Beyond Scalar Distances: Semantic Attribute Gradients from Frozen MLLMs for Visual Embeddings

Shubhang Bhatnagar* **Dheeraj Baiju*** **Narendra Ahuja**
University of Illinois Urbana-Champaign
sb56@illinois.edu dheerajbaiju501@gmail.com n-ahuja@illinois.edu

Abstract

Vision encoders for retrieval are typically trained with class-label supervision: each training pair reduces to a scalar that uniformly pushes the embedding apart or pulls it together, as if every visual attribute either differed or matched. A multimodal large language model (MLLM), shown the same pair, can articulate those attributes and use them to predict whether the images share a class. We propose **SAGA**, a framework that turns this language-grounded, attribute-aware perception into a training signal for the encoder itself. Specifically, we use Group Relative Policy Optimization (GRPO) to reward the MLLM for correct predictions on the vision encoder's tokens. Since correct predictions require those tokens to expose the specific attributes that differ or match between the pair, the gradient pushes the encoder to encode them, replacing the uniform pair-level scalar with attribute-resolved supervision. An auxiliary attention-distillation loss anchors the encoder's embedding to tokens the MLLM attended to, and a standard metric-learning loss shapes the embedding geometry for nearest-neighbour retrieval. The MLLM is frozen throughout and discarded at inference, matching the deployment cost of a metric-learning baseline. SAGA improves Recall@1 by 3 to 6 points over state-of-the-art baselines on CUB-200-2011, Cars-196, FGVC-Aircraft, and iNaturalist Aves on zero-shot image retrieval.

1 Introduction

A visual encoder must embed images along the dimensions that distinguish them: the shape of a bill, the pattern of a wing, the silhouette of a garment, the geometry of a tail. The dominant paradigm (metric learning) trains them with class labels alone [5, 33, 20, 3], a binary signal that acts on every attribute in unison, pulling all of them together when classes match and pushing all of them apart when classes differ, even when two images share most attributes and are distinguished by only a few. This is the wrong inductive bias for zero-shot image retrieval [28, 13], where test classes come from a disjoint label set and are separated by attribute combinations the training signal never required the encoder to represent. Figure 1 makes this concrete: an Indigo Bunting and a Blue Grosbeak (from the CUB200 [30] dataset) share a deep-blue plumage and gray legs, differing only in their wing bars. A class-label scalar reduces this pair to a uniform "different," carrying no information that those few attributes are the ones that matter while the rest agree.

Multimodal large language models (MLLMs) trained on image-text data [16, 1, 4, 8] acquire exactly this perception of visual structure. Asked about an image, an MLLM articulates fine-grained attributes (shapes, patterns, textures, structural proportions) and localizes them on the image while it reasons. For the pair in Figure 1, we can see that the MLLM identifies *wing pattern: two orange wing bars* for

*Equal contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

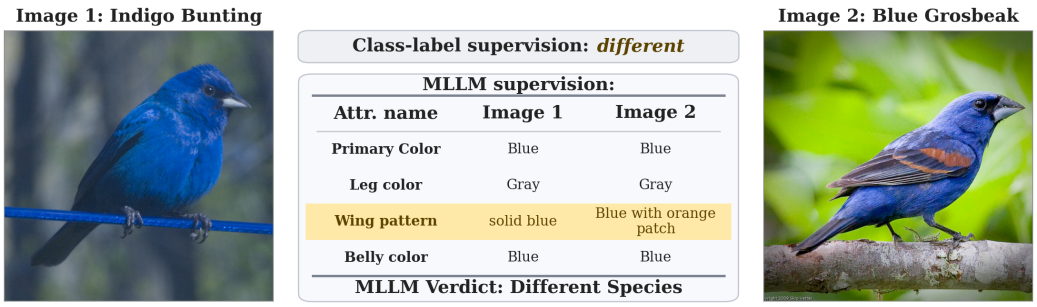

Figure 1: **Using only class labels for images reduces supervision to a scalar, whereas an MLLM resolves it into attributes.** A class-label loss collapses the difference between two very similar-looking bird species into a single 'different' scalar, pushing every embedding dimension apart, even those potentially encoding shared attributes like blue plumage and leg color. A frozen MLLM, by contrast, can identify which attributes match and include them in reaching the same-/different-species verdict. Our method, SAGA, harnesses this by rewarding correct verdicts and reinforcing precisely those feature components (directions) that the MLLM's discrimination relies on, while leaving shared-attribute directions untouched.

Bird 2 against *wing pattern: solid blue* for Bird 1, and concludes 'different species'. We ask whether the MLLM's sensitivity to detail can serve as a training-time supervisor for a visual encoder, turning its high emphasis on certain attributes into learning gradients that reshape the encoder's embedding space.

In this work, we answer this affirmatively and propose **SAGA** (**S**emantic **A**ttribute **G**radients from **A**djudication), a framework that turns a frozen MLLM into a training-time supervisor for the visual encoder of a retrieval system. Our visual encoder is the vision tower of a multimodal LLM, which emits a sequence of patch tokens fed to an MLLM which is asked to compare image pairs by describing their attributes [34]. Correct same/different class verdicts are rewarded via Group Relative Policy Optimization (GRPO) [27]; the resulting gradient flows back through the frozen language backbone into the encoder, pushing it to represent the discriminative attributes the MLLM had to perceive correctly to reach those verdicts. In Figure 1, the correct 'different species' verdict of the MLLM relies on it identifying Bird 2's orange wing bars, so the policy gradient reinforces the encoder to identify and discriminate this attribute, while directions encoding the shared blue plumage and gray legs receive no such reinforcement.

Encoded discriminative attributes still have to be aggregated into a single retrieval vector, so we attach a small pooler that collapses the patch tokens into the embedding used for nearest-neighbor search at inference. The same forward pass also reveals which image regions/tokens the MLLM attended to while reasoning about each image's attributes. We distill [11, 37] this attention into the pooler, a lightweight module that aggregates the encoder's output tokens for an image into a single embedding vector used for nearest-neighbor retrieval at inference. Without this signal the pooler would be left to discover attribute-relevant tokens from class labels alone, the same statistical-discovery problem identified above for the encoder.

The MLLM is frozen throughout training and is used only at training time to produce GRPO rewards and attention targets. Once training is complete only the vision encoder and pooler are retained for deployment. Retrieval is performed by only these two components, matching the deployment cost of any standard metric learning pipeline. The supervisory signal requires only the pairwise class labels already used by the metric learning objective; no attribute annotations are needed.

Our main contributions are:

- We present SAGA, a framework that uses a frozen multimodal LLM as a training-time supervisor to learn a visual encoder using attribute-aware supervision gradients that go beyond the scalar pairwise similarity between class labels.
- We train the encoder with a reinforcement learning objective that rewards the encoder when the MLLM correctly judges image pairs as being from the same class or different classes, and distill the MLLM's attention into the pooler. This (1) teaches the encoder to incorporate

in its representation the attributes the MLLM uses to judge, and (2) teaches the pooler to weight the image regions the MLLM attends to when forming its verdict.

- We evaluate SAGA on four zero-shot image retrieval benchmarks (CUB-200-2011, Cars-196, FGVC-Aircraft, iNaturalist Aves), where it improves Recall@1 by 3–6% over state-of-the-art baselines on the same vision backbone.

2 Related Work

Deep metric learning. Deep metric learning (DML) is the dominant framework for training a vision encoder whose output space is itself a semantic geometry: standard distance metrics over embeddings recover class-level similarity, and the resulting features are intended to generalize to disjoint test classes for downstream tasks such as zero-shot image retrieval [28, 13, 2], face verification [26], and multi-object discovery/recognition [23, 24]. Methods in this family supervise the encoder with a scalar pairwise objective derived from class labels, instantiated either through tuple-based losses operating directly on samples (contrastive [5, 10], triplet [26], multi-similarity [33]) or through proxy-based losses that replace tuple mining with learnable class representatives (Proxy-NCA [20], Proxy-Anchor [13], HIER[14], DDML [21], Potential Field [3]). Across both families, supervision per pair reduces to a single scalar that acts on every attribute dimension in unison, telling the encoder that two images should move closer or farther but not which visual attributes carry the class signal.

Multimodal large language models. MLLMs trained on image-text data, e.g., LLaVA [16], Qwen-VL [1], and InternVL [4], articulate fine-grained visual attributes through language and localize them on the image while reasoning. Group Relative Policy Optimization (GRPO) [27] has become the standard recipe for aligning these models with non-differentiable rewards, including grounded visual reasoning [7, 31] and reconstructive encoder objectives [35]. These works fine-tune the MLLM for VQA or grounded reasoning as a whole. SAGA uses GRPO in the opposite role, treating the frozen MLLM as a loss function whose policy gradient supervises a retrieval encoder. A complementary line observes that an MLLM's internal attention often localizes salient regions even when its textual output is flawed [12], and exploits this at *inference* time by reallocating resolution or tokens toward attended regions [6, 39]; SAGA distills that same attention at *training* time into a retrieval pooler.

Language-guided visual representation learning. A complementary thread uses textual descriptions as supervision for visual representations. CLIP [22] and SigLIP [38] align image and text embeddings via contrastive pretraining, and subsequent work adapts these models with LLM-generated class descriptions for zero-shot recognition [19, 25]; Similar to DML methods, these consume language as fixed targets that align embeddings to text while also requiring caption level labels for such fine-grained images. SAGA does not need such caption level supervision, and only using the class label supervision and GRPO can make encoder learn features about discriminative attributes.

3 Method

3.1 Setup and Notation

Deep metric learning (DML) learns a semantic distance over images from a labelled dataset $\mathcal{D} = \{(\mathbf{I}_i, y_i)\}_{i=1}^{|\mathcal{D}|}$ with $y_i \in \{1, \dots, N\}$, parameterizing an image-to-embedding map $g_{\theta,\phi} : \mathbf{I} \mapsto \mathbf{z} \in \mathbb{R}^{D_e}$ and taking $d(\mathbf{I}_1, \mathbf{I}_2) = \|\mathbf{z}_1 - \mathbf{z}_2\|_2$; d should be small for same-class pairs and large otherwise.

