Omni-SpikeDet: A Spiking Open-Vocabulary Detector with Dynamic Text-Image Alignment

Ziqi Li[1,2] **Tao Gao**[1,*] **Ting Chen**[1,*] **Xin Zhang**[2]
Yuanbo Wen[1] **Yipo Huang**[1] **Robby T. Tan**[2]

[1]Chang'an University
[2]National University of Singapore

lzq@chd.edu.cn, gtnwpu@126.com, tchen@chd.edu.cn
x.zhang@u.nus.edu, wyb@chd.edu.cn, yphuang@chd.edu.cn
robby.tan@nus.edu.sg

[*]Correspondence: gtnwpu@126.com, tchen@chd.edu.cn

Abstract

Open-vocabulary object detection requires recognizing both known and novel categories beyond a fixed label space. Existing open-vocabulary detectors have made strong progress by integrating visual features with language embeddings, but they are predominantly built on dense ANN-based architectures. In contrast, spiking neural networks (SNNs) have been studied for sparse visual computation, yet their use in open-vocabulary detection remains largely unexplored. This paper presents Omni-SpikeDet, a spiking open-vocabulary detector that investigates how text-guided visual recognition can be realized in a spiking framework. Omni-SpikeDet combines spike-based visual feature extraction with text-guided multi-scale alignment. It uses a spiking convolutional module with integer-spike training and spike-based inference to reduce the mismatch between continuous vision-language supervision and discrete spiking representations. It further introduces a Cross-Scale Text-Image Interaction (CTI) module that modulates multi-level spiking features with language embeddings for prompt-guided detection. Experiments on COCO and LVIS show that Omni-SpikeDet achieves competitive open-vocabulary detection performance with a compact model, obtaining 47.2 AP on COCO and 27.9 AP on LVIS with 4.2M parameters. Omni-SpikeDet consumes an estimated 19.8 mJ per image for detector-side convolutions with cached text embeddings.

1 Introduction

Object detection is a cornerstone of computer vision, supporting applications in robotics, autonomous systems, and visual understanding [1–4]. Despite substantial progress [5–8], conventional detectors are typically trained in a closed-vocabulary setting, where the prediction space is limited to predefined categories (*e.g.*, the 80 categories in COCO [9]). This limits their ability to recognize objects beyond the training label set. Open-vocabulary object detection addresses this limitation by introducing language supervision, allowing visual regions to be matched with semantic concepts outside fixed training categories. Recent methods leverage Vision-Language Models (VLMs) [10–21], such as BERT [22] and CLIP [23], to transfer semantic knowledge to detectors. Early approaches rely on limited region-text annotations, while recent region-level vision-language pre-training provides a more scalable solution [24–26], as demonstrated by YOLO-World [27] and OW-OVD [28].

Despite this progress, existing open-vocabulary detectors are predominantly built on dense ANN-based architectures. In contrast, Spiking Neural Networks (SNNs) provide sparse spike-driven representations and have been explored for efficient visual recognition and closed-set object detection

