PINet: Improving the Stability of Prototype Networks via Phantasia-Inspired Uncertain Representations

Ho Kyung Shin[1], Soeun Bae[1], Sang Min Kim[1], Byoung Chul Ko[2], Woo-Jeoung Nam[1]*,

[1]School of Computer Science and Engineering, Kyungpook National University, Daegu, Republic of Korea
[2]Department of Computer Engineering, Keimyung University, Daegu, South Korea
{tlsghrud, qothdms5417, klasskoch}@knu.ac.kr[1], niceko@kmu.ac.kr[2], nwj0612@knu.ac.kr[1]

Abstract

Self-interpretable models are increasingly valued for their inherent explainability. Among them, part-prototype networks stand out by mimicking human reasoning through the use of learned prototypes. However, their explanations often lack stability, becoming sensitive to subtle input perturbations. In this work, we propose Prototype in Imagery Network (PINet), a framework that improves the stability of prototype-based explanations. Rather than training on all possible input variations, which is computationally infeasible, PINet draws inspiration from visual mental imagery. Specifically, we incorporate empty inputs and apply coarse location guidance to simulate the human ability to imagine rough object features (a process akin to Phantasia). PINet mimics this process by incorporating empty inputs and applying coarse location guidance. These imagined, or uncertain, representations are contrasted with those derived from actual inputs (certain representations). We model the differences between the two by computing similarity at both the feature and prototype levels, allowing uncertainty to be explicitly encoded during prototype learning. Comprehensive evaluations on CUB-200-2011 and Stanford Cars demonstrate that PINet consistently achieves robust accuracy and localization, even under noisy conditions. These results represent the ability of PINet to produce stable and interpretable explanations under uncertainty.

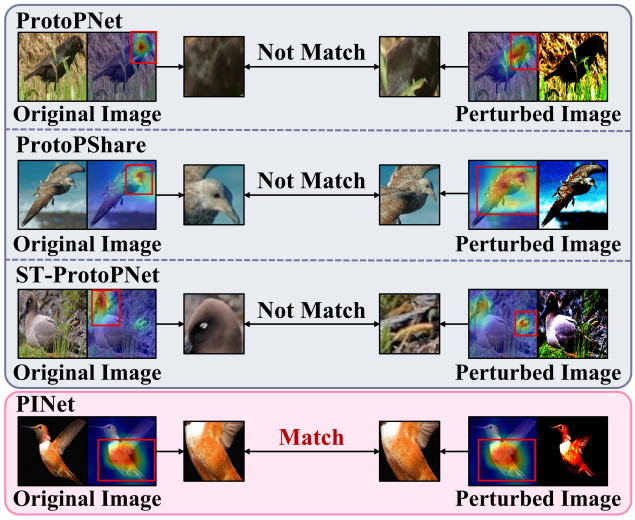

Figure 1: Comparison of the regions activated by the same prototype for original and noise-added inputs. PINet consistently maintains activations within the same features.

Introduction

Deep Neural Networks (DNNs) have demonstrated remarkable performance in diverse domains, including medicine and finance, fields closely associated with safety and sensitive data. This success has amplified the demand for interpretability to ensure faithful decision-making. As a result, Explainable AI (XAI) has emerged as a promising research area, with studies advancing distinct conceptual approaches. A common way to categorize XAI methods is based on when they provide explanations, distinguishing between post-hoc approaches and self-interpretable models. Post-hoc methods interpret the decision-making process after predictions have been made. In contrast, self-interpretable models embed interpretability directly into the network, enabling inherent transparency and explainability. Although post-hoc methods offer considerable flexibility, they have been criticized for inconsistency and lack of faithfulness (Slack et al. 2020; Carmichael and Scheirer 2021). As a result, there is growing interest in self-interpretable models.

Within self-interpretable models, part-prototype networks have gained attention for their human reasoning-inspired mechanisms. Prototypical Part Network (ProtoPNet) (Chen et al. 2019) simulates part-based recognition by comparing image parts to learned prototypes, and has been extended to various domains (Zhang et al. 2022; Wang et al. 2023b). Despite these advances, several limitations persist, including the trade-off between accuracy and interpretability, prototype duplication across classes, and instability. Although various studies (Wang et al. 2021; Rymarczyk et al. 2022; Huang et al. 2023) have proposed methods to address these challenges, relatively few have focused on mitigating instability. Instability occurs when even small input perturbations cause prototypes to generate inconsistent explanations, as illustrated in Fig.1, thus undermining interpretability. As considering every possible perturbed case is infeasible, we

