# CONTROLLING OBJECT-INDUCED SHORTCUTS FOR COMPOSITIONAL ZERO-SHOT LEARNING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Compositional zero-shot learning requires recognizing unfamiliar combinations of familiar attributes and objects. Sparse training combinations make object identity a strong predictor of the composition label, allowing a model to fit observed associations without learning sufficient visual evidence for the attribute. Yet object information also provides the context needed to interpret attribute appearance. We propose SPARC, a framework that regulates this dual role across representation, optimization, and inference. A feedback adapter pools attribute evidence under complementary object-related supports and uses their directional residual to guide token updates. Two training objectives evaluate composition scores relative to object support, while an image-dependent score correction balances competition between object groups at inference. These controls address distinct parts of composition recognition: learning discriminative attribute evidence and balancing its competition across objects. Experiments on three standard benchmarks demonstrate strong compositional generalization, with state-of-the-art AUC and harmonic mean on MIT-States and open-world UT-Zappos. All three regulation stages consistently improve recognition across datasets while adding little inference cost.

## 1 INTRODUCTION

A model can fit its training compositions without learning the visual evidence that defines them. Object identity offers a shortcut: once an object is recognized, its frequent training attributes become plausible answers. An image of a green apple offers clear evidence for *apple*, but selecting *green* requires inspecting that particular apple. If training apples are mostly red, reproducing the familiar association can reduce the loss without resolving the attribute. The model then learns which pairings the training distribution favors, while the evidence needed to recognize a new pairing remains weak.

This distinction is central to compositional zero-shot learning (CZSL). Training covers a subset of attribute–object pairs, and evaluation includes new combinations of the same primitives (Misra et al., 2017; Naeem et al., 2021). A decision rule that works well on observed combinations can therefore fail precisely where compositional generalization is needed. We call the use of object identity as a substitute for composition-specific visual evidence an *object-induced shortcut*. Figure 1 illustrates how a useful object cue can become such a shortcut when training associations dominate the attribute decision.

Removing object information is an unsatisfactory solution. The visual meaning of an attribute depends on what it describes: a *ripe* fruit, a *rusty* tool, and a *wet* garment require different contextual cues. Object-conditioned methods exploit this dependence to improve recognition (Kim et al., 2023; Wu et al., 2025a). The challenge is to preserve the context that helps interpret an attribute while preventing object confidence from standing in for evidence that the attribute is present. This is a question of how object information is used, beyond whether attribute and object features are separated into different branches.

Object influence enters the prediction at several points. In the representation, dominant object-related directions can obscure attribute evidence. During training, object support can make a composition score competitive even when its attribute evidence is weak. At inference, object-dependent score offsets affect which group of compositions wins. These effects require different controls. Better attribute features change the ordering of attributes associated with an object; balancing object-



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

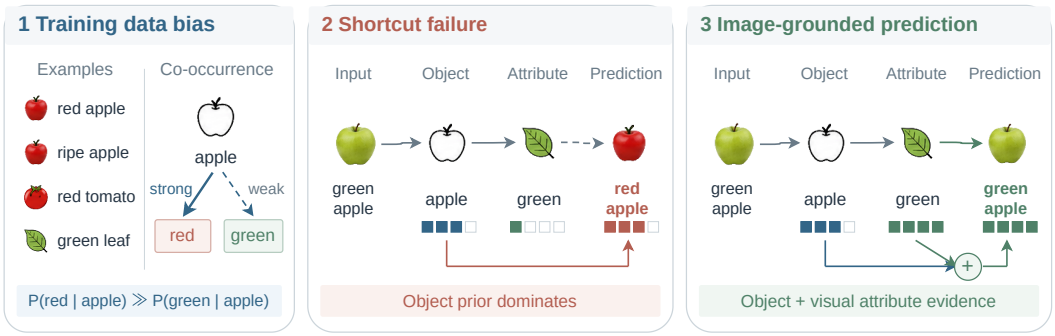


Figure 1: An object-induced shortcut in CZSL. Sparse training combinations favor familiar attribute–object associations (left). A model can recognize the apple correctly yet predict red when the image supports green (middle). The desired prediction retains object context while grounding the attribute in the current image (right).

group scores changes the competition among those groups. A score correction applied uniformly to an object group addresses the latter, so it should complement the learning of attribute evidence.

We develop **SPARC**, **S**hortcut **P**athway-**A**ware **R**egulation for **C**ompositional zero-shot learning, around this division of roles. Its representation module pools attribute evidence under complementary object-related supports. The component of the complementary view that is distinct from the object-related view feeds back into token fusion, preserving evidence beyond their shared direction. At training time, composition scores are evaluated relative to object scores through an object-discounted classification loss and a specificity margin. At inference, a standardized object-alignment score adjusts competition between object groups while retaining the explicit object-recognition branch.

This design links the motivation to the recognition task directly. The adapter improves the evidence from which an attribute–object pair is judged. The score controls then regulate how much object support can contribute to that judgment.

Our contributions are threefold:

- We formulate object-induced shortcuts as a mismatch between fitting observed attribute–object associations and learning the visual evidence needed for new compositions, and distinguish attribute discrimination from object-group competition.
- We introduce SPARC, which couples complementary-evidence feedback with object-relative training and image-dependent score rectification to regulate object influence throughout composition recognition.
- Experiments on MIT-States, UT-Zappos, and C-GQA demonstrate consistent improvements over the adapter baseline in closed- and open-world recognition. Cross-dataset stage ablations and controlled loss and scoring comparisons establish the contribution of the three regulation stages.

## 2 Related Work

**Composition and object-conditioned recognition.** CZSL transfers knowledge of familiar primitives to new attribute–object pairs. Early approaches learn compositional embeddings, transformations, or compatibility structures (Misra et al., 2017; Li et al., 2020; Naeem et al., 2021). Later methods refine primitive features through contrastive learning and attention (Li et al., 2022; Hao et al., 2023). Object conditioning is particularly relevant because attribute appearance varies with the object. Composition Transformer (CoT) (Kim et al., 2023) models this interaction, and the Conditional Probability Framework (CPF) (Wu et al., 2025a) factorizes recognition into object likelihood and conditional attribute likelihood. CPF uses text-enhanced object features to guide attribute learning. SPARC shares the need for contextual attribute recognition and addresses a further issue:

a strong object cue can dominate the features and scores used to judge the whole composition. Its controls act on that dominance while maintaining object support.

**Adapting vision–language models.** CLIP (Radford et al., 2021) supplies transferable image and text representations. CoOp (Zhou et al., 2022) and compositional soft prompting (CSP) (Nayak et al., 2023) adapt textual representations, while DFSP (Lu et al., 2023) and Troika (Huang et al., 2024) improve cross-modal interaction and combine primitive and composition predictions. ClusPro (Qu et al., 2025) learns clustered prototypes; VP-CMJL (Zhang et al., 2025) introduces visual proxies and cross-modal joint learning. LogicZSL (Wu et al., 2025b) supplies logic-induced representations, and recent methods use invertible primitive mappings, expert-based adaptation, or composable flows (Yuan et al., 2026; Zhang et al., 2026; He et al., 2026). These developments improve the features and compatibility functions available to a CZSL model. SPARC focuses on how the model allocates influence between object support and composition-specific evidence. We use the established combination of pair and primitive branches as the scoring framework for this regulation.