Vision encoder and retrieval pooler. We factor $g_{\theta,\phi} = c_\phi \circ f_\theta$ into a vision encoder (vision tower of Qwen3-VL [1]) and a retrieval pooler c_ϕ, producing a sequence of patch tokens $\mathbf{X} = f_\theta(\mathbf{I}) \in \mathbb{R}^{N_p \times D}$ for an image $\mathbf{I}$. The pooler c_ϕ aggregates these into a compact embedding $\mathbf{z} = c_\phi(\mathbf{X}) \in \mathbb{R}^{D_e}$, instantiated as mean, max, or attention-pooling. In the attention-pooling case, a single learnable query attends over the N_p patch tokens to produce spatial weights $\boldsymbol{\beta} \in \Delta^{N_p}$ (the probability simplex over the N_p patches), and the weighted sum of tokens is linearly projected to D_e and ℓ_2-normalized (Appendix B). Both θ and ϕ are trainable.

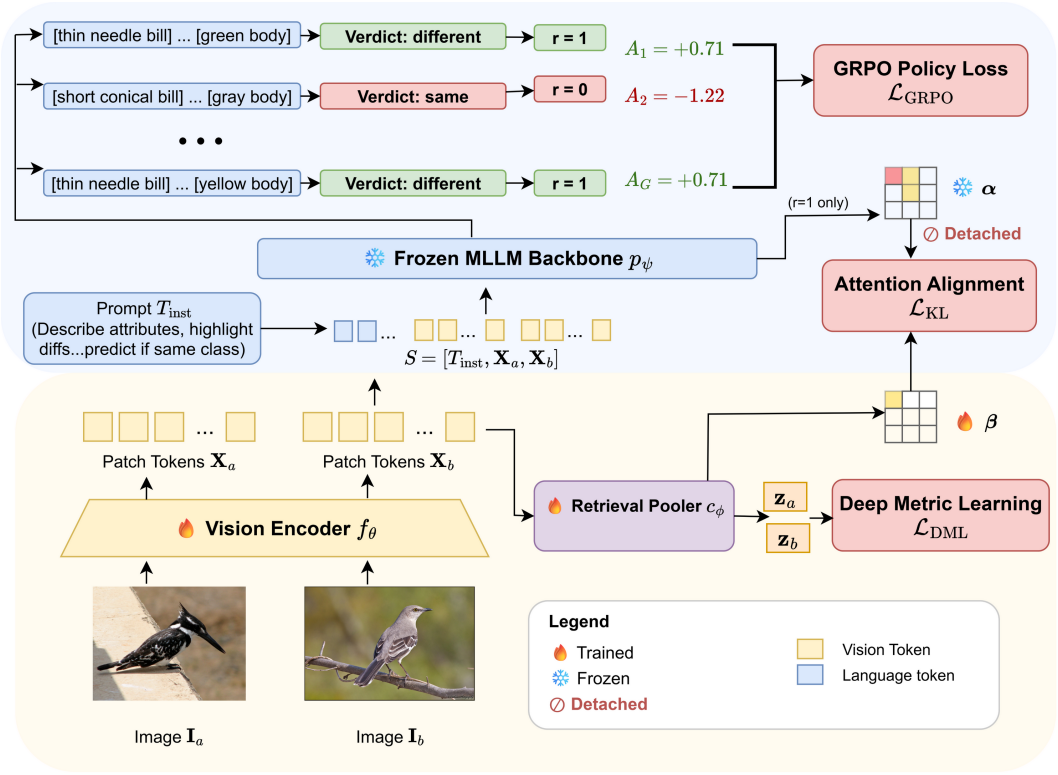

Figure 2: **Overview.** SAGA uses a frozen MLLM as an attribute-aware supervisor for deep metric learning. For an image pair $(\mathbf{I}_a, \mathbf{I}_b)$, the trainable vision encoder f_θ produces patch tokens $\mathbf{X}_a, \mathbf{X}_b$, which feed three losses with complementary roles. (1) The tokens and a comparison prompt T_{inst} are fed to the frozen MLLM p_ψ, which samples G responses ending in a same/different-class verdict; the GRPO loss $\mathcal{L}_{\text{GRPO}}$ rewards correct verdicts and back-propagates through p_ψ into f_θ, pushing it to encode the *discriminative* attributes the MLLM relied on when making correct predictions. (2) The MLLM's attention α from the same forward pass reveals which patch tokens the MLLM attended to while describing each image's attributes; on correct rollouts, the per-image mean attribute-attention $\bar{\alpha}$ is distilled into the pooler's attention β via $\mathcal{L}_{\text{KL}}$, encouraging c_ϕ to pool over those regions when forming embeddings z_a, z_b. (3) The pooler c_ϕ aggregates the tokens into embeddings $\mathbf{z}_a, \mathbf{z}_b$, which a deep metric learning loss $\mathcal{L}_{\text{DML}}$ shapes for nearest-neighbor search. The MLLM is frozen throughout and discarded at inference.

Frozen MLLM-guided supervision To enrich f_θ with semantic reasoning, we use it as the visual front-end for the language backbone p_ψ of the MLLM. Given $\mathbf{X}$ and a text prompt, p_ψ autoregressively generates output tokens; we use $\boldsymbol{\alpha}^{(t)} \in \Delta^{N_p}$ for its attention distribution at an intermediate decoder layer (justified empirically in Sec. 3.4) from a generated token t over the patch positions of $\mathbf{X}$. The parameters ψ are frozen, but gradients flow through p_ψ into θ. The MLLM is used only at training time and discarded at inference, so the embedding model has the same cost as a standard DML pipeline. For proxy-based DML losses, we additionally maintain M trainable proxies per class, $\mathbf{p}_{j,k} \in \mathbb{R}^{D_e}$ for $j \in \{1, \ldots, N\}, k \in \{1, \ldots, M\}$.

3.2 GRPO Attribute Reasoning Loss

The core contribution is using a frozen MLLM as a differentiable, attribute-aware loss function via Group Relative Policy Optimization (GRPO) [27]. Rather than supervised fine-tuning with fixed targets, the MLLM generates freely, and we reward only the final same/different-class verdict. This reward is computed against the ground-truth class labels y used by the DML loss, adding no extra annotation burden. Intermediate attribute descriptions serve as an implicit chain-of-thought, making the resulting gradient attributionally rich despite the binary reward.

Input construction. Given two images $\mathbf{I}_a, \mathbf{I}_b$ from $\mathcal{D}$ with labels y_a, y_b, we construct an input sequence by concatenating their patch tokens with a structured text prompt T_{inst}:

$$S = [\,\mathbf{X}_a,\ \ \mathbf{X}_b,\ \ T_{\text{inst}}\,]. \tag{1}$$

T_{inst} instructs the MLLM to: (1) describe visual attributes in JSON, (2) highlight key differences, and (3) predict if the images share a class.

Rollout and reward. For each pair, we sample G completions without gradients:

$$\hat{Y}^{(g)} = (y_1^{(g)}, \ldots, y_{T_g}^{(g)}) \sim p_\psi(\cdot \mid S), \quad g = 1, \ldots, G, \tag{2}$$

where T_g is the length of the g-th completion. We parse each completion for the verdict field and assign a binary reward:

$$r^{(g)} = \begin{cases} 1 & \text{if the parsed verdict matches } \mathbf{1}[y_a = y_b], \\ 0 & \text{otherwise (including unparseable outputs).} \end{cases} \tag{3}$$

Policy gradient update. We compute group-normalized advantages $A^{(g)} = (r^{(g)} - \bar{r})/(\sigma_r + \epsilon)$ for each completion. Pairs with $\sigma_r = 0$ contribute no policy-gradient signal and are skipped, with additional pairs drawn from subsequent micro-batches to maintain a target contributing-pair count per optimizer step (DAPO Dynamic Sampling [36]). Because each rollout is generated and consumed within a single gradient update, the GRPO importance ratio $\rho_t = \pi_\theta(y_t)/\pi_{\theta_{\text{old}}}(y_t) \equiv 1$ at update time, the surrogate's clip is vacuous, and we use no reference policy ($\beta = 0$), so the GRPO loss reduces to its first-order, advantage-weighted negative log-likelihood form:

$$\log \pi_\theta(y_t^{(g)} \mid y_{<t}^{(g)}, S) \quad \text{for each generated token } t. \tag{4}$$

The GRPO loss is the advantage-weighted negative log-likelihood over all generated tokens:

$$\mathcal{L}_{\text{GRPO}} = -\frac{1}{|\mathcal{T}|} \sum_g \sum_{t=1}^{T_g} A^{(g)} \log \pi_\theta(y_t^{(g)} \mid y_{<t}^{(g)}, S), \tag{5}$$

where $|\mathcal{T}| = \sum_g T_g$ is the total number of generated tokens across contributing completions. Token-level normalization [36] prevents short completions from dominating the gradient.

Why GRPO provides attribute-aware gradients. The policy gradient at each token $\pi(y_t)$ sends signals back to θ through $\partial \log \pi(y_t)/\partial \mathbf{X} \cdot \partial \mathbf{X}/\partial \theta$, exciting only the dimensions of $\mathbf{X}$ used to predict y_t. Shared-attribute tokens are produced with similar probabilities across correct and incorrect rollouts: the visual signal for these attributes is identical in $\mathbf{I}_a$ and $\mathbf{I}_b$ by definition, so the MLLM's belief about them is fixed by perception and does not track the verdict outcome. $\log \pi(y_t)$ is therefore roughly constant in g, and the advantage-weighted sum vanishes by the mean-zero property of group-normalised advantages. Discriminating-attribute tokens, in contrast, force the MLLM to commit to one description per rollout (e.g., "orange wing bars" vs. "solid blue") on the basis of whatever signal $\mathbf{X}$ exposes; if the encoder has not yet cleanly encoded that signal, sampled tokens vary across rollouts and align with the verdict outcome, with rollouts that picked the correct attribute receiving $r^{(g)} = 1$ and the others $r^{(g)} = 0$. The advantage-weighted sum is therefore non-zero on exactly these tokens, flowing into the $\mathbf{X}$-directions that resolve them. As those directions sharpen, the MLLM grows confident, rollout disagreement shrinks, and the gradient decays, producing an automatic curriculum onto attributes the encoder has not yet learned. This mirrors the outcome-only credit-assignment mechanism by which DeepSeek-R1 [9] elicits emergent reasoning from binary correctness rewards.

