OPUSDET: A SIMPLE YET EFFECTIVE UNIFIED FRAMEWORK FOR OPEN-VOCABULARY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent unified open-vocabulary detection (OVD) methods support text queries, visual exemplars, and their combinations. However, they often rely on increasingly complex designs such as heavy cross-modal fusion, staged training, and iterative annotation pipelines. We revisit whether such complexity is necessary with stronger foundation models available. We find that unified OVD can be made much simpler with strong visual features and scalable grounding supervision. We present OpusDet (**O**pen-vocabulary, **P**rompt-**U**nified, **S**imple), a unified detector supporting text, interactive visual, generic visual, and mixed prompting. OpusDet keeps the model, training, and data pipeline simple. It uses one detector for all prompt types, one-stage text–visual training, and a SAM3-based single-pass data engine. From the matched Base to the Full data setting, with the same Swin-T backbone, OpusDet gains 6.5 Visual-I AP on LVIS-minival, compared with 3.1 AP for our T-Rex2-style re-implementation. Our final DINOv3-ConvNeXt-B model achieves state-of-the-art Visual-I performance among comparable unified detectors, reaching 68.1, 69.2, and 54.7 AP on COCO, LVIS-minival, and ODinW35. It also maintains competitive Text and Visual-G performance. It further turns mixed prompting from interference into complementarity, improving over text or visual prompts alone. These results show that a simple unified detector can provide strong performance across prompting modes.

1 INTRODUCTION

Open-vocabulary detection (OVD) localizes and recognizes objects beyond a fixed label set by aligning visual regions with semantic embeddings. Recent work has moved from text-only queries toward unified prompting. In this setting, the same detector also accepts visual prompts, such as reference images, boxes, or points. These prompts provide appearance cues that category names may miss, especially for long-tailed or textually ambiguous objects.

This flexibility often comes at the cost of a more complex system. Some methods add early cross-modal fusion or extra decoder modules to Grounding DINO (Liu et al., 2024; Zhao et al., 2024a)-style models (Chen et al., 2025; Qian et al., 2026). Others use staged text–visual training (Jiang et al., 2024; Fu et al., 2026). Some also use iterative re-annotation that interleaves training with data refinement (Jiang et al., 2024). These designs improve unified OVD, but increase the cost of model design, training, and data preparation. Meanwhile, recent advances in visual foundation models like DINOv3 (Siméoni et al., 2025) for rich representations and SAM3 (Carion et al., 2025) for unified promptable perception inspire a simpler alternative. This motivates us to ask if unified OVD can retain strong text, visual, and mixed prompting without heavy fusion, staged training, or iterative relabeling.

We present OpusDet (**O**pen-vocabulary, **P**rompt-**U**nified, **S**imple), a unified detector for text, interactive visual (Visual-I), generic visual (Visual-G), and mixed prompting. OpusDet simplifies three parts of unified OVD, as shown in Figure 1. First, it pairs a semantic-rich encoder with a GDINO-style prompt-aware decoder. Second, it adopts one-stage text–visual training with Instance-level Contrastive Alignment (ICA) to align visual prompts with matched object queries. And third, it employs a SAM3-based single-pass data engine to convert diverse sources into grounding supervision.

From the matched Base to the Full data setting, under the same Swin-T backbone, OpusDet gains 6.5 LVIS-minival (Gupta et al., 2019) Visual-I AP, compared with 3.1 AP for our T-Rex2-style re-

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.

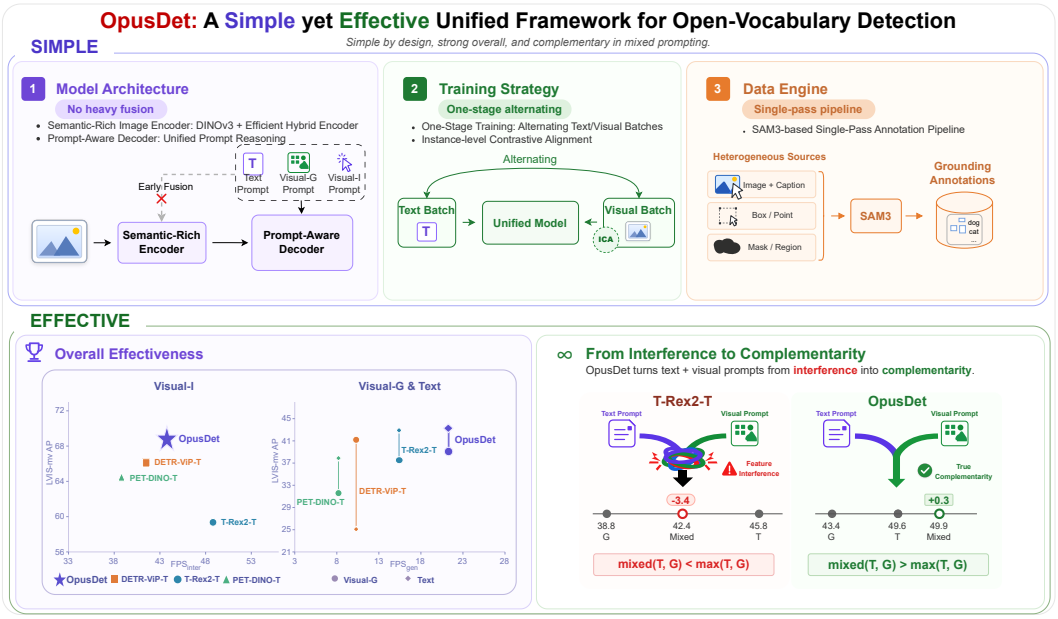

Figure 1: Overview of OpusDet. Top: a simple three-part design across architecture, training, and data engine for unified OVD. Bottom left: accuracy–efficiency trade-off across prompting modes. Bottom right: mixed prompting becomes complementary rather than interfering.

implementation. Our final model achieves state-of-the-art Visual-I performance among comparable unified detectors, reaching 68.1 on COCO (Lin et al., 2014), 69.2 on LVIS-minival, and 54.7 on ODinW35 (Li et al., 2022a). It improves over PET-DINO-T (Fu et al., 2026) by 6.4 AP and over DETR-ViP-T (Qian et al., 2026) by 7.9 AP on ODinW35, while maintaining competitive Text and Visual-G performance.

Mixed prompting provides another benefit. On LVIS-minival, Mixed reaches 45.2 AP, above Text (43.0) and Visual-G (38.6). On COCO, Mixed reaches 49.9 AP, above Text (49.6) and Visual-G (43.4). In contrast, T-Rex2 drops from 45.8 Text AP to 42.4 Mixed AP. As grounding supervision expands, Mixed gains less over Text but more over Visual-G. Our findings reveal a complementary mechanism where visual prompts provide crucial spatial guidance when text supervision is limited, while text prompts help stabilize noisy cross-image visual prototypes.

Our contributions are summarized as follows:

- We present OpusDet, a simple unified OVD framework for text, visual, and mixed prompting without heavy fusion, staged training, or iterative relabeling.
- Experiments on COCO, LVIS-minival, and ODinW35 show that OpusDet achieves state-of-the-art Visual-I performance while maintaining strong Text and Visual-G accuracy within a single unified model.
- We show mixed-prompt complementarity and observe how the contributions of text and visual prompts change as grounding supervision expands.

2 Related Work

From Text to Visual Prompting. Open-vocabulary detection initially used category names or noun phrases as text queries. OWL-ViT (Minderer et al., 2022), GLIP (Li et al., 2022b), and Grounding DINO (Liu et al., 2024) established large-scale region-text alignment. Later methods improved open-vocabulary accuracy or inference efficiency (Wang et al., 2024; Yao et al., 2024; Fu et al., 2025b; Cheng et al., 2024; Zhang et al., 2023). Text prompts, however, provide limited appearance information for fine-grained, long-tailed, or ambiguous categories. Visual prompting addresses this limitation with exemplars, boxes, or points (Li et al., 2024; Jiang et al., 2023; Xu et al., 2023; Amini-Naieni et al., 2024). Prompt-DINO (Guan et al., 2025) further combines text guidance with visual

prompts for generic segmentation. These methods show that visual cues can complement language, especially when category names provide limited appearance information.

Unified Prompting and System Complexity. Recent unified detectors support both text and visual prompts, but often at the cost of higher system complexity. Some add cross-modal fusion or extra decoder modules to Grounding DINO-style architectures (Chen et al., 2025; Qian et al., 2026). Others use staged text–visual training (Jiang et al., 2024; Fu et al., 2026). Grounding supervision is expanded in several ways. Some methods use detector-generated pseudo boxes (Li et al., 2022b; Minderer et al., 2023; Cheng et al., 2024) or MLLM-assisted annotation pipelines (Yao et al., 2024; Fu et al., 2025a). Others rely on large-scale multi-source training (Wu et al., 2024; Ren et al., 2024) or iterative data refinement (Jiang et al., 2024). These designs improve data coverage or prompting capability, but also increase the cost of model design, training, or data preparation. While models like T-Rex2 (Jiang et al., 2024) and VisTex-OVLM (Wu et al., 2025) enable mixed prompting, how text–visual complementarity evolves as grounding training data expands remains rarely studied.

Foundation Models for Simpler Unified OVD. Modern foundation models enable a simpler unified OVD pipeline. Self-supervised visual encoders such as DINOv3 (Siméoni et al., 2025) offer transferable image features that can reduce the need for early cross-modal fusion. Promptable segmentation models such as SAM3 (Carion et al., 2025) can generate large-scale grounding supervision from diverse sources in a single pass. OpusDet follows this route by combining semantically rich visual features with one-stage text–visual training and single-pass grounding supervision. Adopting T-Rex2's prompt aggregation, OpusDet exhibits stronger text–visual complementarity across supervision scales, while revealing a progressive role shift between the two modalities.

3 METHOD

3.1 MODEL ARCHITECTURE

Semantic-Rich Image Encoder. We use a DINOv3 (Siméoni et al., 2025)-pretrained ConvNeXt (Liu et al., 2022) backbone to extract multi-scale image features $\{C_3, C_4, C_5\}$. Instead of performing heavy early cross-modal fusion, OpusDet first processes these visual features with an efficient hybrid encoder (Zhao et al., 2024b), which consists of attention-based intra-scale feature interaction and CNN-based cross-scale feature fusion:

$$\mathbf{F} = \mathrm{HybridEnc}(C_3, C_4, C_5) \in \mathbb{R}^{N_f \times D}, \tag{1}$$

where $N_f = \sum_{l \in \{3,4,5\}} H_l W_l$ denotes the number of visual tokens across all feature scales, and D is the hidden dimension. OpusDet therefore has no encoder-side image-text fusion.

