Master Thesis

Evaluating the Effect of Generated Captions on a Novel Multimodal Prototype-Based Network

by

Yixing Wang
yyg760

First Supervisor: Filip Ilievski

July 20, 2025

Submitted in partial fulfillment of the requirements for
the VU degree of Master of Science in Artificial Intelligence

Evaluating the Effect of Generated Captions on a Novel Multimodal Prototype-Based Network

Yixing Wang

[1] Vrije Universiteit Amsterdam, Amsterdam
[2] The Netherlands
[3]

y41.wang@student.vu.nl

Abstract. This study introduces a novel Multimodal Prototype-Based Network (mPBN), with pre-trained VisualBERT backbone encoder, to investigate the impact of generated natural-language captions on the behavior of prototype-based learning models. Experiments were conducted on the Food101 dataset using a two-step pipeline: first, captions were generated using BLIP and BLIP-2, then the mPBN was trained using these captions. The attention shift caused by the generated captions was qualitatively visualize by comparing the DeepSHAP activations between mPBNs and their unimodal Prototype-Based Networks (PBNs) counterparts that were trained *without* generated captions. Results indicate that generated captions can both enhance or degrade model performance, depending on specific factors such as the number of prototypes per class and the captioning style. These findings suggest that future multimodal systems should carefully consider caption quality and model configuration to fully benefit from language-enhanced representations.

Keywords: Prototype-based Network · Prototype-Based Learning · Multimodaliy · Caption Generation · VisualBERT · DeepSHAP · Attention Visualization

1 Introduction

Prototype-based Networks (PBNs) have emerged as a promising type of machine learning model that classifies inputs by comparing them to learned prototypes, where prototypes are defined as latent representations that resemble typical examples of a given class. These models have demonstrated remarkable success in enhancing the image and text classification performance with the introduction of prototypes, and have better interpretability and robustness compared to traditional black-box models. For example, Sourati et al. used PBNs to improve the robustness of text classification [27], Das et al. implemented PBN to enhance propaganda detection [6], Chen et al. introduced *ProtoPNet* for classifying bird species through visual part matching [5], and Hase et al. introduced a hierarchical prototype network that increases the interpretability in hierarchical image recognition [8] in taxonomical order. More recently, cross-modal

applications have emerged, such as multimodal prototypical networks (mPBNs) that combine visual and textual information for few-shot learning introduced by Pahde et al.[21], and SPANet, which links human-annotated semantic tags with visual features for object recognition by Wan et al..[33]. These two mPBN studies further demonstrate the benefit of combining modalities, showing comparably progressing model improvement, enabling the possibilities of mPBNs for few-shot learning, when data suffers scarcity, and real application scenarios with high-reliability requirements.

However, two main limitations remain in current research. First, existing approaches are often limited in their modality scope and scalability: most PBNs are unimodal, operating on either visual [8] [26] [11] or textual [27] [6] inputs, and the few existing multimodal PBNs — such as those proposed by Pahde et al. [21] and Wan et al. [33] — rely on handcrafted, part-level annotations or project textual features into the visual domain using GAN-based approaches. Such methods limit scalability and generalization to real-world, potentially noisy image-caption datasets where no part-level labels or aligned vocabularies are available. Second, these multimodal approaches largely report performance improvements, but do not investigate the risk of mismatched or inconsistent prototypes across modalities, which may lead to decision-making that does not align with human semantics.

To address these limitations and explore the capacity of PBNs to process natural, unstructured, free-form visual-language input, this study introduces a novel multimodal Prototype-Based Network (mPBN), expanded from the architecture of ProtoPNet [5], along with a two-step pipeline: first generates natural-language captions using pretrained vision-language models (BLIP [16] and BLIP-2 [15]), and then utilizes the generated captions as input to the mPBN, which jointly encodes image-caption pairs using VisualBERT [17]. This enables the model to learn prototypes with the ProtoPNet architecture in a shared visual-semantic space. Unlike prior work that relies on human-annotated semantic parts [33], this method requires *no* manual annotation and is applicable to large, unstructured image-caption datasets. This study further investigates whether incorporating such text enhances or degrades classification performance, and whether multimodal integration improves prototype quality or leads to semantic inconsistencies [10] with DeepSHAP [18] [29], to qualitatively analyze how natural-language captions influence prototype activation and model attention.

The research questions in this study are defined as follows:

1. Does incorporating natural-language text as input improve or degrade the classification performance of a prototype-based network (PBN), compared to a unimodal (image-only) variant?
2. How, and to what extent, does multimodal input influence the model's attention or prototype activation patterns, as revealed through feature attribution techniques?

This study makes the following findings and contributions: (1) Integrating natural-language captions into a PBN can both improve and degrade classification performance, depending on factors such as the number of prototypes and

the style or informativeness of the generated captions (addressing RQ1), (2) Multimodal PBNs often learn more abstract or semantically aligned prototypes than their unimodal counterparts, although this effect varies based on caption style and prototype configuration (addressing RQ2), (3) DeepSHAP analysis reveals that textual input can meaningfully shift model attention—sometimes toward more relevant visual regions, and sometimes away from the target object—highlighting both the benefits and risks of multimodal integration (addressing RQ2), and (4) To the best of my knowledge, this is the first PBN to use a fully automated, two-step pipeline with natural-language caption generation and joint encoding via VisualBERT, enabling scalable multimodal interpretability without human-labeled part annotations.

2 Related Work

The origin of Prototype-Based Networks (PBNs) was inspired by cognitive psychological models of human information processing, where humans recognize objects by identifying salient parts—referred to as "prototypes"—and integrating both visual and linguistic cues for decision-making [28] [32]. This idea motivated efforts to develop algorithms that better imitate such mechanisms [13]. Prior to the formalization of PBNs, Bavaresco et al. [2] and Storrs et al. [30] employed Representational Similarity Analysis (RSA) to compare deep neural network (DNN) representations with brain activity. Their findings revealed strong alignment between model activations and human cortical regions involved in semantic reasoning, suggesting that ML models are capable of approximating certain aspects of human cognition and motivating the development of PBNs.

In the textual domain, recent works by Sourati et al. [27] and Das et al. [6] demonstrated that PBNs outperform standard large language models (LLMs) in robustness, particularly under adversarial and imbalanced conditions [35] [23]. These models classify examples based on their similarity to learned prototypes and offer a level of interpretability by retrieving nearest-neighbor training examples or visualizing softmax-based prototype assignments.

In the visual domain, Chen et al. [5] were the first to introduce PBNs for image classification, namely *ProtoPNet*, using prototypes to provide interpretable visual matching. However, Hoffmann et al. [10] later showed that their explanation methods might be misleading, as prototype activations can be manipulated without altering classification outcomes. Further work by Hase et al. [8] proposed a hierarchical prototype structure that improved traceability of predictions, while Snell et al. [26] introduced Prototypical Networks for few-shot learning, demonstrating better performance over several baselines. Saralajew et al. [25] recently proposed a probabilistically grounded classification head based on radial basis function (RBF) networks, which eliminates reliance on intermediate similarity activations by integrating both reasoning and class priors directly into the output layer. This design enables interpretable and robust classification through the use of both positive and negative evidence.

Two multimodal PBNs (mPBNs) have been proposed recently to enhance visual representations using text-conditioned features during training [21] [33]. Pahde et al. [21] employed a GAN-based encoder and trained the mPBN in a few-shot learning setting, reporting performance improvements over baselines. Wan et al. [33] introduced an mPBN model trained on human-annotated, coarse-grained image-text pairs, achieving strong performance. However, their study did not examine whether such alignment and reasoning could emerge naturally—without manual annotations—from full image-caption pairs. Both mPBNs also evaluated interpretability primarily at the prototype level and *did not assess cross-modal alignment* or the risk of misleading prototype formation as highlighted by Hoffmann et al. [10]. In contrast, the mPBN model proposed in this study addresses these limitations by eliminating the need for *handcrafted annotations*, operating directly on raw image-caption pairs, and enabling cross-modal attribution via DeepSHAP. This approach offers a scalable and fully interpretable multimodal framework that does not depend on aligned concept vocabularies or synthetic visual features.

The existing multimodal ProtoPNet (mPBN) frameworks have employed various Vision-Language Models (VLMs) as backbone encoders to construct complex joint prototypes. To the best of my knowledge, this study is the first to utilize VisualBERT [17] as the encoder within a prototypical network architecture (see Section 3 for implementation details). VisualBERT is a pre-trained, BERT-based transformer that jointly embeds visual and textual inputs into a shared feature space, capturing their contextual interactions through attention mechanisms. It is used by the novel mPBN introduced in this study as the backbone encoder due to its capacity to produce rich, aligned multimodal representations, which serve as a suitable foundation for learning interpretable prototypes in multimodal settings.

Why is it crucial to investigate the cross-modal alignment and feature attribution? Charmers [4] explained from a philosophical perspective that one shall log in to the model's inner architecture to fully interpret its belief and desire, which shall be achieved by visualizing its *attention*. Without this step, the reliability and robustness of a model will *not* be understood - if users unconditionally trusted the model decisions, without evidence of a model being reliable, the potential biases and hazardous behavior of a model would *not* be realized, hence several ethical concerns would occur, especially in fields where *False Negative* predictions have very limited tolerance. To mitigate this issue, DeepSHAP has been selected by this study as a powerful, mathematically-based analysis tool that *estimates* the Shapley Value of each input feature, enabling visualization of the model's *attention points* [18] [29]. An example use case of DeepSHAP was demonstrated by Ahmed et.al. [1], where they utilized this technique to understand why a brain tumor classifier decided whether a patient has a brain tumor by analyzing whether the classifier spotted the significantly maladaptive areas of an MRI image, hence to decide whether the classifier was safe to use.

3 Methodology

3.1 Prototypical Model

ProtoPNet as Base Architecture *ProtoPNet* [5] is the first prototypical network introduced for image classification, and is used as a base for this study due to its comparable robustness and flexibility in combining various encoders from its original study. A general ProtoPNet architecture consists of three parts: a backbone encoder (e.g.: a Convolutional Neural Network (CNN), such as VGG16 [31]), to extract image features, followed by two additional 1×1 additional convolutional layers to compress the encoded latent space; a joint prototype training layer that takes patches of features from the encoded pooled layer to construct partial prototypes of the given class; a fully-connected last layer that performs a convex optimization to filter out the irrelevant partial prototypes that had a low similarity score, and emphasizes the partial prototypes that contributed to the prediction.

