TEXTUAL RESIDUAL PROMPT FOR FINE-GRAINED REAL-TIME OPEN-VOCABULARY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Real-time open-vocabulary detectors combine open-category recognition with efficient inference, but still lag significantly behind heavy open-vocabulary detectors on fine-grained detection tasks. We observe an interesting property in the textual embedding space: semantic residuals induced by the same fine-grained attribute remain highly consistent across different base categories, suggesting that fine-grained attributes correspond to relatively stable and transferable semantic directions. Motivated by this observation, we propose **Textual Residual Prompt (TRP)**, a lightweight adaptation method for fine-grained real-time open-vocabulary detection. Instead of modifying the detector or relearning complete textual representations, TRP learns textual residuals to enhance fine-grained discriminability. The resulting textual representations can still be cached before inference, preserving the efficient prompt-then-detect paradigm. Experiments show that TRP achieves superior fine-grained detection performance among parameter-efficient fine-tuning methods while maintaining real-time inference efficiency.

1 INTRODUCTION

Applications such as robotic perception and interactive visual search require detectors to respond quickly to user queries while distinguishing objects with subtle differences. Open-vocabulary object detection enables detectors to recognize arbitrary object categories specified by natural-language descriptions, extending conventional detection beyond a predefined label space (Radford et al., 2021; Li et al., 2022; Liu et al., 2024b; Zhong et al., 2022; Yao et al., 2022). However, representative open-vocabulary detectors such as GLIP and Grounding DINO typically rely on computationally intensive architectures, limiting their deployment in latency-sensitive applications. Recent real-time open-vocabulary detectors, represented by YOLO-World (Cheng et al., 2024), combine language-guided recognition with efficient one-stage detection and follow a prompt-then-detect paradigm, where textual category embeddings can be pre-computed and cached before inference. Despite achieving competitive performance on general object categories, these detectors remain limited in distinguishing fine-grained categories characterized by subtle differences

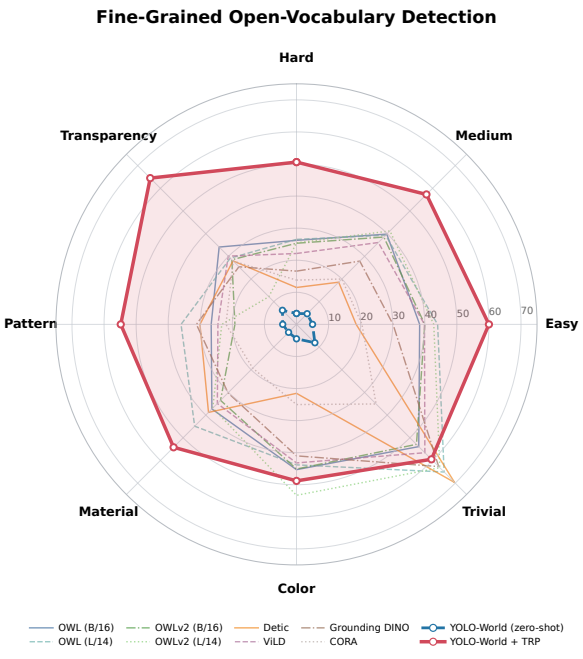


Figure 1: Fine-grained performance of representative open-vocabulary detectors (Wu et al., 2023; Minderer et al., 2023). TRP substantially improves YOLO-World and makes it competitive with heavier detectors. The scores for all eight FG-OVD benchmarks are reported in Appendix Table 8.

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

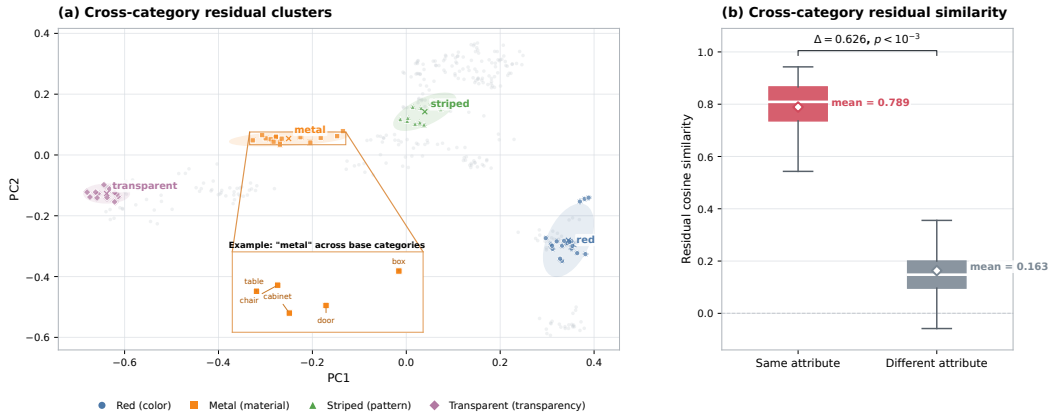


Figure 2: Analysis of fine-grained semantic residuals in the frozen CLIP textual embedding space. **(a)** PCA visualization of cross-category residual clusters. Four representative attributes are highlighted, while the remaining residuals are shown in gray. The inset illustrates the residuals induced by *metal* across different base categories. **(b)** Cross-category cosine similarity between residuals induced by the same or different attributes. White diamonds denote the mean values.

in color, material, pattern, or transparency (Yuksekgonul et al., 2023; Bianchi et al., 2024b). As shown in Fig. 1, a substantial performance gap still exists between real-time detectors and heavier models such as Grounding DINO on fine-grained benchmarks. We therefore investigate fine-grained real-time open-vocabulary detection, aiming to improve fine-grained discriminability without sacrificing inference efficiency.

Existing adaptation strategies remain insufficient for fine-grained real-time open-vocabulary detection. Full-model or detection-head fine-tuning can improve fine-grained performance but may impair the original open-vocabulary capability and increase model-maintenance costs (Medeiros et al., 2025; Yu et al., 2023). Visual-side adaptation methods, such as ModPrompt (Medeiros et al., 2025), adapt input images while keeping the detector frozen, but are primarily designed to bridge visual modality gaps rather than exploit fine-grained distinctions explicitly encoded in language. Existing text-side adaptation methods optimize textual prompts or classifier representations for downstream tasks (Yu et al., 2023; Zhou et al., 2022b;a; Zhu et al., 2023; Yao et al., 2024; 2023; Bulat & Tzimiropoulos, 2023). However, they do not explicitly model the semantic displacement from a base category to its fine-grained variants, which may limit their ability to capture attribute-related structures shared across base categories.