**Distribution bias and score adjustment.** Sparse compositional supervision makes class frequencies and primitive associations uneven. ProLT (Jiang & Zhang, 2024) estimates a proximate class prior from primitive predictions aggregated over training samples and incorporates it during training and inference. Mutual balancing (Jiang et al., 2024) addresses imbalance between state and object components. SPARC couples image-dependent object references with feedback in the visual encoder. Its training discount and inference rectification operate at the object-group level, while complementary-evidence feedback acts on the features that distinguish attributes within each group. This combination targets both the formation of composition evidence and its competition at prediction time.

# 3 Method

## 3.1 From Object Support to Composition Evidence

Let $\mathcal{A}$ and $\mathcal{O}$ be the attribute and object vocabularies. Training images have one composition label $y = (a, o)$ from the seen set $\mathcal{C}^s \subset \mathcal{A} \times \mathcal{O}$. The unseen set $\mathcal{C}^u$ shares these primitives, with $\mathcal{C}^s \cap \mathcal{C}^u = \varnothing$. At evaluation, the candidate set $\mathcal{Y}$ is $\mathcal{C}^s \cup \mathcal{C}^u$ in closed world and $\mathcal{A} \times \mathcal{O}$ in open world. A compatibility function $S(I, y)$ ranks these candidates for image $I$.

Object conditioning provides both visual context and a distributional prior. Bayes' rule makes their roles explicit:

$$\log P(a \mid I, o) = \log P(I \mid a, o) + \log P(a \mid o) - \log P(I \mid o). \tag{1}$$

For a fixed image and object, the last term is constant across attributes. The first term describes how well an attribute–object pair explains the image, whereas the second captures how often that pairing occurs. Sparse training coverage makes the latter an attractive route to predicting seen compositions. Generalizing to a new pairing requires the image evidence to remain decisive when that association changes.

The actual classifier must also compare different objects. For any candidate set, its decision can be written as

$$\hat{o} = \arg\max_{o \in \mathcal{O}_{\mathcal{Y}}} \max_{a:(a,o)\in\mathcal{Y}} S(I, a, o), \qquad \hat{a} = \arg\max_{a:(a,\hat{o})\in\mathcal{Y}} S(I, a, \hat{o}), \tag{2}$$

where $\mathcal{O}_{\mathcal{Y}}$ contains objects represented in $\mathcal{Y}$. This separates two requirements: attributes must be discriminable within a group, and the best-supported group must win overall. An object-dependent offset can change the second decision without changing the first. SPARC assigns representation learning to the quality of attribute evidence and uses object-relative score controls to regulate group competition. Figure 2 summarizes the resulting model.

## 3.2 Complementary-Evidence Feedback

We insert feedback adapters into the last three visual transformer blocks of CLIP ViT-L/14. An adapter maps image tokens $X \in \mathbb{R}^{L \times D}$ to an update $\Delta X$, giving $X' = X + \Delta X$. It organizes attribute evidence using object semantics and feeds back the part of the complementary view distinct from the object-related direction.

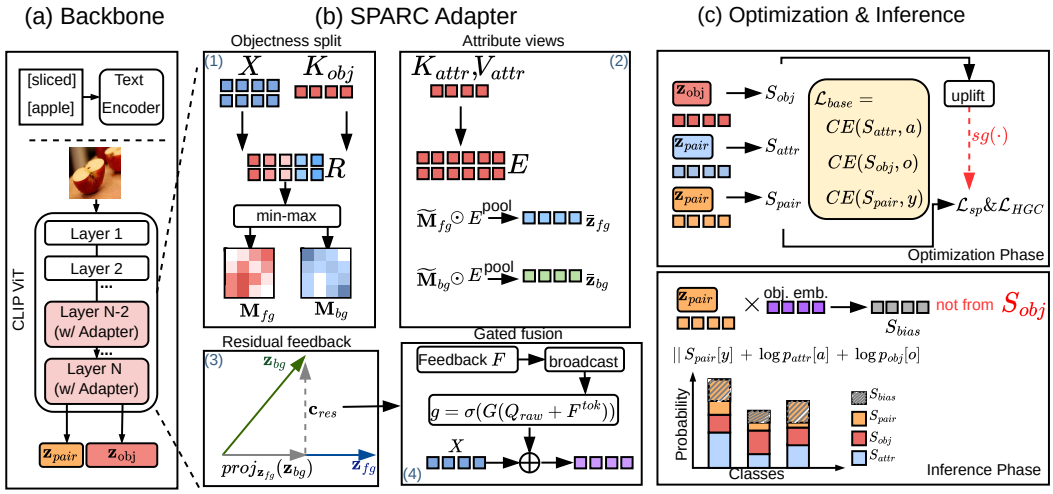


Figure 2: Overview of SPARC. (a) Frozen CLIP encoders with adapters in the last three visual blocks. (b) The adapter computes semantic objectness, pools two attribute views, extracts residual feedback, and gates the token update. Foreground/background subscripts denote the object-related and complementary supports. (c) Pair and primitive scores receive classification supervision; pair–object objectives regulate training, and standardized object alignment in $z_{pair}$ corrects the fused inference score.

**Object-related supports.** Visual tokens and text prototypes are projected into a shared $d$-dimensional adapter space. Let $Q_{\text{raw}}$ be the projected token queries, $Q$ their normalized version, and $K_{obj}$ the normalized object keys. We compute each token's strongest alignment with the object vocabulary:

$$R_i = \max_{o \in \mathcal{O}} \frac{\langle Q_i, K_{obj}[o] \rangle}{\sqrt{d}}, \qquad M_{fg,i} = \frac{R_i - \min_j R_j}{\max_j R_j - \min_j R_j + \epsilon}. \tag{3}$$

The complementary supports are $M_{fg}$ and $M_{bg} = 1 - M_{fg}$, with $\epsilon > 0$ stabilizing the division. A high $M_{fg,i}$ identifies a token aligned with at least one object prototype; $M_{bg}$ emphasizes lower relative alignment. Pooling over all prototypes makes routing available before choosing an object. These scores provide semantic weights for evidence extraction.

**Two views of attribute evidence.** Projected attribute keys $K_{attr}$ and values $V_{attr}$ associate each token with the attribute vocabulary:

$$P_a = \text{softmax}_{\mathcal{A}}(QK_{attr}^{\top}/\sqrt{d}), \qquad E = P_a V_{attr}. \tag{4}$$

Here, $E_i$ is the token's attribute evidence. Normalizing each support over tokens produces two pooled views,

$$\bar{z}_r = \sum_i \widetilde{M}_{r,i} E_i, \qquad \widetilde{M}_{r,i} = \frac{M_{r,i}}{\sum_j M_{r,j} + \epsilon}, \quad r \in \{fg, bg\}. \tag{5}$$

The first view collects evidence at strongly object-related tokens, while the second collects complementary evidence. Both are built from the same attribute vocabulary, so their difference concerns where and how attribute evidence is supported in the image.