Training a ProtoPNet is divided into three stages: *warm-up*, *joint*, and *last-layer optimization*. During the warm-up stage, the model learns a meaningful latent space by clustering the meaningful image patches using the L^2 distance, updated via Stochastic Gradient Descent (SGD). This ensures that the image patches are close to their semantically similar prototypes of the ground-truth image classes and well-separated from different classes. In the joint training stage, the encoder and prototype layer are trained together and the prototype formation layer via Mini-Batch Gradient Descent to further capture the subtle details from the inputs and to adjust the distances of learned prototypes until convergence. In the last-layer optimization stage, the model adjusts the last-layer weights by minimizing the weights of non-relevant connections (i.e.: connections of the input prototypes that did *not* contribute to the prediction with a low similarity score) to approximately zero. This 3-stage training process is achieved by freezing the gradients of the encoder's layers during the warm-up stage, enabling *only* the gradients of the last layer during the last-layer optimization stage, and enabling the gradients of every layer during the joint stage. Hence, only the designated layers are updated in each stage, while other layers remain frozen, and the weights of other layers will remain constant unless they are enabled.

The backbone encoder is the key component of ProtoPNet: it is responsible for producing a latent representation from the inputs (in the case of CNN, the latent representation is a pooled layer of feature maps). When the backbone encoder functions as a model itself (without prototypes), the latent representation is usually forwarded to a dense layer to calculate raw logits. In ProtoPNet, this process is omitted; instead, convolutional layers are used to produce prototypes by the model's joint layer, where the logits are calculated by the similarity scores of each learned prototype. Then, similarly, the SoftMax probabilities of each prototype-based logits will be calculated via Cross Entropy Function that determines the final prediction. During Prototype Formation, the Prototypical Layer projects each learned prototypes onto an *activation map*, and the similar-

ity scores of each prototype against the model's input are calculated by inverting the L^2 distance of the prototypes.

ProtoPNet is modular and supports interchangeable encoders, which enables the construction of *both unimodal and multimodal variants* within a shared architectural framework, which provides an opportunity for this study to achieve the main research goal: investigating the impact of including generated image captions on PBN's behavior. This study leverages this flexibility to implement two models: (1) a unimodal PBN (referred to as PBN in the following of this paper) using a CNN (e.g., VGG16 [31] or ResNet34 [9]) as the encoder to process visual inputs only; and (2) a multimodal PBN (referred to as mPBN), which uses VisualBERT [17] as the encoder to jointly encode image-caption pairs. In both cases, the prototype layer receives pooled embeddings and computes similarity-based logits for classification. Section 4 details the specific encoder choices and prototype configurations.

What is a Prototype? A *prototype* is an abstract, semantically meaningful latent representation of a class (or a salient part of it) learned during the training of a Prototypical Network. In *ProtoPNet*, for instance, these prototypes are localized features that reflect prototypical parts of the training data. The prototypes are integrated into the decision-making process of the model. Each prototype is trained to be representative of a certain class and is encoded as a small latent tensor ($1 \times 1 \times D$ in the original *ProtoPNet study*) extracted from the convolutional feature map. These are then used to reason about input images in a case-based manner.

During training, prototypes are optimized to ensure that they are (1) close to latent features of training examples from their associated class, and (2) far from features of other classes. This is achieved through two losses:

- **Clustering loss** encourages at least one patch from a training image to be close (in latent space) to one of the prototypes of the correct class:
  $\mathcal{L}_{\text{clst}} = \frac{1}{n} \sum_{i=1}^{n} \min_{\substack{j:\mathbf{p}_j \in \mathcal{P}_{y_i} \\ z \in \text{patches}(f(\mathbf{x}_i))}} \|z - \mathbf{p}_j\|_2^2$
- **Separation loss** encourages all patches from a training image to stay distant from prototypes of incorrect classes:
  $\mathcal{L}_{\text{sep}} = -\frac{1}{n} \sum_{i=1}^{n} \min_{\substack{j:\mathbf{p}_j \notin \mathcal{P}_{y_i} \\ z \in \text{patches}(f(\mathbf{x}_i))}} \|z - \mathbf{p}_j\|_2^2$

Here, $\mathcal{P}_{y_i}$ is the set of prototypes assigned to class y_i, $f(\mathbf{x}_i)$ is the latent representation of image $\mathbf{x}_i$, and z is a patch in the feature map.

These losses, together with cross-entropy and convex optimization of the final linear classifier layer, shape the latent space into meaningful clusters where similarity to a prototype leads to an interpretable classification decision.

3.2 Dataset Selection and Caption Generation

Selecting a suitable image dataset for multi-class classification and further generating image-oriented captions for it is a key component of this study. The dataset

must meet the following criteria: (1) it must support single-label classification with mutually exclusive classe; (2) it has a well-balanced class distributionand contain a sufficient number of samples to support training under the time and hardware constraints of this study; and (3) it can accommodate *generated* captions without interference from human supervision.

Two pre-trained caption generation tools are selected based on the following requirements: (1) both can produce naturalistic, textual descriptions that reflect real-world captioning scenarios when applied to the chosen single-label dataset; (2) the captions do *not explicitly* disclose class labels; (3) the captions are *not* human-annotated; and (4) the two sets of generated captions exhibit human-distinguishable stylistic differences. The rationale behind requirement (4) is that the influence of captioning style on prototype formation and multimodal alignment remains underexplored. Therefore, this study incorporates both styles to examine their impact on classification performance and interpretability in multimodal Prototype-Based Networks (mPBNs), providing deeper insights into how text variability influences learned prototypes.

Further implementation details and dataset-specific configurations are provided in Section 4.

3.3 DeepSHAP for Model Attention Visualization

The original ProtoPNet includes a prototype visualization method that identifies the image patches most strongly activating each prototype and maps them to their nearest training samples [5]. However, this technique is specifically designed for image-only inputs and does not generalize to multimodal Prototype-Based Networks (mPBNs), where prototypes exist in a joint visual-textual embedding space. Adapting this method for multimodal settings would require substantial architectural modifications and is considered beyond the scope of this study. Therefore, an alternative, architecture-agnostic attribution method is adopted to fulfill the research objective—namely, comparing the *attention dynamics* between PBNs and mPBNs [29].

DeepSHAP [20] is a model-specific attribution method within the SHAP value family [29], designed for gradient-based networks. It is selected in this study for its strong mathematical grounding and demonstrated stability in producing reliable feature attributions [37] [29]. DeepSHAP is inspired and built on two foundational concepts: *Shapley values*[34] and *DeepLIFT*[14].

Shapley values are rooted in game theory and attribute the total output of a system to its individual contributors by quantifying how much each individual affects the outcome when added to or removed from a coalition [34]. In the context of neural networks, these "individual contributors" correspond to input features.

DeepLIFT builds on this idea by estimating each feature's contribution through a comparison between a *target input* (i.e., the datapoint for which attribution is desired) and a *reference input* (a neutral or baseline example). It propagates the *activation differences* between the target and reference input from the model's

output layer back to the input features, using a principle known as *summation-to-delta* [20]. This process reveals how much each feature in the target input deviates from the reference and contributes to the final output. The mathematical expression of DeepLIFT estimating each feature importance is as follows:

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)]$$

Here: - F denotes the full set of input features, - S is a subset of F that does not contain feature i, - $f_{S \cup \{i\}}$ and f_S are the model's predictions with and without feature i, respectively. This equation computes the marginal contribution of feature i over all possible coalitions $S \subseteq F \setminus \{i\}$, weighted by their occurrence probability.

DeepSHAP extends DeepLIFT by computing such attribution scores for each feature in the target input across a set of reference inputs, collectively referred to as the **background dataset**. By averaging attributions across these references, DeepSHAP produces a more robust estimate of each feature's importance—closely approximating the theoretical Shapley value. Prior studies recommend that this background set be large, balanced, and representative of the true data distribution [36] [19], in order to produce reliable and unbiased attribution maps. This study takes these suggestion into consideration, and further discusses the exact background set choice in Section 4.2.

Although the encoder architecture differs between the unimodal (PBN) and multimodal (mPBN) networks, both rely on a prototype-based reasoning mechanism. Since both models are differentiable, DeepSHAP can effectively backpropagate importance scores from the prototype similarity layer to the input features, allowing us to estimate how different modalities (e.g., image regions or caption tokens) contribute to the final class decision. Several prior works have successfully applied DeepSHAP in medical and vision applications, such as brain tumor detection [1]. Importantly, in this study, DeepSHAP is used as a **qualitative post-hoc analysis tool**. It enables the examination of how the inclusion of text affects prototype attention by comparing attribution maps between PBN and mPBN models. This sheds light on the interpretability of multimodal Prototype-Based Networks and the extent to which textual input influences classification behavior.

4 Experimental Setup

4.1 Data Preparation

This study adopts the *Food101* dataset [3], which is single-labeled, well-balanced, and tailored for image classification without human-annotated captions. This dataset satisfies the requirements listed in Section 3.2. Food101 contains approximately 110,000 images distributed across 101 mutually exclusive cuisine classes, with roughly 1,000 images per class. The dataset is divided into training, validation, and test sets using a 70-15-15 ratio. Although Food101 provides

a predefined train-test split (75-25), a separate validation set was required for model fine-tuning. Therefore, the dataset was reshuffled and re-split accordingly.

To generate captions that adhere to the criteria in Section 3.2, two pretrained vision-language models — BLIP [16] and BLIP-2 [15] — were used. BLIP (Bootstrapping Language-Image Pretraining) is a unified vision-language model that employs a Multimodal Mixture of Encoder-Decoder (MED) architecture and a data bootstrapping method (CapFilt). It refines noisy web data by generating synthetic captions and filtering out irrelevant ones, allowing BLIP to produce rich, diverse, and relevant captions for downstream tasks [16]. BLIP-2 builds upon BLIP but introduces a more computationally efficient approach by bootstrapping from frozen pretrained image encoders and large language models (LLMs). It uses a lightweight Querying Transformer (Q-Former) to bridge the modality gap, enabling zero-shot captioning and improved instruction-following [15].