To understand how fine-grained distinctions can be better exploited, we examine how attributes change the representation of a base category in the text embedding space. Prior work suggests that pretrained vision-language representations already encode fine-grained information, although conventional similarity matching may not fully exploit it (Bianchi et al., 2024a). To examine how such information is organized in the textual space, we analyze the frozen CLIP text embeddings. Let $E(\cdot)$ denote the text encoder, c a base category, and a a fine-grained attribute. We define the attribute-induced textual residual as

$$\mathbf{r}(a, c) = E(a, c) - E(c), \tag{1}$$

where, for example, setting a to *metal* and c to *box* gives the semantic displacement between the embeddings of "a metal box" and "a box".

As shown in Fig. 2, residuals induced by the same attribute remain highly consistent across different base categories. For example, the semantic displacement from "a box" to "a metal box" is similar to that from "a chair" to "a metal chair", with the same pattern also observed for cabinet, door, and table. Residuals induced by the same attribute cluster in nearby regions and exhibit substantially higher cross-category cosine similarity than those induced by different attributes. This suggests that fine-grained attributes induce relatively consistent semantic directions across base categories in the textual embedding space.

Motivated by this observation, we propose **Textual Residual Prompt (TRP)** for fine-grained real-time open-vocabulary detection. TRP uses a lightweight residual generator shared across categories to model the semantic displacement between the frozen text embeddings of each base category and its fine-grained variant, enhancing fine-grained discriminability. Since both the text encoder and detector remain frozen, the enhanced textual representations can be pre-computed and cached, preserving the prompt-then-detect pipeline without additional online text-side computation.

Across the difficulty-based and attribute-based FG-OVD benchmarks (Bianchi et al., 2024b), TRP outperforms the strongest parameter-efficient baseline, TCR. The gains are especially pronounced on Hard and Color, reaching 3.9 and 5.3 mAP points, respectively. TRP also improves the overall mAP of YOLOE-v8s from 38.1% to 51.3%. With cached query features, it retains 113.9 FPS on YOLO-Worldv2-S and 150.5 FPS on YOLOE-v8s.

Our contributions are summarized as follows:

- We reveal a structured property of fine-grained semantics in the textual embedding space: residuals induced by the same attribute exhibit strong consistency across different base categories, indicating shared attribute-induced directions.
- Based on this observation, we propose **Textual Residual Prompt (TRP)**, a lightweight adaptation method that explicitly learns the semantic displacement between base categories and their fine-grained variants while preserving the cacheable prompt-then-detect paradigm.
- Extensive experiments on FG-OVD show that TRP outperforms the strongest parameter-efficient baseline, with gains of 3.9 and 5.3 mAP points on Hard and Color, respectively. TRP also improves YOLOE-v8s while retaining real-time inference speeds on both YOLO-Worldv2-S and YOLOE-v8s.

2 Related Works

2.1 Real-Time Open-Vocabulary Detection

Open-vocabulary detection leverages vision–language models such as CLIP (Radford et al., 2021) to recognize categories beyond a fixed label space (Kuo et al., 2023). ViLD and Detic transfer large-scale vision–language knowledge to object detectors through knowledge distillation or image-level supervision (**?**Zhou et al., 2022c), while GLIP unifies object detection and phrase grounding (Li et al., 2022). OWL-ViT scales contrastive vision–language pre-training for text-conditioned detection (Minderer et al., 2022), and Grounding DINO introduces deep cross-modal fusion for open-set localization (Liu et al., 2024b). Despite their strong performance, these models typically incur substantial computational costs. More recently, YOLO-World introduces the prompt-then-detect paradigm for efficient one-stage open-vocabulary detection (Cheng et al., 2024), OmDet-Turbo employs an efficient multimodal fusion head for real-time Transformer-based detection (Zhao et al., 2024), and YOLOE further improves efficiency through re-parameterizable text alignment (Wang et al., 2025). However, these methods primarily focus on general category-level recognition and remain limited in distinguishing categories defined by subtle attributes.

2.2 Fine-Grained Open-Vocabulary Detection

Fine-grained open-vocabulary detection requires models to distinguish objects described by subtle attributes such as color, material, pattern, and transparency (Liu et al., 2025). FG-OVD (Bianchi et al., 2024b) introduces difficulty-based and attribute-based benchmarks with dynamically constructed hard-negative descriptions, revealing that detectors performing well on standard open-vocabulary benchmarks often struggle with fine-grained semantics. Recent methods improve fine-grained recognition from different perspectives. SHiNe constructs hierarchy-aware classifiers using super- and sub-category knowledge (Liu et al., 2024a), DVDet employs LLM-generated visual descriptors to enhance region–text alignment (Jin et al., 2024), and GUIDED decomposes subject localization and attribute discrimination to alleviate their semantic entanglement (Li et al., 2026). Complementary to these approaches, TRP focuses on lightweight adaptation for real-time detectors by exploiting the structured semantic residuals induced by fine-grained attributes in the textual embedding space.

2.3 Parameter-Efficient Adaptation of Vision-Language Models

Parameter-efficient adaptation updates a small set of task-specific parameters while keeping most pretrained parameters frozen (Gao et al., 2024; Khattak et al., 2023a;b; Li et al., 2023). CoOp learns shared textual context vectors, while CoCoOp generates instance-conditioned prompts to improve adaptability (Zhou et al., 2022b;a). TaskRes learns an independently parameterized additive residual for each downstream category (Yu et al., 2023), and DetPro develops detection-specific prompt learning by explicitly considering foreground and background proposals (Du et al., 2022). On the visual side, ModPrompt adapts vision–language detectors to new visual modalities through learnable input transformations (Medeiros et al., 2025). Although effective, these methods do not explicitly model the semantic relationship between a base category and its fine-grained variants: CoOp and CoCoOp learn task- or instance-conditioned prompt adjustments, while TaskRes learns category-specific residuals independently. In contrast, TRP formulates adaptation as relational residual learning, where a shared generator models the attribute-induced semantic displacement between paired base and fine-grained category representations.

