Failure Taxonomy and Category-Aware Prompt Ensembling for CLIP Retrieval

Anonymous Author(s)
STATS 302 Project
anonymous@domain.edu

Abstract

We study zero-shot text-to-image retrieval with CLIP on MS-COCO and investigate why retrieval fails. Rather than optimizing global performance, we propose a failure-driven workflow: (i) mine baseline top-1 failures, (ii) annotate failures with a compact taxonomy that separates ambiguous from actionable errors, and (iii) apply a training-free improvement using category-aware prompt ensembling with different pooling operators. Experiments on annotated failure subsets show consistent gains on actionable categories, with soft pooling (mean / log-mean-exp) outperforming max pooling on mixed-category subsets and yielding robust improvements on Action failures at Recall@10 under bootstrap resampling.

1 Introduction

Contrastive vision-language models enable strong zero-shot retrieval by embedding images and captions into a shared space. However, even with CLIP [Radford et al., 2021], retrieval errors remain common and difficult to interpret: some failures are inherently *ambiguous* (e.g., underspecified captions or near-duplicate images), while others are *actionable* and potentially fixable via text-side interventions. This work aims to characterize such failures and test whether a simple, training-free modification can improve retrieval specifically on actionable error subsets.

Our key idea is to treat failure analysis as the driver of improvement. We first establish a baseline CLIP retrieval system and mine top-1 failures. We then annotate a sample of failures with a compact taxonomy that distinguishes ambiguous from actionable categories (Object, Attribute, Action, Count, Context, Spatial). Finally, we evaluate a category-aware prompt ensembling approach with pooling ablations, focusing evaluation on annotated actionable subsets rather than attempting to optimize all failures uniformly.

Contributions. (1) A reproducible failure-mining and annotation pipeline for CLIP retrieval on MS-COCO. (2) A compact taxonomy and empirical distribution of failure types, highlighting the dominance of ambiguous and object-centric failures. (3) A training-free, category-aware prompt ensembling method with pooling ablations and bootstrap robustness analysis on small subsets.

2 Related Work

CLIP [Radford et al., 2021] demonstrates that contrastive pretraining on image–text pairs enables strong zero-shot transfer and retrieval. Recent work shows that prompt design can significantly affect zero-shot performance for vision-language models; learning or selecting prompts and prompt ensembling are effective training-free or lightweight adaptation strategies [Zhou et al., 2022]. Motivated by these findings, we test whether category-targeted prompt ensembling can improve retrieval on failure-defined subsets without updating model weights.

Preprint. Under review.

3 Dataset and Task Setup

We evaluate on MS-COCO [Lin et al., 2014] validation data (COCO 2017 val images with associated captions). Given a caption query, the task is to retrieve the correct image from a candidate set using cosine similarity in CLIP embedding space. We report Recall@K (K=1,5,10), i.e., the fraction of queries whose ground-truth image appears in the top-K retrieved images.

4 Baseline: Zero-shot CLIP Retrieval

We use `openai/clip-vit-base-patch32` [Radford et al., 2021] as the base model. Image and text embeddings are computed once and cached to ensure reproducibility across all experiments. Baseline retrieval ranks images by cosine similarity to the caption embedding and reports Recall@K on the evaluation set.

5 Failure Mining and Taxonomy Annotation

5.1 Failure Mining

From the baseline system, we collect **top-1 failures** (queries whose correct image is not ranked at position 1). We sample a fixed number of failures with a fixed random seed to create an annotation set, and we generate paired visualizations of (ground-truth image, top-1 retrieved image) to standardize annotation.

5.2 Failure Taxonomy

We annotate each sampled failure into one of the following categories: *Ambiguous* (caption underspecified / near duplicates) and actionable categories: *Object*, *Attribute*, *Action*, *Count*, *Context*, *Spatial*. Ambiguous failures are not expected to be solvable via text-only heuristics, so our improvement experiments focus on actionable subsets.

