image datasets CUB200-2011 [15] and Stanford Cars [16] with and without the proposed framework. We used ResNet50 [38] and DenseNet161 [39] as the backbones for the convolutional neural network (CNN)-based models and DeiT-Small [40] and CaiT-XXS24 [41] as the backbones for the ViT-based models.

CUB200-2011

Prototype	Class name	k				
		1	2	3	4	5
Shared	-					
Class-specific	Rose breasted Grosbeak					
	House Sparrow					
	Brewer Sparrow					
	California Gull					

Stanford Cars

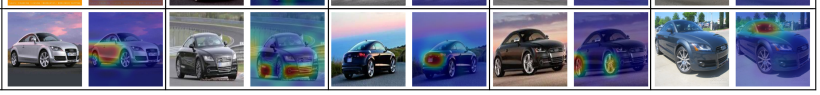
Prototype	Class name	k				
		1	2	3	4	5
Shared	-					
Class-specific	MINI Cooper Roadster Convertible 2012					
	Acura RL Sedan 2012					
	Audi 100 Wagon 1994					
	Audi S5 Coupe 2012					

Figure 4: Visual evaluation of trained prototypes from CUB200-2011 [15] and Stanford Cars [16]. The shared prototypes capture different body parts of a bird for different indices k. The k-th class-specific prototype for each class corresponds to the same body part as the k-th shared prototype.

The learning rates for the newly introduced trainable parameters in the proposed framework were set to 3×10^{-3} for the shared prototypes $\mathbf{R}$ and 5×10^{-4} for the hyperbolic curvature parameter κ. The hyperparameters of the loss function in Equation 14 were set to $\lambda_{\text{entail}} = 0.1$, $\lambda_{\text{ctr}}^{\text{Class}} = 0.8$, and $\lambda_{\text{ctr}}^{\text{Shared}} = 0.1$. We set the number of prototypes per class (i.e., total number of parts comprising the target object) to $K = 5$, set the batch size to 128, set the feature dimension to $D = 512$, and employed the Adam optimizer [42]. For other settings, we mainly followed the original papers or implementations of the released code. Further experimental details are described in Appendix B. For HAPPI [34], the training code was not publicly available, so we implemented it ourselves based on the description in the original paper. The results are summarized in Table 1. The accuracy was assessed by calculating the average and standard deviation across three trials for each condition and model. Overall, applying the proposed framework improved the accuracy in every case.

4.2 Evaluations of Part Hierarchy

We first visually evaluated the prototypes trained by the proposed framework. As shown in Figure 4, shared and class-specific prototypes with the same prototype index k captured the same part across classes while prototypes with different indices k captured different parts.

Table 2: Evaluation of the part hierarchy based on the average IoU between regions with high similarity among prototypes sharing an entailed relationship.

Backbone	Method	CUB200-2011 [15]	Stanford Cars [16]
ResNet50 [38]	HAPPI [34]	9.65 ± 0.27	9.12 ± 0.07
	ProtoPNet [7] + Ours	$\mathbf{74.77 \pm 0.78}$	$\mathbf{62.29 \pm 0.48}$
DenseNet161 [39]	HAPPI [34]	10.94 ± 0.37	7.76 ± 0.21
	ProtoPNet [7] + Ours	$\mathbf{69.42 \pm 1.91}$	$\mathbf{58.39 \pm 1.77}$

Table 3: Metrics used to evaluate the interpretability of prototypes obtained with and without applying the proposed framework to EvalProtoPNet [10] with ResNet50 [38] on CUB200-2011[15].

Ours	Accuracy	Consistency	Stability	Distinctiveness	Comprehensiveness
	81.22 ± 0.19	68.87 ± 2.19	53.38 ± 0.51	52.40 ± 2.96	34.58 ± 0.11
✓	$\mathbf{82.38 \pm 0.19}$	$\mathbf{85.00 \pm 0.49}$	$\mathbf{59.06 \pm 0.40}$	$\mathbf{79.16 \pm 0.37}$	$\mathbf{43.80 \pm 0.02}$

Then, we quantitatively evaluated the part hierarchy by computing the intersection over union (IoU) between a shared prototype and its entailed class-specific prototypes. We introduce the metric S_{part} to evaluate whether a pair of prototypes with an entailment relationship in the hyperbolic space activates a matched region. Let $\hat{M}_i^{\tilde{p}}$ is a binarized activation map for a prototype $\tilde{p}$ of x_i calculated from an activation map $A_i^{\tilde{p}}$. Here, $A_{i,h,w}^{\tilde{p}} = \theta^{\mathrm{hyp}}(\tilde{\mathbf{Z}}_{i,h,w}, \tilde{p})$. Each coordinate of $\hat{M}_i^{\tilde{p}}$ is defined as:

$$\hat{M}_{i,h,w}^{\tilde{p}} = \begin{cases} 1 \ \text{ if } = \bar{A}_{i,h,w}^{\tilde{p}} \geq \rho \\ 0 \ \text{ otherwise.} \end{cases}, \ \text{ where } \bar{A}_{i,h,w}^{\tilde{p}} = \frac{A_{i,h,w}^{\tilde{p}} - \mathrm{Min}(A_i^{\tilde{p}})}{\mathrm{Max}(A_i^{\tilde{p}}) - \mathrm{Min}(A_i^{\tilde{p}})}. \tag{15}$$

In this paper, we set $\rho = 0.5$. Then, S_{part} is computed as:

$$S_{\mathrm{part}} = \frac{\sum_{i \in \mathcal{D}} \sum_{k=1}^{K} \mathbb{1}\{\mathrm{ext}(\tilde{\mathbf{R}}_k, \tilde{\mathbf{P}}_k^{y_i}) - \mathrm{aper}(\tilde{\mathbf{R}}_k) \leq 0\} \mathrm{IoU}(\hat{M}_i^{\mathbf{R}_k}, \hat{M}_i^{\mathbf{P}_k^{y_i}})}{\sum_{i \in \mathcal{D}} \sum_{k=1}^{K} \mathbb{1}\{\mathrm{ext}(\tilde{\mathbf{R}}_k, \tilde{\mathbf{P}}_k^{y_i}) - \mathrm{aper}(\tilde{\mathbf{R}}_k) \leq 0\}}, \tag{16}$$

where $\mathcal{D}$ is a set of test image indices. We compared the proposed framework with HAPPI [34], which also trains a hierarchical feature structure in a hyperbolic space by introducing class-specific and generic prototypes. To compute S_{part} for HAPPI, $\mathbf{R}$ was replaced with the generic prototype. Each value shows the average and standard deviation across three trials. Table 2 presents the results, as the average and standard deviation across three trials, and indicates that the proposed framework more successfully captures the same part region between prototypes with an entailed relationship than HAPPI does. We note that this metric requires prototypes to be organized into hierarchical pairs, which a plain ProtoPNet does not have; hence a bare "ProtoPNet vs. ProtoPNet + Ours" comparison on this metric is undefined, and the meaningful baseline is HAPPI, which does impose such hierarchy pairs.

