

The Value of Being Wrong: Self-Mined Visual In-Context Learning from Errors

Shu Zhao Tan Yu
NVIDIA

shuzhao.cs@gmail.com

Abstract

Visual in-context learning (ICL) enables vision-language models (VLMs) to adapt at test time via demonstration examples, but its effectiveness depends critically on the examples selected. Recent counterfactual selection methods improve ICL by constructing demonstrations that expose how individual attribute changes affect the answer. However, they require multi-stage pipelines involving attribute extraction, caption engineering, and composed image matching. We propose a simpler alternative grounded in a key observation: *a model’s own prediction errors are natural counterfactual demonstrations that directly encode its failure modes*. Our method, SMILE (Self-Mined Visual In-Context Learning from Errors), builds a model-specific pool of hard negatives offline by recording the target VLM’s own mistakes. At test time, the prediction serves as a direction signal that surfaces, in the semantic embedding space, the past mistakes most likely to recur for the current query. This prediction-conditioned identification provides a unified mechanism for both closed-set classification and open-ended visual question answering without task-specific design. Across four benchmarks and four VLMs, SMILE delivers an average task-score gain of $+3.09$ points over the SOTA baseline at $5\times$ lower latency. Project page: <https://tree-shu-zhao.github.io/SMILE/>.

1 Introduction

In-context learning (ICL) has become the dominant paradigm for adapting vision-language models (VLMs) to new tasks at test time without parameter updates [1–5], unlike fine-tuning [6, 7], reinforcement learning [8], or training task-specific models [9–12]. However, the quality of the demonstration set matters as much as the model itself: poorly selected examples can reinforce spurious correlations [13], while carefully curated ones can expose the fine-grained distinctions that determine the correct answer [14–16]. This sensitivity has spurred active research into demonstration-selection strategies, from nearest-neighbor image matching [17] to multimodal heuristics [18–23].

Recently, CIRCLES [24] shows that counterfactual demonstration selection, assembling demonstrations that isolate how individual attribute changes affect the answer, helps VLMs reason about fine-grained visual distinctions. However, constructing each counterfactual requires the VLM to extract task-relevant attributes, generate faithful counterfactual captions, and execute composed image matching against the labeled corpus, a multi-stage pipeline with three potential points of failure.

This paper proposes a simpler path. Our key observation is that *a model’s own prediction errors already function as natural counterfactuals*. As illustrated in Figure 1, a hard negative retrieved from the model’s past errors can expose the same visual failure mode as the query, helping the VLM reject a plausible but wrong answer that positive demonstrations or attribute-based retrieval may not disambiguate. When a VLM predicts “brick” for a building image whose ground-truth answer is “concrete,” that example demonstrates that the visual evidence the model considers sufficient for one answer actually corresponds to a different one: precisely what a counterfactual demonstration should

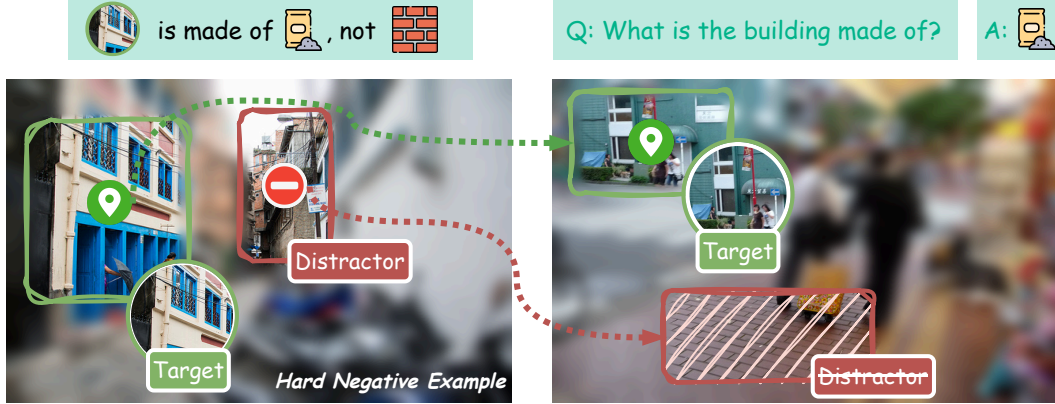


Figure 1: A VLM’s own past mistakes can steer attention away from visual distractors and onto the target region. Direct answering and positive-only retrieval focus on visually similar but misleading evidence, while attribute-based counterfactual retrieval fails to isolate the relevant material distinction. SMILE instead retrieves a hard negative with the same error pattern from the mined pool, making the distractor evidence explicit and steering the VLM toward the correct visual region.

convey. Crucially, these errors are not arbitrary: we show empirically (Section 3) that classification mistakes recur as exact confusion pairs, while VQA mistakes remain semantically close to the correct answer even when exact strings differ. This structure turns the model’s own zero-shot prediction into a useful cue for selecting past errors that resemble its current likely mistake.

Building on this insight, we propose SMILE (Self-Mined Visual In-Context Learning from Errors). An offline mining pass collects incorrectly predicted labeled examples as a reusable hard-negative pool. At test time, positive demonstrations are identified by image-embedding similarity, then the model’s zero-shot prediction guides identification of hard negatives whose stored mistaken predictions are semantically aligned in text-embedding space. Across four benchmarks and four VLMs, SMILE delivers an average task-score gain of +3.09 points over the SOTA baseline at $5\times$ lower latency.

Our contributions are:

- We reformulate counterfactual demonstration selection for visual ICL as model-specific hard-negative mining, a simple yet effective training-free framework.
- We provide empirical analysis showing that model errors are a reusable selection signal: exact confusions recur in closed-set classification, semantic error directions persist in open-ended VQA, and a pool decomposition explains why mined hard examples remain useful rather than noisy.
- We introduce prediction-conditioned selection via text-embedding similarity, a unified mechanism for both closed-set classification and open-ended VQA, and demonstrate consistent improvements across four benchmarks and four VLMs from two model families.

2 Related Work

Visual in-context learning. Vision-language models [1–3] can adapt to new tasks at test time by conditioning on a handful of image–text demonstrations [4, 1], but the resulting performance is highly sensitive to the demonstration set [18, 19, 25, 20, 13]. RICES picks nearest visual neighbors using CLIP features [17, 5, 26], and subsequent visual ICL work has refined this in several directions: Li et al. [14] configure in-context sequences for visual QA, Luo et al. [15] introduce the MUIER family that incorporates textual similarity, MMICES uses a two-stage multimodal selector [21], Chen et al. [27] explore exploration–exploitation strategies, Huang et al. [22], Zhou et al. [28], Yi-Ge et al. [23] learn or curate selection policies tailored to multimodal ICL, Chen et al. [29] apply coreset optimization, and Chen et al. [30] analyze the mechanisms behind visual ICL effectiveness. These approaches share a common thread: they optimize example selection to match the query. CIRCLES [24] departs from this paradigm with *counterfactual* demonstrations constructed via attribute extraction and composed image matching. Our method takes a fundamentally different approach: rather than *synthesizing* counterfactuals via attribute interventions, we leverage the target VLM’s own prediction errors as *natural, model-specific* counterfactuals with minimal pipeline overhead.