3.3 Deep Metric Learning Loss

$\mathcal{L}_{\text{GRPO}}$ shapes which visual signal f_θ encodes, but does not arrange the resulting embeddings $\mathbf{z} = c_\phi(f_\theta(\mathbf{I})) \in \mathbb{R}^{D_e}$ for nearest-neighbor search. We therefore retain a standard deep metric learning loss $\mathcal{L}_{\text{DML}}$ computed over the pooled embeddings of the full training batch, which back-propagates into both θ and ϕ to provide geometric supervision every step.

The two losses are deliberately complementary. $\mathcal{L}_{\text{DML}}$ decides *where* points sit in $\mathcal{Z}$, while $\mathcal{L}_{\text{GRPO}}$ decides *which visual signal* the encoder uses to place them there. Removing $\mathcal{L}_{\text{DML}}$ would leave

the GRPO gradient un-anchored to any explicit metric structure; removing $\mathcal{L}_{\text{GRPO}}$ would leave the geometric supervision attribute-blind, recovering the coarse pairwise signal that motivated this work. Our framework is agnostic to the specific DML objective, and we evaluate three representative variants (MS [33], Proxy-Anchor [13], and Potential Field [3]) to demonstrate that the GRPO supervision composes with both proxy-free and proxy-based metric learning.

3.4 Attention Alignment Loss

$\mathcal{L}_{\text{GRPO}}$ updates f_θ via the frozen LLM, but the pooler c_ϕ only perceives these gradients indirectly. To prevent c_ϕ from weighting non-discriminative regions and erasing attribute information made encodable in f_θ, we introduce an attention-alignment loss. This supervises the pooler's spatial focus by distilling, on correct rollouts only, the MLLM's internal attention during attribute generation.

While a final-layer teacher is intuitive, Qwen3-VL-8B's last-layer attention is dominated by "attention-sink" and register-token artifacts, yielding maps poorly aligned with visual attributes. Our layer selection procedure follows the quantitative grounding analyses used in AttWarp [6] and Zhang et al. [39]. Specifically, to select the layer for distillation to the pooler, we evaluate candidate layers by measuring how well the attention from the attribute tokens localizes onto the object within its annotated visual region, i.e., the attention mass falling in the object area. In our setting, we use CUB's object/part annotations together with its attribute labels, which is how we select $\ell = 26$. Zhang et al. [39] and Dalal et al. [6] use similar objectives to choose a layer that localizes their objects of interest well. This procedure can be applied to any MLLM given the dataset and annotations.

Concretely, let $\mathcal{A}_a, \mathcal{A}_b$ denote the attribute-description tokens describing $\mathbf{I}_a$ and $\mathbf{I}_b$, and $\boldsymbol{\alpha}_a^{(t)}, \boldsymbol{\alpha}_b^{(t)} \in \Delta^{N_p}$ denote the head-averaged attention at layer $\ell = 26$ from token t, renormalized over patches. We aggregate these into a *mean attribute-attention map* per image:

$$\bar{\boldsymbol{\alpha}}_a \;=\; \frac{1}{|\mathcal{A}_a|} \sum_{t \in \mathcal{A}_a} \boldsymbol{\alpha}_a^{(t)}, \qquad \bar{\boldsymbol{\alpha}}_b \;=\; \frac{1}{|\mathcal{A}_b|} \sum_{t \in \mathcal{A}_b} \boldsymbol{\alpha}_b^{(t)}, \tag{6}$$

which represents the union of patch regions the LLM attended to while describing attributes. For each pair with reward $r^{(g)} = 1$, we align these with the pooler's attentions $\boldsymbol{\beta}_a, \boldsymbol{\beta}_b$ via:

$$\mathcal{L}_{\text{KL}} \;=\; D_{\text{KL}}(\bar{\boldsymbol{\alpha}}_a \,\|\, \boldsymbol{\beta}_a) \;+\; D_{\text{KL}}(\bar{\boldsymbol{\alpha}}_b \,\|\, \boldsymbol{\beta}_b)\,. \tag{7}$$

Eq. 7 is gradient-equivalent (in $\boldsymbol{\beta}$) to the per-token average $\frac{1}{|\mathcal{A}_a|}\sum_{t\in\mathcal{A}_a} D_{\text{KL}}(\boldsymbol{\alpha}_a^{(t)} \| \boldsymbol{\beta}_a) + \frac{1}{|\mathcal{A}_b|}\sum_{t\in\mathcal{A}_b} D_{\text{KL}}(\boldsymbol{\alpha}_b^{(t)} \| \boldsymbol{\beta}_b)$, differing only by a $\boldsymbol{\beta}$-independent entropy offset. This formulation is cheaper to compute and targets the parts of the rollout that localize on specific visual regions. Gradients flow only into ϕ; tokens $\mathbf{X}$ and teacher maps $\boldsymbol{\alpha}$ are detached. This teaches c_ϕ *where to look*, complementing $\mathcal{L}_{\text{GRPO}}$'s role in determining *what to encode*.

3.5 Total Objective and Training

The overall loss is:

$$\mathcal{L}_{\text{Total}} = \lambda_{\text{dml}}\,\mathcal{L}_{\text{DML}} + \lambda_{\text{lm}}\,\mathcal{L}_{\text{GRPO}} + \lambda_{\text{kl}}\,\mathcal{L}_{\text{KL}}, \tag{8}$$

where λ_{dml}, λ_{lm}, and λ_{kl} are scalar loss weights. The three losses play complementary roles: $\mathcal{L}_{\text{DML}}$ optimizes the embedding geometry via θ, ϕ, and the proxies $\mathbf{p}_{j,k}$ (when present); $\mathcal{L}_{\text{GRPO}}$ provides attribute-aware gradients to θ via the frozen LLM; and $\mathcal{L}_{\text{KL}}$ teaches ϕ where to attend.

Per-step training flow. Each training step proceeds in three phases: **(A)** compute embeddings for the full batch and apply $\mathcal{L}_{\text{DML}}$; **(B)** sample image pairs from the batch, run G rollouts per pair through the frozen MLLM, and score binary rewards; **(C)** for pairs with non-zero advantage variance, run the differentiable forward pass and apply $\mathcal{L}_{\text{GRPO}}$, additionally applying $\mathcal{L}_{\text{KL}}$ for the rollouts within those pairs that received reward 1. We use gradient accumulation across pairs within a step, followed by gradient clipping and a single optimizer update. Algorithm 1 (Appendix B) provides full pseudocode.

3.6 Inference

At inference, the frozen MLLM p_ψ is discarded entirely. For a query image $\mathbf{I}_q$ and gallery $\mathcal{G} = \{\mathbf{I}_g\}$, retrieval is standard nearest-neighbor search in the embedding space:

$$\mathbf{z}_q = c_\phi(f_\theta(\mathbf{I}_q)), \quad \mathbf{z}_g = c_\phi(f_\theta(\mathbf{I}_g)), \quad \text{rank by } \|\mathbf{z}_q - \mathbf{z}_g\|_2. \tag{9}$$

Table 1: **Main results on zero-shot image retrieval.** Recall@1, Recall@4 (%) and Normalized Mutual Information (NMI, $\in [0, 1]$) on four fine-grained benchmarks (CUB-200-2011, Cars-196, FGVC-Aircraft, and our iNat-Aves subset of iNaturalist-2021). All methods share the same Qwen3-VL-8B vision tower; baselines use mean pooling. Baselines: **PA** = Proxy Anchor [13], **PF** = Potential Field [3]. Best per column in **bold**, second-best underlined. $\pm$ values for SAGA are standard deviation over 3 random seeds

	CUB-200			Cars-196			Aircraft			iNat-Aves		
Method	R@1	R@4	NMI	R@1	R@4	NMI	R@1	R@4	NMI	R@1	R@4	NMI
Pre-trained backbone	75.6	91.8	0.77	70.7	88.5	0.49	53.1	76.1	0.43	42.2	64.8	0.65
PA	79.5	92.0	0.79	93.4	97.3	0.84	73.1	92.8	0.68	54.1	73.1	0.72
PF	81.6	92.9	0.81	93.7	97.8	0.86	77.4	93.2	0.72	55.6	75.0	0.73
SAGA (ours)	$\mathbf{87.9}_{\pm 0.3}$	$\mathbf{96.3}_{\pm 0.2}$	**0.83**	$\mathbf{97.0}_{\pm 0.3}$	$\mathbf{98.6}_{\pm 0.1}$	**0.89**	$\mathbf{83.5}_{\pm 0.4}$	$\mathbf{93.9}_{\pm 0.3}$	**0.77**	$\mathbf{60.1 \pm 0.4}$	$\mathbf{77.1 \pm 0.3}$	**0.80**

The deployed system consists only of the trained vision encoder and pooler, so its inference cost is identical to any standard DML pipeline; the MLLM serves solely as a training-time supervisor.

4 Experiments

4.1 Setup

Datasets: We empirically compare our method (SAGA) against state-of-the-art DML baselines on four zero-shot image retrieval benchmarks: **(1)** the CUB-200-2011 dataset [30] consisting of 11,788 images from 200 bird species, **(2)** the Cars-196 dataset [15] containing 16k images from 196 car model categories, **(3)** the FGVC-Aircraft dataset [18] with 10,000 images from 100 aircraft variants, and **(4)** the **iNat-Aves** benchmark we curate from the iNaturalist-2021 dataset [29]: starting from the `train_mini` split (50 img/ species) we retain only the taxonomic class *Aves*, yielding $\sim$1,486 species and $\sim$74k images.

Classes in all four benchmarks are distinguished by visual attributes that an MLLM can reason about. We exclude the product-retrieval benchmarks SOP [28] and In-Shop [17], since their classes separate on object identity rather than fine-grained attributes. Per-dataset prompt templates, preprocessing details, and full dataset statistics are reported in Appendices A and E.

Evaluation Settings: Following the standard zero-shot retrieval protocol of prior DML work [28, 13, 33, 3], classes are partitioned into disjoint train and test halves and the model is evaluated on *unseen* classes at 224×224 resolution. CUB-200-2011 and Cars-196 use the canonical splits; for FGVC-Aircraft and iNat-Aves we apply the same class-disjoint half-split convention (first half train, second half test; full details in Appendix A). We report Recall@K (fraction of queries with a same-class neighbour among the K nearest) and Normalized Mutual Information (NMI), computed between k-means cluster assignments on the test embeddings (with k equal to the number of test classes) and the ground-truth labels, capturing both nearest-neighbour and global embedding-space structure.