**Residual feedback.** Complementary spatial weights can still produce features dominated by the same direction. We remove the component of $\bar{z}_{bg}$ aligned with $\bar{z}_{fg}$:

$$c_{\text{res}} = \bar{z}_{bg} - \frac{\langle \bar{z}_{bg}, \bar{z}_{fg} \rangle}{\|\bar{z}_{fg}\|_2^2 + \epsilon} \bar{z}_{fg}. \tag{6}$$

This residual supplies a direction that the object-related pooled view does not already explain. It is transformed and broadcast to the tokens, then used to compute a token-channel gate:

$$F = \mathrm{LN}(c_{\mathrm{res}} W_{fb}), \qquad g = \sigma\big(\mathcal{G}(Q_{\mathrm{raw}} + F^{tok})\big), \tag{7}$$

$$\Delta X = s\big[g \odot (M_{fg} \odot E) + (1 - g) \odot (M_{bg} \odot E)\big] W_{out}. \tag{8}$$

$F_i^{tok} = F$ broadcasts the image-level feedback, $\mathcal{G}$ is the gate network, and $s$ scales the update. LN and $\sigma$ denote layer normalization and sigmoid; scalar masks are broadcast across channels. The residual guides the gate, while the update retains the token-level attribute evidence $E$. This preserves local attribute information while weighting it according to complementary image evidence.

The final tokens are pooled into a composition representation $z_{pair}$, which also serves the attribute branch. Object-supported pooling provides a separate representation $z_{obj}$ for object prediction. Thus, object recognition remains explicitly supervised while the composition stream has access to evidence beyond the shared pooled direction.

### 3.3 Object-Relative Training

Pair, attribute, and object cosine classifiers produce logits $S_{pair}$, $S_{attr}$, and $S_{obj}$ with a shared CLIP logit scale. The pair and attribute classifiers use $z_{pair}$, and the object classifier uses $z_{obj}$. For a labeled image $y = (a, o)$, the base objective trains all three branches:

$$\mathcal{L}_{base} = \mathrm{CE}_{\mathcal{C}^s}(S_{pair}, y) + \mathrm{CE}_{\mathcal{A}}(S_{attr}, a) + \mathrm{CE}_{\mathcal{O}}(S_{obj}, o). \tag{9}$$

We add two object-relative objectives that evaluate composition support against an object reference.

**Object-discounted classification.** For a candidate $c = (a', o')$, lift the object logit to the pair space as $S^{\uparrow}_{obj}[c] = S_{obj}[o']$. The shortcut-regulation loss applies cross-entropy after discounting this support:

$$\mathcal{L}_{sp} = \mathrm{CE}_{\mathcal{C}^s}\left(S_{pair} - \lambda\, \mathrm{sg}(S^{\uparrow}_{obj}), y\right). \tag{10}$$

The pair classifier is trained to keep the correct composition competitive after part of its object support is subtracted. Candidates sharing an object receive the same discount, making this an object-group training signal. Within a group, the pair and attribute objectives distinguish the correct attribute from alternatives. Here $\mathrm{sg}$ denotes stop-gradient: the object logits serve as detached references, preserving the object branch's recognition objective.

**Hierarchical specificity.** A composition asserts more than the presence of its object. We express this additional requirement through a margin between the correct pair score and its object reference:

$$\mathcal{L}_{HGC} = \max\big(0, \mathrm{sg}(S_{obj}[o]) + m - S_{pair}[y]\big), \qquad \mathcal{L} = \mathcal{L}_{base} + \mathcal{L}_{sp} + \mathcal{L}_{HGC}. \tag{11}$$

The shared logit scale gives $m$ a common score unit. The two added losses play complementary roles: $\mathcal{L}_{sp}$ evaluates the labeled pair against competing compositions after object discounting, while $\mathcal{L}_{HGC}$ sets a minimum gap over its own object score. Both act on the composition stream while retaining useful object supervision.

### 3.4 Balancing Object Groups at Inference

**Combining branch evidence.** Let $p_{attr} = \mathrm{softmax}(S_{attr})$ and $p_{obj} = \mathrm{softmax}(S_{obj})$. We combine the three branches using

$$S_{fuse}(I, a, o) = S_{pair}(I, a, o) + \log p_{attr}(a \mid I) + \log p_{obj}(o \mid I). \tag{12}$$

For ranking, this is equivalent to adding the log-probabilities of all three branches: replacing $S_{pair}$ by $\log \mathrm{softmax}_{\mathcal{Y}}(S_{pair})$ subtracts one image-dependent constant from every candidate. The rule consequently combines multiplicative support from the pair and primitive predictions in log space. In particular, the positive object term retains evidence that an object is present.

**Image-dependent object correction.** We measure object alignment in the representation used for composition ranking,

$$b_o(I) = \cos(z_{pair}(I), t_o), \qquad \hat{b}_o(I) = \frac{b_o(I) - \mu_I}{\sigma_I + \epsilon}, \tag{13}$$

where $t_o$ is the object text embedding and $\mu_I, \sigma_I$ are the mean and standard deviation across objects. This reference measures object alignment inside $z_{pair}$, complementing the object-recognition evidence from $z_{obj}$. Standardization expresses each alignment relative to the other objects for that image, giving a common interpretation to the correction strength $\alpha$. The final score is

$$S_{final}(I, a, o) = S_{fuse}(I, a, o) - \alpha \hat{b}_o(I). \tag{14}$$

**Connection to composition ranking.** Writing $T_o(I) = \max_{a:(a,o)\in\mathcal{Y}} S_{fuse}(I, a, o)$, the winning object group maximizes $T_o(I) - \alpha \hat{b}_o(I)$. The strongest attribute candidate in each group is preserved, while object alignment changes the margin between groups. For candidates with different objects, the margin changes by $-\alpha(\hat{b}_{o_1} - \hat{b}_{o_2})$. A correction can therefore change a close ranking even when it is small relative to the full score range. This is the final role of regulation: the representation and training objectives establish composition evidence, and rectification balances its competition across objects. Short derivations of these score properties are collected in Appendix A.

# 4 Experiments

## 4.1 Setup

**Benchmarks and metrics.** We evaluate on MIT-States (Isola et al., 2015), UT-Zappos (Yu & Grauman, 2014), and C-GQA (Naeem et al., 2021), following the natural compositional splits and preprocessing used by Troika (Huang et al., 2024). Their attribute/object vocabularies contain 115/245, 16/12, and 413/674 primitives, respectively. Closed-world (CW) evaluation considers seen and designated unseen pairs; open-world (OW) evaluation expands the composition space to all attribute–object combinations. We report area under the seen–unseen accuracy curve (AUC), best harmonic mean (HM), and best seen and unseen top-1 accuracies (S and U). These metrics summarize the standard calibration sweep that trades off seen and unseen predictions. S and U are separate maxima and need not occur at the operating point that maximizes HM. All values are percentages, with AUC on the conventional percentage scale.

**Training.** We freeze the original CLIP ViT-L/14 image and text encoders and train the feedback adapters and prediction heads. AdamW uses a learning rate of $10^{-5}$ and batch size 32 on an NVIDIA A6000. Training runs for at most 30 epochs, with early stopping after five epochs without a decrease in validation loss; the lowest-validation-loss checkpoint is evaluated. Unless varied explicitly, $\lambda = 0.1$, $\alpha = 0.2$, and $m = 2.0$. Our recognition results are means over three independent runs. Comparison results are taken from published CZSL benchmark tables.