4.3 Evaluation of Other Interpretability Scores

We evaluated the interpretability of prototypes obtained by the proposed framework by using four metrics: the consistency [10] of high similarity regions for each prototype across images, the stability [10] of high similarity regions against input perturbations, the distinctiveness [22] of high similarity regions to ensure that they do not overlap across different prototypes, and the comprehensiveness [22] to measure whether prototypes avoid activating irrelevant regions outside the target object such as the background. The detailed descriptions and implementations for this evaluation are stated in Appendix C. Table 3 presents the classification accuracy and interpretability according to these four metrics when the proposed framework was applied to the EvalProtoPNet [10] using ResNet50 [38] as a backbone on CUB200-2011 [15]. Each value shows the average and standard deviation across three trials. Applying our method improves accuracy and all four interpretability metrics. These improvements are in line with how our method works. Using pseudo part IDs that partition the estimated foreground into distinct, image-consistent part regions, our contrastive losses anchor each prototype to its corresponding part region. This can naturally contribute to Consistency (consistent part correspondence across images), Distinctiveness (different prototypes capturing different regions), and Comprehensiveness (coverage of the foreground).

Table 4: Ablation study on entailment loss and patch-level contrastive losses for ProtoPNet [7] with DenseNet161 [39] in CUB200-2011 [15].

	$L_{\text{ctr}}^{\text{Class}}$		Accuracy	Distinctiveness
Training in Euclidean space			73.85 ± 0.09	32.73 ± 4.58
	✓		81.02 ± 0.59	62.07 ± 0.94
	L_{entail}	$L_{\text{ctr}}^{\text{Class}}$ and $L_{\text{ctr}}^{\text{Shared}}$	Accuracy	Distinctiveness
Training in Hyperbolic space			75.51 ± 0.13	6.71 ± 2.24
	✓		76.23 ± 0.47	7.53 ± 3.01
		✓	82.89 ± 0.29	87.41 ± 0.49
	✓	✓	$\mathbf{84.15 \pm 0.17}$	$\mathbf{89.29 \pm 0.08}$

4.4 Ablation Study

We conducted an ablation study on hyperbolic projection and the loss terms comprising the proposed framework for ProtoPNet [7] with DenseNet161 [39] on CUB200-2011 [15]. The upper part of Table 4 (Training in Euclidean space) reports the effect of the patch-level contrastive learning using $L_{\text{ctr}}^{\text{Class}}$ while training ProtoPNet in Euclidean space. L_{entail} and $L_{\text{ctr}}^{\text{Shared}}$ are used only in the settings with hyperbolic projection. The lower part of Table 4 (Training in Hyperbolic space) reports the impact of the entailment loss L_{entail} and the patch-level contrastive learning $L_{\text{ctr}}^{\text{Class}}$ and $L_{\text{ctr}}^{\text{Shared}}$. Each value shows the average and standard deviation across three trials. Using neither the entailment loss nor the patch-level contrastive losses corresponded to applying only the projection into the hyperbolic space. Leveraging hyperbolic space or entailment loss improved classification accuracy but degraded distinctiveness. This suggests that the model is prone to learning redundant prototypes. Meanwhile, introducing patch-level contrastive learning improved distinctiveness and combining the entailment and patch-level contrastive losses with hyperbolic projection achieved the highest accuracy and distinctiveness.

5 Limitations

The patch-level contrastive learning of the proposed framework relies on pseudo patch IDs obtained by clustering DINOv2 features. Consequently, the part-level semantic precision of each ID is limited by the foundation model's representational quality. In specific domains where foundation models are insufficiently pre-trained (e.g. medical images, remote sensing, industrial inspection,) the resulting clusters may not correspond to meaningful object parts and potentially degrade the part-alignment signal performance. Although a robust performance was confirmed on the fine-grained image datasets CUB-200-2011 and Stanford Cars, extending the proposed framework to such specialized domains will likely require domain-adapted foundation models or alternative ID assignment methods. Additionally, our method targets fine-grained tasks in which classes share a common structure (e.g., all birds share wings / beaks / legs), which is precisely what enables shared part prototypes. For coarse-grained settings such as ImageNet [43], where such shared structure is ill-defined, additional design would be needed, for example, grouping classes via super-class information and defining prototype correspondence within groups.

6 Conclusion

The proposed framework learns part hierarchies and cross-class part alignment for fine-grained image classification. By introducing shared and class-specific prototypes, enforcing hyperbolic entailment constraints, and adding patch-level contrastive learning based on pseudo patch IDs derived from foundation-model features, the proposed framework encourages consistent activation of corresponding parts across classes. Extensive experiments across multiple datasets demonstrated that it consistently improved the classification accuracy and interpretability of multiple ProtoPNet variants and backbones as well as achieved cross-class alignment by the hierarchical organization of prototypes.

References

[1] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning Deep Features for Discriminative Localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2921–2929, 2016.

[2] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual Explanations From Deep Networks via Gradient-Based Localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 618–626, 2017.

[3] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International Conference on Machine Learning*, pages 5338–5348, PMLR, 2020.

[4] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International Conference on Machine Learning*, pages 2673–2682, PMLR, 2018.

[5] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. In *Proceedings of the British Machine Vision Conference*, pages 1–17, 2018.