Backbone: We use **Qwen3-VL-8B** [1] as our MLLM for our main results, with its vision tower instantiating the encoder f_θ and its language backbone serving as the frozen supervisor p_ψ. All DML baselines use the same Qwen3-VL-8B vision tower for fair comparison; baselines use mean pooling over patch tokens, while our full method uses our learned attention pooler. The pooler outputs ℓ_2-normalized embeddings of dimension $D_e = 4096$.

Training parameters: The encoder and pooler are trained with AdamW with cosine-annealed learning rates, GRPO group size $G = 8$, and $P = 8$ balanced same/different-class pairs per step. All experiments use a single NVIDIA H200 (141 GB) GPU with bfloat16 mixed precision. Full hyperparameter values are reported in Appendix B.

4.2 Image Retrieval Performance

As seen in Table 1, our method significantly outperforms standard DML baselines on all four fine-grained datasets. It outperforms the best-performing baseline, PotentialField [3], in terms of Recall@1 (R@1) by $\mathbf{6.3}\%$ on CUB-200-2011, $\mathbf{3.3}\%$ on Cars-196, $\mathbf{6.1}\%$ on FGVC-Aircraft, and $\mathbf{4.5}\%$ on

Table 2: **Loss component ablation** on CUB-200-2011 and FGVC-Aircraft (R@1, %). All configurations include the DML term ($\mathcal{L}_{\text{DML}}$), instantiated as PF; ticks indicate which losses are added. The indented italic row swaps the per-dataset attribute list for a generic prompt (App. E.3) to test prompt sensitivity. Baselines: **PF** = Potential Field [3].

$\mathcal{L}_{\text{GRPO}}$	$\mathcal{L}_{\text{KL}}$		CUB	FGVC-A
×	×	PF only	81.6	77.4
✓	×	+ GRPO	87.0	82.3
		(generic prompt)	*84.1*	*79.6*
×	✓	+ KL	82.1	78.1
✓	✓	**SAGA**	**87.9**	**83.5**

Table 3: **DML loss-agnostic ablation** on CUB-200-2011 and FGVC-Aircraft (R@1, %). *bare*: DML loss alone; *SAGA*: same DML loss combined with our GRPO + KL alignment. **SAGA w/ PF** is the headline configuration of the main paper. Baselines: **PA** = Proxy Anchor [13], **MS** = Multi-Similarity [33], **PF** = Potential Field [3].

	CUB-200 R@1		**Aircraft** R@1	
Method	bare	SAGA	bare	SAGA
MS	77.8	86.1	72.5	82.3
PA	79.5	87.3	73.1	82.7
PF	81.6	**87.9**	77.4	**83.5**

iNat-Aves, and shows similar margins over ProxyAnchor [13]. The performance gains are largest on the most attribute-driven benchmarks (birds and aircraft variants), consistent with our hypothesis: these classes are distinguished by subtle attributes (bill shape, wing-bars, eye rings for birds; tail and wing geometry for aircraft) that the MLLM explicitly reasons about during GRPO training, rather than by coarse object identity. The substantial gain on iNat-Aves further demonstrates that this attribute-aware supervision scales to a much larger label space ($\sim$743 training species). The improvement persists at $R@4$ and in NMI, indicating that the benefit extends to the overall structure of the embedding space. Qualitative comparisons are in Sec. 4.4.

4.3 Ablation Studies

Unless otherwise stated, all ablations follow the experimental setting of Sec. 4.1: we use the Qwen3-VL-8B vision tower with our learned attention pooler, train and evaluate on CUB-200-2011 (the most attribute-driven of our four benchmarks) and FGVC-Aircraft, and use the hyperparameters reported in Sec. 4.1 (full values in Appendix B). We report Recall@1 on the held-out test classes of both datasets.

Loss component analysis: Table 2 isolates the contribution of each auxiliary loss on top of the PF baseline (PF = Potential Field [3]). Adding the GRPO term improves R@1 by $\mathbf{5.4}\%$ on CUB-200-2011 and $\mathbf{4.9}\%$ on FGVC-Aircraft, while the KL alignment term alone yields a much smaller 0.5% / 0.7% gain on the same datasets, confirming that the GRPO signal contributes the bulk of the attribute-aware supervision and KL plays a complementary, narrower role of supervising the pooler attention. Combining all three losses (**SAGA**) gives the strongest configuration, exceeding the PF baseline by $\mathbf{6.3}\%$ on CUB-200-2011 and $\mathbf{6.1}\%$ on FGVC-Aircraft.

DML loss-agnostic ablation: Table 3 replaces the PF term inside SAGA with two alternative DML losses, PA (Proxy Anchor [13]) and MS (Multi-Similarity [33]). All three SAGA variants substantially exceed their bare-DML baselines: on CUB-200-2011 the SAGA gains over the bare DML loss are $\mathbf{8.3}\%$, $\mathbf{7.8}\%$, and $\mathbf{6.3}\%$ for MS, PA, and PF respectively, with comparable or larger gains of $\mathbf{9.8}\%$, $\mathbf{9.6}\%$, and $\mathbf{6.1}\%$ on FGVC-Aircraft. The consistency of the gains across DML losses confirms that SAGA is DML loss-agnostic.

Prompt sensitivity: To separate the contribution of the attribute vocabulary from generic MLLM-oracle access, we re-run `+GRPO` with the per-dataset attribute list removed (generic variant in Appendix E.3); the KL term is omitted because its distillation target (attention pooled over named-attribute spans) is undefined without an attribute vocabulary. The italicised *(generic prompt)* row in Table 2 shows the GRPO lift over PF drops from $+5.4/+4.9$ to $+2.5/+2.2$ R@1 on CUB-200-2011 / FGVC-Aircraft, a recovery of roughly 45%. A generic MLLM oracle therefore accounts for about half of the GRPO gain, and the attribute vocabulary contributes the remaining, confirming that attribute-aware reasoning is a meaningful and quantifiable component beyond a generic LLM-as-oracle baseline.

Additional ablations over embedding dimension and MLLM used are reported in Appendix C.

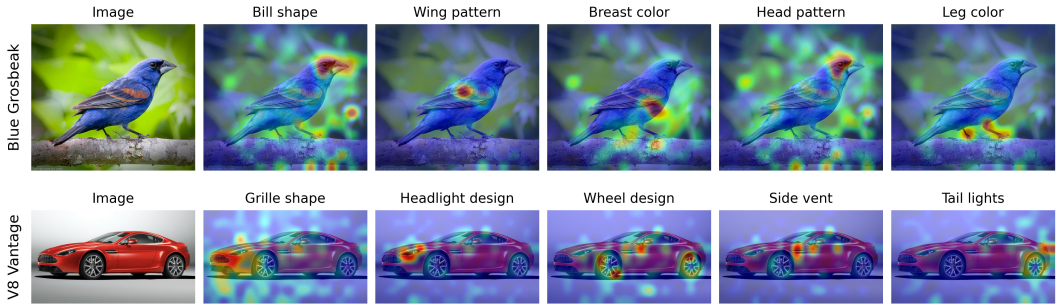

Figure 3: **MLLM supervisor attention over named attributes (KL target).** For a held-out CUB-200-2011 query (top, Blue Grosbeak) and a Cars-196 query (bottom, V8 Vantage), we overlay the MLLM's attention pooled over the reasoning tokens that name each attribute. For the bird, attention localizes on the bill, wing, breast, head, and legs as each attribute is named; for the car, on the grille, headlights, wheels, side vent, and tail lights. These per-attribute spatial maps are exactly the targets that the KL alignment term distills into the vision pooler.

4.4 Attention Analysis

Figure 3 visualizes the MLLM supervisor attention that the KL alignment term distills, on two held-out queries: a CUB-200-2011 Blue Grosbeak and a Cars-196 V8 Vantage. For each attribute the MLLM names in its discriminative-reasoning trace, we pool the supervisor's attention over the tokens corresponding to that attribute and overlay the result on the input image. In every column the mass concentrates on the named attribute region rather than on the bird or car as a whole. This is direct visual evidence that the supervisor signal is attribute-resolved, not a coarse object-vs-background prior, and motivates the per-attribute KL loss in Sec. 3: by aligning the pooler's attention with these maps, the vision encoder inherits the attribute-level spatial discrimination that drives the retrieval gains in Table 1. Retrieval-level qualitative comparisons (top-k images per query, with correct/incorrect class borders, including failure cases) are deferred to Appendix D.

5 Limitations

Training cost. Training under our framework is substantially more expensive than standard DML. Each sampled pair requires G rollouts through the frozen MLLM, and each contributing pair a differentiable replay through the language backbone; on a single H200 this makes SAGA 34 to $55\times$ slower than PF per dataset (31 vs. 0.6 GPU-hours on CUB-200-2011; full breakdown in Appendix B). This cost is paid once at training time; inference uses the vision encoder and pooler alone and is identical in cost to a vanilla DML pipeline. The framework also presumes a supervisor capable of following the structured comparison prompt and resolving a non-trivial fraction of pairs correctly, since GRPO produces gradient signal only when rollouts disagree on the verdict. Open-weight MLLMs that we use meet this requirement on standard fine-grained benchmarks.

Attributes must be verbalizable. The supervision signal is bounded by what the frozen MLLM can detect and describe in language. Cues such as texture statistics, repeating patterns, or relative and metric geometry, which the MLLM cannot reliably name, yield vague descriptions and a weak GRPO signal, making our method unsuitable for some DML applications such as place recognition and Re-ID, face recognition.

Identity retrieval. SAGA targets benchmarks whose classes differ in nameable attributes. In identity-retrieval benchmarks such as SOP and In-Shop, classes are individual product instances with 5 to 6 images each: many class pairs share no attribute while others share nearly all of them, and covering the product variation requires a very large attribute pool with few examples of each combination. Rollouts therefore tend to be unanimous and receive zero advantage. Under identical settings, the contributing-pair rate (the fraction of sampled pairs with $\sigma_r > 0$) is 8% on SOP and 11% on In-Shop, against 56% on CUB-200-2011, so most pairs are skipped and supervision largely

collapses to the DML term. We found that SAGA did not meaningfully improve performance for these benchmarks at all.