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

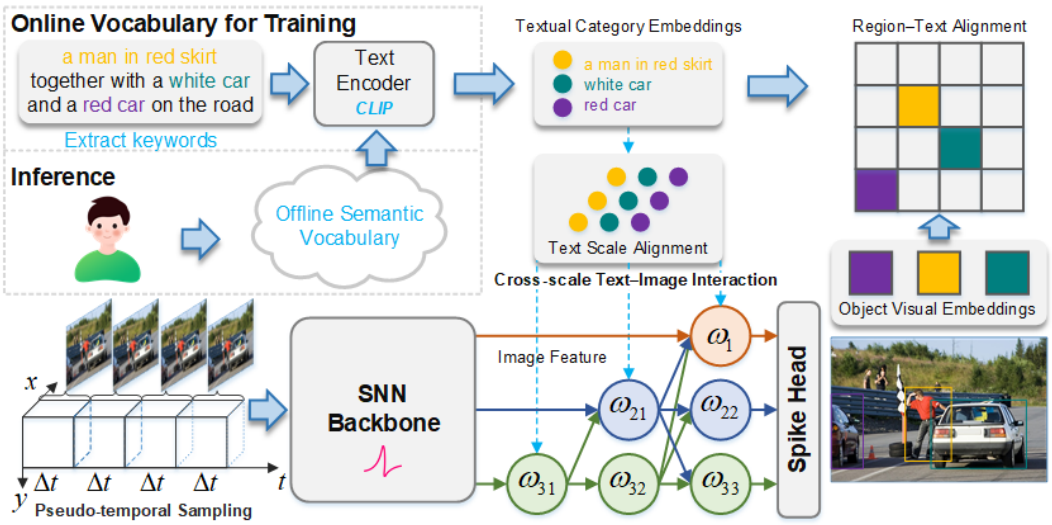


Figure 1: Overview of Omni-SpikeDet. During training, an online textual vocabulary is constructed from region-level annotations and used for text-guided visual-semantic alignment. During inference, pre-computed offline text embeddings provide the semantic vocabulary for open-vocabulary detection. Pseudo-temporal image inputs are processed by the spiking backbone, cross-scale text-image interaction modulates multi-level spiking features with language embeddings, and the spike-based detection head predicts object locations and categories.

[29–32]. However, their use in open-vocabulary detection remains largely unexplored. A key challenge is that open-vocabulary detection relies on continuous language embeddings and multi-scale visual-semantic alignment, whereas SNNs operate through discrete spiking representations. Directly combining these two paradigms can therefore introduce a mismatch between continuous vision-language supervision and discrete spiking features.

This motivates a spiking open-vocabulary detector that supports text-guided recognition while preserving spike-based visual representations. To this end, we present Omni-SpikeDet, which integrates spike-based visual encoding, text-guided multi-scale alignment, and spike-based visual inference within a unified framework.

To reduce the mismatch between continuous vision-language supervision and discrete spike-based features, we introduce a spiking convolutional module with Text-Aligned Spiking Activation (TASA). TASA employs integer-spike training to preserve richer intermediate responses during optimization, while retaining spike-based inference. This provides a practical bridge between continuous text-guided learning and discrete spiking representations. To support cross-scale visual-semantic alignment, we further introduce the Cross-Scale Text–Image Interaction (CTI) module. CTI modulates multi-level spiking features using CLIP text embeddings, allowing textual cues to guide feature fusion across scales. During inference, part of this interaction can be re-parameterized to reduce runtime overhead.

As shown in Figure. 1, Omni-SpikeDet integrates spike-based feature extraction, text-guided multi-scale alignment, and spike-compatible inference for prompt-guided detection of seen and unseen categories. The main contributions are summarized as follows:

- We propose Omni-SpikeDet, a spiking open-vocabulary detection framework that realizes text-guided object recognition with spike-based visual representations.
- We introduce Text-Aligned Spiking Activation (TASA), which adapts integer-spike training and binary-spike inference to open-vocabulary detection by reducing the mismatch between continuous language supervision and discrete spiking features.
- We devise a Cross-Scale Text-Image Interaction (CTI) module that modulates multi-level spiking features with language embeddings, enabling text-guided visual-semantic alignment across scales.
- Experiments on COCO and LVIS show competitive open-vocabulary detection with 4.2M parameters.

2 Related Work

Closed-Set Object Detection. Most detectors learn a fixed vocabulary defined by datasets such as COCO [9] and Objects365 [33]. They include region-based methods such as Faster R-CNN [8], dense one-stage detectors over anchors or grids [7, 34, 35], and query-based DETR-style detectors [36, 37]. Real-time YOLO variants [7, 38, 39] improve efficiency with path aggregation [40], cross-stage partial networks [41], and re-parameterization [42]. AFPN [43] progressively fuses multi-level features for dense prediction. Multiscale fusion also supports remote-sensing [44] and cluttered-traffic detection [45]. MobileNetV2 [46] and EfficientNet [47] reduce dense-backbone cost through lightweight operators and compound scaling, respectively. In contrast, Omni-SpikeDet targets open-vocabulary detection with spike-based visual representations.

Open-Vocabulary Object Detection. Open-vocabulary object detection recognizes categories beyond a fixed label space [14]. Early methods train on base classes and evaluate on novel ones [10–13, 48], while Detic expands the vocabulary using image-level supervision [49]. Recent methods use large-scale vision-language pretraining and region-text alignment [23, 50–56], with representative systems including OWL-ViTs [57, 58], GLIP [59], and Grounding DINO [60]. Many rely on heavy backbones or detectors such as Swin-L [61], ATSS [35], and DINO [62], while YOLO-World provides a real-time YOLO-based alternative. Omni-SpikeDet instead studies open-vocabulary detection in a spiking framework.