*Corresponding author

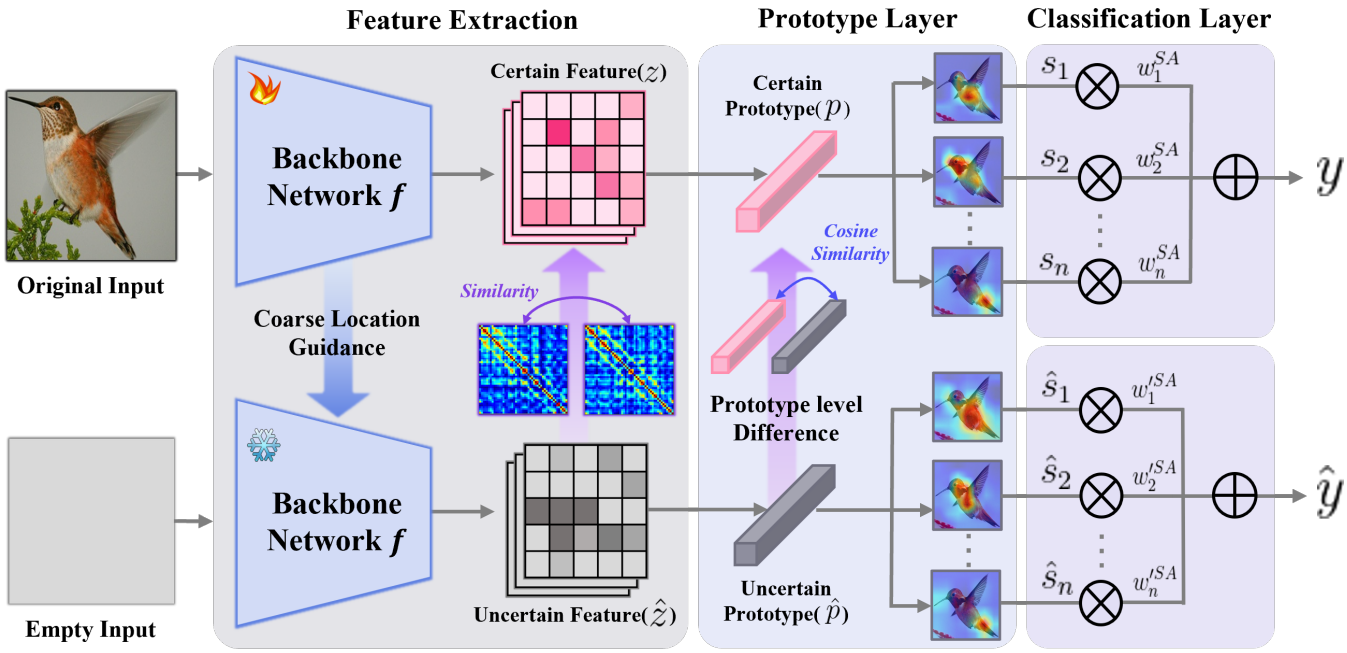

Figure 2: Overview of PINet. The conventional feature map and prototype are designed to incorporate the differences from the rough feature map and uncertain prototype obtained from empty input. The difference at the feature map level is computed using self-similarity, whereas at the prototype level, cosine similarity is employed.

propose Prototype in Imagery Network (PINet), a framework inspired by visual mental imagery, known as Phantasia (Larner, Leff, and Nachev 2024). Phantasia refers to the ability to generate rough visual images in the mind, allowing individuals to internally perceive approximate objects without external stimuli. PINet mimics this phenomenon during feature extraction by providing an empty input together with the original input. To prevent diminished activations from propagating the empty input alone, we employ coarse location guidance to indicate the approximate location of the activation in the original input. Rough features from the empty input, regraded as incomplete representations, are assigned to a prototype as uncertain representations reflecting random noise. Our goal is to integrate such uncertainty into the features and prototype derived from the original input, referred to as certain representations. To this end, we compute the differences between the representations and incorporate these differences into the framework. To validate the effectiveness of our approach, we conduct extensive experiments on CUB-200-2011 (Wah et al. 2011) and Stanford Cars (Krause et al. 2013). The results demonstrate that our approach achieves superior performance in accuracy and localization for original and perturbed inputs. In summary, our contributions are as follows:

- We propose the Prototype in Imagery Network (PINet), a framework motivated by Phantasia. PINet simulates this phenomenon by incorporating an additional empty input and coarse location guidance in feature extraction.
- We improve the stability of prototypes by treating uncertain representations as additional references and incorporating their differences from certain representations, thereby ensuring reliability in the presence of noise.
- We demonstrate through extensive experiments that our method achieves superior performance in accuracy and localization, consistently outperforming existing approaches under both original and perturbed inputs.

Related Work

Methods for interpreting the decision-making processes of DNNs are broadly divided into two groups based on when explanations are provided: post-hoc approaches and self-interpretable models. Post-hoc methods (Selvaraju et al. 2017; Nam, Choi, and Lee 2021; Nam and Lee 2024) explain pre-trained networks by employing additional explainers to reveal distinctive patterns hidden within the black-box model. However, these methods often yield unfaithful and unreliable explanations. Therefore, self-interpretable models, which inherently incorporate explainability into training mechanisms or architectures, have attracted attention as promising alternatives. In this study, we focus on part-prototype networks that reflect human behavior.

Part-Prototype Networks ProtoPNet (Chen et al. 2019) first introduced the concept of prototypes and the principle of "this looks like that", mirroring common human reasoning. Building on this foundation, various models have been proposed to address the underlying limitations. To improve both accuracy and prototype quality, TesNet (Wang et al. 2021) embeds prototypes on the Grassmann manifold and introduces loss terms to enforce orthogonality among pro-

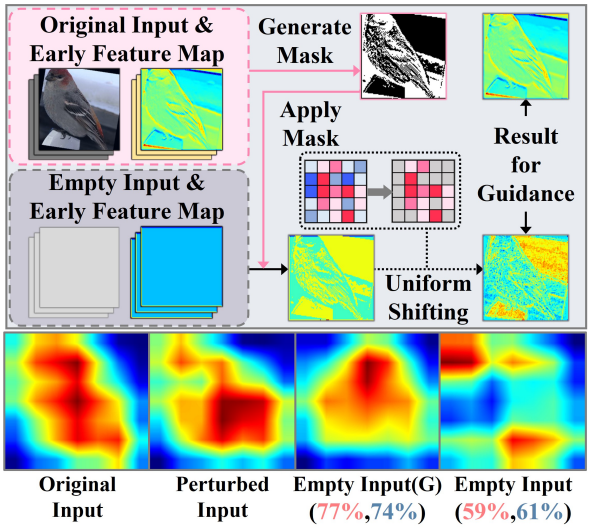

Figure 3: Procedure of coarse location guidance and feature map comparison. The feature maps below correspond to: (i) the original input, (ii) the perturbed input, (iii) an empty input with guidance, and (iv) an empty input without guidance. The values shown beneath the third and fourth maps indicate similarity scores with the first and second maps.