6 Conclusion

We introduced SAGA, a framework that turns a frozen MLLM into a training-time supervisor for the vision encoder of a retrieval system. Where class-label DML reduces a pair to a scalar that acts on every embedding direction in unison, GRPO over the MLLM's verdict yields a gradient whose group-normalized advantages cancel on tokens the rollouts agree on and concentrate on the discriminating ones, routing signal into precisely the directions that resolve the attributes the supervisor used to judge. A KL term distills the supervisor's attention over its discriminative-reasoning tokens into the pooler, and a standard metric loss shapes the geometry. The MLLM is frozen throughout and discarded at inference, so deployment cost matches a vanilla DML pipeline; on CUB-200-2011, Cars-196, FGVC-Aircraft, and iNaturalist Aves, this lifts Recall@1 by 3 to 6 points over the strongest baselines on the same backbone. We view the binary verdict as the simplest instance of a broader principle, that coarse rewards adjudicated by a reasoning supervisor can carry far more structure into the gradient than they appear to, and see this as a promising lever for representation learning whenever fine-grained annotation is unavailable.

Acknowledgments and Disclosure of Funding

This work was supported by the Office of Naval Research under grant N00014-24-1-2169, the IBM-Illinois Discovery Accelerator Institute (IIDAI), and the Amazon-Illinois Center on AI for Interactive Conversational Experiences. This research used the Delta system at NCSA through allocation CIS230360 from the ACCESS program, which is supported by National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296.

References

[1] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025. URL `https://arxiv.org/abs/2511.21631`.

[2] Shubhang Bhatnagar and Narendra Ahuja. Piecewise-linear manifolds for deep metric learning. In *Conference on Parsimony and Learning (Proceedings Track)*, 2023. URL `https://openreview.net/forum?id=Z0Fk5MyxkY`.

[3] Shubhang Bhatnagar and Narendra Ahuja. Potential field based deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 25549–25559, June 2025.

[4] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24185–24198, 2024.

[5] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005.

[6] Dwip Dalal, Gautam Vashishtha, Utkarsh Mishra, Jeonghwan Kim, Madhav Kanda, Hyeonjeong Ha, Svetlana Lazebnik, Heng Ji, and Unnat Jain. Constructive distortion: Improving MLLMs

with attention-guided image warping. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=SRl0xy0UOj.

[7] Yue Fan, Xuehai He, Diji Yang, Kaizhi Zheng, Ching-Chen Kuo, Yuting Zheng, Sravana Jyothi Narayanaraju, Xinze Guan, and Xin Eric Wang. Grit: Teaching mllms to think with images, 2025. URL https://arxiv.org/abs/2505.15879.

[8] Aaron Grattafiori et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

[9] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[10] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, pages 1735–1742. IEEE Computer Society, 2006. doi: 10.1109/CVPR.2006.100.

[11] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[12] Yifan Hou, Buse Giledereli, Yilei Tu, and Mrinmaya Sachan. Do vision-language models really understand visual language?, 2025. URL https://arxiv.org/abs/2410.00193.

[13] Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. Proxy anchor loss for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[14] Sungyeon Kim, Boseung Jeong, and Suha Kwak. Hier: Metric learning beyond class labels via hierarchical regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19903–19912, 2023.

[15] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

[16] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 34892–34916. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/6dcf277ea32ce3288914faf369fe6de0-Paper-Conference.pdf.

[17] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1096–1104, 2016. doi: 10.1109/CVPR.2016.124.

[18] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[19] Sachit Menon and Carl Vondrick. Visual classification via description from large language models, 2022. URL https://arxiv.org/abs/2210.07183.

[20] Yair Movshovitz-Attias, Alexander Toshev, Thomas K Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *Proceedings of the IEEE international conference on computer vision*, pages 360–368, 2017.

[21] Jinhee Park, Jisoo Park, Dagyeong Na, and Junseok Kwon. Deep disentangled metric learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 19830–19838, 2025.

[22] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

[23] Samyak Rawlekar, Shubhang Bhatnagar, Vishnuvardhan Pogunulu Srinivasulu, and Narendra Ahuja. Improving multi-label recognition using class co-occurrence probabilities. In *International Conference on Pattern Recognition*, pages 424–439. Springer, 2024.

[24] Samyak Rawlekar, Amitabh Swain, Yujun Cai, Yiwei Wang, Ming-Hsuan Yang, and Narendra Ahuja. Finding distributed object-centric properties in self-supervised transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 31326–31335, 2026.

[25] Oindrila Saha, Grant Van Horn, and Subhransu Maji. Improved zero-shot classification by adapting vlms with text descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17542–17552, 2024.

[26] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[27] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y.K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

[28] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[29] Grant Van Horn, Elijah Cole, Sara Beery, Kimberly Wilber, Serge Belongie, and Oisin MacAodha. Benchmarking Representation Learning for Natural World Image Collections . In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12879–12888, Los Alamitos, CA, USA, June 2021. IEEE Computer Society. doi: 10.1109/CVPR46437.2021.01269. URL https://doi.ieeecomputersociety.org/10.1109/CVPR46437.2021.01269.

[30] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. Technical report, California Institute of Technology, 2011.

[31] Haochen Wang, Xiangtai Li, Zilong Huang, Anran Wang, Jiacong Wang, Tao Zhang, Jiani Zheng, Sule Bai, Zijian Kang, Jiashi Feng, Zhuochen Wang, and Zhaoxiang Zhang. Traceable evidence enhanced visual grounded reasoning: Evaluation and methodology, 2026. URL https://arxiv.org/abs/2507.07999.

[32] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

[33] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R. Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[34] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

[35] Zhiyuan Yan, Kaiqing Lin, Zongjian Li, Junyan Ye, Hui Han, Haochen Wang, Zhendong Wang, Bin Lin, Hao Li, Xinyan Xiao, Jingdong Wang, Haifeng Wang, and Li Yuan. Unified multimodal models as auto-encoders, 2026. URL https://arxiv.org/abs/2509.09666.

[36] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, YuYue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=2a36EMSSTp.

[37] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Sks9_ajex.

[38] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11975–11986, October 2023.

[39] Jiarui Zhang, Mahyar Khayatkhoei, Prateek Chhikara, and Filip Ilievski. MLLMs know where to look: Training-free perception of small visual details with multimodal LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2502.17422.

Supplementary Material for
Beyond Scalar Distances: Semantic Attribute Gradients from Frozen MLLMs for Visual Embeddings

In this supplementary material, we provide additional information that did not fit in the main paper. We do so in five sections: Sec. A gives full statistics and licenses for the four image-retrieval benchmarks; Sec. B reports implementation and optimization details (training algorithm, optimizer, loss weights, batching, hardware); Sec. C reports additional ablations omitted from Sec. 4.3 for space, including ablations over embedding dimension and vision backbone; Sec. D shows top-5 nearest-neighbor rankings on held-out queries with class-correctness color coding; finally, Sec. E reproduces the structured pair-comparison prompt T_{inst} template used by the supervisor MLLM, with per-dataset substitutions and the attribute vocabularies that instantiate the template for the four datasets in our main experiments.

A Dataset Details

We provide additional details for each of the four fine-grained image retrieval benchmarks used in our main experiments. All datasets are publicly available under their original licenses; we use them solely for non-commercial academic research. Across all benchmarks we follow the standard zero-shot retrieval protocol of Song et al. [28]: classes are partitioned into disjoint training and evaluation halves, and models are evaluated on classes never seen during training. The structured comparison prompt T_{inst} used by the supervisor MLLM and the per-dataset attribute vocabularies that instantiate it for each dataset above are reported at the end of this supplement in Appendix E.

CUB-200-2011 [30]. The Caltech-UCSD Birds 200-2011 dataset contains $11{,}788$ images of 200 bird species (≈ 59 images per class). Each image is annotated with 312 binary attributes spanning 28 attribute groups (bill shape, plumage color, wing pattern, etc.), 15 part-location keypoints, and a single bounding box. We use the first 100 species for training and the remaining 100 for evaluation. Birds are distinguished by subtle attribute combinations such as bill shape, plumage patterning, wing-bar presence, and eye-ring color, making CUB the canonical benchmark for attribute-aware retrieval.

Cars-196 [15]. The Stanford Cars dataset contains $16{,}185$ images of 196 car classes defined at the make-model-year level (e.g., *2012 Tesla Model S Sedan*). Following the zero-shot DML convention of Song et al. [28], the first 98 classes are used for training and the remaining 98 for evaluation. Classes are distinguished by external visual cues such as body style, grille and headlight design, side profile, badge placement, and apparent era.

FGVC-Aircraft [18]. The Fine-Grained Visual Classification of Aircraft dataset [18] ships approximately $10{,}000$ images organized hierarchically (manufacturer, family, variant). We retrieve at the variant level using the standard 100-variant release. Following the same disjoint-class convention as CUB and Cars, we sort variants alphabetically and split the 100 classes into the first 50 for training and the remaining 50 for evaluation, pooling FGVC's own `trainval` and `test` image partitions before the class-level split (since our train/eval classes are already disjoint, the original image-level split is irrelevant). Discriminative cues include wing configuration (low / mid / high / T-tail), engine count and mounting position, fuselage profile, and vertical-stabilizer geometry.

iNaturalist 2021 Aves [29]. We use the Aves (birds) supercategory from the `train_mini` split of the iNaturalist 2021 competition, comprising $1{,}486$ species at 50 images per species ($\approx 74{,}300$ images total). Following the same disjoint-class protocol, we sort species directories lexicographically (zero-padded iNat category-id prefix) and use the first 743 species for training and the remaining 743 for evaluation. Compared to CUB, iNat-Aves covers a substantially broader taxonomic range and contains images captured by the iNaturalist citizen-science community under highly varied conditions (lighting, pose, partial occlusion, cluttered natural backgrounds), making it an open-set fine-grained benchmark much closer to real-world species identification.

Image preprocessing. All images are resized to 224×224 before being passed to the Qwen3-VL-8B vision encoder, matching the pre-training resolution of the base model. We do not crop using bounding-box annotations, so the encoder sees the full image including background context.