Spiking Neural Networks for Object Detection. SNNs provide sparse, event-driven computation for low-power vision. Existing training strategies include ANN-to-SNN conversion [63, 64] and direct training with surrogate gradients [65, 66], with tdBN [67] improving deep SNN training and supporting competitive classification and regression [68, 69]. For detection, conversion-based methods [70, 71], hybrid SNN-ANN designs [72, 73], and integer-valued SNNs with spike-driven inference [31, 74] have been explored. However, EMS-YOLO [30], SpikeYOLO [31], MSD [32], and SpikeDet [75] remain limited to predefined categories. Omni-SpikeDet extends directly trained SNNs to open-vocabulary detection with text-guided multi-scale alignment.

3 Proposed Method

We present Omni-SpikeDet, a spiking open-vocabulary detection framework that combines spike-based visual feature extraction with text-guided multi-scale alignment. Given an image and textual category prompts, Omni-SpikeDet extracts multi-scale spiking features and aligns them with language embeddings for prompt-guided detection. It introduces two core components: a Spiking Convolution Neuron (SCN) with Text-Aligned Spiking Activation (TASA) for spike-based visual representation learning, and a Cross-Scale Text–Image Interaction (CTI) module for cross-scale visual-semantic alignment. We first formulate the task and input representations, then describe the designs of SCN/TASA and CTI.

3.1 Preliminary Definition

The input to the SNN is represented as $X \in \mathbb{R}^{T\times C\times H\times W}$, where T denotes the number of timesteps, C the number of channels, and $H \times W$ the spatial resolution.

Static Images. For static images, we follow the common direct-input setting in SNNs [76, 77], where the same image is repeated across T timesteps. This provides a temporal dimension for spike-based processing, allowing the first spiking layer to convert continuous pixel values into spike trains and subsequent layers to accumulate responses over timesteps. Since the input frames are pseudo-temporal rather than event streams [78, 79] or videos [80], this setting does not model explicit motion dynamics.

Pre-training Formulation. Traditional object detectors are trained with instance-level annotations $\Omega = \{B_i, c_i\}_{i=1}^{N}$, where B_i is a bounding box and c_i is its category. Following open-vocabulary detection, we reformulate annotations as region–text pairs $\Omega = \{B_i, t_i\}_{i=1}^{N}$, where t_i is a textual description of B_i (*e.g.*, category name or noun phrase). Given an image I and a set of text prompts $\mathcal{P}$, the detector predicts bounding boxes $\{\hat{B}_k\}$ and corresponding visual–semantic embeddings $\{e_k\}$, where $e_k \in \mathbb{R}^D$.

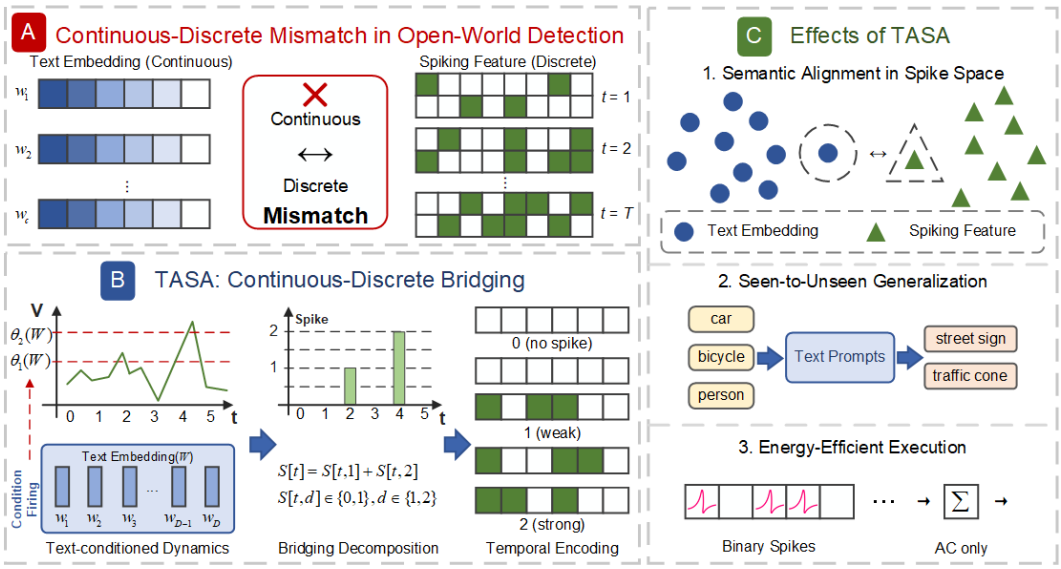


Figure 2: Illustration of Text-Aligned Spiking Activation (TASA) for bridging continuous language supervision and discrete spiking representations.

