NeProV: Compositional Zero-Shot Learning with Negative Prompts and Progressive Visual Feature Disentanglement

Anonymous ACL submission

Abstract

Compositional Zero-Shot Learning (CZSL) targets the recognition of unseen attribute-object compositions by generalizing from seen ones. Existing prompt-based methods encode attributes, objects, and pairs only with positive prompts, limiting language supervision and thus compositional adaptability. In this paper, we propose NEPROV, a CLIP-based framework that introduces negative soft prompts for attributes and objects to provide complementary counter-evidence. Positive prompts are encouraged to respond to present concepts, whereas negative prompts capture evidence for their absence. On the visual side, we propose progressive visual feature disentanglement, which aggregates features across transformer layers and decomposes them into attribute-, object-, and composition-sensitive representations. We further introduce *logic-based constraints* to enforce consistent relationships between positive and negative predictions, together with *structured label smoothing* to reduce overfitting to seen compositions. Experiments on UT-Zappos, MIT-States, and C-GQA under closed-world and open-world settings show that NEPROV achieves state-of-the-art or competitive performance.

1 Introduction

Inspired by human compositional cognition, Compositional Zero-Shot Learning (CZSL) has become a central challenge in machine learning. CZSL aims to recognize novel compositions of familiar concepts, such as identifying *"ripe apple"* without having seen the specific combination during training. It requires two key abilities: (1) **generalization**, i.e., transferring knowledge to unseen attribute-object pairs; and (2) **relational reasoning**, i.e., understanding the semantic compatibility between attributes and objects (e.g., *"ripe"* is more likely to modify *"apple"* than *"rock"*).

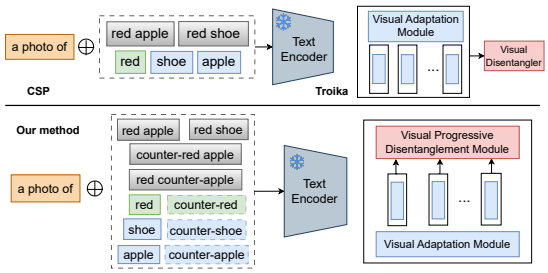


Figure 1: Comparison between our method and previous approaches. CSP (Nayak et al., 2023) constructs compositional soft prompts for attributes and objects, whereas our method further introduces negative prompts to provide additional discriminative supervision. Troika (Huang et al., 2024) performs visual adaptation before disentanglement, whereas our method enables progressive visual disentanglement to proceed jointly with visual adaptation.

Recent CZSL approaches can be broadly categorized into four paradigms: (i) *Compositional soft prompts and factorized classifiers*, including compositional soft prompting methods (Nayak et al., 2023) and classical factorized models that learn separate attribute and object classifiers (Nagarajan and Grauman, 2018); (ii) *Graph-based compositional reasoning*, which captures attribute–object dependencies through graph neural networks or leverages external knowledge to constrain and prune implausible compositions (Naeem et al., 2021); (iii) *Logic-guided semantic alignment*, which aligns visual and textual embeddings of attributes, objects, and compositions under explicit logical constraints (Wu et al., 2025); and (iv) *Clustering and cross-modal joint learning*, which learns composition-aware prototypes and jointly aligns visual and textual representations (Qu et al., 2025; Zhang et al., 2025).

These CZSL methods have two main limitations. First, the compositional concept space is substantially larger than the individual attribute and object spaces. As a result, attribute and object prompts can be prone to overfitting when trained only

through their respective branches, while lacking explicit textual concept supervision for attributes and objects themselves. Second, their visual disentanglement strategy is either applied only after adapter-based adaptation or confined to the deepest transformer layer, thereby failing to fully exploit informative signals distributed across the entire network depth.

To address these challenges, we introduce two key designs: *Negative Prompt Tuning* and *Progressive Visual Disentanglement*. The former complements positive prompts with negative soft prompts, enabling the model to learn both concept-presence and concept-absence signals, while the latter aggregates and separates visual features across transformer layers, allowing attribute-, object-, and composition-sensitive representations to evolve jointly with visual adaptation.

Building on these ideas, we propose NEPROV, a CLIP-based (Radford et al., 2021) framework for CZSL, as illustrated in Figure 1. Unlike CSP (Nayak et al., 2023), which constructs only positive soft prompts for attributes, objects, and their compositions, NEPROV further introduces negative prompts to provide richer discriminative supervision. Unlike Troika (Huang et al., 2024), which performs visual adaptation before disentanglement, NEPROV integrates progressive visual adaptation and disentanglement within a unified pipeline, enabling them to evolve jointly across transformer layers. On the textual side, NEPROV augments positive prompts with *negative soft prompts* for each attribute and object while keeping the CLIP text encoder frozen. On the visual side, a progressive disentanglement module aggregates *stage-wise* features from the CLIP image encoder and decomposes them into attribute-, object-, and composition-sensitive token embeddings. During training, NEPROV aligns visual and textual representations through *multi-level* contrastive objectives together with logic-based constraints; during inference, predictions from different branches are fused to obtain the final compositional prediction.

Our contributions are summarized as follows.

- **Negative soft prompting.** We introduce negative soft prompts for attributes and objects to complement conventional positive prompts. The positive and negative prompts are learned jointly under logic-based regularization, allowing NEPROV to capture both supporting evidence and counter-evidence for each concept.

- **Progressive visual adaptation and disentanglement.** We combine lightweight visual adapters with a progressive visual disentanglement module that aggregates stage-wise features across transformer depths and separates them into attribute-, object-, and composition-aware branches.

- **Structured label smoothing and logic-based constraints.** We propose a structured label smoothing scheme that redistributes part of the probability mass to semantically related compositions sharing the same attribute or object, thereby improving robustness and compositional generalization. We further introduce logic-based constraints to contrastively model positive and negative prompts.

2 Related Work

Compositional prompt learning. Prompt tuning has been widely explored to adapt pretrained vision-language models (VLMs) such as CLIP for CZSL. Methods like CSP (Nayak et al., 2023) and DFSP (Lu et al., 2023) introduce compositional soft prompts of attributes and objects to better align visual and textual spaces. However, these methods largely rely on *positive* prompts and lack mechanisms to explicitly *adapt* and *disentangle* attribute–object factors within the visual pathway, which limits transfer to unseen compositions.

Graph-based prompt integration. Another line of work augments textual embeddings with relational priors via GCNs. These approaches often assume static graphs or hand-crafted textual features, constraining adaptability and expressiveness. Hierarchical prompting (Geng et al., 2025; Liu et al., 2023) encodes structured relations among attributes and objects, while other methods like conditional prompting (Bao et al., 2025; Zhou et al., 2022; Huang et al., 2024) exploit linguistic or visual context to refine prompts. Although helpful on seen compositions, improvements on unseen pairs remain modest, indicating that semantic entanglement is insufficiently addressed.

Visual adapters for CZSL. Recent works insert lightweight adapters into the CLIP image encoder (Huang et al., 2024; Zheng et al., 2024) to enable task-specific tuning. While effective for adaptation, most adapter designs do not enforce explicit factorization of attribute and object cues; as a result, they can overfit seen combinations and

generalize poorly to novel pairs.

Clustering-, logic-, and proxy-based approaches. Prototype mining via clustering (Qu et al., 2025) builds composition-aware prototypes, logic-induced learning (Wu et al., 2025) injects symbolic constraints to regularize compositional reasoning, and visual proxy learning (Zhang et al., 2025) constructs proxy targets to ease supervision. These strategies enrich supervision from complementary angles, yet they do not systematically disentangle attribute and object representations or guarantee robustness on unseen compositions.