B Additional Implementation Details

Algorithm 1 SAGA: One Training Step

Require: Batch stream $\mathcal{B}$, target contributing pairs K, group size G, loss weights $\lambda_{\text{dml}}, \lambda_{\text{lm}}, \lambda_{\text{kl}}$

Phase A: Embedding & DML (per micro-batch)
$\{z_i\}_{i=1}^{B} \leftarrow c_\phi(f_\theta(I_i))$; backward $\lambda_{\text{dml}} \cdot \mathcal{L}_{\text{DML}}(\{z_i\}, \{y_i\})$
Phase B: GRPO Rollouts (no grad, DAPO Dynamic Sampling)
Buffer $\mathcal{C} \leftarrow \emptyset$
while $|\mathcal{C}| < K$ **do**
  Sample image pair (I_a, I_b) from $\mathcal{B}$ (refill micro-batches as needed)
  Sample G completions $\{\hat{Y}^{(g)}\}_{g=1}^{G}$ from p_ψ; parse rewards $\{r^{(g)}\}$, advantages $\{A^{(g)}\}$
  if $\sigma_r > 0$ **then**
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{(\{\hat{Y}^{(g)}\}, \{A^{(g)}\}, \{r^{(g)}\})\}$
  end if
end while
Phase C: Policy Update (with grad)
Recompute log-probs through $f_\theta \rightarrow p_\psi$ for all rollouts in $\mathcal{C}$ (c_ϕ bypassed)
Compute $\mathcal{L}_{\text{GRPO}}$ via Eq. (5) over $\mathcal{C}$ (token-level normalization across the buffer); backward $\lambda_{\text{lm}} \cdot \mathcal{L}_{\text{GRPO}}$
For each rollout in $\mathcal{C}$ with $r^{(g)} = 1$: compute $\mathcal{L}_{\text{KL}}$ via Eq. (7) ($\ell = 26$ teacher attention); backward $\lambda_{\text{kl}} \cdot \mathcal{L}_{\text{KL}}$
Gradient clip; optimizer step

Inference. At test time the GRPO rollouts, KL alignment, and the frozen MLLM supervisor are discarded; only the vision encoder f_θ and attention pooler c_ϕ remain, producing a single ℓ_2-normalized embedding per image used directly for nearest neighbor retrieval (Fig. 4).

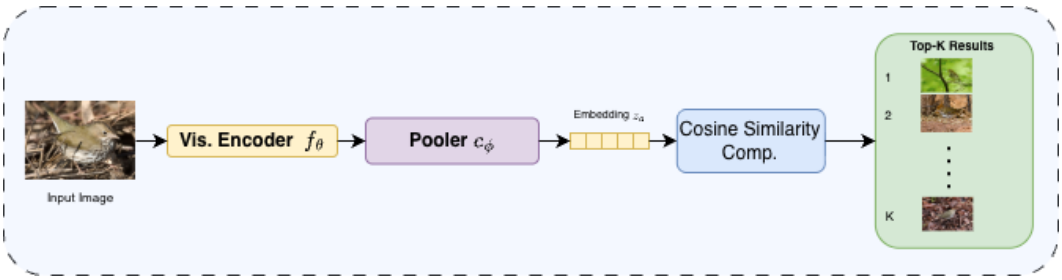

Figure 4: Inference-time pipeline. Only f_θ and c_ϕ remain; the frozen MLLM, GRPO rollouts, and KL alignment are dropped. Each image yields a single $\ell_2$normalized embedding used directly for retrieval.

Attention pooler architecture. The attention pooler c_ϕ is a single-query cross-attention head over the patch tokens $\mathbf{X} \in \mathbb{R}^{N_p \times D}$. A learnable query vector $\mathbf{q} \in \mathbb{R}^D$ and a linear key projection $W_k \in \mathbb{R}^{D \times D}$ produce the patch-attention distribution $\boldsymbol{\beta} = \text{softmax}\Big((\mathbf{q}W_k\mathbf{X}^\top)/\sqrt{D}\Big) \in \Delta^{N_p}$, the pooled vector $\boldsymbol{\beta}\mathbf{X}$ is mapped to the embedding dimension D_e by a linear projection, and the resulting embedding is $\ell_2$normalized before being passed to either $\mathcal{L}_{\text{DML}}$ the retrieval or the distance. The attention weights $\boldsymbol{\beta}$ supervised by $\mathcal{L}_{\text{KL}}$ (Eq. 7) are exactly the softmax output of this single cross-attention layer; no additional importance-scoring head is used. The DML baselines in Table 1 use mean pooling over patch tokens; we verified on CUB-200-2011 and FGVC-Aircraft that swapping mean for max pooling moves PA and PF R@1 by less than $\sim$0.5% with no consistent winner, so mean was selected for its conventionality in the DML literature.

Optimisation. The vision encoder f_θ and the attention pooler c_ϕ are trained with AdamW at learning rates 2×10^{-5} and 1×10^{-4} respectively, with cosine annealing over 3 epochs and a linear warm-up over the first 5% of steps. Gradients are clipped at global norm 1.0. The frozen language backbone p_ψ receives no gradient updates throughout training.

Loss weights. We use $\lambda_{\text{dml}} = \lambda_{\text{lm}} = \lambda_{\text{kl}} = 1$. Each loss term is internally normalized before the weight is applied: $\mathcal{L}_{\text{DML}}$ is averaged over the batch, $\mathcal{L}_{\text{GRPO}}$ over generated tokens (Eq. 5), and $\mathcal{L}_{\text{KL}}$ over attribute-token positions per image (Eq. 7); these per-loss normalizations bring the gradient magnitudes of the three terms to within roughly an order of magnitude of each other at initialization, so a unit weight on each performs well without further tuning. We did not perform a formal sweep over the weights; preliminary CUB-200 runs at $\lambda_{\text{lm}}, \lambda_{\text{kl}} \in \{0.5, 1, 2\}$ produced indistinguishable R@1.

Attention extraction for $\mathcal{L}_{\text{KL}}$. The teacher attention $\boldsymbol{\alpha}$ in Eq. 7 is taken from layer $\ell = 26$ of the Qwen3-VL-8B language backbone, head-averaged, and renormalized over the patches of the corresponding image. As discussed in Sec. 3.4, the last layer is dominated by attention-sink artefacts; layer $\ell = 26$ is the layer selected by the attention-mass criterion of Sec. 3.4.

Batching and GRPO sampling. At each micro-batch we draw a class-balanced batch of size 64 and sample candidate same-/different-class pairs from it; for each pair we roll out $G = 8$ MLLM completions with temperature 0.7, top-p 0.95, and at most 1024 generated tokens. Following DAPO Dynamic Sampling [36], we accumulate contributing pairs ($\sigma_r > 0$) across successive micro-batches until $K = 8$ have buffered, then take a single optimizer step over the buffered rollouts. Pairs with $\sigma_r = 0$ contribute neither to $\mathcal{L}_{\text{GRPO}}$ nor $\mathcal{L}_{\text{KL}}$ but still receive the DML gradient. $\mathcal{L}_{\text{KL}}$ is computed only on the rollouts within a buffered pair that received reward $r^{(g)} = 1$. The Qwen3-VL-8B supervisor produces well-formed JSON essentially always, so a fallback parser was unnecessary.

Hardware. All experiments use a single NVIDIA H200 (141 GB) GPU with bfloat16 mixed precision.

Training cost. Table 4 reports the training cost of SAGA against PF, measured as wall-clock hours on a single H200. Proxy Anchor's training cost is within 15% of PF's (both are pure-DML runs over the same Qwen3-VL-8B vision tower), so we report PF, which is also the DML loss used to train SAGA, as the DML reference. The contributing-pair rate is the fraction of sampled pairs with $\sigma_r > 0$ that survive DAPO Dynamic Sampling and enter $\mathcal{L}_{\text{GRPO}}$, averaged over the first 100 steps on each dataset.

Table 4: Training cost per dataset (GPU-h = wall-clock hours on a single H200).

Dataset	PF (GPU-h)	SAGA (GPU-h)	Slowdown	Contributing-pair rate (%)
CUB-200-2011	0.6	31	$52\times$	56.5
Cars-196	0.7	33	$47\times$	35.3
FGVC-Aircraft	0.4	22	$55\times$	41.4
iNat-Aves	3.3	112	$34\times$	43.8

Per-step timing. A step is one optimizer update, and its unit differs between the two methods: a PF step is a single batch of 64 images, while a SAGA step buffers $K = 8$ contributing pairs across successive DAPO refill micro-batches (Algorithm 1). On CUB-200-2011 we measure 0.47 s per PF step and 31 s per SAGA step, averaged over the first 100 steps. Each sampled pair receives $G = 8$ rollouts averaging 622.2 generated tokens, and every buffered rollout is then replayed once through the frozen 8B backbone with gradients enabled to recompute log-probabilities (Phases B and C of Algorithm 1). Pairs with $\sigma_r = 0$ are skipped and cost only no-grad generation. The resulting one-time increase of roughly $50\times$ in training cost is not incurred at retrieval time, where the deployed model is the encoder and pooler alone (Sec. 3.6).

C Additional Ablations

This section reports extended ablations and per-dataset breakdowns for the analyses presented in Sec. 4.3 of the main paper. Unless otherwise stated, all experiments follow the setting of Sec. 4.1 of the main paper: the Qwen3-VL-8B vision tower with our learned attention pooler, AdamW with the schedule and hyperparameters reported in Appendix B, GRPO group size $G = 8$ and DAPO target $K = 8$ contributing pairs per step, and the zero-shot retrieval evaluation protocol with Recall@K and NMI on held-out classes. Each subsection below extends a specific ablation from Sec. 4.3 of the main paper to additional datasets, hyperparameter ranges, or design choices that did not fit in the main text.

C.1 Pooling Strategy Across Datasets

Context: Sec. 3.1 introduces our learned attention pooler c_ϕ, which the KL alignment term ($\mathcal{L}_{\text{KL}}$) supervises against the supervisor's discriminative attention. To isolate the contribution of the pooler from that of the GRPO + KL training, we compare against mean pooling under matched training settings.

Experiment: We run PF + our attention pooler (Appendix B), with the pooler trained with only DML and no GRPO and no KL, on 3 datasets following the setting and protocol of Sec. 4.1.

Table 5: **Pooler control.** R@1 (%) on CUB-200-2011, Cars-196, and FGVC-Aircraft. **PA** = Proxy Anchor [13], **PF** = Potential Field [3].