For each image, two distinct captions are generated using both models. To prevent accidental data leakage, all generated captions are automatically filtered to ensure that the ground-truth class name is *not* present in the text - if a caption explicitly contained the class label, it was replaced with a neutral term such as "dish" or "cuisine." For instance, given an image of *caesar salad*, a generated caption such as *"a plate of caesar salad topped with cheese and tomato on the table"* will be rewritten as *"a plate of dish topped with cheese and tomato on the table"*.

Note that in Food101, some class labels exhibit partial lexical overlap (e.g., *caesar salad* and *beet salad*). In these cases, generic terms like salad may appear in captions without being exact label matches. Such partially overlapping terms are not filtered out, based on the following rationale: (1) a generic term like salad is non-exclusive and does not uniquely identify a class; (2) such terms often appear as side components in many unrelated classes (e.g., a burger image with salad on the side), making them semantically plausible across categories; and (3) removing too many generic or overlapping terms would degrade the linguistic richness of the captions, which may hinder the multimodal encoder's ability to form robust visual-semantic associations. Excessive filtering could therefore destabilize training and reduce the potential for meaningful cross-modal alignment—the very phenomenon this study aims to investigate. Filbrandt et al. [7] demonstrated a case that learning from partially overlapping labels can be both realistic and beneficial in weakly supervised learning. While their research objective differs from that of this study, the underlying principle supports the decision to retain semantically relevant generic terms. Nonetheless, to ensure that this choice does not lead to label leakage, a preliminary overfitting test is conducted to confirm that the model cannot trivially infer ground-truth classes from unfiltered generic terms in the captions.

4.2 Models

Model Architecture Addressing the Section 3.1, this study trains and evaluates a total of **18 models**, varying across three main design dimensions: (1) the type of encoder, (2) the number of prototypes per class (1, 2, 5, or 10), and

(3) the type of caption input (i.e., BLIP or BLIP-2). The models fall into the following three categories:

1. **Non-prototypical Baseline Models:** Two standard convolutional classifiers trained without prototype reasoning. In this study, these models are denoted as: **a) VGG16-Baseline** [31] and **b)ResNet34-Baseline** [9].

2. **Unimodal Prototype-Based Networks (PBNs):** Eight models based on the ProtoPNet architecture [5], using either VGG16 or ResNet34 as the visual encoder and varying the number of prototypes per class.

   In this study, these models are denoted as:
   a) PBN-VGG16-1P, **b) PBN-VGG16-2P**, **c) PBN-VGG16-5P**, **d) PBN-VGG16-10P**, and
   e) PBN-ResNet34-1P, **f) PBN-ResNet34-2P**, **g) PBN-ResNet34-5P**, **h) PBN-ResNet34-10P**.

3. **Multimodal Prototype-Based Networks (mPBNs):** Eight models based on the ProtoPNet architecture [5], with **VisualBERT** [17] as a multimodal encoder. Each model is trained with either **BLIP** [16] or **BLIP-2** [15] captions and uses 1, 2, 5, or 10 prototypes per class.

   in this study, these models are denoted as:
   a) mPBN-1P-BLIP, **b) mPBN-2P-BLIP**, **c) mPBN-5P-BLIP**, **d) mPBN-10P-BLIP**,
   and
   e) mPBN-1P-BLIP2, **f) mPBN-2P-BLIP2**, **g) mPBN-5P-BLIP2**, **h) mPBN-10P-BLIP2**.

The following sections of this paper adhere to the notations defined above. The suffix P indicates the number of prototypes per class that a model is trained with. *BLIP* and *BLIP2* refer to the captioning style—i.e., which caption generator was used to produce the captions used during training.

Unimodal Prototype-Based Network (PBN) The unimodal Prototype-Based Network (PBN) in this study adopts the original *ProtoPNet* architecture, which is designed to operate exclusively on visual inputs without textual information [5]. ProtoPNet supports various convolutional architectures (e.g., VGG, ResNet, DenseNet) as backbone encoders. In this work, VGG16 [31] and ResNet34 [9] are selected based on their strong performance in image classification and their moderate architectural complexity. These models offer a balance between expressive capacity and computational efficiency, making them suitable for training under the time and hardware constraints of this project.

The backbone encoder extracts a feature map from the input image and applies spatial pooling to produce a compact latent representation. The prototype layer then retrieves and clusters local patches (partial prototypes) from this feature map, as discussed in Section 3.1. These prototypes are compared to the

input features via L^2 distance, and the resulting similarity scores are projected onto an activation map. The class logits are computed based on the prototype similarities and passed through a final softmax layer to yield predictions. A schematic overview of the full architecture is provided in Figure 4 in Section 8.2.

Multimodal Prototype-Based Network (mPBN) Figure 5 in Section 8.2 illustrates the model architecture of mPBN. In contrast to the unimodal PBN, the *Multimodal Prototype-Based Network (mPBN)* takes both an image and its corresponding generated caption as input for classification. These two modalities are jointly embedded using a pre-trained *VisualBERT* model, which functions as a multimodal Transformer encoder. Captions are tokenized using a BERT tokenizer, while image features are extracted via a pre-trained *VGG16* network. The convolutional output from VGG16 is passed through a global average pooling layer, followed by a linear projection that transforms the resulting 512-dimensional visual vector into a 768-dimensional embedding, aligning it with VisualBERT's expected token size. This projection bridges the CNN and Transformer domains, enabling multimodal fusion.

The textual and visual embeddings are then concatenated and passed through VisualBERT's Transformer layers, which apply multi-head self-attention over the entire sequence. The final output is a sequence of contextualized embeddings, where the special `[CLS]` token (768 dimensions) serves as a pooled semantic representation capturing cross-modal interactions between caption tokens and visual features.

This `[CLS]` embedding is used as input to the prototype layer. Unlike traditional ProtoPNet, where prototypes represent spatial visual patches, the prototypes in mPBN are learned in the joint semantic space produced by VisualBERT. Specifically, the model learns K prototype vectors, each in the same 768-dimensional space as the `[CLS]` token. During inference, the model computes the squared ℓ_2 distance between the `[CLS]` embedding and each prototype, and uses these distances to compute similarity scores. A final linear layer maps these scores to class logits. This enables mPBN to perform interpretable classification based on *multimodal semantic similarity*, rather than purely spatial visual features.

Non-prototypical Baseline Models VGG16 [31] and ResNet34 [9] are used as baseline CNN classifiers *without* prototype layers. These models were chosen due to their popularity, proven performance, and use as backbones in the original ProtoPNet [5]. They serve as benchmarks for evaluating the benefit of prototype-based learning. ResNet34 includes residual connections for better gradient flow, while VGG16 uses a deeper feedforward structure. Both models output class logits via a final dense layer and are trained using cross-entropy loss.

Hyperparameters This study focuses on two key hyperparameters: the number of prototypes per class and the learning rate.

Prior work has used both single and multiple prototypes per class to balance simplicity and intra-class variance modeling [26] [6] [5] [27]. Following recommendations from ProtoPNet [5], this study evaluates four settings—1, 2, 5, and 10 prototypes per class—to examine how prototype granularity affects performance and interpretability.

While ProtoPNet used a default learning rate, fine-tuning revealed this was suboptimal for all models. In particular, ResNet-based PBNs and some mPBNs showed instabilities (e.g., exploding gradients). To address this, learning rates were empirically reduced per model type until stable convergence was achieved. Gradient clipping was also applied for stabilization [22]. For baseline (non-prototypical) CNNs, a standard learning rate of 1×10^{-4} is used [12] [24]. Section 8.1 summarizes the exact learning rate values each model adapted in this study.

Model Training All models in Table 2 are initially trained using the regular learning rate configuration to observe whether this configuration can adapt to the novel dataset in this study. If not, subsequent fine-tuning is conducted iteratively, adjusting learning rates to achieve convergence and maximize performance stability.

For all models, training follows the three-phase procedure described in Section 3.1, consisting of a warm-up, joint training, and final-layer optimization phase. Early stopping was employed with a patience threshold of 10 epochs.

Training is conducted on the DAS-6 supercomputing infrastructure [4]. at Vrije Universiteit Amsterdam.

4.3 Posthoc Analysis

The posthoc analysis begins by examining the overall performance of the models through two steps: (1) evaluating their test accuracies and evaluation losses, and (2) identifying *model-exclusive* true positive prediction counts—i.e., datapoints correctly classified by one model but misclassified by others. This analysis provides insight into the relative strengths of each model and serves as a basis for the subsequent case-based comparison.

To interpret how different factors affect model behavior, DeepSHAP is used to generate attention visualizations on selected datapoints. These attention maps help reveal how prototype activation and class decisions are influenced across the following scenarios:

1. **Contribution of generated natural-language text (addressing Question 1):** Compares mPBNs and their unimodal PBN counterparts (with identical prototype counts) to assess whether textual input shifts the model's attention and decision-making behavior.

[4] Henri Bal, Dick Epema, Cees de Laat, Rob van Nieuwpoort, John Romein, Frank Seinstra, Cees Snoek, and Harry Wijshoff: "A Medium-Scale Distributed System for Computer Science Research: Infrastructure for the Long Term", IEEE Computer, Vol. 49, No. 5, pp. 54-63, May 2016.

2. **Impact of captioning style and quality (addressing Question 2):** Compares mPBNs trained with BLIP versus BLIP-2 captions, isolating the impact of captioning style while holding all other variables constant.
3. **Effect of prototype count (addressing Question 2):** Examines the influence of different prototype counts per class (1P, 2P, 5P, 10P) on model interpretability and classification performance within the same captioning setup.

In each case above, DeepSHAP is employed to compare the attention distributions of the models on selected datapoints, providing insights into (1) modality contributions, (2) caption dependencies, and (3) the role of prototype diversity.

Background Set Selection for DeepSHAP As discussed in Section 3.3, the choice of background dataset is critical for accurate and reliable attribution using DeepSHAP. Prior work [36] [19] recommends that an ideal background set should: (1) exhibit class balance and sufficient sample size to approximate the true data distribution, and (2) preferably consist of samples from the dataset's validation or held-out partition to avoid overfitting or distributional shift.

While using the full validation set would be optimal in theory, the size of Food101's validation set exceeds the computational capacity of the DAS-6 infrastructure used in this study. As a compromise, a stratified subset of 202 examples is sampled from the validation set, maintaining class balance by selecting two samples per class. This design provides sufficient diversity and class coverage for stable Shapley value estimation, while remaining computationally tractable.

