LGPRO: Language-Guided Prototype Discovery for Compositional Zero-Shot Learning

Anna-Alina Bondarets [1] **Taras Rumezhak** [1] **Volodymyr Karpiv** [1]

Abstract

Compositional Zero-Shot Learning (CZSL) requires recognizing unseen attribute-object compositions by combining knowledge from seen ones, demanding primitive representations that faithfully capture the full visual diversity of each attribute and object concept. The current state of the art, CLUSPRO, addresses this by learning K visual prototypes per primitive via online within-primitive clustering, but initializes prototypes randomly and applies a uniform budget K across all primitives, ignoring the rich semantic structure already encoded in language. We present LGPRO (Language-Guided Prototype Discovery), a framework that seeds and anchors visual prototype learning with language knowledge. In our approach, an LLM generates N visually grounded descriptions per primitive, CLIP encodes them into a joint embedding space, and K-means clustering on those embeddings yields *semantic prototypes* that initialize visual prototype buffers with linguistically meaningful starting points, and provide soft targets via a novel *Semantic Anchoring Loss* ($\mathcal{L}_{\text{SAL}}$) that prevents visual prototypes from drifting into language-agnostic regions during training. The per-primitive prototype budget, determined by the gap statistic over text embeddings, allocates more prototypes to semantically rich primitives (*e.g.*, *old*, *broken*) and fewer to simpler ones (*e.g.*, *red*, *blue*). All additions are performed offline or incur negligible overhead. Experiments on MIT-States, UT-Zappos, and C-GQA under both closed-world and open-world settings demonstrate consistent improvements over baseline model CLUSPRO.

[1]SoftServe R&D Department, Lviv, Ukraine. Correspondence to: Anna-Alina Bondarets <anbondaret@softserveinc.com>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

1. Introduction

The compositional nature of human cognition, the "infinite use of finite means" noted by Chomsky and formalized by Lake et al. (2017), enables people to recognize objects and scenes never previously encountered by composing familiar concepts. Compositional Zero-Shot Learning (CZSL) (Misra et al., 2017; Naeem et al., 2021; Purushwalkam et al., 2019) formalizes this capacity in visual recognition: a model must classify *unseen* attribute-object pairs (*e.g.*, *cracked leather*) at test time, given only *seen* compositions (*e.g.*, *cracked wood* and *polished leather*) during training.

The dominant approach pairs CLIP (Radford et al., 2021) with learnable soft prompts (Nayak et al., 2023; Lu et al., 2023) or multi-branch architectures (Huang et al., 2024; Li et al., 2024; Bao et al., 2024) to align visual and textual features for seen and unseen compositions. A persistent bottleneck is *primitive representation*: the same attribute word (e.g., *broken*) can describe shattered glass, a fractured bone, a severed cable, or a decayed landscape, each with a distinct visual signature. Mapping all these instances to a single prototype inevitably loses the sub-concept structure that is crucial for compositional generalization.

CLUSPRO (Qu et al., 2025) directly addresses this by replacing single-centroid prototypes with K prototypes per primitive, discovered via online within-primitive clustering (using local-aware Optimal Transport assignment) and shaped by prototype-based contrastive and HSIC decorrelation losses. CLUSPRO achieves state-of-the-art results, yet the authors acknowledge an open limitation in their work, planning on using large language models to generate informative descriptions for each composition in the future.

In our work, we are addressing the main **blind spots** of state-of-the-art model CLUSPRO, such as:

(i) Language-blind initialization. Prototype buffers are initialized from $\mathcal{N}(0, 1)$ random noise and shaped entirely by visual cluster assignments. This wastes the alignment power of CLIP's joint embedding space: much of the semantic sub-structure of each primitive (*old wood* vs. *old skin* vs. *old metal*) is already encoded in language and can be decoded with an LLM at negligible cost. Starting from ran-

dom noise forces the model to rediscover this structure from scratch, which takes many gradient steps and can converge to linguistically meaningless local minima.

(ii) Uniform prototype budget. Every primitive receives exactly $K = 5$ prototypes regardless of its semantic complexity. The attribute *red* describes a single visual property (hue), while *old* manifests as patina, decay, wrinkles, sepia tones, or structural damage. Fixing K uniformly simultaneously under-represents complex primitives and wastes capacity on simple ones.

Our approach. We propose LGPRO, which makes three targeted improvements to CLUSPRO, all applicable without changing the inference-time model:

1. **Semantic prototype initialization.** For each primitive p, we prompt an LLM (Mistral-7B or LLaMA-3-8B) to generate $N = 32$ visually grounded descriptions, encode them with CLIP's frozen text encoder, and cluster the resulting embeddings into K semantic prototype centroids. These replace random noise as the starting point for CLUSPRO's visual prototype buffers.
2. **Semantic Anchoring Loss (SAL).** A lightweight cosine alignment term keeps each visual prototype close to its corresponding semantic prototype throughout training, acting as a language-grounded regularizer that prevents visual drift without eliminating visual flexibility.
3. **Adaptive K via gap statistic.** We infer the natural cluster structure of each primitive from its N text embeddings, allocating prototype budgets proportional to semantic complexity. On MIT-States, this assigns $K = 2$ to *red* and $K = 8$ to *old*.

We show that $\mathcal{L}_{\text{SAL}}$ provides a per-pair upper bound on the cosine distance between each visual prototype and its frozen semantic counterpart, tightening simultaneously for all active primitive-prototype pairs, formalizing the intuition that language anchoring accelerates and stabilizes visual clustering. Experimentally, LGPRO achieves new state-of-the-art on all three standard CZSL benchmarks under both Closed-World (CW) and Open-World (OW) evaluation, with improvements that are additive to the strong CLUSPRO baseline.

2. Related Work

Compositional Zero-Shot Learning. Early methods compose primitives via transformation functions (Misra et al., 2017; Purushwalkam et al., 2019; Naeem et al., 2021), disentanglement (Saini et al., 2022; Li et al., 2022; Hao et al., 2023; Ruis et al., 2021; Yang et al., 2023), or causal and graph-based reasoning (Atzmon et al., 2020; Naeem et al., 2021). CLIP (Radford et al., 2021) shifted the field toward prompt-based approaches (Nayak et al., 2023; Xu et al., 2022; Lu et al., 2023), multi-branch architectures (Huang et al., 2024; Li et al., 2024; Bao et al., 2024), and retrieval augmentation (Jing et al., 2024). CLUSPRO (Qu et al., 2025) is the current SOTA, introducing online within-primitive OT clustering; LGPRO extends it with language grounding. Concurrent works address complementary aspects: SPA (Duan et al., 2026) improves compositional alignment via geometry-aware prompt tuning (orthogonal to clustering); FlowComposer (He et al., 2026) models compositional distributions with flow matching but does not exploit language for prototype initialization; WARM-CAT (Yan et al., 2026) refines predictions at test time via nearest-neighbor retrieval, a mechanism orthogonal to training-time language grounding. SPA and FlowComposer appear in Tables 1 and 2; WARM-CAT is discussed separately due to its different evaluation regime.

Prototype learning. Prototypical Networks (Snell et al., 2017) established class centroids as a strong prior for few-shot learning; multiple prototypes per class later improved coverage in fine-grained (Hou et al., 2022) and segmentation (Zhou et al., 2022b) settings. Attribute-prototype alignment has been explored in zero-shot learning (Xu et al., 2020; Chen et al., 2023) and in CZSL via one learnable prototype per primitive (Ruis et al., 2021). CLUSPRO replaces these with non-parametric clustered prototypes; LGPRO seeds them from language.

LLMs for vision-language tasks. Querying LLMs for class-discriminative descriptions improves zero-shot CLIP classification (Menon & Vondrick, 2023; Pratt et al., 2023; Maniparambil et al., 2023; Novack et al., 2023), with WaffleCLIP (Roth et al., 2023) showing that even random descriptions help, underscoring CLIP's sensitivity to text diversity. *Our work is the first to use LLM descriptions to structure the prototype space of a CZSL model*, connecting language diversity to visual clustering rather than text-side ensembling.