3.2 Omni-SpikeDet Architecture

The overall architecture is illustrated in Figure. 1. Omni-SpikeDet consists of four components: an SCN-based spiking backbone, a CLIP text encoder, a Cross-Scale Text–Image Interaction (CTI) neck, and a spike-based detection head. Given an input image and a set of text prompts $\mathcal{P}$, the spiking backbone extracts multi-scale visual features, while the text encoder produces language embeddings. The CTI neck modulates multi-level spiking features with text embeddings to support cross-scale visual–semantic alignment for open-vocabulary detection.

Text Encoder. Given a set of text prompts $\mathcal{P} = \{p_j\}_{j=1}^{C}$, we use the CLIP [23] text encoder to obtain language embeddings $W = \text{TextEncoder}(\mathcal{P}) \in \mathbb{R}^{C \times D}$, where C denotes the number of textual concepts and D is the embedding dimension. For caption-based supervision, noun phrases are extracted and encoded as text prompts.

Spike Head. Following [81], a decoupled head predicts bounding boxes $\{b_k\}_{k=1}^{K}$ and object embeddings $\{e_k\}_{k=1}^{K}$. The similarity between object and text embeddings is computed as:

$$s_{k,j} = \alpha \cdot \text{Norm}(e_k) \cdot \text{Norm}(w_j)^\top + \beta,$$

where α and β are learnable scalars for contrastive training.

Training and Inference. During training, each mosaic sample builds an online vocabulary from positive noun phrases and sampled negatives, capped at M tokens. At inference, the category vocabulary is encoded once and its embeddings are cached across images.

3.3 Spiking Convolution Neuron with Text-Aligned Spiking Activation (TASA)

To address the mismatch between continuous language supervision and discrete visual spikes in open-vocabulary detection, we introduce a Spiking Convolution Neuron (SCN) with Text-Aligned Spiking Activation (TASA), as illustrated in Figure. 2. Binary spikes are efficient for inference, but their limited precision can cause quantization loss during training, especially when visual features are optimized against continuous text embeddings. As shown in the figure, TASA uses integer-valued spike responses during training to preserve richer intermediate information for text-aligned optimization, while retaining binary spike-based inference for efficient computation. The resulting SCN consists of TASA encoding, spike-driven convolution, and temporal-domain batch normalization, providing a spike-compatible visual representation for subsequent text-guided alignment.

Problem Formulation. Given spiking visual features S and text embeddings $W \in \mathbb{R}^{C\times D}$, open-vocabulary detection jointly optimizes detection and visual-semantic alignment:

$$\mathcal{L} = \mathcal{L}_{det} + \lambda \mathcal{L}_{align}(S, W), \tag{1}$$

where $\mathcal{L}_{align}$ encourages semantic consistency between visual and textual representations. Since W is continuous while binary spikes are discrete, directly optimizing binary spiking features under language supervision can introduce quantization error and make cross-modal optimization more difficult.

TASA Encoding. The standard LIF neuron with soft reset is formulated as:

$$U[t] = H[t-1] + X[t], \tag{2}$$

$$S[t] = \Theta(U[t] - V_{\text{th}}), \tag{3}$$

$$H[t] = \beta(U[t] - S[t]V_{\text{th}}), \tag{4}$$

where $U[t]$ is the membrane potential, $H[t]$ is the hidden membrane state, $\Theta(\cdot)$ denotes the Heaviside function, V_{th} is the firing threshold, and β is the decay factor.

To reduce the mismatch between continuous optimization and binary spike responses, TASA adopts bounded integer-valued spike responses during training:

$$\tilde{S}[t] = L\left(\text{round}\left(\frac{U[t]}{V_{\text{th}}}\right), 0, D\right), \tag{5}$$

where $L(\cdot)$ clips the spike count to $[0, D]$, and D denotes the maximum spike count. Surrogate gradients are used within the bounded interval to enable differentiable optimization. During inference, the integer response can be decomposed into binary spikes, allowing the model to retain spike-based inference while benefiting from higher-precision training.