Summary. Across prompt-, graph-, adapter-, and auxiliary supervision paradigms, a recurring limitation is the lack of *explicit semantic disentanglement*. This limitation motivates NEPROV, which *progressively adapts and disentangles* visual representations while leveraging enriched supervision from negative prompts to learn more factorized features, thereby improving generalization to unseen attribute–object pairs.

3 Methodology

We present NEPROV, a CZSL framework that: i) augments the text encoder with *positive* and *negative* soft prompts; ii) adapts and disentangles visual features using visual adapters and a progressive visual disentanglement module (Visual PD); iii) trains the model with *structured label smoothing* over semantically related compositions and logic-based constraints; and iv) performs inference through weighted fusion of attribute, object, and composition predictions. By coupling attribute/object scores with composition cues, NEPROV improves generalization to both seen and unseen attribute-object pairs. The overall architecture is shown in Figure 2.

3.1 Text Encoding

Positive and negative soft prompts. Following CSP (Nayak et al., 2023), we learn prompt embeddings for *attributes* and *objects* in CLIP's text encoder. For both attributes and objects, each concept embedding c is initialized as the mean of the token embeddings associated with its name, excluding special tokens. The corresponding *negative* counterpart is then defined as the mean embedding of all remaining concepts in the same concept space.

$$\begin{aligned} \mathbf{e}_c^+ &= \frac{1}{|T_c|} \sum_{t \in T_c} \mathrm{TokEmb}(t), \\ \mathbf{e}_c^- &= \frac{1}{|\mathcal{C}(c)| - 1} \sum_{c' \in \mathcal{C}(c) \setminus \{c\}} \mathbf{e}_{c'}^+, \end{aligned} \tag{1}$$

where T_c denotes the name tokens of concept c (excluding special tokens); $\mathcal{C}(c)$ denotes the set of concepts in the same category as c, such as the attribute set or the object set.

For each concept space, namely attributes, objects, and compositions, we additionally construct a prefix embedding as shared context corresponding to the prompt phrase "`a photo of`":

$$\mathbf{E}_{\text{ctx}} = \mathrm{TokEmb}(\texttt{"a photo of"}).$$

To obtain branch-specific representations, we use the following prompt templates: (1) `[a photo of] [attribute]`, (2) `[a photo of] [object]`, (3) `[a photo of] [attribute] [object]`, (4) `[a photo of] [neg. attribute] [object]`, and (5) `[a photo of] [attribute] [neg. object]`. These branches are intended to disentangle attribute semantics, object semantics, and their compositional interactions.

Let $E_a, E_o, E_c \in \mathbb{R}^{n_{\text{ctx}} \times D}$ denote the learnable context token sequences for the attribute, object, and composition branches, respectively. The positive textual representations are defined as

$$\begin{aligned} \mathbf{z}_a^+ &= \mathrm{TextEncode}\big(E_a \oplus \mathbf{e}_a^+\big), \\ \mathbf{z}_o^+ &= \mathrm{TextEncode}\big(E_o \oplus \mathbf{e}_o^+\big), \\ \mathbf{z}_c^+ &= \mathrm{TextEncode}\big(E_c \oplus \mathbf{e}_a^+ \oplus \mathbf{e}_o^+\big). \end{aligned} \tag{2}$$

For *negative* encoding, we instantiate templates with the negative concept vectors (replacing $\mathbf{e}^+$ by $\mathbf{e}^-$) to obtain attribute-only and object-only negatives:

$$\begin{aligned} \mathbf{z}_a^- &= \mathrm{TextEncode}\big(E_a \oplus \mathbf{e}_a^-\big), \\ \mathbf{z}_o^- &= \mathrm{TextEncode}\big(E_o \oplus \mathbf{e}_o^-\big). \end{aligned} \tag{3}$$

Similarly, the negative compositional representations are constructed as

$$\begin{aligned} \mathbf{z}_c^{a-} &= \mathrm{TextEncode}\big(E_c \oplus \mathbf{e}_a^- \oplus \mathbf{e}_o^+\big), \\ \mathbf{z}_c^{o-} &= \mathrm{TextEncode}\big(E_c \oplus \mathbf{e}_a^+ \oplus \mathbf{e}_o^-\big). \end{aligned} \tag{4}$$

where $\oplus$ denotes token concatenation. During training, $\mathbf{z}_c^+$, $\mathbf{z}_c^{a-}$ and $\mathbf{z}_c^{o-}$ are instantiated only for seen attribute–object pairs.

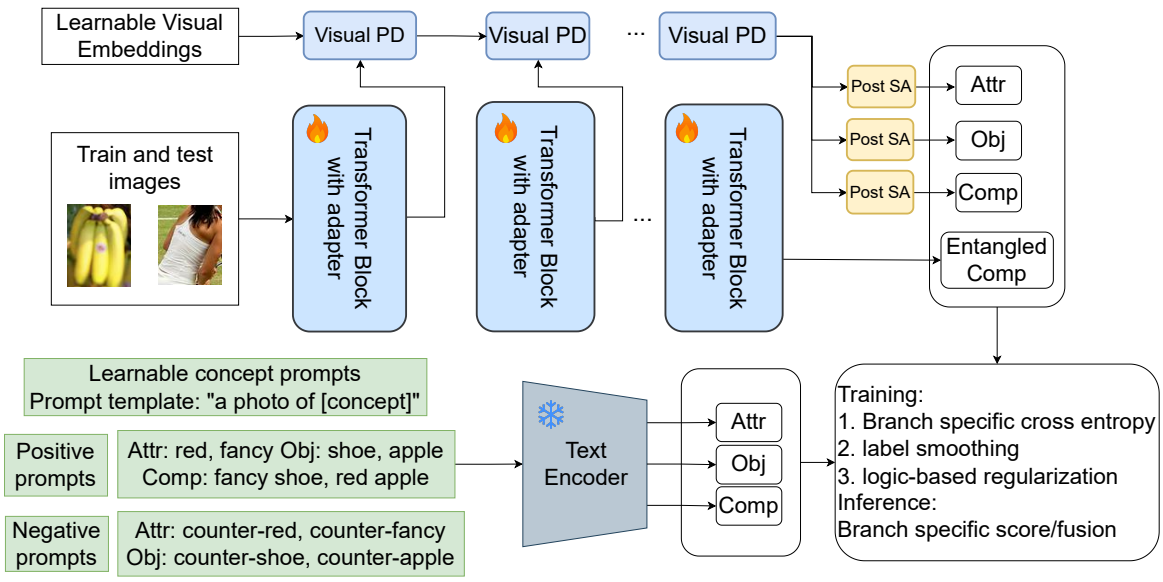


Figure 2: **Overall architecture of NEPROV.** (1) **Textual prompt encoding.** The prompt prefix "a photo of" is combined with learnable concept tokens and fed into the frozen CLIP text encoder to produce branch-specific textual representations, including composition-, attribute-, and object-level prompts, as well as their corresponding negative prompts. (2) **Progressive visual disentanglement (Visual PD).** Given training or test images, the CLIP visual encoder is augmented with adapters and learnable visual embeddings. Visual PD modules are inserted progressively across the transformer blocks to disentangle branch-specific visual features, and post-self-attention layers further refine the resulting branch representations. (3) **Multi-branch supervision and prediction.** The branch-specific visual and textual features are aligned to produce logits for different semantic branches. During training, the model is optimized with multi-branch cross-entropy loss and logic-based regularization. During inference, predictions from multiple branches are combined by weighted fusion.