## 4.2 Comparison with Prior Methods

Table 1 compares CLIP ViT-L/14 methods using the standard benchmark protocols. It covers prompt learning, cross-modal fusion, prior adjustment, prototype learning, and recent logical, invertible, expert-based, and flow-based models. ProLT and CPF are included using their published CLIP variants. They provide direct comparisons with distribution-prior adjustment and object-conditioned attribute learning, respectively.

**Closed world.** SPARC reaches 24.8 AUC and 41.8 HM on MIT-States, the best values among the listed methods. Relative to ClusPro, it improves AUC by 1.0, 2.0, and 2.5 points on MIT-States, UT-Zappos, and C-GQA. It also exceeds VP-CMJL and Logic-Troika on all three datasets. The strongest results are distributed across methods: EVA leads on UT-Zappos and C-GQA, while SPARC leads on MIT-States. Against the closest object-conditioned comparison, CPF, SPARC improves C-GQA AUC from 15.4 to 17.4 and HM from 33.6 to 35.8. These results show that regulating object influence is effective alongside the strong contextual representations supplied by CLIP.

Table 1: CZSL comparison with CLIP ViT-L/14. Best AUC and HM among the listed methods are bold. Our results are means over three runs; other entries are published results. CPF reports its CLIP variant on C-GQA.

| | MIT-States | | UT-Zappos | | C-GQA | |
|---|---|---|---|---|---|---|
| Method | AUC | HM | AUC | HM | AUC | HM |
| *Closed world* | | | | | | |
| CLIP (2021) | 11.0 | 26.1 | 5.0 | 15.6 | 1.4 | 8.6 |
| CoOp (2022) | 13.5 | 29.8 | 18.8 | 34.6 | 4.4 | 17.1 |
| DFSP (t2i) (2023) | 20.6 | 37.3 | 36.0 | 47.2 | 10.5 | 27.1 |
| ProLT (2024) | 21.1 | 38.2 | 36.1 | 49.4 | 11.0 | 27.7 |
| Troika (2024) | 22.1 | 39.3 | 41.7 | 54.6 | 12.4 | 29.4 |
| ClusPro (2025) | 23.8 | 40.7 | 46.6 | 58.5 | 14.9 | 32.8 |
| VP-CMJL (2025) | 23.3 | 40.4 | 47.9 | 58.5 | 16.3 | 34.9 |
| Logic-Troika (2025b) | 23.4 | 40.5 | 45.8 | 57.8 | 15.3 | 33.3 |
| CPF (2025a) | – | – | – | – | 15.4 | 33.6 |
| I2CD (2026) | 23.3 | 40.0 | 46.4 | 57.9 | 16.3 | 35.0 |
| EVA (2026) | 24.0 | 41.0 | **50.2** | **60.2** | **18.8** | **36.9** |
| FlowComposer (Troika) (2026) | 23.5 | 40.2 | 46.8 | 58.6 | 15.9 | 34.0 |
| **SPARC** | **24.8** | **41.8** | 48.6 | 59.5 | 17.4 | 35.8 |
| *Open world* | | | | | | |
| CLIP (2021) | 3.0 | 12.8 | 2.2 | 11.2 | 0.3 | 4.0 |
| CoOp (2022) | 2.8 | 12.3 | 13.2 | 28.9 | 0.7 | 5.5 |
| DFSP (t2i) (2023) | 6.8 | 19.3 | 30.3 | 44.0 | 2.4 | 10.4 |
| Troika (2024) | 7.2 | 20.1 | 33.0 | 47.8 | 2.7 | 10.9 |
| ClusPro (2025) | 9.3 | 23.0 | 39.5 | 54.1 | 3.0 | 11.6 |
| VP-CMJL (2025) | 8.3 | 22.0 | 41.4 | 54.5 | 4.6 | 15.5 |
| Logic-Troika (2025b) | 8.7 | 22.4 | 36.2 | 50.8 | 3.4 | 12.6 |
| CPF (2025a) | – | – | – | – | 3.6 | 13.0 |
| I2CD (2026) | 8.2 | 21.4 | 35.6 | 50.2 | 4.3 | 14.9 |
| EVA (2026) | 9.4 | 22.8 | 40.2 | 54.2 | **5.6** | **17.9** |
| FlowComposer (Troika) (2026) | 7.5 | 20.3 | 35.5 | 51.0 | 3.5 | 12.6 |
| **SPARC** | **9.5** | **23.4** | **41.8** | **56.2** | 4.9 | 16.0 |

Table 2: Cumulative stage ablations. Representation replaces the linear adapter; optimization adds the two object-relative losses; rectification adds Eq. (14). The last row is full SPARC.

| | MIT-States (CW) | | UT-Zappos (HM) | | C-GQA (HM) | |
|---|---|---|---|---|---|---|
| Variant | AUC | HM | CW | OW | CW | OW |
| Linear adapter | 22.1 | 37.5 | 49.7 | 43.2 | 27.5 | 9.8 |
| + Representation | 23.3 | 39.6 | 54.3 | 48.1 | 31.9 | 13.8 |
| + Optimization | 24.0 | 40.8 | 56.8 | 52.4 | 33.7 | 14.9 |
| + Rectification | **24.8** | **41.8** | **59.5** | **56.2** | **35.8** | **16.0** |

**Open world.** With the larger candidate space, SPARC obtains the best listed AUC and HM on MIT-States and UT-Zappos. Its UT-Zappos results reach 41.8 AUC and 56.2 HM, improving over VP-CMJL by 0.4 and 1.7 points. On C-GQA, it improves over CPF by 1.3 AUC and 3.0 HM points, and over VP-CMJL by 0.3 AUC and 0.5 HM points; EVA remains the strongest method on that dataset. The gains over the adapter baseline, examined next, establish that the regulation stages improve recognition while holding the pretrained backbone fixed.

### 4.3 What Each Regulation Stage Contributes

**Cumulative stage comparison.** The starting model uses the frozen CLIP backbone with a linear adapter and the three-branch classification objective. Table 2 successively replaces the linear adapter with complementary-evidence feedback, adds $\mathcal{L}_{sp}$ and $\mathcal{L}_{HGC}$, and enables the object-score correc-

Table 3: Training objectives (MIT-States CW). Representation and inference are fixed.

| Losses added to $\mathcal{L}_{base}$ | AUC | HM |
|---|---|---|
| None | 23.7 | 39.8 |
| $\mathcal{L}_{sp}$ | 24.1 | 41.2 |
| $\mathcal{L}_{HGC}$ | 24.0 | 40.7 |
| Both | **24.8** | **41.8** |

Table 4: Inference correction (MIT-States CW). Representation and training are fixed.

| Correction | AUC | HM |
|---|---|---|
| None | 24.0 | 40.8 |
| Pair alignment, min–max | 24.3 | 41.4 |
| Object branch, z-score | 24.6 | 41.7 |
| Pair alignment, z-score | **24.8** | **41.8** |

tion in Eq. (14). The inference step denotes this rectification. This comparison measures the gain from each stage added to the preceding model.