totypes within each class. Following this, ProtoTree(Nauta, Van Bree, and Seifert 2021) incorporates a decision tree mechanism into the prototype learning. Deformable ProtoPNet(Donnelly, Barnett, and Chen 2022) constructs prototypes in a deformable manner, enabling adaptive spatial changes. PIP-Net (Nauta et al. 2023) employs self-supervised learning to generate semantically meaningful prototypes. ST-ProtoPNet (Wang et al. 2023a) utilizes two types of prototypes that serve different roles in controlling distances from the classification boundary. PixPNet (Carmichael et al. 2024a) overcomes the limitation of focusing solely on parts by utilizing a receptive field-based spacing matching, enabling explanations that cover the entire object. ProtoFlow (Carmichael et al. 2024b) proposes a perspective by combining the concept of prototypes with generative classification. Limitations concerning prototype consistency and stability were first presented in (Huang et al. 2023), which also introduced evaluation metrics for these aspects. Lastly, Prototype redundancy has been addressed by ProtoShare (Rymarczyk et al. 2021) and ProtoPool (Rymarczyk et al. 2022), which propose pruning and assignment strategies. Despite the development of numerous methods, relatively little research has focused on improving stability, particularly robustness to noise. To address this gap, we enhance the robustness of explanations by emulating the human phenomenon of visual mental imagery.

Method

Preliminaries

Existing part-prototype networks are generally built upon the design principles of ProtoPNet, with extensions adapted to their specific approaches. This section outlines the typical components of part-prototype networks. These networks primarily comprise three parts: a backbone network f, a prototype layer g with learnable prototypes $P = \{p_n \in \mathbb{R}^{1\times1\times D}\}_{n=1}^{N}$, where D is the prototype dimension, and a classification layer h. Given an input image $x \in \mathbb{R}^{H\times W\times C}$, where H, W, and C represent the height, width, and channel size, respectively. The backbone network f extracts the feature map $z = f(x)$ of shape $\tilde{H}\times\tilde{W}\times D$. The prototype layer g produces N similarity maps, $S_n^{(i,j)} = sim(z(i,j,:), p_n)$, for $i \in \{1,2,...,\tilde{H}\}$ and $j \in \{1,2,...,\tilde{W}\}$ by computing the similarity between the feature map z and each prototype p_n. The similarity maps $S_n^{(i,j)}$ are converted to scores s_n via max pooling, which are aggregated in the classification layer using a weight matrix w_n to yield the prediction. This architecture inherently supports interpretability by associating each prototype with features, enabling the model to provide explanations.

Prototype in Imagery Network

We propose Prototype in Imagery Network (PINet), a framework designed to improve stability by modeling Phantasia. The overall architecture is depicted in Fig. 2. Structurally, PINet resembles conventional part-prototype networks. The key contribution lies in leveraging an empty input to simulate Phantasia during feature extraction, thereby generating uncertain representations. These uncertain representations are integrated with certain representations to handle incomplete features, which can be caused by inputs with noise.

Modeling Phantasia in feature extraction Phantasia refers to the human ability to generate visual mental imagery, enabling the perception of objects internally. Inspired by the observation that such imagery facilitates the recognition of rough characteristics of imagined objects, we propose a method that mimics this process within our framework. Specifically, feature extraction is performed using the original input x and an empty image $\hat{x}$ of identical shape, both processed through the backbone network f. $\hat{x}$ simulates the absence of external stimuli, and extracting features from $\hat{x}$ emulates the internal generation of visual mental imagery. However, the features extracted from $\hat{x}$ tend to capture unrelated representations due to the lack of explicit evidence of the target object. We guide f to encode rough representations from $\hat{x}$ using coarse location guidance, which provides approximate information about activated regions in x. Coarse location guidance operates in three steps: (i) generating a binary mask $\mathcal{B}_{ijk}$ from the early feature of x, (ii) retaining only the features of $\hat{x}$ aligned with $\mathcal{B}_{ijk}$, and (iii) applying uniform shifting. $\mathcal{B}_{ijk}$ is derived from the positively activated regions in the activation map $\mathcal{A}_{ijk}$ of x at the first layer, as detailed in Eq. 1.

$$\mathcal{B}_{ijk} = \begin{cases} 1 & \mathcal{A}_{ijk} > 0 \\ 0 & \mathcal{A}_{ijk} \leq 0 \end{cases} \tag{1}$$

Here, i, j and k denote spatial indices. As defined in Eq. 2, $\mathcal{B}_{ijk}$ is applied to the activation map $\hat{\mathcal{A}}_{ijk}$ of $\hat{x}$ at the same