3.2 Image Encoding, Adaptation and Disentanglement

Inject image encoder with adapters. Recent work (Geng et al., 2025; Zheng et al., 2024; Huang et al., 2024) introduces lightweight adapter blocks for CLIP/ViT backbones to enable parameter-efficient fine-tuning. Following this line, we insert two *bottleneck adapter* blocks into *each* transformer block of the CLIP vision encoder: (i) a residual **adapter** on the self-attention sublayer and (ii) a residual **adapter** on the feed-forward sublayer. Each adapter is a bottleneck MLP ($D \to d \to D$ with ReLU as activation, $d \ll D$); to mitigate overfitting, we apply dropout inside the bottleneck and on the adapter output before the residual merge. Figure 3 depicts the architecture of the bottleneck adapter.

Given a batch of images $\mathbf{X} \in \mathbb{R}^{B\times C\times H\times W}$, CLIP convolution produce latent visual embedding $\mathrm{V}_{\text{ent}}^{(0)}$, the adapted encoder produces token-level visual features

$$\mathrm{V}_{\text{ent}}^{(i)} = \mathrm{ImgEncode}_{\theta}(\mathrm{V}_{\text{ent}}^{(i-1)}) \in \mathbb{R}^{B\times L\times D}, \quad (5)$$

where L is the number of visual tokens (including a class token, if present) and D is the embedding width.

To capture depth-wise behavior, we leverage the intermediate token features after each vision transformer block,

$$\mathbf{V}_{\text{ent}}^{(i)} \in \mathbb{R}^{B\times L\times D}, \quad i = 1, \ldots, N,$$

where N is the number of blocks (e.g., N=24 for a ViT-L/14 CLIP backbone). These entangled features feed visual progressive disentanglement modules. The final image embedding is obtained from the class token of the last block, followed by a projection:

$$\mathrm{v}_{\text{ent}} = \mathrm{Proj}\left(\mathrm{CLS}(\mathrm{V}_{\text{ent}}^{(N)})\right) \in \mathbb{R}^{B\times d}, \quad (6)$$

where $\mathrm{CLS}(\cdot)$ selects the class token and $\mathrm{Proj} : \mathbb{R}^{D} \to \mathbb{R}^{d}$ maps to the downstream embedding width d.

Progressive visual feature disentanglement. We instantiate M learnable disentanglement tokens $\mathbf{V}_{\text{dis}}^{(0)} \in \mathbb{R}^{B\times M\times D}$ as a compact carrier of factorized visual information. At each transformer

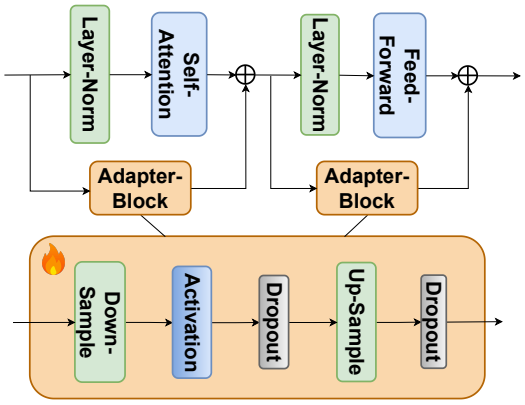


Figure 3: **Transformer block adapters in the CLIP vision encoder.** Two adapters are added per transformer block: an attention adapter on the self-attention path and a feed-forward adapter on the MLP path. Each adapter is a bottleneck MLP with dropout, enabling parameter-efficient fine-tuning across the entire depth of the backbone.

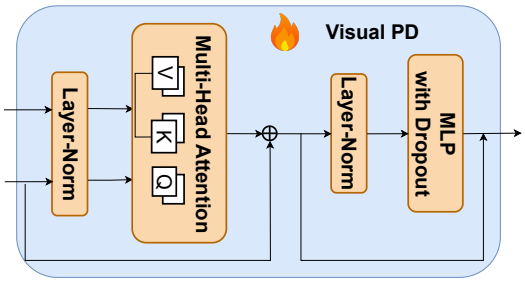


Figure 4: **Visual PD module.** At transformer block i, Visual PD updates disentangled tokens $\mathbf{V}_{\text{dis}}^{(i-1)}$ by attending to entangled visual tokens $\mathbf{V}_{\text{ent}}^{(i)}$, followed by a pre-normalized residual MLP. The module is shared across transformer layers to progressively aggregate depth-wise visual cues.

block i, Visual PD uses the previous disentangled tokens as queries and the current entangled tokens $\mathbf{V}_{\text{ent}}^{(i)} \in \mathbb{R}^{B\times L\times D}$ as visual evidence:

$$\mathbf{V}_{\text{dis}}^{(i)} = \text{PD}\left(\mathbf{V}_{\text{dis}}^{(i-1)}, \mathbf{V}_{\text{ent}}^{(i)}\right), \quad i = 1, \ldots, N. \tag{7}$$

By updating the same compact token set across depths, Visual PD integrates both low- and high-level cues while encouraging factorized representations for compositional reasoning.

Generating multi-branch visual representations. Given last layer of $\text{V}_{\text{dis}} \in \mathbb{R}^{B\times M\times D}$, we derive three role-specific representations—composition, attribute, and object—using parallel, parameter-disjoint branches, one transformer layer each:

$$\begin{aligned} \mathbf{V}_b &= \text{SA}_b(\text{LN}(\mathbf{V}_{\text{dis}})), \\ \mathbf{v}_b &= \text{Proj}_b(\text{LN}(\text{MeanPool}(\mathbf{V}_b))), b \in \{c, a, o\}, \end{aligned} \tag{8}$$

where $b \in \{c, a, o\}$ denotes the composition, attribute, and object branches, respectively. $\text{SA}_b(\cdot)$ is the branch-specific self-attention module, and $\text{Proj}_b(\cdot)$ maps the pooled representation into the corresponding branch embedding.

3.3 Training

The overall training pipeline of NEPROV is illustrated in Figure 2. It consists of multi-branch text and image encoding, together with modules for feature disentanglement and representation adaptation. In the following, we first describe how branch-specific prediction logits are computed and then present the corresponding training objectives.

Structured label smoothing for compositions. CZSL training data covers only a subset of attribute–object pairs. To curb overconfidence under label sparsity and imbalance, we apply *label smoothing* (Li et al., 2020) to the composition classifier, but redistribute probability mass only to *semantically related* classes.

Let C denote the number of composition classes, and let $\mathbf{y}_c^i \in \{0,1\}^C$ denote the one-hot label vector for the i-th composition class. For each class i, we define its relation set as

$$\mathcal{R}(i) = \{\, j \neq i \;:\; a_j = a_i \;\vee\; o_j = o_i \,\},$$

namely, the set of all other composition classes that share either the same attribute or the same object with class i, where a_i and o_i denote the attribute and object associated with class i, respectively.

We then construct a binary relation mask $\mathbf{m}^{(i)} \in \{0,1\}^C$ such that

$$m_j^{(i)} = \begin{cases} 1, & j \in \mathcal{R}(i), \\ 0, & \text{otherwise}, \end{cases} \qquad m_i^{(i)} = 0,$$

and let

$$K_i = \|\mathbf{m}^{(i)}\|_1$$

denote the number of related classes for class i. Given a smoothing coefficient $\varepsilon \in [0, 1)$, we define the structured smoothed target as

$$\tilde{\mathbf{y}}_c^i = (1-\varepsilon)\,\mathbf{y}_c^i \;+\; \varepsilon\,\frac{\mathbf{m}^{(i)}}{\max(1, K_i)}. \tag{9}$$

In this way, a probability mass of $1-\varepsilon$ is assigned to the ground-truth class, while the remaining mass

ε is distributed uniformly over all related classes that share the same attribute or object. If no such related class exists, i.e., $K_i = 0$, the second term becomes zero, and the target reduces to the original one-hot label.