On MIT-States, the representation stage raises AUC from 22.1 to 23.3 and HM from 37.5 to 39.6. Object-relative training brings these to 24.0 and 40.8; rectification yields 24.8 and 41.8. The gains extend beyond one dataset. Moving from the base model to the full method increases CW HM by 9.8 points on UT-Zappos and 8.3 points on C-GQA. OW HM rises by 13.0 and 6.2 points, respectively. Every added stage improves the reported metric in each setting. The representation stage supplies the first improvement, and the training and ranking controls add gains after those features are in place.

**Training objectives.** Table 3 keeps the feedback adapter and final inference rule fixed. Its CE-only row uses $\mathcal{L}_{base}$, so it differs from the linear-adapter baseline in Table 2. Adding only the object-discounted loss raises MIT-States CW AUC from 23.7 to 24.1 and HM from 39.8 to 41.2. Adding only the specificity margin gives 24.0 AUC and 40.7 HM. Their combination achieves 24.8 and 41.8. Comparing these rows separates competition against alternative compositions from the margin over the correct object's score, and shows that their combination is more effective than either objective alone.

**Inference choices.** Table 4 fixes the representation and training objectives, then varies only the object correction applied to the fused score. "None" is the complete trained model before rectification. "Object branch" substitutes $S_{obj}$ for the cosine alignment computed from $z_{pair}$; "Min–max" keeps the pair-representation alignment but changes its normalization. Pair-representation alignment with standardization achieves the highest AUC and HM. The modest difference from the object-branch reference favors measuring the object component in the representation being ranked. Both alternatives improve on no correction, supporting the usefulness of regulating object-group offsets after training.

**Reading the stage gains.** The final-stage comparison is especially informative because all features and learned classifiers are held fixed. Its improvement follows solely from changed competition between object groups: candidates within the same group retain their relative scores. The earlier representation and objective comparisons concern learning, where attribute and composition evidence can change. Together, these experiments evaluate the distinct roles identified in Eq. (2): constructing useful composition evidence and balancing the candidate groups against which it competes.

### 4.4 Adaptive Fusion, Sensitivity, and Cost

**Specificity margin.** Figure 3 varies $m$ over $\{0.5, 1, 2, 3, 4\}$. On MIT-States, AUC stays between 24.3 and 24.9 and HM between 41.3 and 41.9. C-GQA AUC peaks at $m = 3$, while its highest HM occurs at $m = 0.5$, illustrating the different emphasis of an integrated trade-off metric and the best harmonic-mean point. UT-Zappos reaches its highest AUC and HM at $m = 2$, with larger margins reducing both. The shared default $m = 2$ gives strong performance across the three datasets without selecting a separate margin for each test metric.

**Adaptive evidence weighting.** We group MIT-States attributes into coarse semantic types and average each sample's token-channel gates into a scalar. Table 5 reports their means and standard deviations. Color and shape/size receive higher mean gates, whereas material and state have lower means and greater variation. Larger gates favor the object-related support in Eq. (8). The pattern

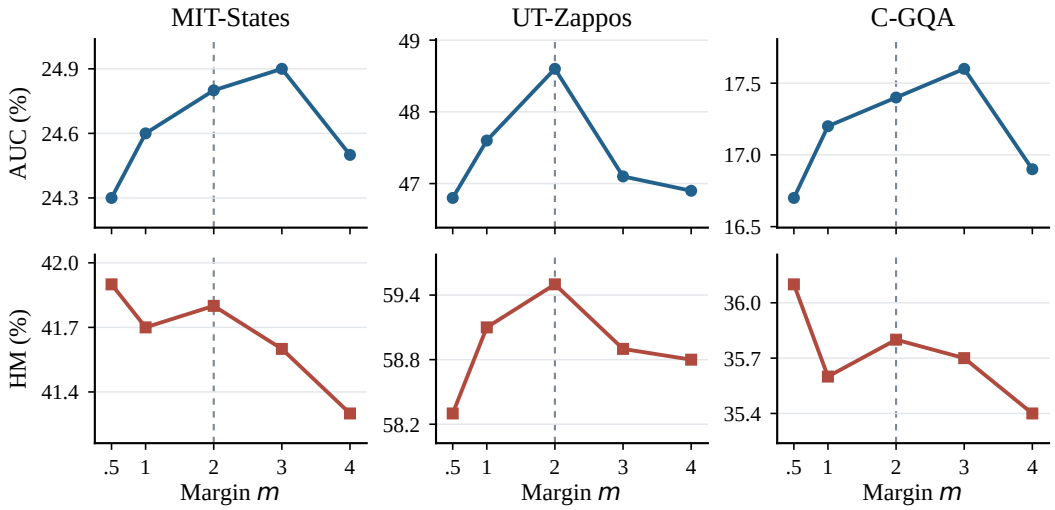


Figure 3: CW sensitivity to the specificity margin $m$. The dashed line marks the shared default $m = 2$; each panel retains its dataset's metric scale.

Table 5: Gate statistics on MIT-States. Variation is measured across samples.

| Attribute type | Mean | Std. |
|---|---|---|
| Color | 0.72 | 0.12 |
| Shape/size | 0.65 | 0.15 |
| Material | 0.45 | 0.21 |
| State | 0.38 | 0.24 |

Table 6: Evaluation cost on MIT-States. Total parameters include frozen CLIP.

| Method | Params (M) | Time (s) |
|---|---|---|
| Linear adapter | 442.46 | 170.58 |
| SPARC | 444.40 | 171.89 |

shows that the adapter uses different mixtures for different visual properties, jointly weighting tokens through the gates and semantic masks.

**Efficiency.** Table 6 compares SPARC with its adapter baseline on MIT-States. The method adds 1.94M parameters and increases evaluation time from 170.58 to 171.89 seconds, approximately 0.8%. Feedback and score regulation therefore retain the efficiency of the underlying adapter framework.

**Scope of regulation.** Regulation targets strong object support that competes with weaker composition evidence. Unambiguous attributes and weak training associations leave less shortcut dependence to regulate. Useful object semantics remain available through the object branch and partial score correction. Our experiments assess compositional transfer on standard CLIP-based benchmarks.

The supplementary material collects the remaining S/U metrics and short derivations of the score properties.

## 5 Conclusion

Object context is essential to compositional recognition, but sparse supervision can make it a substitute for attribute evidence. SPARC combines complementary-evidence feedback, object-relative training, and score rectification to regulate this influence from feature learning to candidate ranking. Experiments across three benchmarks show cumulative gains from the three stages while retaining the efficiency of adaptation on a frozen vision–language backbone.

## AI USE STATEMENT

Generative AI tools were used as supporting tools during the preparation of this work. They assisted with improving the language, structure, and presentation of the manuscript. They were also used for literature and reference verification, and consistency checks among the manuscript text, equations, tables, and figures.

All AI-assisted material was reviewed by the authors before inclusion. References were checked against the original sources, mathematical and technical suggestions were independently verified, and manuscript descriptions were compared with the reported experiments and results. Generative AI tools were not used to generate experimental results or determine the scientific conclusions of the work.

The authors take full responsibility for the final manuscript, including its technical content, scientific claims, and reported results.

## REFERENCES

Shaozhe Hao, Kai Han, and Kwan-Yee K. Wong. Learning Attention As Disentangler for Compositional Zero-Shot Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15315–15324, June 2023.