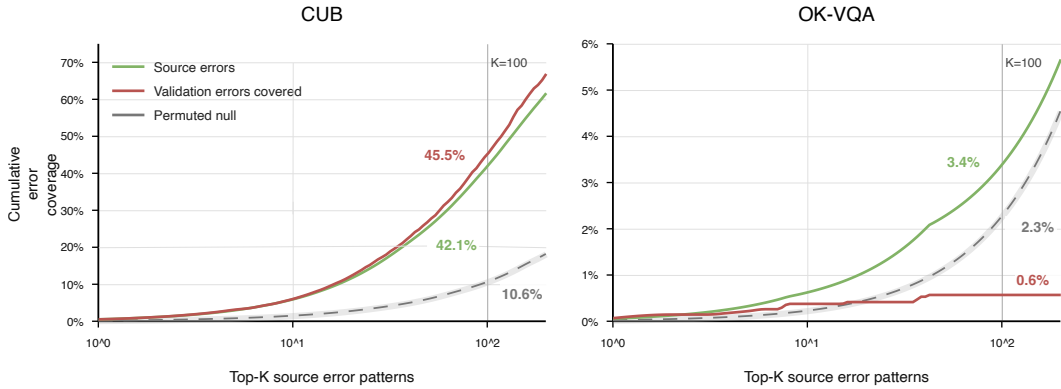


Figure 2: **Exact confusion-pair reuse is concentrated for classification but sparse for VQA.** **Green**: source-corpus exact error-pair distribution; **Red**: validation errors covered by the top- K most frequent source pairs; **Grey**: permuted-prediction null.

Hard negative mining. Hard negative mining has a long history in metric and contrastive representation learning [31–33]. In vision–language pretraining, hard negatives are constructed by perturbing captions [34], generating synthetic image–text pairs [35–37], or filtering false negatives in noisy web data [38–40]. A parallel line treats model errors as a useful learning signal: LLMs cannot reliably self-correct without external feedback [41], motivating critique routing [42] and online self-correction policies [43]; visual self-improvement methods construct preference pairs or reflection trajectories from VLM mistakes for fine-tuning [44, 45]; and a complementary line uses failures to drive curriculum design and data synthesis [46–48]. Theoretically, ICL itself behaves like error-driven learning, with surprising demonstrations exerting stronger effects than common ones [49]. All of these approaches use errors to drive parameter updates. We differ in two ways: hardness is defined by the target VLM’s own predictions, making the pool model-specific; and hard negatives are injected directly into the in-context prompt rather than used to drive parameter updates, surfacing targeted contradictions at inference time without any fine-tuning or external verifiers.

3 Model Errors Reveal Reusable Failure Modes

Our central assumption is that a wrong answer can still be informative. When a VLM predicts *fox sparrow* for an image of a *song sparrow*, or answers *circle* when the target answer is *round*, the prediction reveals the answer region the model considered plausible. If similar mistakes recur, then a labeled example the model previously answered incorrectly can serve as a useful warning at test time: it shows the model a case where its likely answer was tempting but wrong.

We test this premise through two questions. First, do VLM mistakes repeat as stable ground-truth/prediction confusions? Second, when exact answer strings do not repeat, do wrong predictions still point toward the right semantic region? These two cases match the two regimes our method must handle: closed-set classification and open-ended VQA.

Takeaway: Classification mistakes repeat exactly; VQA mistakes repeat semantically

Classification errors often repeat exactly: the same true class is confused with the same wrong class on unseen examples. VQA errors rarely repeat exactly, but wrong answers still land near the correct answer in semantic space.

Closed-set errors repeat as exact confusions. We first test the strongest form of reuse: whether the same ground-truth answer is repeatedly confused with the same wrong prediction. For each wrong zero-shot prediction, we record an error pair $(y, \hat{y})$, where y is the ground truth and $\hat{y}$ is the model prediction. On fine-grained classification, exact confusions are highly reusable: the top-100 source error pairs cover 45.5% of validation mistakes, and 94.2% of wrong validation predictions repeat a pair seen in the source corpus (Figure 2). This is the cleanest case for hard negatives: the model repeatedly falls into the same decision-boundary traps.

Open-ended VQA needs semantic rather than exact reuse. In VQA, exact strings fragment across synonyms, paraphrases, and partially specified answers. Only 3.0% of wrong validation predictions repeat an exact source pair, so exact pair identity is too brittle. However, this does not mean the errors are arbitrary. The relevant question is whether a wrong prediction still lands near the correct answer

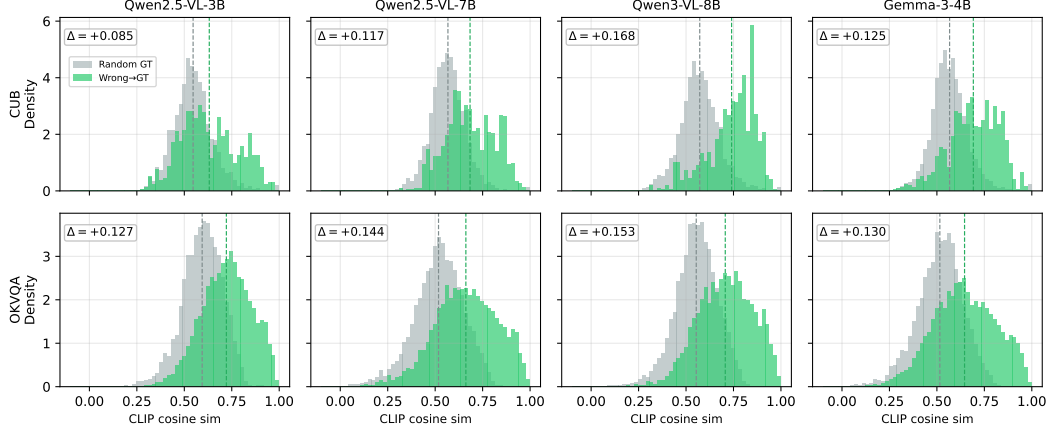


Figure 3: **Wrong predictions remain semantically closer to the ground truth than random answers.** Rows are task regimes; columns span model size/generation/family. Every cell shows the wrong-prediction distribution shifted to the right of the random-GT baseline, with a positive gap Δ .

semantically. We test this by embedding incorrect predictions and their ground-truth answers, and comparing their cosine similarities against a random ground-truth baseline.

Wrong predictions tend to concentrate near the truth. Across all VLMs and both task regimes, wrong predictions are consistently closer to the ground truth than random answers (Figure 3). For the reference model, the mean gap is $\Delta=+0.117$ on classification and $\Delta=+0.144$ on VQA; across all models, every gap is positive. Qualitatively, the mistakes usually preserve the coarse answer type: bird species are confused with visually related species, and VQA answers often differ by synonym, granularity, or category membership. The model is wrong, but it is usually wrong in a meaningful neighborhood. Full counts and significance tests are in Section C.

Design implication. These results show that a mined hard negative is not just a failed example; it is a labeled example attached to a known likely mistake $\hat{y}_h$. At test time, the query’s zero-shot prediction $\hat{y}_q$ reveals the model’s current likely mistake. Matching $\hat{y}_q$ to stored mistakes therefore gives SMILE a direct way to select examples from the relevant failure mode: exact confusion matching is natural for classification, while semantic matching is necessary for VQA.

4 Method

4.1 Problem Setup

Let $\mathcal{K} = \{(I_j, Q_j, y_j)\}_{j=1}^M$ denote a labeled corpus of images, questions, and answers, and let Φ denote a frozen target VLM. Given a test query (I_q, Q_q) , our goal is to construct an in-context demonstration set that both supports the correct answer and warns the model away from its most likely error. We use a frozen embedding model $E = (\text{Emb}_I, \text{Emb}_T)$, where Emb_I embeds images and Emb_T embeds text. The framework is agnostic to the particular embedding model; the only requirement is that E provide comparable image and text representations for similarity-based selection.