3 Method

3.1 Problem Formulation

Let $\mathcal{D} = \{(x_n, Y_n)\}_{n=1}^{N}$ denote a fine-grained detection dataset, where x_n is an input image and Y_n contains its detection annotations. For each fine-grained category, we define a pair of textual descriptions (q_i^b, q_i^f), where q_i^b denotes the base category (e.g., "car") and q_i^f denotes its fine-grained variant (e.g., "red car"). A frozen CLIP text encoder $E(\cdot)$ produces their embeddings as

$$\mathbf{t}_i^b = E(q_i^b), \qquad \mathbf{t}_i^f = E(q_i^f), \tag{2}$$

where $\mathbf{t}_i^b, \mathbf{t}_i^f \in \mathbb{R}^D$.

To improve fine-grained discrimination while keeping the detector frozen, we propose **Textual Residual Prompt (TRP)**. TRP uses the semantic displacement between each base category and its fine-grained variant to adapt their text embeddings. Given K category pairs in the current query set, it produces adapted representations as follows:

$$\widehat{\mathbf{t}}_i = \mathrm{TRP}_\phi(\mathbf{t}_i^b, \mathbf{t}_i^f), \qquad \widehat{\mathbf{T}}_\phi = \{\widehat{\mathbf{t}}_i\}_{i=1}^{K}, \tag{3}$$

where ϕ denotes the learnable parameters of the shared residual generator.

Let F_θ denote a frozen real-time open-vocabulary detector. TRP is optimized by

$$\min_{\phi} \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_{\mathrm{det}} \left(F_\theta(x_n; \widehat{\mathbf{T}}_\phi), Y_n \right), \tag{4}$$

while keeping both the text encoder E and detector parameters θ frozen. The detailed construction of $\widehat{\mathbf{t}}_i$ is introduced in the following subsection.

3.2 Attribute-Induced Textual Residual Prompt

As shown in Fig. 2, textual residuals induced by the same fine-grained attribute exhibit similar directions across different base categories. Based on this observation, we represent the semantic displacement between a fine-grained category and its base category as

$$\mathbf{d}_i = \mathbf{t}_i^f - \mathbf{t}_i^b. \tag{5}$$

For example, $E(\text{"red car"}) - E(\text{"car"})$ describes the semantic displacement induced by the attribute "red". Rather than learning an unconstrained residual independently for each category, TRP models this paired relation using a residual generator shared across categories, encouraging recurring attribute-induced structures to be captured.

As illustrated in the top part of Fig. 3, the semantic displacement is fed into a lightweight multilayer residual generator g_ϕ, implemented as a stack of linear layers interleaved with ReLU activations, to generate a textual residual prompt:

$$\mathbf{r}_i = g_\phi(\mathbf{d}_i). \tag{6}$$

The generated residual is then combined with the frozen fine-grained textual embedding $\mathbf{t}_i^f$:

$$\widehat{\mathbf{t}}_i = \mathrm{Norm}\left(\mathbf{t}_i^f + \mathbf{r}_i\right), \tag{7}$$

where $\mathrm{Norm}(\cdot)$ denotes ℓ_2 normalization. Here, $\mathbf{t}_i^f$ preserves the original fine-grained textual semantics, while $\mathbf{r}_i$ provides a relation-conditioned semantic correction. Since the bias terms of g_ϕ are shared across all categories, they naturally capture the category-agnostic task-level adjustment without requiring an additional shared prompt. Applying this process to all queries produces the adapted textual representations $\widehat{\mathbf{T}}_\phi$ defined in Eq. 3.

As shown in the bottom panel of Fig. 3, the adapted textual representations can be computed and cached after training and subsequently reused by the lightweight detector.

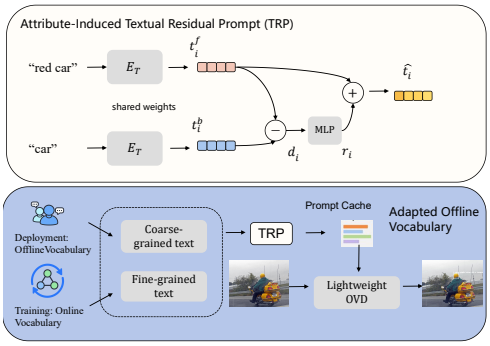


Figure 3: Overview of TRP. **Top:** Residual generation. **Bottom:** Offline vocabulary construction.

3.3 Optimization and Adapted Offline Vocabulary

During training, the text encoder E and detector F_θ remain frozen. Since the paired textual descriptions are fixed, their base and fine-grained embeddings can be pre-computed and cached before training. At each iteration, TRP constructs the adapted textual representations $\widehat{\mathbf{T}}_\phi$ and feeds them into the frozen detector. We retain the original detection loss in Eq. 4 and update only the residual generator g_ϕ.

After training, the adapted textual embeddings for the current query vocabulary are computed once as

$$\widehat{\mathbf{T}}^* = \left\{\mathrm{TRP}_{\phi^*}(\mathbf{t}_i^b, \mathbf{t}_i^f)\right\}_{i=1}^{K}, \tag{8}$$

which forms the *adapted offline vocabulary*. Following the prompt-then-detect paradigm (Cheng et al., 2024), these residual-enhanced textual embeddings are cached and directly reused for online detection:

$$\widehat{Y} = F_\theta(x; \widehat{\mathbf{T}}^*). \tag{9}$$

Therefore, neither the text encoder nor the residual generator is executed during online detection. When the query vocabulary changes, its adapted textual embeddings are regenerated and cached offline, preserving the original detector architecture and inference efficiency.

4 Experiments

4.1 Experimental Settings

Dataset and evaluation metrics. We conduct all experiments on the Fine-Grained Open-Vocabulary Detection (FG-OVD) benchmark (Bianchi et al., 2024b), which is constructed from COCO images (Lin et al., 2014) to evaluate fine-grained recognition in open-vocabulary object detection. FG-OVD contains four difficulty-based benchmarks (*Hard*, *Medium*, *Easy*, and *Trivial*) and four attribute-based benchmarks (*Color*, *Material*, *Pattern*, and *Transparency*). We train all methods on the corresponding FG-OVD training sets and evaluate them on the official test sets.