[6] Yu Zhang, Peter Tino, Aleš Leonardis, and Ke Tang. A survey on neural network interpretability. In *IEEE Transactions on Emerging Topics in Computer Intelligence*. volume 5, no. 5, pages 726–742, 2021.

[7] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: Deep learning for interpretable image recognition. *Advances in Neural Information Processing Systems*, pages 8928–8939, 2019.

[8] Agnar Aamodt and Enric Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. In AI AI Communications, volume 7, pages 39–59, 1994.

[9] Chiyu Ma, Brandon Zhao, Chaofan Chen, and Cynthia Rudin. This Looks Like Those: Illuminating Prototypical Concepts Using Multiple Visualizations. *Advances in Neural Information Processing Systems*, volume 36, pages 39212–39235, 2023.

[10] Qihan Huang, Mengqi Xue, Wenqi Huang, Haofei Zhang, Jie Song, Yongcheng Jing, and Mingli Song. Evaluation and improvement of interpretability for self-explainable partprototype networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2011–2020, 2023.

[11] Jon Donnelly, Alina Jade Barnett, and Chaofan Chen. Deformable ProtoPNet: An Interpretable Image Classifier Using Deformable Prototypes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10265–10275, 2022.

[12] Chong Wang, Yuyuan Liu, Yuanhong Chen, Fengbei Liu, Yu Tian, Davis McCarthy, Helen Frazer, and Gustavo Carneiro. Learning support and trivial prototypes for interpretable image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2062–2072, 2023.

[13] Xue Mengqi, Huang Qihan, Zhang Haofei, Hu Jingwen, Song Jie, Song Mingli, and Jin Canghong. ProtoPFormer: Concentrating on Prototypical Parts in Vision Transformers for Interpretable Image Recognition. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 1516–1524, 2022.

[14] Ma Chiyu, Donnelly Jon, Liu Wenjun, Vosoughi Soroush, Rudin Cynthia, and Chen Chaofan. Interpretable Image Classification with Adaptive Prototype-based Vision Transformers. *Advances in Neural Information Processing Systems*. volume 37, pages 41447–41493, 2024.

[15] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltechucsd birds–200–2011 dataset. Technical Report CNS–TR–2011–001, California Institute of Technology, 2011.

[16] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR–13)*, Sydney, Australia, 2013.

[17] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *The Ninth International Conference on Learning Representations*, 2021.

[19] Bor-Shiun Wang, Chien-Yi Wang, and Wei-Chen Chiu. MCPNet: An Interpretable Classifier via Multi-level Concept Prototypes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10885–10894, 2024.

[20] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and Gretchen Krueger, Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763, PMLR, 2021.

[21] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021.

[22] Zhijie Zhu, Lei Fan, Maurice Pagnucco, and Yang Song. Interpretable Image Classification via Non-parametric Part Prototype Learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9762–9771, 2025.

[23] Dawid Rymarczyk, Łukasz Struski, Jacek Tabor, and Bartosz Zielinski. ProtoPShare: Prototypical Parts Sharing for Similarity Discovery in Interpretable Image Classification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1420–1430, 2021.

[24] Meike Nauta, Ron Van Bree, and Christin Seifert. Neural Prototype Trees for Interpretable Fine-grained Image Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14933–14943, 2021.

[25] Dawid Rymarczyk, Łukasz Struski, Michał Górszczak, Koryna Lewandowska, Jacek Tabor, and Bartosz Zielinski. Interpretable Image Classification with Differentiable Prototypes Assignment. In *European Conference on Computer Vision*, pages 351–368. Springer, 2022.

[26] Yuki Ukai, Tsubasa Hirakawa, Takayoshi Yamashita, and Hironobu Fujiyoshi. This looks like it rather than that: Protoknn for similarity-based classifier. In *The Eleventh International Conference on Learning Representations*, 2023.

[27] Pascal Mettes, Mina Ghadimi Atigh, Martin Keller-Ressel, Jeffrey Gu, and Serena Yeung. Hyperbolic Deep Learning in Computer Vision: A Survey. In *International Journal of Computer Vision*, volume 132, pages 3484–3508, 2024.

[28] Octavian Ganea, Gary Becigneul, and Thomas Hofmann. Hyperbolic Entailment Cones for Learning Hierarchical Embeddings. In *International Conference on Machine Learning*, volume 80, pages 1646–1655, PMLR, 2018.

[29] Karan Desai, Maximilian Nickel, Tanmay Rajpurohit, Justin Johnson, and Ramakrishna Vedantam. Hyperbolic Image-Text Representations. In *Proceedings of the International Conference on Machine Learning*, volume 202, pages 7694–7731, PMLR, 2023.

[30] Avik Pal, Max van Spengler, Guido Maria D'Amely di Melendugno, Alessandro Flaborea, Fabio Galasso, and Pascal Mettes. Compositional Entailment Learning for Hyperbolic Vision-Language Models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[31] Valentin Khrulkov, Leyla Mirvakhabova, Evgeniya Ustinova, Ivan Oseledets, and Victor Lempitsky. Hyperbolic image embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6418–6428, 2020.

[32] Samuele Fonio, Roberto Esposito, and Marco Aldinucci. Hyperbolic Prototypical Entailment Cones for Image Classification. In *International Conference on Artificial Intelligence and Statistics*, volume 258, pages 3358–3366, 2025.

[33] Paul Berg, Léo Buecher, Björn Michele, Minh-Tan Pham, Laetitia Chapel, and Nicolas Courty. Multi-prototype Hyperbolic Learning Guided by Class Hierarchy. In *International Journal of Computer Vision*, volume 133, pages 7969–7984, 2025.

[34] Hooman Vaseli, Victoria Wu, Nima Kondori, Nguyen Nhat Minh To, Andrea Fung, Ang Nan Gu, and Purang Abolmaesumi. HAPPI: Hyperbolic Hierarchical Part Prototypes for Image Recognition. In *Beyond Euclidean Workshop at International Conference on Computer Vision*, 2025.