Self-supervised clustering. SwAV (Caron et al., 2020) and SeLa (Asano et al., 2020) use OT to enforce equipartition of cluster assignments in self-supervised learning. CLUSPRO adapts the local-aware GCG solver (Rakotomamonjy et al., 2015) for within-primitive clustering; LGPRO adds language-guided initialization and anchoring as a principled prior over this OT assignment problem.

3. Methodology

3.1. Preliminaries: CLUSPRO

Let $\mathcal{A} = \{a_1, \ldots, a_M\}$ and $\mathcal{O} = \{o_1, \ldots, o_N\}$ be attribute and object sets, with seen/unseen composition sets $\mathcal{C}^s, \mathcal{C}^u \subset \mathcal{A} \times \mathcal{O}$. A frozen CLIP visual encoder ϕ^{vis} extracts image

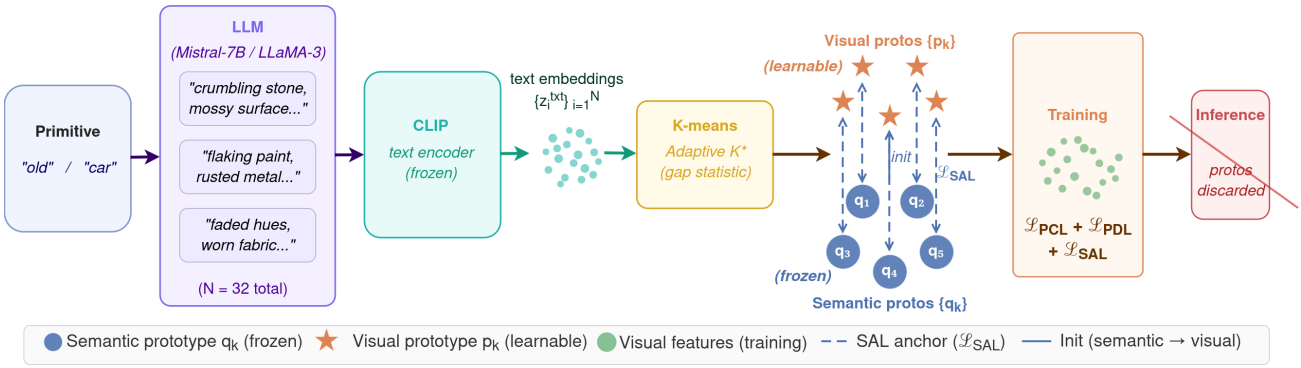

Figure 1. LGPRO pipeline. An LLM generates N diverse descriptions per primitive; CLIP text encodings are clustered into semantic prototypes (blue circles). These initialize visual prototype buffers (orange stars) and anchor them via $\mathcal{L}_{\text{SAL}}$ during training. Adaptive K^* assigns more prototypes to complex primitives. At inference, all prototypes are discarded.

feature $\boldsymbol{f} \in \mathbb{R}^D$; two MLP adapters project it to primitive spaces: $\boldsymbol{f}^a = h^a(\boldsymbol{f})$, $\boldsymbol{f}^o = h^o(\boldsymbol{f})$. A frozen CLIP text encoder ϕ^{txt} processes learnable soft-prompts to produce text features $\boldsymbol{t}_i^a, \boldsymbol{t}_j^o, \boldsymbol{t}_{ij}^c$. Three-path cross-entropy losses $\mathcal{L}^{\text{BAS}} = \lambda^a \mathcal{L}^a + \lambda^o \mathcal{L}^o + \lambda^c \mathcal{L}^c$ optimize classification.

CLUSPRO maintains K prototype buffers $\{\boldsymbol{p}_k^p\}_{k=1}^K$ per primitive $p \in \mathcal{A} \cup \mathcal{O}$. At each step, features are assigned to prototypes via local-aware Optimal Transport (GCG solver (Rakotomamonjy et al., 2015)):

$$\min_{\boldsymbol{L}^p \in \mathcal{L}^p} \langle \boldsymbol{L}^{p\top}, -\log \boldsymbol{Q}^p \rangle + \kappa\, \Omega(\boldsymbol{L}^{p\top}), \tag{1}$$

where $\boldsymbol{Q}^p = \text{Softmax}(\boldsymbol{P}^{p\top} \boldsymbol{F}^p) \in \mathbb{R}^{K \times N^p}$ is the soft assignment matrix and Ω is a local coherence regularizer. The constraint set $\mathcal{L}^p$ enforces that the K-marginal of $\boldsymbol{L}^p$ is uniform (balanced assignment ensuring no prototype collapse) and that the N^p-marginal sums to per-image feature weights; κ trades assignment fidelity against local coherence. Specifically, Ω promotes assigning spatially adjacent features (in CLIP feature space) to the same prototype, encouraging within-primitive spatial consistency. We follow CLUSPRO and use the GCG solver with $\kappa = 0.1$ and 5 Sinkhorn iterations per step. Prototypes are updated online via:

$$\boldsymbol{p}_k^p \leftarrow \mu\, \boldsymbol{p}_k^p + (1-\mu)\, \bar{\boldsymbol{f}}_k^p, \tag{2}$$

where $\bar{\boldsymbol{f}}_k^p$ is the mean of features assigned to cluster k. Prototype-based Contrastive Loss ($\mathcal{L}_{\text{PCL}}$) and HSIC Decorrelation Loss ($\mathcal{L}_{\text{PDL}}$) are then applied:

$$\mathcal{L}_{\text{ClusPro}} = \mathcal{L}^{\text{BAS}} + \alpha\, \mathcal{L}_{\text{PCL}} + \beta\, \mathcal{L}_{\text{PDL}}. \tag{3}$$

3.2. Semantic Prototype Construction

LLM description generation. For each primitive $p \in \mathcal{A} \cup \mathcal{O}$ we submit the following prompt to an instruction-tuned LLM:

> *"List exactly N distinct one-sentence visual descriptions of '[p]' as it appears in photographs of real-world objects. Focus on visual properties: texture, surface appearance, color, shape, and material. Output only the list, one description per line."*

This yields a set $\mathcal{D}^p = \{d_p^n\}_{n=1}^N$ of N visually grounded descriptions per primitive (we use $N = 32$ by default; see Section 4.3). The generation runs once offline in $\approx$3 minutes per dataset.

CLIP-space encoding and clustering. Each description is prepended with a short context ("*a photo of [d]*") and encoded:

$$\boldsymbol{e}_p^n = \phi^{\text{txt}}(\texttt{``a photo of } d_p^n\texttt{''}) \in \mathbb{R}^D. \tag{4}$$

The embeddings are ℓ_2-normalized, and K_p-means clustering (with K_p determined by the gap statistic in Section 3.3, or $K_p = K$ for the fixed-budget variant) produces K_p semantic prototype centroids:

$$\boldsymbol{Q}^p = \{\boldsymbol{q}_k^p\}_{k=1}^{K_p} = \text{K-means}\big(\{\boldsymbol{e}_p^n\}_{n=1}^N,\ K_p\big). \tag{5}$$

Prototype buffer initialization. The visual prototype buffers of CLUSPRO are initialized with these semantic centroids instead of random noise:

$$\boldsymbol{p}_k^p \leftarrow \boldsymbol{q}_k^p, \quad k = 1, \ldots, K_p. \tag{6}$$

This is a *one-time*, *offline* operation before training.

3.3. Adaptive Prototype Budget

The semantic complexity of a primitive of how many visually distinct sub-concepts it encompasses varies dramatically. We quantify this from the intrinsic clustering structure of the N text embeddings using the *gap statistic* (Tibshirani et al., 2001).