Following the query-wise FG-OVD protocol, each vocabulary contains one target category and N hard negatives, with one detector inference per vocabulary. We pair each category with a coarse description extracted by a frozen OpenAssistant SFT LLaMA-30B model (Köpf et al., 2023) (Appendix A.1). All methods use 640×640 inputs; benchmark-specific values of N and other evaluation settings are given in Appendix B.4.

We use COCO-style $\mathrm{mAP}_{50:95}$ as the primary metric and additionally report the mean target-category rank. For each ground-truth object, candidate categories are scored by the maximum confidence among predictions with IoU ≥ 0.5, and the rank of the target category among the $N+1$ candidates is recorded. The rank is averaged over all objects, with lower values indicating better fine-grained discrimination. All methods use identical candidate sets, NMS, post-processing, and checkpoint-selection settings.

Base detectors. We evaluate TRP on YOLO-Worldv2-S (Cheng et al., 2024) and YOLOE-v8s (Wang et al., 2025). Both support offline textual vocabularies, allowing TRP to retain their *prompt-then-detect* inference paradigm.

Implementation details. We train for 20 epochs on four NVIDIA A100 GPUs, updating only the residual generator. YOLO-World uses SGD and YOLOE uses AdamW. Batch sizes, learning rates, and augmentation settings are detailed in Appendix B.3.

Compared methods. We compare TRP with conventional detector baselines and representative parameter efficient adaptation methods.

- **Detector fine-tuning.** *Zero-shot (ZS)* directly evaluates the pretrained base detector without FG-OVD adaptation. *Full-FT* optimizes the visual backbone, neck, and detection head while keeping the text encoder frozen, whereas *Head-FT* updates only the detection head.
- **Text-side parameter-efficient adaptation.** Inspired by TaskRes (Yu et al., 2023), *TPT* learns a single textual residual shared across all queried categories: $\widehat{\mathbf{t}}_i = \mathbf{t}_i + \mathbf{r}$. Inspired by CoCoOp (Zhou et al., 2022a), *TCR* uses an MLP to generate a residual from each frozen text embedding: $\widehat{\mathbf{t}}_i = \mathbf{t}_i + g_\psi(\mathbf{t}_i)$.
- **Visual-side parameter-efficient adaptation.** Following Medeiros et al. (Medeiros et al., 2025), *VPT* learns a shared, image-independent pixel prompt: $\widehat{x} = x + \mathbf{v}$. In contrast, *ModPrompt* employs a lightweight generator to produce an input-dependent visual prompt: $\widehat{x} = x + h_\vartheta(x)$. We implement its visual prompting component while keeping the detector frozen.

All methods use the same training data, detection objective, training schedule, and optimization settings. For TPT, TCR, VPT, ModPrompt, and TRP, all pretrained detector parameters are frozen and only the corresponding adaptation module is optimized.

4.2 Comparison with Adaptation Methods

We compare TRP with detector-level fine-tuning and parameter-efficient adaptation methods using YOLO-World as the base detector. We first report performance averaged over all valid hard-negative settings, then examine how performance changes as the number of negative descriptions increases.

Overall Performance. Tables 1 and 2 compare detection accuracy and target category ranking on YOLO-World. While keeping the detector frozen, TRP achieves an overall mAP of 56.2%, compared with 55.6% for Head-FT and 54.6% for Full-FT. Its post-NMS mean rank is also lower, at 1.7 versus 2.2 and 2.1, respectively. Thus, TRP reaches the detection accuracy of detector fine-tuning while showing a clearer advantage in selecting the correct fine-grained description among candidate categories.

Among the evaluated parameter-efficient adaptation methods, TRP achieves the highest overall mAP and the lowest post-NMS mean rank. In this comparison, text-side methods generally outperform visual-side prompting methods, suggesting that adapting textual category representations is effective when queries contain hard negatives. TCR is the closest baseline, but TRP exceeds it by 1.8 mAP points on the difficulty-based benchmarks and 2.6 points on the attribute-based benchmarks, while also achieving lower target-category ranks. Its larger advantage on the attribute-based benchmarks is consistent with TRP's focus on fine-grained attribute discrimination.

Performance Trends under Increasing Hard Negatives. Figure 4 shows how mAP changes as the number of hard-negative descriptions increases. Adding more candidates generally makes fine-

Table 1: YOLO-World results on FG-OVD ($\mathrm{mAP\text{-}Avg}_N$, %). Avg. is the macro-average within each benchmark group. Higher is better.

		Difficulty-based benchmarks					Attribute-based benchmarks				
Category	Method	Hard	Medium	Easy	Trivial	Avg.	Color	Material	Pattern	Transparency	Avg.
	Zero-shot	3.4	4.7	5.0	8.1	5.3	4.5	3.5	4.3	6.2	4.6
Detector fine-tuning	Full-FT	49.2	56.4	59.7	56.8	55.5	**48.8**	52.7	52.9	60.4	53.7
	Head-FT	48.1	56.8	**60.1**	**60.0**	56.3	47.1	53.3	**55.1**	64.4	55.0
Visual-side adaptation	VPT	12.3	17.2	20.4	31.1	20.2	18.1	10.8	10.9	16.2	14.0
	ModPrompt	10.9	15.9	17.8	27.8	18.1	15.7	11.2	13.2	11.6	12.9
Text-side adaptation	TPT	25.3	37.0	41.1	56.6	40.0	35.9	23.9	27.0	32.7	29.9
	TCR	46.7	55.5	59.1	59.0	55.1	43.5	51.9	53.9	62.9	53.0
	TRP	**50.6**	**57.3**	60.0	59.5	**56.9**	**48.8**	**54.2**	54.8	**64.5**	**55.6**

Table 2: YOLO-World post-NMS mean rank on FG-OVD. Avg. is the macro-average within each benchmark group. Lower is better. Bold values indicate the best results based on unrounded scores.