4.2 Framework Overview

Figure 4 gives an overview. SMILE decomposes the in-context demonstrations into two complementary branches:

$$\mathcal{P}_q = \text{TopK}_{x \in \mathcal{K}} \cos(\text{Emb}_I(I_q), \text{Emb}_I(I_x)), \quad (1)$$

$$\mathcal{N}_q = \text{TopN}_{h \in \mathcal{H}} \cos(\text{Emb}_T(\hat{y}_q), \text{Emb}_T(\hat{y}_h)), \quad (2)$$

where $\mathcal{P}_q$ is a set of K positive demonstrations identified by image-embedding similarity, $\mathcal{H}$ is a mined hard-negative pool (Section 4.3), $\hat{y}_q$ is the model’s zero-shot prediction for the query, and $\hat{y}_h$ is the mistaken prediction stored with each hard negative. The final prompt concatenates $\mathcal{P}_q$ and

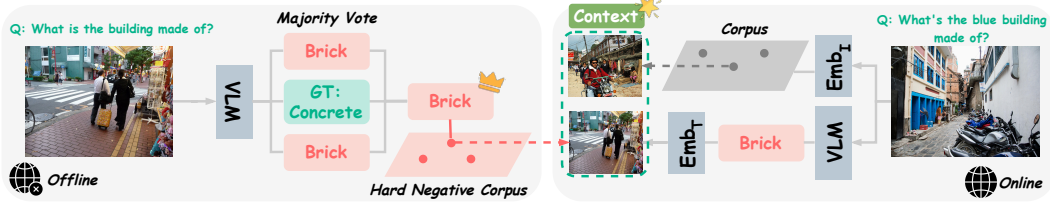


Figure 4: **SMILE mines failures offline and reuses them online as counterfactual demonstrations.** Left: the target VLM is run on the labeled corpus; examples whose voted prediction disagrees with the ground truth are stored with both the gold answer and the mistaken answer. Right: for a new query, image similarity retrieves positive demonstrations, while the query’s zero-shot prediction is matched in text-embedding space to stored mistaken predictions; the resulting hard negatives are appended to the context before final inference.

$\mathcal{N}_q$ before the query, with the hard-negative block labeled as “tricky examples that were previously confused with the predicted answer”; any hard negative already in the positive set is excluded.

4.3 Offline Hard-Negative Mining

The hard-negative pool $\mathcal{H}$ is constructed once per target model. For every labeled example (I_j, Q_j, y_j) , we run zero-shot inference with self-consistency voting [50, 51]:

$$\hat{y}_j = \text{vote}(\Phi(I_j, Q_j)^{(1)}, \dots, \Phi(I_j, Q_j)^{(R)}). \quad (3)$$

If $\hat{y}_j \neq y_j$, we add the example to the pool:

$$\mathcal{H} = \{(I_j, Q_j, y_j, \hat{y}_j) \mid \hat{y}_j \neq y_j\}. \quad (4)$$

Each mined example records both the ground-truth answer y_j and the model’s wrong prediction $\hat{y}_j$. When shown in context, a hard negative’s ground-truth answer contradicts the model’s likely prediction, which is the core function of a counterfactual demonstration. Mining is model-specific by design: different VLMs have distinct failure profiles, and mining separately ensures the pool captures each model’s unique error patterns. We empirically validate this choice and characterize how far a pool mined for one VLM can be reused on another in Section 5.4.

4.4 Prediction-Conditioned Selection

At test time, we first identify K positive demonstrations with image-embedding similarity (Equation (1)), following the standard RICES retrieval setup [17, 5]. We then obtain a zero-shot prediction $\hat{y}_q = \Phi(I_q, Q_q)$ for the query and use it as a *direction indicator* for hard-negative selection.

The selection is performed in text-embedding space. We encode $\hat{y}_q$ and each mined $\hat{y}_h$ with Emb_T and rank hard negatives by cosine similarity (Equation (2)). The top N hard negatives, those whose mistaken predictions are most semantically similar to the query’s predicted answer, are selected as contrastive demonstrations.

Why semantic text matching? We use embedding similarity rather than exact matching or token F1 because open-ended VQA predictions exhibit high lexical variation (e.g., “park,” “public garden,” “green space” for the same error mode); see Table 5.

5 Experiments

Our experiments are organized around three research questions. **RQ1:** Does SMILE improve visual ICL across task regimes, model families, and metrics? **RQ2:** Are mined mistakes useful because they contain reusable task signals rather than simply adding more examples? **RQ3:** Which parts of prediction-conditioned hard-negative selection are necessary?

Table 1: **Main comparison with both primary metric and F1 reported per dataset.** The **best** and **second best** scores per model and metric are highlighted.

Model	Method	CUB		Flowers		OK-VQA		VizWiz		Average	
		Acc	F1	Acc	F1	EM	F1	EM	F1	Score	F1
Gemma3-4B	None	10.56	8.19	46.76	46.50	19.12	26.70	39.85	57.26	29.07	34.66
	Random	9.98	8.10	37.01	39.51	24.81	30.74	51.84	67.08	30.91	36.36
	RICES	65.40	67.62	86.70	87.43	26.65	32.72	56.08	70.40	58.71	64.54
	MUIER	65.21	67.42	86.39	87.28	26.87	32.82	56.59	70.75	58.77	64.57
	MMICES	13.95	10.98	38.09	38.20	26.24	32.41	52.72	68.21	32.75	37.45
	CIRCLES	71.97	72.39	93.32	93.49	31.27	36.89	57.61	71.35	63.54	68.53
	SMILE (ours)	72.22	72.75	96.89	96.88	35.01	39.94	69.69	76.20	68.45	71.44
Gemma3-12B	None	30.34	25.02	67.51	64.33	25.47	33.89	56.82	71.26	45.04	48.62
	Random	29.27	25.82	71.28	71.56	33.59	39.56	70.13	80.54	51.07	54.37
	RICES	76.37	76.25	96.44	96.29	36.86	42.61	73.98	83.43	70.91	74.65
	MUIER	76.51	76.39	96.42	96.28	36.58	42.30	73.72	83.04	70.81	74.50
	MMICES	36.37	31.61	71.36	69.13	35.12	40.85	71.50	81.56	53.59	55.79
	CIRCLES	77.03	76.90	97.77	97.75	37.75	43.23	74.30	82.35	71.71	75.06
	SMILE (ours)	76.98	76.89	98.85	98.86	43.67	48.29	78.58	84.40	74.52	77.11
Qwen2.5-VL-3B	None	7.09	5.97	22.57	24.04	42.29	46.12	75.83	78.58	36.95	38.68
	Random	15.81	15.63	41.76	44.57	41.54	45.30	74.48	77.40	43.40	45.73
	RICES	72.26	73.88	93.06	92.84	42.57	46.26	70.80	73.37	69.67	71.59
	MUIER	72.25	73.87	93.06	92.83	41.66	45.82	71.61	74.74	69.64	71.81
	MMICES	17.26	15.38	33.84	35.47	39.32	42.95	73.23	76.21	40.91	42.50
	CIRCLES	74.89	76.34	94.70	94.77	43.24	47.27	72.93	75.33	71.44	73.43
	SMILE (ours)	76.75	77.75	94.79	95.25	43.75	47.53	78.26	82.72	73.39	75.81
Qwen2.5-VL-7B	None	14.83	11.79	43.47	42.14	33.31	38.33	68.70	73.30	40.08	41.39
	Random	26.27	24.42	47.15	49.74	41.66	46.20	76.11	78.34	47.80	49.67
	RICES	82.15	81.91	98.83	98.85	43.66	48.32	73.79	76.13	74.61	76.30
	MUIER	82.14	81.90	98.93	98.95	44.29	48.76	74.67	77.08	75.01	76.67
	MMICES	34.57	29.72	49.15	47.62	43.40	47.66	70.69	72.77	49.45	49.44
	CIRCLES	82.17	82.13	98.99	99.04	43.54	48.53	77.63	82.36	75.58	78.02
	SMILE (ours)	82.55	82.30	99.33	99.37	47.80	51.61	83.42	87.49	78.28	80.19

5.1 Experimental Setup

Datasets. We evaluate on four standard visual ICL benchmarks: **CUB-200-2011** [52] (fine-grained bird classification), **Flowers-102** [53] (flower classification), **OK-VQA** [54] (knowledge-based VQA), and **VizWiz** [55] (real-world VQA, ~32% unanswerable). Classification reports accuracy / weighted F1; VQA reports exact match / token F1.