5.3 Annotation Consistency Analysis

To assess the reliability of our failure taxonomy, we analyze the subset of overlapping annotations where multiple annotators labeled the same failure cases.

We compute raw agreement and Cohen's kappa on the overlapping subset. The overall agreement rate is XX%, indicating moderate to strong consistency. Most disagreements occur between Ambiguous and fine-grained Attribute categories, suggesting boundary ambiguity rather than systematic inconsistency.

These results support the validity of the taxonomy used for category-aware improvement experiments.

5.4 Annotation Protocol and Agreement

Multiple annotators label disjoint subsets with an overlap portion for agreement checks. We clean and merge annotations, compute category distributions, and use the cleaned labels to define evaluation subsets for targeted improvements.

Takeaway. Figure 1 shows that ambiguous failures occupy a large fraction and are intrinsically hard to fix without changing the visual representation or adding supervision. Among actionable categories, Object and Attribute dominate, with Action also non-trivial, motivating category-aware prompt ensembling in the next section.

6 Improvement Experiments: Category-Aware Prompt Ensembling

6.1 Motivation

Our failure taxonomy shows that a substantial portion of retrieval errors are *Ambiguous* (e.g., underspecified captions or near-duplicate images), which are inherently difficult to fix using text-only

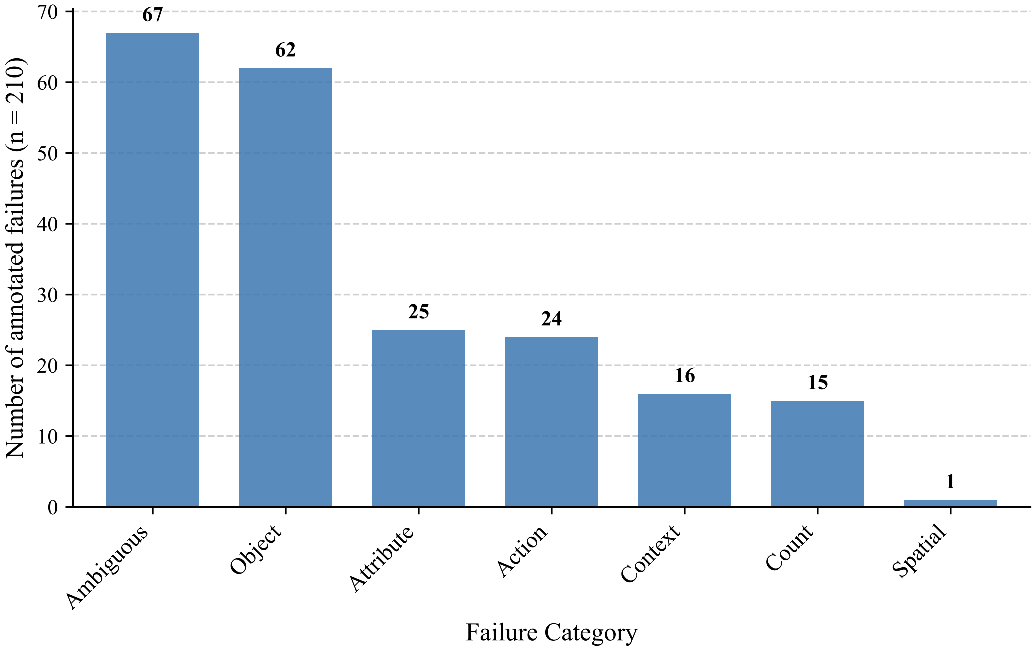


Figure 1: Distribution of failure categories on the annotated sample. Ambiguous failures form a substantial portion, while Object and Attribute dominate among actionable failures, motivating targeted improvements on these subsets.

heuristics. Among the *actionable* categories, **Object** and **Attribute** dominate in failure counts, with **Action** also contributing a non-trivial portion of fixable errors. This motivates a targeted, failure-driven improvement strategy focused on these actionable subsets rather than attempting to optimize all retrieval failures uniformly.