		Difficulty-based benchmarks					Attribute-based benchmarks				
Category	Method	Hard	Medium	Easy	Trivial	Avg.	Color	Material	Pattern	Transparency	Avg.
	Zero-shot	4.8	4.8	4.7	4.5	4.7	4.8	4.7	3.7	1.8	3.8
Detector fine-tuning	Full-FT	2.6	1.9	1.6	2.9	2.2	2.7	2.3	1.8	1.1	2.0
	Head-FT	2.8	2.2	1.7	2.7	2.4	2.9	2.5	1.7	1.1	2.0
Visual-side adaptation	VPT	4.0	3.8	3.4	3.4	3.7	3.8	3.8	2.9	1.6	3.0
	ModPrompt	4.0	3.9	3.6	3.6	3.8	3.9	3.8	2.9	1.5	3.0
Text-side adaptation	TPT	3.5	3.2	2.8	3.0	3.1	3.2	3.5	2.6	1.4	2.7
	TCR	2.3	1.6	1.3	2.4	1.9	2.5	1.7	1.5	1.1	1.7
	TRP	**2.1**	**1.5**	**1.2**	**2.3**	**1.8**	**2.4**	**1.6**	**1.3**	**1.1**	**1.6**

grained detection harder, yet TRP remains competitive with Full-FT and Head-FT across the evaluated settings while keeping the detector frozen. The comparison with TCR is particularly informative. On *Color*, TRP maintains higher mAP throughout the evaluated range of negative counts, rather than benefiting from only a particular query size. Its average gains over TCR are also more pronounced on *Hard* and *Color*, at 3.9 and 5.3 mAP points, respectively, than on the easier benchmarks. These results suggest that conditioning the residual on the base-to-fine textual displacement is especially useful when the candidate descriptions are difficult to distinguish.

4.3 TRANSFER TO YOLOE

We further apply TRP to YOLOE-v8s to assess its applicability beyond YOLO-World. Both versions of YOLOE are evaluated under the same FG-OVD protocol. As shown in Table 3, TRP raises the overall mAP across eight benchmarks from 38.1% to 51.3% and lowers the post-NMS mean rank from 2.7 to 2.1. Both metrics improve on seven benchmarks, with particularly large mAP gains on Hard (+18.4 points), Material (+22.9 points), and Pattern (+22.5 points).

Trivial is the sole exception: its mAP decreases from 59.7% to 54.3%, while mean rank rises from 1.6 to 1.9. In FG-OVD, the positive and negative descriptions in this split do not differ in fine-grained attributes (Bianchi et al., 2024b). The substantial gains on the other benchmarks show that TRP improves fine-grained discrimination when applied to YOLOE, extending its effectiveness beyond YOLO-World.

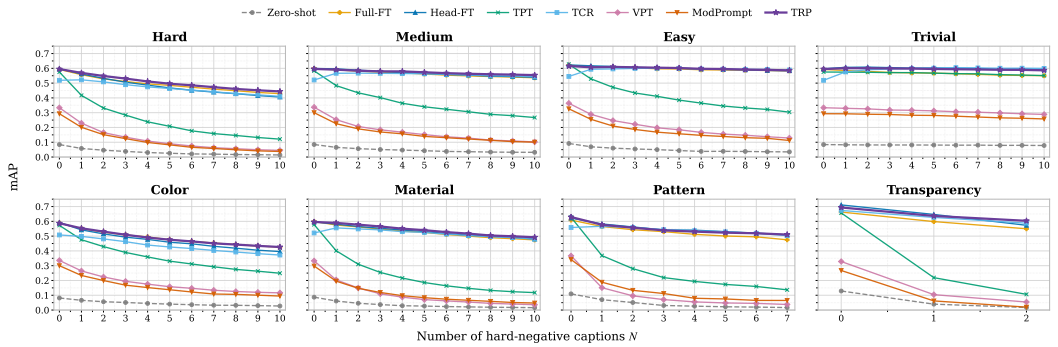


Figure 4: mAP as the number of hard-negative captions N increases across the eight FG-OVD benchmarks. Higher is better. Pattern and Transparency contain up to 7 and 2 negatives, respectively.

Table 3: Transfer to YOLOE-v8s on FG-OVD. Left: $\mathrm{mAP\text{-}Avg}_N$ (%); right: post-NMS mean rank. Parentheses show mAP gains in points or rank reductions relative to YOLOE; positive values indicate improvement.

	mAP (%)					Mean rank				
Method	Hard	Medium	Easy	Trivial	Avg.	Hard	Medium	Easy	Trivial	Avg.
YOLOE	28.1	40.3	40.0	**59.7**	42.0	3.6	2.7	2.7	**1.6**	2.6
YOLOE + TRP	**46.5** (+18.4)	**53.1** (+12.8)	**52.2** (+12.2)	54.3 (-5.4)	**51.5** (+9.5)	**2.4** (+1.2)	**2.0** (+0.7)	**2.2** (+0.5)	1.9 (-0.3)	**2.1** (+0.5)
Method	Color	Material	Pattern	Transparency	Avg.	Color	Material	Pattern	Transparency	Avg.
YOLOE	38.2	27.2	27.5	43.6	34.1	2.8	3.7	3.0	1.6	2.8
YOLOE + TRP	**44.8** (+6.6)	**50.1** (+22.9)	**50.0** (+22.5)	**59.5** (+15.9)	**51.1** (+17.0)	**2.5** (+0.3)	**2.2** (+1.5)	**2.2** (+0.8)	**1.2** (+0.4)	**2.0** (+0.8)

4.4 Parameter-Efficient and Non-Destructive Adaptation

As shown in Table 4, TRP updates only 0.328M parameters, accounting for 2.51% of the complete model. This is 7.1× fewer trainable parameters than Head-FT and 38.9× fewer than Full-FT. Nevertheless, TRP achieves 56.2% overall FG-OVD mAP, compared with 55.6% for Head-FT and 54.6% for Full-FT. Thus, adapting textual representations with a small number of parameters reaches the performance level of directly fine-tuning the detector. TRP also keeps the pretrained detector weights unchanged and stores task-specific adaptation in a separate textual adapter. By disabling the adapter and restoring the original vocabulary, it recovers 37.3% COCO mAP without reloading detector weights; the fine-grained Head-FT and Full-FT checkpoints obtain only 2.6% and 2.5% when evaluated directly on COCO. This result demonstrates reversible task switching, rather than COCO performance with the adapter enabled. In terms of training cost, TRP takes 163.2 ms per forward–backward step, 26.8% less than Full-FT but 17.2% more than Head-FT. Head-FT is faster because its backward computation is confined to the detection head, whereas updating TRP requires gradients to pass through subsequent frozen detector operations before reaching the textual adapter.