Unified Prompt Generator. Following T-Rex2 (Jiang et al., 2024), OpusDet supports text and visual prompts under a unified prompt representation. Let M denote the number of prompted categories in the current forward pass, not the full dataset vocabulary. For text prompts, category names or noun phrases are encoded by a CLIP (Radford et al., 2021) text encoder and projected into the detector hidden space, producing text prompt features $\mathbf{P}^t \in \mathbb{R}^{M \times D}$. For visual prompts, a Visual Prompt Encoder (VPE) encodes user-provided boxes or points. Given K_m visual exemplars for category $m \in \{1, \ldots, M\}$, VPE stacks three Transformer layers with multi-scale deformable cross-attention to extract features from multi-scale image feature maps conditioned on these exemplars, yielding instance-level content prompt features $\mathbf{P}^v_{m,\mathrm{con}}$ and an aggregated class-level prompt feature $\mathbf{P}^v_{m,\mathrm{cls}}$ via a learnable class token:

$$\mathbf{P}^v_m = [\mathbf{P}^v_{m,\mathrm{con}}; \mathbf{P}^v_{m,\mathrm{cls}}] \in \mathbb{R}^{(K_m+1) \times D}. \tag{2}$$

We denote all instance-level content prompt features by $\mathbf{P}^v_{\mathrm{con}} \in \mathbb{R}^{K \times D}$, where $K = \sum_{m=1}^{M} K_m$. Class-level features of all M categories are stacked into $\mathbf{P}^v_{\mathrm{cls}} \in \mathbb{R}^{M \times D}$. At inference, Visual-I uses same-image exemplars, while Visual-G averages cross-image category prototypes into $\mathbf{P}^v_{\mathrm{cls}}$, and the final prompt $\mathbf{P} \in \mathbb{R}^{M \times D}$ is selected by the prompting setting:

$$\mathbf{P} = \begin{cases} \mathbf{P}^t, & \text{Text}, \\ \mathbf{P}^v_{\mathrm{cls}}, & \text{Visual}, \\ (\mathbf{P}^t + \mathbf{P}^v_{\mathrm{cls}})/2, & \text{Mixed}. \end{cases} \tag{3}$$

Prompt-Aware Decoder. To support unified reasoning over heterogeneous prompts, we adopt a prompt-aware decoder based on the Grounding DINO (Liu et al., 2024) decoder. Given image features $\mathbf{F} \in \mathbb{R}^{N_f \times D}$ and prompt features $\mathbf{P} \in \mathbb{R}^{M \times D}$, we initialize N_q object queries $\mathbf{Q} \in \mathbb{R}^{N_q \times D}$ through prompt-guided proposal selection. Each decoder layer l iteratively refines the queries via query self-attention, prompt cross-attention over $\mathbf{P}$, and image cross-attention over $\mathbf{F}$:

$$\mathbf{Q}^{(l)} = \mathrm{DecLayer}\Big(\mathbf{Q}^{(l-1)}, \mathbf{F}, \mathbf{P}\Big). \tag{4}$$

The same decoder is shared across text, interactive visual, generic visual, and mixed prompting, without prompt-specific branches.

3.2 Training Strategy and Objective

One-Stage Text-Visual Training. OpusDet is trained with a one-stage alternating strategy. For text-prompt batches, we use object detection and visual grounding data, where category names or noun phrases serve as text prompts. For visual-prompt batches, we sample ground-truth boxes or points as visual prompts and train the model to detect objects conditioned on visual exemplars. Text and visual batches are alternated during training.

Detection and Alignment Losses. The detection objective follows DETR-style detection training and includes classification, box regression, IoU, and denoising losses:

$$\mathcal{L}_{\mathrm{det}} = \mathcal{L}_{\mathrm{cls}} + \mathcal{L}_{\mathrm{bbox}} + \mathcal{L}_{\mathrm{iou}} + \mathcal{L}_{\mathrm{dn}}. \tag{5}$$

For visual-prompt batches, we use the global prompt alignment loss from T-Rex2 (Jiang et al., 2024). It aligns the aggregated class-level visual prompt $\mathbf{P}^{v}_{\mathrm{cls},m}$ with the text prompt $\mathbf{P}^{t}_{m}$ for the same category m. The base objective is therefore

$$\mathcal{L}_{\mathrm{base}} = \mathcal{L}_{\mathrm{det}} + \mathcal{L}_{\mathrm{align}}. \tag{6}$$

While this global alignment encourages text-visual category consistency, it does not explicitly preserve instance-specific appearance cues, which are important for visual and mixed prompting.

Instance-level Contrastive Alignment (ICA). ICA complements the class-level alignment loss by preserving instance-specific appearance cues. While $\mathcal{L}_{\mathrm{align}}$ aligns the aggregated visual prompt with the text prompt for each category, $\mathcal{L}_{\mathrm{ICA}}$ aligns content prompts with decoder queries at the instance level. For each category, we randomly select one or more ground-truth instances as visual prompts. After Hungarian matching, only selected instances with a matched query enter ICA. For each such instance, the matched query and content prompt form a positive pair $(Q_i, P^{v}_{\mathrm{con},i})$. All other content prompts $\{P^{v}_{\mathrm{con},j}\}_{j \neq i}$ in the local mini-batch serve as negatives, regardless of category. This contrastive objective distinguishes individual instances, while $\mathcal{L}_{\mathrm{align}}$ preserves category-level semantic alignment. The ICA loss is then defined as:

$$\mathcal{L}_{\mathrm{ICA}} = -\frac{1}{N^{+}} \sum_{i=1}^{N^{+}} \log \frac{\exp(\mathrm{sim}(Q_i, P^{v}_{\mathrm{con},i})/\tau)}{\sum_{j=1}^{N^{+}} \exp(\mathrm{sim}(Q_i, P^{v}_{\mathrm{con},j})/\tau)}, \tag{7}$$

Here, N^{+} is the number of matched instances in the local mini-batch, sim is cosine similarity, and τ is the temperature. In practice we use the symmetric form that also contrasts content prompts against queries and averages the two directions. The final objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{base}} + \mathcal{L}_{\mathrm{ICA}}, \tag{8}$$

where $\mathcal{L}_{\mathrm{ICA}}$ is applied only to visual-prompt batches and set to zero otherwise.

3.3 Data Engine

We construct additional grounding data with a shared SAM3-based pipeline. The resulting corpus combines existing detection and grounding datasets with annotated text-image, classification, and segmentation data, as shown in Figure 2.

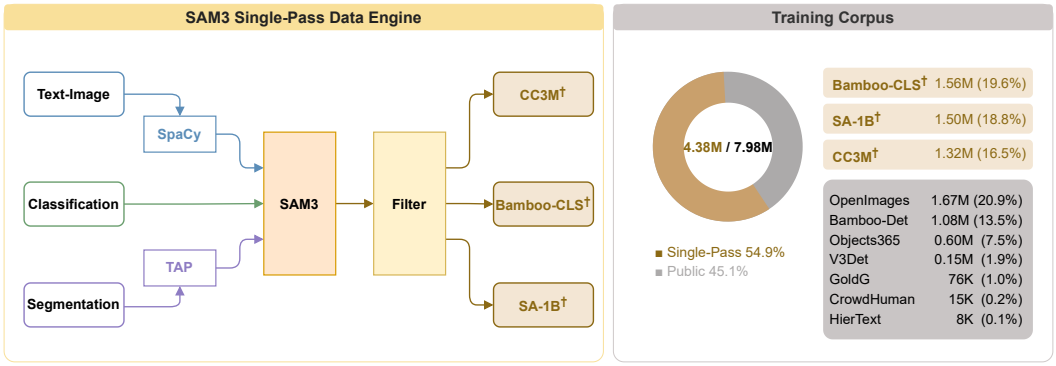

Figure 2: Overview of the single-pass data engine and the OpusDet training corpus. The data engine processes text-image data with spaCy, classification data with image-level labels, and segmentation data with TAP. SAM3 converts all three sources into grounding annotations. The final corpus contains 7.98M images, including 4.38M images annotated by this pipeline.

For text-image data, we extract noun phrases from captions with spaCy (Honnibal et al., 2020). SAM3 (Carion et al., 2025) uses these phrases as prompts and returns the corresponding object boxes. For classification data, SAM3 uses image-level labels as prompts to generate box annotations. For segmentation data, we first apply TAP (Pan et al., 2024) to assign unified semantic labels to existing masks or regions. We then use the labeled masks and their category names as prompts for SAM3 to identify additional instances and complete missing box annotations.

This pipeline generates grounding annotations for Bamboo-CLS (Zhang et al., 2025), CC3M (Sharma et al., 2018), and SA-1B (Kirillov et al., 2023). We filter the generated boxes using confidence thresholds, category-wise and image-wise NMS, and geometric rules. The geometric rules remove near full-image boxes and large crowd boxes that enclose many smaller instances. Appendix C reports the full pipeline, annotation quality, and representative success and failure cases.

4 Experiments

4.1 Experimental Setup

Model and Training Details. Unless otherwise specified, OpusDet uses a DINOv3-pretrained ConvNeXt-B backbone, a CLIP-B text encoder, and a T-Rex2-style VPE. We uses AdamW (Loshchilov & Hutter, 2017) and alternates visual and text batches at a ratio of 1:8. For Table 6, Full models use 16 A100 GPUs, a global batch size of 128, and 400K iterations. Base models use 8 A100 GPUs, a global batch size of 64, and 120K iterations. Appendix D.1 provides the full settings, and Appendix E.1 studies the alternating ratio.

Evaluation Protocol. We report zero-shot AP on COCO (Lin et al., 2014), LVIS-minival (Gupta et al., 2019), and ODinW35 (Li et al., 2022a) following the protocol of T-Rex2 (Jiang et al., 2024). Unless otherwise noted, baseline AP results are taken from the original papers. We evaluate Text, Visual-I (same-image exemplars), Visual-G (mean prototypes from N=16 training references), and Mixed (average of Text and Visual-G). We measure $\mathrm{FPS}_{\mathrm{gen}}$ and $\mathrm{FPS}_{\mathrm{inter}}$ on COCO using reproduced public code or architectures. Appendix D.2 provides the full protocols.

4.2 Main Results

Accuracy and Efficiency. Table 1 compares OpusDet with T-Rex2-T(Jiang et al., 2024), CP-DETR-T(Chen et al., 2025), DETR-ViP-T(Qian et al., 2026), PET-DINO-T(Fu et al., 2026), and YOLOE-v8-L(Wang et al., 2025) under Visual-I, Visual-G, and Text prompting. We also include T-Rex-Omni (Jiang et al., 2026) as a Visual-G-specialized reference. Its training-free variant (T-Rex-Omni-TF) improves T-Rex2-T from 38.8/37.4 to 41.6/40.6 Visual-G AP on COCO/LVIS-minival. The fine-tuned version (T-Rex-Omni-FT) further reaches 43.6/43.0 AP through hard-negative mining for fine-grained Visual-G.

Table 1: **Zero-shot detection under Visual-I, Visual-G, and Text prompting on COCO, LVIS-minival, and ODinW35.** $\mathrm{FPS}_{\mathrm{inter}}$ and $\mathrm{FPS}_{\mathrm{gen}}$ denote inference speeds with cached image and prompt features, respectively. **Bold** / underline: best / second best.