[35] Hooman Vaseli, Victoria Wu, Diane Kim, Michael Y. Tsang, Ang Nan Gu, Christina Luong, Purang Abolmaesumi, and Teresa S. M. Tsang. HiProtoNet: Hyperbolic Hierarchy-aware Part Prototypes for Aortic Stenosis Severity Classification. In *ASMUS Workshop at International Conference on Medical Image Computing and Computer Assisted Intervention*, 2025.

[36] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep ViT Features as Dense Visual Descriptors. In *European Conference on Computer Vision Workshop What is Motion For?*, 2022.

[37] Wei-Chih Hung, Varun Jampani, Sifei Liu, Pavlo Molchanov, Ming-Hsuan Yang, and Jan Kautz. Scops: Self-supervised co-part segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[38] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016

[39] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017.

[40] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.

[41] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 32–42, 2021.

[42] Kingma Diederik and Ba Jimmy. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 2015.

[43] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: a Large-Scale Hierarchical Image Database In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

A Additional Evaluations

A.1 Using DINOv2 as a Backbone

The proposed framework leverages the feature extraction capability of DINO-ViT to assign pseudo patch IDs, resulting in improving the backbone-agnostic classification accuracy. Therefore, it is also applicable to other methods using DINO-ViT. Table 5 compares its performance with that of ProtoPNet [7] and the method from Zhu et al. [22] in a single-run setting. Because the method from Zhu et al. [22] trains class-specific prototypes in a non parametric manner, directly applying the proposed framework by introducing shared prototypes, projecting features into a hyperbolic space, and adding our loss terms would be inappropriate. Accordingly, we applied the proposed framework after switching the prototype learning mechanism to a parametric one as is standard ProtoPNet, while keeping the original model architecture unchanged. The results showed that the proposed framework achieved a higher classification accuracy.

Table 5: Classification accuracy of ProtoPNet [7], Zhu et al.'s method[22], and the proposed framework using DINOv2 backbones.

Backbone	Method	Accuracy
DINOv2 ViT-B/14 [17]	ProtoPNet [7]	85.45
	Zhu et al. [22]	89.66
	Ours	**90.13**
DINOv2 ViT-S/14 [17]	ProtoPNet [7]	82.78
	Zhu et al. [22]	85.21
	Ours	**87.30**

A.2 Sensitivity to PCA Dimensions when assigning Pseudo Patch IDs

The pseudo patch IDs of the proposed framework are obtained by clustering DINOv2 feature vectors after dimensionality reduction with PCA. Table 6 presents the relationship between the number of retained PCA dimensions and accuracy in a single-run setting. along with the classification accuracy obtained by assigning IDs using the unsupervised part segmentation method of Amir et al. [36]. The classification accuracy was highest when the first five principal components were used, and using a larger number of dimensions degraded the classification accuracy. The pseudo patch IDs do not require detailed information on the parts and features necessary for fine-grained classification but should capture features of common parts shared across classes. Thus, a smaller number of dimensions is more suitable.

Table 6: PCA dimensions vs. Accuracy.

PCA dim.	3	5	16	28	768 (full)	Amir et al. [36]
Cumulative contribution	15.89	21.91	39.67	50.02	100.00	-
Accuracy	78.24	**78.32**	78.20	77.01	76.99	77.93

A.3 Extended Visualization of Prototypes

Figure 5 and Figure 6 demonstrate the visualized trained prototypes with $K = 5$ for CUB200-2011 and Stanford Cars as extended results of Figure 4. Shared and class-specific prototypes with the same prototype index k captured the same part regions across broader classes, while prototypes with different indices k captured different part regions. Additionally, Figure 7 and Figure 8 show the visualizations for $K = 10$. Some prototypes capture very similar regions between different k (e.g., $k = 2$ and $k = 5$ for CUB200-2011), and some prototypes capture irrelevant regions (e.g., $k = 6$ for Stanford Cars). These results indicate that a larger K may harm prototype quality from the perspective of interpretability.

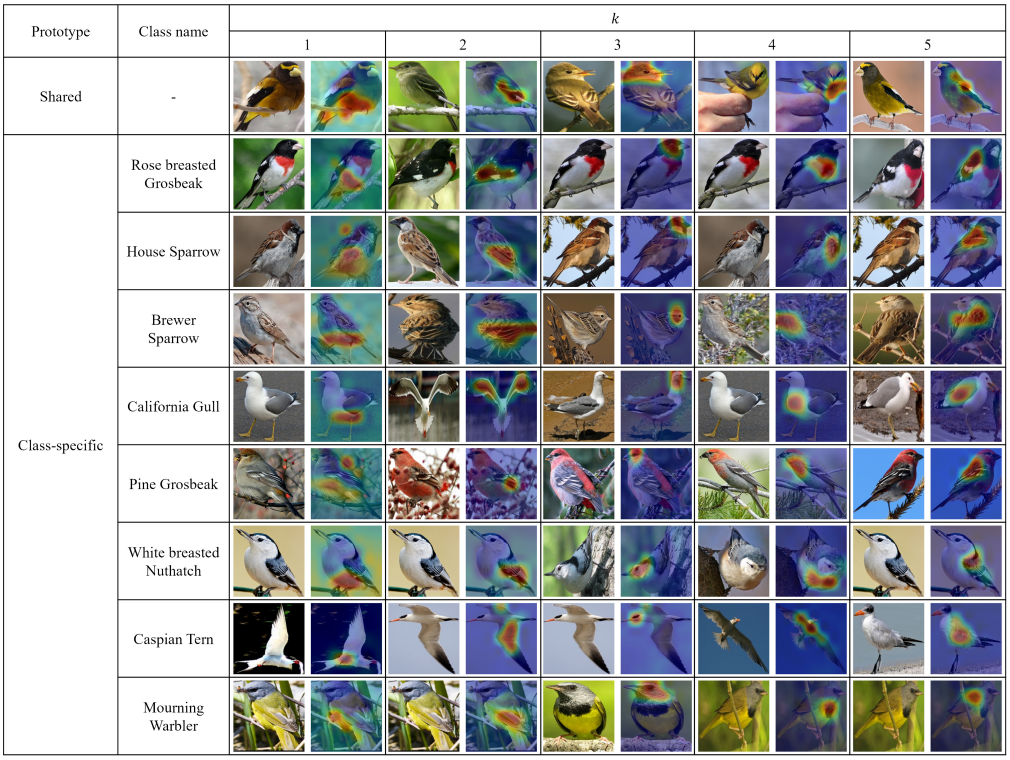


Figure 5: Extended visualization of prototypes for $K = 5$ from CUB200-2011 [15].