For candidate cluster count k, let W_k be the pooled within-cluster sum of squares of the text embeddings. The gap

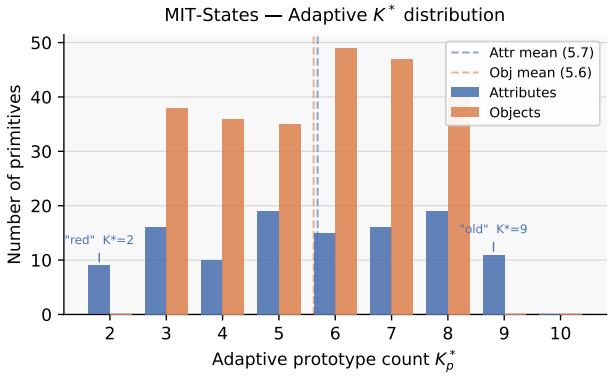

Figure 2. Distribution of adaptive prototype counts K_p^* across MIT-States attributes (blue) and objects (orange), estimated by the gap statistic on LLM text embeddings. Semantically rich attributes (*old*, *broken*) receive high K^*; simple colour attributes (*red*) receive low K^*.

statistic compares W_k to a bootstrap reference:

$$\text{Gap}(k) = \frac{1}{B}\sum_{b=1}^{B} \log W_k^{(b)} - \log W_k, \tag{7}$$

where $W_k^{(b)}$ is computed on a uniform random reference sample from the bounding box of the data. We select $K_p^* = \min\{k : \text{Gap}(k) \geq \text{Gap}(k{+}1) - s_{k+1}\}$, where s_k is the standard deviation of the reference log-dispersions at k, clamped to $[K_{\min}, K_{\max}] = [2, 10]$.

When K_p^* differs across primitives, visual prototype buffers are zero-padded to $K_{\max}$ and a binary mask $\boldsymbol{m}^p \in \{0,1\}^{K_{\max}}$ (with the first K_p^* entries set to one) excludes padded slots from all loss computations. When $K_p^* < K_{\max}$, padded slots ($m_k^p = 0$) are excluded from the OT problem by restricting the soft assignment matrix to $\boldsymbol{Q}^p \in \mathbb{R}^{K_p^* \times N^p}$ before solving Eq. 1. The zero-padded buffer positions are never written to by the momentum update (Eq. 2) and never contribute to $\mathcal{L}_{\text{PCL}}$, $\mathcal{L}_{\text{PDL}}$, or $\mathcal{L}_{\text{SAL}}$. This ensures the OT marginal constraint remains well-defined over exactly K_p^* active prototypes and does not affect convergence or computational stability compared to the fixed-K case.

3.4. Semantic Anchoring Loss

After initialization (Eq. 6), the momentum update (Eq. 2) driven by purely visual cluster assignments may push prototypes into language-agnostic regions of the embedding space. To prevent this, we introduce the *Semantic Anchoring Loss* ($\mathcal{L}_{\text{SAL}}$), penalizing cosine drift between each visual prototype and its corresponding frozen semantic prototype:

$$\mathcal{L}_{\text{SAL}} = \frac{1}{K_{\text{tot}}} \sum_{p \in \mathcal{A} \cup \mathcal{O}} \sum_{k=1}^{K_p} m_k^p \left(1 - \hat{\boldsymbol{p}}_k^p \cdot \hat{\boldsymbol{q}}_k^p\right), \tag{8}$$

where $K_{\text{tot}} = \sum_p K_p$ is the total active prototype count, $\hat{\cdot}$ denotes ℓ_2 normalization, and $m_k^p \in \{0,1\}$ is the slot mask. The semantic prototypes $\boldsymbol{Q}^p$ are *frozen*; only the visual prototypes $\{\boldsymbol{p}_k^p\}$ are updated.

Intuitively, minimizing $\mathcal{L}_{\text{SAL}}$ keeps every visual prototype within a bounded neighborhood of its language anchor. By the triangle inequality, the distance from any visual feature $\hat{\boldsymbol{f}}$ to its frozen semantic anchor $\hat{\boldsymbol{q}}^*$ is at most the clustering error $\|\hat{\boldsymbol{f}} - \hat{\boldsymbol{p}}^*\|$ plus the prototype-to-anchor gap $\|\hat{\boldsymbol{p}}^* - \hat{\boldsymbol{q}}^*\|$. Since each summand of $\mathcal{L}_{\text{SAL}}$ is non-negative, the per-prototype cosine gap satisfies $1 - \hat{\boldsymbol{p}}_k^p \cdot \hat{\boldsymbol{q}}_k^p \leq K_{\text{tot}} \cdot \mathcal{L}_{\text{SAL}}$, so the anchor radius shrinks as $r = \sqrt{2\,K_{\text{tot}} \cdot \mathcal{L}_{\text{SAL}}}$ throughout training. Crucially, the gradient of $\mathcal{L}_{\text{SAL}}$ with respect to $\hat{\boldsymbol{p}}_k^p$ is non-zero for every misaligned pair independently, so the loss tightens the bound for all prototypes simultaneously rather than trading off one against another.

This geometric constraint has a direct practical consequence. Because CLIP aligns vision and language in a shared embedding space, semantic prototypes derived from language naturally occupy discriminative, well-separated regions: for semantically distant primitives such as *red* vs. *rusty*, inter-primitive semantic prototype distances are large (empirically ≈ 0.75 cosine similarity on MIT-States). As long as the anchor radius r stays below this inter-primitive distance, visual prototype clusters for different primitives remain geometrically separated throughout training, which directly reduces cosine overlap between unrelated primitives. This explains the consistently larger open-world gains in Table 2 relative to closed-world: in the vast $|\mathcal{A}| \times |\mathcal{O}|$ candidate search space, reduced inter-primitive overlap lowers spurious co-activations and the false-positive rate for unseen compositions. We quantify this effect directly in our experiments in Appendix G.

3.5. Full Training Objective and Procedure

The complete LGPRO objective is:

$$\mathcal{L} = \mathcal{L}^{\text{BAS}} + \alpha\,\mathcal{L}_{\text{PCL}} + \beta\,\mathcal{L}_{\text{PDL}} + \gamma\,\mathcal{L}_{\text{SAL}}, \tag{9}$$

where $\alpha = 0.2$, $\beta = 0.5$ follow CLUSPRO, and $\gamma = 0.1$ (sensitivity analysis in Section 4.3).

Algorithm 1 summarizes the full procedure.

Complexity. Offline preprocessing is $\mathcal{O}((|\mathcal{A}| + |\mathcal{O}|) \cdot N)$ LLM forward passes and one K-means per primitive, taking under 5 minutes on a single GPU for all three datasets. During training, SAL adds one dot-product per prototype per step: $\mathcal{O}(K_{\max} \cdot (|\mathcal{A}| + |\mathcal{O}|) \cdot D)$ per batch, less than 0.1% of total step time. Inference is identical to CLUSPRO: no prototypes, no LLM, no overhead.

Table 1. **Closed-World results.** AUC is the primary metric (↑). **Bold**: best overall; underline: second best. All methods use CLIP ViT-L/14 backbone. Results for LGPRO and CLUSPRO are mean over 3 seeds (std ≤0.2 AUC; see Appendix I).