Model	Backbone	Input Size	Visual-I				Visual-G			Text			$\mathrm{FPS}_{\mathrm{gen}}$
			COCO	LVIS-mv	ODinW	$\mathrm{FPS}_{\mathrm{inter}}$	COCO	LVIS-mv	ODinW	COCO	LVIS-mv	ODinW	
T-Rex2-T	Swin-T	800×1333	56.6	59.3	37.7	***48.8***	38.8	37.4	23.6	45.8	42.8	18.0	*15.4*
CP-DETR-T	Swin-T	800×1333	61.8	64.1	41.0	–	–	–	–	**52.0**	**47.6**	**27.3**	–
DETR-ViP-T	Swin-T	800×1333	65.4	66.1	46.8	*41.5*	43.2	41.1	**31.2**	–	25.0	–	*10.3*
PET-DINO-T	Swin-T	800×1333	64.3	64.5	48.3	*38.8*	38.4	31.5	25.5	49.8	37.8	20.6	*8.2*
YOLOE-v8-L	YOLOv8-L	640×640	–	–	–	–	–	34.2	–	–	35.9	–	–
T-Rex-Omni-TF	Swin-T	800×1333	–	–	–	–	41.6	40.6	–	–	–	–	–
T-Rex-Omni-FT	Swin-T	800×1333	–	–	–	–	**43.6**	**43.0**	25.2	–	–	–	–
OpusDet (Swin-T)	Swin-T	800×1333	**68.1**	**70.6**	53.7	*42.1*	35.8	35.8	24.8	47.1	44.6	22.0	*14.5*
OpusDet	ConvNeXt-B	640×640	**68.1**	69.2	**54.7**	*43.5*	43.4	38.6	25.9	49.6	43.0	22.1	***21.3***

Table 2: **Swin-L baselines under Visual-I (our single model).** Gray: COCO/LVIS-trained.

Model	Backbone	Input Size	COCO	LVIS-mv	ODinW
T-Rex2-L	Swin-L	800 × 1333	58.5	62.5	39.7
CP-DETR-L	Swin-L	800 × 1333	68.4	71.6	50.6
DETR-ViP-L	Swin-L	800 × 1333	**71.1**	**71.9**	51.2
PET-DINO-L	Swin-L	800 × 1333	66.5	65.8	49.7
OpusDet	ConvNeXt-B	640 × 640	68.1	69.2	**54.7**

Table 3: **Zero-shot detection under Mixed prompting on COCO, LVIS-minival, and LVIS.**

Model	COCO			LVIS-mv			LVIS		
	Text	Visual-G	Mixed	Text	Visual-G	Mixed	Text	Visual-G	Mixed
T-Rex2-T	45.8	38.8	42.4	42.8	37.4	–	34.8	34.9	37.0
OpusDet	49.6	43.4	**49.9**	43.0	38.6	**45.2**	34.8	31.2	**37.6**

OpusDet achieves the best Visual-I performance among the compared unified detectors on COCO, LVIS-minival, and ODinW35, surpassing DETR-ViP-T by 2.7 AP on COCO and 3.1 AP on LVIS-minival, as well as exceeding PET-DINO-T and DETR-ViP-T by 6.4 and 7.9 AP on ODinW35 When compared to the specialized T-Rex-Omni-FT, OpusDet is 0.2 AP lower on COCO Visual-G and 4.4 AP lower on LVIS-minival, but outperforms it by 0.7 AP on ODinW35. Importantly, while T-Rex-Omni-FT explicitly targets fine-grained Visual-G, OpusDet maintains competitive Visual-G quality while supporting Text, Visual-I, Visual-G, and Mixed prompting in a unified model.

For a comparison under the same backbone and input size, we report OpusDet (Swin-T) using Swin-T (Liu et al., 2021) with an 800×1333 resolution. Compared with T-Rex2-T, it improves Visual-I AP on LVIS-minival by 11.3 to 70.6, with a 1.6 AP trade-off in Visual-G. Our primary OpusDet architecture adopts a DINOv3-pretrained ConvNeXt-B backbone with hybrid encoding and a 640×640 input. Compared with OpusDet (Swin-T), it recovers Visual-G and provides a more balanced profile across generic and interactive detection, while reaching 21.3 $\mathrm{FPS}_{\mathrm{gen}}$, the highest reported speed in Table 1.

Table 2 compares OpusDet with the larger Swin-L variants of T-Rex2, CP-DETR, DETR-ViP, and PET-DINO. Despite using a smaller ConvNeXt-B backbone and a 640×640 input, OpusDet remains within 3 AP of DETR-ViP-L on COCO and LVIS-minival, while outperforming all Swin-L baselines on ODinW35 by at least 3.5 AP.

Mixed Prompting Complementarity. Table 3 evaluates Mixed prompting. On COCO, T-Rex2-T drops from 45.8 (Text) to 42.4 (Mixed), showing interference. In contrast, OpusDet maintains its strong performance on COCO at 49.9 AP (from 49.6 Text AP) and from 43.0 to 45.2 on LVIS-minival. On full LVIS, OpusDet also achieves a higher Mixed score than T-Rex2-T despite its lower Visual-G score. These results suggest that OpusDet combines text and visual cues more effectively. Section 4.5 further analyzes how this complementarity evolves as grounding supervision expands.

Table 4: **Cumulative ablation on LVIS-minival.** Δ_T/Δ_G: Mixed gains over Text / Visual-G.

ID	Evolution Step	FPS_{gen}	Visual-I		Visual-G		Text		Mixed		Δ_{T}	Δ_{G}
			AP	AP_r	AP	AP_r	AP	AP_r	AP	AP_r		
A	T-Rex2-style baseline	15.4	56.3	63.8	33.6	28.0	26.5	16.0	35.2	22.8	+8.7	+1.6
B	Backbone and encoder	22.0	57.5	66.8	35.3	29.1	29.5	20.4	36.2	33.1	+6.7	+0.9
C	Prompt-aware decoder	21.3	63.0	73.9	33.3	26.1	32.7	20.5	36.5	30.8	+3.8	+3.2
D	ICA	21.3	63.2	74.7	33.5	30.0	33.2	20.3	38.0	31.1	+4.8	+4.5
E	Data engine	21.3	69.2	78.3	38.6	32.8	43.0	36.6	45.2	38.7	+2.2	+6.6

Table 5: **Accuracy–efficiency decomposition of Step B.** B_1 changes the backbone, B_2 changes the input resolution, and B_3 changes the encoder.

ID	Evolution Step	Params	Input	GFLOPs	FPS_{gen}	Visual-I	Visual-G	Text	Mixed
A	T-Rex2-style baseline	115M	800×1333	257	15.4	56.3	33.6	26.5	35.2
B_1	DINOv3-ConvNeXt-B	173M	800×1333	439	11.6	63.2	38.0	36.4	39.3
B_2	640×640 input	173M	640×640	208	19.4	59.1	33.4	28.6	34.4
B_3	Hybrid encoder	172M	640×640	164	22.0	57.5	35.3	29.5	36.2

4.3 OpusDet Design Analysis

We study how each configuration change contributes to the final OpusDet model. Since T-Rex2 has not released the code and weights needed for reproduction, we implement its architecture based on the published paper and train it on O365 and GoldG as the **T-Rex2-style baseline** in Step A. Table 4 reports a cumulative progression from the T-Rex2-style baseline to the final OpusDet configuration. Each row applies the listed change to the previous row.

Backbone and encoder. Table 5 factorizes step B into three substeps. Replacing Swin-T with DINOv3-ConvNeXt-B at 800×1333 (B_1) improves all prompting modes, but increases computation from 257 to 439 GFLOPs and reduces FPS_{gen} from 15.4 to 11.6. Reducing the input to 640×640 (B_2) restores efficiency at an accuracy cost. The hybrid encoder (B_3) further reduces computation to 164 GFLOPs and raises FPS_{gen} to 22.0. Together, these changes define Step B and increase FPS_{gen} by 43% over the baseline.

Prompt-aware decoder. Replacing the baseline decoder with prompt-aware decoding improves Visual-I from 57.5 to 63.0 and Text from 29.5 to 32.7, but reduces Visual-G from 35.3 to 33.3. This drop is consistent with a mismatch between Visual-I training and Visual-G inference. Visual-I uses same-image exemplars during training and evaluation, whereas Visual-G uses averaged prototypes from cross-image references. The mismatch is most severe for rare categories, whose prototypes are averaged from fewer and more diverse instances. The decoder adds 2M parameters and 1 GFLOP, changing FPS_{gen} from 22.0 to 21.3.

ICA. Rows C→D isolate ICA under O365+GoldG setting. The class-level alignment loss matches visual class prompts with text prompts of the same category. ICA adds instance-level alignment between content prompts and matched decoder queries. It improves Visual-G AP_r from 26.1 to 30.0 and Visual-I AP_r from 73.9 to 74.7. Mixed AP also increases from 36.5 to 38.0. Overall Visual-I, Visual-G, and Text AP change by at most 0.5, so the clearest gain appears in long-tailed visual prompting. Section 4.5 further analyzes this effect.

Data engine. Adding the single-pass data engine improves all four prompting modes. Visual-I increases from 63.2 to 69.2, Visual-G from 33.5 to 38.6, Text from 33.2 to 43.0, and Mixed from 38.0 to 45.2. The Text AP_r gain from 20.3 to 36.6 indicates a strong improvement in long-tailed semantic coverage. Together, ICA and the data engine address complementary difficulties in visual alignment and semantic coverage.

4.4 Grounding Supervision and Data Scaling

Data Engine Annotation Quality. The SAM3-based data engine contributes 4.38M of the 7.98M training images. We randomly sample 200 images from each of Bamboo-CLS, CC3M, and SA-1B

Table 6: **Matched and progressive data scaling on COCO and LVIS-minival. Base**: O365+GoldG; **Mid**: +OI+V3Det+HT+CH+Bamboo; **Full**: +CC3M+SA-1B(7.98M images, dataset breakdown in Table 7). Compares official baselines (top), T-Rex2-T re-impl. scaling (mid), and OpusDet scaling (bottom). T/V: Text/Visual training images.