Figure 6: Extended visualization of prototypes for $K = 5$ from Stanford Cars [16].

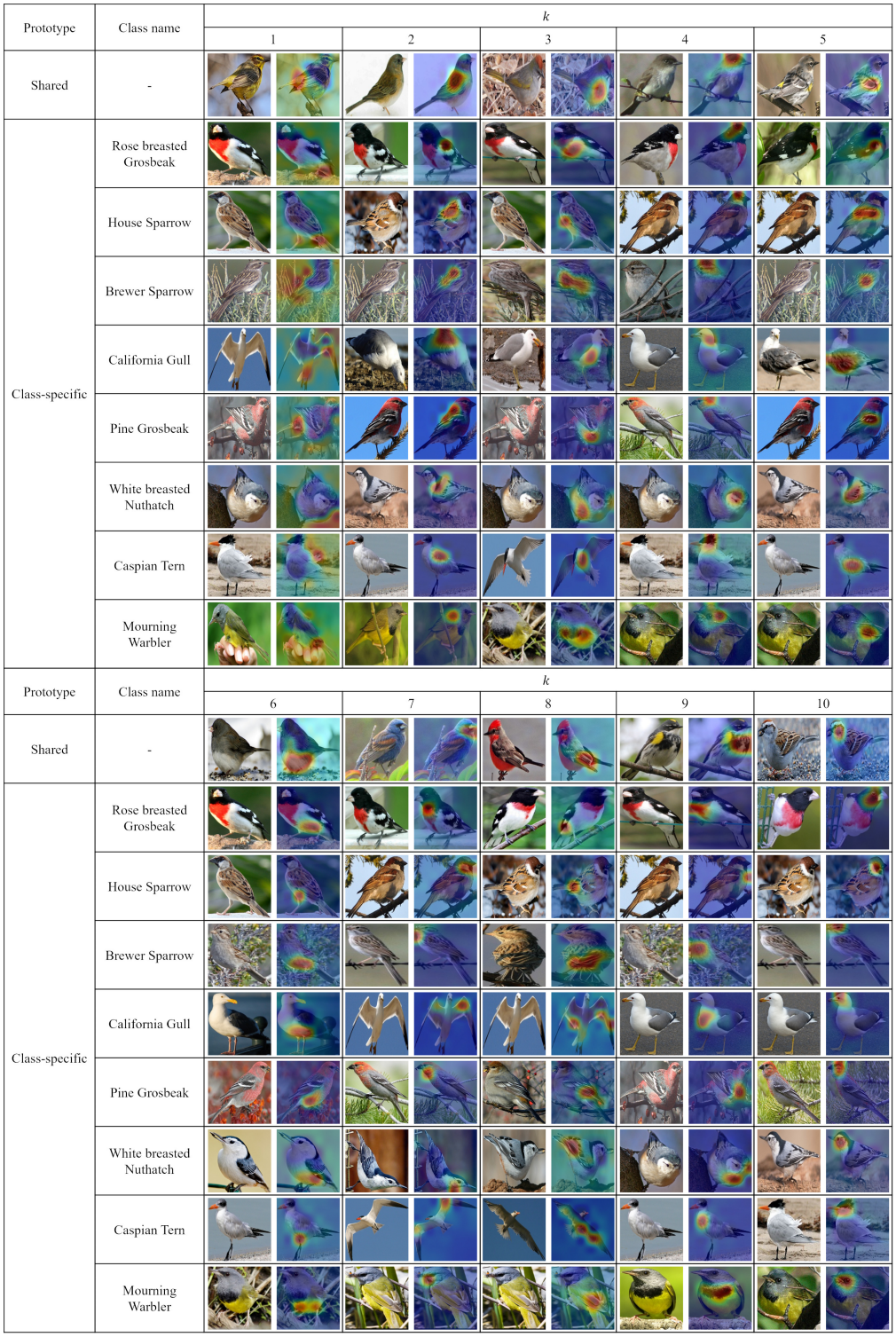


Figure 7: Extended visualization of prototypes for $K = 10$ from CUB200-2011 [15].