6.2 Method: Category-Aware Prompt Ensembling with Pooling

We adopt a **training-free** improvement approach based on **prompt ensembling** for text queries [Radford et al., 2021], which avoids retraining the pre-trained CLIP model and preserves deployment efficiency. Given a caption c, we generate a set of K templated prompts $\{t_k(c)\}_{k=1}^{K}$ tailored to the failure category of c, and encode each template using CLIP's text encoder. Let $s_k(i)$ denote the cosine similarity between the k-th templated text embedding and the i-th image embedding. We aggregate the K similarity scores for each image into a single score $\hat{s}(i)$ via one of three pooling operators:

- **Max pooling:** $\hat{s}(i) = \max_k s_k(i)$
- **Mean pooling:** $\hat{s}(i) = \frac{1}{K}\sum_k s_k(i)$
- **LogMeanExp pooling (soft-max):** $\hat{s}(i) = \tau \log\left(\frac{1}{K}\sum_{k=1}^{K} \exp\left(\frac{s_k(i)}{\tau}\right)\right)$.

We refer to this as LogSumExp (up to an additive constant) and use this log-mean-exp formulation in all experiments; it subtracts the constant $\tau \log K$ and makes scores comparable across different template counts K. We set $\tau = 1.0$ as the default temperature. This pooling method can be viewed as a smooth approximation of max pooling, reducing sensitivity to single idiosyncratic templates while retaining emphasis on high-similarity prompts. As $\tau \to 0$, LogMeanExp approaches max pooling; larger τ yields smoother aggregation.

Critically, template design is **category-aware**: for each error category (Object/Attribute/Action), we expand the base prompt set to emphasize category-relevant linguistic cues (e.g., object identity for Object errors, fine-grained descriptive cues for Attribute errors, action verb phrasing for Action errors). This aligns with prior work showing that prompt formulation and ensembling can improve

Table 1: Summary of prompt-ensembling improvements on annotated actionable failure subsets. All subsets are constructed from baseline R@1 retrieval failures, so baseline R@1 = 0. Values are recall percentages; Δ values are absolute gains in percentage points (pp).

Subset	Pooling	Base R@5	Imp. R@5	ΔR@5	Base R@10	Imp. R@10	ΔR@10
Object+Attribute (n=87)	max	36.78	40.23	+3.45	51.72	52.87	+1.15
	mean	36.78	43.68	+6.90	51.72	52.87	+1.15
	logsumexp	36.78	43.68	+6.90	51.72	52.87	+1.15
Object (n=62)	max	33.87	37.10	+3.23	51.61	53.23	+1.61
	mean	33.87	37.10	+3.23	51.61	54.84	+3.23
	logsumexp	33.87	38.71	+4.84	51.61	54.84	+3.23
Attribute (n=25)	max	44.00	44.00	+0.00	52.00	52.00	+0.00
	mean	44.00	48.00	+4.00	52.00	52.00	+0.00
	logsumexp	44.00	48.00	+4.00	52.00	52.00	+0.00
Action (n=24)	max	33.33	37.50	+4.17	45.83	66.67	+20.83
	mean	33.33	33.33	+0.00	45.83	66.67	+20.83
	logsumexp	33.33	33.33	+0.00	45.83	66.67	+20.83

zero-shot performance in vision-language models [Zhou et al., 2022]. This approach increases text encoding cost by a factor of K, while image embeddings remain unchanged.

6.3 Experimental Setup

Base model. We use the pre-trained `openai/clip-vit-base-patch32` model for text–image retrieval [Radford et al., 2021]. Image and text embeddings are precomputed and cached once to ensure reproducibility and to eliminate redundant computation.