Model	Training Data	Data Size (T/V)	COCO			LVIS-mv		
			Visual-I	Visual-G	Text	Visual-I	Visual-G	Text
PET-DINO-T	official	1.56M / 0.6M	64.0	40.3	49.8	61.8	29.6	37.8
PET-DINO-T	official	1.56M / 2.5M	64.3	38.4	-	64.5	31.5	-
T-Rex2-T	official	6.5M / 2.4M	41.1	41.1	-	40.6	38.1	-
T-Rex2-T	official	6.5M / 3.1M	56.6	38.8	45.8	59.3	37.4	42.8
T-Rex2-T (re-impl.)	Base	0.68M / 0.6M	57.2	41.3	45.8	56.3	33.6	26.5
	Mid	3.6M / 3.37M	56.6	42.1	47.3	57.2	36.6	39.0
	Full	7.98M / 4.87M	56.0	41.4	47.1	59.4	40.5	45.2
OpusDet (Swin-T)	Base	0.68M / 0.6M	61.2	37.6	45.2	64.1	34.4	33.9
	Full	7.98M / 4.87M	68.1	35.8	47.1	70.6	35.8	44.6
OpusDet	Base	0.68M / 0.6M	64.6	43.0	48.2	63.2	33.5	33.2
	Full	7.98M / 4.87M	68.1	43.4	49.6	69.2	38.6	43.0

and manually check every pseudo box–phrase pair after filtering. A box is marked incorrect if either its localization or label is wrong. Box accuracies are 91.9%, 89.5% and 91.2% on the three sources. The remaining errors mainly include nested or redundant background boxes in Bamboo-CLS and CC3M, and incorrect box–label assignments in SA-1B. Appendix C provides detailed statistics, the filtering procedure and qualitative examples.

Progressive Data Scaling. Table 6 first studies progressive data scaling using our T-Rex2-T re-implementation. On COCO, Visual-I, Visual-G, and Text change only sightly from Base to Full, moving from 57.2, 41.3, and 45.8 to 56.0, 41.4, and 47.1. On LVIS-minival, however, scaling brings large gains, boosting Visual-G to 40.5 (+6.9) and Text to 45.2 (+18.7). These results show that expanded grounding supervision mainly improves long-tailed coverage. Visual-I is already strong with O365 visual training and further improves on LVIS-minival after adding SA-1B. In contrast, the official staged T-Rex2 results in Appendix E.3 show a drop in Visual-G after adding visual data.

Data Scaling Across Frameworks. Table 6 also compares T-Rex2-T (re-impl.), OpusDet (Swin-T), and OpusDet under the matched Base and Full data settings. On LVIS-minival, both OpusDet variants already lead Visual-I at Base, and scaling from Base to Full brings a larger Visual-I gains for OpusDet (+6.0 AP) and OpusDet (Swin-T) (+6.5 AP) than for T-Rex2-T (+3.1 AP), proving that OpusDet better leverages expanded grounding supervision even with the same Swin-T backbone.

With the Full set, OpusDet reaches 68.1 Visual-I AP on COCO and over 69 AP on LVIS-minival across both configurations. However, Visual-G scales unevenly across OpusDet variants. OpusDet (Swin-T) gains only 1.4 AP on LVIS-minival and slightly drops on COCO. In contrast, the final OpusDet configuration scales effectively, boosting LVIS-minival Visual-G by 5.1 AP to 38.6 AP and reaching 43.4 AP on COCO. This recovery aligns with the benefit of the DINOv3-ConvNeXt-B backbone and hybrid encoder. Together with competitive Text performance, the final OpusDet configuration provides a more balanced profile across prompting modes.

4.5 Further Analysis

Fine-grained Category Analysis. Figure 3 breaks down the A→E progression by category frequency on LVIS-minival. Overall AP and most frequency-group AP improve throughout the progression, showing that the gains are not limited to a single category group.

The main exception appears when adding the prompt-aware decoder. It improves Text and Visual-I, but temporarily reduces Visual-G, especially for rare categories. This behavior is consistent with

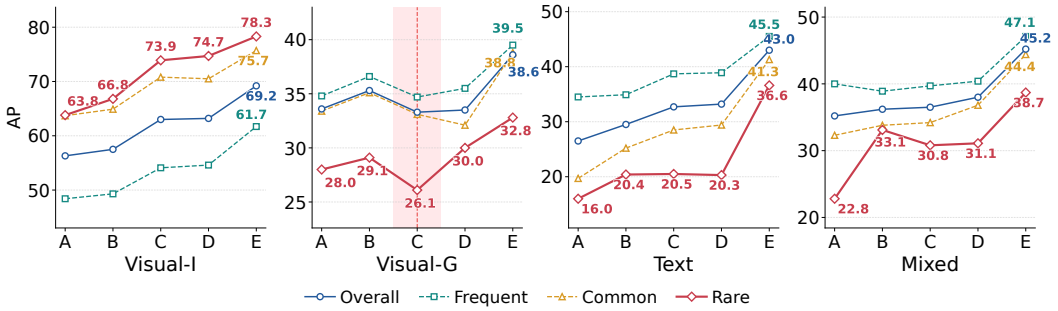

Figure 3: Overall AP and frequent, common, and rare AP on LVIS-minival under Visual-I, Visual-G, Text, and Mixed prompting across the A→E progression.

the mismatch between same-image Visual-I training and cross-image Visual-G inference. Rare categories are more sensitive because their prototypes are averaged from fewer and more diverse instances.

ICA partially recovers this drop, with the clearest gain on rare categories. The class-level alignment loss anchors class-level visual prompts to text prompts. ICA instead aligns content-level visual prompts with matched decoder queries. It therefore provides finer instance-level supervision when class-level anchors or visual prototypes are weak.

Evolving Complementarity Between Text and Visual Prompting. Table 4 also reveals how text-visual complementarity evolves across the A→E progression. Mixed prompting consistently outperforms both Text and Visual-G inference, showing that the two prompt types provide complementary cues. However, the source of this complementarity changes over time: Δ_T decreases from 8.7 to 2.2, while Δ_G increases from 1.6 to 6.6.

This trend reflects *a progressive role shift between the two prompting modalities*. At early stages, Text prompting suffers from limited long-tail semantic coverage, so Visual-G provides useful appearance-level guidance. As the data engine expands grounding supervision, Text prompting becomes much stronger and eventually surpasses Visual-G. Mixed prompting then shifts from using visual prompts to compensate for weak text semantics toward using text prompts to stabilize noisier cross-image visual prototypes. This evolving interaction explains why mixed prompting remains consistently beneficial, and also exposes a limitation of current generic visual prompting, suggesting that the Visual-G paradigm itself deserves further study, including reference selection, prototype aggregation, and prototype denoising.

5 Conclusion

We presented OpusDet, a simple framework for unified open-vocabulary detection. OpusDet supports Text, Visual-I, Visual-G, and Mixed prompting in one detector. It combines a semantic-rich encoder and prompt-aware decoder with one-stage text–visual training, Instance-level Contrastive Alignment, and a SAM3-based single-pass data engine. This design avoids heavy cross-modal fusion, staged training, and iterative relabeling. OpusDet achieves state-of-the-art Visual-I performance among comparable unified detectors on COCO, LVIS-minival, and ODinW35. It also maintains competitive Text and Visual-G performance. Under matched Base and Full settings, OpusDet converts expanded grounding supervision into larger Visual-I gains than our T-Rex2-style reimplementation. Mixed prompting also outperforms either individual prompt mode.

Our analysis further shows that this complementarity evolves with grounding supervision. As Text prompting recovers long-tailed semantic coverage, Visual-G increasingly relies on textual semantics in the mixed setting, exposing the limitation of noisy cross-image category prototypes. Improving generic visual prompting through better reference selection, prototype aggregation, and prototype denoising remains an important direction for future unified OVD systems. Overall, OpusDet demonstrates that simplicity and strong unified prompting capability can be achieved together.

REPRODUCIBILITY STATEMENT

The model architecture and training objectives are described in Section 3, with forward pass, alternating training loop, and inference given as algorithms in Appendix B. Full hyperparameters, including optimizer settings, schedule, loss weights, and the visual-to-text batch ratio, are listed in Appendix D.1, and the evaluation protocol for each prompting mode is specified in Appendix D.2. The SAM3-based data engine is the component that is hardest to reproduce from the paper alone, so Appendix C reports the per-source preprocess filtering and post-processing stages, the resulting dataset sizes, and a manual audit of pseudo-label accuracy with representative success and failure cases. Because T-Rex2 releases neither code nor weights, the baseline used in our component-wise and data-scaling studies is our own re-implementation; its training configuration is reported alongside ours so that the comparison can be repeated. Code, configuration files, and trained weights will be released.

AI USE STATEMENT

We used generative AI tools to aid and polish the writing of this paper, including improving grammar, clarity, and presentation. We also used AI-assisted retrieval and discovery tools to identify potentially relevant related work. All cited papers were independently read and verified by the authors, who made the final decisions regarding citation and characterization. AI tools were not used to generate experimental results, or reported numbers. All authors reviewed the final manuscript and take full responsibility for its content.

REFERENCES

Niki Amini-Naieni, Tengda Han, and Andrew Zisserman. Countgd: Multi-modal open-world counting. *Advances in Neural Information Processing Systems*, 37:48810–48837, 2024.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts, 2025. URL https://arxiv.org/abs/2511.16719.

Qibo Chen, Weizhong Jin, Jianyue Ge, Mengdi Liu, Yuchao Yan, Jian Jiang, Li Yu, Xuanjiang Guo, Shuchang Li, and Jianzhong Chen. Cp-detr: Concept prompt guide detr toward stronger universal object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 2141–2149, 2025.

Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world: Real-time open-vocabulary object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16901–16911, 2024.

Shenghao Fu, Yukun Su, Fengyun Rao, Jing Lyu, Xiaohua Xie, and Wei-Shi Zheng. Wedetect: Fast open-vocabulary object detection as retrieval. *arXiv preprint arXiv:2512.12309*, 2025a.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14987–14997, 2025b.

Weifu Fu, Jinyang Li, Bin-Bin Gao, Jialin Li, Yuhuan Lin, Hanqiu Deng, Wenbing Tao, Yong Liu, and Chengjie Wang. Pet-dino: Unifying visual cues into grounding dino with prompt-enriched training. *arXiv preprint arXiv:2604.00503*, 2026.

Yuchen Guan, Chong Sun, Canmiao Fu, Zhipeng Huang, Chun Yuan, and Chen Li. Text-guided visual prompt dino for generic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 21288–21298, 2025.

Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5356–5364, 2019.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength natural language processing in Python. Zenodo, 2020. URL https://doi.org/10.5281/zenodo.1212303.

Qing Jiang, Feng Li, Tianhe Ren, Shilong Liu, Zhaoyang Zeng, Kent Yu, and Lei Zhang. T-rex: Counting by visual prompting. *arXiv preprint arXiv:2311.13596*, 2023.

Qing Jiang, Feng Li, Zhaoyang Zeng, Tianhe Ren, Shilong Liu, and Lei Zhang. T-rex2: Towards generic object detection via text-visual prompt synergy. In *European Conference on Computer Vision*, pp. 38–57. Springer, 2024.

Qing Jiang, Feng Li, Zhaoyang Zeng, Tianhe Ren, Shilong Liu, and Lei Zhang. T-Rex-Omni: Integrating negative visual prompt in generic object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026.

Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetr-modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1780–1790, 2021.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4015–4026, 2023.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, 128:1956–1981, 2020.