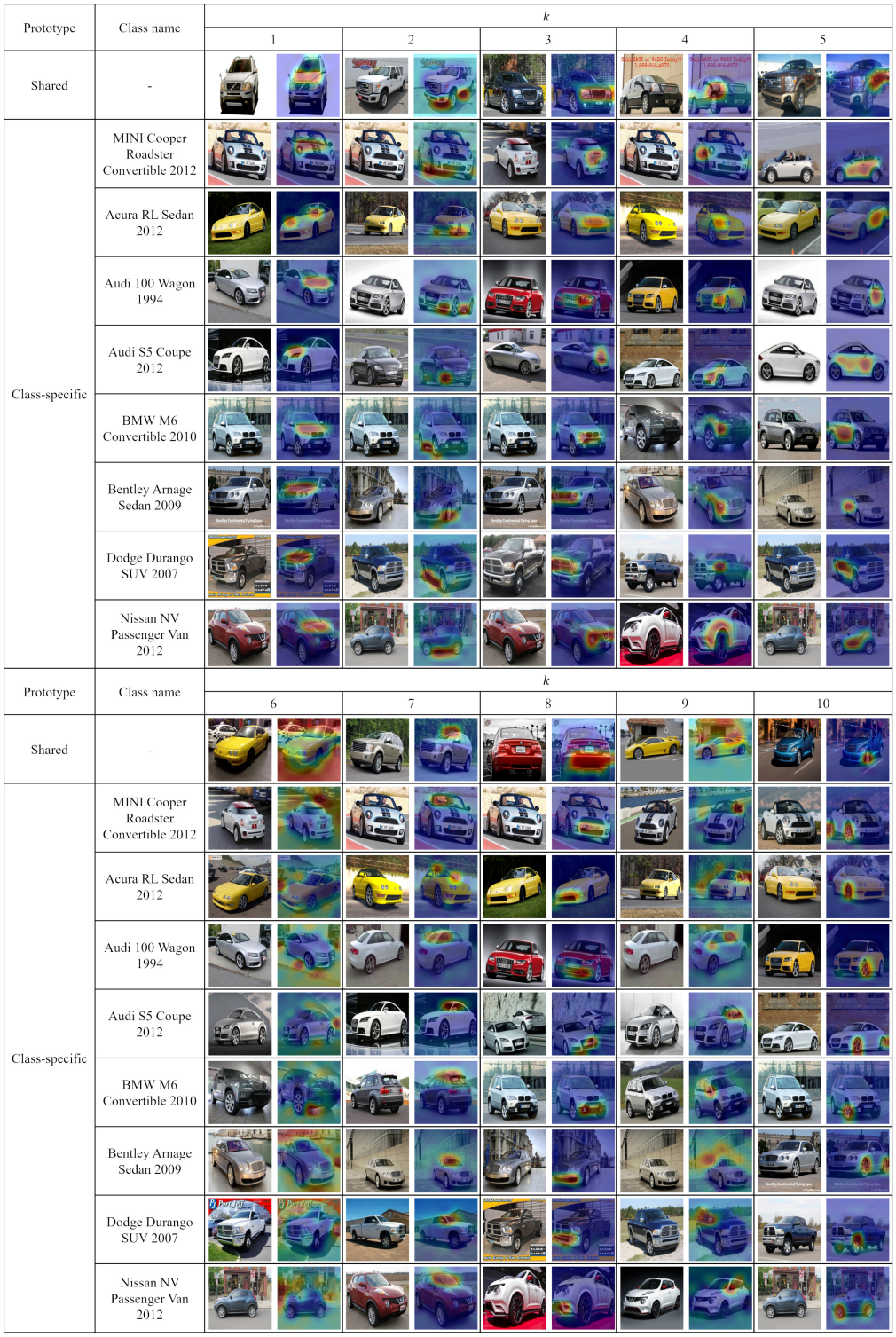


Figure 8: Extended visualization of prototypes for $K = 10$ from Stanford Cars [16].

A.4 Sensitivity to K

In the main paper, the experiments were conducted with the setting the number of class-specific prototypes per class to $K = 5$. Table 7 reports the classification accuracy for $K = 10$ of ProtoPNet [7], ProtoPFormer [13], and HAPPI [34] when applied to CUB200-2011 [15] and Stanford Cars [16] with and without the proposed framework. We used ResNet50 [38] and DeiT-Small [40] as the backbones. These results were obtained from a single run, not averaged over three trials. We confirm that the proposed framework also improved classification accuracy in the different K setting.

Table 7: Classification accuracy with/without the proposed framework for different baseline methods and backbones applied to CUB200-2011 [15] and Stanford Cars [16] for $K = 10$.

Base Method	Backbone	Ours	CUB200-2011[15]	Stanford Cars [16]
ProtoPNet [7]	ResNet50 [38]		75.32	83.99
		✓	**77.96**	**89.22**
ProtoPFormer [13]	DeiT-Small [40]		81.46	80.30
		✓	**83.43**	**85.74**
HAPPI [34]	ResNet50 [38]		71.45	82.59
		✓	**79.81**	**88.86**
	DeiT-Small [40]		81.67	82.74
		✓	**82.69**	**85.77**

We further conducted a sensitivity analysis of K on CUB200-2011 and Stanford Cars using ResNet50. As shown in Table 8, classification accuracy remained relatively stable across the evaluated values of K in a single-run setting, whereas S_{part} decreased by more than 10 points on both datasets. Importantly, from the viewpoint of interpretability for humans together with the visualization results in Appendix A.3, a large K is undesirable: since our method encourages the foreground region to be organized into K distinct pseudo-part groups, too many parts yield an over-fine decomposition that is hard to interpret (e.g., splitting a bird's wing or head into several pieces blurs the semantic distinction). A small $K(= 5)$ thus balances accuracy, spatial part hierarchy, and human interpretability.

Table 8: Sensitivity analysis on K against classification accuracy and part hierarchy score S_{part}.

CUB200-2011 [15]	K	3	5	7	10	12	15
	Prototypes / Class	600	1,000	1,400	2,000	2,400	3,000
	Accuracy	78.01	78.32	**78.41**	77.96	78.39	77.93
	S_{part}	**78.07**	75.36	73.67	69.97	71.05	67.24
Stanford Cars [16]	K	3	5	7	10	12	15
	Prototypes / Class	588	980	1,372	1,960	2,352	2,940
	Accuracy	88.06	89.26	89.44	89.22	89.33	**89.47**
	S_{part}	**69.61**	62.88	62.01	62.87	63.54	57.13

A.5 Evaluation of Interpretability Scores on ProtoPNet and EvalProtoPNet

As an extension of Table3, we evaluated interpretability metrics for both ProtoPNet[7] and EvalProtoPNet[10] using ResNet50[38] and DenseNet161[39]. As shown in Table9, our method improves the accuracy and all four interpretability metrics overall. The only exception is Consistency on EvalProtoPNet with DenseNet161, although the baseline exhibits a large standard deviation.

Table 9: Metrics used to evaluate the interpretability of prototypes obtained with and without applying the proposed framework to ProtoPNet [7] and EvalProtoPNet [10] with ResNet50 [38] and DenseNet161 [39] on CUB200-2011[15].

