CATEGORIES AS PROTOTYPE SETS: CONSTRUCTING TEXT PROTOTYPES FROM DESCRIPTIONS FOR OPEN-VOCABULARY OBJECT DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Open-vocabulary object detection uses text representations to recognize categories that have no bounding-box annotations during training. Existing approaches often represent each category with a class-name embedding or a fixed average of description embeddings, limiting how descriptions contribute to region-level recognition. We propose ProtoSet-DETR, a DETR-based detector that learns multiple text prototypes per category from a fixed set of offline-generated descriptions. A shared Text Prototype Aggregator constructs the prototypes from frozen CLIP description embeddings using attention and a learned value projection. The detector uses this bank for proposal scoring, decoder query initialization, and classification, weighting prototype contributions according to each visual region. Adaptive Prototype Regularization encourages diversity and balanced usage, while a training-only region–prototype alignment objective connects the bank to clustered encoder features. The same aggregator constructs prototypes for novel categories. Our diagnostics show concentrated description attention and reduced redundancy among the projected embeddings. On LVIS, ProtoSet-DETR achieves 46.0 rare-category AP (AP_r) and 46.1 AP, improving over LaMI-DETR with the same backbone by 2.6 AP_r and 4.8 AP. Without fine-tuning, the LVIS-trained model transfers to COCO and Objects365 with 46.3 and 25.4 AP. These results support learning how category descriptions are represented and matched to regions. Code will be released here.

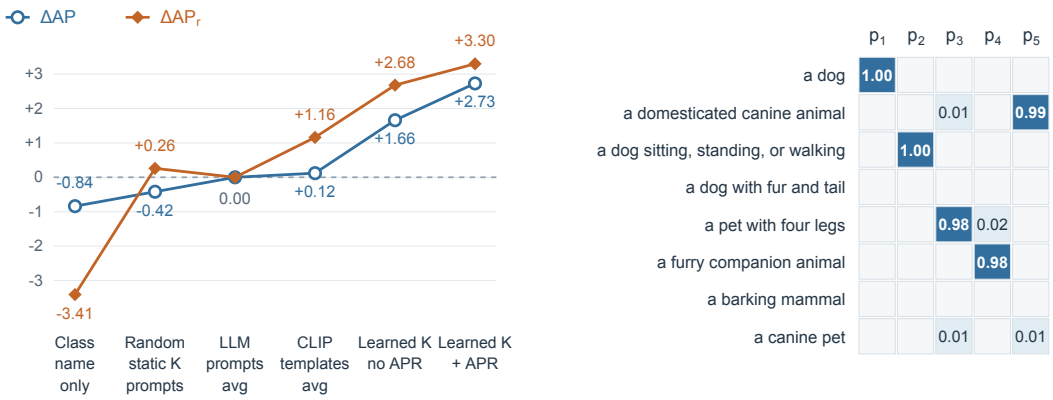
1 INTRODUCTION

Open-vocabulary object detection (OVD) aims to recognize and localize categories without their bounding-box annotations during training (Zareian et al., 2021; Gu et al., 2022; Minderer et al., 2022). Pre-trained vision–language models such as CLIP (Radford et al., 2021) provide transferable features for matching visual content with category text. Prompt learning (Du et al., 2022) and feature adapters (Gao et al., 2024) adapt these representations to downstream tasks. LaMI-DETR (Du et al., 2024) enriches category text with language-model descriptions, using a single category embedding for detector classification and language-guided query initialization. Descriptions provide cues about appearance, state, and context beyond a category name, raising the question of how a detector should organize and use these cues for individual regions.

The relevant cues can vary across instances of the same category. A cup, for example, may have a visible handle, contain liquid, or show both cues. Averaging description embeddings combines these cues with fixed weights before any region is observed. Learning a single category vector can adapt this representation to detection, but still supplies only one reference for matching. A small set of prototypes instead allows the detector to weight separate references according to each region's visual evidence. With the same description bank, learned prototypes outperform fixed averaging in the one-epoch comparison in Fig. 1(a), motivating a closer study of how the detector represents these descriptions.

Learning such a representation for novel categories poses a further challenge: only base categories have bounding-box annotations during detector training. We therefore learn a prototype construction rule shared across categories. Supervision on base categories shapes how this rule selects and transforms descriptions, and the same rule constructs prototypes from novel-category descriptions at inference.

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.



(a) Learning category representations (b) Description selection (dog)

Figure 1: **(a)** Changes in AP and AP_r after one epoch from released LaMI-DETR weights, using detector scores only. The reference, *LLM prompts avg*, averages eight LLM descriptions. *Random static K prompts* and both learned variants use the same description bank with $K = 5$. APR denotes Adaptive Prototype Regularization. **(b)** Attention of five prototype slots over eight descriptions, including the fixed slot prior, after twelve epochs of training (Appendix B.2). Blank cells denote weights below 0.005.

We introduce **ProtoSet-DETR**, a DETR-based detector whose **Text Prototype Aggregator (TPA)** learns a small prototype bank for each category. TPA combines fixed CLIP description embeddings through attention and a learned value projection, with parameters shared across categories. The dog example in Fig. 1(b) illustrates how different descriptions can be retained in separate slots.

The bank is built independently of the image, while its use depends on the region. Region–prototype similarities determine the text contribution to decoder query initialization and the category scores for encoder proposals and decoder outputs. During training, prototype regularization discourages redundant directions and encourages balanced assignment of descriptions to slots. In addition, a region–prototype alignment objective ties the aggregated encoder features to the bank. The descriptions are generated offline, and the bank can be cached for inference.

Using the same ConvNeXt-L backbone and DINO detector as LaMI-DETR, ProtoSet-DETR achieves 46.0 AP_r and 46.1 AP after twelve epochs of LVIS training, exceeding the published LaMI-DETR results by 2.6 and 4.8 points. Without fine-tuning, it achieves 46.3 AP on COCO and 25.4 AP on Objects365, improving AP by 3.5 on each dataset. The full-training comparison in Fig. 3 further shows that increasing K_p from 1 to 5 raises AP_r from 42.5 to 46.0. Because the descriptions, the learnable projection, and the training and evaluation protocols are all held fixed, the improvement supports using multiple prototypes in this setting.

Our main contributions are as follows:

- We learn multiple text prototypes from fixed category descriptions through an aggregator shared across categories, enabling the same construction for novel categories.
- We integrate region-dependent prototype matching into query initialization and classification, with training objectives for prototype diversity and alignment to encoder features.
- We demonstrate improved novel-category detection on LVIS and transfer to COCO and Objects365, with controlled ablations examining the benefit of training with multiple prototypes.

2 Related Work

2.1 Open-vocabulary object detection

Open-vocabulary object detection (OVD) extends recognition and localization to categories without bounding-box annotations during detector training (Zareian et al., 2021; Gu et al., 2022). To generalize beyond box-supervised categories, detectors draw on knowledge from image–caption pairs (Zareian et al., 2021), pre-trained vision–language models (Gu et al., 2022), or image-level labels (Zhou et al., 2022c). A key challenge in transferring pre-trained vision–language models is adapting their features

to region recognition. F-VLM (Kuo et al., 2022) reuses frozen features within a detector, while CORA (Wu et al., 2023b) adapts the model through region prompting and anchor pre-matching. CLIPSelf (Wu et al., 2024) addresses region recognition at the feature level, using self-distillation to improve dense representations.

Another approach is to align regions with text during language–image pre-training. RegionCLIP (Zhong et al., 2022) learns region–text alignment, and GLIP (Li et al., 2022) unifies detection and phrase grounding during pre-training. To broaden the language supervision available for alignment, DetCLIP (Yao et al., 2022) introduces dictionary descriptions, while DetCLIPv2 (Yao et al., 2023) scales word–region alignment with detection, grounding, and image–text data. Language generation provides another form of supervision: LLMDet (Fu et al., 2025) co-trains a grounding detector with a language model, using image-level captions and region-level phrases as generation targets alongside the grounding objective.

Language representations also guide intermediate detection stages. Grounding DINO (Liu et al., 2024b) integrates language into feature fusion, query selection, and decoding. Our work focuses on the category representations used for classification and query initialization in a DETR-style detector built on frozen vision–language features.

2.2 Learning and using category representations

Category representations can be adapted at different stages of the vision–language pipeline. Prompt learning changes the encoder inputs: CoOp (Zhou et al., 2022b) learns continuous text prompts, CoCoOp (Zhou et al., 2022a) conditions them on images, and MaPLe (Khattak et al., 2023) couples prompts across visual and language branches. In detection, PromptDet (Feng et al., 2022) and DetPro (Du et al., 2022) use region-level supervision to learn prompts, with DetPro averaging representations learned across foreground context levels. Adaptation can also operate after encoding: CLIP-Adapter (Gao et al., 2024) learns residual bottleneck transformations of visual features or text classifier weights after frozen CLIP encoders.