Accuracy(Original/Perturbed Input)							
	ProtoPNet	**TesNet**	**ProtoPShare**	**ProtoPool**	**EvalProtoPNet**	**ST-ProtoPNet**	**PINet**
VGG16	76.1 / 63.7	81.3 / 68.9	71.8 / 65.4	76.3 / 68.2	80.9 / 73.1	82.9 / 73.6	**83.4 / 78.8**
VGG19	78 / 67.6	81.4 / 71.2	75.8 / 68.3	78.4 / 73.1	82.5 / 74.7	83.2 / 75.7	**83.8 / 78.9**
Res34	79.2 / 69.4	82.8 / 72.8	74.7 / 71.8	80.3 / 72.6	84 / 74.9	83.5 / 74.6	**85.1 / 80.4**
Res152	78 / 69.9	82.7 / 70.6	73.6 / 64.5	81.5 / 69.8	85.1 / 72.8	84.1 / 75.8	**85.3 / 82.3**
Den121	80.2 / 70.1	84.8 / 74.2	74.7 / 71.1	81.5 / 68.7	85.4 / 74.7	85.4 / 76.1	**86.2 / 79.5**
Den161	80.1 / 71.8	84.6 / 74.6	76.4 / 72.3	82 / 71.4	86.5 / 75.8	86.1 / 77.8	**86.7 / 80.8**
Consistency Score(Original/Perturbed Input)							
	ProtoPNet	**TesNet**	**ProtoPShare**	**ProtoPool**	**EvalProtoPNet**	**ST-ProtoPNet**	**PINet**
VGG16	17.4 / 9.3	31.8 / 20.8	14.5 / 13.3	25 / 13.8	56.7 / 33.1	38.8 / 17.7	**58.1 / 43.3**
VGG19	31.6 / 10.4	46.8 / 22.1	21.4 / 12.2	36.2 / 14.8	56.5 / 36.7	39.9 / 21.3	**59.4 / 48.9**
Res34	15.1 / 8.8	53.3 / 30.1	19 / 13.7	32.4 / 12.5	70.6 / 36.2	26.1 / 14.6	**70.9 / 52.1**
Res152	28.3 / 7.6	48.6 / 27.7	26.1 / 15.6	35.7 / 20	62.1 / 32.3	31.4 / 18.9	**62.3 / 57.4**
Den121	24.9 / 12.5	63.1 / 36.5	23.8 / 21.4	48.5 / 19.9	68.1 / 41.1	36.5 / 16.4	**75.6 / 61.3**
Den161	21.2 / 7.2	62.2 / 34.4	29.6 / 24.6	40.6 / 15.4	72 / 37.5	45.8 / 27.8	**73.5 / 59.6**
Stability Score(Original Input)							
	ProtoPNet	**TesNet**	**ProtoPShare**	**ProtoPool**	**EvalProtoPNet**	**ST-ProtoPNet**	**PINet**
VGG16	68.1	69.2	64.6	66	70.36	52.5	**75.6**
VGG19	60.4	58.2	68.8	62.7	63.5	52.7	**77.9**
Res34	53.8	65.4	70.9	57.6	72.1	45.7	**76.1**
Res152	56.7	60	59.1	58.4	70.8	48.3	**75.7**
Den121	58.9	66.1	65.3	55.3	67.6	49.5	**73.6**
Den161	58.2	67.5	69.4	61.2	71.8	59.1	**76.3**

Table 1: The results of quantitative evaluations on CUB-200-2011. For accuracy and consistency score, results for perturbed inputs are also reported. The second value in each cell indicates the metric for perturbed inputs.

Accuracy(Original/Perturbed Input)							
	ProtoPNet	**TesNet**	**ProtoPShare**	**ProtoPool**	**EvalProtoPNet**	**ST-ProtoPNet**	**PINet**
VGG16	88.3 / 80.4	90.3 / 84.7	84.1 / 76.2	84.7 /79.1	90.5 / 83	91.1 / 82.6	**91.8 / 89.1**
VGG19	89.4 / 79.7	90.6 / 84.8	88.4 / 81.5	87.8 / 81	90.6 / 83.1	**91.7 / 84.3**	**91.7 / 90**
Res34	88.8 / 80.2	90.9 / 83.7	87.6 / 82.2	89.2 / 81.9	91.1 / 85.6	91.4 / 83.6	**91.5 / 88.9**
Res152	88.5 / 81	92 / 83.8	80.1 / 71.2	90.1 / 82.1	91.8 / 84.8	92 / 86.4	**92.6 / 89.6**
Den121	87.7 / 78.9	91.9 / 84.3	85.6 / 78.4	88.2 / 83.6	92.2 / 86.4	92.3 / 85.2	**92.7 / 88.8**
Den161	89.5 / 81.1	92.6 / 83.5	88.9 / 80.3	90.3 / 82.8	92.5 / 85.5	92.7 / 85.7	**93.1 / 90.7**

Table 2: Performance of various backbone networks on original and noise-added inputs in the StanfordCars. The second value in each cell corresponds to noise-added inputs.

layer using the Hadamard product $\otimes$, preserving only values corresponding to the activated regions in x.

$$\hat{\mathcal{A}} = \hat{\mathcal{A}} \otimes \mathcal{B} \tag{2}$$

Directly applying the mask often results in residual negative values in $\hat{\mathcal{A}}_{ijk}$, which tend to diminish through propagation to deeper layers. To address this, we apply uniform shifting along the channel dimension, as shown to be effective in (Nam et al. 2020). This shifting enables negative values to be cast into positive while maintaining the distributional gap between values. After shifting, $\mathcal{B}$ is reapplied to eliminate any shifted values beyond the masked regions:

$$\hat{\mathcal{A}} = (\hat{\mathcal{A}}_k - \min(\hat{\mathcal{A}}_k)) \otimes \mathcal{B} \tag{3}$$

Fig. 3 illustrates the process of coarse location guidance and provides a visual comparison of feature maps obtained from empty input, with and without guidance. The comparison demonstrates that the guided feature map $\hat{z} = f(\hat{x})$ captures rough but partially similar features obtained from the original input, compared to those without guidance. Accordingly, we utilize $\hat{z}$ with an additional prototype to construct uncertain representations. During propagation of the empty input, the weights of f are not updated, as our intention is to introduce uncertainty only through the obtained uncertain representations.