Method	CUB	Cars	Aircraft
PA, mean pool	79.5	93.4	73.1
PF, mean pool	81.6	93.7	77.4
PF + attn pooler, DML only	81.5	93.6	77.8
SAGA (PF + attn pooler + GRPO + KL)	**87.9**	**97.0**	**83.5**

Results: The learned pooler on its own contributes essentially nothing (-0.1 / -0.1 / $+0.4$ R@1 over mean-pooled PF), and SAGA's improvement over this architecture-matched baseline holds. This is consistent with the motivation for $\mathcal{L}_{\text{KL}}$ (Sec. 3.4), as the pooler does give consistent performance gains when trained with it (Table 2).

C.2 Lower-Dimensional Embeddings

Context: The main paper uses embeddings of dimension $d = 4096$. In storage- or compute-constrained retrieval settings (*e.g.* on-device species recognition, large-scale gallery indexing) lower-dimensional embeddings are preferable. We verify that SAGA's gain over the baselines is preserved when the embedding is compressed to $d \in \{512, 128\}$, with $d = 512$ matching the standard DML choice in prior work and $d = 128$ being a more aggressive compression target.

Experiment: We re-train SAGA and the PotentialField [3] baseline, and evaluate the zero-shot encoder, at $d \in \{512, 128\}$, otherwise following the standard setting of Sec. 4.1. Results are reported on CUB-200-2011 and FGVC-Aircraft.

Results: Table 6 reports R@1 and R@4 at $d \in \{512, 128\}$ on CUB-200-2011 and FGVC-Aircraft. The method ordering mirrors Table 1 (which uses the main paper's $d = 4096$): SAGA retains its R@1 margin over PF at both compressed dimensions, indicating that the attribute-aware GRPO signal continues to deliver gains in the small-embedding regime relevant to deployment.

C.3 Vision Backbone Transfer

Context: The main paper uses the Qwen3-VL-8B [1] vision tower throughout. To test whether SAGA's gain transfers beyond a single MLLM family, we re-train the pipeline with the vision tower swapped to InternVL3.5-8B [32], keeping the attention pooler, GRPO, and KL alignment unchanged.

Experiment: We train SAGA and the PotentialField [3] baseline on CUB-200-2011 using the InternVL3.5-8B vision tower as the encoder f_θ. Note that this means that supervisor LM p_ψ is also

Table 6: **Lower-dimensional embeddings.** R@1 and R@4 (%) at $d = 512$ and $d = 128$ on CUB-200-2011 and FGVC-Aircraft (the main paper uses $d = 4096$; see Sec. 4.1). The method ordering is preserved at both compressed dimensions. **PF** = Potential Field [3].

	$d = 512$				$d = 128$			
	CUB-200-2011		FGVC-Aircraft		CUB-200-2011		FGVC-Aircraft	
Method	R@1	R@4	R@1	R@4	R@1	R@4	R@1	R@4
Pre-trained backbone	75.0	91.6	50.5	74.1	71.7	90.2	45.8	70.2
PF	79.4	91.8	76.3	92.2	78.3	91.5	75.5	91.5
SAGA	**86.5**	**93.1**	**80.1**	**93.3**	**83.8**	**92.7**	**79.5**	**91.7**

the InternVL3.5-8B language backbone in these runs. We additionally report the zero-shot retrieval recall of the InternVL3.5-8B encoder (no fine-tuning) as a no-training reference. Due to compute constraints, this transfer study is limited to CUB-200-2011.

Table 7: **Vision backbone transfer.** R@1, R@2, R@4, and R@8 (%) on CUB-200-2011 when the vision tower of f_θ is swapped from Qwen3-VL-8B (main paper) to InternVL3.5-8B [32]. For the trained runs, the supervisor LM p_ψ is the InternVL3.5-8B language backbone. **PF** = Potential Field [3].

Method (encoder = InternVL3.5-8B)	R@1	R@2	R@4	R@8
Zero-shot	50.8	64.8	76.4	85.8
PF only	78.0	86.2	91.8	95.1
SAGA	**80.1**	**87.6**	**92.3**	**95.6**

Results: Table 7 reports R@1 through R@8 on CUB-200-2011 for the zero-shot encoder, the PF baseline, and SAGA under the alternate encoder/MLLM combination. The SAGA gain over PF persists with the swapped backbone, indicating that the attribute-aware GRPO signal is not tied to a single MLLM family.

C.4 Attribute-Token Gradient Analysis

Context: Sec. 3.2 argues that GRPO gradients excite the visual-token directions of attributes that differ between the two images, while on shared attributes the advantage-weighted sum vanishes. Here we provide direct gradient evidence for this mechanism through an analysis of attribute-token gradient norms.

Experiment: All encoder updates flow through the gradient of $\mathcal{L}_{\text{GRPO}}$ with respect to the visual tokens $\mathbf{X}$ (Eq. (5)). For each rollout group and each attribute, let s denote the positions of that attribute's value tokens in the structured JSON completion (Appendix E). We compute the norm of the advantage-weighted gradient

$$g_s = \Big\| \frac{\partial}{\partial \mathbf{X}} \Big[\sum_g A^{(g)} \sum_{t \in s} \log \pi_\theta\big(y_t^{(g)} \mid y_{<t}^{(g)}, S\big) \Big] \Big\|_2, \tag{10}$$

which is the exact quantity over which the argument of Sec. 3.2 is stated. We use the model at the start of training with the setting of Sec. 4.1 on CUB-200-2011, whose per-image attribute annotations (Appendix A) give both which attributes are visible in each image and their values. Every attribute of a pair is therefore labeled *shared* or *differing* from ground truth, independently of the MLLM's output. This yields 2,306 (pair, attribute) records over 114 pairs.

Table 8: Ratio of mean gradient norm g_s for differing vs. shared attributes on CUB-200-2011, grouped by CUB attribute type.

Attribute type	CUB attribute groups	#Groups	Ratio (differing/shared)
All attributes	all of the below	28	**5.9**
Part-level color	wing, upperparts, underparts, back, upper tail, under tail, breast, throat, belly, nape, forehead, crown, eye, bill, leg, primary	16	8.5
Part-level pattern	breast, head, back, tail, belly, wing	6	3.7
Shape/size	bill shape, bill length, wing shape, tail shape, overall shape, size	6	1.7

Results: Differing attributes receive a mean $5.9\times$ larger gradient norm than shared ones, with 23 of 28 attribute groups above $1\times$ (Table 8). The effect scales with how localized the attribute is: it is strongest on the fine part-level color traits that separate confusable species and smaller on the coarse morphology such pairs share. Sec. 3.2 states the mechanism for the idealized case in which the MLLM's descriptions faithfully track the true attributes. Since our shared/differing labels come from ground truth alone, any attribute the MLLM consistently misreads counts against the effect, so these ratios are a conservative estimate.

D Qualitative Retrieval Gallery

Context: We complement the attention-overlay analysis of Sec. 4.4 with retrieval-level qualitative results: top-5 nearest-neighbor rankings produced by the SAGA embedding on held-out test images.

Experiment: For each of the four benchmarks we report two held-out query images, both drawn from the standard zero-shot retrieval split (classes disjoint from training). To avoid both trivial wins (queries surrounded by easy same-class neighbors) and degenerate cases (queries whose image content is dominated by background), we stratify the candidate pool by the SAGA top-5 hit count: the first row per dataset is drawn from queries whose SAGA top-5 contains four or five same-class neighbors (*clean*), and the second from queries whose top-5 contains one to three same-class neighbors (*informative*). For each query we display the original image (leftmost column, neutral border) followed by its five nearest neighbors in descending cosine similarity, with green borders for same-class retrievals and red borders for cross-class errors. All embeddings are ℓ_2-normalized, and the query is excluded from its own retrieval set.

Results: On the *clean* rows (Fig. 5) SAGA returns same-class neighbors that are also visually consistent with the query. On the *informative* rows the wrong-class neighbors are visually plausible (similar pose, color, or silhouette), so the residual errors sit at the boundary between visually adjacent classes rather than across coarse categories.

E Prompts and Attributes

E.1 Prompt Template

The supervisor MLLM is queried with a structured pair-comparison prompt T_{inst} that asks it to (i) describe each of the two input images along a fixed list of visual attribute groups, (ii) summarize the key visual differences between the two images, and (iii) emit a same/different verdict in JSON. The verdict field of the JSON is parsed by the GRPO reward function (Sec. 3) to produce the binary reward $r \in \{0, 1\}$ used in the group-relative advantage estimate.

We use the same prompt structure across all four datasets, parameterized by (i) the expert role assumed by the model, (ii) the item word for the photographed object, (iii) the verdict question, and (iv) the dataset-specific attribute list reported in Sec. E.2 below. The generic template is reproduced verbatim below, exactly as it appears in our codebase; per-dataset substitutions are given in Table 9.

```
You are assisting an {EXPERT_NOUN} in identifying {WHAT_TO_IDENTIFY}
from photographs.
```