When a category has multiple prompts, their representations must also be combined or selected for recognition. ProDA (Lu et al., 2022) learns multiple continuous prompts and models their output classifier embeddings with a Gaussian distribution. It encourages diversity through an orthogonality loss and uses the distribution mean for inference in its experiments. Other approaches obtain this diversity from the text itself. SHiNe (Liu et al., 2024a) and Zhou & Chen (2025) aggregate hierarchy- or state-aware descriptions into a single vector per category without training. DeCo-DETR (Wang et al., 2026) instead clusters region descriptions into a centroid pool queried by a DETR detector. In these approaches, the prototypes are not optimized with the detection objective.

In detection, the resulting category representation can also guide query construction. Our closest reference, LaMI-DETR (Du et al., 2024), uses a single description embedding per category for detector classification and query fusion. It also refines descriptions to distinguish confusable categories in its VLM scoring branch at inference. ProtoSet-DETR retains multiple prototypes for classification and query initialization, allowing individual regions to match different prototypes of the same category. A category-agnostic selector and shared projection construct these prototypes from fixed description embeddings. Detection supervision, prototype regularization, and training-only region–prototype alignment jointly shape the bank, while the shared aggregator extends its construction to novel categories.

3 Method

3.1 Problem Setup and Overview

We consider open-vocabulary object detection: training uses box annotations from base categories $\mathcal{C}_{\text{base}}$, evaluation covers $\mathcal{C} = \mathcal{C}_{\text{base}} \cup \mathcal{C}_{\text{novel}}$, and textual descriptions are available for all categories. As shown in Fig. 2, ProtoSet-DETR builds on a two-stage DETR detector with DINO-style query selection (Carion et al., 2020; Zhang et al., 2022). A category-agnostic aggregator maps each category's fixed descriptions to a bank of K_p prototypes (Sec. 3.2.1). The detector uses this bank to score encoder proposals, initialize decoder queries, and classify decoder outputs, weighting the prototypes according to each region (Secs. 3.2.2–3.2.3). Two training-only objectives shape the bank: a prototype regularizer (Sec. 3.3) and a region–prototype alignment on clustered encoder features

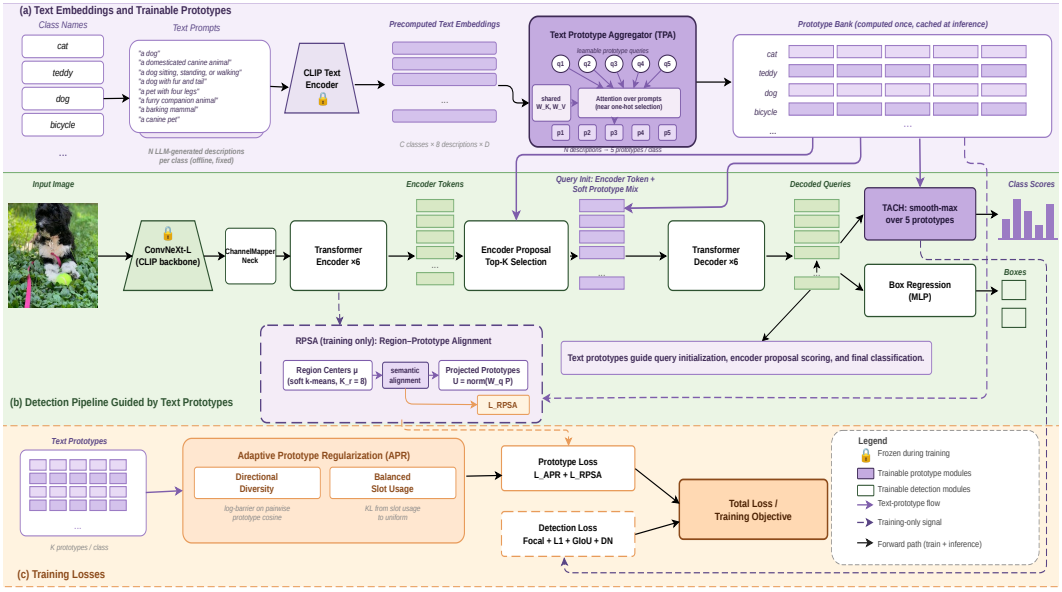


Figure 2: Overview of ProtoSet-DETR. (a) TPA builds five prototypes per category from eight fixed descriptions, encoded by a frozen CLIP text encoder and mapped through a shared learned projection; the bank is cached at inference. (b) TACH scores encoder proposals and decoder outputs by log-mean-exp pooling over prototypes, and the same bank guides query initialization; during training only, RPSA aligns clustered encoder tokens with the projected prototypes. (c) APR encourages directional diversity and balanced slot usage; the objective sums the detection loss with $\mathcal{L}_{\text{APR}}$ and $\mathcal{L}_{\text{RPSA}}$.

(Sec. 3.4). Because the aggregator is shared across categories, the construction applies unchanged to novel categories at inference.

3.2 Text-Prototype Interface for Detection

Following the federated protocol of Zhou et al. (2022c), each training iteration uses a sampled category subset $\mathcal{C}_t \subset \mathcal{C}_{\text{base}}$, with $C_t = |\mathcal{C}_t|$, that includes all categories annotated in the batch. Category-level computations during training use this subset. At inference, proposal scoring, query initialization, and classification use the full evaluation vocabulary $\mathcal{C}$.

3.2.1 Text Prototype Aggregation

For each category c, a frozen CLIP text encoder encodes N_p fixed descriptions generated offline by a language model. This yields unit-norm features $\{\mathbf{T}_{c,n}\}_{n=1}^{N_p}$, with $\mathbf{T}_{c,n} \in \mathbb{R}^d$. The description roles are summarized in Appendix A.1.

To construct the prototypes, we introduce K_p learnable slot queries $\mathbf{q}_k \in \mathbb{R}^{d_h}$, a key projection $W_K \in \mathbb{R}^{d_h \times d}$, and a value projection $W_V \in \mathbb{R}^{d \times d}$, all shared across categories. Each slot attends to the description features and aggregates their value projections:

$$a_{c,k,n} = \frac{\exp\left((\mathbf{q}_k^\top W_K \mathbf{T}_{c,n} + b_{k,n})/(\sqrt{d_h}\,\tau_p)\right)}{\sum_{n'=1}^{N_p} \exp\left((\mathbf{q}_k^\top W_K \mathbf{T}_{c,n'} + b_{k,n'})/(\sqrt{d_h}\,\tau_p)\right)}, \qquad \mathbf{P}_{c,k} = \sum_{n=1}^{N_p} a_{c,k,n}\, W_V \mathbf{T}_{c,n}. \quad (1)$$

Here, τ_p controls attention sharpness. The fixed slot prior $b_{k,n}$ gives each slot an initial preference for a predefined description position, breaking symmetry without fixing its eventual selection (Sec. 4.5). We initialize W_V to the identity so that the value projection initially preserves the CLIP text features. The prior construction and remaining initialization details are provided in Appendix A.1.

The attention weights determine the contribution of each description, while W_V adapts its representation to detection. The bank itself is independent of the image. The classification head below determines how its prototypes contribute to each region's score. We analyze the learned attention patterns and prototype geometry in Sec. 4.5.

3.2.2 TEXT-AWARE CLASSIFICATION HEAD

The Text-Aware Classification Head (TACH) uses the category prototypes to compute scores that depend on each visual region. Given a visual feature $\mathbf{h}_j$, we project it into the prototype space as $\mathbf{z}_j = \mathrm{norm}(W_t \mathbf{h}_j)$ and normalize each prototype as $\hat{\mathbf{P}}_{c,k} = \mathrm{norm}(\mathbf{P}_{c,k})$, where norm denotes ℓ_2 normalization. The logit for category c aggregates the region's cosine similarities to its prototypes through log-mean-exp pooling:

$$S_{j,c} = s\,\tau_{\mathrm{cls}} \log\Big(\frac{1}{K_p}\sum_{k=1}^{K_p} \exp\big(\mathbf{z}_j^\top \hat{\mathbf{P}}_{c,k}/\tau_{\mathrm{cls}}\big)\Big). \tag{2}$$

Here, s controls the logit scale, while τ_{cls} controls pooling sharpness. Pooling approaches the maximum similarity as $\tau_{\mathrm{cls}} \to 0$ and the mean similarity as $\tau_{\mathrm{cls}} \to \infty$. The normalization by K_p ensures that identical prototypes yield the same score as a single prototype. Both s and τ_{cls} are shared across categories, and the same formulation applies to base and novel categories. We use this scoring rule for encoder proposal scoring and decoder classification.