Incorporating uncertainties In most conventional part-prototype networks, a single prototype P is used to compute similarity with the feature map z. We consider z and P as certain representations derived from x. To encode uncertainty, we define an additional prototype, the uncertain prototype $\hat{P} = \{\hat{p}_n \in \mathbb{R}^{1\times1\times D}\}_{n=1}^{N}$, which are used to com-

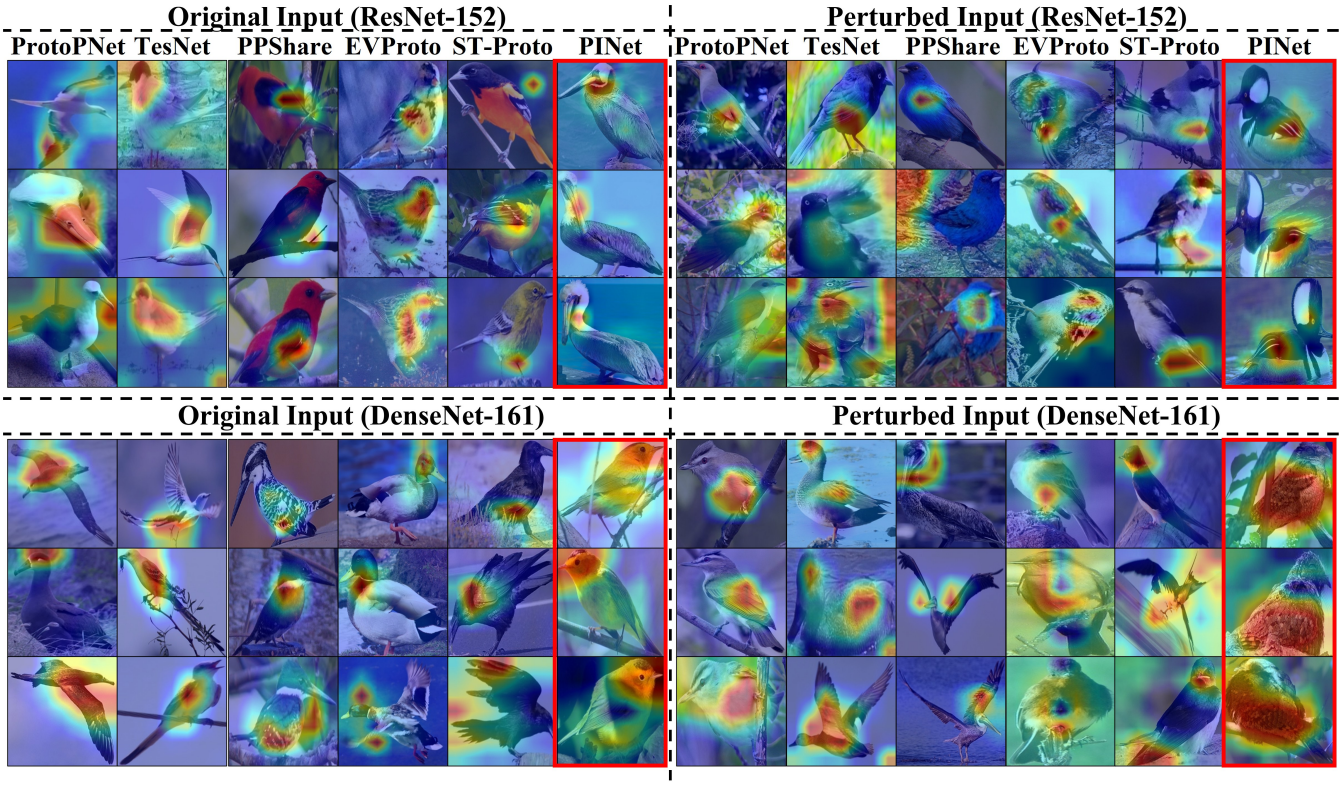

Figure 4: Qualitative evaluation results in terms of consistency for ResNet-152 and DenseNet-161. The left side shows the results for original inputs, while the right side presents the results for perturbed inputs. Each column visualizes the same prototype part for the same bird species. PPShare, EVProto and St-Proto denote ProtoPShare, EvalProtoPNet and ST-ProtoPNet.

pute similarity with $\hat{z}$. As shown in Fig. 3, $\hat{z}$ exhibits similar activation patterns to the feature map obtained from the perturbed input. Therefore, $\hat{z}$ and $\hat{P}$ together constitute the uncertain representations, serving as proxies for perturbed conditions. To integrate uncertain representations, we compute their differences from certain representations at both the feature map and prototype levels, incorporating these as loss terms. At the feature map level, we employ a self-similarity-based approach to reflect discrepancies in spatial relational patterns. First, for original input x, we identify spatial location r that exhibits the highest activation for the ground-truth class within P. After normalization along the channel dimension, self-similarity matrices are generated for z and $\hat{z}$, denoted as G and $\hat{G}$. The row corresponding to r is extracted from each matrix to characterize the relational pattern between r and all other spatial locations. The absolute difference between these rows defines the feature map discrepancy loss $\mathcal{L}_{feat}$:

$$\mathcal{L}_{feat} = |G_r - \hat{G}_r| \tag{4}$$

This quantifies the degree of change in relational patterns, measuring the magnitude of structural discrepancies. At the prototype level, the difference is measured using cosine similarity between prototypes and converted into the prototype loss $\mathcal{L}_{proto}$:

$$\mathcal{L}_{proto} = 1 - \sum cos(P, \hat{P}) \tag{5}$$

$\mathcal{L}_{proto}$ is included in the total loss $\mathcal{L}_{total}$ to encourage robustness of P under uncertainty.