5 Results

5.1 Captioning Style Differences

To enhance caption diversity, prompt tuning was applied to BLIP-2. Notably, the two captioning models produced distinct stylistic outputs, even after prompt tuning. *BLIP* typically generated longer, more descriptive captions that often included contextual information about the environment surrounding the cuisine. In contrast, *BLIP-2* tended to produce shorter, more object-centric captions with less contextual detail.

Given that the effect of caption style on the performance of Prototype-Based Networks (PBNs) has not been previously explored, this study investigated how stylistic differences influence the behavior and performance of the multimodal PBN (mPBN). A comparative evaluation was conducted using the same mPBN architecture trained separately on captions generated by BLIP and BLIP-2. The results offer preliminary insights into the impact of caption variability on multimodal interpretability and classification performance. Example captions illustrating the stylistic contrast are provided in the case-based visualizations in Section 5.3.

5.2 Overall Model Performance

Table 3 in Section 8.3 summarizes the test accuracy and evaluation loss for all model configurations. A key observation is that the number of prototypes per class significantly influences performance. In unimodal PBNs, classification accuracy improves as the prototype count increases beyond one. In contrast, mPBNs experience a decline in performance with higher prototype counts—regardless of whether captions are generated by BLIP or BLIP-2. This suggests that when incorporating generated natural-language captions, fewer prototypes support better generalization, whereas visual-only models benefit from a more granular prototype representation.

Among all models, the highest-performing configuration is **mPBN-BLIP2-1P**, which achieves a test accuracy of 77.34% and an evaluation loss of 1.37. The lowest-performing model is **PBN-ResNet34-1P**, with a test accuracy of only 26.92% and a high loss of 4.62—indicating ineffective prototype learning and poor training stability.

All models were initially trained using ProtoPNet's original learning rate configuration (denoted as *Regular*; see Section 4.2 and Table 2). While this setting was sufficient for most PBNs, several mPBN configurations—particularly mPBN-BLIP-1P, -5P, -10P and mPBN-BLIP2-2P—exhibited instability during training, likely due to exploding gradients. These issues were identified based on two consistent indicators: (1) validation accuracy plateaued around 0.01, and (2) sudden drops in accuracy following initial improvement. Training stability was restored in these cases by lowering the learning rate and applying gradient clipping by setting the maximum allowed L^2 norm to 1.0[22].

PBN-ResNet34-1P was the only model that consistently failed to converge, even after aggressive learning rate reduction - while other models achieved a test accuracy of at least 0.3 with much less aggressive learning rate configurations, this model reached solely 0.27. Its poor performance may stem from architectural incompatibility with the 1-prototype-per-class setting or ineffective feature extraction under unstable training conditions. Further investigation is required to better understand the performance limitations of this model.

Overall, these findings reveal two key trends: (1) mPBNs generally require lower learning rates due to their increased architectural complexity and sensitivity to gradient instability, (2) The optimal number of prototypes depends on the input modality: multimodal models perform best with fewer prototypes (1–2P), while unimodal models benefit from more prototypes (5–10P) to better capture intra-class visual variance.

5.3 Case-Based Visualization

Addressing the three hypothesized scenarios outlined in Section 4.3, this section presents case-based analyses to explore model behavior in greater depth. Before delving into specific qualitative examples, a quantitative overview of model-exclusive true positive predictions is provided. Specifically, pairwise counts are reported to indicate how often one model correctly classifies samples that its

counterpart does not. This overview contextualizes the impact of generated captions, captioning style, and prototype count prior to individual case inspection, and summarizes *the number of occurrences of each possible case in each hypothesized scenario* in Section 4.3. Additionally, to ensure a fair and stable comparison, only the PBN models with VGG16 encoders are used in this analysis, as one of the ResNet34-based PBNs (PBN-ResNet34-1P) failed to train successfully.

In this section, each one of the three hypothesized scenarios is provided with visualization examples. Each example includes the original image, its generated caption, and the attention maps of selected model pairs for comparison. The attention visualizations were generated using DeepSHAP, following the procedure described in Section 3.3. These visualizations highlight the image regions that contributed most to the model's prediction, effectively illustrating where each model focused its attention during decision-making. Each visualization example includes a color bar indicating the strength of feature attribution; higher values highlight regions that contributed more significantly to the model's decision.

Model-Exclusive True Positive Count Figure 7 in Section 8.4 illustrates the number of model-exclusive true positive predictions made by each PBN and mPBN pair with identical prototype counts. Addressing the first research question in Section 1 and the first hypothesis in Section 4.3, the results show that mPBNs outperform their unimodal counterparts when using 1 or 2 prototypes per class, whereas PBNs achieve better results with 5 or 10 prototypes per class. This finding further enhances the result in Section 5.2, which indicates that textual input can improve model decisions when the number of prototype choices is limited to 1 and 2, and with larger prototype sets, this advantage fades or reverses, suggesting diminishing returns or possible redundancy.

Addressing Research Question 2 and the second hypothesized scenario of the posthoc analysis (Section4.3), Figure8 in Section 8.4 compares model-exclusive true positive predictions between mPBN-BLIP and mPBN-BLIP2 across varying prototype counts. The results indicate that mPBN-BLIP makes the most true positive predictions with 5 prototypes per class, while mPBN-BLIP2 makes the most with 10 prototypes per class. Prediction counts across other prototype settings for mPBN-BLIP2 remain relatively stable. These findings suggest that captioning style plays a substantial role in shaping prototype effectiveness in multimodal networks. As such, careful consideration of caption generation methods is essential when designing mPBNs for real-world applications.

Addressing Research Question 2 and the third hypothesized scenario in Section 4.3, Figure 6 in Section 8.4 compares model-exclusive true positive predictions across different prototype counts within each captioning style. For both mPBN-BLIP (top row) and mPBN-BLIP2 (bottom row), a clear pattern emerges: models with fewer prototypes per class (e.g., 1P or 2P) tend to outperform models with more prototypes (5P or 10P) in terms of unique true positive classifications. This supports the notion that for mPBNs, decreasing prototype granularity improves class coverage to a certain degree.

Taken together, these findings highlight that both the number of prototypes per class and the tuning of this hyperparameter are critical, as they collectively impact the model performance. Neither one of these criteria should be treated as a standalone fixed design choice but rather optimized for each application scenario.

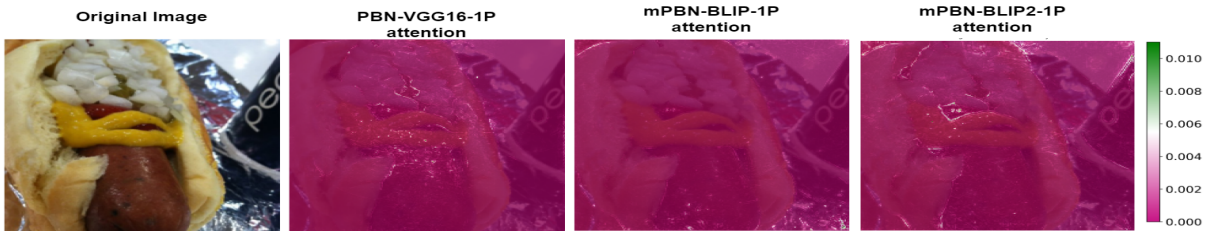


Fig. 1: Attention visualization of Case 1. True label: hotdog. BLIP caption: "a dish sitting in an aluminum foil container with some conal on it"; BLIP2 caption: "dish with mustard and onions on a bun."PBN-VGG16-1P prediction: breakfast burrito; mPBN-BLIP-1P and mPBN-BLIP2-1P prediction: hotdog

Case 1: mPBN-1P-BLIP and mPBN-BLIP-1P outperforms PBN-VGG16-1P This example shows an image labeled as *hot dog*, where both mPBN-BLIP-1P and mPBN-BLIP2-1P made correct predictions, while PBN-VGG16-1P incorrectly predicted it as *breakfast burrito*. The BLIP-generated caption is "a dish sitting in an aluminum foil container with some conal on it", while BLIP2 caption is "dish with mustard and onions on a bun."

From the attention visualizations, it is evident that all three models focus on different parts of the image, suggesting they rely on different prototype activations for the hot dog class. Interestingly, mPBN-BLIP-1P appears to attend more to the foil wrapping rather than the hot dog itself, indicating that the model may have associated the concept of a hot dog with the aluminum foil described in the caption. In contrast, mPBN-BLIP2-1P directs its attention to the mustard and onions, which aligns more directly with the semantic content of its caption.

Case 2: mPBN-2P-BLIP and mPBN-2P-BLIP2 outperform PBN-VGG16-2P This example shows an image labeled as pizza, where both mPBN-BLIP-2P and mPBN-BLIP2-2P correctly predicted the class, while PBN-VGG16-2P misclassified it as *grilled salmon*. The caption generated by BLIP is "a dish box with an uncoded slice of dish in it and two forks", whereas BLIP2 is "a slice of dish sitting in a cardboard box." Notably, the mention of “two forks” in the BLIP caption is a generated noise. The attention visualizations reveal that both mPBN models directed attention primarily to the shape and content of the pizza

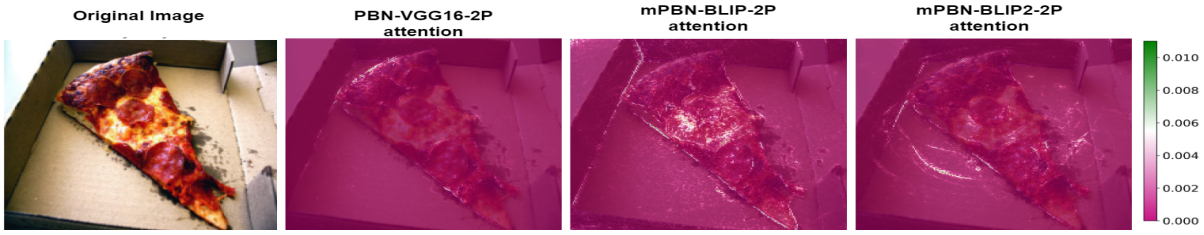


Fig. 2: Attention visualization of Case 2. True label: pizza. BLIP caption: "a dish box with an uncoded slice of dish in it and two forks"; BLIP2 caption: "a slice of dish sitting in a cardboard box." PBN-VGG16-2P prediction: grilled salmon; mPBN-BLIP-2P and mPBN-BLIP2-2P prediction: pizza