Models. We evaluate four open VLMs from two families: Gemma-3 [56] (4B, 12B) and Qwen2.5-VL [57] (3B, 7B). Additional results on Qwen3-VL [58] (4B, 8B) are reported in Section E. Unless otherwise specified, all analyses, ablations, and probes use Qwen2.5-VL-7B as the default backbone.

Baselines. We compare against **None** (zero-shot), **Random**, **RICES** [5], **MUIER** [15], **MMICES** [21], and **CIRCLES** [24]. Following Xiong et al. [24], all baselines use a total budget of 32 in-context examples, and we directly report the results from CIRCLES.

Implementation. Hard-negative pools are mined offline once per target VLM with self-consistency voting. SMILE uses $K+N=12$ in-context examples for classification ($K=8$, $N=4$) and 20 for VQA ($K=12$, $N=8$). All experiments report means over 3 runs; full setup, per-model settings, and dataset statistics are in Section A.

5.2 Main Results

Table 1 presents the main comparison of SMILE against six in-context learning baselines across four datasets and four VLM backbones, reporting both the primary metric (accuracy for classification; exact match for VQA) and the corresponding F1 score. From the results, we observe that:

SMILE consistently outperforms all baselines. It achieves the highest average task score and average F1 for every backbone, winning against the strongest counterfactual baseline CIRCLES with an average task-score gain of +3.09 points, and improving over the strongest similarity-only

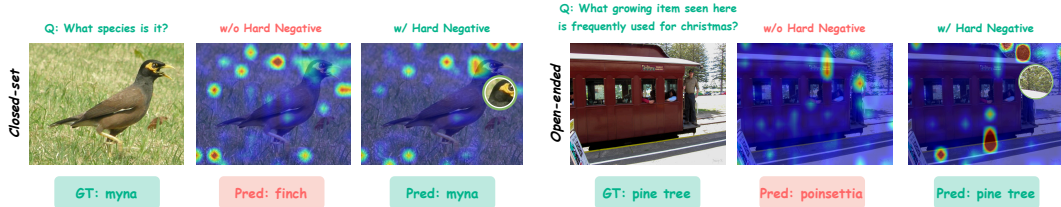


Figure 5: **Hard negatives concentrate attention on the discriminating region and flip the prediction.** Attention from the final prompt token to query-image patches, with vs. without the prediction-conditioned hard-negative block.

baseline (RICES or MUIER) by +3.3 to +9.7 points. Notably, SMILE achieves this with only 12–20 in-context examples while all baselines use 32, so the gain is not a function of longer context.

Gains are largest on open-ended VQA. SMILE delivers its biggest improvements over CIRCLES on VQA benchmarks, where plausible wrong answers are diverse; classification gains are smaller because visual neighbors already provide a strong label signal, and accuracy is closer to its ceiling.

The effect is consistent across model scales and families. SMILE improves over the SOTA method by +4.91, +2.81, +1.95, and +2.70 average task-score points on Gemma and Qwen models with different sizes. Additional results on Qwen3-VL are reported in Section E.

5.3 Why Mined Errors Help

The main results show that mined errors improve performance; we now ask why. To probe the mechanism, we (a) partition the CUB labeled corpus by the model’s own behavior into a *clean pool* (1,113 images it classifies correctly under self-consistency voting) and a *hard pool* (4,881 images it gets wrong), then re-run RICES retrieval restricted to each pool (Table 2); and (b) extract attention from the final prompt token to the query-image patch grid, averaged over heads at late decoder layers, with vs. without the hard-negative block (Figure 5). From these analyses, we observe that:

Hard examples are load-bearing precisely on the classes the model gets wrong. Restricting RICES to the smaller clean pool collapses overall EM from 82.15 to 46.12, yet restricting it to the much larger hard pool costs only 1.81 points. The asymmetry comes from a coverage gap: 77 of CUB’s 200 species have zero clean examples because the model misclassified every image of those classes, and on the 39% of queries drawn from these classes, hard-only retrieval beats clean-only by +68.34 EM, since clean retrievals are structurally off-class. On the fair slice where both pools cover the true class, clean-only beats hard-only by +9.15 EM, confirming that hard examples are not intrinsically better, but are the only available positives precisely where the model needs them. See Section F for full slice definitions.

Table 2: **Pool decomposition** on CUB. *Uncovered*: the query’s true class has no clean examples. *Fair*: both pools contain ≥ 4 examples of the true class.

	Full	Uncovered	Fair
Full corpus	82.15	76.27	89.77
Hard only	80.34	80.00	83.27
Clean only	46.12	11.66	92.42

Visual similarity implicitly performs hard-example coverage. On average 9.74 of the top-12 RICES retrievals are drawn from the hard pool, because the visually most-confusable images are exactly the ones the model misclassifies; the very property that makes “hard” examples seem risky as demonstrations is what makes them naturally retrievable. This motivates SMILE’s two-axis design: positives are retrieved from the full labeled corpus, while the hard-negative branch operates on semantic similarity between predicted answers $\hat{y}_h$ and $\hat{y}_q$, so the two branches provide complementary signals rather than duplicating each other.

Hard negatives redirect attention to the discriminating region and flip the prediction. With the positive set fixed and only the hard-negative block toggled, attention sharpens away from distractor regions and onto the feature that separates the correct answer from the model’s plausible mistake. The closed-set and open-ended examples are consistent with the mechanism in Section 4.4: a hard negative tied to the model’s likely mistake forces it to seek query-image evidence that distinguishes the right answer from that mistake.

Table 3: **Cross-model hard-negative pool transfer heatmap.** Each cell reports the change in CUB accuracy or OK-VQA EM relative to the self-mined pool for the same inference backbone and dataset.

HN-pool source	CUB					OK-VQA				
	Q2.5-3B	Q2.5-7B	Q3-8B	G-4B	G-12B	Q2.5-3B	Q2.5-7B	Q3-8B	G-4B	G-12B
Qwen2.5-VL-3B	+0.00	+0.77	-0.23	+1.69	-1.20	+0.00	+0.24	-1.47	-1.98	-0.71
Qwen2.5-VL-7B	+0.27	+0.00	-0.12	-0.05	-0.65	-0.38	+0.00	-1.19	-1.89	-0.79
Qwen3-VL-8B	+0.26	+0.38	+0.00	+0.67	-0.58	-0.12	+0.29	+0.00	-0.50	+0.52
Gemma-3-4B	+0.30	+1.18	-0.09	+0.00	-0.02	-0.43	-0.36	-1.20	+0.00	-0.67
Gemma-3-12B	+0.53	+0.01	+0.32	+0.31	+0.00	-0.30	+0.04	-0.71	-1.55	+0.00

5.4 Cross-Model Hard-Negative Pool Transfer

SMILE mines a model-specific hard-negative pool, but deployment settings may involve multiple VLMs or frequent model upgrades. We therefore ask whether another model can use a pool mined by one model at inference time. Table 3 reports the change in performance when replacing each inference model’s self-mined pool with a pool mined by another backbone, across five backbones (Qwen2.5-VL-3B/7B, Qwen3-VL-8B, Gemma-3-4B/12B).

Transfer works robustly across architecture families. This supports a practical “mine once, reuse broadly” interpretation: the stored wrong predictions $\hat{y}_h$ encode semantic neighborhoods of plausible mistakes that are partly shared across model families. The heatmap also reveals the boundary of transfer, so a self-mined pool remains the safest choice when maximum accuracy is required. However, when mining cost or deployment simplicity matters, an existing pool from a nearby or readily available VLM is a viable substitute.

5.5 Ablations

Effect of hard negatives. Table 4 separates the positive and contrastive branches. Positive demonstrations alone provide the largest single jump, especially on CUB (14.83→81.18 Acc), confirming that nearest visual neighbors remain the backbone of visual ICL. Hard negatives then add further gains. On OK-VQA, prediction-conditioned negatives outperform random negatives (47.80 vs. 42.86 EM), showing the value of semantically aligned failures in open-ended answer spaces; on CUB, random negatives remain competitive because broader label coverage is enough in this closed-set setting.