Compositional classification logits. For each image i, we separately compare the entangled visual feature v_{ent} and the composition branch feature v_c with compositional text features $\mathrm{z}_c \in \mathbb{R}^{K\times d}$ for K training attribute–object pairs. Each comparison produces a vector of CLIP-style logits via temperature-scaled cosine similarity. The CLIP-style logits are given by

$$\begin{aligned} \text{logits}_e &= \tau\,(\mathrm{v}_{\text{ent}}\,\mathrm{z}_c^{+\top}) \in \mathbb{R}^K, \\ \text{logits}_c &= \tau\,(\mathrm{v}_c\,\mathrm{z}_c^{+\top}) \in \mathbb{R}^K, \end{aligned} \tag{10}$$

where τ denotes the learnable logit scale. We then apply a cross-entropy loss with smoothed labels to encourage accurate compositional prediction.

$$\begin{aligned} \mathcal{L}_e &= -\frac{1}{B}\sum_{i=1}^{B}\big(\tilde{\mathrm{y}}_c^i\big)^\top \log\big(\mathrm{softmax}(\text{logits}_e^i)\big), \\ \mathcal{L}_c &= -\frac{1}{B}\sum_{i=1}^{B}\big(\tilde{\mathrm{y}}_c^i\big)^\top \log\big(\mathrm{softmax}(\text{logits}_c^i)\big), \end{aligned} \tag{11}$$

where y_c^i is the one-hot encoding of the ground-truth attribute–object pair and $\tilde{\mathrm{y}}_c^i$ is the smoothed version of y_c^i.

Attribute classification logits. To encourage accurate attribute prediction, we compute attribute logits by matching the attribute-branch visual feature v_a^i with a matrix of attribute textual prototypes $\mathrm{z}_a \in \mathbb{R}^{A\times d}$:

$$\text{logits}_a^i = \tau\,(\mathrm{v}_a^i\,\mathrm{z}_a^{+\top} - \mathrm{v}_a^i\,\mathrm{z}_a^{-\top}) \in \mathbb{R}^A, \tag{12}$$

where τ is the learnable logit scale and A is the number of attributes. We then apply a cross-entropy loss to supervise attribute classification. Let $\mathbf{y}_{\text{attr}}^i$ be the one-hot attribute label:

$$\mathcal{L}_a = -\frac{1}{B}\sum_{i=1}^{B}\big(\mathrm{y}_a^i\big)^\top \log\big(\mathrm{softmax}(\text{logits}_a^i)\big). \tag{13}$$

Object classification logits Similarly, we supervise object prediction with a cross-entropy loss. Given the object-branch visual feature v_o^i and the matrix of object textual prototypes $\mathrm{z}_o \in \mathbb{R}^{O\times d}$, the object logits are computed as

$$\text{logits}_o^i = \tau\,(\mathrm{v}_o^i\,\mathrm{z}_o^{+\top} - \mathrm{v}_o^i\,\mathrm{z}_o^{-\top}) \in \mathbb{R}^O, \tag{14}$$

where τ is the learnable logit scale and O is the number of object categories. Let $\mathrm{y}_{\text{obj}}^i$ denote the one-hot object label. The loss is defined as

$$\mathcal{L}_o = -\frac{1}{B}\sum_{i=1}^{B}\big(\mathrm{y}_o^i\big)^\top \log\big(\mathrm{softmax}(\text{logits}_o^i)\big). \tag{15}$$

Logic-based regularization. To better distinguish positive and negative prompts, we introduce logic-based regularization that orders them by target confidence and predictive uncertainty. For branch $b \in \{a, o, c\}$ and prompt type r, we compute

$$p_b^r = \mathrm{softmax}\Big(\tau \mathbf{v}_b \mathbf{Z}_b^{r\top}\Big),\, s_b^r = \langle \mathbf{y}_b, p_b^r\rangle, \tag{16}$$

where τ is the temperature and $\mathbf{y}_b$ is the one-hot target. We use $r \in \{+, -\}$ for attribute/object branches and $r \in \{+, a-, o-\}$ for the composition branch, where $a-$ and $o-$ replace the attribute or object with its negative counterpart.

We compare positive and negative prompt pairs

$$\mathcal{P} = \{(a, +, -), (o, +, -), (c, +, a-), (c, +, o-)\}. \tag{17}$$

For each $(b, r^+, r^-) \in \mathcal{P}$, let

$$\Delta_s = s_b^{r^+} - s_b^{r^-},\, \Delta_H = \mathcal{H}(p_b^{r^-}) - \mathcal{H}(p_b^{r^+}), \tag{18}$$

where $\mathcal{H}(p) = -\sum_i p_i \log p_i$. The regularization loss is

$$\mathcal{L}_{\text{reg}} = \sum_{(b,r^+,r^-)\in\mathcal{P}} [\phi(m_s - \Delta_s) + \lambda_H \phi(m_H - \Delta_H)] \tag{19}$$

where m_s and m_H are margins, λ_H weights the entropy term, and $\phi(\cdot)$ is implemented as $\mathrm{softplus}(\cdot)$.

Total training loss. To learn disentangled and compositional representations for CZSL, we optimize a multi-objective loss that supervises both individual components (attributes and objects) and their compositions. The total loss is defined as:

$$\mathcal{L}_{\text{total}} = \lambda_c(\mathcal{L}_e + \mathcal{L}_c) + \lambda_a\mathcal{L}_a + \lambda_o\mathcal{L}_o + \lambda_r\mathcal{L}_{reg} \tag{20}$$

where λ is a hyperparameter that controls the weight of each loss.

Each component encourages the model to capture a distinct aspect of compositional structure: $\mathcal{L}_c$ and $\mathcal{L}_e$ focus on joint composition classification, $\mathcal{L}_a$ and $\mathcal{L}_o$ supervise disentangled attribute and object recognition. This formulation ensures effective generalization to unseen attribute-object compositions during inference.

Inference with multi-branch logits. During inference, the model predicts compositional concepts by fusing semantic signals from three branches: *composition*, *attribute*, and *object*. Given an image, we first obtain the branch-specific logits logits_c, logits_a, and logits_o from Eq. (10), Eq. (12), and Eq. (14), respectively.

To improve compositional consistency, we integrate these predictions using a weighted fusion strategy. For each candidate pair (a_i, o_i), the final score is computed as a combination of the pair-level logits and the product of independent attribute and object probabilities:

$$\begin{aligned} \log_c &= \alpha \text{logits}_c + (1-\alpha)\text{logits}_e \\ p_a &= \text{softmax}(\text{logits}_a), \\ p_o &= \text{softmax}(\text{logits}_o), \\ s_{\text{final}}^{(a_i,o_i)} &= \log_c^{(a_i,o_i)} + p_a(a_i)\, p_o(o_i). \end{aligned} \tag{21}$$

where α is a weighting coefficient. This fusion encourages agreement between holistic (pair-level) and factorized (attribute/object) predictions, enabling robust inference on unseen compositions.

4 Experiments

Datasets. We evaluate NEPROV on three widely used CZSL benchmarks: MIT-States (Isola et al., 2015), UT-Zappos (Yu and Grauman, 2014), and C-GQA (Naeem et al., 2021), the latter being derived from GQA (Hudson and Manning, 2019). We follow the standard dataset splits used in prior work (Purushwalkam et al., 2019) for fair comparison.