3.2.3 LANGUAGE-GUIDED QUERY FUSION

The encoder scores in Eq. (2) also determine which visual features initialize decoder queries. We rank encoder tokens by their maximum category score and select the top-scoring tokens as query seeds. For seed i, let $\mathbf{m}_i$ denote the selected encoder feature. Its predicted proposal box is detached and used as the initial reference box.

Following LaMI-DETR (Du et al., 2024), we combine each visual seed with a language representation to form the query content. Here, the language representation is drawn from the prototype bank. To accommodate ambiguity in each seed's category prediction, we retain the set $\mathcal{R}_i$ of the R categories with the highest classification scores and project the prototypes into the detector hidden space, $\mathbf{U}_{c,k} = \mathrm{norm}(W_q \mathbf{P}_{c,k})$. We weight the retained categories by $\gamma_{i,c} = \mathrm{softmax}_{c\in\mathcal{R}_i}\big(S_{i,c}/\tau_s\big)$. Within each category, prototype weights depend on similarity to the visual seed: $\alpha_{i,c,k} = \mathrm{softmax}_k\big(\cos(\mathbf{m}_i, \mathbf{U}_{c,k})/\tau_q\big)$. The temperatures τ_s and τ_q control these two weight distributions. The initial content query combines the visual seed with the resulting language term:

$$\mathbf{q}_i^0 = \mathrm{sg}(\mathbf{m}_i) + \sum_{c\in\mathcal{R}_i} \gamma_{i,c} \sum_{k=1}^{K_p} \alpha_{i,c,k}\,\mathbf{U}_{c,k}. \tag{3}$$

The stop-gradient operator sg blocks gradients through the visual residual only. The language term still propagates gradients to the encoder through the selected category logits and prototype similarities used to compute $\gamma_{i,c}$ and $\alpha_{i,c,k}$.

3.3 ADAPTIVE PROTOTYPE REGULARIZATION

Regularizer. We regularize the prototype bank to encourage complementary representations for classification and query initialization. The regularizer combines directional diversity with balanced slot usage:

$$\begin{aligned} \mathcal{L}_{\mathrm{APR}} &= \lambda_{\mathrm{div}}\mathcal{L}_{\mathrm{div}} + \lambda_{\mathrm{bal}}\mathcal{L}_{\mathrm{bal}}, \qquad \mathcal{L}_{\mathrm{bal}} = \frac{1}{C_t}\sum_{c\in\mathcal{C}_t} \mathrm{KL}\big(\bar{\boldsymbol{\pi}}_c \,\|\, \mathcal{U}(K_p)\big), \\ \mathcal{L}_{\mathrm{div}} &= -\frac{1}{C_t K_p(K_p-1)} \sum_{c\in\mathcal{C}_t}\sum_{k\neq m} \log\Big(1-(1-\epsilon)\big(\hat{\mathbf{P}}_{c,k}^\top \hat{\mathbf{P}}_{c,m}\big)^2\Big). \end{aligned} \tag{4}$$

$\mathcal{L}_{\mathrm{div}}$ is a log barrier on the squared cosine between prototypes of the same category; it penalizes redundant directions, with ϵ keeping the penalty finite. $\mathcal{L}_{\mathrm{bal}}$ is the KL divergence from the slot-usage distribution $\bar{\boldsymbol{\pi}}_c$ to the uniform distribution, where $\bar{\pi}_{c,k}$ is the share of the category's descriptions assigned to slot k by a softmax over slots of the temperature-scaled attention scores in Eq. (1) (Appendix A.2). Both weights remain constant throughout training, and both terms are defined as zero for $K_p = 1$.

Prototype formation. The detection objective does not require distinct prototypes, so task gradients on the aggregator can conflict with $\mathcal{L}_{\mathrm{APR}}$ early in training. Instability in learning to aggregate descriptions has also been reported by Zhou & Chen (2025). During the first four epochs, we apply a PCGrad-style projection (Yu et al., 2020) to remove the component of the aggregator's task gradient that opposes the regularizer. Ordinary joint optimization then resumes, with the regularizer still active. APR adapts the gradient through this per-step conflict test; Appendix A.2 gives the projection and its analysis.

3.4 Region–Prototype Semantic Alignment

APR regularizes relationships within the prototype bank. RPSA complements it with a training-only objective that aligns aggregated encoder features with the bank, supplementing the proposal-level supervision provided by the detection losses.

Visual centers and category targets. Let $\mathbf{x}_{b,n} \in \mathbb{R}^{d_e}$ denote the encoder token features of image b, and $S_{b,n,c}$ their category scores under Eq. (2). We obtain token-to-category weights $M_{b,n,c}$ by applying softmax over $\mathcal{C}_t$ and retaining each token's L largest probabilities. We select up to M_{tok} valid tokens, ranked by their maximum category weight, and group them into K_r visual centers $\boldsymbol{\mu}_{b,g}$ with one step of soft k-means. Farthest-point sampling supplies the initial centers $\boldsymbol{\mu}^0_{b,g}$. Responsibilities $r_{b,n,g}$ are a softmax over centers of $-\|\mathbf{x}_{b,n} - \boldsymbol{\mu}^0_{b,g}\|^2/\sigma^2$, where σ controls assignment softness. Each updated center is the responsibility-weighted mean of the selected encoder features (Appendix A.3). Each center then inherits category weights from the tokens it aggregates, $\pi_{b,g,c} = \sum_n r_{b,n,g} M_{b,n,c} / \sum_n r_{b,n,g}$, which we normalize over $\mathcal{C}_t$ to obtain a soft alignment target $\tilde{\pi}_{b,g,c}$, treated as fixed without gradient propagation. A confidence filter on the unnormalized $\pi_{b,g,c}$ removes low-confidence centers, yielding the retained set $\mathcal{V}$ for the current mini-batch; Appendix A.3 specifies the filtering rule.

Objective. Using the projected prototypes $\mathbf{U}_{c,k}$ introduced in Sec. 3.2.3, we align each retained center with the prototype bank through a weighted InfoNCE objective:

$$\mathcal{L}_{\mathrm{RPSA}} = -\frac{1}{|\mathcal{V}|} \sum_{(b,g)\in\mathcal{V}} \sum_{c\in\mathcal{C}_t} \tilde{\pi}_{b,g,c} \log \frac{\sum_{k=1}^{K_p} \exp\big(\cos(\boldsymbol{\mu}_{b,g}, \mathbf{U}_{c,k})/\tau_r\big)}{\sum_{c'\in\mathcal{C}_t} \sum_{k=1}^{K_p} \exp\big(\cos(\boldsymbol{\mu}_{b,g}, \mathbf{U}_{c',k})/\tau_r\big)}. \tag{5}$$

Here, τ_r is the alignment temperature. The numerator aggregates matching evidence across a category's prototypes, while the denominator includes all prototypes in the sampled category set. Within each category, better-matching prototypes contribute more to the aggregated score, following the soft aggregation principle used by TACH.

3.5 Training Objective and Inference

The training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{det}} + \lambda_{\mathrm{APR}}\mathcal{L}_{\mathrm{APR}} + \lambda_{\mathrm{RPSA}}\,\rho(t)\,\mathcal{L}_{\mathrm{RPSA}}, \qquad \rho(t) = \mathrm{clip}\big((t - T_0)/T_1,\, 0,\, 1\big). \tag{6}$$

Here, $\mathcal{L}_{\mathrm{det}}$ includes DINO's matched focal classification, ℓ_1/GIoU box regression, auxiliary encoder and decoder supervision, and denoising losses. The schedule $\rho(t)$ delays RPSA for T_0 iterations, then increases its weight linearly over T_1 iterations.

At inference, the prototype bank is computed once for the evaluation vocabulary and cached; no language-model call or clustering is needed. For comparison with prior work we follow the CLIP score ensemble of LaMI-DETR (Du et al., 2024), which fuses detector and frozen-CLIP region scores for novel categories; the formula, coefficients, and category scope are given in Sec. 4.1 and Appendix B.1.

4 Experiments

4.1 Experimental Setup

Datasets and metrics. We use LVIS (Gupta et al., 2019) as our main benchmark. Following prior work (Gu et al., 2022; Zhou et al., 2022c; Liu et al., 2024b), we divide its 1,203 categories into 866 base categories (common and frequent) and 337 novel categories (rare), withholding box annotations

Table 1: Comparison with open-vocabulary detectors on OV-LVIS. †: reports mask AP, following LaMI-DETR (Du et al., 2024).