Zhenqi He, Lin Li, and Long Chen. FlowComposer: Composable Flows for Compositional Zero-Shot Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

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

Phillip Isola, Joseph J Lim, and Edward H Adelson. Discovering states and transformations in image collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1383–1391, 2015.

Chenyi Jiang and Haofeng Zhang. Revealing the proximate long-tail distribution in compositional zero-shot learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 2498–2506, 2024.

Chenyi Jiang, Qiaolin Ye, Shidong Wang, Yuming Shen, Zheng Zhang, and Haofeng Zhang. Mutual Balancing in State-Object Components for Compositional Zero-Shot Learning. *Pattern Recognition*, 152:110451, 2024. ISSN 0031-3203.

Hanjae Kim, Jiyoung Lee, Seongheon Park, and Kwanghoon Sohn. Hierarchical visual primitive experts for compositional zero-shot learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5675–5685, 2023.

Xiangyu Li, Xu Yang, Kun Wei, Cheng Deng, and Muli Yang. Siamese Contrastive Embedding Network for Compositional Zero-Shot Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9326–9335, 2022.

Yong-Lu Li, Yue Xu, Xiaohan Mao, and Cewu Lu. Symmetry and group in attribute-object compositions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11316–11325, 2020.

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

Ishan Misra, Abhinav Gupta, and Martial Hebert. From red wine to red tomato: Composition with context. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1792–1801, 2017.

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

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

Hongyu Qu, Jianan Wei, Xiangbo Shu, and Wenguan Wang. Learning Clustering-based Prototypes for Compositional Zero-shot Learning. In *International Conference on Learning Representations*, 2025.

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

Peng Wu, Qiuxia Lai, Hao Fang, Guo-Sen Xie, Yilong Yin, Xiankai Lu, and Wenguan Wang. A Conditional Probability Framework for Compositional Zero-shot Learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3673–3683, 2025a.

Peng Wu, Xiankai Lu, Hao Hu, Yongqin Xian, Jianbing Shen, and Wenguan Wang. LOGICZSL: Exploring Logic-induced Representation for Compositional Zero-shot Learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 30301–30311, 2025b.

Aron Yu and Kristen Grauman. Fine-grained visual comparisons with local learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 192–199, 2014.

Zhaoquan Yuan, Zining Wang, Yuankang Pan, Ao Luo, Wei Li, Xiao Wu, and Changsheng Xu. I2CD: An Invertible Causal Framework for Compositional Zero-Shot Learning via Disentangle-Compose-Disentangle. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 12295–12303, 2026.

Shiyu Zhang, Cheng Yan, Yang Liu, Chenchen Jing, Lei Zhou, and Wenjun Wang. Learning Visual Proxy for Compositional Zero-Shot Learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2793–2802, October 2025.

Xiao Zhang, Haodong Jing, Yongqiang Ma, and Nanning Zheng. Decoupling primitive with experts: Dynamic feature alignment for compositional zero-shot learning. In *The Fourteenth International Conference on Learning Representations*, 2026.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022.

# APPENDIX

## A SCORE PROPERTIES

The score controls in SPARC separate two decisions: which attribute best matches an object in the image, and which object group contains the strongest composition candidate. This section derives the identities used in Section 3 and explains how they connect the representation, training, and inference stages.

### A.1 RESIDUAL FEEDBACK AND ATTRIBUTE EVIDENCE

Taking the inner product of Eq. (6) with the object-related pooled view gives

$$\langle c_{\text{res}}, \bar{z}_{fg} \rangle = \langle \bar{z}_{bg}, \bar{z}_{fg} \rangle \left( 1 - \frac{\|\bar{z}_{fg}\|^2}{\|\bar{z}_{fg}\|^2 + \epsilon} \right) = \frac{\epsilon \langle \bar{z}_{bg}, \bar{z}_{fg} \rangle}{\|\bar{z}_{fg}\|^2 + \epsilon}. \tag{15}$$

For nonzero $\bar{z}_{fg}$, the residual is orthogonal to that view as $\epsilon$ tends to zero. The operation extracts the part of the complementary evidence that lies outside the shared linear direction. Both pooled views come from the same attribute-evidence matrix $E$: their difference arises from the spatial weights under which this evidence is collected.

The residual serves as a control signal for the gate. The token update itself uses $E$, preserving the spatially varying attribute evidence available before pooling. This distinction matters when an attribute appears only in a small part of the image. Broadcasting the residual makes the complementary information available to every token, while the token queries allow the resulting gates to vary across locations and channels. Evidence at object-related tokens remains in the update through the $M_{fg}$ term.

### A.2 TRAINING DISCOUNT AND GROUP COMPETITION

Let $\widetilde{S}(a, o) = S_{pair}(a, o) - \lambda \, \text{sg}(S_{obj}[o])$. For two attributes paired with the same object,

$$\widetilde{S}(a_1, o) - \widetilde{S}(a_2, o) = S_{pair}(a_1, o) - S_{pair}(a_2, o). \tag{16}$$

The identical object reference cancels. The loss nevertheless changes the training gradients through its normalization across all seen compositions. Writing $\widetilde{p} = \text{softmax}_{\mathcal{C}^s}(\widetilde{S})$ gives the score-level derivatives

$$\frac{\partial \mathcal{L}_{sp}}{\partial S_{pair}[c]} = \widetilde{p}(c) - \mathbf{1}[c = y], \qquad \frac{\partial \mathcal{L}_{sp}}{\partial S_{obj}} = 0. \tag{17}$$

The discount reweights each group's contribution to the softmax denominator by $\exp(-\lambda S_{obj}[o])$. Groups with larger object scores are discounted more strongly. The pair classifier is trained to keep the labeled composition competitive under this reweighting. Each candidate still has its own pair logit and classification gradient. The object reference therefore changes the competition faced during learning, while the pair and attribute supervision supplies the distinctions within a group. The object classifier retains its own supervised loss.

The specificity objective adds a different requirement. It is active when the gap between the labeled pair score and its object reference is below $m$, and it pushes that pair score upward until the margin is met. This comparison concerns the labeled pair and its own object; $\mathcal{L}_{sp}$ also involves alternative compositions through cross-entropy. Their combination asks for a composition that is competitive against alternatives and sufficiently specific relative to its object support.

### A.3 BRANCH FUSION AND INFERENCE MARGINS

For a fixed image and candidate set, write $Z_{pair} = \sum_{y \in \mathcal{Y}} \exp S_{pair}(I, y)$ and $p_{pair} = \text{softmax}_{\mathcal{Y}}(S_{pair})$. Equation (12) satisfies

$$S_{fuse}(I, a, o) = \log[p_{pair}(a, o \mid I) p_{attr}(a \mid I) p_{obj}(o \mid I)] + \log Z_{pair}. \tag{18}$$

Since $\log Z_{pair}$ is constant across candidates, the two expressions produce the same ranking. The pair term evaluates the complete composition, while the primitive terms contribute evidence for its

Table 7: Quantities inside one feedback adapter. Scalar token supports are broadcast across channels when they weight $E$.

| Quantity | Shape | Role |
|---|---|---|
| $X$ | $L \times D$ | Incoming visual tokens. |
| $Q_{\text{raw}}, Q$ | $L \times d$ | Projected queries and their normalized version. |
| $K_{obj}$ | $\lvert\mathcal{O}\rvert \times d$ | Normalized object keys for token routing. |
| $K_{attr}, V_{attr}$ | $\lvert\mathcal{A}\rvert \times d$ | Attribute keys and values. |
| $M_{fg}, M_{bg}$ | $L$ | Complementary object-related supports. |
| $E$ | $L \times d$ | Token-level attribute evidence. |
| $\bar{z}_{fg}, \bar{z}_{bg}, c_{\text{res}}$ | $d$ | Pooled views and their directional residual. |
| $F^{tok}, g$ | $L \times d$ | Broadcast feedback and token-channel gates. |
| $\Delta X$ | $L \times D$ | Update returned to the visual stream. |

attribute and object. The positive object term keeps useful object-recognition evidence in the final decision.