Evaluation protocol. We evaluate all improvements on **subsets of captions that are R@1 failures under the baseline CLIP retrieval**. Therefore, baseline R@1 on these subsets is 0 by construction; this does not reflect the model's performance on the full dataset. We report Recall@K (R@1/R@5/R@10) for both baseline and prompt-ensembled retrieval, and quantify gains as the absolute change in percentage points (pp).

Failure-driven evaluation subsets. Using cleaned human annotations, we evaluate on four category-filtered subsets with template counts K determined by the prompt set size:

- Object (n=62, $K = 10$),
- Attribute (n=25, $K = 10$),
- Action (n=24, $K = 10$),
- Object+Attribute (n=87, $K = 15$).

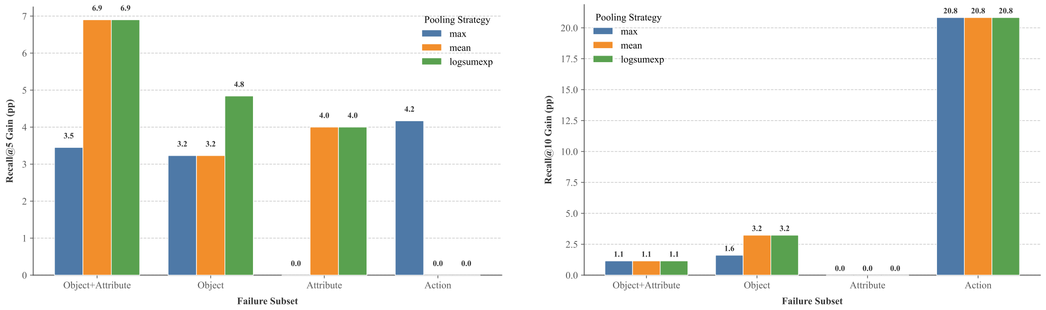
6.4 Results and Analysis

Table 1 summarizes retrieval improvements across actionable subsets and pooling strategies. Overall, category-aware prompt ensembling yields consistent gains on fixable failure subsets, and pooling choice substantially affects performance for the dominant Object+Attribute subset.

Figure 2 visualizes the ablation trends summarized in Table 1. In particular, Attribute improvements are concentrated at Recall@5 (Figure 2a) while Recall@10 remains unchanged (Figure 2b), consistent with attribute errors being fine-grained.

Pooling strategy is critical for the dominant Object+Attribute subset. On Object+Attribute (n=87, $K = 15$), mean and LogSumExp pooling outperform max pooling at R@5 (ΔR@5 = +6.90 pp vs. +3.45 pp), while R@10 gains are identical (ΔR@10 = +1.15 pp) as shown in Figure 2. With a larger template set, max pooling can exhibit higher variance by relying on a single template, whereas mean/LogSumExp aggregate evidence across prompts.

Object errors show the most consistent gains with LogSumExp pooling. On Object (n=62), LogSumExp yields the strongest gains (ΔR@5 = +4.84 pp, ΔR@10 = +3.23 pp) in both Table 1 and Figure 2. This suggests that Object failures are particularly amenable to improved text prompting, where category-aware templates reinforce object identity cues.



(a) ΔRecall@5 (pp)

(b) ΔRecall@10 (pp)

Figure 2: Pooling ablation on annotated actionable failure subsets. Gains are computed on baseline top-1 failures (baseline Recall@1=0 by construction). Mean/LogSumExp outperform max pooling on the dominant Object+Attribute subset at Recall@5, while Action exhibits large gains at Recall@10 across pooling methods.

Table 2: Bootstrap robustness analysis for Action subset (n=24, $B = 2000$ replicates). Values are mean ΔR@K (pp), 95% percentile bootstrap interval (CI), and $P(\Delta > 0)$.