For the classification layer h, we utilize the score aggregation (SA) module introduced in (Huang et al. 2023). The SA module aggregates similarity scores only within their allocated classes and assigns a learnable weight w_n^{SA} to each score. The final class score is calculated as a weighted sum of similarity scores from class-relevant prototypes. This approach enhances performance by preventing cross-class interference. Class scores for both x and $\hat{x}$ are computed using the SA module with shared weights. Cross-entropy loss is then applied to each, denoted as $\mathcal{L}_{certain}$ and $\mathcal{L}_{uncertain}$. The total loss $\mathcal{L}_{total}$ combines losses from existing methods: $\mathcal{L}_{certain}$, $\mathcal{L}_{clst}$, $\mathcal{L}_{sep}$ (Chen et al. 2019), and $\mathcal{L}_{ortho}$ (Donnelly, Barnett, and Chen 2022), with additional terms for uncertainty: $\mathcal{L}_{uncertain}$, $\mathcal{L}_{feat}$, and $\mathcal{L}_{proto}$. The contribution of these additional losses for incorporating uncertainty into the certain representation is controlled by the parameter λ. We constrain the range of λ to $0 \leq \lambda \leq 1$ and set its initial value to 0.3 during both training and evaluation. The results for additional values are presented in the supplementary material. $\mathcal{L}_{total}$ is defined as follows:

$$\begin{aligned}\mathcal{L}_{total} = \mathcal{L}_{certain} + \mathcal{L}_{clst} + \mathcal{L}_{sep} + \mathcal{L}_{ortho} \\ +\lambda(\mathcal{L}_{uncertain} + \mathcal{L}_{feat} + \mathcal{L}_{proto})\end{aligned} \tag{6}$$

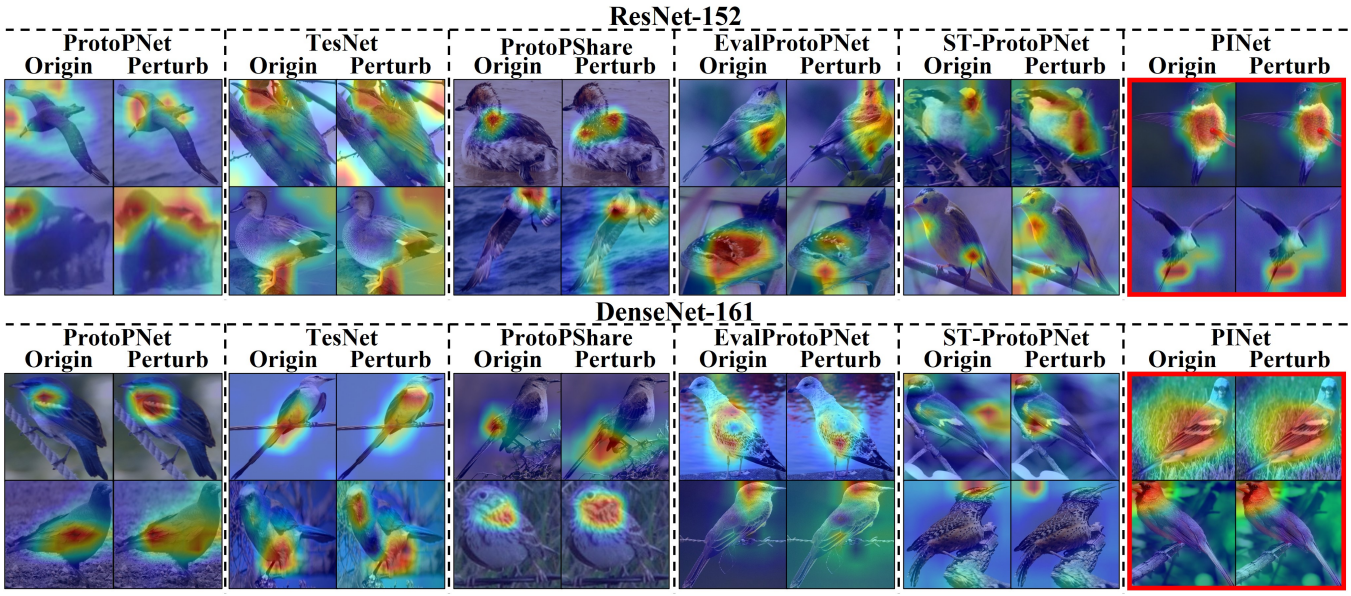

Figure 5: Qualitative results evaluating stability for ResNet-152 and DenseNet-161. The same prototype, obtained from original inputs, is applied to both original and perturbed inputs to compare changes in explanations according to the addition of noise.

Experiments

Experimental Settings

To evaluate the effectiveness of PINet, we conducted extensive experiments focusing on accuracy and interpretability. We assessed the performance across various backbone networks f, including VGG(VGG16, VGG19) (Simonyan and Zisserman 2014), ResNet(Res34, Res50) (He et al. 2016), DenseNet(Den121, Den161) (Huang et al. 2017), using two widely used datasets: CUB-200-2011 and Stanford Cars. For comparison, we selected representative part-prototype networks, including ProtoPNet, TesNet, ProtoPShare, ProtoPool, EvalProtoPNet (Huang et al. 2023), and ST-ProtoPNet.

Quantitative Results

CUB-200-2011 Following previous works, we evaluated our frameworks using three metrics: accuracy, consistency score, and stability score (Huang et al. 2023). Each metric was assessed under standard evaluation scenarios. In addition, accuracy and consistency score were further evaluated under the perturbed scenario. The perturbed scenario was implemented by adding random noise to the input, where the noise is sampled from various Gaussian distributions with different σ. Results for each metric are summarized in Tab. 1, which reports the outcomes with noise sampled at $\sigma = 0.2$. Results for higher noise levels($\sigma > 0.2$) are provided in the supplementary material.