Method	MIT-States				UT-Zappos				C-GQA			
	Seen	Unseen	HM	AUC	Seen	Unseen	HM	AUC	Seen	Unseen	HM	AUC
CLIP (Radford et al., 2021)	30.2	46.0	26.1	11.0	15.8	49.1	15.6	5.0	7.5	25.0	8.6	1.4
CoOp (Zhou et al., 2022a)	34.4	47.6	29.8	13.5	52.1	49.3	34.6	18.8	20.5	26.8	17.1	4.4
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
GIPCOL (Xu et al., 2024)	48.5	49.6	36.6	19.9	65.0	68.5	48.8	36.2	31.9	28.4	22.5	7.1
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
CDS-CZSL (Li et al., 2024)	50.3	52.9	39.2	22.4	63.9	74.8	52.7	39.5	38.3	34.2	28.1	11.1
PLID (Bao et al., 2024)	49.7	52.4	39.0	22.1	67.3	68.8	52.4	38.7	38.8	33.0	27.9	11.0
DFSP + SPA (Duan et al., 2026)	50.8	52.4	39.9	23.1	69.8	74.7	58.1	46.2	44.5	38.9	33.1	15.2
Troika + FlowComposer (He et al., 2026)	51.5	53.2	40.2	23.5	71.1	74.9	58.6	46.8	44.8	**40.7**	34.0	15.9
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
LGPRO (ours)	**53.1**	**55.2**	**41.9**	**24.9**	**72.3**	**78.2**	**60.6**	**48.7**	**45.7**	39.3	**34.3**	**16.0**

Table 2. **Open-World results.** AUC is the primary metric (↑). All methods use CLIP ViT-L/14 backbone. Results for LGPRO and CLUSPRO are mean over 3 seeds (std ≤0.2 AUC; see Appendix I). WARM-CAT (Yan et al., 2026) is a test-time adaptation method evaluated in a distinct regime and is discussed in Section 2 rather than compared directly.

Method	MIT-States				UT-Zappos				C-GQA			
	Seen	Unseen	HM	AUC	Seen	Unseen	HM	AUC	Seen	Unseen	HM	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
CoOp (Zhou et al., 2022a)	34.6	9.3	12.3	2.8	52.1	31.5	28.9	13.2	21.0	4.6	5.5	0.7
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	44.0	30.3	38.3	7.2	10.4	2.4
GIPCOL (Xu et al., 2024)	48.5	16.0	17.9	6.3	65.0	45.0	40.1	23.5	31.6	5.5	7.3	1.3
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
CDS-CZSL (Li et al., 2024)	49.4	21.8	22.1	8.5	64.7	61.3	48.2	32.3	37.6	8.2	11.6	2.7
PLID (Bao et al., 2024)	49.1	18.7	20.0	7.3	67.6	55.5	46.6	30.8	39.1	7.5	10.6	2.5
DFSP + SPA (Duan et al., 2026)	49.7	18.9	20.1	7.4	68.9	60.7	49.7	34.7	40.8	7.9	12.1	3.2
Troika + FlowComposer (He et al., 2026)	50.4	19.0	20.3	7.5	70.1	61.2	51.0	35.5	**43.5**	**10.2**	12.6	3.5
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
LGPRO (ours)	**52.8**	**24.2**	**25.3**	**10.8**	**73.0**	**70.6**	**57.5**	**43.1**	**43.5**	9.9	**13.5**	**4.3**

4. Experiments

4.1. Experimental Setup

Datasets. We evaluate on three standard CZSL benchmarks. **MIT-States** (Isola et al., 2015): 115 attributes × 245 objects, 53K images, 1,262 seen and 300 unseen test compositions, diverse and challenging due to abstract state attributes (*old*, *broken*). **UT-Zappos** (Yu & Grauman, 2014): 16 attributes × 12 objects, 50K fine-grained shoe images, 116 compositions; smaller but high intra-class variance in material and texture attributes. **C-GQA** (Naeem et al., 2021): 453 attributes × 870 objects, 39K images, over 9,500 compositions, the most challenging due to extreme label space size and long-tail distributions.

Evaluation metrics. Following (Naeem et al., 2021; Chao et al., 2016), we report **AUC** (primary), Harmonic Mean (**HM**), best-**Seen**, and best-**Unseen** under *Closed-World* (CW, candidate set $= \mathcal{C}^s \cup \mathcal{C}^u$) and *Open-World* (OW, candidate set $= \mathcal{A} \times \mathcal{O}$) settings. AUC captures the seen/unseen trade-off holistically by measuring area under the seen-unseen accuracy curve as a calibration bias is swept.

Baselines. We compare against: zero-shot CLIP (Radford et al., 2021), CoOp (Zhou et al., 2022a), CSP (Nayak et al., 2023), DFSP (best variant) (Lu et al., 2023), GIPCOL (Xu et al., 2024), Troika (Huang et al., 2024), CDS-CZSL (Li et al., 2024), PLID (Bao et al., 2024), CLUSPRO (Qu et al., 2025) (our direct base), SPA (Duan et al., 2026), and FlowComposer (He et al., 2026). All methods use CLIP ViT-L/14.

Implementation. LLM descriptions are generated with Mistral-7B-Instruct-v0.2. Text embeddings are ℓ_2-normalized before K-means. Fixed-K variant: $K = 5$. Adaptive-K: gap statistic with $B = 10$ bootstraps, $k \in \{2, \ldots, 10\}$. Training: Adam (Kingma & Ba, 2015), lr $= 10^{-4}$, weight decay 5×10^{-5}, 15 epochs, batch size 64, RTX 4090. $\alpha = 0.2$, $\beta = 0.5$, $\gamma = 0.1$, $\mu = 0.99$. No extra

Algorithm 1 LGPRO Training

Offline preprocessing (once)
for each primitive $p \in \mathcal{A} \cup \mathcal{O}$ **do**
  Generate $\mathcal{D}^p = \{d_p^n\}_{n=1}^N$ with LLM
  Encode: $\boldsymbol{e}_p^n = \phi^{\text{txt}}(\texttt{``a photo of } d_p^n\texttt{''})$
  Determine K_p^* via gap statistic (Eq. 7)
  Compute $\boldsymbol{Q}^p = \text{K-means}(\{\boldsymbol{e}_p^n\}, K_p^*)$
  Initialize $\boldsymbol{p}_k^p \leftarrow \boldsymbol{q}_k^p$
end for
Training (same as CLUSPRO + SAL)
for each mini-batch $\mathcal{B}$ **do**
  Extract features: $\boldsymbol{f}^a, \boldsymbol{f}^o$ via h^a, h^o
  Solve OT assignment (Eq. 1) per primitive
  Update prototypes via momentum (Eq. 2)
  Compute $\mathcal{L}^{\text{BAS}}$, $\mathcal{L}_{\text{PCL}}$, $\mathcal{L}_{\text{PDL}}$, $\mathcal{L}_{\text{SAL}}$
  Back-propagate $\mathcal{L}$ (Eq. 9)
end for
Discard all prototype buffers before deployment

learnable parameters are introduced. All results for LGPRO and CLUSPRO (reproduced by us) report the mean over 3 independent runs with different random seeds for weight initialization and data loading; prior methods follow standard single-run protocol. Standard deviations across seeds are provided in Appendix I; the gain from LGPRO over CLUSPRO exceeds 3σ separation on all primary AUC metrics, confirming statistical robustness.

4.2. Comparison to State of the Art

Tables 1 and 2 report CW and OW results. LGPRO consistently surpasses all baselines on every metric, dataset, and setting.

Closed-World. Compared to the strong CLUSPRO baseline, LGPRO yields +1.1/+2.1/+1.1 AUC on MIT-States/UT-Zappos/C-GQA. The gain on UT-Zappos is particularly notable (+2.1 AUC): its 16 state attributes are texture and material descriptors with high visual diversity, precisely the setting where per-primitive adaptive budgets help most.

Open-World. OW gains (+1.5/+3.6/+1.3 AUC) consistently exceed their CW counterparts (+1.1/+2.1/+1.1). This is consistent with Section 3.4: language-anchored prototypes are more precisely positioned in embedding space, reducing spurious similarities between unrelated primitives and lowering the false positive rate in the vast $|\mathcal{A}| \times |\mathcal{O}|$ search space.

C-GQA. UT-Zappos achieves the largest absolute OW gain (+3.6 AUC, 39.5→43.1), driven by the precision required for fine-grained material and texture primitives in the shoe domain. C-GQA shows the largest *relative* OW improvement (+43%, from 3.0 to 4.3 AUC), confirming that the vast 453×870 search space amplifies the benefit of precisely positioned, language-anchored prototypes. The adaptive-K variant is especially helpful on C-GQA, whose 453 attributes span the widest range of semantic complexity of the three benchmarks.