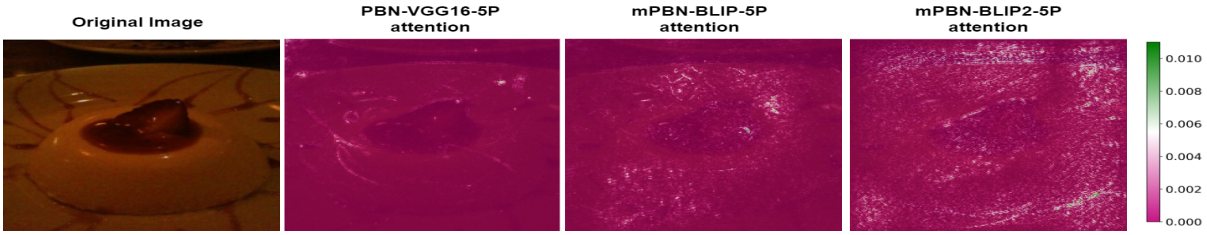


Fig. 3: Attention visualization of Case 3. True label: panna cotta. BLIP caption: "a plate that has some food on it and is lit up with candles in the background"; BLIP2 caption: "a dessert with a caramel sauce is on the plate". PBN-VGG16-5P prediction: panna cotta; mPBN-5P-BLIP prediction: scallops; mPBN-5P-BLIP2 prediction: chocolate mousse

slice, including its crust and pepperoni. This focus suggests that they learned discriminative features even under a low-prototype regime. Interestingly, the mPBN-BLIP2-2P model showed relatively strong attention on the cardboard box background as well, likely associating it with common pizza packaging. This indicates that the caption mentioning a “cardboard box” may have guided the model’s attention to a broader visual context that was still class-informative, enabling correct classification. In contrast, the PBN-VGG16-2P model exhibited low and poorly localized attention, failing to activate meaningful regions of the image.

Case 3: PBN-VGG16-5P outperforms mPBN-5P-BLIP and mPBN-5P-BLIP2 This example presents an image labeled as *panna cotta*, where only PBN-VGG16-5P produced the correct prediction. In contrast, mPBN-5P-BLIP misclassified it as *scallops*, and mPBN-5P-BLIP2 as *chocolate mousse*. The caption generated by BLIP reads: "a plate that has some food on it and is lit up with candles in the background", while BLIP2 states: "a dessert with a caramel sauce on the plate".

Unlike the previous two cases, the BLIP caption includes contextual noise, such as the mention of "candles in the background", which is unrelated to the object of interest. The attention visualizations suggest that mPBN-5P-BLIP and BLIP2 may have been influenced by such caption content. Specifically, mPBN-5P-BLIP focuses around the dessert but lacks sharp localization, and mPBN-5P-BLIP2 exhibits diffused attention across the entire image. In contrast, PBN-VGG16-5P centers its attention precisely on the shape and location of the dessert, which may have contributed to its accurate prediction.

This case raises the possibility that multimodal models, while potentially more expressive, are also more susceptible to misleading cues from noisy or irrelevant caption tokens. Further investigation is needed to determine which textual elements may have influenced these models' decision-making.

Case: 4: mPBN-5P-BLIP outperforms mPBN-5P-BLIP2 This example shows an image labeled as *panna cotta*, where mPBN-5P-BLIP made a correct prediction, and mPBN-5P-BLIP incorrectly predicted it as *chocolate mousse*. The caption generated by BLIP is "a plate with some desserts on it and some orange slices on top of the plate", whereas BLIP2 is "a plate with white pudding and orange slices on it". The captions generated by BLIP and BLIP2 are semantically similar, both referencing a white dessert and orange slices. The attention maps also show highly overlapping activation around the dessert and the plate, with no clear visual distinction between the two models' focus. This suggests that the difference in classification outcome may not stem directly from the captions or visual attention mechanisms, but rather from subtle factors in the learned representations or prototype associations. This case highlights the limitation of interpretability when attention visualizations do not yield clearly differentiable insights. The attention visualization is attached in figure 9 in Section 3.3.

Case 5: mPBN-2P-BLIP2 outperforms mPBN-2P-BLIP This example shows an image labeled as *lobster bisque*, where mPBN-2P-BLIP2 correctly predicted the class, while mPBN-2P-BLIP misclassified it as *chicken curry*. The caption generated by BLIP is: "a white bowl filled with soup next to bread and vegetables on a yellow napkin", while BLIP2 produces a more concise description: "a bowl of soup on a table". The attention visualization is provided in figure 10 in Secction 8.5.

From the attention visualizations, it can be observed that mPBN-2P-BLIP2 places strong emphasis within the bowl, capturing the soup's texture effectively. In contrast, mPBN-2P-BLIP appears to miss key visual cues, despite having a more descriptive caption. This suggests that the concise and visually grounded caption from BLIP2 may have guided the model to attend to more discriminative features of lobster bisque, resulting in a correct prediction. This case illustrates that more detailed descriptions do not necessarily improve model performance, and in some cases, may introduce distracting or irrelevant elements that mislead the model's attention.

Case 6: mPBN-1P-BLIP outperforms mPBN-2P-BLIP and mPBN-5P-BLIP This case is divided into 6-1 and 6-2. Attention visualization is provided in figure 11 in Section 8.5. Case 6-1 shows an image labeled as *chicken quesadilla*, where mPBN-1P-BLIP made a correct prediction, while mPBN-2P-BLIP incorrectly predicted it as *breakfast burrito*. The caption generated by BLIP is: "two burritoos and some salsa on a plate at a restaurant in los". Despite the presence of a spelling error ("burritoos") and noisy context ("at a restaurant in los"), mPBN-1P-BLIP successfully focused its attention on the actual food item in the center of the plate, aligning with the correct class. In contrast, the attention of mPBN-2P-BLIP appears to spread more broadly across the plate, including the side salsa and background, potentially diluting the core discriminative features. This suggests that increasing the number of prototypes may shift attention toward a wider set of visual cues, which can be harmful when the caption introduces ambiguity or irrelevant content. Additionally, in future work, it is advised to investigate whether the misspelled "burritoos" drive decisions - if it does, then both models might fail the same way, but this is not the observed case.

Case 6-2 presents an image labeled as *baklava*, where mPBN-1P-BLIP correctly identified the food item, while mPBN-5P-BLIP misclassified it as *spring roll*. The BLIP caption is: "two pieces of meat pie on a white plate with some sauce on top of the sandwich". Despite the caption containing irrelevant and inconsistent elements (e.g., "meat pie" and "sandwich"), mPBN-1P-BLIP maintained concentrated attention on the key pastry region of the image. In contrast, mPBN-5P-BLIP exhibited more dispersed attention across the plate, which may have caused it to associate the image with visually similar yet incorrect classes. This case demonstrates that a higher prototype count, in combination with noisy captions, can sometimes hinder precise attention and lead to semantically misleading decisions.

6 Discussion

This study explored the effectiveness of integrating natural-language captions into a Prototype-Based Network (PBN) architecture, forming a novel multimodal framework (mPBN) that jointly leverages visual and textual modalities. It contributes valuable insights into how generated natural-language captions can affect the behavior of prototype-based classifiers. Compared to prior multimodal prototypical networks by Pahde et al.[21] and Wan et al.[33], this study extends the investigation by incorporating VisualBERT as a backbone encoder. In contrast to GAN-based or part-annotated approaches, the proposed mPBN framework operates on free-form, generated captions without requiring human-annotated semantic parts or manually constructed visual prompts. Additionally, this study provides counterevidence that, while multimodal input can enhance model performance, its effectiveness is highly dependent on the linguistic style of the captions and the model's configuration, both of which can, in some cases, lead to degraded performance.

The results demonstrate that mPBNs can effectively incorporate natural-language descriptions and, when applying fewer prototypes per class (1 and 2, in this study), even outperform purely visual models. This suggests that language-based contextual information can compensate for limited visual prototype diversity. Conversely, as the number of prototypes increases, the benefits of textual input tend to diminish—and may even hinder performance—depending on the quality, specificity, and alignment of the generated captions. The case-based analyses further underscore how differences in captioning style and content can either enhance or mislead model attention, depending on the scenario.

These findings pave the way for future research on multimodal interpretability, controllable captioning, and robust architectural design in weakly supervised settings. Importantly, since this study relies on generated rather than human-annotated captions, the proposed mPBN framework presents a scalable approach for multimodal tasks where (1) textual supervision is scarce or costly, or (2) the task demands low tolerance for annotation errors. Potential applications include visual question answering, weakly supervised image segmentation, or multimodal retrieval systems.

6.1 Limitations and Future Work

Impaired PBN-ResNet34-1P One notable limitation was the training failure of the PBN-ResNet34-1P model, even under significantly reduced learning rates. While other ResNet-based models performed adequately, the single-prototype configuration exhibited persistent instability. Future work is encouraged to investigate the learning dynamics of ResNet-based PBNs in low-prototype settings, potentially by modifying the feature encoder or introducing stronger regularization mechanisms.

Exact Prototype Visualization for mPBN Although DeepSHAP was employed to visualize attention maps as a proxy of the original ProtoPNet's prototype visualization method [5] and interpret model decisions from a broader view of how captions shift the models'attention, its primary design projects importance scores onto the image space alone. With the introduction of mPBN, prototype representations now lie in a shared visual-language embedding space (VisualBERT), increasing model complexity and making direct projection of learned prototypes onto either modality non-trivial. This complexity hindered the ability to visualize precise prototype activations, particularly in the textual domain. Developing tools for multimodal prototype visualization—capable of attributing importance to both image regions and caption tokens—remains an open challenge. Such tools would not only improve transparency but would also help unravel the exact interactions of visual and textual components during decision making, especially in cases where the impact of a *specific* text token needs to be investigated.