Method	Pretrain	Backbone	AP_r	AP_c	AP_f	AP
ViLD (Gu et al., 2022)	ViT-B/32	RN50	16.7	26.5	34.2	27.8
BARON (Wu et al., 2023a)	ViT-B/32	RN50	23.2	29.3	32.5	29.5
DetPro (Du et al., 2022)	ViT-B/32	RN50	20.8	27.8	32.4	28.4
OV-DETR (Zang et al., 2022)	ViT-B/32	RN50	17.4†	25.0†	32.5†	26.6†
PromptDet (Feng et al., 2022)	ViT-B/32	RN50	21.4†	23.3†	29.3†	25.3†
OADP (Wang et al., 2023)	ViT-B/32	RN50	21.9	28.4	32.0	28.7
LBP (Li et al., 2024)	ViT-B/32	RN50	24.1	29.5	32.8	29.9
RegionCLIP (Zhong et al., 2022)	RN50x4	RN50x4	22.0†	32.1†	36.9†	32.3†
CORA (Wu et al., 2023b)	RN50x4	RN50x4	22.2	-	-	-
F-VLM (Kuo et al., 2022)	RN50x64	RN50x64	32.8†	35.4†	35.4†	34.9†
RO-ViT (Kim et al., 2023b)	ViT-L/16	ViT-L/16	33.6	-	-	36.2
CFM-ViT (Kim et al., 2023a)	ViT-L/16	ViT-L/16	35.6	-	-	38.5
OWL-ViT (Minderer et al., 2022)	ViT-L/14	ViT-L/14	25.6	-	-	34.7
OV-DQUO (Wang et al., 2025a)	ViT-L/14	ViT-L/14	39.3	-	-	-
OV-DQUO+DeCLIP (Wang et al., 2025b)	ViT-L/14	ViT-L/14	41.5	-	-	34.6
ExDet (Zhang et al., 2026)	ViT-L/14	ViT-L/14	40.1	37.0	38.2	38.0
LaMI-DETR (Du et al., 2024)	ConvNeXt-L	ConvNeXt-L	43.4	-	-	41.3
Ours	ConvNeXt-L	ConvNeXt-L	**46.0**	**44.4**	**48.0**	**46.1**

for the novel categories during training. We report box AP averaged over IoU thresholds from 0.5 to 0.95, both overall and separately for rare (AP_r), common (AP_c), and frequent (AP_f) categories. Rare-category AP is our primary measure of open-vocabulary performance.

For cross-dataset transfer, we evaluate the LVIS-trained detector on all categories of COCO (Lin et al., 2014) and Objects365 (Shao et al., 2019) without fine-tuning. We do not train on the OV-COCO base/novel split; Appendix B.1 discusses this distinction. For both datasets, we report AP, AP_{50}, and AP_{75}.

Implementation details. ProtoSet-DETR is built on DINO (Zhang et al., 2022) with a CLIP ConvNeXt-Large backbone (Liu et al., 2022; Radford et al., 2021). We freeze the backbone trunk and fine-tune its output normalization parameters, while training the transformer, aggregator, and detection heads from scratch. For each category, a frozen CLIP text encoder encodes eight descriptions generated offline. We use $K_p = 5$ textual prototypes and $K_r = 8$ visual centers by default.

We train for 12 epochs with an effective batch size of 32, using AdamW with a base learning rate and weight decay of 10^{-4}. APR gradient routing is applied during the first four epochs and disabled afterwards, and RPSA follows the activation schedule in Sec. 3.5. Appendix B.1 provides the complete training settings and hyperparameter-selection procedure.

Evaluation protocol. Benchmark comparisons, component ablations, and sensitivity experiments use models trained for 12 epochs and the full validation set of each dataset. Cross-dataset transfer uses the same LVIS-trained model.

For score ensembling, a frozen CLIP branch scores each region against fixed text embeddings. On LVIS, we combine these scores with detector scores for rare categories using $\beta = 0.3$ and $\eta = 3$; common and frequent categories keep the detector scores. We apply the same ensemble to all target categories on COCO and Objects365. Appendix B.1 gives the fusion formula.

4.2 Main Results on LVIS

ProtoSet-DETR achieves 46.0 AP_r and 46.1 AP on OV-LVIS (Table 1), improving on the published LaMI-DETR (Du et al., 2024) results by 2.6 and 4.8 points, respectively. LaMI-DETR is the closest comparison because it uses the same ConvNeXt-Large backbone and DINO detector. The other methods in the table differ in pre-training scale, architecture, or post-processing. The effects of individual components are evaluated separately in Sec. 4.4.

Table 2: Cross-dataset transfer from LVIS to the full validation sets of COCO and Objects365. ProtoSet-DETR is trained on LVIS for 12 epochs and transferred without fine-tuning, with CLIP score ensembling applied to all target categories ($\beta = 0.3$, $\eta = 3$).

Method	Backbone	Detector	COCO			Objects365		
			AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}
ViLD (Gu et al., 2022)	RN50	Faster R-CNN	36.6	55.6	39.8	11.8	18.2	12.6
BARON (Wu et al., 2023a)	RN50	Faster R-CNN	36.2	55.7	39.1	13.6	21.0	14.5
DetPro (Du et al., 2022)	RN50	Faster R-CNN	34.9	53.8	37.4	12.1	18.8	12.9
LBP (Li et al., 2024)	RN50	Faster R-CNN	36.8	56.5	39.8	14.3	21.8	15.1
CFM (Kim et al., 2023a)	ViT-L/16	Faster R-CNN	-	-	-	18.7	28.9	20.3
CLIPSelf (Wu et al., 2024)	ViT-L/14	Faster R-CNN	40.5	**63.8**	44.3	19.5	31.3	20.7
CoDet (Ma et al., 2023)	EVA02-L	Mask R-CNN	39.1	57.0	42.3	14.2	20.5	15.3
OV-DQUO (Wang et al., 2025a)	-	DINO	39.2	-	42.5	18.4	-	19.6
LaMI-DETR (Du et al., 2024)	ConvNeXt-L	DINO	42.8	57.6	46.9	21.9	30.0	23.5
Ours	ConvNeXt-L	DINO	**46.3**	62.8	**51.1**	**25.4**	**34.2**	**27.5**

Table 3: Effects of TPA, APR, and RPSA on LVIS box AP.

TPA	APR	RPSA	AP_r	AP_c	AP_f	AP
-	-	-	42.0	40.2	44.0	42.0
✓	-	-	43.4	41.2	45.0	43.1
✓	✓	-	44.3	42.0	45.8	43.9
-	-	✓	43.1	41.0	44.9	42.9
✓	✓	✓	**46.0**	**44.4**	**48.0**	**46.1**

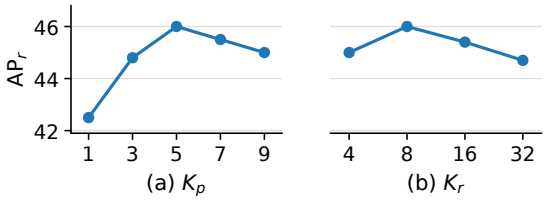


Figure 3: Sensitivity to (a) textual prototypes K_p and (b) RPSA visual centers K_r, measured by AP_r.

Both experiments use full LVIS validation after 12 epochs of training, following the protocol in Sec. 4.1.

4.3 Cross-Dataset Transfer

For transfer, we replace the LVIS prototype bank with one constructed from the target-category descriptions using the same aggregator. All learned parameters remain fixed, and evaluation follows the protocol in Sec. 4.1.

ProtoSet-DETR achieves 46.3 AP on COCO and 25.4 AP on Objects365 (Table 2), a 3.5-point improvement on each dataset over LaMI-DETR with the same ConvNeXt-L backbone and DINO detector. The gains extend to AP_{50} and AP_{75} on both datasets; AP_{75} improves by 4.2 points on COCO and 4.0 on Objects365.

4.4 Ablation Studies

Effect of individual components. We start from a detector that uses one CLIP class-name embedding per category (Table 3). With TPA, we learn prototypes from the category descriptions and use them for region-level matching, increasing AP_r from 42.0 to 43.4. We then add APR regularization, with gradient routing applied during the first four epochs, and obtain 44.3 AP_r. RPSA improves both the class-name baseline and the model with TPA and APR. With all three components, our method improves over the baseline by 4.0 AP_r and 4.1 AP, with gains on common and frequent categories as well.

Number of textual prototypes and visual centers. We next vary the number of prototypes learned from the same eight descriptions under the same training and evaluation protocol (Fig. 3(a)). Even with $K_p = 1$, we retain the learnable projection, so this model differs from the class-name baseline in Table 3. Increasing K_p from 1 to 5 raises AP_r from 42.5 to 46.0, while larger banks perform slightly worse. We also vary the number of visual centers and find that $K_r = 8$ gives the highest AP_r among the tested settings (Fig. 3(b)). We use $K_p = 5$ and $K_r = 8$ by default and provide the full sweep values in Appendix B.1.

4.5 Mechanism and Qualitative Analysis

How prototypes are learned. The shared value projection reduces redundancy among each category's eight description embeddings, increasing their effective rank from 2.20 to 4.07. Attention concentrates



Figure 4: Prototype-conditioned retrieval on full LVIS validation. Each row shows four contextual crops around ground-truth instances, ranked by relative prototype response. Row labels indicate the highest-attention description of each slot.