As shown in the upper part of Tab. 1, PINet outperforms all comparative models in accuracy, both with original and noise-added inputs. ProtoPShare and ProtoPool, designed to address the prototype redundancy, show relatively lower performance compared to the other models. Notably, our framework demonstrates consistent superiority on original inputs, with a marked improvement of 1.6% on ResNet-34. This performance gap widens under perturbed conditions. PINet achieves nearly 80% accuracy in this setting, whereas models that perform competitively with PINet on the original input exhibit declines of at least 3.1%.

The middle section of Tab. 1 presents the consistency score, which measures how consistently a prototype is mapped to the same part across different images. PINet achieves the highest consistency score among all models, followed by EvalProtoPNet, which originally proposed this evaluation metric. Under perturbed inputs, PINet maintains the highest score, even surpassing the performance of several models evaluated on original inputs. Comparing the performance change across scenarios, ProtoPNet and ST-ProtoPNet exhibit a significant drop, whereas our framework remains relatively stable.

The last section of Tab. 1 reports the stability score, which quantifies robustness to noise. PINet also demonstrates superior performance, achieving an especially high score of 77.9 with VGG-19. ST-ProtoPNet, which employs two prototypes with different roles, exhibits the lowest performance across most backbone networks. This indicates that the assignment of roles to prototypes can significantly affect performance. These results demonstrate that our uncertain prototypes and representations enhance robustness to noise by treating them as imperfect proxies for perturbed inputs.

Stanford Cars Unlike CUB-200-2011, Stanford Cars does not provide bounding box annotations for each object part, which are necessary for calculating consistency and stability scores. Thus, only accuracy was quantitatively evaluated for this dataset, as summarized in Tab. 2. PINet demonstrates superior performance on Stanford Cars as well. Most models, excluding ProtoPShare and ProtoPool,

	Guidance						Prototype					
	Acc		Con		Sta		Acc		Con		Sta	
	w/o Guide	Guide	w/o Guide	Guide	w/o Guide	Guide	Un	Ce	Un	Ce	Un	Ce
VGG16	52.2	**83.4**	42.4	**58.1**	67	**75.6**	79.6	**83.4**	56.3	**58.1**	69.3	**75.6**
VGG19	51.9	**83.8**	40.7	**59.4**	68.2	**77.9**	79.9	**83.8**	58	**59.4**	72.2	**77.9**
Res34	53.6	**85.1**	45	**70.9**	67	**76.1**	82.2	**85.1**	65.6	**70.9**	74.5	**76.1**
Res152	50.4	**85.3**	36.6	**62.3**	50.9	**75.7**	82.9	**85.3**	58.1	**62.3**	72.6	**75.7**
Den121	54.5	**86.2**	53.4	**75.6**	56.1	**73.6**	83.8	**86.2**	72.2	**75.6**	71.7	**73.6**
Den161	59.6	**86.7**	51.3	**73.5**	61.1	**76.3**	83.1	**86.7**	69.5	**73.5**	73.5	**76.3**

Table 3: Ablation study evaluating the effectiveness of coarse location guidance and the extent to which the uncertain prototype reflects the characteristics of the certain prototype. The columns 'w/o Guide' and 'Guide' indicate the absence and application of coarse location guidance, respectively, while 'Un' and 'Ce' denote the uncertain and certain prototypes.

	Acc	Con	Sta
	Original/Perturb	Original/Perturb	Original
ViT-T	83.3 / 80.5	45.2 / 39.7	61.4
ViT-S	85.5 / 82.1	46.8 / 41.2	66.9

Table 4: Ablation study of the proposed method applied to ViTs, with quantitative evaluation on CUB-200-2011.

achieve comparable accuracy on the original inputs. However, substantial differences appear under perturbation, with PINet exhibiting the smallest performance drop.

Qualitative Results

Qualitative evaluation results for PINet and other comparative models using ResNet-152 and DenseNet-161 are presented in Fig.4 and Fig.5. Each figure illustrates visualization in terms of consistency and stability. Fig.4 presents the results for both original and noise-added inputs. For original inputs, all models tend to highlight specific parts within the object; however, comparative models often emphasize adjacent regions depending on the object's orientation or posture. In contrast, PINet consistently highlights the same region, demonstrating qualitative results that are consistent with its quantitative performance. Fig.5 shows the results when prototype information obtained from original inputs is directly applied to noise-added inputs. In comparative models, noise often leads to broader or shifted highlighted regions, or even entirely different parts. However, PINet, there are subtle differences in the values within the highlighted region, consistently emphasizes the same part regardless of noise. These results demonstrate that incorporating uncertain features contributes to enhance robustness to noise. Additional results are provided in the supplementary material.

Ablation Study

We performed ablation studies to assess the effectiveness of coarse location guidance and to evaluate the extent to which the uncertain prototype reflects the characteristics of the certain prototype. The proposed mechanism was also validated using Vision Transformer (ViT) (Dosovitskiy et al. 2020). Here, Acc refers to accuracy, Con to consistency score, and Sta to stability score. In Tab. 3, 'W/o Guide' refers to the absence of coarse location guidance, while 'Guide' indicates its application. As shown in Tab. 3, removing coarse location guidance leads to a significant decrease in all evaluation metrics, especially for ResNet and DenseNet. This demonstrates that coarse location guidance enables backbone networks to capture rough features of the input. Additionally, to assess how much the uncertain prototype reflects the characteristics of the certain prototype, we applied the uncertain prototype to the original input, with results shown in Tab. 3. 'Un' stands for the uncertain prototype and 'ce' for the certain prototype. While some performance degradation is observed compared to using the certain prototype, PINet still outperforms several models and maintains superior performance under perturbed conditions. Tab. 4 presents the results of applying the proposed mechanism to ViTs. While the accuracy is comparable to that of other backbone networks, scores for the remaining evaluation metrics were relatively reduced. This appears to be due to the structural and operational differences of ViTs. We plan to address this in future work to enhance general applicability.