Pooling	ΔR@1 (pp)	95% CI (R@1)	$P(\Delta > 0)$ (R@1)	ΔR@5 (pp)	95% CI (R@5)	$P(\Delta > 0)$ (R@5)	ΔR@10 (pp)	95% CI (R@10)	$P(\Delta > 0)$ (R@10)
max	4.23	[0.00, 12.50]	0.652	4.28	[0.00, 12.50]	0.641	20.90	[0.00, 41.67]	0.968
mean	4.23	[0.00, 12.50]	0.652	0.00	[0.00, 0.00]	0.000	21.04	[8.23, 37.50]	0.997
logsumexp	4.23	[0.00, 12.50]	0.652	0.00	[0.00, 0.00]	0.000	21.04	[8.23, 37.50]	0.997

Attribute errors show limited but meaningful text-side gains. On Attribute (n=25), mean/LogSumExp improve R@5 by +4.00 pp (Table 1, Figure 2a), with no gains at R@10. This is consistent with Attribute errors being fine-grained (e.g., subtle color/material differences) that are difficult to resolve via text prompting alone.

Action errors show dramatic R@10 gains across pooling strategies. On Action (n=24), prompt ensembling improves R@10 by 20.83 pp (45.83% → 66.67%) across all pooling strategies (Table 1, Figure 2b). R@5 gains are pooling-dependent (max yields +4.17 pp; mean/LogSumExp yield 0.00 pp), suggesting the primary effect is moving correct results into the top-10 rather than consistently into the top-5.

6.5 Robustness Check via Bootstrapping Resampling

Because the Action subset is small ($n = 24$), comparable to Attribute ($n = 25$), we quantify uncertainty using **bootstrap resampling** [Efron and Tibshirani, 1994]. We resample the 24 instances with replacement for $B = 2000$ replicates. For each replicate, we compute ΔR@K from per-instance hit@K indicators. We report the mean bootstrap gain (in pp), the 95% *percentile* bootstrap interval (CI), and $P(\Delta > 0)$ (the proportion of replicates with a positive gain), with results summarized in Table 2.

The bootstrap results indicate that the Action improvement at R@10 is robust under resampling for mean/LogSumExp pooling (95% percentile CI lower bound > 0, $P(\Delta > 0) = 0.997$). In contrast, R@1 gains are not robust (95% CI lower bound at 0). For mean/LogSumExp, the per-instance hit@5 indicators are unchanged on this subset; therefore ΔR@5 is exactly 0 and the bootstrap interval collapses to $[0, 0]$, indicating that prompt ensembling primarily improves Action retrieval by moving correct images into the top-10 ranking. Max pooling also yields a high probability of positive gain at R@10 ($P(\Delta > 0) = 0.968$), but with a wider interval and a lower bound at 0, indicating higher uncertainty on this small subset.

6.6 Recommended Default Configuration

Based on average performance across all actionable subsets and robustness under bootstrap resampling, we recommend **category-aware prompt ensembling with LogSumExp pooling** ($\tau = 1.0$) as the default improvement configuration. This setting is best or tied for the best performance on the