Evaluation metrics. We follow the standard CZSL evaluation protocol in both the CW and OW settings (Purushwalkam et al., 2019; Mancini et al., 2021; Nagarajan and Grauman, 2018; Misra et al., 2017). Let $b \in \mathbb{R}$ denote a calibration bias that adjusts the scores of seen compositions relative to unseen ones. We report four standard metrics: **S**, the best accuracy on seen compositions as $b \to +\infty$; **U**, the best accuracy on unseen compositions as $b \to -\infty$; **H**, the best harmonic mean between seen and unseen accuracies over all values of b; and **AUC**, the area under the seen–unseen accuracy curve obtained by sweeping b from $+\infty$ to $-\infty$. Together, these metrics reflect both absolute performance and the trade-off between seen and unseen compositions. Among them, AUC is often regarded as the most informative single-number summary for CZSL.

Implementation details. We implement NEPROV with a pretrained CLIP ViT-L/14 backbone (Radford et al., 2021), which has become a common backbone choice in recent CLIP-based CZSL studies. We optimize the model with Adam (Kingma and Ba, 2014) and a step-based learning-rate scheduler. The temperature τ, i.e., CLIP's logit scale, is initialized from the pretrained checkpoint and remains learnable during training. The bottleneck adapter uses a hidden dimension of 128 and a dropout rate of 0.1. Both the visual progressive disentanglement module and the post-SA MLP use a dropout rate of 0.3. We train the model for 15 epochs on each dataset with a learning rate of 1×10^{-4} and set both the training and evaluation batch sizes to 64. The remaining hyperparameters are reported in Appendix A. All experiments are conducted on a single NVIDIA A100 GPU.

We compare NEPROV with recent CLIP-based CZSL methods, including CSP (Nayak et al., 2023), DFSP (Lu et al., 2023), CDS-CZSL (Li et al., 2024b), Troika (Huang et al., 2024), MSCI (Wang et al., 2025), CAILA (Zheng et al., 2024), TsCA (Li et al., 2024a), ClusPro (Qu et al., 2025), LogiCZSL (Wu et al., 2025), and VP-CMJL (Zhang et al., 2025), under both the CW and OW settings across three benchmarks. We report Seen accuracy (S), Unseen accuracy (U), Harmonic Mean (H), and AUC. Quantitative results are presented in Table 1 and Table 2, followed by qualitative analyses of compositional disentanglement and generalization. For competing methods, we report the results from their original papers. To ensure a fair comparison, all methods are based on the same CLIP backbone configuration, i.e., CLIP ViT-L/14, and are evaluated under the standard CZSL dataset split protocol. For our method, the random seed is set to 0 by default across all experiments.

CW evaluation. In the CW setting, prediction is restricted to a benchmark-provided set of valid compositions (typically the union of seen and unseen pairs); no distractor pairs outside this set are considered. The results demonstrate that NEPROV consistently delivers strong performance under CW. On C-GQA, it achieves the best scores across all metrics, with a harmonic mean (H) of 36.48% and an AUC of 17.94%. On MIT-States, it sets a new state-of-the-art with 41.33% H and 24.51% AUC. On UT-Zappos, it leads in AUC (48.4%), ranks first in H (61.7%), and remains competitive in seen accuracy (70.6%) and unseen accuracy (76.2%). These results indicate that NEPROV effectively aligns vi-

Table 1: CW CZSL results on **UT-Zappos**, **MIT-States**, and **C-GQA**. Metrics include Seen Accuracy (S), Unseen Accuracy (U), Harmonic Mean (H), and Area Under the Curve (AUC). Best results are in **bold**, and second-best results are underlined.

Method	MIT-States				UT-Zappos				C-GQA			
	S(↑)	U(↑)	H(↑)	AUC(↑)	S(↑)	U(↑)	H(↑)	AUC(↑)	S(↑)	U(↑)	H(↑)	AUC(↑)
CLIP (Radford et al., 2021)	30.2	40.0	26.1	11.0	15.8	49.1	15.6	5.0	7.5	25.0	8.6	1.4
CSP (Nayak et al., 2023)	46.6	49.9	36.3	19.4	64.2	66.2	46.6	33.0	28.8	26.8	20.5	6.2
DFSP (Lu et al., 2023)	46.9	52.0	37.3	20.6	66.7	71.7	47.2	36.0	38.2	32.0	27.1	10.5
CDS-CZSL (Li et al., 2024b)	50.3	52.9	39.2	22.4	63.9	74.8	52.7	39.5	38.3	34.2	28.1	11.1
Troika (Huang et al., 2024)	49.0	53.0	39.3	22.1	66.8	73.8	54.6	41.7	41.0	35.7	29.4	12.4
MSCI (Wang et al., 2025)	50.2	53.4	39.9	22.8	67.4	75.5	59.2	45.8	42.4	38.2	31.7	14.2
CAILA (Zheng et al., 2024)	51.0	53.9	39.9	23.4	67.8	74.0	57.0	44.1	43.9	38.5	32.7	14.8
TsCA (Li et al., 2024a)	51.2	52.9	39.9	23.0	68.7	75.8	58.5	46.1	43.8	38.9	33.1	15.2
LogiCZSL (Wu et al., 2025)	50.8	53.9	40.5	23.4	69.6	74.9	57.8	45.8	44.4	39.4	33.3	15.3
ClusPro (Qu et al., 2025)	52.1	**54.0**	40.7	23.8	70.7	76.0	58.5	46.6	44.3	37.8	32.8	14.9
VP-CMJL (Zhang et al., 2025)	51.8	52.6	40.4	23.3	**71.0**	76.0	58.5	47.9	46.0	40.2	34.9	16.3
NEPROV (Ours)	**53.57**	53.56	**41.33**	**24.51**	70.6	**76.2**	**61.7**	**48.4**	**46.43**	**43.71**	**36.48**	**17.94**

Table 2: OW CZSL results on **UT-Zappos**, **MIT-States**, and **C-GQA**. Metrics include Seen Accuracy (S), Unseen Accuracy (U), Harmonic Mean (H), and Area Under the Curve (AUC). Best results are in **bold**, and second-best results are underlined.

Method	MIT-States				UT-Zappos				C-GQA			
	S(↑)	U(↑)	H(↑)	AUC(↑)	S(↑)	U(↑)	H(↑)	AUC(↑)	S(↑)	U(↑)	H(↑)	AUC(↑)
CLIP (Radford et al., 2021)	30.1	14.3	12.8	3.0	15.7	20.6	11.2	2.2	7.5	4.6	4.0	0.3
CSP (Nayak et al., 2023)	46.3	15.7	17.4	5.7	64.1	44.1	38.9	22.7	28.7	5.2	6.9	1.2
DFSP (Lu et al., 2023)	47.5	18.5	19.3	6.8	66.8	60.0	44	30.3	38.3	7.2	10.4	2.4
CDS-CZSL (Li et al., 2024b)	49.4	21.8	22.1	8.5	64.7	61.3	48.2	32.3	37.6	8.2	11.6	2.7
Troika (Huang et al., 2024)	48.8	18.7	20.1	7.2	66.4	61.2	47.8	33.0	40.8	7.9	10.9	2.7
MSCI (Wang et al., 2025)	49.2	20.6	21.2	7.9	67.4	63.0	53.2	37.3	42.0	10.6	13.7	3.8
CAILA (Zheng et al., 2024)	51.0	21.6	20.2	8.2	67.8	59.7	49.4	32.8	43.9	8.0	11.5	3.1
TsCA (Li et al., 2024a)	50.8	21.7	22.3	8.7	69.8	63.4	52.2	37.1	44.3	11.4	14.7	4.5
LogiCZSL (Wu et al., 2025)	50.7	21.4	22.4	8.7	69.6	63.7	50.8	36.2	43.7	9.3	12.6	3.4
ClusPro (Qu et al., 2025)	51.2	**22.1**	23.0	9.3	71.0	66.2	54.1	39.5	41.6	8.3	11.6	3.0
VP-CMJL (Zhang et al., 2025)	51.8	22.0	19.9	8.3	71.9	**66.6**	54.5	**41.4**	46.0	11.5	15.5	4.6
NEPROV(Ours)	**52.4**	**22.1**	**23.1**	**9.4**	**72.3**	65.7	**56.4**	**41.4**	**46.2**	**12.6**	**16.7**	**5.1**

sual and textual primitives for both seen and unseen compositions in the CW regime.