Degree of Linguistic Noise on Model Performance and Caption Quality Control A further challenge stemmed from the linguistic noise present in generated

captions. As demonstrated in the case-based visualizations, mPBN performance varied depending on caption informativeness and clarity. Certain captions contained misspellings, irrelevant phrases, or ambiguous descriptions that may have misdirected the attention of multimodal models, which was inevitable upon completion of this study when using pre-trained text generators. Moreover, as demonstrated in Section 5.3, the influence of such noise was *not monotonic* — some models remained robust despite imperfect input, while others exhibited clear degradation. These observations point to the need for future research to systematically evaluate and/or control caption quality. This includes (1) defining a golden standard of mPBN-usable caption that would *not* negatively impact the model performance (2) investigating prompt engineering strategies exquisitely for generating such captions, (3) improving interpretability tools for the novel mPBN in this study to pinpoint specific linguistic influences on model behavior, and (4) consider including human judgements to score the caption quality, and dividing them into more subtle categories to further investigate their impacts on mPBN.

7 Conclusion

This study proposed and evaluated a novel Multimodal Prototype-Based Network (mPBN) that extends classical Prototype-Based Networks by incorporating generated natural-language captions using VisualBERT as a joint visual-text encoder. Through systematic experiments, the study demonstrated that mPBNs can outperform their unimodal counterparts, particularly when trained with fewer prototypes per class, by leveraging contextual cues from language, but can also underperform their unimodal counterparts when gradually increasing the number of prototypes. The case-based analysis further revealed that the impact of generated, natural-language captions is not monotonic: both captioning style and prototype configuration can mutually influence model effectiveness in different ways. In some cases, linguistic noise or semantic misalignment between image and caption led to performance degradation, highlighting the need for more precise control and interpretability. Furthermore, a golden-standard of mPBN-usable caption that would not negatively impact the model performance.

This study also identified critical challenges—including instability in specific model configurations (PBN-ResNet34-1P), limitations in visualizing joint prototype activations, and the unpredictable influence of caption noise—which require further investigation.

Overall, this work contributes to a deeper understanding of multimodal reasoning in prototype-based classifiers, offering a new direction for interpretable and data-efficient multimodal learning.

References

1. Ahmed, S., Nobel, S., Ullah, O.: An effective deep cnn model for multiclass brain tumor detection using mri images and shap explainability (04 2023). https://doi.org/10.1109/ECCE57851.2023.10101503
2. Bavaresco, A., Fernández, R.: Experiential semantic information and brain alignment: Are multimodal models better than language models? arXiv (2025), https://arxiv.org/abs/2504.00942
3. Bossard, L., Guillaumin, M., Gool, L.V.: Food-101 – mining discriminative components with random forests. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 446–461. Springer, Zurich, Switzerland (2014)
4. Chalmers, D.J.: Propositional interpretability in artificial intelligence. ArXiv (2025), https://arxiv.org/abs/2501.15740
5. Chen, C., Li, O., Tao, A., Barnett, A., Su, J., Rudin, C.: This looks like that: Deep learning for interpretable image recognition. arXiv preprint arXiv:1806.10574 (2018), https://arxiv.org/abs/1806.10574
6. Das, A., Gupta, C., Kovatchev, V., Lease, M., Li, J.J.: ProtoTEx: Explaining model decisions with prototype tensors. In: Muresan, S., Nakov, P., Villavicencio, A. (eds.) Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 2986–2997. Association for Computational Linguistics, Dublin, Ireland (May 2022). https://doi.org/10.18653/v1/2022.acl-long.213, https://aclanthology.org/2022.acl-long.213/
7. Filbrandt, G., Kamnitsas, K., Bernstein, D., Taylor, A., Glocker, B.: Learning from partially overlapping labels: Image segmentation under annotation shift (2021), https://arxiv.org/abs/2107.05938
8. Hase, P., Chen, C., Li, O., Rudin, C.: Interpretable image recognition with hierarchical prototypes. In: Proceedings of the AAAI Conference on Human Computation and Crowdsourcing. vol. 7, pp. 32–40 (2019). https://doi.org/10.1609/hcomp.v7i1.5265, https://doi.org/10.1609/hcomp.v7i1.5265
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition (2015), https://arxiv.org/abs/1512.03385
10. Hoffmann, A., Fanconi, C., Rade, R., Kohler, J.: This looks like that... does it? shortcomings of latent space prototype interpretability in deep networks. arXiv preprint arXiv:2105.02968 (2021), https://arxiv.org/abs/2105.02968
11. Huang, Q., Xue, M., Huang, W., Zhang, H., Song, J., Jing, Y., Song, M.: Evaluation and improvement of interpretability for self-explainable part-prototype networks. arXiv preprint arXiv:2212.05946 (2022), https://arxiv.org/abs/2212.05946
12. Jepkoech, J., Mugo, D.M., Kenduiywo, B.K., Too, E.C.: The effect of adaptive learning rate on the accuracy of neural networks. International Journal of Advanced Computer Science and Applications **12**(8) (2021). https://doi.org/10.14569/IJACSA.2021.0120885, http://dx.doi.org/10.14569/IJACSA.2021.0120885
13. Ji, Z., Zou, X., Liu, X., Huang, T., Mi, Y., Wu, S.: Neural information processing in hierarchical prototypical networks. In: Cheng, L., Leung, A.C.S., Ozawa, S. (eds.) Neural Information Processing. pp. 603–611. Springer International Publishing, Cham (2018)
14. Li, J., Zhang, C., Zhou, J.T., Fu, H., Xia, S., Hu, Q.: Deep-lift: Deep label-specific feature learning for image annotation. IEEE Transactions on Cybernetics **52**(8), 7732–7741 (2022). https://doi.org/10.1109/TCYB.2021.3049630

15. Li, J., Li, D., Savarese, S., Hoi, S.: BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In: Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., Scarlett, J. (eds.) Proceedings of the 40th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 202, pp. 19730–19742. PMLR (23–29 Jul 2023), https://proceedings.mlr.press/v202/li23q.html
16. Li, J., Li, D., Xiong, C., Hoi, S.: Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: Proceedings of the 39th International Conference on Machine Learning (ICML) (2022), https://arxiv.org/abs/2201.12086
17. Li, L.H., Yatskar, M., Yin, D., Hsieh, C.J., Chang, K.W.: Visualbert: A simple and performant baseline for vision and language. arXiv preprint arXiv:1908.03557 (2019), https://arxiv.org/abs/1908.03557
18. Li, Z.: Extracting spatial effects from machine learning model using local interpretation method: An example of shap and xgboost. Computers Environment and Urban Systems **96**, 101845 (06 2022). https://doi.org/10.1016/j.compenvurbsys.2022.101845
19. Liu, M., Ning, Y., Yuan, H., Ong, M.E.H., Liu, N.: Balanced background and explanation data are needed in explaining deep learning models with shap: An empirical study on clinical decision making (2022), https://arxiv.org/abs/2206.04050
20. Lundberg, S., Lee, S.I.: A unified approach to interpreting model predictions (2017), https://arxiv.org/abs/1705.07874
21. Pahde, F., Puscas, M., Klein, T., Nabi, M.: Multimodal prototypical networks for few-shot learning. arXiv preprint arXiv:2011.08899 (2020), https://arxiv.org/abs/2011.08899
22. Qian, J., Wu, Y., Zhuang, B., Wang, S., Xiao, J.: Understanding gradient clipping in incremental gradient methods. In: Banerjee, A., Fukumizu, K. (eds.) Proceedings of The 24th International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 130, pp. 1504–1512. PMLR (13–15 Apr 2021), https://proceedings.mlr.press/v130/qian21a.html
23. Raina, V., Liusie, A., Gales, M.: Is llm-as-a-judge robust? investigating universal adversarial attacks on zero-shot llm assessment. ArXiv (2024), https://arxiv.org/abs/2402.14016
24. Rathore, P.S., Dadich, N., Jha, A., Pradhan, D.: Effect of learning rate on neural network and convolutional neural network. International Journal of Engineering Research & Technology (IJERT) **6**(17), 1–5 (2019). https://doi.org/10.17577/IJERTCONV6IS17007, https://www.ijert.org/effect-of-learning-rate-on-neural-network-and-convolutional-neural-network, v-Impact – 2018 (Volume 06 – Issue 17)
25. Saralajew, S., Rana, A., Villmann, T., Shaker, A.: A robust prototype-based network with interpretable rbf classifier foundations (2025), https://arxiv.org/abs/2412.15499
26. Snell, J., Swersky, K., Zemel, R.S.: Prototypical networks for few-shot learning. ArXiv (2017), https://arxiv.org/abs/1703.05175
27. Sourati, Z., Deshpande, D., Ilievski, F., Gashteovski, K., Saralajew, S.: Robust text classification: Analyzing prototype-based networks. ArXiv (2023), https://arxiv.org/abs/2311.06647
28. Sporns, O.: Brain networks and cognitive architectures. Neuron **88**(1), 166–175 (2015). https://doi.org/10.1016/j.neuron.2015.09.027

29. Stadlhofer, A., Mezhuyev, V.: Approach to provide interpretability in machine learning models for image classification. Industrial Artificial Intelligence **1** (08 2023). https://doi.org/10.1007/s44244-023-00009-z
30. Storrs, K.R., Kietzmann, T.C., Walther, A., Mehrer, J., Kriegeskorte, N.: Diverse deep neural networks all predict human inferior temporal cortex well, after training and fitting. Journal of Cognitive Neuroscience **33**(10), 2044–2064 (09 2021). https://doi.org/10.1162/jocn_a_01755, https://doi.org/10.1162/jocn_a_01755
31. Tammina, S.: Transfer learning using vgg-16 with deep convolutional neural network for classifying images. vol. 9, p. p9420 (10 2019). https://doi.org/10.29322/IJSRP.9.10.2019.p9420
32. Vigliocco, G.: Language processing: The anatomy of meaning and syntax. Current Biology **10**(2), R78–R80 (2000). https://doi.org/https://doi.org/10.1016/S0960-9822(00)00282-7, https://www.sciencedirect.com/science/article/pii/S0960982200002827
33. Wan, Q., Wang, R., Chen, X.: Interpretable object recognition by semantic prototype analysis. pp. 789–798 (01 2024). https://doi.org/10.1109/WACV57701.2024.00085
34. Winter, E.: Chapter 53 the shapley value. Handbook of Game Theory with Economic Applications, vol. 3, pp. 2025–2054. Elsevier (2002). https://doi.org/https://doi.org/10.1016/S1574-0005(02)03016-3, https://www.sciencedirect.com/science/article/pii/S1574000502030163
35. Xu, X., Kong, K., Liu, N., Cui, L., Wang, D., Zhang, J., Kankanhalli, M.: An llm can fool itself: A prompt-based adversarial attack. ArXiv (2023), https://arxiv.org/abs/2310.13345
36. Yuan, H., Liu, M., Kang, L., Miao, C., Wu, Y.: An empirical study of the effect of background data size on the stability of shapley additive explanations (shap) for deep learning models (2023), https://arxiv.org/abs/2204.11351
37. Zhou, Y., Booth, S., Ribeiro, M.T., Shah, J.: Do feature attribution methods correctly attribute features? (2021), https://arxiv.org/abs/2104.14403

8 Appendix

8.1 Learning Rate Specification

This section summarizes the exact learning rate values used for each model in this study, addressing the Section 4.2, Table **??** notes the exact value of each learning rate type, and table 2 summarizes the type of learning rates each model configurated.