Table 4: Ablation on **component**.

Pos.	Rand. HN	Pred. HN	CUB	OK-VQA
✗	✗	✗	14.83	33.31
✓	✗	✗	81.18	43.16
✓	✓	✗	81.23	42.86
✓	✗	✓	82.55	47.80

Selection strategy. A central design choice in SMILE is how to match the query’s zero-shot prediction to the mined pool. Table 5 compares three candidates (exact string match, token-F1 overlap, and CLIP text similarity) on CUB and OK-VQA. On the closed-set classification task, all three matchers perform comparably (within 0.25 Acc) because bird species labels have limited lexical variation. On open-ended OK-VQA, exact-match collapses (40.46 EM) while semantic matching dominates (47.80, +7.3 EM): open-ended predictions fragment across synonyms and paraphrases of the same error mode, and only embedding similarity reliably detects them. Text-embedding similarity, therefore, provides a single mechanism that serves both regimes.

Table 5: Ablation on **hard selection**.

	CUB	OK-VQA	Avg.
EM	82.34	40.46	61.40
F1	82.39	46.65	64.52
Semantic	82.55	47.80	65.18

5.6 Hyperparameter Sensitivity

Figure 6 sweeps the positive (K) and negative (N) demonstration counts on CUB and OK-VQA. The two regimes prefer different optima: CUB peaks sharply at $K=8$, $N=4$, while OK-VQA prefers larger K paired with more negatives ($K=12$, $N \geq 8$) and shows a smooth plateau where any ($K \geq 8$, $N \geq 8$) combination lies within 0.3 EM of the best cell. Closed-set classification needs precise

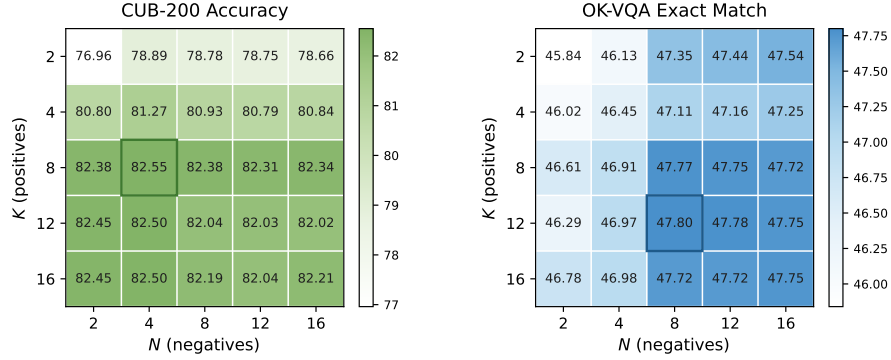


Figure 6: **Hyperparameter sweep** over positive (K) and negative (N) demonstration counts.

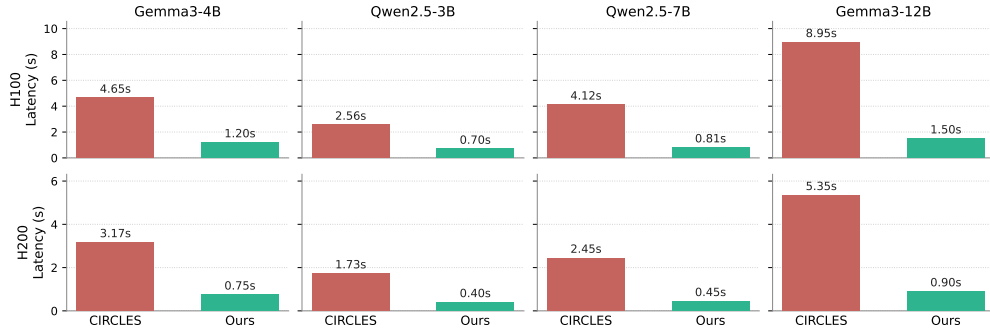


Figure 7: **Per-query inference latency** across four VLMs. SMILE is 3.6–6.0 $\times$ faster than CIRCLES on H100 (top) and 4.2–6.0 $\times$ faster on H200 (bottom).

visual exemplars with a few targeted contradictions; open-ended VQA benefits from broader context and more diverse contrastive signals. Performance also saturates and eventually degrades when $K+N$ exceeds the model’s effective context.

5.7 Inference-Time Cost and Latency

SMILE separates one-time mining cost from per-query inference cost. Offline mining runs the target VLM over the labeled corpus once; on our largest dataset (VizWiz, $\sim 20K$ examples) this completes in 6.7–13.3 minutes on 8 H100 GPUs depending on backbone size (full breakdown in Table 6 of Section B), and the cross-model transfer above means a previously mined pool can often be reused for a new VLM. At inference time, SMILE adds only a short zero-shot direction call and a nearest-neighbor lookup over pre-encoded predictions ($< 5\%$ overhead); end-to-end latency is dominated by VLM generation. Figure 7 compares both methods inside a shared vLLM pipeline [24] on H100 and H200: SMILE runs 3.6–6.0 $\times$ faster than CIRCLES on H100 and 4.2–6.0 $\times$ on H200, with absolute per-query cost below 1 s on every tested backbone on H200 (0.40–0.90 s) versus 1.7–5.3 s for CIRCLES. SMILE dominates CIRCLES on the accuracy–latency frontier.

6 Conclusion

We presented SMILE, a visual in-context learning framework that replaces attribute-engineered counterfactual selection with model-specific hard-negative mining. By collecting labeled examples that the target VLM incorrectly predicts and selecting a subset whose mistaken predictions are semantically aligned with the query’s likely error, SMILE surfaces targeted contradictions that challenge the model’s failure modes without requiring attribute extraction, caption generation, or composed image matching. On four benchmarks and four VLMs, SMILE consistently outperforms all compared baselines, including the state-of-the-art counterfactual selection method CIRCLES, with the strongest gains on open-ended VQA where model failure modes are most diverse.

References

- [1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a visual language model for few-shot learning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022.
- [2] Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning (ICML)*, 2023.
- [3] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023.
- [4] Maria Tsimpoukelli, Jacob Menick, Serkan Cabi, S. M. Ali Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.
- [5] Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, Jenia Jitsev, Simon Kornblith, Pang Wei Koh, Gabriel Ilharco, Mitchell Wortsman, and Ludwig Schmidt. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*, 2023.
- [6] Shu Zhao, Nilesh Ahuja, Tan Yu, Tianyi Shen, and Vijaykrishnan Narayanan. MoRA: Missing modality low-rank adaptation for visual recognition. In *International Conference on Learning Representations (ICLR)*, 2026.
- [7] Shu Zhao, Tan Yu, Xiaoshuai Hao, Wenchao Ma, and Vijaykrishnan Narayanan. KALAHASH: Knowledge-anchored low-resource adaptation for deep hashing. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2025.
- [8] Shu Zhao, Tan Yu, Anbang Xu, Japinder Singh, Aaditya Shukla, and Rama Akkiraju. ParallelSearch: Train your LLMs to decompose query and search sub-queries in parallel with reinforcement learning. *arXiv preprint arXiv:2508.09303*, 2025.
- [9] Shu Zhao, Dayan Wu, Wanqian Zhang, Yu Zhou, Bo Li, and Weiping Wang. Asymmetric deep hashing for efficient hash code compression. In *ACM International Conference on Multimedia (ACM MM)*, 2020.
- [10] Shu Zhao, Dayan Wu, Yucan Zhou, Bo Li, and Weiping Wang. Rescuing deep hashing from dead bits problem. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.
- [11] Xiaoshuai Hao, Yunfeng Diao, Mengchuan Wei, Yifan Yang, Peng Hao, Rong Yin, Hui Zhang, Weiming Li, Shu Zhao, and Yu Liu. MapFusion: A novel BEV feature fusion network for multi-modal map construction. *Information Fusion*, 119:103018, 2025.
- [12] Xiaoshuai Hao, Lingdong Kong, Rong Yin, Pengwei Wang, Jing Zhang, Yunfeng Diao, and Shu Zhao. SafeMap: Robust HD map construction from incomplete observations. In *International Conference on Machine Learning (ICML)*, 2025.
- [13] Hrayr Harutyunyan, Rafayel Darbinyan, Samvel Karapetyan, and Hrant Khachatryan. In-context learning in presence of spurious correlations. *Transactions on Machine Learning Research (TMLR)*, 2026.
- [14] Li Li, Jiawei Peng, Huiyi Chen, Chongyang Gao, and Xu Yang. How to configure good in-context sequence for visual question answering. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2024.