OW evaluation. In the OW setting, the candidate space is expanded to the full Cartesian product of attributes and objects (or a large superset), introducing many implausible/distractor pairs and making recognition substantially harder. The results demonstrate that NEPROV remains competitive across datasets. On C-GQA—the most challenging benchmark in our suite—it attains the best seen accuracy (46.2%), unseen accuracy (12.6%), harmonic mean H (16.7%), and AUC (5.1%), establishing a new state of the art. On MIT-States, NEPROV also ranks first (H = 23.1%, AUC = 9.4%). On UT-Zappos, it maintains strong generalization (H = 56.4%, AUC = 41.4%). These findings suggest that NEPROV supports robust generalization to novel attribute–object compositions under both CW and OW protocols by improving structured alignment and learning more disentangled representations. Additional experiments and ablation studies are provided in Appendix A, Appendix B, and Appendix C.

5 Conclusion

We introduced NEPROV, a unified CZSL framework that combines positive and negative prompts with progressive visual feature disentanglement. Experiments on standard benchmarks show improved generalization to unseen attribute–object compositions. Future work will explore stronger VLM backbones, open-vocabulary attributes, and more explicit compositional reasoning.

Limitations

Two key limitations are identified in NEPROV: (1) The current fusion mechanism relies on fixed weights, which may limit adaptability to diverse

visual domains; (2) The model assumes clean attribute-object annotations during training, which may not generalize well to noisy or weakly labeled data. Addressing these limitations presents opportunities for future work in adaptive fusion and weakly supervised compositional learning.

References

Wentao Bao, Lichang Chen, Heng Huang, and Yu Kong. 2025. Prompting language-informed distribution for compositional zero-shot learning. In *Computer Vision – ECCV 2024*, pages 107–123, Cham. Springer Nature Switzerland.

Yuxia Geng, Runkai Zhu, Jiaoyan Chen, Jintai Chen, Xiang Chen, Zhuo Chen, Shuofei Qiao, Yuxiang Wang, Xiaoliang Xu, and Sheng-Jun Huang. 2025. Graph-guided cross-composition feature disentanglement for compositional zero-shot learning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 2678–2690.

Siteng Huang, Biao Gong, Yutong Feng, Min Zhang, Yiliang Lv, and Donglin Wang. 2024. Troika: Multi-path cross-modal traction for compositional zero-shot learning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24005–24014.

Drew A. Hudson and Christopher D. Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6693–6702.

Phillip Isola, Joseph J Lim, and Edward H Adelson. 2015. Discovering states and transformations in image collections. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1383–1391.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Miaoge Li, Jingcai Guo, Richard Yi Da Xu, Dongsheng Wang, Xiaofeng Cao, Zhijie Rao, and Song Guo. 2024a. Tsca: On the semantic consistency alignment via conditional transport for compositional zero-shot learning. *arXiv preprint arXiv:2408.08703*.

Weizhi Li, Gautam Dasarathy, and Visar Berisha. 2020. Regularization via structural label smoothing. In *international conference on artificial intelligence and statistics*, pages 1453–1463. PMLR.

Yun Li, Zhe Liu, Hang Chen, and Lina Yao. 2024b. Context-based and diversity-driven specificity in compositional zero-shot learning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17037–17046.

Yajing Liu, Yuning Lu, Hao Liu, Yaozu An, Zhuoran Xu, Zhuokun Yao, Baofeng Zhang, Zhiwei Xiong, and Chenguang Gui. 2023. Hierarchical prompt learning for multi-task learning. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10888–10898.

Xiaocheng Lu, Song Guo, Ziming Liu, and Jingcai Guo. 2023. Decomposed soft prompt guided fusion enhancing for compositional zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23560–23569.

Massimiliano Mancini, Muhammad Ferjad Naeem, Yongqin Xian, and Zeynep Akata. 2021. Open world compositional zero-shot learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5218–5226.

Ishan Misra, Abhinav Gupta, and Martial Hebert. 2017. From red wine to red tomato: Composition with context. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1160–1169.

Muhammad Ferjad Naeem, Yongqin Xian, Federico Tombari, and Zeynep Akata. 2021. Learning graph embeddings for compositional zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 953–962.

Tushar Nagarajan and Kristen Grauman. 2018. Attributes as operators: Factorizing unseen attribute-object compositions. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part I*, page 172–190, Berlin, Heidelberg. Springer-Verlag.

Nihal V. Nayak, Peilin Yu, and Stephen H. Bach. 2023. Learning to compose soft prompts for compositional zero-shot learning. In *International Conference on Learning Representations*.

Senthil Purushwalkam, Maximilian Nickel, Abhinav Gupta, and Marc'Aurelio Ranzato. 2019. Task-driven modular networks for zero-shot compositional learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3593–3602.

Hongyu Qu, Jianan Wei, Xiangbo Shu, and Wenguan Wang. 2025. Learning clustering-based prototypes for compositional zero-shot learning. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

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

Yue Wang, Shuai Xu, Xuelin Zhu, and Yicong Li. 2025. Msci: Addressing clip's inherent limitations for compositional zero-shot learning. *arXiv preprint arXiv:2505.10289*.

Peng Wu, Xiankai Lu, Hao Hu, Yongqin Xian, Jianbing Shen, and Wenguan Wang. 2025. Logiczsl: Exploring logic-induced representation for compositional zero-shot learning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 30301–30311.

Aron Yu and Kristen Grauman. 2014. Fine-grained visual comparisons with local learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Shiyu Zhang, Cheng Yan, Yang Liu, Chenchen Jing, Lei Zhou, and Wenjun Wang. 2025. Learning visual proxy for compositional zero-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2793–2802.

Zhaoheng Zheng, Haidong Zhu, and Ram Nevatia. 2024. Caila: concept-aware intra-layer adapters for compositional zero-shot learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1721–1731.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022. Conditional prompt learning for vision-language models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16795–16804.

NEPROV: Compositional Zero-Shot Learning with Negative Prompts and Progressive Visual Feature Disentanglement

Supplementary Material

This appendix provides additional details for the experimental evaluation of NEPROV. We first describe the datasets, baselines, and implementation settings. We then report additional ablation and sensitivity analyses, followed by model complexity and qualitative results.

A Datasets and Experimental Details

A.1 Datasets

We evaluate NEPROV on three widely used compositional zero-shot learning (CZSL) benchmarks: MIT-States, UT-Zappos, and C-GQA. As summarized in Table 3, these datasets differ substantially in scale, concept vocabulary, and compositional diversity, thereby providing a comprehensive testbed for evaluating compositional generalization.

MIT-States is a standard CZSL benchmark consisting of object images annotated with state attributes. It contains a large set of visually diverse attribute–object compositions, many of which are fine-grained or visually subtle. The dataset is therefore challenging because models must distinguish compositions that may share either similar attributes or similar objects.

UT-Zappos focuses on shoe images and provides a relatively clean and controlled CZSL benchmark. Compared with MIT-States, it has a smaller concept space and lower semantic diversity, making it useful for evaluating whether a method can learn reliable attribute–object composition under less noisy visual variation.