For candidates $y_1 = (a_1, o_1)$ and $y_2 = (a_2, o_2)$,

$$S_{final}(I, y_1) - S_{final}(I, y_2) = S_{fuse}(I, y_1) - S_{fuse}(I, y_2) - \alpha(\hat{b}_{o_1} - \hat{b}_{o_2}). \tag{19}$$

When $o_1 = o_2$, the correction cancels. For different objects, the signed difference of standardized alignments shifts the decision margin. The relevant quantity is the margin between competing candidates: a small correction can change a close decision even when the scores themselves are much larger.

Per-image centering removes a common offset in all object alignments. If every $b_o$ increases by the same value, its mean increases by that value and $\hat{b}_o$ is unchanged. Division by the standard deviation then expresses each object's alignment relative to the spread across objects in that image. This gives $\alpha$ a consistent role as the strength of relative object-group correction. The object-recognition term and the correction use different representations, $z_{obj}$ and $z_{pair}$, respectively: the former supplies positive object evidence, while the latter measures object alignment within the composition stream.

## B Computation and Data Flow

This section follows one image through the model. Table 7 collects the quantities used inside an adapted visual block; $L$ is the number of tokens, $D$ is the incoming visual width, and $d$ is the adapter width.

### B.1 One Adapted Visual Block

The following sequence corresponds to the four operations in Figure 2(b).

1. **Build the two supports.** Project and normalize the token queries and object keys. For each token, take its largest similarity over the object vocabulary, then normalize these responses across tokens to obtain $M_{fg}$. Set $M_{bg} = 1 - M_{fg}$. This routing step uses the object vocabulary before a particular object prediction has been selected.
2. **Extract attribute evidence.** Compare each query with the attribute keys, normalize over attributes, and use the resulting weights to combine the attribute values into $E$. Every token therefore has an attribute-evidence vector before either pooled view is formed.
3. **Pool and residualize.** Normalize each support over tokens and use it to pool $E$. Remove the component of $\bar{z}_{bg}$ aligned with $\bar{z}_{fg}$, giving $c_{\text{res}}$. Transform and normalize this residual to produce the feedback vector $F$.
4. **Gate the token update.** Broadcast $F$ to all token locations and combine it with $Q_{\text{raw}}$ to compute $g$. Weight the token evidence through the two supports and complementary gates, project it back to width $D$, and add the scaled update to $X$.

The supports are complementary before pooling, but each is normalized separately for its own weighted average. Consequently, $\bar{z}_{fg}$ and $\bar{z}_{bg}$ summarize evidence under different spatial emphasis;

neither vector is obtained by subtracting the other from a global pooled feature. This is why both views can contain a shared direction even though their underlying supports sum to one.

**Reading the gate.** For token $i$ and channel $k$, the coefficient multiplying $E_{ik}$ before output projection is $g_{ik}M_{fg,i} + (1 - g_{ik})M_{bg,i}$. A larger gate favors the object-related support, and a smaller gate favors its complement. The effective weight depends on both the gate and the support at that token. The sample-average gate in Table 5 thus summarizes a preference in the fusion mechanism; it is distinct from the fraction of the image assigned to a foreground region. The full token-channel gate retains the variation that this scalar summary averages over.

### B.2 Training and Recognition

The visual encoder contains these adapters in its last three transformer blocks. After the visual forward pass, the model forms $z_{pair}$ for composition and attribute prediction and $z_{obj}$ for the object branch. Training computes the three classification terms in $\mathcal{L}_{base}$, the object-discounted composition loss, and the specificity margin from the same labeled image. Composition cross-entropy uses the seen set $\mathcal{C}^s$, while primitive losses use their attribute and object vocabularies. The encoder parameters stay frozen as the adapters and prediction heads are optimized.

Recognition reuses the learned representations and heads. It computes primitive probabilities, combines them with the pair score for each candidate, and subtracts the standardized object alignment obtained from $z_{pair}$. For a fixed object, this correction is computed once and shared by all its candidate compositions. The training-only losses are absent from this scoring procedure; $\lambda$ and $m$ govern learning, whereas $\alpha$ governs the final ranking correction.

## C Evaluation and Additional Recognition Metrics

### C.1 Candidate Sets and Calibration

Each test image has one target composition. A prediction is correct when both its attribute and object match that label. Seen and unseen accuracies refer to different subsets of test images, distinguished by whether their target composition belongs to the training composition set. Both subsets are classified against the joint evaluation candidate set.

In closed world, that set contains the seen and designated unseen compositions. Open world expands the composition space to $\mathcal{A} \times \mathcal{O}$, introducing additional competing combinations. The primitive vocabularies remain the same. This makes the evaluation sensitive to how evidence is distributed across candidates, beyond whether the two primitives have individually appeared during training.

The calibration sweep trades off predictions of seen and unseen compositions. A convenient notation is $S_\gamma(I, c) = S_{final}(I, c) + \gamma \mathbf{1}[c \in \mathcal{C}^s]$, where increasing $\gamma$ favors seen candidates. Let $s(\gamma)$ and $u(\gamma)$ be the seen and unseen accuracies at the same operating point. The reported summaries are

$$S = \max_\gamma s(\gamma), \qquad U = \max_\gamma u(\gamma), \qquad HM = \max_\gamma \frac{2s(\gamma)u(\gamma)}{s(\gamma) + u(\gamma)}. \tag{20}$$

AUC summarizes the area under the resulting seen–unseen accuracy curve. Best HM describes its strongest balanced operating point; S and U describe separate extremes of the trade-off. They need not be attained at the same value of $\gamma$.

The sweep and SPARC's object correction act on different partitions. Calibration distinguishes seen from unseen compositions. Rectification distinguishes object groups and applies the same shift to all compositions sharing an object. Seen and unseen pairs can coexist within one object group, so the two operations serve different roles. Likewise, checkpoint selection and curve reporting are separate steps: validation loss selects the trained checkpoint, and the calibration sweep summarizes recognition using its fixed scores.