- [15] Yang Luo, Zangwei Zheng, Zirui Zhu, and Yang You. How does the textual information affect the retrieval of multimodal in-context learning? In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024.
- [16] Jeonghwan Kim and Heng Ji. Finer: Investigating and enhancing fine-grained visual concept recognition in large vision language models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024.
- [17] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. An empirical study of GPT-3 for few-shot knowledge-based VQA. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2022.
- [18] Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for gpt-3? In *Annual Meeting of the Association for Computational Linguistics Workshops (ACL Workshop)*, 2022.
- [19] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022.
- [20] Ivana Balazevic, David Steiner, Nikhil Parthasarathy, Relja Arandjelovic, and Olivier J. Hénaff. Towards in-context scene understanding. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023.
- [21] Shuo Chen, Zhen Han, Bailan He, Jianzhe Liu, Mark Buckley, Yao Qin, Philip Torr, Volker Tresp, and Jindong Gu. Can multimodal large language models truly perform multimodal in-context learning? In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025.
- [22] Brandon Huang, Chancharik Mitra, Leonid Karlinsky, Assaf Arbelle, Trevor Darrell, and Roei Herzig. Multimodal task vectors enable many-shot multimodal in-context learning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2024.
- [23] Ellen Yi-Ge, Jiechao Gao, Wei Han, and Wei Zhu. DRUM: learning demonstration retriever for large multi-modal models. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025.
- [24] Guangzhi Xiong, Sanchit Sinha, Zhenghao He, and Aidong Zhang. Retrieving counterfactuals improves visual in-context learning. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2026.
- [25] Ohad Rubin, Jonathan Herzig, and Jonathan Berant. Learning to retrieve prompts for in-context learning. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2022.
- [26] 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 *International Conference on Machine Learning (ICML)*, 2021.
- [27] Cheng Chen, Yunpeng Zhai, Yifan Zhao, Jinyang Gao, Bolin Ding, and Jia Li. Provoking multi-modal few-shot LVLm via exploration-exploitation in-context learning. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2025.
- [28] Yucheng Zhou, Xiang Li, Qianning Wang, and Jianbing Shen. Visual in-context learning for large vision-language models. In *Findings of the Association for Computational Linguistics (ACL Findings)*, 2024.
- [29] Huiyi Chen, Jiawei Peng, Kaihua Tang, Xin Geng, and Xu Yang. Enhancing multimodal in-context learning for image classification through coreset optimization. In *ACM International Conference on Multimedia (ACM MM)*, 2025.

- [30] Shuo Chen, Zhen Han, Bailan He, Mark Buckley, Philip Torr, Volker Tresp, and Jindong Gu. Understanding and improving in-context learning on vision-language models. In *International Conference on Learning Representations Workshops (ICLRW)*, 2024.
- [31] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2015.
- [32] Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *International Conference on Learning Representations (ICLR)*, 2021.
- [33] Albert Xu, Jhih-Yi Hsieh, Bhaskar Vundurthy, Nithya Kemp, Eliana Cohen, Lu Li, and Howie Choset. Mathematical justification of hard negative mining via isometric approximation theorem. In *International Conference on Learning Representations (ICLR)*, 2024.
- [34] Le Zhang, Rabiul Awal, and Aishwarya Agrawal. Contrasting intra-modal and ranking cross-modal hard negatives to enhance visio-linguistic compositional understanding. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2024.
- [35] Maitreya Patel, Abhiram Kusumba, Sheng Cheng, Changhoon Kim, Tejas Gokhale, Chitta Baral, and Yezhou Yang. Tripletclip: Improving compositional reasoning of CLIP via synthetic vision-language negatives. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2024.
- [36] Chengen Lai, Shengli Song, Sitong Yan, and Guangneng Hu. Improving vision and language concepts understanding with multimodal counterfactual samples. In *European Conference on Computer Vision (ECCV)*, 2024.
- [37] Amita Kamath, Cheng-Yu Hsieh, Kai-Wei Chang, and Ranjay Krishna. The hard positive truth about vision-language compositionality. In *European Conference on Computer Vision (ECCV)*, 2024.
- [38] Jaeseok Byun, Dohoon Kim, and Taesup Moon. MAFA: managing false negatives for vision-language pre-training. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2024.
- [39] Jaehyun Kwak, Ramahdani Muhammad Izaaz Inhar, Se-Young Yun, and Sung-Ju Lee. Qure: Query-relevant retrieval through hard negative sampling in composed image retrieval. In *International Conference on Machine Learning (ICML)*, 2025.
- [40] Haonan Wang, Minbin Huang, Runhui Huang, Lanqing Hong, Hang Xu, Tianyang Hu, Xiaodan Liang, Zhenguo Li, Hong Cheng, and Kenji Kawaguchi. Getting more juice out of your data: Hard pair refinement enhances visual-language models without extra data. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2025.
- [41] Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *International Conference on Learning Representations (ICLR)*, 2024.
- [42] Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: large language models can self-correct with tool-interactive critiquing. In *International Conference on Learning Representations (ICLR)*, 2024.
- [43] Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D. Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M. Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal M. P. Behbahani, and Aleksandra Faust. Training language models to self-correct via reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2025.
- [44] Yi Ding and Ruqi Zhang. Sherlock: Self-correcting reasoning in vision-language models. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

- [45] Kanzhi Cheng, Yantao Li, Fangzhi Xu, Jianbing Zhang, Hao Zhou, and Yang Liu. Vision-language models can self-improve reasoning via reflection. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2025.
- [46] Nayoung Lee, Ziyang Cai, Avi Schwarzschild, Kangwook Lee, and Dimitris Papailiopoulos. Self-improving transformers overcome easy-to-hard and length generalization challenges. In *International Conference on Machine Learning (ICML)*, 2025.
- [47] Xiao Liang, Zhong-Zhi Li, Yeyun Gong, Yang Wang, Hengyuan Zhang, yelong shen, Ying Nian Wu, and Weizhu Chen. Sws: Self-aware weakness-driven problem synthesis in reinforcement learning for LLM reasoning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2025.
- [48] Zhiyuan Zeng, Yizhong Wang, Hannaneh Hajishirzi, and Pang Wei Koh. Evaltree: Profiling language model weaknesses via hierarchical capability trees. In *Conference on Language Modeling (COLM)*, 2025.
- [49] Zhenghao Zhou, Robert Frank, and R. Thomas McCoy. Is in-context learning a type of error-driven learning? evidence from the inverse frequency effect in structural priming. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2025.
- [50] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2023.
- [51] Ying Mo, Jiahao Liu, Jian Yang, Qifan Wang, Shun Zhang, Jingang Wang, and Zhoujun Li. C-ICL: Contrastive in-context learning for information extraction. In *Findings of the Association for Computational Linguistics: EMNLP (EMNLP Findings)*, 2024.
- [52] 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.
- [53] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP)*, 2008.
- [54] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. OK-VQA: A visual question answering benchmark requiring external knowledge. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2019.
- [55] Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2018.
- [56] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Godeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Pettrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh,

Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

- [57] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.
- [58] 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, Sibao 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. *arXiv preprint arXiv:2511.21631*, 2025.