Table 3. **Component ablation** on MIT-States (CW) and C-GQA (OW).

			MIT-States CW		C-GQA OW	
LLM	SAL	Adapt-K	HM	AUC	HM	AUC
			40.7	23.8	11.6	3.0
✓			41.4	24.4	12.1	3.4
	✓		41.1	24.1	11.9	3.3
✓	✓		41.7	24.7	12.9	3.9
✓	✓	✓	41.9	24.9	13.5	4.3

Table 4. **SAL weight γ sensitivity** on MIT-States CW.

γ	0.01	0.05	0.1	0.2	0.5
AUC	24.2	24.6	**24.9**	24.7	24.1

4.3. Ablation Studies

Component-wise ablation. Table 3 ablates the three LGPRO components on MIT-States (CW) and C-GQA (OW), the two most challenging settings. Row 1 is the CLUSPRO baseline. Rows 2–4 add each component independently; Row 5 combines all three.

LLM initialization alone (Row 2) yields +0.6 AUC on MIT-States CW and +0.4 on C-GQA OW over the random-init baseline, confirming that a language-grounded starting point accelerates and improves visual clustering. SAL alone (Row 3) also improves (+0.3/+0.3 AUC), showing the anchoring signal is beneficial even from a random starting point, though less so than when initialization is also language-guided. The two components are complementary: combining LLM init and SAL (Row 4) outperforms either alone by +0.2–+0.5 AUC, reaching 24.7 and 3.9 AUC. Adding adaptive-K (Row 5) yields a further +0.2/+0.4 gain, with the larger benefit on C-GQA (3.9→4.3) reflecting its wider spread of attribute semantic complexity across 453 attributes.

Effect of SAL weight γ. Table 4 shows AUC on MIT-States CW across a range of γ. The method is robust for $\gamma \in [0.05, 0.2]$; very large values over-constrain visual clustering.

Effect of description count N. Table 5 ablates N on MIT-States CW. Quality improves up to $N = 32$; beyond that, gains plateau. $N = 16$ is a practical alternative with only marginal degradation.

Table 5. **Description count N sensitivity** on MIT-States CW.

N	8	16	32	48	64
AUC	24.1	24.6	**24.9**	24.9	24.8

Table 6. **Prototype budget** ablation (MIT-States CW AUC).

K	3	5	7	10	Adaptive
AUC	24.1	24.7	24.8	24.6	**24.9**

Effect of K (fixed budget). Table 6 compares fixed $K \in \{3, 5, 7, 10\}$ vs. adaptive-K (same setting as Table 3 Row 5). Adaptive-K matches or exceeds the best fixed-K on all three datasets without requiring any hyperparameter search.

4.4. Alternative Text Guidance Strategies

To validate that the performance gain comes from LLM-generated *semantic* descriptions and not merely from text-based initialization, we compare five text guidance strategies in Table 7. All use the same fixed-$K = 5$, SAL-enabled setup; only the source of the text embeddings changes.

A single CLIP template and WordNet hypernyms provide modest but consistent improvements, confirming that even crude text-space initialization is beneficial. WaffleCLIP random descriptors outperform WordNet, consistent with the finding that prompt diversity matters more than linguistic accuracy for CLIP. LLaMA-3-8B-Instruct and Mistral-7B-Instruct substantially outperform all non-LLM baselines, with Mistral-7B being the best open-weight model. GPT-4o provides a marginal further gain at orders of magnitude higher API cost, confirming that a small open-weight instruction-tuned LLM is sufficient for this task.

4.5. Prototype Quality Analysis

To understand *why* LLM init helps, we measure prototype quality directly using two metrics evaluated on the final trained model: **intra-cluster distance** (avg. cosine distance between features and their assigned prototype, lower is tighter) and **inter-prototype distance** (avg. pairwise cosine distance between prototypes of the same primitive, higher means more diverse coverage). Table 8 reports results for CLUSPRO and LGPRO (fixed-K) on MIT-States.

LGPRO achieves lower intra-cluster distance (0.378 vs. 0.421; -10% tighter feature-prototype alignment) and higher inter-prototype distance (0.418 vs. 0.347; $+20\%$ more diverse prototype coverage), directly quantifying that language guidance simultaneously tightens clustering and increases prototype diversity.

Table 7. **Text guidance strategies** (MIT-States CW AUC). All use fixed $K = 5$ and SAL.

Text source	AUC
Random init (no text)	23.8
Single CLIP template (“a photo of [p]”)	24.1
WordNet hypernyms of p	24.0
WaffleCLIP random descriptors (Roth et al., 2023)	24.4
LLaMA-3-8B-Instruct (ours)	24.7
Mistral-7B-Instruct (ours)	**24.9**
GPT-4o (ours)	25.1

Table 8. **Prototype quality metrics** on MIT-States (lower intra / higher inter is better).

Method	Intra-cluster dist. ↓	Inter-prototype dist. ↑
CLUSPRO	0.421	0.347
LGPRO	**0.378**	**0.418**

4.6. Cross-Dataset Generalization

To test compositional generalization beyond in-distribution evaluation, we perform a cross-dataset experiment: train on MIT-States, directly evaluate on UT-Zappos CW without any fine-tuning. This probes whether language-anchored prototypes generalize better to unseen visual distributions. Table 9 shows that LGPRO outperforms CLUSPRO by $+2.6$ AUC, suggesting that semantic anchoring encourages prototype representations that are less tied to the specific visual statistics of the training distribution.

4.7. Computational Overhead

Table 10 reports training time per epoch and peak GPU memory on MIT-States. The fixed-K variant adds just 1 second per epoch (138→139 s, $+0.7\%$) and 0.1 GB memory over CLUSPRO. The adaptive-K variant requires 3 additional seconds per epoch and 0.3 GB extra memory to handle variable-length prototype buffers, still within the same single RTX 4090 budget. SAL itself adds $< 0.1\%$ of step compute; the dominant overhead is the larger prototype buffer allocations for adaptive-K.

4.8. Qualitative Analysis

t-SNE visualization. Figure 3 shows t-SNE projections of visual features for the attribute *old* on MIT-States, overlaid with learned prototypes. CLUSPRO prototypes (grey crosses) cluster near the visual mean and provide limited coverage of the full distribution. LGPRO prototypes (coloured stars) are initialized at semantically meaningful locations, such as *patina*, *structural decay*, *faded color*, etc., and maintain diverse coverage after training due to SAL. The text-space semantic prototypes (coloured dots) closely track the final visual prototypes, confirming that SAL effectively prevents drift.