Base Model : ProtoPNet [7]					
Backbone : ResNet50 [38]					
Ours	Accuracy	Consistency	Stability	Distinctiveness	Comprehensiveness
	74.39±0.19	41.10±4.04	49.83±1.38	10.15±1.18	19.65±0.17
✓	**78.67±0.78**	**95.07±0.25**	**54.68±0.37**	**91.33±0.13**	**37.06±0.16**
Backbone : DenseNet161 [39]					
Ours	Accuracy	Consistency	Stability	Distinctiveness	Comprehensiveness
	73.85±0.09	82.20±1.47	38.49±2.10	32.73±4.58	30.80±0.30
✓	**84.15±0.17**	**90.83±0.17**	**66.25±0.23**	**89.20±0.08**	**40.24±0.10**

Base Model : EvalProtoPNet [10]					
Backbone : ResNet50 [38]					
Ours	Accuracy	Consistency	Stability	Distinctiveness	Comprehensiveness
	81.22 ± 0.19	68.87 ± 2.19	53.38 ± 0.51	52.40 ± 2.96	34.58 ± 0.11
✓	**82.38 ± 0.19**	**85.00 ± 0.49**	**59.06 ± 0.40**	**79.16 ± 0.37**	**43.80 ± 0.02**
Backbone : DenseNet161 [39]					
Ours	Accuracy	Consistency	Stability	Distinctiveness	Comprehensiveness
	83.52 ± 0.68	**86.37 ± 4.13**	64.15 ± 3.03	61.07 ± 1.50	39.58 ± 0.52
✓	**84.65 ± 0.48**	84.83 ± 0.46	**68.16 ± 0.36**	**80.39 ± 0.17**	**43.99 ± 0.03**

A.6 Evaluations of Part Hierarchy on HAPPI and the Proposed Method on multiple base methods

As an extention of Table 2, we evaluated the part hierarchy score S_{part} introduced in Section 4.2 on HAPPI [34] and the proposed framework on ProtoPNet [7], ProtoConcepts [9], and HAPPI [34]. Table 10 shows the average and standard deviation across three trials, and indicates that the proposed framework stably achieves higher hierarchy score than HAPPI without our framework does. For the HAPPI with our framework, the improvement is smaller than for base methods without a hyperbolic entailment of their own, such as ProtoPNet and ProtoConcepts. This is expected by design. HAPPI already employs an entailment mechanism of a different kind from ours: it encourages each specific prototype to be contained in the cone of at least one generic prototype of its class. These specific prototypes coincide with the class-specific prototypes of our proposed hierarchy. On these, we additionally impose our own constraint: each must be contained within the cone of its corresponding shared prototype, paired in advance to form the part hierarchy. The same prototypes are therefore subject to two entailment constraints at the same time. We believe this competition between the two constraints leads to the smaller gain on HAPPI.

Table 10: Extended evaluation of the part hierarchy based on the average IoU between regions with high similarity among prototypes sharing an entailed relationship.

Backbone	Method	CUB200-2011 [15]	Stanford Cars [16]
ResNet50 [38]	HAPPI [34]	9.65 ± 0.27	9.12 ± 0.07
	ProtoPNet [7] + Ours	**74.77 ± 0.78**	**62.29 ± 0.48**
	ProtoConcepts [9] + Ours	**55.99 ± 7.93**	**57.17 ± 3.59**
	HAPPI [34] + Ours	**23.72 ± 3.64**	**10.00 ± 10.29**
DenseNet161 [39]	HAPPI [34]	10.94 ± 0.37	7.76 ± 0.21
	ProtoPNet [7] + Ours	**69.42 ± 1.91**	**58.39 ± 1.77**
	ProtoConcepts [9] + Ours	**66.06 ± 0.85**	**59.81 ± 2.97**
	HAPPI [34] + Ours	**18.80 ± 1.77**	**10.29 ± 0.82**

B Implementation Details

We use the standard train/test splits of CUB-200-2011 [15] and Stanford Cars [16], following the base methods; we did not carve out a separate validation split. Our tunable hyperparameters are limited, as detailed below.

Loss coefficients. The loss coefficients for contrastive learning ($L_{\text{ctr}}^{\text{Class}}$ and $L_{\text{ctr}}^{\text{Shared}}$) were tuned once and then fixed on a single reference configuration (ProtoPNet + ResNet50 on CUB), and reused without any re-tuning for every other backbone, ProtoPNet variant, and dataset. We performed no per-backbone, per-dataset, or per-variant search; the fact that one fixed setting transfers across all backbones, variants, and both datasets and still yields consistent gains indicates the improvements are not the product of benchmark-specific tuning.

Settings inherited from prior work. Following each base method's released code, prototype projection is used for ProtoPNet [7], ProtoConcepts [9], and HAPPI [34], but not for the others. The learning rate for the shared prototypes R is set equal to that for the class-specific prototypes P, which are already present in the base ProtoPNet-style method; and the entailment weight λ_{entail}, the curvature κ, and the half-aperture angle Q follow HAPPI [34].

Method-specific implementation. The learning rate decays by $\times$ 0.5 every 5 epochs for ProtoPNet and ProtoConcepts. For ProtoPFormer we disable L_{PPC}, a loss that enforces the concentration and mutual separation of prototype activation regions (also disabled in HAPPI, and redundant with our contrastive loss). For EvalProtoPNet, we adapt the Shallow-Deep Feature Alignment (SDFA) loss proposed in that method, replacing the Euclidean distance used to align the shallow and deep feature vectors with the hyperbolic (Lorentzian) distance, consistent with our prototypes being learned in hyperbolic space; the Euclidean orthogonal loss L_{orth} is disabled.

A minor auxiliary regularizer. We also specify a minor auxiliary regularizer used throughout our experiments: a separation loss (auxiliary to entailment), based on the hyperbolic similarity θ^{hyp} (Eq. 8), that regularizes the geometric arrangement of prototypes:

$$L_{\text{sep}}^{\text{hyp}} = \frac{1}{K(K-1)/2} \sum_{k \neq k'} \theta^{\text{hyp}}(\tilde{\mathrm{R}}_k, \tilde{\mathrm{R}}_{k'}) + \frac{1}{CK} \sum_c \sum_k (\tilde{\mathrm{R}}_k, \tilde{\mathrm{P}}_k^c). \tag{17}$$

We set $\lambda_{\text{sep}}^{\text{hyp}} = 0.1$ as the coefficient for this term. It is intended to help construct the entailment relation $\text{ext}(\tilde{\mathrm{R}}_k, \tilde{\mathrm{P}}_k^c) < \text{aper}(\tilde{\mathrm{R}}_k)$ in the feature space. The first term prevents distinct shared prototypes from collapsing near the origin, so that each can serve as the apex of a well-separated entailment cone; the second increases the separation between each shared prototype and its corresponding class-specific one, which, since the entailment cone widens away from its apex, places the class-specific prototype in the wider region of the cone and thereby stabilizes the satisfaction of the entailment. It is not intended to assign part-level semantics to prototypes; assigning part-level semantics is instead the role of the patch-level contrastive loss (Section 3.4). It was enabled identically in all configurations that use the hyperbolic embedding, i.e., every ablation row of Table 4 except the Euclidean baseline, so it does not affect the relative comparisons among the entailment/contrastive ablations, and its effect on accuracy is marginal.