Regular

Stage	Feat. Ext.	Proto Vec	Add-on
Warmup	NA	3.00E-03	3.00E-03
Joint	1.00E-04	3.00E-03	3.00E-03
Last	1.00E-04	–	–

Low

Stage	Feat. Ext.	Proto Vec	Add-on
Warmup	NA	1.00E-03	1.00E-03
Joint	5.00E-05	1.00E-03	1.00E-03
Last	1.00E-04	–	–

Very Low

Stage	Feat. Ext.	Proto Vec	Add-on
Warmup	NA	1.00E-03	1.00E-03
Joint	1.00E-05	5.00E-04	5.00E-04
Last	1.00E-04	–	–

Extreme Low 1

Stage	Feat. Ext.	Proto Vec	Add-on
Warmup	NA	1.00E-03	1.00E-03
Joint	1.00E-06	1.00E-05	1.00E-05
Last	1.00E-04	–	–

Extreme Low 2

Stage	Feat. Ext.	Proto Vec	Add-on
Warmup	NA	1.00E-04	1.00E-04
Joint	1.00E-07	1.00E-06	1.00E-06
Last	1.00E-04	–	–

High

Stage	Feat. Ext.	Proto Vec	Add-on
Warmup	NA	3.00E-02	3.00E-02
Joint	1.00E-03	3.00E-02	3.00E-02
Last	1.00E-04	–	–

For Non-prototypical Baseline Models

Type	Learning Rate
High	1.00E-03
Regular	1.00E-04
Low	1.00E-05

Table 1: Explanation of Learning Rate Type in table 2. Learning rate schedules applied to prototypical models during training. Each setting (e.g., Regular, Low, Extreme Low) specifies the learning rates used in different training phases: warmup, joint training, and add-on layer. The schedule separately controls the feature extractor (Feat. Ext.), prototype vectors (Proto Vec), and additional components (Add-on). Lower schedules (e.g., "Extreme Low") were introduced to stabilize training for sensitive models such as PBN-ResNet34-1P. Baseline learning rates are summarized for comparison.

Model Name	Learning Rate Type	Model Name	Learning Rate Type
Baseline VGG16	Regular	Baseline ResNet34	High
PBN-VGG16-1P	Regular	PBN-VGG16-5P	Regular
PBN-VGG16-2P	Regular	PBN-VGG16-10P	Regular
PBN-ResNet34-1P	Extreme Low 2	PBN-ResNet34-5P	Regular
PBN-ResNet34-2P	Regular	PBN-ResNet34-10P	Regular
mPBN-1P-BLIP	Low	mPBN-5P-BLIP	Low
mPBN-2P-BLIP	Regular	mPBN-10P-BLIP	Low
mPBN-1P-BLIP2	Regular	mPBN-5P-BLIP2	Regular
mPBN-2P-BLIP2	Low	mPBN-10P-BLIP2	Regular

Table 2: Learning rate configuration for all model variants. *Baseline* refers to non-prototypical CNN models. *PBN* refers to unimodal Prototype-Based Networks, and *mPBN* to multimodal variants. The suffix "1P", "2P", etc., denotes the number of prototypes per class. The suffix "BLIP" and "BLIP2" denote the caption style that the model is trained with. *Regular* corresponds to the learning rate from the original ProtoPNet paper [5], while *High*, *Low*, and *Extreme Low* are reduced values derived empirically through fine-tuning.

8.2 PBN and mPBN Architecture Illustrations

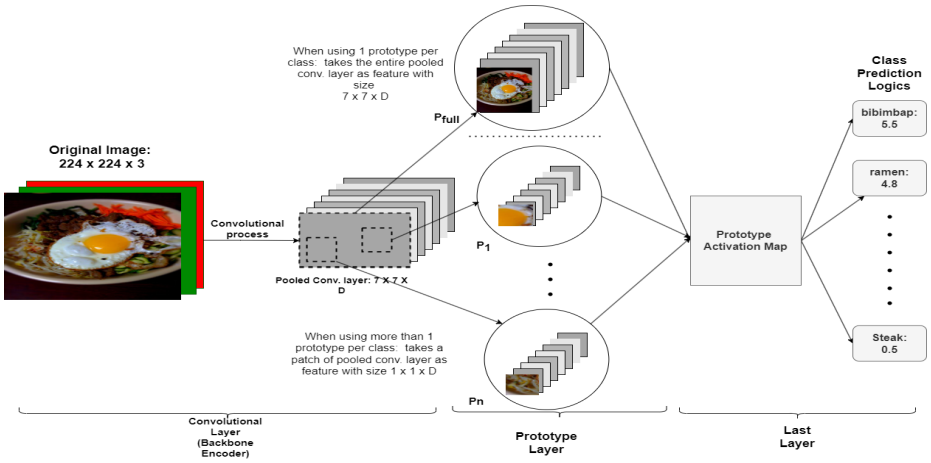


Fig. 4: Illustration of the Unimodal Prototype-Based Network (PBN) architecture based on ProtoPNet [5]. The input image is passed through a convolutional encoder to produce a pooled latent feature map. Depending on the number of prototype choices, either the full feature map ($7 \times 7 \times D$) or smaller patches ($1 \times 1 \times D$) (see Section 4.2 for details) are selected to form prototypes. These learned prototypes are compared to the input activation map using inverted ℓ_2 distance to generate similarity scores. The activation map aggregates these distances, and the model assigns class logits based on which class-specific prototypes are most similar to the input. Prototype reasoning enables case-based interpretability by aligning class decisions with learned prototype regions.

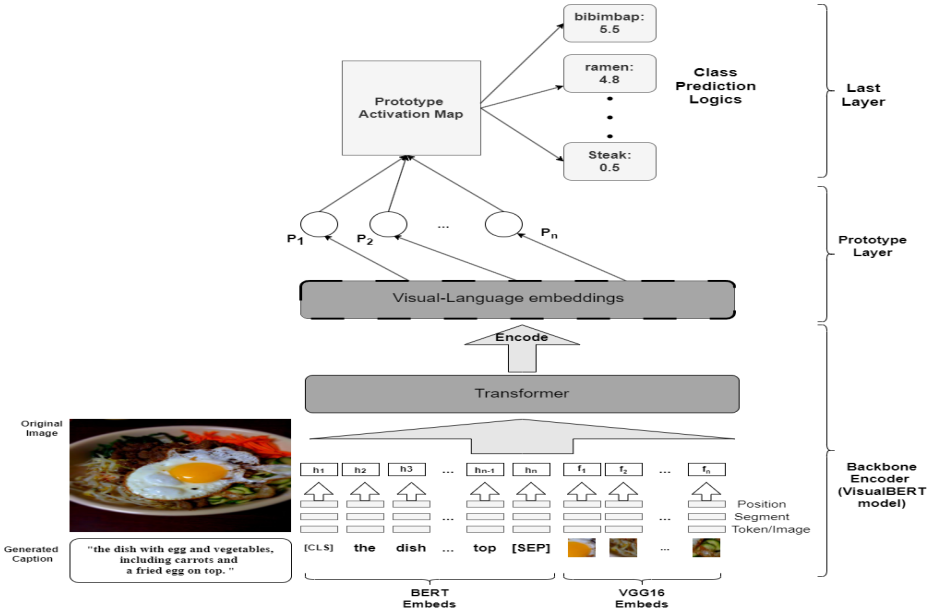


Fig. 5: Illustration of the Multimodal Prototype-Based Network (mPBN) architecture. The input image and its generated caption are embedded using VGG16 and a BERT tokenizer, respectively. These are concatenated and processed by a pre-trained VisualBERT Transformer to produce a joint visual-language representation. The [CLS] token output serves as the semantic embedding compared against learned prototypes in the prototype layer. Class logits are computed based on the similarity scores between the [CLS] embedding and the prototypes.

8.3 Overall Model Performance summary

Model	Test Accuracy (%)	Evaluation Loss
Baseline VGG16	55.37	2.21
Baseline ResNet34	47.04	3.15
PBN-VGG16 1P	31.95	2.82
PBN-VGG16 2P	65.64	1.50
PBN-VGG16 5P	66.78	1.48
PBN-VGG16 10P	66.00	1.43
PBN-ResNet34 1P	26.92	4.62
PBN-ResNet34 2P	69.74	1.55
PBN-ResNet34 5P	66.89	2.06
PBN-ResNet34 10P	69.58	1.54
mPBN-BLIP 1P	72.65	1.72
mPBN-BLIP 2P	70.39	1.78
mPBN-BLIP 5P	46.87	4.21
mPBN-BLIP 10P	37.42	3.37
mPBN-BLIP2 1P	77.34	1.37
mPBN-BLIP2 2P	76.87	1.55
mPBN-BLIP2 5P	31.80	2.82
mPBN-BLIP2 10P	41.75	2.79

Table 3: Test accuracy and evaluation loss of all baseline, unimodal PBN, and multimodal mPBN models trained under different prototype counts and learning rate settings. Each model variant is evaluated on a held-out test set. PBNs generally benefit from increasing the number of prototypes, while mPBNs perform better with fewer prototypes (1–2 per class). The best-performing model is mPBN-BLIP2-1P with 77.34% accuracy, while the worst is PBN-ResNet34-1P with 26.92% accuracy and high evaluation loss.