Chunyuan Li, Haotian Liu, Liunian Harold Li, Pengchuan Zhang, Jyoti Aneja, Jianwei Yang, Ping Jin, Houdong Hu, Zicheng Liu, Yong Jae Lee, and Jianfeng Gao. Elevater: A benchmark and toolkit for evaluating language-augmented visual models. *Advances in Neural Information Processing Systems*, 35:9287–9301, 2022a.

Feng Li, Qing Jiang, Hao Zhang, Tianhe Ren, Shilong Liu, Xueyan Zou, Huaizhe Xu, Hongyang Li, Jianwei Yang, Chunyuan Li, Lei Zhang, and Jianfeng Gao. Visual in-context prompting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12861–12871, 2024.

Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10965–10975, 2022b.

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.

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

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11976–11986, 2022.

Shangbang Long, Siyang Qin, Dmitry Panteleev, Alessandro Bissacco, Yasuhisa Fujii, and Michalis Raptis. Towards end-to-end unified scene text detection and layout analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1049–1059, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, Xiao Wang, Xiaohua Zhai, Thomas Kipf, and Neil Houlsby. Simple open-vocabulary object detection with vision transformers. In *European Conference on Computer Vision*, pp. 728–755. Springer, 2022.

Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. In *Advances in Neural Information Processing Systems*, 2023.

Ting Pan, Lulu Tang, Xinlong Wang, and Shiguang Shan. Tokenize anything via prompting. In *European Conference on Computer Vision*, pp. 330–348. Springer, 2024.

Bo Qian, Dahu Shi, and Xing Wei. Detr-vip: Detection transformer with robust discriminative visual prompts. *arXiv preprint arXiv:2604.14684*, 2026.

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

Tianhe Ren, Yihao Chen, Qing Jiang, Zhaoyang Zeng, Yuda Xiong, Wenlong Liu, Zhengyu Ma, Junyi Shen, Yuan Gao, Xiaoke Jiang, Xingyu Chen, Zhuheng Song, Yuhong Zhang, Hongjie Huang, Han Gao, Shilong Liu, Hao Zhang, Feng Li, Kent Yu, and Lei Zhang. DINO-X: A unified vision model for open-world object detection and understanding. *arXiv preprint arXiv:2411.14347*, 2024.

Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. Crowdhuman: A benchmark for detecting human in a crowd. *arXiv preprint arXiv:1805.00123*, 2018.

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 *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8430–8439, 2019.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2556–2565, 2018.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3, 2025. URL https://arxiv.org/abs/2508.10104.

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

Hao Wang, Pengzhen Ren, Zequn Jie, Xiao Dong, Chengjian Feng, Yinlong Qian, Lin Ma, Dongmei Jiang, Yaowei Wang, Xiangyuan Lan, and Xiaodan Liang. Ov-dino: Unified open-vocabulary detection with language-aware selective fusion, 2024. URL https://arxiv.org/abs/2407.07844.

Jiaqi Wang, Pan Zhang, Tao Chu, Yuhang Cao, Yujie Zhou, Tong Wu, Bin Wang, Conghui He, and Dahua Lin. V3det: Vast vocabulary visual detection dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19844–19854, 2023.

Junfeng Wu, Yi Jiang, Qihao Liu, Zehuan Yuan, Xiang Bai, and Song Bai. General object foundation model for images and videos at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3783–3795, 2024.

Yongjian Wu, Yang Zhou, Jiya Saiyin, Bingzheng Wei, and Yan Xu. Visual textualization for image prompted object detection. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20900–20910. IEEE, 2025.

Yifan Xu, Mengdan Zhang, Chaoyou Fu, Peixian Chen, Xiaoshan Yang, Ke Li, and Changsheng Xu. Multi-modal queried object detection in the wild. *Advances in Neural Information Processing Systems*, 36:4452–4469, 2023.

Lewei Yao, Renjie Pi, Jianhua Han, Xiaodan Liang, Hang Xu, Wei Zhang, Zhenguo Li, and Dan Xu. Detclipv3: Towards versatile generative open-vocabulary object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 27391–27401, 2024.

Hao Zhang, Feng Li, Xueyan Zou, Shilong Liu, Chunyuan Li, Jianwei Yang, and Lei Zhang. A simple framework for open-vocabulary segmentation and detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1020–1031, 2023.

Yuanhan Zhang, Qinghong Sun, Yichun Zhou, Zexin He, Zhenfei Yin, Kun Wang, Lu Sheng, Yu Qiao, Jing Shao, and Ziwei Liu. Bamboo: Building mega-scale vision dataset continually with human–machine synergy. *International Journal of Computer Vision*, 133(8):5806–5821, 2025.

Xiangyu Zhao, Yicheng Chen, Shilin Xu, Xiangtai Li, Xinjiang Wang, Yining Li, and Haian Huang. An open and comprehensive pipeline for unified object grounding and detection. *arXiv preprint arXiv:2401.02361*, 2024a.

Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. Detrs beat yolos on real-time object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16965–16974, 2024b.

A Limitations

Although OpusDet achieves strong overall performance, generic visual prompting remains less improved than text and interactive visual prompting. Our training uses Visual-I-style prompts sampled from ground-truth instances, whereas Visual-G inference relies on cross-image category prototypes aggregated from reference images. This training-inference mismatch makes Visual-G more sensitive to reference quality and prototype noise, especially for rare categories. Our analysis suggests that current generic visual prompting still requires more targeted modeling, such as better reference image selection, prototype aggregation, and prototype denoising.

In addition, our SAM3-based data engine is designed to scale heterogeneous grounding supervision in a single pass, but its effectiveness depends on the quality and domain coverage of the underlying promptable model. We validate OpusDet on standard natural-image detection benchmarks, including COCO, LVIS-minival, and ODinW35, but have not evaluated the annotation pipeline on specialized domains such as medical, industrial, or remote-sensing imagery. Extending and validating the pipeline in such domains remains future work.

B Method Details

B.1 Forward Pass and Training Objective

Algorithm 1 summarizes the forward propagation and loss computation of our unified detector during training. Given an input image, the model encodes category prompts with the text encoder and extracts multi-scale visual features through the backbone and hybrid encoder. Under the VISUAL mode, the Visual Prompt Encoder (VPE) produces class- and content-level visual prompt features,

Algorithm 1 Unified forward pass and training objective.

Require: Image I, annotations $\mathcal{Y}$, text prompts P^t, sampled visual prompts V with source-GT indices $\mathcal{S}_V$, mode $m \in \{\text{TEXT}, \text{VISUAL}\}$
Ensure: $\mathcal{L} = \mathcal{L}_{\text{det}} + \mathcal{L}_{\text{align}} + \mathcal{L}_{\text{ica}}$
$\{F_\ell\}_{\ell\in\{3,4,5\}} \leftarrow \text{BACKBONE}(I)$
$\{F_\ell\} \leftarrow \text{CHANNELMAPPER}(\{F_\ell\})$
$F \leftarrow \text{HYBRIDENCODER}(\{F_\ell\})$
if $m = \text{VISUAL}$ **then**
  $(P^v_{\text{cls}}, P^v_{\text{con}}) \leftarrow \text{VPE}(F, V)$
  $P \leftarrow P^v_{\text{cls}}$ {use visual prompts}
  $\mathcal{A} \leftarrow$ category correspondence between P^t and P^v_{cls}
else
  $P \leftarrow P^t$ {use text prompts}
end if
$(Q_0, R_0) \leftarrow \text{TOPKPROPOSALS}(F, P, N_q)$
$Q_0 \leftarrow \text{CONCAT}(\text{BUILDDNQUERIES}(\mathcal{Y}, P), Q_0)$
for $l = 1$ to 6 **do**
  $Q_l \leftarrow \text{SELFATTN}(Q_{l-1})$
  $Q_l \leftarrow \text{CROSSATTNPROMPT}(Q_l, P)$
  $Q_l \leftarrow \text{CROSSATTNIMAGE}(Q_l, F, R_{l-1})$
  $Q_l \leftarrow \text{FFN}(Q_l)$
  $R_l \leftarrow \text{REFINEREFERENCEPOINTS}(Q_l, R_{l-1})$
end for
Split Q_l into $Q^{\text{match}}, Q^{\text{dn}}$
$(\hat{y}, \hat{b}), (\hat{y}_{\text{dn}}, \hat{b}_{\text{dn}}) \leftarrow \text{DETHEAD}(Q^{\text{match}}, Q^{\text{dn}}, P)$
$\Pi \leftarrow \text{HUNGARIANMATCH}(\hat{y}, \hat{b}, \mathcal{Y})$
$\mathcal{L}_{\text{det}} \leftarrow \text{DETLOSS}(\hat{y}, \hat{b}, \hat{y}_{\text{dn}}, \hat{b}_{\text{dn}}, \mathcal{Y}, \Pi)$
if $m = \text{VISUAL}$ **then**
  $\mathcal{L}_{\text{align}} \leftarrow \text{ALIGNLOSS}(P^t, P^v_{\text{cls}}; \mathcal{A})$
  $\mathcal{C} \leftarrow \{(Q^{\text{match}}_{\Pi(s)}, P^v_{\text{con},s}) \mid s \in \mathcal{S}_V \cap \text{dom}(\Pi)\}$ {sampled GTs with matched queries}
  $\mathcal{L}_{\text{ica}} \leftarrow \text{SYMINFONCE}(\mathcal{C})$ {other valid prompts are negatives}
else
  $\mathcal{L}_{\text{align}}, \mathcal{L}_{\text{ica}} \leftarrow 0$
end if
return $\mathcal{L}_{\text{det}} + \mathcal{L}_{\text{align}} + \mathcal{L}_{\text{ica}}$

Algorithm 2 Alternating text-visual training loop.

Require: text dataloader $\mathcal{D}_{\text{text}}$, visual dataloader $\mathcal{D}_{\text{vis}}$, total iterations $T_{\max}$, alternating ratio 1:8 (VISUAL : TEXT)
Ensure: Parameters Θ
for $t = 0$ **to** $T_{\max} - 1$ **do**
  if $t \bmod 9 = 8$ **then**
    $(I, \mathcal{Y}) \sim \mathcal{D}_{\text{vis}}$
    mode $\leftarrow$ VISUAL
  else
    $(I, \mathcal{Y}) \sim \mathcal{D}_{\text{text}}$
    mode $\leftarrow$ TEXT
  end if
  $\mathcal{L} \leftarrow \text{FORWARDANDLOSS}(I, \mathcal{Y}, \textit{mode}; \Theta)$ {Algorithm 1}
  $\Theta \leftarrow \text{ADAMWUPDATE}(\Theta, \nabla_\Theta \mathcal{L})$
end for

and the decoder consumes $P{=}P^v_{\text{cls}}$ (text mode uses $P{=}P^t$). The decoder adopts alternating self-attention, prompt cross-attention, and image cross-attention for iterative query refinement. Following the two-stage query initialization and denoising strategy, the model computes both matching and denoising detection losses. Additional alignment and instance-level contrastive objectives are introduced only for visual prompt training.

B.2 ALTERNATING TRAINING LOOP

Algorithm 3 Open-vocabulary inference under text, visual, and mixed prompts.