SCN Formulation. Given an input feature sequence $\mathbf{X} \in \mathbb{R}^{T\times C_{\text{in}}\times H\times W}$, SCN first applies TASA encoding independently at each timestep:

$$\tilde{\mathbf{X}}_t = \text{TASA}(\mathbf{X}_t). \tag{6}$$

Here, $\tilde{\mathbf{X}}_t$ denotes the TASA-encoded spike response at timestep t. The encoded responses are then processed by convolution over spike activations:

$$\mathbf{Y}_t^c = \sum_{k=1}^{C_{\text{in}}} \mathbf{W}^{c,k} * \tilde{\mathbf{X}}_t^k, \tag{7}$$

where $\mathbf{W}^{c,k}$ denotes the convolution kernel from input channel k to output channel c. Finally, temporal-domain batch normalization is applied to stabilize feature distributions across timesteps:

$$\mathbf{Y} = \text{tdBN}\left(\{\mathbf{Y}_t\}_{t=1}^T\right). \tag{8}$$

This SCN block provides a spike-compatible convolutional representation that can be integrated into the multi-scale backbone used for text-guided detection.

Algorithm 1: Spiking Convolutional Neuron (SCN)

Require: Input $\mathbf{X} \in \mathbb{R}^{T\times C_{\text{in}}\times H\times W}$
Ensure: Output $\mathbf{Y} \in \mathbb{R}^{T\times C_{\text{out}}\times H'\times W'}$

Integer Leaky Integrate-and-Fire Encoding
for $t = 1$ to T **do**
  $\tilde{\mathbf{X}}_t \leftarrow \text{TASA}(\mathbf{X}_t)$
end for
Spike-driven Convolution
for $t = 1$ to T **do**
  for $c = 1$ to C_{out} **do**
    $\mathbf{Y}_t^c \leftarrow \mathbf{0}$
    for $k = 1$ to C_{in} **do**
      $\mathbf{Y}_t^c \mathrel{+}= \mathbf{W}^{c,k} * \tilde{\mathbf{X}}_t^k$
    end for
  end for
end for
Temporal-domain Batch Normalization
$\mathbf{Y} \leftarrow \text{tdBN}(\{\mathbf{Y}_t\}_{t=1}^T)$

Train-Inference Consistency. TASA uses bounded integer-valued spike responses during training to preserve richer intermediate information under continuous language supervision. During inference, each integer response is decomposed into a sum of binary spikes:

$$\tilde{S}[t] = \sum_{d=1}^{D} S[t,d], \quad S[t,d] \in \{0,1\}. \tag{9}$$

Accordingly, the next-layer input is computed as:

$$X^{l+1}[t] = \sum_{d=1}^{D} W^l S^l[t,d], \tag{10}$$

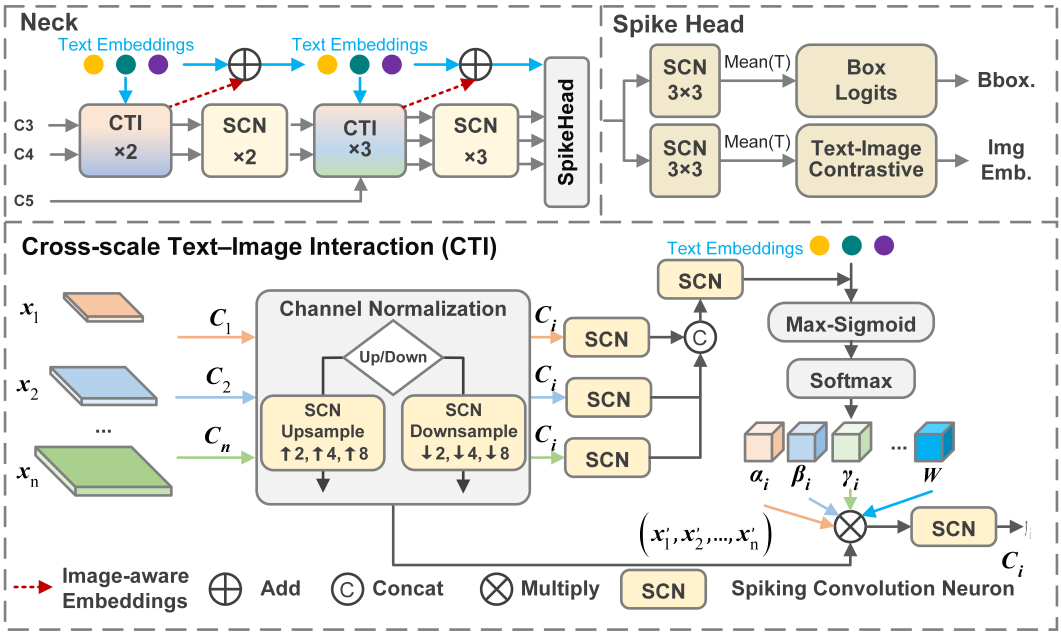


Figure 3: Illustration of Cross-Scale Text–Image Interaction Architecture.

which can be implemented through sparse accumulation over binary spikes rather than dense multiplication. Here D bounds the training-time response, while deployed convolution receives binary spikes over T steps.