### C.2 Reading the Additional Metrics

Table 8 gives the best seen and unseen accuracies corresponding to Table 1. Their separate maxima should not be combined to reconstruct the reported best HM. For example, SPARC's MIT-States

Table 8: Best seen (S) and unseen (U) accuracies for the comparison in Table 1.

| | MIT-States | | UT-Zappos | | C-GQA | |
|---|---|---|---|---|---|---|
| Method | S | U | S | U | S | U |
| *Closed world* | | | | | | |
| CLIP | 30.2 | 46.0 | 15.8 | 49.1 | 7.5 | 25.0 |
| CoOp | 34.4 | 47.6 | 52.1 | 49.3 | 20.5 | 26.8 |
| DFSP (t2i) | 46.9 | 52.0 | 66.7 | 71.7 | 38.2 | 32.0 |
| ProLT | 49.1 | 51.0 | 66.0 | 70.1 | 39.5 | 32.9 |
| Troika | 49.0 | 53.0 | 66.8 | 73.8 | 41.0 | 35.7 |
| ClusPro | 52.1 | 54.0 | 70.7 | 76.0 | 44.3 | 37.8 |
| VP-CMJL | 51.8 | 52.6 | 71.9 | 76.3 | 46.0 | 40.2 |
| Logic-Troika | 50.8 | 53.9 | 69.6 | 74.9 | 44.4 | 39.4 |
| CPF | – | – | – | – | 44.8 | 39.6 |
| I2CD | 50.6 | 54.1 | 69.1 | 77.1 | 44.1 | 41.2 |
| EVA | 51.2 | 55.0 | 71.2 | 79.6 | 47.1 | 44.6 |
| FlowComposer (Troika) | 51.5 | 53.2 | 71.1 | 74.9 | 44.8 | 40.7 |
| SPARC | 51.7 | 53.9 | 70.1 | 75.6 | 45.3 | 42.6 |
| *Open world* | | | | | | |
| CLIP | 30.1 | 14.3 | 15.7 | 20.6 | 7.5 | 4.6 |
| CoOp | 34.6 | 9.3 | 52.1 | 31.5 | 21.0 | 4.6 |
| DFSP (t2i) | 47.5 | 18.5 | 66.8 | 60.0 | 38.3 | 7.2 |
| Troika | 48.8 | 18.7 | 66.4 | 61.2 | 40.8 | 7.9 |
| ClusPro | 51.2 | 22.1 | 71.0 | 66.2 | 41.6 | 8.3 |
| VP-CMJL | 51.8 | 19.9 | 71.9 | 66.6 | 46.0 | 11.5 |
| Logic-Troika | 50.7 | 21.4 | 69.6 | 63.7 | 43.7 | 9.3 |
| CPF | – | – | – | – | 44.5 | 9.3 |
| I2CD | 50.6 | 20.6 | 69.1 | 62.1 | 44.0 | 10.9 |
| EVA | 50.8 | 23.2 | 71.6 | 66.5 | 46.9 | 13.3 |
| FlowComposer (Troika) | 50.4 | 19.0 | 70.1 | 61.2 | 43.5 | 10.2 |
| SPARC | 49.9 | 22.6 | 72.1 | 66.5 | 47.6 | 10.1 |

CW endpoints are 51.7 and 53.9, while its best HM is 41.8. The endpoints describe performance under different preferences for seen and unseen candidates; HM requires the two accuracies to be achieved together.

The endpoint comparison also explains why leading AUC or HM does not require the highest S and U separately. On MIT-States, SPARC leads the listed methods in both CW AUC and HM even though other methods have larger endpoint accuracies. Its advantage concerns the trade-off between the two populations. Under OW evaluation, C-GQA shows another pattern: SPARC has higher best seen accuracy than EVA, while EVA has higher best unseen accuracy, AUC, and HM. Reporting all four metrics makes these differences visible.

The ablations answer complementary questions. Table 2 measures cumulative changes to the model, Table 3 compares objectives with the feedback representation and inference rule fixed, and Table 4 compares scoring choices after training. Their baseline rows therefore represent different models. The CE-only row in the loss comparison already has the feedback adapter and final scoring rule; the linear-adapter row in the stage comparison is the starting architecture. This distinction explains why those rows have different accuracies.

## D A Worked Example of Object-Group Rectification

Table 9 illustrates how a small correction changes a composition decision. The scores are constructed for this example, with the default $\alpha = 0.2$. Consider an image labeled *green apple*. Before rectification, *green pear* leads by 0.15 score units. The pear group has a larger standardized object alignment, so its correction is larger. After rectification, *green apple* becomes the highest-scoring composition.

Table 9: Illustrative application of Eq. (14) using constructed scores and $\alpha = 0.2$. Bold entries mark the highest score before and after rectification.

| Candidate | $S_{fuse}$ | $\hat{b}_o$ | $S_{final}$ |
|---|---|---|---|
| Green apple | 5.00 | −1.00 | **5.20** |
| Red apple | 4.40 | −1.00 | 4.60 |
| Green pear | **5.15** | 1.00 | 4.95 |
| Red pear | 4.10 | 1.00 | 3.90 |

Two features of the example carry over to the scoring rule. The apple candidates retain their 0.60 gap, and the pear candidates retain their 1.05 gap: attribute ordering within each object group is preserved. Across groups, the 0.40 difference in corrections exceeds the original 0.15 winning margin. The scale that matters for this decision is therefore the gap between the leading candidates.

Choosing *green* over *red* within the apple group remains the responsibility of the composition and attribute evidence learned during training. Rectification then compares that group's strongest candidate with the strongest candidates from other objects. This division of labor connects the feedback adapter to the final ranking: the adapter and objectives improve the evidence available for each composition, and the inference control adjusts the object-dependent offsets under which that evidence competes.

## E Design Choices and Reproducibility

**Using the object vocabulary for routing.** The strongest token–object alignment in Eq. (3) supplies a vocabulary-level semantic response. Taking the maximum allows different tokens to align with different object prototypes and produces supports before the final object decision. The complementary support keeps lower-alignment locations available for attribute evidence. This design is useful when recognizing the object is easier than determining its current property: the model can use object semantics for organization while collecting attribute evidence from both supports.

**Preserving context for composition.** An attribute's appearance depends on its object: the visual evidence for *old*, for example, differs between a building and an item of clothing. Object context therefore remains useful for interpreting attributes. SPARC retains this information in its semantic routing, primitive supervision, and positive object term at inference. The regulation stages address the additional problem created by sparse composition coverage, where a familiar object can make a familiar attribute association competitive despite weak evidence for that attribute in the current image.

**Computation over vocabularies.** Inside an adapter, token–object matching requires $O(L|\mathcal{O}|d)$ operations, and attribute-evidence extraction requires $O(L|\mathcal{A}|d)$. The two pooled vectors and their residual then require no matching over attribute–object pairs. This keeps token-level routing separate from the potentially much larger composition candidate space. At inference, the correction is an object-length vector shared across candidates, while pair classification supplies the composition-specific scores. These operations explain the structure of the added computation measured in Table 6.

**Experimental procedure.** The reported runs use frozen CLIP ViT-L/14, AdamW with learning rate $10^{-5}$, batch size 32, and an NVIDIA A6000. Training lasts at most 30 epochs, stopping after five consecutive epochs without a decrease in validation loss and retaining the checkpoint with the lowest validation loss. Recognition results average three independent runs. The shared regulation settings are $\lambda = 0.1$, $\alpha = 0.2$, and $m = 2.0$, except in the corresponding parameter analysis. Table 5 instead reports variation of the gate across samples, as described in Section 4.4.