Require: input image I; mode m; text prompts P^t; visual prompts V {V: same-image exemplars (Visual-I) or cached mean prototypes (Visual-G)}
Ensure: detections $\{(b_j, s_j, c_j)\}_{j=1}^{N}$
$F \leftarrow \text{ENCODER}(I)$
if $m = \text{TEXT}$ **then**
  $P \leftarrow P^t$
else
  $(P^v_{\text{cls}}, P^v_{\text{con}}) \leftarrow \text{VPE}(F, V)$ **or** use cached P^v_{cls} {Visual-G: offline mean over N reference prompts}
  if $m \in \{\text{VISUAL-I}, \text{VISUAL-G}\}$ **then**
    $P \leftarrow P^v_{\text{cls}}$
  else
    $P \leftarrow (P^t + P^v_{\text{cls}})/2$ {mixed; no learned fusion}
  end if
end if
$(Q_0, R_0) \leftarrow \text{TOPKPROPOSALS}(F, P, N_q)$
$Q, R \leftarrow \text{DECODER}(Q_0, P, F, R_0)$
$\{(b_j, s_j, c_j)\} \leftarrow \text{DETHEAD.PREDICT}(Q, P)$
return $\{(b_j, s_j, c_j)\}$

Table 7: Training datasets in OpusDet. †: annotations from the SAM3 single-pass data engine.

Dataset	Prompt Type	Images	Ratio
Objects365 (Shao et al., 2019)	Text / Visual	0.6M	7.5%
OpenImages (Kuznetsova et al., 2020)	Text / Visual	1.67M	20.9%
GoldG (Kamath et al., 2021)	Text	76K	1.0%
V3Det (Wang et al., 2023)	Text	0.15M	1.9%
CrowdHuman (Shao et al., 2018)	Text / Visual	15K	0.2%
HierText (Long et al., 2022)	Text / Visual	8K	0.1%
Bamboo-Det (Zhang et al., 2025)	Text / Visual	1.08M	13.5%
Bamboo-CLS† (Zhang et al., 2025)	Text	1.56M	19.6%
CC3M† (Sharma et al., 2018)	Text	1.32M	16.5%
SA-1B† (Kirillov et al., 2023)	Text / Visual	1.5M	18.8%
Total	Text / Visual	7.98M	100.0%

Algorithm 2 presents the alternating optimization strategy used in training. To jointly support text prompting and visual prompting learning under a unified detector, we alternate between visual batches and text prompt batches with a ratio of 1:8. Text batches optimize conventional open-vocabulary detection objectives using enhanced category prompts and sampled negative categories, while visual-prompt batches use VPE features as P and enable auxiliary visual prompt losses. All parameters are optimized end-to-end using AdamW.

B.3 Inference

Algorithm 3 describes the unified inference workflow supporting four inference modes: text prompting, interactive visual prompting, generic visual prompting, and mixed prompting. Depending on the mode, the detector selects $P{=}P^t$, $P{=}P^v_{\text{cls}}$, or $P{=}(P^t + P^v_{\text{cls}})/2$ as in Equation 3; Mixed uses slot-wise averaging with no learned fusion module. The decoder then performs iterative object query refinement to produce final open-vocabulary detections. For generic visual prompting, visual embeddings can be precomputed and cached to improve inference efficiency.

C Data Engine

Section 3 describes our SAM3-based single-pass pipeline that converts text-image, classification, and segmentation sources into Bamboo-CLS, CC3M, and annotated SA-1B supervision, respectively. Here we detail the filtering stages and show qualitative cases. Dataset statistics are summarized in Table 7.

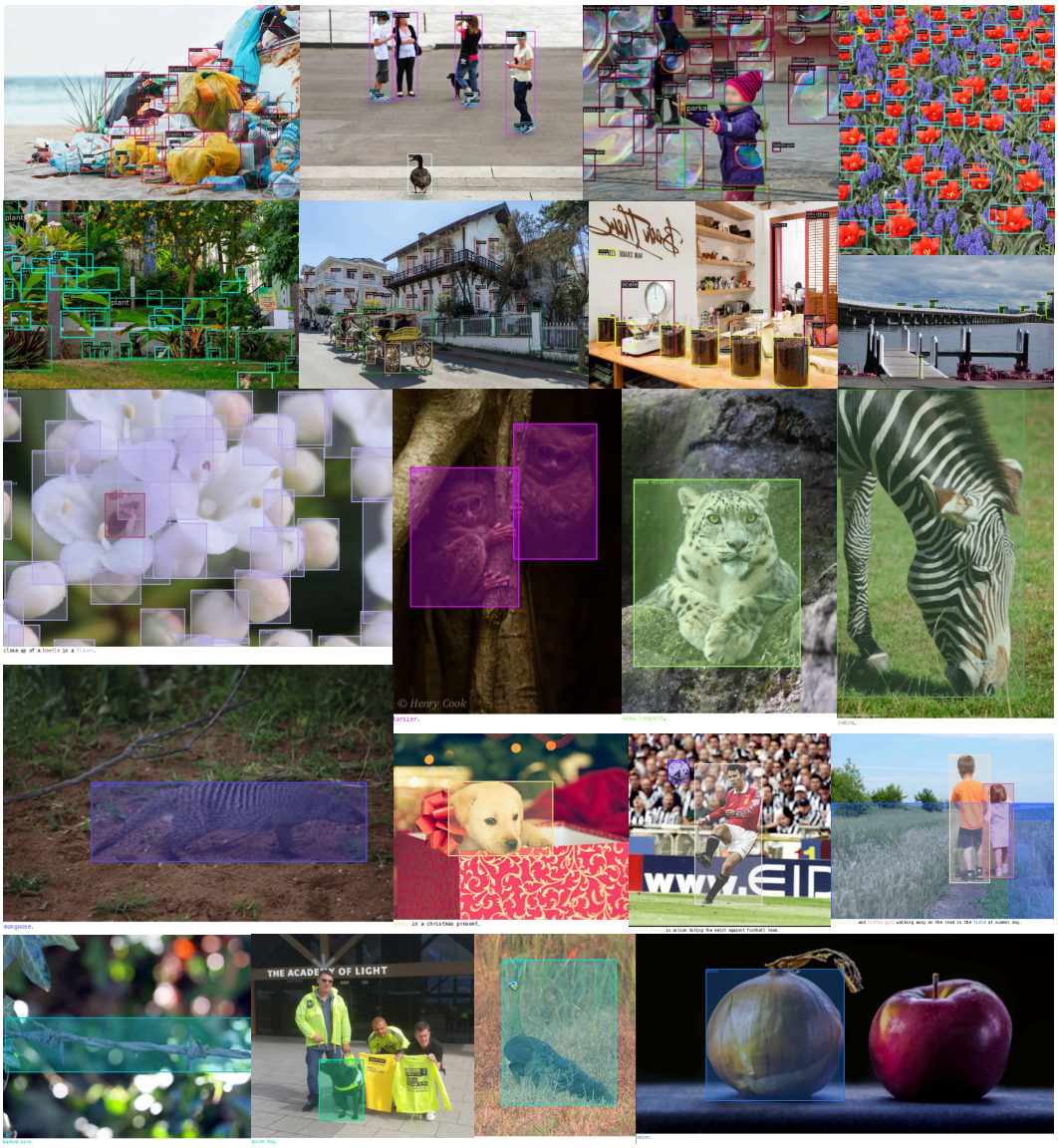

Figure 4: SAM3 pseudo-label success cases after filtering (columns: Bamboo-CLS, CC3M, SA-1B).

Table 8: Human verification of SAM3 pseudo-labels after filtering. A box is incorrect if localization or the label is wrong.

Source	#Images	#Boxes	Box Acc. (%)
Bamboo-CLS	200	653	91.9
CC3M	200	1291	89.5
SA-1B	200	7897	91.2

Human verification. We randomly sample 200 images from each generated source and manually verify every pseudo box–phrase pair after filtering. A box counts as incorrect if localization or the assigned label is wrong. Table 8 reports box accuracies near 90% on Bamboo-CLS, CC3M, and SA-1B.

Source-specific preprocess filtering. Different sources introduce different noise. For CC3M, noun phrases extracted from captions are used only as candidate text prompts and must be visually grounded by SAM3, which removes many ungroundable phrases. For Bamboo-CLS, image-

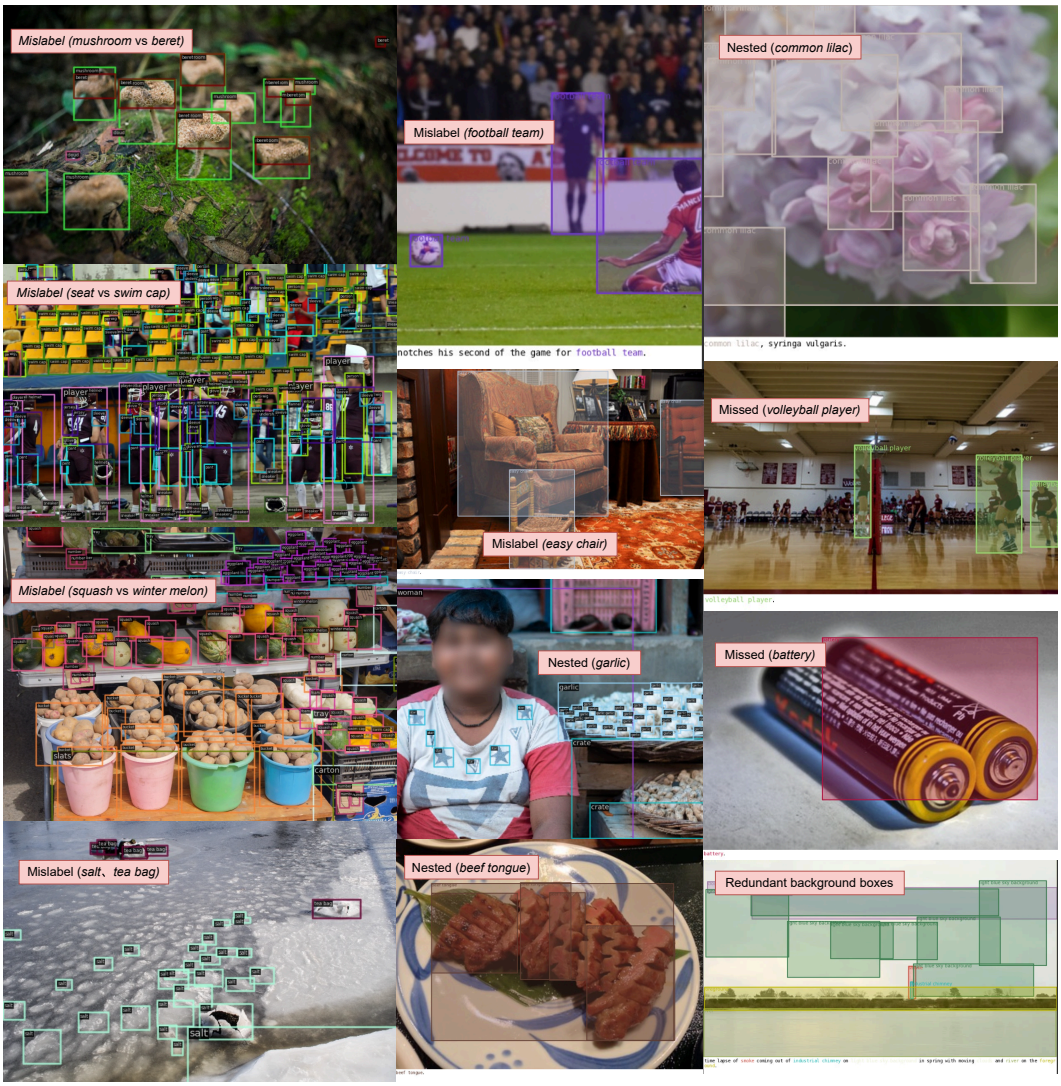

Figure 5: SAM3 pseudo-label failure cases after filtering (columns: Bamboo-CLS, CC3M, SA-1B): mislabels, nested boxes, and redundant background boxes.

level labels can be coarser than object regions or cover only part of the visible content; subsequent grounding and filtering mitigate this mismatch while retaining scalable supervision. For SA-1B, we first obtain semantic labels with TAP over a vocabulary of 2560 common categories and discard unreliable tags with category-dependent confidence thresholds, then use the retained labels as prompts for SAM3 to complete category-level boxes.