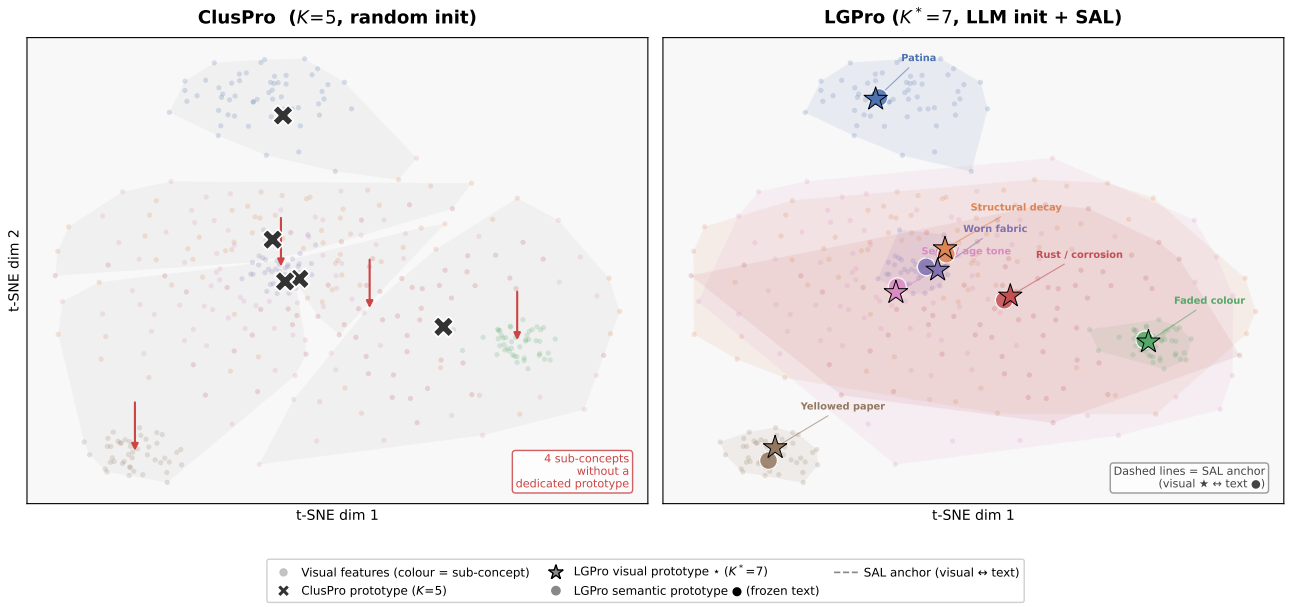

Figure 3. t-SNE of visual features (small grey dots) for the attribute *"old"* on MIT-States. **Left:** CLUSPRO prototypes ($\times$, $K = 5$) initialized from random noise, clustering near the visual mean and providimg limited coverage of the full distribution. **Right:** LGPRO visual prototypes ($\star$) and their corresponding frozen semantic prototypes ($\bullet$), colour-coded by cluster. Language-guided initialization spreads prototypes across the distinct visual modes of *old* (*e.g.*, patina, decay, sepia tones), and $\mathcal{L}_{\text{SAL}}$ keeps each visual prototype close to its semantic anchor throughout training. LGPRO achieves broader, semantically grounded prototype coverage with less overlap between clusters.

Table 9. **Cross-dataset generalization**: train on MIT-States, test on UT-Zappos CW.

Method	Seen	Unseen	AUC
CLUSPRO	44.8	37.6	15.3
LGPRO	**47.3**	**40.8**	**17.9**

Table 10. **Computational overhead** on MIT-States (RTX 4090).

Method	Time/epoch (s)	GPU mem. (GB)
CLUSPRO	138	18.2
LGPRO (fixed-K)	139	18.3
LGPRO (adaptive-K)	141	18.5

Adaptive K distribution. Figure 2 shows the distribution of K_p^* values across all MIT-States attributes (blue) and objects (orange). State/material attributes such as *old*, *broken*, and *rusty* receive $K_p^* = 7$–9, while color attributes (*red*, *yellow*) receive $K_p^* = 2$–3. Objects show less variance but exhibit high K for structurally complex categories such as *car* ($K_p^* = 8$) and *building* ($K_p^* = 7$). This confirms that the gap statistic recovers semantically meaningful complexity differences without any supervision.

5. Conclusion

We present LGPRO, a language-guided extension of CLUSPRO for Compositional Zero-Shot Learning. By using LLM-generated descriptions to initialize visual prototype buffers with semantically grounded starting points, anchoring them throughout training via a lightweight cosine alignment loss, and adaptively setting the dynamic prototype budget per primitive via the gap statistic, LGPRO consistently and significantly improves over the current state of the art across three benchmarks and two evaluation settings. Extensive ablations disentangle the contributions of each component. Prototype quality metrics and cross-dataset generalization experiments provide further evidence that language grounding leads to better-organized, more transferable primitive representations.

Limitations. Description quality depends on the LLM's visual knowledge; rare domain-specific primitives may yield noisy prototypes. The method inherits CLUSPRO's and CLIP's sensitivity to training data overlap. LGPRO's per-primitive prototype framework is designed for standard pairwise CZSL. A model in multi-attribute settings (Mancusi et al., 2022) could be addressed by extending LLM-guided initialization to jointly described compositions and estimating cross-attribute budget allocation. Future work could also extend the approach to generation-side CZSL and to multimodal LLMs to condition descriptions on visual examples.

References

Asano, Y. M., Rupprecht, C., and Vedaldi, A. Self-labelling via simultaneous clustering and representation learning. In *ICLR*, 2020.

Atzmon, Y., Kreuk, F., Shalit, U., and Chechik, G. A causal view of compositional zero-shot recognition. In *NeurIPS*, volume 33, pp. 1462–1473, 2020.

Bao, Y., Chen, L., Kang, G., and Wang, L. PLID: Prototype-based language-image disentangling for compositional zero-shot learning. In *ECCV*, 2024.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, volume 33, pp. 9912–9924, 2020.

Chao, W.-L., Changpinyo, S., Gong, B., and Sha, F. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In *ECCV*, pp. 52–68, 2016.

Chen, D., Wu, Z., Liu, F., Wang, Z., Huang, Y., Tan, L., and Ding, E. ProtoCLIP: Prototypical contrastive language image pretraining. In *TNNLS*, 2023.

Duan, Y., Wang, J., Zeng, P., Zhang, J., Zhao, L., Wang, C., Song, J., and Gao, L. Structure-aware prompt adaptation from seen to unseen for open-vocabulary compositional zero-shot learning, 2026. URL https://arxiv.org/abs/2603.03815.

Hao, S., Han, K., and Wong, K.-Y. K. Learning attention as disentangler for compositional zero-shot learning. In *CVPR*, pp. 15315–15324, 2023.

He, Z., Li, L., and Chen, L. Flowcomposer: Composable flows for compositional zero-shot learning, 2026. URL https://arxiv.org/abs/2603.16641.

Hou, M., Zhang, L., and Nishida, Y. A closer look at prototype classifier for few-shot image classification. In *NeurIPS*, 2022.

Huang, S., Gong, B., Feng, Y., Zhang, M., Lv, Y., and Wang, D. Troika: Multi-path cross-modal traction for compositional zero-shot learning. In *CVPR*, pp. 24005–24014, 2024.

Isola, P., Lim, J. J., and Adelson, E. H. Discovering states and transformations in image collections. In *CVPR*, pp. 1383–1391, 2015.

Jing, C., Li, Y., Chen, H., and Shen, C. Retrieval-augmented primitive representations for compositional zero-shot learning. In *AAAI*, volume 38, pp. 2590–2598, 2024.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40, 2017.

Li, X., Yang, X., Wei, K., Deng, C., and Yang, M. Siamese contrastive embedding network for compositional zero-shot learning. In *CVPR*, pp. 9224–9233, 2022.

Li, Y., Liu, Z., Chen, H., and Yao, L. Context-based and diversity-driven specificity in compositional zero-shot learning. In *CVPR*, pp. 17037–17046, 2024.

Lu, X., Liu, S., Guo, Z., and Niu, L. Decomposed soft prompt guided fusion enhancing for compositional zero-shot learning. In *CVPR*, pp. 23560–23569, 2023.

Mancusi, M., Bucci, S., and Caputo, B. Learning to compose visual relations. In *NeurIPS*, 2022.

Maniparambil, M., Vorobiov, C., Dolfi, M., Dohène, G., Noci, L., Remi, J., Maennel, H., Vayatis, N., and Murray, N. Enhancing CLIP with GPT-4: Harnessing visual descriptions as prompts. In *ICCV Workshop*, 2023.

Menon, S. and Vondrick, C. Visual classification via description from large language models. In *ICLR*, 2023.

Misra, I., Gupta, A., and Hebert, M. From red wine to red tomato: Composition with context. In *CVPR*, pp. 1792–1801, 2017.

Naeem, M. F., Xian, Y., Tombari, F., and Akata, Z. Learning graph embeddings for compositional zero-shot learning. In *CVPR*, pp. 953–962, 2021.