Conclusion

In this study, we propose the Prototype in Imagery Network (PINet), a framework designed to enhance the stability of prototype-based explanations. PINet is inspired by the concept of Phantasia, which describes the ability to recognize the coarse characteristics of objects through mental imagery. By modeling this process, our framework aims to improve robustness against input perturbations. We consider the coarse characteristics identified by Phantasia as features that can be extracted under perturbed conditions. To achieve this, we utilize empty inputs and coarse location guidance in feature extraction to capture rough but meaningful features. These features are combined with an additional prototype, termed the uncertain prototype, to construct an uncertain representation. During training, this uncertain representation is integrated into the conventional representation via similarity-based difference computations at both the feature map and prototype levels. We evaluate the effectiveness of PINet through qualitative and quantitative analyses on original and noise-perturbed inputs. Experimental results demonstrate that PINet exhibits superior robustness to noise compared to existing part-prototype networks.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. RS-2024-00449891), the IITP(Institute of Information & Coummunications Technology Planning & Evaluation)-ITRC(Information Technology Research Center) grant funded by the Korea government(Ministry of Science and ICT)(IITP-2025-RS-2024-00437718), and the Regional Innovation System & Education (RISE) Glocal 30 Program through the Daegu RISE Center, funded by the Ministry of Education (MOE) and the Daegu, Republic of Korea.(2025-RISE-03-001)

References

Carmichael, Z.; Lohit, S.; Cherian, A.; Jones, M. J.; and Scheirer, W. J. 2024a. Pixel-grounded prototypical part networks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 4768–4779.

Carmichael, Z.; Redgrave, T.; Cedre, D. G.; and Scheirer, W. J. 2024b. This probably looks exactly like that: An invertible prototypical network. In *European Conference on Computer Vision*, 221–240. Springer.

Carmichael, Z.; and Scheirer, W. J. 2021. A framework for evaluating post hoc feature-additive explainers. *arXiv preprint arXiv:2106.08376*.

Chen, C.; Li, O.; Tao, D.; Barnett, A.; Rudin, C.; and Su, J. K. 2019. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32.

Donnelly, J.; Barnett, A. J.; and Chen, C. 2022. Deformable protopnet: An interpretable image classifier using deformable prototypes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10265–10275.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708.

Huang, Q.; Xue, M.; Huang, W.; Zhang, H.; Song, J.; Jing, Y.; and Song, M. 2023. Evaluation and improvement of interpretability for self-explainable part-prototype networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2011–2020.

Krause, J.; Stark, M.; Deng, J.; and Fei-Fei, L. 2013. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, 554–561.

Larner, A.; Leff, A.; and Nachev, P. 2024. Phantasia, aphantasia, and hyperphantasia: Empirical data and conceptual considerations. *Neuroscience & Biobehavioral Reviews*, 105819.

Nam, W.-J.; Choi, J.; and Lee, S.-W. 2021. Interpreting deep neural networks with relative sectional propagation by analyzing comparative gradients and hostile activations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 11604–11612.

Nam, W.-J.; Gur, S.; Choi, J.; Wolf, L.; and Lee, S.-W. 2020. Relative attributing propagation: Interpreting the comparative contributions of individual units in deep neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 2501–2508.

Nam, W.-J.; and Lee, S.-W. 2024. Illuminating Salient Contributions in Neuron Activation with Attribution Equilibrium. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Nauta, M.; Schlötterer, J.; Van Keulen, M.; and Seifert, C. 2023. Pip-net: Patch-based intuitive prototypes for interpretable image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2744–2753.

Nauta, M.; Van Bree, R.; and Seifert, C. 2021. Neural prototype trees for interpretable fine-grained image recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 14933–14943.

Rymarczyk, D.; Struski, Ł.; Górszczak, M.; Lewandowska, K.; Tabor, J.; and Zieliński, B. 2022. Interpretable image classification with differentiable prototypes assignment. In *European Conference on Computer Vision*, 351–368. Springer.

Rymarczyk, D.; Struski, Ł.; Tabor, J.; and Zieliński, B. 2021. Protopshare: Prototypical parts sharing for similarity discovery in interpretable image classification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 1420–1430.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, 618–626.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Slack, D.; Hilgard, S.; Jia, E.; Singh, S.; and Lakkaraju, H. 2020. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 180–186.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The caltech-ucsd birds-200-2011 dataset.

Wang, C.; Liu, Y.; Chen, Y.; Liu, F.; Tian, Y.; McCarthy, D.; Frazer, H.; and Carneiro, G. 2023a. Learning support and trivial prototypes for interpretable image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2062–2072.

Wang, G.; Li, J.; Tian, C.; Ma, X.; and Liu, S. 2023b. A Novel Multimodal Prototype Network for Interpretable Medical Image Classification. In *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2577–2583. IEEE.

Wang, J.; Liu, H.; Wang, X.; and Jing, L. 2021. Interpretable image recognition by constructing transparent embedding space. In *Proceedings of the IEEE/CVF international conference on computer vision*, 895–904.

Zhang, Z.; Liu, Q.; Wang, H.; Lu, C.; and Lee, C. 2022. Protgnn: Towards self-explaining graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, 9127–9135.