C-GQA is a large-scale compositional benchmark derived from GQA. It contains a much larger vocabulary of attributes and objects, as well as substantially more valid compositions. Owing to its scale and semantic diversity, C-GQA poses the most challenging compositional generalization setting among the three datasets.

Following prior work, we adopt the standard compositional splits and evaluate under both closed-world (CW) and open-world (OW) protocols. In the CW setting, predictions are restricted to feasible compositions in the test split, whereas the OW setting allows a much larger candidate composition space.

Table 3: Dataset statistics.

Dataset	#Images	#Attr.	#Obj.	#Comp.
MIT-States	53,753	115	245	1,962
UT-Zappos	50,025	16	12	116
C-GQA	39,298	453	870	9.5K+

A.2 Baselines and Evaluation Protocol

We compare NEPROV with recent CLIP-based CZSL methods and representative compositional learning approaches. Specifically, we include: CLIP (Radford et al., 2021), which serves as the zero-shot vision–language baseline; CSP (Nayak et al., 2023), which learns composable soft prompts for attributes and objects; DFSP (Lu et al., 2023), which decomposes soft prompts into primitive-level representations and fuses them with visual features; CDS-CZSL (Li et al., 2024b), which models context-based and diversity-driven attribute specificity for composition recognition; Troika (Huang et al., 2024), which introduces multi-path cross-modal traction to calibrate visual and textual representations; CAILA (Zheng et al., 2024), which inserts concept-aware intra-layer adapters into CLIP; TsCA (Li et al., 2024a), which aligns patches, primitives, and compositions through conditional transport; MSCI (Wang et al., 2025), which exploits multi-stage cross-modal interactions to incorporate intermediate CLIP visual features; ClusPro (Qu et al., 2025), which learns diversified clustering-based prototypes for primitive concepts; LogiCZSL (Wu et al., 2025), which injects logic-induced semantic relations into compositional representations; and VP-CMJL (Zhang et al., 2025), which learns visual proxies to reduce modality gaps between textual prototypes and visual features.

For competing methods, we report results from their original papers whenever available. All methods are based on the same CLIP ViT-L/14 backbone and follow the standard CZSL split protocol, where attribute–object compositions are divided into seen pairs for training and disjoint unseen pairs for evaluation. All methods are evaluated under the same CW/OW evaluation protocol and report the standard CZSL metrics: seen accuracy (S), unseen accuracy (U), harmonic mean (HM),

and area under the seen–unseen curve (AUC). Unless otherwise specified, the random seed is set to 0 by default. These consistent backbone, split, seed, and evaluation settings ensure a fair comparison across methods.

A.3 Implementation Details and Hyperparameters

We implement NEPROV with a pretrained CLIP ViT-L/14 backbone, following common practice in recent CLIP-based CZSL research. The CLIP text encoder is kept frozen, whereas the visual encoder is adapted using lightweight residual adapters and the proposed progressive visual disentanglement (Visual PD) modules. Unless otherwise specified, all experiments use a random seed of 0 and are conducted on a single NVIDIA A100 GPU. The default hyperparameter configuration is summarized in Table 4.

We train NEPROV for 15 epochs using Adam with a learning rate of 1×10^{-4} and a batch size of 64 for both training and evaluation. The bottleneck adapter hidden dimension is set to 128, with a dropout rate of 0.1. Both Visual PD and the post-SA MLP use a dropout rate of 0.3. For the training objectives, we set the structured label smoothing coefficient to $\epsilon = 0.025$, the composition, attribute, and object loss weights to $\lambda_c = \lambda_a = \lambda_o = 1.0$, and the logic-based regularization weight to $\lambda_r = 1.0$. The score and entropy margins are set to $m_s = m_H = 0$, and the inference fusion coefficient is set to $\alpha = 0.85$.

Table 4: Default hyperparameter configuration of NEPROV.

Hyperparameter	Value
Backbone	CLIP ViT-L/14
Text encoder	Frozen
Visual encoder	Adapter-tuned
Random seed	0
Optimizer	Adam
Learning rate	1×10^{-4}
Training epochs	15
Batch size	64
Adapter hidden dim.	128
Adapter dropout	0.1
Visual PD dropout	0.3
Post-SA MLP dropout	0.3
Label smoothing ϵ	0.025
Composition loss λ_c	1.0
Attribute loss λ_a	1.0
Object loss λ_o	1.0
Regularization loss λ_r	1.0
Entropy reg. weight λ_H	1.0
Score margin m_s	0
Entropy margin m_H	0
Fusion coefficient α	0.85
Visual PD tokens M	32
Context tokens n_{ctx}	3
Visual PD depth	24 layers

B Additional Ablation and Sensitivity Analyses

This section provides additional analyses of the main components and hyperparameters in NEPROV. We first ablate negative prompts, Visual PD depth, and structured label smoothing. We then analyze the sensitivity of the fusion coefficient α, the smoothing coefficient ϵ, and the regularization weight λ_r.

B.1 Ablation on Negative Prompts

We examine the contribution of negative prompts by removing attribute-level and object-level negative prompts from the training objective. As shown in Table 5, using both negative attribute prompts and negative object prompts yields the best overall performance on both MIT-States and C-GQA. On MIT-States, removing either type of negative prompt reduces AUC from 24.51 to 24.30 or 24.03, and removing both further decreases AUC to 23.99. On C-GQA, the same pattern is more pronounced: AUC decreases from 17.94 to 17.58 or 17.40 when one type of negative prompt is removed, and further drops to 17.03 when both are removed.

These results indicate that attribute-level and object-level negative prompts provide complementary supervision. Negative attribute prompts help distinguish state or attribute evidence, whereas negative object prompts help suppress object-level confusion. Their joint use improves the balance between seen and unseen recognition, as reflected by the stronger AUC and HM values.

B.2 Ablation on Visual PD Depth

We further study where Visual PD should be inserted in the visual encoder. We apply Visual PD to the last k ViT blocks and vary k from 3 to 24. As shown in Table 6, applying Visual PD to all 24 layers yields the best AUC and HM on both MIT-States and C-GQA. On MIT-States, AUC increases from 23.83 with the last 3 layers to 24.51 with all 24 layers. On C-GQA, AUC increases from 16.83 to 17.94 as Visual PD is extended from the last 3 layers to all layers.

These results suggest that compositional cues are distributed across multiple representation depths

Table 5: Ablation on negative prompts. We evaluate the effect of removing attribute-level and object-level negative prompts.

Negative Prompts		MIT-States in CW Setting			
negative attributes	negative objects	AUC (↑)	HM (↑)	S (↑)	U (↑)
✓	✓	**24.51**	**41.33**	**53.57**	**53.56**
	✓	24.30	41.40	52.73	53.49
✓		24.03	41.22	52.23	53.40
		23.99	40.83	52.90	53.03
		CGQA in CW Setting			
✓	✓	**17.94**	**36.48**	46.43	**43.71**
	✓	17.58	35.91	46.36	42.57
✓		17.40	35.85	**46.69**	42.22
		17.03	35.74	46.26	41.35

Table 6: Ablation on the layers to which Visual PD is applied.

MIT-States in the CW Setting				
Visual PD Layers	AUC (↑)	HM (↑)	S (↑)	U (↑)
All 24 layers	**24.51**	**41.33**	**53.57**	53.56
Last 12 layers	24.20	40.74	53.11	53.76
Last 6 layers	24.08	40.83	52.10	53.21
Last 3 layers	23.83	35.74	51.72	**53.94**
CGQA in the CW Setting				
Visual PD Layers	AUC (↑)	HM (↑)	S (↑)	U (↑)
All 24 layers	**17.94**	**36.48**	46.43	**43.71**
Last 12 layers	17.28	35.95	**46.46**	42.22
Last 6 layers	17.02	35.30	46.38	41.35
Last 3 layers	16.83	34.76	46.36	40.56

rather than being concentrated only in the final layer. Full-depth Visual PD allows the model to progressively refine disentangled attribute-sensitive and object-sensitive evidence throughout the visual backbone. Although partial insertion into the last 12 or 6 layers is already competitive, full-depth insertion provides the strongest overall generalization.