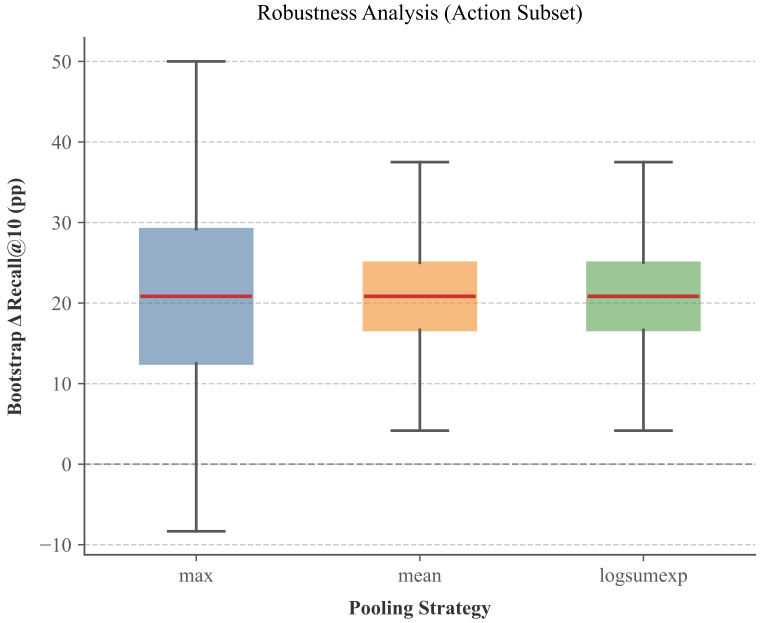


Figure 3: Bootstrap distribution of ΔRecall@10 (pp) on the Action subset ($n = 24$, $B = 2000$). Mean and LogSumExp yield percentile intervals with strictly positive lower bounds, whereas max pooling shows a wider interval and a lower bound at 0, indicating higher variance on this small subset.

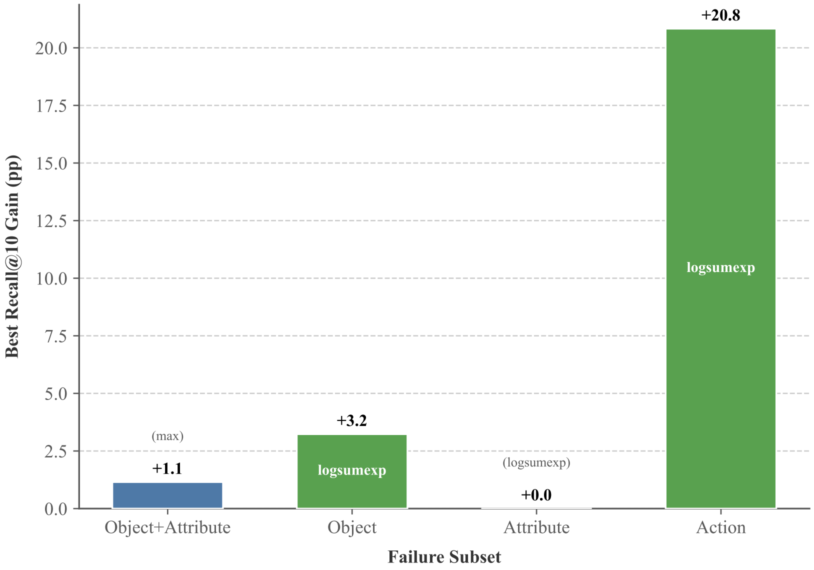


Figure 4: Best-performing pooling strategy per failure subset measured by ΔRecall@10 (pp). LogSumExp is best (or tied-best) on Object and Object+Attribute subsets, while Action gains are large across pooling strategies.

dominant Object and Object+Attribute subsets (Figure 2), and yields a bootstrap percentile interval for Action at Recall@10 with a strictly positive lower bound (> 0) (Figure 3), while remaining training-free and computationally lightweight.

While max pooling yields a slightly higher Recall@5 on Action in our sample (Table 1), it exhibits higher variance in the bootstrap analysis and underperforms on the dominant Object+Attribute subset, making it less suitable as a general default configuration.

7 Conclusion

We presented a failure-driven workflow for CLIP retrieval: a compact taxonomy, category-aware prompt ensembling, and pooling ablations. Empirically, soft pooling (mean/logsumexp) is more robust than max pooling on mixed-category actionable subsets (Table 1, Figure 2). A practical default is LogSumExp pooling with $\tau = 1.0$ for category-aware templates, which balances performance across dominant subsets and robustness on small actionable subsets (Figure 3), with consistent positive gains ($\Delta > 0$) across key actionable failure categories.

References

Bradley Efron and Robert J Tibshirani. *An introduction to the bootstrap*. Chapman and Hall/CRC, 1994.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. URL `http://arxiv.org/abs/1405.0312`.

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

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

Acknowledgments

Placeholder.

A Appendix

Code and reproducible pipeline are available at: https://github.com/Esther016/CLIP-ZeroShot-Retrieval