Figure 5: Post-hoc clustering visualization of encoder features on LVIS validation images. Top: inputs. Bottom: colored feature groups, not instance or part masks.

on individual descriptions (mean peak weight 0.964, including the fixed slot prior). With detector weights fixed, replacing the bank with all eight projected descriptions or the five-prototype centroid lowers AP across rare, common, and frequent categories on LVIS minival. These inference-time substitutions affect both classification and query initialization. Appendix B.2 gives the slot-prior analysis and full diagnostic results.

What prototypes respond to. Figure 4 shows ground-truth instances retrieved by relative prototype response. The *cup* rows highlight handles or liquid, matching the selected descriptions. For *dog*, the prototype associated with the pose description retrieves full-body animals outdoors, whereas the companion prototype retrieves mostly indoor scenes. The class-name row also includes plush toys and figurines. The *coat* rows contrast cold-weather jackets with formal overcoats. These examples suggest that prototype responses reflect both object appearance and context associated with the descriptions.

Encoder features after alignment. RPSA aligns aggregated encoder features with the prototype bank during training. Figure 5 visualizes these learned encoder features through post-hoc clustering, revealing spatially coherent groups around objects and their surroundings. Appendix B.3 explains the scope of this visualization.

5 Conclusion

We presented **ProtoSet-DETR**, which constructs multiple text prototypes from fixed category descriptions through an aggregator shared across categories. The shared construction applies to novel categories, while region-dependent matching determines how each prototype contributes to detection. Experiments on LVIS and transfer to COCO and Objects365 demonstrate improvements in rare-category recognition and cross-dataset detection. The diagnostics show how description selection and the shared projection shape the learned bank.

AI USE STATEMENT

Claude was used offline in October 2025 to generate the eight descriptions for each category in the fixed description bank (Appendix A.1); the bank was reviewed by the authors, and no language model is called during training or inference. AI coding assistants were used for implementation support and for editing the manuscript, including migration to the ICLR template. The authors were responsible for the research questions, method design, experiments, analyses, and final text, and verified all numbers and claims against the source code and experiment logs.

ETHICS STATEMENT

This study uses existing public object-detection benchmarks and does not involve new human-subject studies or crowdsourcing. Open-vocabulary detectors can support beneficial applications, but also raise concerns about surveillance, biased predictions, and downstream misuse. We discuss these broader impacts in Appendix D and the scope and limitations of our evaluation in Appendix C.

REPRODUCIBILITY STATEMENT

Section 3 describes the prototype construction, detector interface, regularization, and training objectives. Appendix A provides the description bank, slot prior and initialization, gradient-routing rule and schedule, and RPSA token selection and center filtering. The datasets and evaluation protocol are specified in Sec. 4.1. Appendix B.1 lists module and optimization hyperparameters, score-ensemble settings, and the parameter-selection protocol, while Sec. 4.4 reports component and hyperparameter ablations. Appendix C discusses the scope and limitations of the method. The full description bank is provided as supplementary material, and code and configuration files will be released upon acceptance.

REFERENCES

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

Penghui Du, Yu Wang, Yifan Sun, Luting Wang, Yue Liao, Gang Zhang, Errui Ding, Yan Wang, Jingdong Wang, and Si Liu. Lami-detr: Open-vocabulary detection with language model instruction. In *ECCV*, pp. 312–328. Springer, 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 *CVPR*, pp. 14084–14093, 2022.

Chengjian Feng, Yujie Zhong, Zequn Jie, Xiangxiang Chu, Haibing Ren, Xiaolin Wei, Weidi Xie, and Lin Ma. Promptdet: Towards open-vocabulary detection using uncurated images. In *ECCV*, pp. 701–717. Springer, 2022.

Shenghao Fu, Qize Yang, Qijie Mo, Junkai Yan, Xihan Wei, Jingke Meng, Xiaohua Xie, and Wei-Shi Zheng. Llmdet: Learning strong open-vocabulary object detectors under the supervision of large language models. In *CVPR*, pp. 14987–14997, 2025.

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. *IJCV*, 132:581–595, 2024. doi: 10.1007/s11263-023-01891-x.

Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. In *ICLR*, 2022. URL `https://openreview.net/forum?id=lL3lnMbR4WU`.

Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *CVPR*, pp. 5356–5364, 2019.

Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *CVPR*, pp. 19113–19122, 2023.

Dahun Kim, Anelia Angelova, and Weicheng Kuo. Contrastive feature masking open-vocabulary vision transformer. In *ICCV*, pp. 15602–15612, 2023a.

Dahun Kim, Anelia Angelova, and Weicheng Kuo. Region-aware pretraining for open-vocabulary object detection with vision transformers. In *CVPR*, pp. 11144–11154, 2023b.

Weicheng Kuo, Yin Cui, Xiuye Gu, AJ Piergiovanni, and Anelia Angelova. F-vlm: Open-vocabulary object detection upon frozen vision and language models. *arXiv preprint arXiv:2209.15639*, 2022.

Jiaming Li, Jiacheng Zhang, Jichang Li, Ge Li, Si Liu, Liang Lin, and Guanbin Li. Learning background prompts to discover implicit knowledge for open vocabulary object detection. In *CVPR*, pp. 16678–16687, 2024.

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 *CVPR*, pp. 10965–10975, 2022.

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 *ECCV*, 2014.

Mingxuan Liu, Tyler L Hayes, Elisa Ricci, Gabriela Csurka, and Riccardo Volpi. Shine: Semantic hierarchy nexus for open-vocabulary object detection. In *CVPR*, pp. 16634–16644, 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 *ECCV*, pp. 38–55. Springer, 2024b.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, pp. 11966–11976, 2022.

Yuning Lu, Jianzhuang Liu, Yonggang Zhang, Yajing Liu, and Xinmei Tian. Prompt distribution learning. In *CVPR*, pp. 5206–5215, 2022.

Chuofan Ma, Yi Jiang, Xin Wen, Zehuan Yuan, and Xiaojuan Qi. Codet: Co-occurrence guided region-word alignment for open-vocabulary object detection. *NeurIPS*, 36:71078–71094, 2023.

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 *ECCV*, pp. 728–755. Springer, 2022.

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 *ICML*, pp. 8748–8763. PMLR, 2021.

Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *ICCV*, pp. 8430–8439, 2019.

Junjie Wang, Bin Chen, Bin Kang, Yulin Li, Weizhi Xian, Yichi Chen, and Yong Xu. Ov-dquo: Open-vocabulary detr with denoising text query training and open-world unknown objects supervision. In *AAAI*, volume 39, pp. 7762–7770, 2025a.

Junjie Wang, Bin Chen, Yulin Li, Bin Kang, Yichi Chen, and Zhuotao Tian. Declip: Decoupled learning for open-vocabulary dense perception. *arXiv preprint arXiv:2505.04410*, 2025b.

Luting Wang, Yi Liu, Penghui Du, Zihan Ding, Yue Liao, Qiaosong Qi, Biaolong Chen, and Si Liu. Object-aware distillation pyramid for open-vocabulary object detection. In *CVPR*, pp. 11186–11196, 2023.

Siheng Wang, Yanshu Li, Bohan Hu, Zhengdao Li, Haibo Zhan, Linshan Li, Weiming Liu, Ruizhi Qian, Guangxin Wu, Hao Zhang, Jifeng Shen, Piotr Koniusz, Zhengtao Yao, Junhao Dong, and Qiang Sun. Deco-detr: Decoupled cognition detr for efficient open-vocabulary object detection. In *ICLR*, pp. 124043–124063, 2026.

Size Wu, Wenwei Zhang, Sheng Jin, Wentao Liu, and Chen Change Loy. Aligning bag of regions for open-vocabulary object detection. In *CVPR*, pp. 15254–15264, 2023a.

Size Wu, Wenwei Zhang, Lumin Xu, Sheng Jin, Xiangtai Li, Wentao Liu, and Chen Change Loy. Clipself: Vision transformer distills itself for open-vocabulary dense prediction. In *ICLR*, 2024.

Xiaoshi Wu, Feng Zhu, Rui Zhao, and Hongsheng Li. Cora: Adapting clip for open-vocabulary detection with region prompting and anchor pre-matching. In *CVPR*, pp. 7031–7040, 2023b.

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. *NeurIPS*, 35:9125–9138, 2022.

Lewei Yao, Jianhua Han, Xiaodan Liang, Dan Xu, Wei Zhang, Zhenguo Li, and Hang Xu. Detclipv2: Scalable open-vocabulary object detection pre-training via word-region alignment. In *CVPR*, pp. 23497–23506, 2023.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *NeurIPS*, volume 33, 2020.

Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy. Open-vocabulary detr with conditional matching. In *ECCV*, pp. 106–122. Springer, 2022.

Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-vocabulary object detection using captions. In *CVPR*, pp. 14393–14402, 2021.

Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022.

Yupeng Zhang, Yuzhong Feng, Ruize Han, Zhiwei Chen, Wei Feng, and Liang Wan. Exdet: Open-domain open-vocabulary detection with cross-modal extrapolation and rectification. *arXiv preprint arXiv:2606.09360*, 2026.

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 *CVPR*, pp. 16793–16803, 2022.

Jiaying Zhou and Qingchao Chen. State and scene enhanced prototypes for weakly supervised open-vocabulary object detection. *arXiv preprint arXiv:2511.18012*, 2025.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *CVPR*, pp. 16816–16825, 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 *ECCV*, pp. 350–368. Springer, 2022c.

A METHOD DETAILS

A.1 DESCRIPTION BANK, SLOT PRIOR, AND INITIALIZATION

Description bank. Each category uses $N_p = 8$ descriptions in a fixed order: (1) the class name, (2) a definition, (3) a state or usage phrase, (4) an attribute or context phrase, and (5)–(8) four paraphrases that do not contain the class name. For *dog*: "a dog", "a domesticated canine animal", "a dog sitting, standing, or walking", "a dog with fur and tail", "a pet with four legs", "a furry companion animal", "a barking mammal", "a canine pet". We generated the bank once in October 2025 with Claude (Anthropic), using interactive sessions with batches of 100–200 categories and a single role template. The descriptions were not edited afterwards. The generation instruction was not archived; the full bank is released as a supplementary file. Descriptions (3) and (4) emphasize state or usage information and attributes or context, respectively, while the paraphrases in (5)–(8) may also contain visual cues. The bank and its CLIP text features remain fixed. The aggregator learns to construct prototypes from these features, with no language-model calls during training or inference.

Slot prior and initialization. The fixed prior in Eq. (1) is

$$b_{k,n} = \beta_{\text{slot}} \mathbb{1}[n = p_k], \qquad p_k = \begin{cases} 1 + \text{round}\left((k-1)\dfrac{N_p - 1}{K_p - 1}\right), & K_p > 1, \\ 1, & K_p = 1, \end{cases}$$

so that with $N_p = 8$ and $K_p = 5$ the slots are seeded on positions $1, 3, 5, 6, 8$: the class name, the state/usage phrase, and three paraphrases. β_{slot} controls the prior strength; attention weights remain learnable, and preferred positions can repeat when $K_p > N_p$. The slot queries $\mathbf{q}_k$ and the key projection W_K are Xavier-initialized, and the value projection W_V is initialized to the identity.

A.2 PROTOTYPE REGULARIZATION AND GRADIENT ROUTING

Regularizer behavior. For a prototype pair with cosine u, the diversity penalty $-\log(1-(1-\epsilon)u^2)$ increases with $|u|$ and stays finite for $0 < \epsilon < 1$. Its gradient with respect to the prototype vectors vanishes when the two are exactly parallel or antiparallel, as for any function of the cosine, so the penalty does not by itself separate an already collapsed pair.

The usage distribution in $\mathcal{L}_{\text{bal}}$ is

$$\bar{\pi}_{c,k} = \frac{1}{N_p} \sum_{n=1}^{N_p} \text{softmax}_k \left(\ell_{c,k,n} / (\sqrt{d_h}\tau_p)\right), \qquad \ell_{c,k,n} = \mathbf{q}_k^\top W_K \mathbf{T}_{c,n} + b_{k,n},$$

so that $\mathcal{L}_{\text{bal}} = \frac{1}{C_t} \sum_{c \in \mathcal{C}_t} \sum_k \bar{\pi}_{c,k} \log(K_p \bar{\pi}_{c,k})$. Identical scores across slots give uniform usage even if the prototypes coincide, so balanced usage alone does not ensure directional diversity.

Gradient routing and optimization. Let $\mathbf{g}_{\text{task}} = \nabla_\theta(\mathcal{L} - \lambda_{\text{APR}}\mathcal{L}_{\text{APR}})$, which includes the detection and scheduled RPSA terms, and $\mathbf{g}_{\text{APR}} = \nabla_\theta(\lambda_{\text{APR}}\mathcal{L}_{\text{APR}})$, with θ the aggregator parameters. The routed task gradient is

$$\tilde{\mathbf{g}}_{\text{task}} = \mathbf{g}_{\text{task}} - \frac{\min\left(0,\ \langle \mathbf{g}_{\text{task}}, \mathbf{g}_{\text{APR}} \rangle\right)}{\|\mathbf{g}_{\text{APR}}\|^2} \mathbf{g}_{\text{APR}}, \tag{7}$$

and $\tilde{\mathbf{g}}_{\text{task}} + \mathbf{g}_{\text{APR}}$ is passed to the optimizer for the aggregator. The gradients are flattened into one vector, so the conflict test and projection apply to the aggregator as a whole. When the two gradients conflict, the projection gives $\langle \tilde{\mathbf{g}}_{\text{task}}, \mathbf{g}_{\text{APR}} \rangle = 0$; otherwise the inner product is unchanged. The routed direction therefore does not increase the APR objective to first order; if the gradients are exactly opposed, the routed task gradient vanishes and the APR gradient remains. This concerns the gradient handed to the optimizer and does not guarantee a decrease of either objective after the AdamW update. Computing the APR gradient separately requires one additional backward pass through the aggregator (0.79M parameters at the dimensions of Table 4); in a paired measurement over the first 50 updates this costs 2.53 versus 2.49 s per update, about 2%, and only during the epochs in which routing is active. The projection introduces no additional weighting coefficient. The

combined routed gradient is formed before gradient clipping and the AdamW update; the remaining detector parameters keep their ordinary gradients.

Routing is applied during the first four epochs and disabled afterwards, with the regularizer still active. The classification score does not penalize collapse: because of the $1/K_p$ normalization in Eq. (2), identical prototypes score exactly as one. Once the prototypes coincide, the diversity gradient vanishes and uniform usage can satisfy the balance term without restoring separation. The initial bank is already highly correlated (effective rank 3.12, mean pairwise cosine 0.89), making early prototype formation a particular concern. In a paired comparison from the same initialization on identical batches, the effective rank after 50 updates is 3.05 with routing and 2.59 without it, and the mean pairwise cosine is 0.90 versus 0.94; an independent run without routing reproduces 2.59 at the same point. These early measurements motivate applying routing while the bank forms. With routing, its effective rank reaches 4.72 of 5 at the end of the second epoch. After routing is disabled, the bank retains an effective rank of 4.38 at the end of training (Appendix B.2).

The switch is implemented as a resume: the run is launched with routing and a four-epoch stop, then resumed from update $28{,}400$ with routing disabled and the stop extended to twelve epochs. All other settings, including the APR terms and weights and the twelve-epoch learning-rate horizon and decay point, are identical across the two stages, so the schedule matches a single twelve-epoch run with routing switched off at update $28{,}400$, except that the resume does not restore the data-loader position or the random-number state.

A.3 Region–Prototype Alignment Details

Token selection and initialization. We first compute category probabilities $\mathrm{softmax}_{c\in\mathcal{C}_t}(S_{b,n,c})$ and form $M_{b,n,c}$ by retaining the L largest values for each token and setting the remaining entries to zero, without renormalization. Token confidence is $\kappa_{b,n} = \max_c M_{b,n,c}$. Eligible tokens are non-padding positions with finite generated proposals. We retain the highest-confidence eligible tokens, up to M_{tok} per image; for batching, the retained count is also capped by the smallest eligible-token count among the images. This construction requires at least K_r eligible tokens per image. Farthest-point sampling initializes the first center with the selected token of largest ℓ_2 norm, then repeatedly selects the token with the largest minimum squared distance to the existing centers. The seed features are detached before the single soft clustering update in Eq. (8).

The single soft clustering update and the center category weights are

$$r_{b,n,g} = \frac{\exp\big(-\|\mathbf{x}_{b,n} - \boldsymbol{\mu}^0_{b,g}\|^2/\sigma^2\big)}{\sum_{g'=1}^{K_r}\exp\big(-\|\mathbf{x}_{b,n} - \boldsymbol{\mu}^0_{b,g'}\|^2/\sigma^2\big)}, \qquad \boldsymbol{\mu}_{b,g} = \frac{\sum_n r_{b,n,g}\,\mathbf{x}_{b,n}}{\sum_n r_{b,n,g}}, \tag{8}$$

$$\pi_{b,g,c} = \frac{\sum_n r_{b,n,g}\,M_{b,n,c}}{\sum_n r_{b,n,g}}, \qquad \tilde{\pi}_{b,g,c} = \frac{\pi_{b,g,c}}{\sum_{c'\in\mathcal{C}_t}\pi_{b,g,c'}}, \tag{9}$$

with sums over the selected tokens of image b.