4.5 Ablation Studies

Effect of residual conditioning. We compare three textual residual formulations using the same base detector, training data, detection loss, and optimization settings. TPT learns a residual shared across all categories. TCR generates a residual from the fine-grained text embedding $\mathbf{t}^f$, whereas TRP conditions the residual generator on the coarse–fine displacement $\mathbf{t}^f - \mathbf{t}^b$.

Table 5 shows that replacing TPT's shared residual with a text-conditioned residual raises overall mAP from 34.9% to 54.1%. Conditioning on the coarse–fine displacement further improves mAP to 56.2%, exceeding TCR by 2.1 points. This gain supports the use of the relationship between base and fine-grained descriptions as the conditioning signal, rather than the fine-grained embedding alone.

Table 4: Parameter efficiency and task switching on YOLO-World. Ratio is the percentage of model parameters trained; F+B is the forward–backward time per step. FG-OVD mAP is averaged over all eight benchmarks. COCO mAP is measured after restoring the original vocabulary: TRP is disabled, while Head-FT and Full-FT retain their fine-tuned weights.

Method	Updated component	Trainable params (M)	Model ratio (%)	F+B time (ms/step)	FG-OVD mAP (%)	COCO mAP (%)
TRP	Textual adapter	**0.328**	**2.51**	163.2	**56.2**	**37.3**
Head-FT	Detection head	2.317	17.70	**139.2**	55.6	2.6
Full-FT	Backbone and head	12.760	97.48	223.1	54.6	2.5

Table 5: Effect of residual conditioning. Overall mAP is $\mathrm{mAP\text{-}Avg}_N$ across eight FG-OVD benchmarks; Δ TPT is measured in mAP points.

Method	Condition	Overall mAP (%)	Δ TPT
TPT	Shared	34.9	0.0
TCR	$\mathbf{t}^f$	54.1	+19.2
TRP	$\mathbf{t}^f - \mathbf{t}^b$	**56.2**	**+21.3**

Table 6: Inference speed on one A100 GPU with cached query features. Latency includes preprocessing, detector forward propagation, and NMS.

Method	ms/img	FPS
YOLO-Worldv2-S	8.71	114.8
YOLO-Worldv2-S + TRP	8.78	113.9
YOLOE-v8s	6.53	153.1
YOLOE-v8s + TRP	6.64	150.5

Effect of residual generator capacity. We compare the current four-layer residual generator with a linear projection and a shallower two-layer MLP. All variants use $\mathbf{t}^f - \mathbf{t}^b$ as input, with the same residual addition, feature normalization, training data, and optimization settings.

Table 7: Effect of residual generator architecture. All variants are conditioned on $\mathbf{t}^f - \mathbf{t}^b$. Overall mAP is averaged over the eight FG-OVD benchmarks.

Generator	Architecture	Trainable params (M)	Overall mAP (%)
Linear	$512 \to 512$	0.263	56.1
Shallow MLP	$512 \to 256 \to 512$	0.263	56.0
Current MLP	$512 \to 256 \to 128 \to 256 \to 512$	**0.328**	**56.2**

As shown in Table 7, the three generators achieve similar performance. The current MLP exceeds the linear projection and shallow MLP by only 0.1 and 0.2 mAP points, respectively. Within the evaluated range, TRP is therefore relatively insensitive to generator depth and parameter count. Together with the conditioning ablation, this result indicates that the choice of conditioning signal has a greater effect than increasing generator capacity in these experiments.

4.6 Real-Time Open-Vocabulary Detection with TRP

TRP preserves the prompt-then-detect paradigm by adapting and caching query features before image inference. As shown in Table 6, it adds only 0.07 and 0.11 ms per image to YOLO-Worldv2-S and YOLOE-v8s, respectively, while retaining 113.9 and 150.5 FPS on one A100 GPU. The timing protocol is detailed in Appendix B.5.

5 Conclusion

We observe consistent attribute-induced directions across base categories in the textual embedding space and propose Textual Residual Prompt (TRP), a simple and efficient method for real-time open-vocabulary detection. TRP achieves strong parameter-efficient adaptation through lightweight textual residual learning without additional online text-side computation.

AI USE STATEMENT

We used OpenAI Codex to assist with drafting and organizing the appendix, editing its language, and aligning its terminology and notation with the main text. Codex also assisted with inspecting the existing training and evaluation scripts to check the appendix's descriptions of reward functions, model components, training configurations, and evaluation protocols, and with preparing numerical checks of the reward functions. The authors take responsibility for the final content of this work, including text, claims, and artifacts prepared with AI assistance.

ETHICS STATEMENT

This work uses the existing FG-OVD benchmark, which is built from COCO images, and does not involve human-subject experiments. TRP may inherit biases from pretrained models and datasets, and may produce false detections or miss objects outside the evaluated settings. Its reliability should therefore be assessed before use in safety-sensitive applications, with appropriate human oversight. This work adheres to the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

The method and experimental settings are described in the main text. The appendix provides the coarse–fine pair construction (Appendix A.1), training configuration (Appendix B.3), evaluation protocol (Appendix B.4), and inference timing protocol (Appendix B.5).

REFERENCES

Lorenzo Bianchi, Fabio Carrara, Nicola Messina, and Fabrizio Falchi. Is clip the main roadblock for fine-grained open-world perception? In *2024 International Conference on Content-Based Multimedia Indexing (CBMI)*, pp. 1–8. IEEE, 2024a.

Lorenzo Bianchi, Fabio Carrara, Nicola Messina, Claudio Gennaro, and Fabrizio Falchi. The devil is in the fine-grained details: Evaluating open-vocabulary object detectors for fine-grained understanding. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22520–22529. IEEE, 2024b.

Adrian Bulat and Georgios Tzimiropoulos. Lasp: Text-to-text optimization for language-aware soft prompting of vision & language models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23232–23241. IEEE, 2023.

Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world: Real-time open-vocabulary object detection. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16901–16911. IEEE, 2024.

Yu Du, Fangyun Wei, Zihe Zhang, Miaojing Shi, Yue Gao, and Guoqi Li. Learning to prompt for open-vocabulary object detection with vision-language model. In *2022 IEEE/CVF IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14064–14073. IEEE, 2022.

Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision*, 132(2):581–595, 2024.

Sheng Jin, Xueying Jiang, Jiaxing Huang, Lewei Lu, and Shijian Lu. Llms meet vlms: Boost open vocabulary object detection with fine-grained descriptors. In *International Conference on Learning Representations*, volume 2024, pp. 8734–8745, 2024.

Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19113–19122. IEEE, 2023a.

Muhammad Uzair Khattak, Syed Talal Wasim, Muzammal Naseer, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Self-regulating prompts: Foundational model adaptation without forgetting. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15144–15154. IEEE, 2023b.

Andreas Köpf, Yannic Kilcher, Dimitri Von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, et al. Openassistant conversations-democratizing large language model alignment. *Advances in Neural Information Processing Systems*, 36:47669–47681, 2023.

Weicheng Kuo, Yin Cui, Xiuye Gu, AJ Piergiovanni, and Anelia Angelova. F-vlm: Open-vocabulary object detection upon frozen vision and language models. In *International Conference on Learning Representations*, 2023.

Jiaming Li, Zhijia Liang, Weikai Chen, Lin Ma, and Guanbin Li. Guided: Granular understanding via identification, detection, and discrimination for fine-grained open-vocabulary object detection. *Advances in Neural Information Processing Systems*, 38:41320–41339, 2026.

Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10955–10965. IEEE, 2022.

Xin Li, Dongze Lian, Zhihe Lu, Jiawang Bai, Zhibo Chen, and Xinchao Wang. Graphadapter: Tuning vision-language models with dual knowledge graph. *Advances in Neural Information Processing Systems*, 36:13448–13466, 2023.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pp. 740–755. Springer, 2014.

Mingxuan Liu, Tyler L Hayes, Elisa Ricci, Gabriela Csurka, and Riccardo Volpi. Shine: Semantic hierarchy nexus for open-vocabulary object detection. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16634–16644. IEEE, 2024a.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pp. 38–55. Springer, 2024b.

Ying Liu, Yijing Hua, Haojiang Chai, Yanbo Wang, and TengQi Ye. Fine-grained open-vocabulary object detection with fined-grained prompts: Task, dataset and benchmark. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13860–13867. IEEE, 2025.

Heitor R Medeiros, Atif Belal, Srikanth Muralidharan, Eric Granger, and Marco Pedersoli. Visual modality prompt for adapting vision-language object detectors. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2172–2182. IEEE, 2025.

Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, et al. Simple open-vocabulary object detection. In *European Conference on Computer Vision*, pp. 728–755. Springer, 2022.

Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. *Advances in Neural Information Processing Systems*, 36:72983–73007, 2023.

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

Ao Wang, Lihao Liu, Hui Chen, Zijia Lin, Jungong Han, and Guiguang Ding. Yoloe: Real-time seeing anythi. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 24591–24602. IEEE, 2025.

Xiaoshi Wu, Feng Zhu, Rui Zhao, and Hongsheng Li. Cora: Adapting clip for open-vocabulary detection with region prompting and anchor pre-matching. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7031–7040. IEEE, 2023.

Hantao Yao, Rui Zhang, and Changsheng Xu. Visual-language prompt tuning with knowledge-guided context optimization. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6757–6767. IEEE, 2023.

Hantao Yao, Rui Zhang, and Changsheng Xu. Tcp: Textual-based class-aware prompt tuning for visual-language model. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23438–23448. IEEE, 2024.

Lewei Yao, Jianhua Han, Youpeng Wen, Xiaodan Liang, Dan Xu, Wei Zhang, Zhenguo Li, Chunjing Xu, and Hang Xu. Detclip: Dictionary-enriched visual-concept paralleled pre-training for open-world detection. *Advances in Neural Information Processing Systems*, 35:9125–9138, 2022.

Tao Yu, Zhihe Lu, Xin Jin, Zhibo Chen, and Xinchao Wang. Task residual for tuning vision-language models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10899–10909. IEEE, 2023.

Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? *International Conference on Learning Representations*, 2023.

Tiancheng Zhao, Peng Liu, Xuan He, Lu Zhang, and Kyusong Lee. Real-time transformer-based open-vocabulary detection with efficient fusion head. *arXiv preprint arXiv:2403.06892*, 2024.

Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, et al. Regionclip: Region-based language-image pretraining. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16772–16782. IEEE, 2022.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16795–16804. IEEE, 2022a.

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, 2022b.

Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *European Conference on Computer Vision*, pp. 350–368. Springer, 2022c.

Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. Prompt-aligned gradient for prompt tuning. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15613–15623. IEEE, 2023.

A Supplementary Method Details

A.1 Constructing Coarse-Grained Descriptions

Before training, we construct a coarse–fine pair for each FG-OVD category. A frozen OpenAssistant SFT LLaMA-30B model is prompted to extract the main object noun from the category name, using only words already present and omitting attribute modifiers. We use greedy decoding. For example, "a dark green fabric pillow" is paired with "pillow". The extracted descriptions retain the original category identities, allowing both target and hard-negative queries to be paired with their corresponding coarse descriptions. Only category names are provided to the extractor; images and box annotations are not used for this step. At inference, the implementation accepts an explicit coarse description for each fine-grained query. This can be supplied by the user or produced by an optional upstream noun-phrase extractor before the query features are cached.

B Supplementary Experimental Details

B.1 Comparison with Other Open-Vocabulary Detectors

Table 8 gives the eight benchmark values plotted in Fig. 1. The reference-detector entries are the comparison values used to construct that figure; they provide context for the locally evaluated YOLO-World rows and were not rerun under our implementation. On these values, YOLO-World with TRP is the highest-scoring method on six of the eight benchmarks. The exceptions are *Trivial* and *Color*; the former contains shuffled, relatively easy-to-distinguish negatives and does not isolate subtle attribute discrimination. The table supports a benchmark-level comparison, rather than a controlled comparison of model size or runtime.

Table 8: Values underlying the radar plot in Fig. 1: FG-OVD $\mathrm{mAP\text{-}Avg}_N$ (%). Reference-detector scores are the supplied comparison values used in that plot; YOLO-World rows are our evaluations. Higher is better.