8.4 Model-Exclusive True Positive Count Figures

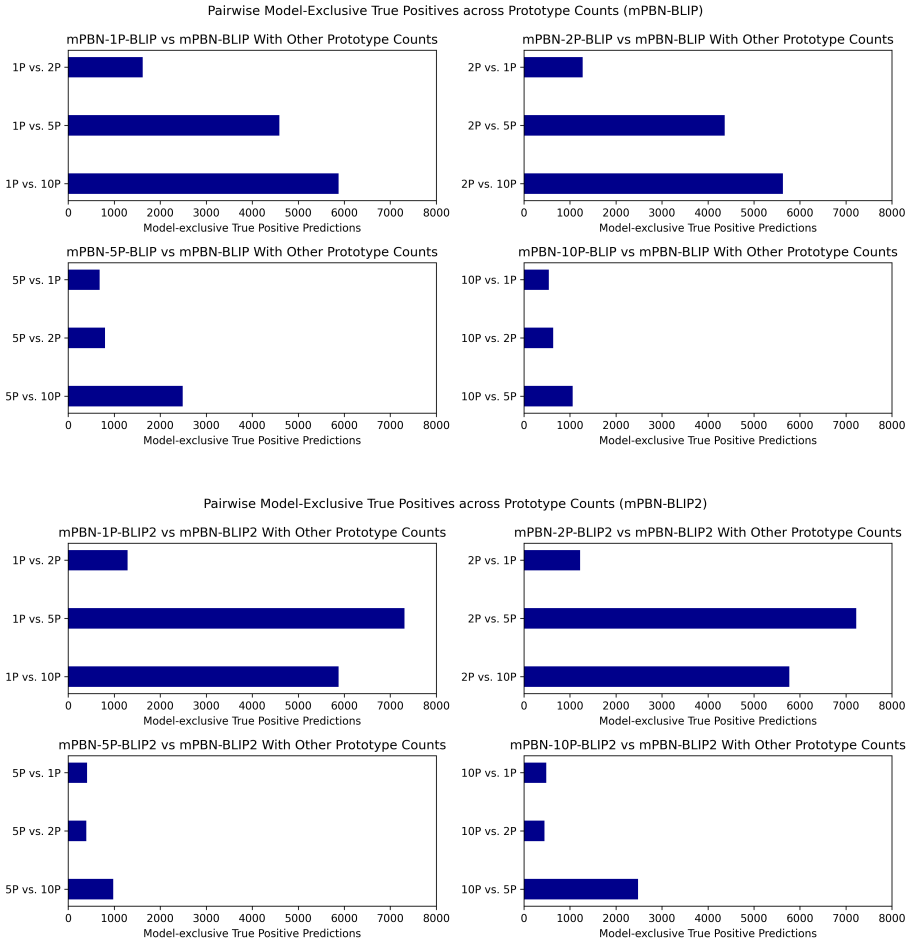
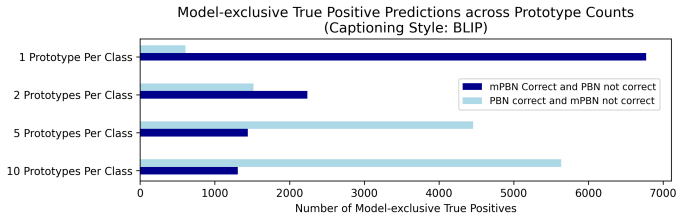
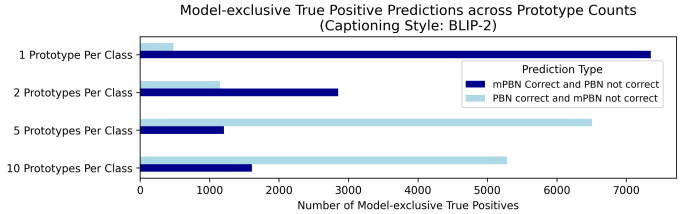


Fig. 6: Pairwise model-exclusive true positive predictions across different prototype counts within each captioning style. Each subplot compares a pair of mPBN models with the same captioning style (BLIP or BLIP2) but different prototype counts. Bars indicate the number of test samples that were correctly classified by one model and not the other, revealing how prototype granularity affects model behavior under fixed language conditions. The figure shows that for mPBNs, gradually decreasing the number of prototypes can improve the model performance to a certain degree.



(a) BLIP-based captioning result (mPBN-BLIP vs. PBN-VGG16)



(b) BLIP-2-based captioning result (mPBN-BLIP2 vs. PBN-VGG16)

Fig. 7: Model-exclusive true positive predictions across prototype counts for mPBN and PBN models, based on two captioning styles. Each bar shows the number of samples correctly classified by only one model (either mPBN or PBN) but not the other, reflecting model-specific strengths under different prototype configurations. For instance, the dark-blue bar in Figure (a) for "1 Prototype Per Class" indicates that nearly 6,900 test samples were correctly classified by mPBN-BLIP-1P but not by its PBN counterpart (PBN-VGG16-1P).

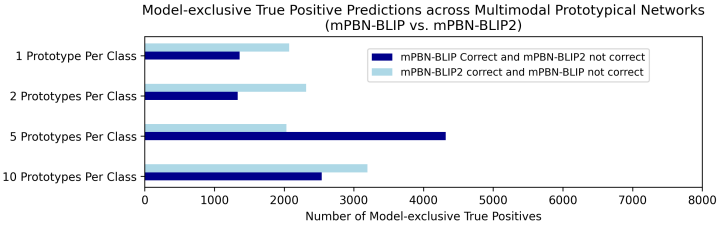


Fig. 8: Model-exclusive true positive predictions for mPBN-BLIP and mPBN-BLIP2 across different prototype counts. Each bar shows the number of samples correctly classified by only one model (either mPBN-BLIP or mPBN-BLIP2) but not the other. This comparison highlights that captioning style influences the predictive behavior of multimodal prototypical networks under different prototype configurations.

8.5 Attention Visualization Plots for Case 4, 5, and 6

This section illustrates the attention visualization of case 4, 5, and 6 (Section 5.3, 5.3, and 5.3) respectively.

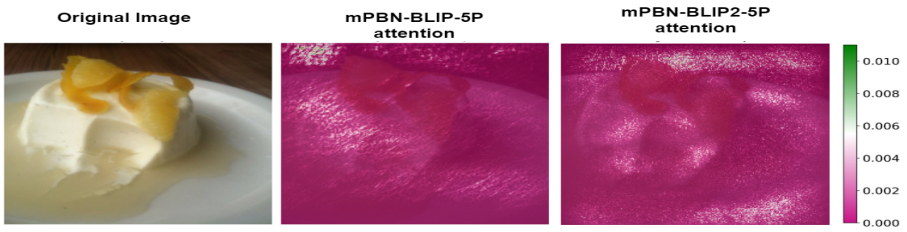


Fig. 9: Attention visualization of Case 4. True label: panna cotta. BLIP caption: "a plate with some desserts on it and some orange slices on top of the plate"; BLIP2 caption: "a plate with white pudding and orange slices on it". mPBN-5P-BLIP prediction: panna cotta; mPBN-5P-BLIP2 prediction: chocolate mousse

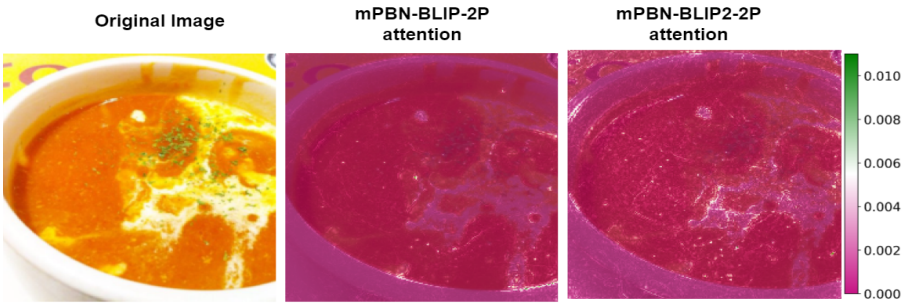
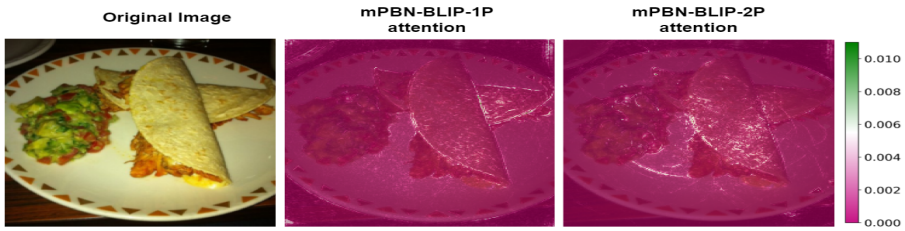
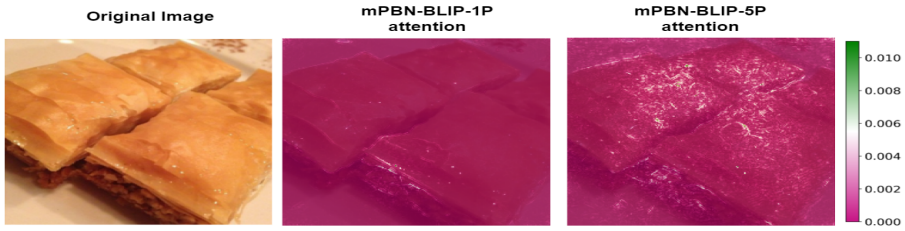


Fig. 10: Attention visualization of Case 5. True label: lobster bisque. BLIP caption: "a white bowl filled with soup next to bread and vegetables on a yellow napkin"; BLIP2 caption: "a bowl of soup on a table". mPBN-2P-BLIP prediction: lobster bisque; mPBN-2P-BLIP2 prediction: chicken curry



(a) Attention visualization of Case 6-1. True label: chicken quesadilla. BLIP caption: "two burritoos and some salsa on a plate at a restaurant in los". mPBN-1P-BLIP prediction: chicken quesadilla; mPBN-2P-BLIP prediction: breakfast burrito



(b) Attention visualization of Case 6-2. True label: baklava. BLIP caption: "two pieces of meat pie on a white plate with some sauce on top of the sandwich". mPBN-1P-BLIP prediction: baklava; mPBN-2P-BLIP prediction: spring roll

Fig. 11: Attention Visualization of Case 6