Unified post-processing. All generated boxes then pass through a shared post-processing stage: confidence thresholding, category-wise and image-wise NMS, removal of near full-image boxes, and filtering of large boxes that enclose many small same-category instances.

Qualitative cases. Figure 4 and Figure 5 show representative success and failure cases after filtering, with one column each for Bamboo-CLS, CC3M, and SA-1B. Successful examples typically produce tight object-level boxes with consistent labels. Residual failures include mislabeled boxes, nested boxes, and redundant background boxes on regions without clear object features (e.g., sky). On Bamboo-CLS and CC3M, nested or redundant boxes are more common; on SA-1B, mislabels appear more often.

Table 9: Training configuration.

Parameter	Value
Backbone	DINOv3-ConvNeXt-B
Text encoder	CLIP ViT-B/32
Image size	640×640
Batch size per GPU	8
GPUs	16
Global batch size	128
Training iterations	400K
Alternating ratio	1:8 (VISUAL:TEXT)
Optimizer	AdamW
Base learning rate	4×10^{-4}
Backbone / text encoder LR	4×10^{-5}
Weight decay	1×10^{-4}
Gradient clipping	0.1
Warmup	1K iterations, start factor 0.1
LR scheduler	MultiStep (50%, 90%; $\gamma = 0.1$)
Precision	FP16 (AMP)
Detection queries	900
DN queries	100
Max categories per image	256
Loss weights	cls / bbox / iou / align / ica
	1.0 / 5.0 / 2.0 / 1.0 / 1.0
Align loss	Cross-entropy w/ learnable log-scale
ICA temperature	InfoNCE ($\tau = 0.07$)
Validation interval	20K iterations

D Experimental Setup

D.1 Training Hyperparameters

Our training hyperparameter configuration is summarized in Table 9. Unless otherwise specified, all models are trained under the same optimization setting, including input resolution, learning-rate schedule, alternating training ratio, and loss weighting strategy, to ensure fair comparison under a fixed efficiency budget.

D.2 Evaluation Protocol

D.2.1 Accuracy

Text Protocol. Under the text prompt protocol, category names are used as textual inputs and encoded by a pretrained text encoder. The resulting text embeddings are projected into the shared prompt space and used as category prompts for open-vocabulary detection. This protocol evaluates the detector's ability to recognize categories from language descriptions alone.

Visual-I Protocol. Under the interactive visual prompt protocol, one ground-truth instance is randomly selected for each category in a test image and used as the visual prompt. The model then detects all instances of the corresponding categories in the same image. This setting simulates interactive applications such as annotation and object counting, where users provide instance-level guidance at inference time.

Visual-G Protocol. Under the generic visual prompt protocol, category-level visual prompts are constructed offline from reference images. For each category, we randomly sample a fixed number of training images, encode their visual prompts, and average them into a category-level visual prototype. The resulting prototypes are reused across all test images during inference. This proto-

Table 10: Visual-G and Mixed prompting vs. number of references (N) on the final OpusDet model. Text-only: 49.6 AP (COCO) / 43.0 AP (LVIS-minival). Bold marks the default N=16.

N	Prompt	COCO	LVIS-mv			
		AP	AP	AP_f	AP_c	AP_r
1	Visual-G	27.6	25.0	25.6	25.4	20.0
4	Visual-G	38.6	34.5	35.1	34.8	28.8
8	Visual-G	41.6	37.1	38.0	36.9	33.6
16	Visual-G	**43.4**	**38.6**	**39.5**	**38.8**	**32.8**
32	Visual-G	44.3	39.7	40.1	40.7	32.3
64	Visual-G	44.7	39.7	40.4	40.3	33.0
1	Mixed	46.2	40.5	42.6	39.4	34.2
4	Mixed	49.1	44.7	46.1	44.5	38.6
8	Mixed	49.4	45.1	46.7	44.2	39.9
16	Mixed	**49.9**	**45.2**	**47.1**	**44.4**	**38.7**
32	Mixed	50.2	45.4	47.2	44.8	37.8
64	Mixed	50.3	45.6	47.4	45.0	38.6

col evaluates whether visual prompts can serve as generic category descriptions without per-image interaction.

Mixed Protocol (Text + Visual-G). Under the mixed protocol, text prompt embeddings and Visual-G prototypes are independently constructed and jointly used as category prompts during inference. The two embeddings are averaged at matched class slots without a learned fusion module. This setting evaluates whether semantic text cues and appearance-level visual cues can complement each other in a unified detector.

D.2.2 Efficiency

Interactive Speed ($\mathrm{FPS}_{\mathrm{inter}}$). This metric measures inference throughput in the interactive visual prompting setting. Following prior work, image features are cached for each test image, and the measured runtime includes visual prompt encoding and decoder inference for each user interaction. This reflects the latency-sensitive use case where users iteratively provide visual prompts on the same image.

Generic Speed ($\mathrm{FPS}_{\mathrm{gen}}$). This metric measures inference throughput for Text and Visual-G prompting. In this setting, category prompt embeddings are precomputed offline and reused across test images. The measured runtime therefore includes image feature extraction and decoder inference, but excludes repeated prompt encoding. This reflects standard generic detection, where a fixed set of category prompts is applied to many images.

E Additional Results

E.1 Sensitivity Analysis

We study sensitivity to Visual-G reference sampling (N), the visual:text alternating ratio, and ICA temperature τ.

E.1.1 Visual-G Reference Sampling

Table 10 varies $N \in \{1, 4, 8, 16, 32, 64\}$ to verify the default N=16 choice. Visual-G needs enough references for stable category-level prototypes, rising from 25.0 AP at N=1 to 38.6 AP at N=16 on LVIS-minival, and then saturates beyond N=16. For rare categories, the training set may contain fewer usable images than N, so larger N does not always mean more distinct references. Under Mixed prompting, text provides an additional anchor, so performance varies less with N and already exceeds Text-only from N=4 on LVIS-minival and from N=16 on COCO. Five random seeds at the default N=16 confirm that this choice is stable: Visual-G is 43.3 ± 0.1 on COCO and 38.6 ± 0.3 on LVIS-minival, and Mixed is 49.9 ± 0.1 and 45.5 ± 0.2.

Table 11: Sensitivity to the visual:text alternating ratio on LVIS-minival.

visual:text	Visual-I	Visual-G	Text	Mixed
1:4	**64.6**	31.5	31.6	36.8
1:8	63.2	33.5	33.2	38.0
1:16	58.8	**34.6**	**33.7**	**38.5**

Table 12: ICA temperature sensitivity (τ) on LVIS-minival.

τ	Visual-I	Visual-G	Text	Mixed
0.05	63.3	33.7	32.7	38.4
0.07	63.2	33.5	33.2	38.0
0.10	63.1	33.4	33.2	38.5

E.1.2 VISUAL:TEXT ALTERNATING RATIO

Table 11 ablates the visual:text alternating ratio on the ICA-equipped model under the O365+GoldG setting (main-text Table 4, step D). A larger visual share (1:4) favors Visual-I, while a larger text share (1:16) favors Text, Visual-G, and Mixed. No single ratio dominates all modes, so we use 1:8 as the default in the main experiments.

E.1.3 ICA TEMPERATURE

Table 12 varies the ICA temperature τ under the same setting as Table 11, with $w_{\mathrm{ICA}}{=}1.0$. Across $\tau \in \{0.05, 0.07, 0.10\}$, Visual-I, Text, and Mixed change only slightly, and Visual-G stays within 0.3 AP. We therefore keep $\tau{=}0.07$ as the default.

E.2 FINE-GRAINED COMPONENT ABLATION

Table 13 keeps the same cumulative configurations as Table 4, and breaks each step into AP_f/AP_c/AP_r. Overall, the progression from the baseline to the full OpusDet model improves performance across most category groups and prompting modes, showing that the gains are not limited to frequent categories.

A notable exception appears when replacing the baseline decoder with prompt-aware decoding in Row C. While the decoder improves Text and Visual-I prompting, it temporarily reduces Visual-G performance, with the largest drop on rare categories. This supports our interpretation that Visual-G is more sensitive to the mismatch between Visual-I-style training prompts and cross-image category prototypes used during Visual-G inference. Rare categories are particularly affected because their prototypes are aggregated from fewer and more diverse examples, making them less stable than frequent-category prototypes.

Adding ICA in Row D produces a selective recovery pattern under Visual-G. Although the overall Visual-G AP recovers only modestly, AP_r improves substantially from 26.1 to 30.0, while AP_f gains slightly and AP_c remains below the pre-decoder level. This pattern suggests that ICA is particularly useful when category-level visual prototypes are unreliable, especially for rare categories.

To understand this effect, recall that the base alignment objective $\mathcal{L}_{\mathrm{align}}$ operates at the class level. It trains the aggregated visual prompt feature $\mathbf{P}^v_{\mathrm{cls}}$ by pulling it toward the corresponding text prompt feature $\mathbf{P}^t$. While this alignment encourages text-visual category consistency, it also makes Visual-G prototypes depend on a global semantic anchor. Such an anchor is coarse by design: text features mainly encode category-level semantics rather than instance-level appearance variations. This limitation becomes more pronounced for rare categories, where both the text representations and the cross-image visual prototypes receive less reliable supervision. As a result, aligning Visual-G features primarily through $\mathcal{L}_{\mathrm{align}}$ may weaken fine-grained visual details and make rare-category prototypes inherit the limitations of the text modality.