B.3 Ablation on Structured Label Smoothing

We vary the structured label smoothing coefficient ϵ to examine how smoothing strength affects compositional recognition. As shown in Table 7, moderate smoothing performs best on both benchmarks. AUC peaks at $\epsilon = 0.025$, reaching 24.51 on MIT-States and 17.94 on C-GQA. HM follows the same pattern, reaching 41.33 and 36.48, respectively.

These results indicate that structured label smoothing provides a useful regularization effect by redistributing probability mass to semantically related compositions that share the same attribute or object. When ϵ is too small, the model receives limited benefit from semantic smoothing. When ϵ is too large, the supervision can become overly diffuse and blur fine-grained distinctions between compositions. We therefore use $\epsilon = 0.025$ as the default setting.

Table 7: Ablation on label smoothing strength in the CW setting.

ϵ	MIT-States				CGQA			
	AUC	HM	S	U	AUC	HM	S	U
0.01	24.27	41.05	52.98	53.87	17.58	35.91	46.36	42.57
0.02	24.39	41.20	53.11	53.40	17.70	36.30	**46.70**	43.50
0.025	**24.51**	**41.33**	**53.57**	53.56	**17.94**	**36.48**	46.43	**43.71**
0.03	24.35	41.22	53.03	54.08	17.40	35.85	46.69	42.22
0.04	24.18	40.94	52.27	**54.33**	17.37	35.67	46.03	42.57

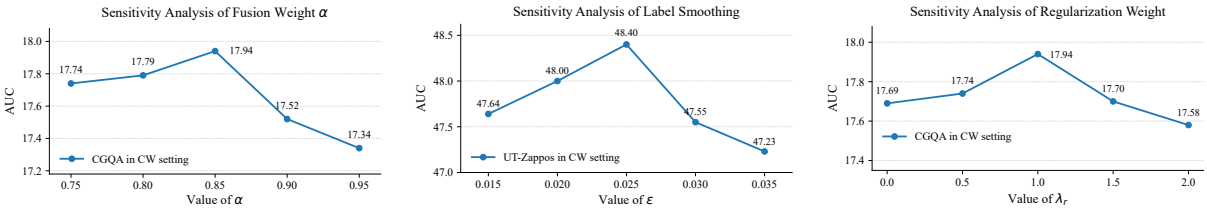
B.4 Sensitivity to Key Hyperparameters

We further examine the sensitivity of NEPROV to three key hyperparameters: the branch-fusion coefficient α, the structured smoothing coefficient ϵ, and the logic-based regularization weight λ_r. The results are shown in Figure 5.

For the fusion coefficient α, NEPROV achieves the best AUC at $\alpha = 0.85$ and remains relatively stable within a moderate range. This indicates that the model benefits from combining holistic composition-level evidence with factorized attribute/object predictions. When α is either too small or too large, the contribution of one source of evidence is weakened, leading to inferior performance.

For the smoothing coefficient ϵ, performance first improves as ϵ increases and then declines when the smoothing becomes excessive. This trend is consistent with the results in Table 7: moderate structured smoothing alleviates overconfidence and improves robustness, whereas over-smoothing weakens class discriminability.

For the regularization weight λ_r, the best performance is obtained at $\lambda_r = 1.0$. Smaller values provide insufficient separation between positive and negative prompts, while overly large values may impose excessive constraints on the score distributions. Overall, these sensitivity results support the default hyperparameter configuration used in the main experiments.



(a) Fusion coefficient α (b) Smoothing coefficient ϵ (c) Regularization weight λ_r

Figure 5: Sensitivity analyses of key hyperparameters under the closed-world setting.

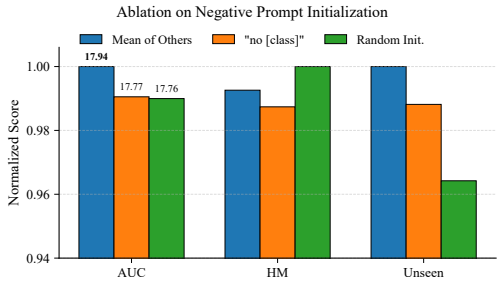


Figure 6: Ablation on negative prompt initialization on C-GQA under the closed-world setting. Scores are normalized by the best value for each metric to highlight relative differences.

B.5 Negative Prompt Initialization

For each attribute or object concept, we construct its positive prompt from the token embeddings of the corresponding concept name. The negative prompt is initialized as the mean representation of all other concepts within the same concept space. This mean-of-others strategy provides a simple way to introduce contrastive counter-evidence: instead of relying on a manually specified negation phrase, the negative prompt is anchored by the semantic neighborhood formed by competing attributes or objects.

To assess whether NEPROV depends on this particular initialization, we compare the proposed mean-of-others strategy with two alternatives: initializing negative prompts using the textual template "no [class]" and random initialization. As shown in Figure 6, all three strategies produce competitive results, suggesting that NEPROV is not overly sensitive to the specific initialization scheme. Nevertheless, the mean-of-others initialization achieves the best AUC and unseen accuracy, indicating that aggregating competing concepts provides a more informative semantic anchor for learning negative prompts. These results support our default design while showing that the effectiveness of negative prompting does not rely on a brittle initialization choice.

Table 8: Model complexity and runtime comparison on C-GQA under the closed-world setting.

Method	Total Params (M)	Trainable Params (M)	Runtime (min/epoch)
CSP	427.84	304.24	3.03
Troika	450.10	22.54	0.48
NEPROV	493.58	66.02	4.90

C Model Complexity and Qualitative Results

C.1 Model Complexity and Runtime

We evaluate the efficiency of NEPROV in terms of parameter size and training runtime. Compared with existing CLIP-based CZSL methods, NEPROV introduces additional trainable parameters through residual adapters, Visual PD modules, and branch-specific post-SA layers. Nevertheless, the overall parameter increase remains moderate because the CLIP backbone is largely frozen.

As shown in Table 8, NEPROV has more trainable parameters and a longer training time than CSP and Troika. This additional cost mainly comes from progressive visual disentanglement across the visual encoder and branch-specific prediction modules. However, the training remains practical on a single A100 GPU, and the added cost is consistent with the stronger compositional generalization observed in the main results.

C.2 Qualitative Results

We provide qualitative examples to illustrate how NEPROV behaves on unseen attribute–object compositions. Figure 7 presents representative success and failure cases from MIT-States and C-GQA. Each example reports the ground-truth composition and the model prediction. Green boxes denote correct predictions, while red boxes denote failure cases.

The successful cases show that NEPROV can recognize unseen compositions when the attribute

Figure 7: Qualitative results of NEPROV on unseen attribute–object compositions. Green boxes denote correct predictions, and red boxes denote failure cases.

cue is visually grounded and compatible with the object, such as color–object or state–object combinations. This suggests that negative prompting and progressive visual disentanglement help the model align attribute-sensitive and object-sensitive evidence. The failure cases are also informative: they often occur when the attribute cue is ambiguous, when multiple attributes are plausible for the same object, or when the model identifies the object correctly but confuses fine-grained attribute states. These cases suggest that remaining errors are mainly caused by ambiguous visual evidence rather than a complete failure to localize the object.