Center filtering. We measure center confidence using the unnormalized category weights from Eq. (9), $h_{b,g} = \max_c \pi_{b,g,c}$. The retained set is

$$\mathcal{V} = \big\{(b,g) : h_{b,g} \geq \max(\theta_{\mathrm{bg}}, Q_b)\big\},$$

where θ_{bg} is a fixed floor and Q_b is the q-th percentile of $h_{b,g}$ over the centers of image b. Filtering precedes category normalization, so a low-confidence center cannot pass solely because its weights are rescaled to sum to one. These scores express relative category confidence and are not foreground probabilities. The loss is averaged over retained centers in the current mini-batch; if none are retained, it is zero. Default settings are listed in Table 4.

Annotations and gradients. Center construction and category assignment use encoder features and predicted category scores, without direct assignments from ground-truth boxes, masks, or part labels. The detector remains trained with the base-category annotations described in Sec. 3.1. Token selection, seed initialization, and center filtering do not propagate gradients through their discrete choices. The token-to-category weights $M_{b,n,c}$ and normalized category targets $\tilde{\pi}_{b,g,c}$ are detached, while the soft assignments and center updates remain differentiable with respect to the selected encoder features. The alignment objective also propagates gradients through the projected prototypes.

B Experimental Details and Additional Results

B.1 Hyperparameter and Optimization Settings

Section 4.1 describes the model, training setup, and evaluation protocol. Tables 4 and 5 give the detailed settings for the complete model. Component ablations change only the corresponding component, and the sensitivity experiments vary the parameter under study.

Table 4: Default module hyperparameters. Symbols follow Sec. 3. The TPA attention denominator is $\sqrt{d_h}\,\tau_p$; the reported 0.07 refers to this product.

Setting	Symbol	Value
Text prototype aggregation		
Descriptions per category	N_p	8
Text feature dimension	d	768
Textual prototypes per category	K_p	5
Slot-query dimension	d_h	256
Attention denominator	$\sqrt{d_h}\,\tau_p$	0.07
Slot-prior strength	β_{slot}	0.2
Aggregator dropout	—	0.1
Classification and query fusion		
Classifier logit scale	s	50
Prototype-pooling temperature	τ_{cls}	0.07
Candidate categories per query seed	R	3
Category-weight temperature	τ_s	1.0
Prototype-weight temperature	τ_q	0.15
Adaptive prototype regularization		
Overall APR weight	λ_{APR}	1
Diversity weight	λ_{div}	0.10
Usage-balance weight	λ_{bal}	0.03
Log-barrier stabilizer	ϵ	10^{-4}
Region–prototype semantic alignment		
Visual centers per image	K_r	8
Retained categories per token	L	5
Selected encoder tokens per image	M_{tok}	1024
Soft clustering temperature	σ	1
Background-confidence floor	θ_{bg}	0.05
Confidence percentile	q	60
Contrastive temperature	τ_r	0.06
RPSA loss weight	λ_{RPSA}	0.05

Table 5: Optimization, training schedules, and inference calibration. Section 4.1 specifies the benchmark and component-ablation protocol; Appendix B.2 describes the diagnostic evaluations.

Setting	Value
Optimizer	AdamW
Full training schedule	12 epochs
Hardware	4 A100 GPUs
Effective batch size	32 (16 images per step with 2-step gradient accumulation)
Base learning rate	10^{-4}
Backbone normalization learning rate	$0.1\times$ the base learning rate
Aggregator learning rate	$10\times$ the base learning rate
Weight decay	10^{-4}
Gradient clipping	Norm 0.5, applied separately to the aggregator and the remaining detector parameters
Numerical precision	Mixed precision

Table 5: Optimization, training schedules, and inference calibration (continued).

Setting	Value
Random seed	42
Federated category sampling	100 categories per iteration, including categories present in the batch
APR gradient routing	Applied during the first 4 epochs (updates 0–28,400), then disabled
RPSA warm-up	$T_0 = 20{,}000$ iterations with RPSA disabled
RPSA activation ramp	$T_1 = 8{,}000$ iterations of linear ramp-up after warm-up
CLIP score-ensemble coefficients	$\alpha = 0$ (base categories, detector scores only), $\beta = 0.3$ (novel categories), $\eta = 3$

Score ensemble. Let $p_{j,c}^{\mathrm{det}} = \mathrm{sigmoid}(S_{j,c})$ denote the detector score and $p_{j,c}^{\mathrm{clip}}$ the softmax score obtained by comparing a frozen CLIP image encoder's feature for the predicted box with fixed text embeddings. We use the description embeddings released with LaMI-DETR on LVIS and the mean of the eight description embeddings per category on COCO and Objects365. The fused score is $\eta\,(p_{j,c}^{\mathrm{det}})^{1-\beta}(p_{j,c}^{\mathrm{clip}})^{\beta}$. On LVIS, we apply it only to novel categories; base categories retain $p_{j,c}^{\mathrm{det}}$. For COCO and Objects365 transfer, we apply it to all target categories.

Parameter selection. Hyperparameter values were selected using four-epoch screening runs and then fixed for the full twelve-epoch experiments. The sensitivity results in Sec. 4.4 evaluate the numbers of textual prototypes K_p and visual centers K_r after twelve epochs of training; the four-epoch runs are used only for screening.

Sensitivity sweep values. For $K_p = 1, 3, 5, 7, 9$, the corresponding AP_r values in Fig. 3(a) are 42.5, 44.8, 46.0, 45.5, and 45.0, respectively. For $K_r = 4, 8, 16, 32$, Fig. 3(b) reports 45.0, 46.0, 45.4, and 44.7 AP_r, respectively.

Benchmark scope. Our evaluation protocol follows LaMI-DETR, which reports OV-LVIS and cross-dataset transfer to COCO and Objects365 and does not report the OV-COCO 48/17 split. We do not train on that split, although several methods in Table 1 report it; on OV-COCO, reported differences additionally depend on the use of caption or image-level data, which is orthogonal to the representation studied here. Whether the description-level representation offers similar benefits on its 17 novel categories remains untested (Appendix C). They are also all frequent LVIS categories, each with at least 700 training images carrying box supervision in our training set, so the LVIS-trained detector cannot be evaluated as open-vocabulary on them; an OV-COCO result would require a separate detector trained on the 48 base categories. LaMI-DETR additionally reports a zero-shot LVIS setting trained on Objects365 and Visual Genome, which uses different training data and is not part of our protocol.

B.2 Prototype Diagnostics

Checkpoint and scope. Unless otherwise stated, diagnostics use the same 12-epoch model as the benchmark results. Statistics of description attention and projected text features cover all 1,203 LVIS categories. The bank-substitution study uses LVIS minival (4,809 images); region retrieval and alignment statistics use LVIS full validation (19,809 images). These are checkpoint analyses, not additional variants of the twelve-epoch component-ablation table.

Description geometry. For each category, we normalize its eight description vectors and compute the mean cosine over distinct pairs. Effective rank is $\exp(-\sum_i p_i \log p_i)$, where $p_i = \sigma_i^2 / \sum_j \sigma_j^2$ and σ_i are the singular values of the matrix of normalized description vectors. We compute the same quantities after the learned value projection and average over categories. Figure 6(b) shows a decrease in mean pairwise cosine from 0.791 to 0.496 and an increase in effective rank from 2.20 to 4.07. The mean cosine between each category's description centroid and its nearest other category centroid also decreases, from 0.912 to 0.767. Thus, the projection reduces similarity both within categories and between their nearest centroids. The mean peak attention is 0.964 over all 6,015 category–slot

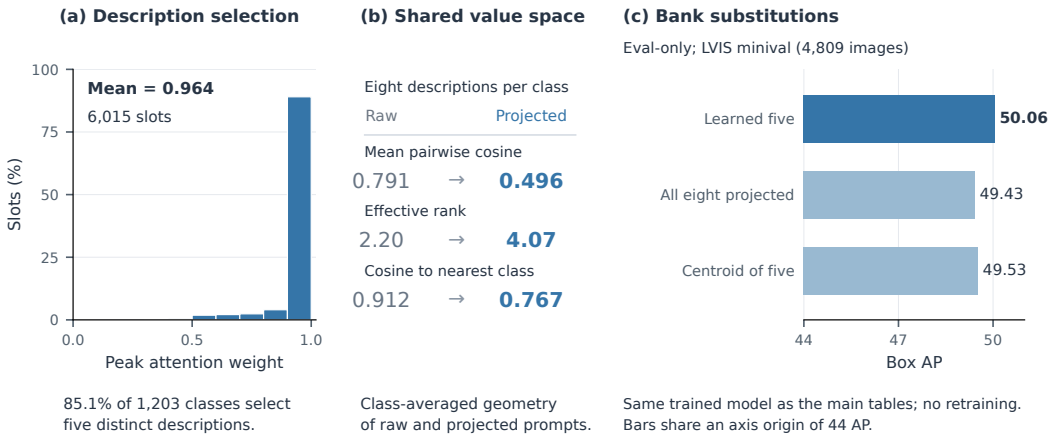


Figure 6: Prototype diagnostics. (a) Peak description-attention weights across 1,203 categories and five slots. (b) Geometry of the eight descriptions per category before and after value projection. (c) LVIS minival AP under evaluation-only prototype-bank substitutions.

pairs (Fig. 6(a)); 85.1% of categories select five distinct descriptions. Distinct-description counts use the maximum-attention description of each slot, including the fixed slot prior. Using the entropy of normalized singular values rather than their squares, the bank reaches an effective rank of 4.72 after two epochs (Appendix A.2) and retains a rank of 4.38 after training (Table 6).