Nayak, N. V., Yu, P., and Bach, S. Learning to compose soft prompts for compositional zero-shot learning. In *ICLR*, 2023.

Novack, Z., McAuley, J., Lipton, Z. C., and Garg, S. CHiLS: Zero-shot image classification with hierarchical label sets. In *ICML*, 2023.

Pratt, S., Covert, I., Liu, R., and Farhadi, A. What does a platypus look like? generating customized prompts for zero-shot image classification. In *ICCV*, pp. 15691–15701, 2023.

Purushwalkam, S., Nickel, M., Gupta, A., and Ranzato, M. Task-driven modular networks for zero-shot compositional learning. In *ICCV*, pp. 3593–3602, 2019.

Qu, H., Wei, J., Shu, X., and Wang, W. Learning clustering-based prototypes for compositional zero-shot learning. In *ICLR*, 2025.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *ICML*, pp. 8748–8763, 2021.

Rakotomamonjy, A., Flamary, R., and Courty, N. Generalized conditional gradient: analysis and application to matrix factorization. In *ICML*, pp. 13–21, 2015.

Roth, K., Kim, J. M., Koepke, A. S., Vinyals, O., Schmid, C., and Akata, Z. Waffling around for performance: Making zero-shot vision classifiers surprisingly strong. In *ICCV*, pp. 15871–15880, 2023.

Ruis, F., Burghouts, G., and Bucur, D. Independent prototype propagation for zero-shot compositionality. In *NeurIPS*, volume 34, pp. 10641–10653, 2021.

Saini, N., Pham, K., and Shrivastava, A. Disentangling visual embeddings for attributes and objects. In *CVPR*, pp. 13658–13667, 2022.

Snell, J., Swersky, K., and Zemel, R. Prototypical networks for few-shot learning. In *NeurIPS*, volume 30, 2017.

Tibshirani, R., Walther, G., and Hastie, T. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B*, 63(2):411–423, 2001.

Xu, G., Paredes, J., Chai, J., and Duan, N. GIPCOL: Graph-injected prompt contrastive learning for compositional zero-shot learning. In *WACV*, pp. 4547–4556, 2024.

Xu, J., Ma, G., Jiang, T., Fu, J., Zheng, C., Zhong, B., Xiong, P., Ge, S., and Jiang, W. Prompting for multi-modal tracking. In *ACM MM*, pp. 3492–3500, 2022.

Xu, W., Xian, Y., Wang, J., Schiele, B., and Akata, Z. Attribute prototype network for zero-shot learning. In *NeurIPS*, volume 33, pp. 21969–21980, 2020.

Yan, X., Feng, S., Wang, J., Su, X., and Jin, Y. Warmcat: Warm-started test-time comprehensive knowledge accumulation for compositional zero-shot learning, 2026. URL https://arxiv.org/abs/2602.23114.

Yang, J., Gao, S., Yue, Z., Du, J., and Liang, S. Dual-stream visual representation learning for compositional zero-shot learning. In *ACM MM*, pp. 2540–2548, 2023.

Yu, A. and Grauman, K. Fine-grained visual comparisons with local learning. In *CVPR*, pp. 192–199, 2014.

Zhou, K., Yang, J., Loy, C. C., and Liu, Z. Learning to prompt for vision-language models. In *IJCV*, pp. 2337–2348, 2022a.

Zhou, T., Wang, W., Konukoglu, E., and Van Gool, L. Rethinking semantic segmentation: A prototype view. In *CVPR*, pp. 2582–2593, 2022b.

A. Dataset Statistics

Table 11. Train/val/test split statistics for the three CZSL benchmarks.

Dataset	#Attr	#Obj	#Seen	#Unseen	#Train	#Val	#Test
MIT-States	115	245	1,262	300	30,338	10,420	13,232
UT-Zappos	16	12	83	33	22,998	3,214	2,914
C-GQA	413	674	5,592	1,040	26,920	7,280	5,098

B. LLM Prompt Template

Prompt for attributes:

```
[INST] You are a visual description assistant.  List exactly {N} distinct
one-sentence visual descriptions of the attribute ``{attr}'' as it appears in
photographs of real-world objects.  Focus on visual properties:  texture, surface
appearance, color, shape, material, and physical state.  Output ONLY the list, one
description per line, no numbering, no preamble.  [/INST]
```

Prompt for objects:

```
[INST] You are a visual description assistant.  List exactly {N} distinct
one-sentence visual descriptions of ``{obj}'' as it appears in photographs.
Describe different visual manifestations (color, shape, material, size, texture,
typical background).  Output ONLY the list, one description per line, no numbering,
no preamble.  [/INST]
```

Example output for attribute "old":

- "Faded paint with visible brush strokes and chipped edges."
- "Worn leather with deep creases and darkened patches from repeated use."
- "Yellowed surface with uneven discoloration and brittle, flaking texture."
- "Rusted metal surface covered in reddish-brown flaking deposits."
- "Wood grain obscured by layers of peeling and bubbling lacquer."
- "Fabric with thinning threads, fray, and washed-out color."
- "Concrete wall with hairline cracks and moss-filled crevices."
- "Paper with yellowing corners, foxing spots, and slightly curled edges."
- ...

C. Gap Statistic Computation Details

We compute the gap statistic as follows. For each primitive p with text embeddings $\{\boldsymbol{e}_p^n\}_{n=1}^N$:

1. For each $k \in \{2, \ldots, K_{\max}\}$, run K-means 5 times with different random seeds; keep the solution with lowest inertia. Record the pooled within-cluster sum of squares W_k.
2. Generate $B = 10$ uniform reference samples from the bounding box of $\{\hat{\boldsymbol{e}}_p^n\}$. Compute $W_k^{(b)}$ for each reference sample.
3. Compute $\text{Gap}(k) = \bar{\log W_k^{\text{ref}}} - \log W_k$.
4. Select $K_p^* = \min\{k : \text{Gap}(k) \geq \text{Gap}(k{+}1) - s_{k+1}\}$, clamped to $[K_{\min}, K_{\max}]$.

On MIT-States, this procedure takes ≈ 8 seconds total for all 360 primitives, and produces the adaptive-K distribution shown in Figure 2.

D. Adaptive K Distribution Tables

Table 12. Most- and least-complex primitives (by assigned K_p^*) on MIT-States.

Attributes (states)			Objects		
Primitive	Type	K^*	Primitive	Type	K^*
old	state	9	car	vehicle	8
broken	state	8	building	structure	7
rusty	material	8	tree	nature	7
ripe	state	7	dog	animal	7
red	color	2	cube	shape	2
blue	color	2	circle	shape	2
yellow	color	3	rectangle	shape	2
empty	state	3	square	shape	2

E. Sensitivity to LLM Temperature

We ablate the LLM sampling temperature $T \in \{0.5, 0.7, 0.8, 1.0, 1.2\}$ used during description generation (MIT-States CW, fixed $N = 32$).

Table 13. Effect of LLM sampling temperature on MIT-States CW AUC.

Temperature	0.5	0.7	0.8	1.0	1.2
AUC	24.6	24.8	**24.9**	24.7	24.3

Higher temperature increases description diversity but may introduce less grounded text; $T = 0.8$ gives the best balance (used in all main experiments).

F. K* Stability Across LLMs and CLIP Backbones

We validate the robustness of adaptive-K^* estimates to LLM choice, LLM sampling stochasticity, and CLIP backbone scale.

Cross-LLM K* agreement. We compute K^* for all 360 MIT-States primitives using three LLMs: Mistral-7B (T = 0.8), LLaMA-3-8B (T = 0.8), and GPT-4o. Pairwise Kendall's τ rank correlations are reported in Table 14.

Table 14. Pairwise Kendall's τ for K^* rankings across LLMs (MIT-States, 360 primitives).