Method	Hard	Medium	Easy	Trivial	Color	Material	Pattern	Transp.
OWL (B/16)	26.2	39.8	38.4	53.9	45.3	37.3	26.6	34.1
OWL (L/14)	26.5	39.3	44.0	65.1	43.8	44.9	36.0	29.2
OWLv2 (B/16)	25.3	38.5	40.0	52.9	45.1	33.5	19.2	28.5
OWLv2 (L/14)	25.4	41.2	42.8	63.2	**53.3**	36.9	23.3	12.2
Detic	11.5	18.6	18.6	**69.7**	21.5	38.8	30.1	28.0
ViLD	22.1	36.1	39.9	56.6	43.2	34.9	24.5	30.1
Grounding DINO	16.6	27.9	30.1	62.7	41.0	30.2	31.2	25.4
CORA	13.8	20.0	20.4	35.1	25.0	19.3	22.0	27.9
YOLO-Worldv2-S (zero-shot)	3.4	4.7	5.0	8.1	4.5	3.5	4.3	6.2
YOLO-Worldv2-S + TRP	**50.6**	**57.3**	**60.0**	59.5	48.8	**54.2**	**54.8**	**64.5**

B.2 Mean-Rank Trends with More Hard Negatives

Figure 5 complements the mAP curves in Fig. 4 with the post-NMS mean-rank proxy for the same eight FG-OVD benchmarks. The rank of the target description is computed from the surviving detections whose boxes overlap a ground-truth object by at least 0.5 IoU; the highest confidence for each candidate is used, and a missing candidate receives zero. Because only the detector's surviving top-class predictions are retained, this is a post-NMS proxy rather than a ranking of all raw class logits. Lower values indicate better selection of the target description.

At $N = 0$ the only candidate has rank one. As candidates are added, TRP's rank generally rises much more slowly than the zero-shot YOLO-World rank. For example, at $N = 10$ the *Hard* ranks are 3.57 and 8.70, respectively, and the *Material* ranks are 2.37 and 8.52. TCR remains a close text-side baseline, so the figure should be read alongside the mAP results rather than as an independent detection accuracy metric. The same valid N ranges as in the main text are used: 0–10 except *Pattern* (0–7) and *Transparency* (0–2).

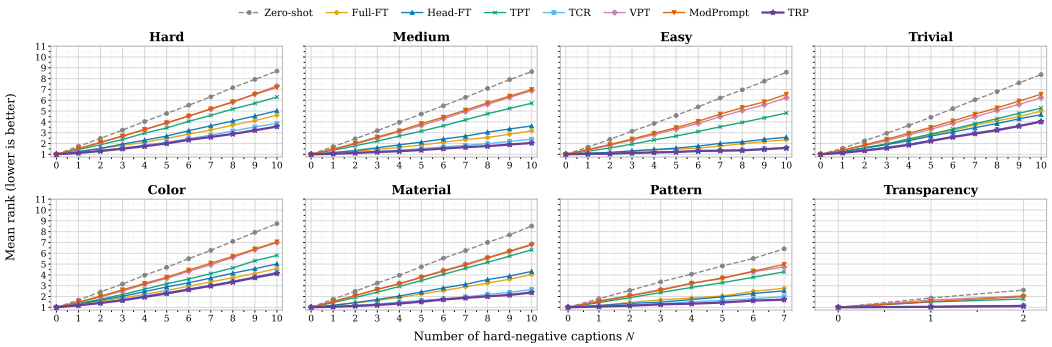


Figure 5: Post-NMS mean target-category rank as the number of hard-negative descriptions increases. The panels and method labels match Fig. 4; lower is better. The reported YOLO outputs yield a post-NMS proxy.

Table 9: Logged TRP training settings. Batch is the global batch over four GPUs; nominal batch controls gradient accumulation and decay scaling.

Setting	YOLO-Worldv2-S	YOLOE-v8s
Trainable parameters	0.328M	0.329M
Epochs / seed	20 / 0	20 / 0
Global batch (per GPU)	128 (32)	32 (8)
Nominal batch	64	128
Optimizer	SGD (auto-selected)	AdamW
Initial learning rate	0.01	0.001
Weight decay (configured)	5×10^{-4}	5×10^{-4}
Warm-up / final LR factor	3 epochs / 0.01	3 epochs / 0.01
Mixed precision	on	off
Data workers per process	8	2

B.3 Training Configuration

Both TRP variants are trained on the eight FG-OVD training subsets built from COCO training images, using four NVIDIA A100-SXM4-80GB GPUs. The logged runs use 20 epochs, and 640×640 input. Only the shared residual generator is optimized; pretrained detector and text-encoder weights remain frozen. YOLOE additionally keeps frozen detector batch-normalization statistics in evaluation mode. The 512-dimensional generator has layer widths $512 \to 256 \to 128 \to 256 \to 512$ with ReLU between linear layers. We use the last-epoch checkpoint for evaluation, without choosing an epoch from the FG-OVD test results. Table 9 records the detector-specific training settings. YOLO-World uses no gradient accumulation, whereas YOLOE accumulates four global batches before an optimizer step. The trainer scales the configured weight decay by global batch times accumulation divided by nominal batch, yielding effective values of 10^{-3} and 5×10^{-4}, respectively.

The active training augmentations include random translation and scaling, HSV color jitter, and horizontal flips.

B.4 Evaluation and Inference Configuration

Each FG-OVD query contains one target and N negative descriptions. For mAP and mean-rank aggregation, we evaluate every valid N: 0–10 for six benchmarks, 0–7 for *Pattern*, and 0–2 for *Transparency*. Per-benchmark results average over those N values; overall values then macro-average the eight benchmarks. Detection uses a 640-pixel image-size setting, confidence threshold 0.001, detector NMS IoU threshold 0.7, and at most 100 predictions per query. We report COCO-style $\mathrm{mAP}_{50:95}$; mean rank uses 0.5 IoU for matching a surviving detection to a ground-truth box. The

candidate vocabulary, including the coarse pair of every negative description, is constructed in the same order for both the baseline and TRP detector.

B.5 Inference Efficiency Protocol

For the speed comparison in Table 6, we use one A100-SXM4-80GB GPU, batch size one, FP32, and the same 64 FG-OVD images and 11 descriptions ($N = 10$) for all four methods. Input images are already decoded in memory. Text encoding and TRP adaptation are completed once before timing, and the resulting features are cached. The timed path covers image preprocessing, detector forward propagation, and NMS, with disk I/O excluded. The benchmark uses a 640-pixel letterboxed input, confidence threshold 0.001, NMS IoU threshold 0.7, and a 100-detection limit. After 20 warm-up images, we time three passes through the image set with GPU synchronization around each image.