Appendix

A Implementation Details	15
B Offline Mining Time Breakdown	16
C Additional Error-Structure Analysis	16
D Effect of the Similarity Encoder	17
E Additional Backbone Results	17
F Pool Decomposition: Phase Transition with k	18
G Prompt Template	18
H Limitations	19

A Implementation Details

Datasets. **CUB-200-2011** [52] contains 5,994 labeled and 5,794 test images across 200 fine-grained bird species. **Flowers-102** [53] uses 2,040 labeled images and 6,149 test images across 102 flower categories. **OK-VQA** [54] contains 9,009 labeled and 5,046 validation questions requiring external knowledge. **VizWiz** [55] contains 20,523 labeled and 4,319 validation questions collected by blind photographers, with $\sim 32\%$ of questions being unanswerable.

Backbone VLMs. All baselines and SMILE are instantiated with the same backbone VLMs to ensure fair comparison. We use Gemma-3 [56] with 4B and 12B parameters, Qwen2.5-VL [57] with 3B and 7B parameters, and Qwen3-VL [58] with 4B and 8B parameters. The complete baseline matrix in Table 1 uses the four Gemma-3 and Qwen2.5-VL backbones; additional Qwen3-VL results appear in Section E. All models are used in their instruction-tuned variants with greedy decoding (temperature 0) for final answer generation and served via the vLLM library with BFloat16 precision.

Baselines. We compare against six in-context learning strategies. **None**: zero-shot, no demonstrations. **Random**: uniformly sampled demonstrations. **RICES** [5]: image-similarity selection. **MUIER** [15]: multimodal selection combining image and text similarity. **MMICES** [21]: two-stage multimodal selection. **CIRCLES** [24]: attribute-guided counterfactual selection via composed image matching. All baselines use CLIP ViT-G/14 as the shared similarity encoder with a total budget of 32 in-context examples.

Similarity encoder. For all similarity-based methods (RICES, MUIER, MMICES, CIRCLES, SMILE), we use CLIP ViT-G/14 (“laion/CLIP-ViT-g-14-laion2B-s12B-b42K”) [26] as the shared image-text encoder. All visual and textual features are L2-normalized before similarity computation.

Hard-negative mining setup. For each target VLM, we run zero-shot inference on the labeled corpus with self-consistency voting ($R=3$ generations per example, temperature 0.7, majority vote). Examples where the voted prediction does not match the ground-truth answer are collected into the model-specific hard-negative pool. Pool sizes range from $\sim 20\%$ of the labeled corpus (for strong 7B+ models on easy tasks) to $\sim 80\%$ (for small models on difficult tasks).

Prompting setup. We use a unified prompting template across all methods and tasks. The prompt includes the query image and question, followed by in-context demonstrations, and repeats the input at the end. All VLMs use greedy decoding (temperature 0) for the final answer generation. For classification tasks (CUB and Flowers), we enforce a closed answer set by appending the candidate class names. For VQA tasks (OK-VQA and VizWiz), the model generates answers freely. VizWiz questions may be unanswerable from the image: each question asks the model to answer

Table 6: **Offline mining time on 8 H100 GPUs.** Wall-clock minutes for one full $R=3$ -vote zero-shot pass over each dataset’s labeled corpus.

Dataset	# samples	Gemma-3-12B	Gemma-3-4B	Qwen2.5-VL-7B	Qwen2.5-VL-3B
CUB-200	5,994	3.3	1.9	2.4	1.6
Flowers-102	2,040	0.9	0.7	0.8	0.6
OK-VQA	9,009	5.1	3.2	3.5	2.5
VizWiz	20,523	13.3	6.7	11.3	6.8

“unanswerable” in that case, and hard negatives whose stored prediction $\hat{y}_h$ is “unanswerable” are excluded from the pool. On VizWiz, each in-context example is labeled with its most frequent annotator answer (ignoring answers marked as not confident) instead of the first annotator’s answer, or with “unanswerable” if the question is not marked answerable, and the final instruction is “Answer the question using a single word or phrase.” After generation, an answerability vote replaces the answer with “unanswerable” when at least half of the query’s $K+N$ in-context examples are labeled “unanswerable”.

Section 3 analysis setup. Unless otherwise specified, the analyses in Section 3 use Qwen2.5-VL-7B with one representative fine-grained classification dataset (CUB-200) and one representative VQA dataset (OK-VQA). For the confusion-pair analysis, morphological near-duplicates (*jean/jeans*, *park meter/parking meter*) are treated as correct so that the reported pairs focus on semantic errors. This normalization is only an analytical lens: the mining pipeline of Section 4.3 uses official VQA equality, and applying the same normalization to mining does not change downstream SMILE performance.

Compute resources. All experiments were conducted on NVIDIA H100 GPUs. Mining the hard-negative pool is fast: on 8 H100 GPUs with vLLM and self-consistency $R=3$, even VizWiz (the largest dataset, 20K examples) finishes in 6.7–13.3 minutes per backbone; full per-dataset, per-model timings are in Table 6 of Section B. Inference-time overhead is also minimal: one additional zero-shot VLM call per query (for the direction prediction) plus a text-embedding similarity lookup, which adds $< 5\%$ wall-clock time compared to RICES.

B Offline Mining Time Breakdown

Table 6 reports wall-clock mining times measured on 8 NVIDIA H100 GPUs with vLLM serving (BFloat16, default concurrency) and self-consistency $R=3$ generations per example. All entries are end-to-end measurements from this configuration. Mining is one-time per backbone and per dataset, and the cross-model transfer result (Section 5.4) means a previously mined pool can often be reused for a new VLM, further amortizing this cost.

Even on the largest corpus (VizWiz, 20,523 examples), mining completes in well under 15 minutes for every backbone. Smaller datasets like Flowers-102 are essentially free (under one minute). The implication is that SMILE’s offline cost is not a deployment bottleneck: it is incurred once per backbone-dataset pair and is fully amortized across all subsequent test queries.

C Additional Error-Structure Analysis

Exact confusion-pair reuse. For every wrong zero-shot prediction, we record the pair $(y, \hat{y})$ consisting of the ground-truth answer and the model prediction. The source corpus summarizes past mistakes, and the validation split tests whether those mistakes reappear. As a chance baseline, we randomly re-pair ground-truth answers with wrong predictions, which estimates the repetition expected if errors were unrelated to the true answer.

For CUB, the 100 most frequent source pairs cover 42.1% of 4,766 source error patterns, which is $4.0\times$ the shuffle baseline (10.6%). These pairs transfer strongly to validation: 94.2% of wrong validation predictions repeat some pair seen in the source corpus, and the top-100 source pairs cover 45.5% of validation mistakes. For OK-VQA, exact strings are much more fragmented: the top-100 source pairs cover 3.4% of 4,407 source error patterns, compared with a 2.3% shuffle baseline, and only 3.0% of wrong validation predictions repeat an exact source pair.

Table 7: **Effect of the similarity encoder on SMILE performance.** CUB-200 with Qwen2.5-VL-7B, swapping the encoder used for both positive retrieval and hard-negative selection while keeping all other settings fixed. CLIP ViT-G/14 is the paper baseline used in all other experiments.

Encoder	EM	F1
CLIP ViT-B/32	70.85	70.50
CLIP ViT-B/16	74.63	74.53
CLIP ViT-L/14	79.79	79.72
CLIP ViT-H/14	83.88	83.54
CLIP ViT-G/14	82.55	82.30
SigLIP 2 base/16-224	76.68	76.36
SigLIP 2 large/16-256	78.68	78.41

Semantic direction of wrong predictions. For each wrong prediction, we encode the predicted answer and the ground-truth answer with CLIP ViT-G/14 and compute cosine similarity. The random-GT baseline re-pairs each wrong prediction with an arbitrary answer from the same task. Across all four VLMs and both task regimes in Figure 3, the wrong-prediction distribution is shifted right of this random baseline; Welch t -statistics range from 32.8 to 64.6 ($p < 10^{-100}$ in every cell). For the reference model, the mean gap is $\Delta = +0.117$ on classification (0.684 vs. 0.567) and $\Delta = +0.144$ on VQA (0.662 vs. 0.518); gaps across all eight cells fall in $[+0.085, +0.168]$.