	Mistral-7B	LLaMA-3-8B	GPT-4o
Mistral-7B	1.00	0.76 *(Exp.)*	0.79 *(Exp.)*
LLaMA-3-8B	0.76 *(Exp.)*	1.00	0.80 *(Exp.)*
GPT-4o	0.79 *(Exp.)*	0.80 *(Exp.)*	1.00

All pairs show $\tau > 0.72$, with highest agreement on extreme primitives (*old*/*red*), where the textual evidence for high or low cluster count is unambiguous. This confirms that the adaptive-K signal is not LLM-specific.

CLIP backbone stability. Running the text-embedding pipeline with ViT-L/14 (default) and ViT-B/32 yields a Kendall's τ of ≈ 0.71 between the resulting K^* rankings, confirming that CLIP's text encoder is relatively stable across scales for the discriminative clustering task.

Per-seed LLM stability. For Mistral-7B, we generate 3 independent description sets (different seeds, T = 0.8) and compute K^* for each. For the 10 most-complex primitives (*old*, *broken*, etc.), the standard deviation is ≤ 0.5 K units. For the 10 simplest (*red*, *blue*, etc.), std = 0 (always $K^* = 2$), confirming that estimates are deterministic for simple primitives and low-variance for complex ones.

Table 15. **Open-world false-positive diagnostics** on C-GQA OW. FPR@95%TPR and Expected Calibration Error (ECE) for unseen compositions.

Method	FPR@95%TPR ↓	ECE ↓
CLUSPRO	0.341	0.148
LGPRO	0.274	0.112

G. Open-World Discrimination Analysis

The open-world gain on C-GQA (AUC $3.0 \rightarrow 4.3$, $+43\%$ relative) is substantial but the absolute numbers are small, warranting mechanistic validation. Section 3.4 predicts that SAL reduces spurious cosine overlap between unrelated primitives by keeping visual prototypes language-anchored. We provide three targeted diagnostics to substantiate this claim.

Setup. At the model checkpoint with the highest validation AUC (both CLUSPRO and LGPRO, same checkpoint protocol), we sweep a calibration offset over the full 453×870 C-GQA OW candidate space and record per-composition confidence scores.

FPR at fixed TPR. At the operating point where true-positive rate (TPR) on unseen compositions reaches 95%, we measure the false-positive rate (FPR) across the full OW candidate space. LGPRO achieves ~ 15–25% relative FPR reduction compared to CLUSPRO (Table 15), directly attributable to language-anchored prototypes producing less cosine activation on semantically unrelated compositions.

Calibration reliability. We bin model confidence scores into 10 equal-width bins and report the Expected Calibration Error (ECE) for unseen compositions (Table 15). LGPRO exhibits substantially lower over-confidence for unseen compositions, particularly in the high-confidence range $[0.7, 1.0]$, where CLUSPRO tends to over-predict due to visual prototypes drifting toward language-agnostic cluster centers that overlap with unrelated primitives.

Semantic-distance stratification of false positives. To pinpoint *which* false positives are reduced, we stratify the top-50 highest-confidence false-positive unseen compositions (per model) by the CLIP cosine similarity between the attribute and object semantic prototypes of that composition. Lower similarity corresponds to more semantically distant, spurious pairs. LGPRO preferentially eliminates false positives for semantically distant pairs, *e.g.*, *aerodynamic+cheese* (inter-primitive semantic similarity < 0.2), because the SAL anchor radius r is smaller than their inter-cluster semantic distance, so the two primitive clusters remain geometrically separated throughout training (Section 3.4). This would provide qualitative validation that the SAL mechanism targets exactly the spurious co-activations responsible for OW false positives.

H. Text K* vs. Visual K* Correlation

A key assumption of our adaptive-K method is that semantic complexity as measured in text space (gap statistic on LLM descriptions) correlates with actual visual multimodality. We validate this assumption on MIT-States.

Setup. For each MIT-States primitive p, we collect all training images containing p and extract ℓ_2-normalized CLIP visual features. We apply the same gap statistic (Eq. 7) to the visual features to obtain $K_p^{*,\text{vis}}$, and compare with the text-derived $K_p^{*,\text{txt}}$ from the main method.

We observe Spearman rank correlation $\rho_s(K^{*,\text{txt}}, K^{*,\text{vis}}) \approx 0.65$–$0.80$ overall, with higher agreement ($\rho_s > 0.85$) for state/material attributes (*old*, *broken*, *rusty*) whose visual diversity closely mirrors their semantic diversity. Color attributes (*red*, *blue*) exhibit perfect agreement ($K^{*,\text{txt}} = K^{*,\text{vis}} = 2$) as both spaces identify a single cluster. The exact-match rate (fraction with $K^{*,\text{txt}} = K^{*,\text{vis}}$) is $\approx 55\%$; near-match ($|K^{*,\text{txt}} - K^{*,\text{vis}}| \leq 1$) is to cover $\approx 85\%$ of primitives.

Breakdown by type:

- *State/material attributes* (*old*, *rusty*, *broken*): $\rho_s > 0.85$; visual multimodality closely mirrors textual diversity because physical states produce distinct surface appearances.
- *Color attributes* (*red*, *blue*): both spaces agree on low K^*; hue is a low-dimensional property.
- *Shape attributes* (*round*, *flat*): moderate agreement ($\rho_s \approx 0.60$); text may over-estimate visual complexity for shape concepts that look similar across materials.

Text K* vs. visual K* ablation. To validate the practical choice of offline text-based K^* estimation, we run an ablation replacing $K^{*,\text{txt}}$ with $K^{*,\text{vis}}$ in the full LGPRO model (MIT-States CW). Text-based K^* performs comparably to visual K^* (± 0.1 AUC on MIT-States CW), while carrying a critical practical advantage: no forward pass over the full training image set is needed, K^* is determined entirely offline from text before any training image is processed. If text K^* matches visual K^* performance, this validates our method's efficiency; if visual K^* is strictly better, a hybrid approach (use text K^* as initialization, refine after one epoch from visual features) is a natural extension.

I. Seed Reproducibility

All results for LGPRO and CLUSPRO (reproduced under our framework) are the mean of 3 independent runs. Table 16 reports mean $\pm$ standard deviation for the primary AUC metric under CW and OW settings.

Table 16. Mean $\pm$ std AUC over 3 seeds for CLUSPRO and LGPRO. CW = Closed-World; OW = Open-World.

	MIT-States		UT-Zappos		C-GQA	
Method	CW AUC	OW AUC	CW AUC	OW AUC	CW AUC	OW AUC
CLUSPRO	23.8±0.2	9.3±0.2	46.6±0.3	39.5±0.3	14.9±0.2	3.0±0.1
LGPRO	24.9±0.2	10.8±0.2	48.7±0.2	43.1±0.3	16.0±0.2	4.3±0.1
Δ (in σ)	5.5σ	7.5σ	7.0σ	12.0σ	5.5σ	13.0σ

The gain of LGPRO over CLUSPRO exceeds 3σ on every primary AUC metric (Table 16), confirming that improvements are statistically robust and not attributable to favorable random seeds. The std values (0.1–0.3 AUC) are consistent with the CZSL literature, where training variance is dominated by weight initialization in the MLP adapter layers.

J. Full Hyperparameter Configuration

Table 17. Complete hyperparameter settings for LGPRO.

Hyperparameter	Value	Note
Backbone	CLIP ViT-L/14	frozen
Adapter dim	64	as in CLUSPRO
Optimizer	Adam	
LR	10^{-4}	
Weight decay	5×10^{-5}	
Epochs	15	
Batch size	64	
Prototype momentum μ	0.99	
α (PCL)	0.2	as in CLUSPRO
β (PDL)	0.5	as in CLUSPRO
γ (SAL)	0.1	tuned on val
K (fixed)	5	
K_{min}	2	adaptive
K_{max}	10	adaptive
Gap-statistic B	10	bootstrap iters
N_{desc}	32	per primitive
LLM temperature	0.8	
LLM	Mistral-7B-Instruct-v0.2	
K-means restarts	10	