Overall, these are clarifications rather than architectural changes, so the "base method + plug-in module" characterization still holds.

C Metrics for Evaluating Prototype Interpretability

We conducted a evaluation of prototypes with four quantitative metrics in Section 4.3: the consistency, the stability, the distinctiveness, and the comprehensiveness. The detailed definitions of them are as following equations: The consistency and stability required part-level keypoint annotations for each image, and comprehensiveness required object-level segmentation labels. Therefore, the experiments were conducted by using the CUB-200-2011 [15] because it included these annotations. Note that while the original paper's implementation for evaluating the consistency and stability filters the high similarity region using a fixed-size bounding box centered on the pixel with maximum similarity for each image and prototype, we do not filter them for proper evaluations in this paper.

Consistency. Huang et al. [10] proposed the consistency score to evaluate how high similarity area match across images for each prototypes as follows.

$$S_{\text{con}} = \frac{1}{J}\sum_{j=1}^{J} \mathbb{1}\{\max a^{p_j} \geq \mu\}, \;\; a^{p_j} = \frac{\sum_{i\in\mathcal{D}^{p_j}} o^{p_j}(x_i)}{\|\mathcal{D}^{p_j}\|}, \tag{18}$$

where, $o^{p_j}(x_i)$ is a indicator vector that $o^{p_j}(x_i)_l = 1$ if the high similarity area of a prototype p_j for input image x_i includes part l, $o^{p_j}(x_i)_l = 0$ for others, and $\mathcal{D}^{p_j}$ is a set of input image indices such that p_j belongs to y_i.

Stability. Huang et al. [10] proposed the stability score to evaluate how high similarity area is robust against perturbation ξ to input images as follows.

$$S_{\text{sta}} = \frac{1}{J}\sum_{j=1}^{J} \frac{\sum_{x\in\mathcal{D}^{p_j}} \mathbb{1}\{o^{p_j}(x_i) = o^{p_j}(x_i+\xi)\}}{\|\mathcal{D}^{p_j}\|} \tag{19}$$

Distinctiveness. Zhu et al. [22] proposed the distinctiveness score, defined as follows, to evaluate overlap of high similarity regions between prototypes in each image.

$$S_{\text{dist}} = 1 - \frac{1}{|\mathcal{D}|}\sum_{i\in\mathcal{D}} \frac{1}{K(K-1)/2} \sum_{k=1}^{K-1}\sum_{k'=k+1}^{K} \text{IoU}(\hat{M}_i^{\mathbf{P}_k^{y_i}}, \hat{M}_i^{\mathbf{P}_{k'}^{y_i}}). \tag{20}$$

High distinctiveness means prototypes capture separate parts each other.

Comprehensiveness. Zhu et al. [22] proposed the comprehensiveness score to evaluate whether prototypes activate meaningful features.

$$S_{\text{comp}} = \frac{1}{|\mathcal{D}|}\sum_{i\in\mathcal{D}} \text{IoU}(M_i^{\text{FG}}, \bigcup_{k=1}^{K} \hat{M}_i^{\mathbf{P}_k^{y_i}}), \tag{21}$$

where M_i^{FG} is the ground truth foreground mask of x_i. This metric degrades when prototypes activate on irrelevant features such as background.

D Computational Information

We implemented our experiments using Pytorch and run on 1 NVIDIA H100 GPU or 1 A100 GPU. By applying our method to a baseline model, the training time increases by 1.47 times. It takes 67 minutes to assign pseudo patch IDs for each image in the entire training data. Note that the standard setting of ProtoPNet and its variants generates augmented training image files in advance with 30 different patterns; since we assign an ID to every augmented instance across all patterns, this increases the total number of items to process and thus leads to a longer processing time.

E Code References

We referenced the following released codes for baseline experiments and applying the proposed framework.

- ProtoPNet [7]: https://github.com/cfchen-duke/ProtoPNet
- ProtoConcepts [9]: https://github.com/Henrymachiyu/This-looks-like-those_ProtoConcepts
- ProtoPFormer [13]: https://github.com/zju-vipa/ProtoPFormer
- EvalProtoPNet [10]: https://github.com/hqhQAQ/EvalProtoPNet
- Zhu et al. [22]: https://github.com/zijizhu/proto-non-param
- HAPPI [34]: https://github.com/DeepRCL/HAPPI
  - Note : There is no code for training, loss, and model architecture at the time of this paper's submission.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The Abstract and the Introduction state the issues with the existing methods and the contributions of this paper. Section 3 describes the proposed framework and Section 4 demonstrates the effectiveness of the proposed framework.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: In the Limitations section of the main paper, we discuss the limitations of the proposed framework.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A].

Justification: There is no theoretical result and proof involved.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes].

   Justification: In the Experiments section, we describe the baseline methods, the settings for applying the proposed framework, the datasets, the implementations, and the evaluation metrics.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No].

Justification: We used open dataset for evaluations in this paper. We cannot release the codes and trained weights because of the policies of the collaborating company.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification:The paper describes the models, datasets, hyperparameters, and optimizers used in the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes].

   Justification: We conducted every experiments in the main paper with multiple trials for each condition, and reported mean and standard deviation.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: We provide information for the computational conditions in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: We confirmed that our proposed framework and the experiments described in our paper do not violate the ethical guidelines of this conference.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes].

    Justification: In our paper, we address the challenge of interpretability, a key issue when humans use deep learning technologies, propose a new technique that contributes to this field, and discuss the limitations of the proposed framework.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes].

    Justification: We properly reference papers, datasets, and released codes.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A].

    Justification: We do not release any datasets, codes, or trained weights in the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: The paper neither involves crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: The paper neither involves crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A].

    Justification: Our proposed framework and the contributions of the paper are not based on LLMs.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.