Slot prior versus learned selection. Table 6 compares each slot's maximum-attention description with its prior position. At initialization the prior determines the selection for 99.9% of slots. After training, 55.1% of slots remain on their prior position, well above the 12.5% a uniform choice would give, so the prior still shapes which description a slot takes. Of the 2,699 slot–category pairs that do move, 54% move between paraphrases and a further 13% from the state/usage phrase to the attribute/context one, indicating that reassignment is concentrated among these descriptions. Agreement with the prior is lowest for rare categories (47.1%). This difference alone does not establish better adaptation to rare categories, which receive no box supervision. Removing the prior at inference lowers the share by 3.8 points and the bank's effective rank by 0.04. What the prior does not supply is the separation itself: at initialization, 99.9% of slots select their prior positions, yet the bank's effective rank is near 3.1 (App. A.2). Training raises it to 4.38 while moving nearly half of the slots off those positions; removing the prior from the trained aggregator changes the rank by only 0.04. The trained bank therefore retains its separation without the fixed prior at inference, although this does not isolate the prior's influence during training. Slot roles are consistent across categories: the first slot selects the class name for 1,138 of 1,203 categories; the second selects the state/usage phrase for 772 and the attribute/context phrase for 349; the third, fourth, and fifth slots select their prior paraphrase for 496, 447, and 463 categories and otherwise spread mainly over the other paraphrases and the definition.

What bank substitutions establish. Figure 6(c) and Table 7 measure the behavior of a fixed detector under changes to its text-side representation, on LVIS minival and under the evaluation protocol used throughout this paper. The learned five-prototype bank outperforms all eight value-projected descriptions by 0.63 AP and 0.47 AP_r, showing that its representation is not equivalent to retaining every description. Collapsing the bank to its centroid also costs accuracy, by 0.53 AP and 0.91 AP_r, so the five prototypes are not interchangeable with their mean at inference. These are evaluation-time substitutions on a fixed checkpoint and do not predict what training each representation from initialization would give.

Region collection and filtering. For each annotated validation instance, we independently select the final decoder query with the highest box IoU and retain matches with IoU ≥ 0.5; a correct predicted category is not required. This is not a one-to-one Hungarian assignment. The collection contains 221,298 matched instances. For retrieval and alignment statistics, we retain boxes whose

Table 6: Effect of the slot prior. Left: share of category–slot pairs whose maximum-attention description is the slot's prior position. Right: effective rank of the five-prototype bank, class-averaged, from the entropy of the normalized singular values of the normalized prototypes. This differs from the description effective rank in Fig. 6(b), which uses squared singular values over the eight description vectors. The rank at initialization depends on the random draw and is reported for the training run in App. A.2. Rows are the aggregator at initialization, after training, and after training with the prior removed at inference.

	On prior position (%)				Bank effective rank			
	All	Freq.	Com.	Rare	All	Freq.	Com.	Rare
Initialization	99.9	100.0	99.9	99.9	—	—	—	—
Trained	55.1	65.5	51.9	47.1	4.38	4.56	4.43	4.09
Trained, prior removed	51.3	63.2	47.8	41.7	4.34	4.52	4.37	4.08

Table 7: Evaluation-only bank substitutions on LVIS minival ($4{,}809$ images), under the evaluation protocol used throughout this paper (CLIP score ensembling with $\beta = 0.3$ and $\eta = 3$). Detector weights are unchanged; the bank is replaced jointly in classification and query initialization. These interventions differ from training each representation from initialization.

Prototype bank	AP	AP_r	AP_c	AP_f
Learned five-prototype bank	**50.06**	**46.49**	**50.82**	**50.02**
All eight value-projected descriptions	49.43	46.02	49.93	49.60
Centroid of the five prototypes	49.53	45.58	50.43	49.44

shorter side is at least 64 pixels and whose area is at most 90% of the image. Boxes touching image boundaries are retained. Each displayed crop is a square context window around the ground-truth box, enlarged by 15%, shifted inside the image boundary, and resized to 256×256 pixels.

Retrieval and figure selection. Retrieval uses detector features before CLIP score ensembling. For a ground-truth category c and its associated query feature $\mathbf{z}_i$, we compute the relative prototype score

$$r_{i,k} = \cos(\mathbf{z}_i, \mathbf{P}_{c,k}) - \frac{1}{K_p} \sum_{j=1}^{K_p} \cos(\mathbf{z}_i, \mathbf{P}_{c,j}). \tag{10}$$

Each slot ranks eligible instances by this score. Slots take candidates in round-robin order, allowing each instance to appear in at most one row and each image to contribute at most one instance to a given row. This deduplication rule does not require a row's slot to be the highest-scoring slot for every displayed instance. Figure 4 shows the first four exported instances for selected rows: cup slots 1, 4, and 3; dog slots 1, 3, and 0; and coat slots 2, 4, and 1 (zero-based indices). The row label is the original description receiving the largest attention weight from that slot. The figure is a selected illustration, while the statistics below use all eligible instances of each included category.

CLIP-based correspondence. We compare $r_{i,k}$ with the frozen CLIP region feature's cosine similarity to the description selected by slot k. CLIP features are extracted from ground-truth regions. Within each category, CLIP similarities are centered both across the selected descriptions for each instance and across instances for each description, to reduce shared instance effects and description-specific score biases. For each slot whose selected description is unique within the category, we compute Spearman correlation across eligible instances with its own description and with the other distinct slots' descriptions. The analyzed subset contains 130 frequent categories with at least 100 eligible instances each, giving 650 category–slot pairs. Mean corresponding correlation is 0.126, compared with -0.031 for other descriptions; 81.7% of pairs have a larger corresponding correlation, and 93.8% of categories have a positive mean corresponding correlation. This is a frozen-CLIP proxy on a frequent-category subset, not an independent semantic annotation or a test of rare-category performance.

B.3 Encoder-Feature Visualization Details

Figure 5 provides a separate post-hoc clustering visualization of encoder features on selected LVIS validation images. Its colors identify feature groups, not predicted instance or part masks. This illustration is auxiliary to the component results; it does not compare RPSA with a single-vector baseline or establish that these assignments exactly match those used during training.

C Limitations

Our evaluation covers open-vocabulary object detection on LVIS and cross-dataset transfer to COCO and Objects365. These results do not establish whether the same prototype representation would benefit other tasks, such as open-vocabulary segmentation, phrase grounding, or video object detection.

The value of this representation for small vocabularies also remains unclear. A class-name embedding may already distinguish their categories well, but we have not tested whether this reduces the benefit of learning from descriptions. Small-vocabulary settings, including the OV-COCO split (Appendix B.1), remain future work.

ProtoSet-DETR also relies on offline category descriptions and a frozen vision-language text encoder. The quality, diversity, and visual relevance of these descriptions may therefore affect the learned prototype bank. Categories with ambiguous names, highly context-dependent meanings, or insufficiently diverse descriptions may still be difficult to represent with a fixed number of prototypes.

Text prototype aggregation, adaptive prototype regularization, and region–prototype semantic alignment increase implementation complexity during training. APR and RPSA are used only during training, and the prototype bank can be cached for inference without online language-model calls. The method also introduces several hyperparameters, such as the number of textual prototypes, the number of visual centers, and the weights and schedules of the auxiliary losses.

Our analysis uses standard benchmark splits and leaves robustness to severe domain shift, noisy vocabularies, and biased category descriptions largely untested. Evaluating these conditions would clarify how well the learned prototypes handle categories and appearances beyond those represented in the training data.

D Broader Impact

Open-vocabulary object detection lowers the cost of building visual recognition systems for long-tail categories and emerging concepts, which can benefit applications in robotics, accessibility, content moderation, and scientific imaging. At the same time, more capable detectors could be misused for unauthorized surveillance or to amplify existing biases in benchmark annotations (e.g., demographic imbalance in COCO/LVIS). The method inherits risks associated with its training benchmarks and CLIP backbone. We encourage deployers to (i) audit the category vocabulary for sensitive classes, (ii) follow standard responsible-AI guidelines for surveillance-adjacent applications, and (iii) treat detector outputs as probabilistic estimates rather than authoritative judgments.