ICA addresses this issue from a different direction. Instead of aligning only the aggregated class-level visual prototype to text, ICA aligns instance-level visual prompt features $\mathbf{P}^v_{\mathrm{con}}$ directly with their matched decoder queries. These decoder queries are grounded in image regions through de-

Table 13: $AP_f/AP_c/AP_r$ breakdown of the A→E ablation on LVIS-minival (four prompting modes; cf. Table 4).

ID	Visual-I				Visual-G				Text				Mixed			
	AP	**AP$_f$**	**AP$_c$**	**AP$_r$**	**AP**	**AP$_f$**	**AP$_c$**	**AP$_r$**	**AP**	**AP$_f$**	**AP$_c$**	**AP$_r$**	**AP**	**AP$_f$**	**AP$_c$**	**AP$_r$**
A	56.3	48.4	63.7	63.8	33.6	34.8	33.4	28.0	26.5	34.5	19.7	16.0	35.2	40.0	32.3	22.8
B	57.5	49.3	64.9	66.8	35.3	36.6	35.1	29.1	29.5	34.9	25.2	20.4	36.2	38.9↓	33.8	33.1
C	63.0	54.1	70.8	73.9	33.3↓	34.7↓	33.1↓	26.1↓	32.7	38.7	28.5	20.5	36.5	39.7	34.2	30.8↓
D	63.2	54.6	70.5↓	74.7	33.5	35.5	32.1↓	30.0	33.2	38.9	29.4	20.3↓	38.0	40.4	36.8	31.1
E	**69.2**	**61.7**	**75.7**	**78.3**	**38.6**	**39.5**	**38.8**	**32.8**	**43.0**	**45.5**	**41.3**	**36.6**	**45.2**	**47.1**	**44.4**	**38.7**

Table 14: Official T-Rex2-T second-stage visual-prompt training results from T-Rex2 Table 6. Compare with our one-stage progressive scaling in Table 6.

Model	Training data	COCO			LVIS-mv		
		Visual-I	Visual-G	Text	Visual-I	Visual-G	Text
T-Rex2-T	O365, OI, HT, CH	41.1	41.1	–	40.6	38.1	–
	O365, OI, HT, CH, SA-1B	56.6	38.8	45.8	59.3	37.4	42.8

tection supervision, and therefore provide a more localized and appearance-sensitive target than text embeddings. This instance-query alignment helps preserve fine-grained visual information without being constrained solely by the semantic coarseness of the text anchor. It also provides a stronger learning signal for rare categories, where class-level text-visual alignment is least reliable. This explains why ICA yields its clearest Visual-G recovery on AP_r rather than uniformly improving all frequency groups.

E.3 Comparison with Staged Visual-Prompt Training

Table 14 recalls official T-Rex2-T second-stage visual-prompt results from T-Rex2 Table 6. These numbers are a reference comparison against reported results, not a controlled re-training under the same implementation.

With only O365 for visual-prompt training, our one-stage re-impl. already reaches Visual-I 57.2/56.3 on COCO/LVIS-minival (Table 6), while official staged training reports 41.1/40.6 before SA-1B and 56.6/59.3 after. Under Visual-G, adding SA-1B lowers official T-Rex2-T from 41.1 to 38.8 on COCO and from 38.1 to 37.4 on LVIS-minival. The one-stage re-impl. instead stays flat on COCO (41.3→41.4) and rises on LVIS-minival (33.6→40.5).

E.4 Visualizations

We provide qualitative results for interactive visual prompting, generic visual prompting, and text prompting. These examples complement the quantitative results by showing how OpusDet behaves across dense scenes, cross-image exemplar transfer, and open-vocabulary text-guided detection.

Figure 6 compares OpusDet with T-Rex2 and PET-DINO under interactive visual prompting. OpusDet demonstrates stronger visual prompt understanding for both single-category and multi-category targets in diverse and dense scenes.

Figure 7 further compares OpusDet with T-Rex2 on challenging interactive visual prompting cases. OpusDet produces fewer background false positives and more accurate instance localization.

Figure 8 shows generic visual prompting results, where exemplar prompts from reference images are transferred to different test images. OpusDet effectively preserves category-level visual semantics across changes in scene layout, object scale, and instance appearance.

Figure 9 presents text-prompted zero-shot detection examples.

Together, these visualizations show that OpusDet provides robust prompt understanding across text, interactive visual, and generic visual prompting settings.



Figure 6: Interactive visual prompting: OpusDet vs. T-Rex2 and PET-DINO on single- and multi-category targets in diverse dense scenes.

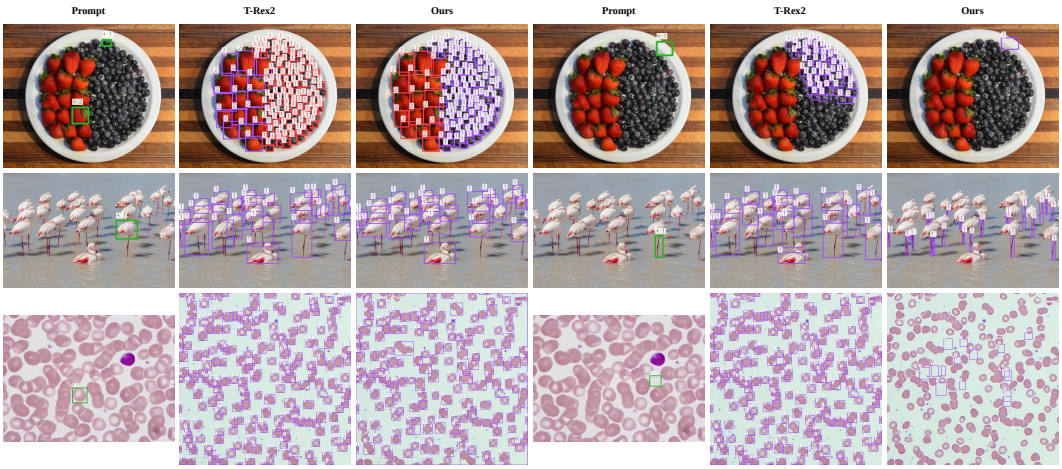

Figure 7: Challenging interactive visual detection cases: OpusDet vs. T-Rex2, with tighter localization and fewer false positives.

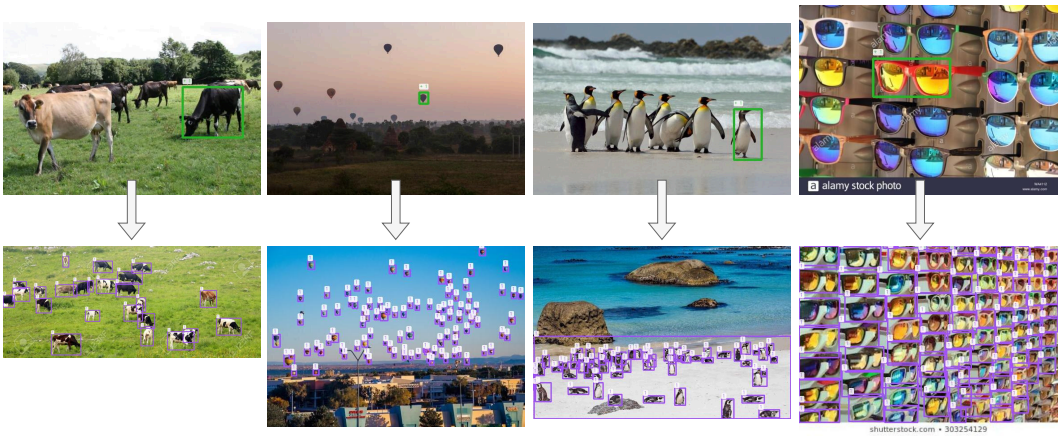

Figure 8: Generic visual prompting with cross-image exemplars (top: prompts; bottom: detections).

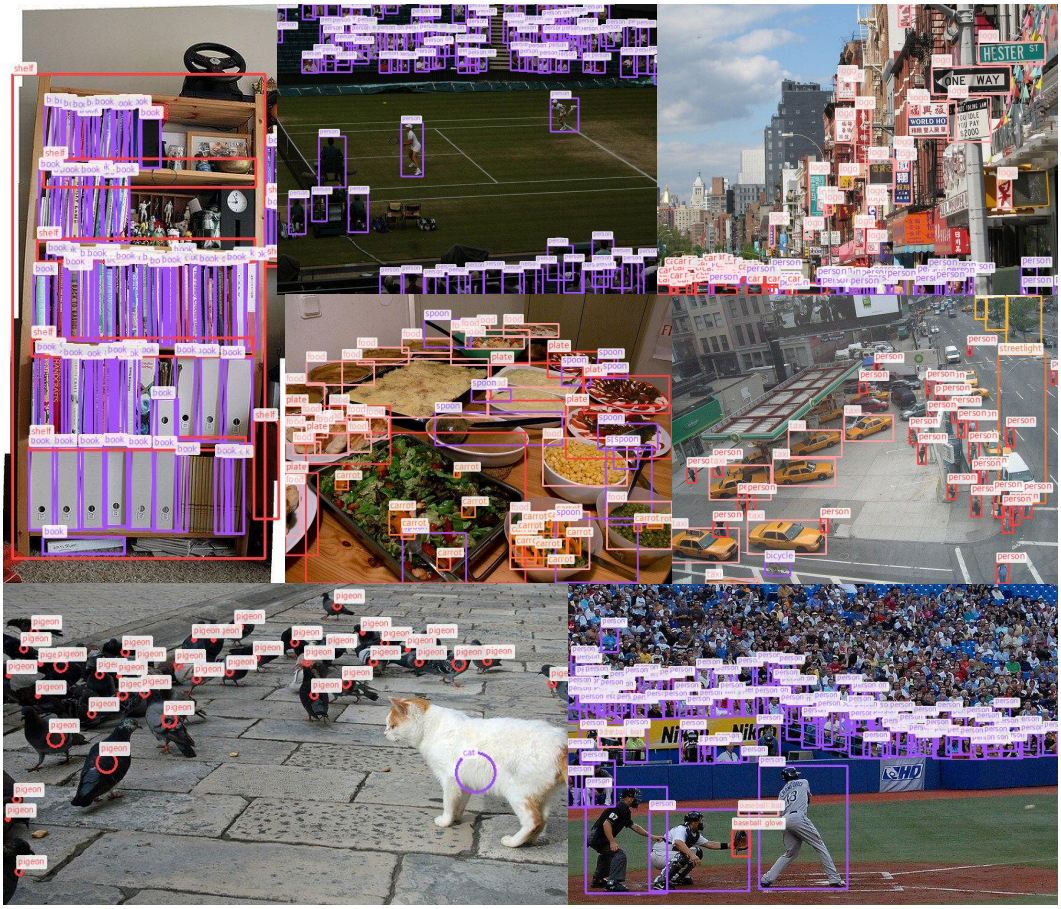

Figure 9: Zero-shot text-prompted detection examples.