Mid-similarity examples illustrate what the aggregate statistics capture. Classification mistakes often confuse labels within the same visual family, such as *song sparrow* with *fox sparrow* and *brown pelican* with *white pelican*. VQA mistakes are often synonym- or category-level confusions, such as *children* with *kid*, *round* with *circle*, and *apple* with *mac*. These examples are still incorrect under the task metric, but they preserve enough semantic direction to make prediction-conditioned hard-negative selection plausible.

D Effect of the Similarity Encoder

SMILE relies on a frozen similarity encoder $E = (\text{Emb}_I, \text{Emb}_T)$ for two operations: image-similarity positive retrieval and text-similarity hard-negative selection. The main experiments use CLIP ViT-G/14 (LAION-2B). Table 7 reports SMILE performance on CUB-200 with Qwen2.5-VL-7B as we swap this encoder across seven open alternatives spanning two families (CLIP and SigLIP 2) and four parameter scales, holding all other settings fixed.

Three observations emerge. *First, performance scales smoothly with encoder capacity within a family.* On CLIP, EM rises from 70.85 at ViT-B/32 to 83.88 at ViT-H/14, with strict monotonicity in encoder size; the SigLIP 2 base→large step shows the same direction (76.68 → 78.68). *Second, the paper baseline is near the top.* CLIP ViT-G/14 (LAION-2B) at 82.55 EM is within 1.3 points of the best encoder we tested (CLIP ViT-H/14 LAION-2B, 83.88) and is the second-best entry overall, confirming that our headline numbers are not propped up by a uniquely strong retriever. *Third, SMILE is robust across the full encoder range:* even the smallest CLIP encoder (ViT-B/32) yields 70.85 EM, far above zero-shot (14.83) and the Random baseline (26.27) on the same Qwen2.5-VL-7B backbone (Table 1). The benefit of mining and contrasting model errors is therefore preserved even when the retriever is significantly downsized, suggesting that the dominant source of SMILE’s gain is the prediction-conditioned hard-negative signal rather than encoder strength.

E Additional Backbone Results

Table 8 reports SMILE results on two additional Qwen3-VL backbones. We use the same CLIP encoder and task-specific K, N settings as in the main experiments.

Table 8: **Additional SMILE results on Qwen3-VL backbones.** CUB and Flowers report accuracy / weighted F1; OK-VQA and VizWiz report exact match / token-level F1.

Model	CUB		Flowers		OK-VQA		VizWiz	
	Acc	F1	Acc	F1	EM	F1	EM	F1
Qwen3-VL-4B	84.08	89.42	99.45	99.46	44.15	49.03	81.94	87.95
Qwen3-VL-8B	83.87	89.41	99.61	99.62	47.50	51.70	82.39	87.51

Table 9: **Fair-subset analysis: queries restricted to classes with $\geq k$ examples in both pools.** Natural wins at small k (best individual visual matches); clean-only dominates at $k \geq 4$ (consistent class signal beats individual visual quality); hard-only never wins on a fair subset.

k	queries	natural	hard-only	clean-only	clean-hard
2	2,161	90.19	83.80	86.49	+2.69
4	1,584	89.77	83.27	92.42	+9.15
8	844	89.22	79.74	94.55	+14.81
12	385	87.79	78.44	96.10	+17.66

F Pool Decomposition: Phase Transition with k

This appendix expands the main-paper pool-decomposition (Section 5.3, Table 2) with formal slice definitions and a phase-transition analysis. All numbers on CUB with Qwen2.5-VL-7B; primary metric is exact match.

Pool partition and slice definitions. Hard-negative mining via $R=3$ -vote zero-shot (Section 4) places a CUB labeled image in the hard pool if the model’s majority vote disagrees with the gold label, and in the clean pool otherwise. This yields 1,113 clean and 4,881 hard images out of 5,994 total (81% hard). The class coverage of the two pools is asymmetric: only 123 of CUB’s 200 species appear in the clean pool, and the remaining 77 species have *zero* model-correct labeled examples. We define three test-side slices accordingly:

- **Uncovered** (2,255 queries, 39% of test): the query’s true class has 0 examples in the clean pool, so any clean retrieval is structurally off-class.
- **Fair $\geq k$** (at $k=4$: 1,584 queries, 27% of test): the query’s true class has $\geq k$ examples in *both* pools, so clean and hard retrieval can be compared apples-to-apples.
- **Full test** (5,794 queries): no slicing.

Phase transition with k . Table 9 shows how the clean-vs-hard gap sharpens as the fair-subset threshold k grows; each row is restricted to classes with $\geq k$ examples in both pools.

The hard pool is therefore not intrinsically better than the clean pool: it is essential precisely because it covers the classes the model gets wrong, and CLIP image similarity surfaces it implicitly without any need for explicit pool-routing logic at inference time.

G Prompt Template

We adopt a unified prompting template across all backbones and tasks. The SMILE template, shown below, is structured as follows: the query is presented first; K positive demonstrations identified by image-embedding similarity follow; N prediction-conditioned hard negatives are appended under the explicit label “*tricky examples that were previously confused with the predicted answer*”; and the query is repeated at the end for clarity. Each in-context example is rendered as the original Image / Question / Answer triple. For hard negatives we always show the *ground-truth* answer of the labeled example, not the model’s stored mistaken prediction $\hat{y}_h$ (the latter serves only as the selection key during retrieval, Equation (2)). All VLMs use greedy decoding (temperature 0); for classification (CUB and Flowers) we additionally enforce a closed-set answer space by appending the candidate class names, while OK-VQA and VizWiz are evaluated open-ended.

Prompt template for VLM inference under SMILE

Your task is to perform {{Task Type}}.

{{Query Image}}

Question: {{Query Question}}

Here are {{K}} in-context examples to help you answer the question:

{{Positive Image 1}}

Question: {{Positive Question 1}}

Answer: {{Positive Answer 1}}

{{Positive Image 2}}

.....

Below are {{N}} tricky examples that were previously confused with the predicted answer:

{{Hard-Negative Image 1}}

Question: {{Hard-Negative Question 1}}

Answer: {{Hard-Negative Ground-Truth Answer 1}}

{{Hard-Negative Image 2}}

.....

Here is the original question again.

{{Query Image}}

Question: {{Query Question}}

Please provide your response by directly outputting the answer.

H Limitations

SMILE has several boundaries that we make explicit here.

Dependence on the informativeness of the zero-shot prediction. The quality of hard-negative selection depends on the query’s zero-shot prediction $\hat{y}_q$ being a meaningful error direction. Some predictions will be semantically unrelated to the ground truth, overly generic, or driven by missing world knowledge rather than a local visual or semantic confusion; in these cases, prediction-conditioned selection cannot identify corrective hard negatives because the model has not produced a usable direction to follow. We view this as a boundary of inference-time demonstration selection rather than a flaw specific to SMILE: when the initial prediction is entirely off-manifold, the underlying limitation is likely in the model’s internal knowledge or reasoning capability, which may require parameter updates such as supervised fine-tuning or reinforcement learning rather than better demonstration selection alone.

Prompt-structure choices are not adaptive. Our current implementation uses a fixed prompt structure with a capped number of hard negatives. More sophisticated prompt organizations or adaptive negative budgets that respond to per-query difficulty may yield further improvements but are outside the scope of this paper.

Benchmark scope. Our evaluation focuses on four benchmarks with available labeled corpora. Extending the approach to settings where examples must be identified from web-scale corpora, or where no labeled pool is available, remains an open direction.