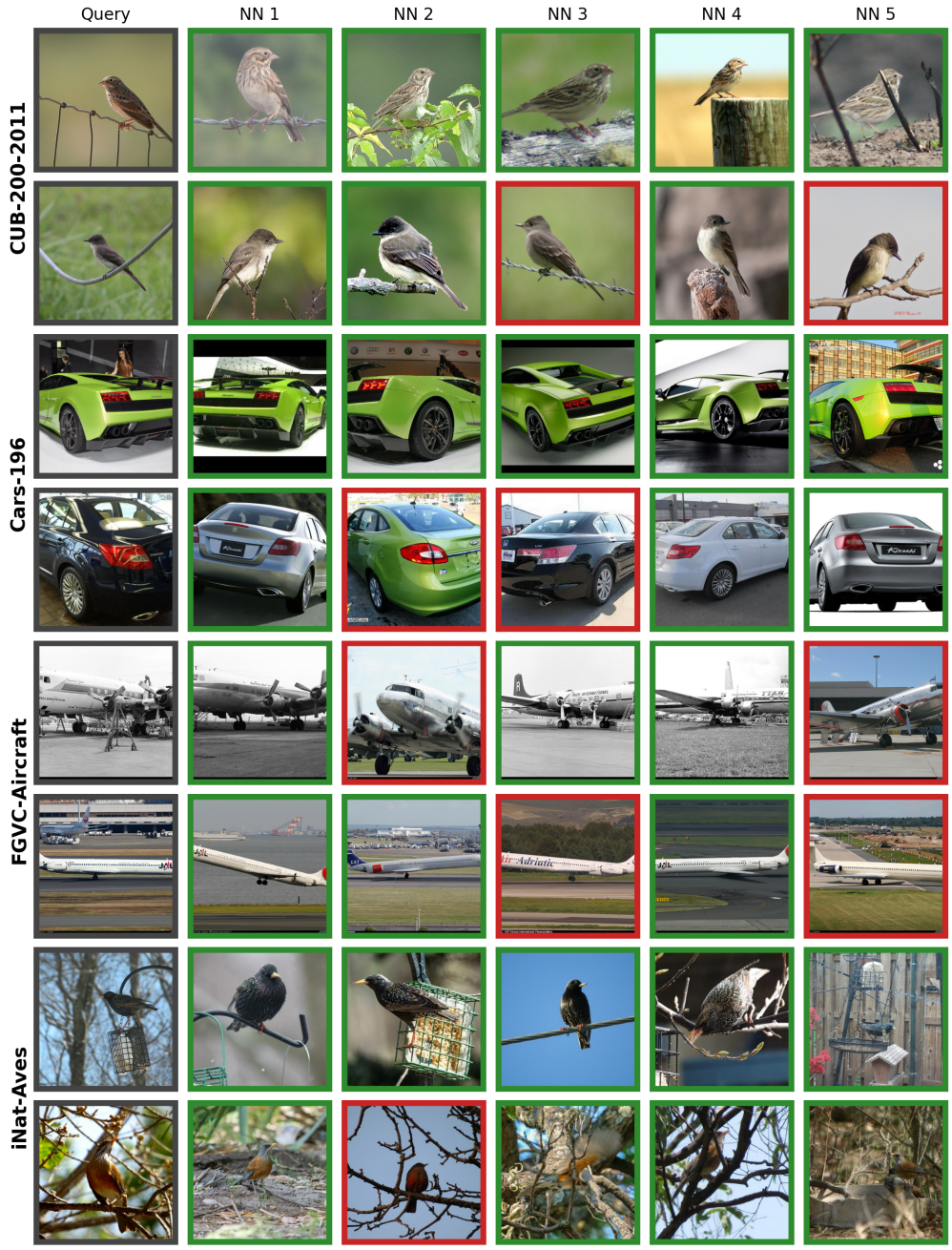

Figure 5: **Qualitative retrieval gallery on held-out test classes.** Two queries per dataset (rows), top-5 nearest neighbors under the SAGA embedding in descending cosine similarity (columns 2 to 6). The leftmost column is the query (neutral border). Green borders mark same-class retrievals; red borders mark cross-class errors.

```
{EXPERT_PREFIX} specialists use the following visual attributes to
distinguish between {TARGET_PLURAL}:
{attr_list}

You are given two {item} photographs ({ITEM} 1 and {ITEM} 2).

Please do the following:
```

```
1. **Describe {ITEM} 1**: For each of the attributes listed above,
   describe what you observe in {ITEM} 1. Use natural, concise
   language (e.g. "{EXAMPLE_DESCRIPTION}").

2. **Describe {ITEM} 2**: Do the same for {ITEM} 2.

3. **Key Differences**: Summarize the most important visual
   differences between the two {item_plural}. Focus on the
   attributes that would help a specialist tell them apart.

4. **{VERDICT_LABEL} Prediction**: Based on your observations, are
   these two {item_plural} the {VERDICT_QUESTION}?

Respond in JSON with the following structure:
{
  "{item}_1": {
    "<attr_1>": "...",
    "<attr_2>": "...",
    ... (one entry per attribute)
  },
  "{item}_2": { ... same attribute keys ... },
  "differences": "key visual differences between the two {item_plural}",
  "confidence": "high", "medium", or "low",
  "reasoning": "one-sentence justification based on the attributes",
  "{VERDICT_KEY}": "yes" or "no"
}
```

Table 9: **Per-dataset parameterisation of the comparison prompt template.** Substituting each column's values into the placeholders of the template above yields the exact prompt used for that dataset. CUB-200-2011 and iNaturalist Aves share a single prompt (both are bird benchmarks). The verdict key is the JSON Boolean field whose value is parsed into the binary GRPO reward.

Placeholder	CUB-200-2011 / iNat-Aves	Cars-196	FGVC-Aircraft
`{EXPERT_NOUN}`	ornithologist	automotive expert	aviation expert
`{WHAT_TO_IDENTIFY}`	bird species	vehicle make, model, and year	aircraft make, model, and variant
`{EXPERT_PREFIX}`	Bird	Automotive	Aviation
`{TARGET_PLURAL}`	species	vehicle models	aircraft variants
`{ITEM} / {item}`	Bird / bird	Car / car	Aircraft / aircraft
`{item_plural}`	birds	cars	aircraft
`{EXAMPLE_DESCRIPTION}`	short, cone-shaped, dark grey	four-door sedan, mid-size, chrome horizontal-slat grille	twin turbofan, under-wing, swept low wing, t-tail
`{VERDICT_LABEL}`	Same Species	Same Model	Same Variant
`{VERDICT_QUESTION}`	same species	same make and model	same make, model, and variant
`{VERDICT_KEY}`	`same_species`	`same_model`	`same_variant`
# attributes	28	17	17

E.2 Per-Dataset Attribute Vocabularies

The supervisor MLLM is asked to describe each input image along a fixed list of visual attribute groups before producing its same/different verdict. The list of attribute groups is dataset-specific and was chosen to capture the visual cues that domain specialists actually use to discriminate at the relevant taxonomic level: species for the two bird benchmarks, make and model for Cars-196, and variant (*e.g.* Boeing 737-700 vs. 737-800) for FGVC-Aircraft. The full per-dataset vocabularies are listed below; together with the prompt template above they fully specify the input passed to the supervisor.

CUB-200-2011 [30] and iNaturalist Aves [29]. For both bird benchmarks we use the same 28-attribute vocabulary, derived by collapsing CUB's 312 binary attributes into 28 groups: *bill shape, bill length, bill color, head pattern, crown color, forehead color, eye color, nape color, throat color, breast color, breast pattern, belly color, belly pattern, back color, back pattern, upperparts color, wing color, wing pattern, wing shape, tail shape, tail pattern, upper tail color, under tail color, underparts color, shape, size, primary color, leg color*.

Cars-196 [15]. 17 attribute groups, chosen to discriminate at the make-and-model level: *body style, number of doors, front grille, headlights, front bumper, side profile, roofline, greenhouse, wheels,*

fenders and wheel arches, rear lights, rear bumper, exhaust, badging, overall proportions, apparent era, paint finish.

FGVC-Aircraft [18]. 17 attribute groups, chosen to disambiguate variants (e.g. Boeing 737-700 vs. 737-800), not just manufacturers or families: *number of engines, engine type (turbofan / turboprop / piston / jet), engine mount position (under-wing / rear-fuselage / tail / in-wing), engine nacelle shape and size, wing configuration (high / mid / low mounted), wing planform (swept / straight / delta / variable), winglets or wingtip shape, tail configuration (conventional / t-tail / cruciform / v-tail), vertical stabilizer shape, fuselage length and proportions, nose shape, cockpit window layout, cabin window count and spacing, landing gear layout, overall size class (light / regional / narrow-body / wide-body), apparent era, livery and markings*.

E.3 Generic Comparison Prompt

The prompt-sensitivity ablation in Sec. 4.3 replaces the structured prompt above with the generic variant below: the expert role and the attribute list are dropped, and the model is asked only for a free-form difference description and the same/different verdict. The verdict key is preserved, so the GRPO reward parser operates without modification. We use the same placeholder convention as the structured template above; per-dataset substitutions are read from Table 9.

```
You are given two {item} photographs ({ITEM} 1 and {ITEM} 2).

Describe the most important visual differences between the
two {item_plural}, then decide whether they are the
{VERDICT_QUESTION}.

Respond in JSON with the following structure:
{
  "differences": "key visual differences between the two {item_plural}",
  "reasoning": "one-sentence justification",
  "{VERDICT_KEY}": "yes" or "no"
}
```

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 enumerate three explicit contributions (frozen-MLLM supervisor+ GRPO objective, KL attention alignment, and empirical gains on four DML benchmarks); each is supported by the corresponding method subsection (§3.2–§3.4) and experimental result (Table 1, §4.3).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 5 discusses limitations.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper is primarily empirical and does not include formal theorems or proofs. All mathematical content consists of definitional equations for the loss functions and training objective, which are derived directly from the GRPO formulation [27].

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Section 4.1 and Appendix B specify the backbone, pooler, optimizer, learning rate, schedule, GRPO group size, loss weights, batch size, pair sampling strategy and all other hyperparametrs. Algorithm 1 (supplement) provides per-step pseudocode, and Appendix E/E.2 contains the full prompt templates used for each dataset.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Code is not released with the submission. Section 4.1, Appendix B, and Algorithm 1 specify all hyperparameters and give full training pseudocode, and the complete prompt templates and attribute vocabularies are provided in Appendix E.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4.1 (Implementation details) and Appendices A & B provide data splits (standard zero-shot DML protocol: first half of classes for training, second half for testing), optimizer (AdamW), learning rates, scheduler, GRPO group size, loss weight selection, and hyperparameter sensitivity ranges.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report three-run std deviations for our main experiments in Section 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See Section 4.1 (Setup) and Appendix B (Additional Implementation Details).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics. The paper uses only publicly released datasets and pre-trained models under their permitted licenses, involves no human subjects, and preserves author anonymity in this submission.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

Justification: This paper is foundational research on training algorithms for image-retrieval representations. It does not introduce a deployed system, and any downstream impact follows from image-retrieval applications in general rather than from properties specific to our method.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release new pre-trained models, generative systems, or scraped datasets. Our contribution is a training algorithm; any released artifacts will be checkpoints of image-retrieval embedding models with low misuse potential.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All datasets (CUB-200-2011, Cars-196, FGVC-Aircraft, iNat-Aves) and the Qwen3-VL-8B backbone are cited using their original papers.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The submission does not release new assets (no new datasets or pre-trained models are released at submission time). If model checkpoints are released after publication, they will be accompanied by standard model-card documentation.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects. All data comes from existing public image-retrieval benchmarks.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve human subjects, so no IRB approval is required.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: A frozen multimodal LLM (Qwen3-VL-8B) is a central, non-standard component of our method.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.