Prior integer-valued spiking neurons mainly target closed-set recognition or detection. TASA adapts the same general integer-spike training and binary-spike inference principle to open-vocabulary detection, where spiking visual features are optimized under continuous language supervision. Incorporated into SCN, TASA reduces the quantization mismatch between text embeddings and spiking features, producing sparse visual representations for subsequent text-guided alignment in CTI.

3.4 Cross-Scale Text-Image Interaction (CTI)

To support visual–semantic alignment across feature levels, we introduce the Cross-Scale Text–Image Interaction (CTI) module, as shown in Figure. 3. CTI modulates multi-scale spiking features with language embeddings, allowing textual cues to influence feature fusion at different resolutions. This is particularly important for open-vocabulary detection, where category information is provided by text embeddings rather than a fixed classifier. Unlike dense cross-modal attention, CTI uses lightweight text-conditioned modulation and cross-scale spatial weighting.

Formally, given text embeddings $W = \{w_j\}_{j=1}^{C}$, $w_j \in \mathbb{R}^D$, and multi-scale visual features $X_l \in \mathbb{R}^{H_l \times W_l \times D}$, $l \in \{3, 4, 5\}$, CTI first injects textual cues into each scale using max-sigmoid modulation:

$$X_l' = X_l \odot \delta\left(\max_{j \in \{1,\dots,C\}} X_l w_j^\top\right), \tag{11}$$

where $\delta(\cdot)$ is the sigmoid function and $\odot$ denotes element-wise modulation with broadcasting over channels. The modulated features X_l' are then used for cross-scale fusion.

To adaptively combine information across scales, CTI employs a spatial weighting mechanism. Let $x_{ij}^{n \to l}$ denote the feature at position (i, j) projected from scale n to scale l, and let w_l^* denote the text-conditioned feature used at scale l. The fused feature y_{ij}^l is computed as

$$y_{ij}^l = \alpha_{ij}^l \cdot (x_{ij}^{1 \to l} + w_l^*) + \beta_{ij}^l \cdot (x_{ij}^{2 \to l} + w_l^*) + \gamma_{ij}^l \cdot (x_{ij}^{3 \to l} + w_l^*), \tag{12}$$

where $\alpha_{ij}^l + \beta_{ij}^l + \gamma_{ij}^l = 1$. The coefficients $\{\alpha^l, \beta^l, \gamma^l\}$ are produced by compressing the projected features, concatenating them across scales, reducing them to three channels with a lightweight convolution, and applying Softmax normalization.

CTI combines convolutional projections of spiking features with dense text-conditioned gates and similarity scores; vocabulary embeddings are cached at inference.

4 Experiments

We evaluate Omni-SpikeDet on LVIS and COCO under zero-shot and fine-tuning settings. Text embeddings are cached at inference. GPU latency is measured on NVIDIA A800 hardware, and detector-side energy is estimated from convolutional operations as described in Appendix A.

Table 1: Pre-training datasets used for Omni-SpikeDet. The table summarizes the dataset type, vocabulary size, number of images, and annotations employed during large-scale pre-training.

Dataset	Type	Vocab.	Images	Anno.
Objects365V1 [33]	Detection	365	609k	9,621k
GQA [82]	Grounding	-	621k	3,681k
Flickr30k [83]	Grounding	-	149k	641k
CC3M-Lite [84]	Image-Text	-	246k	821k

Pre-training Data. Omni-SpikeDet is pre-trained on Objects365-v1 [33], GQA [82], Flickr30k [83], and CC3M-Lite [84], as summarized in Table 1. COCO-overlapping GoldG images are removed. We denote Objects365 by O, GQA+Flickr30k by G, and CC3M-Lite by C.

Zero-shot Evaluation. We evaluate zero-shot detection on LVIS [85], which contains 1,203 categories. Following prior work, we use the LVIS minival split, report Fixed AP, and set the maximum number of predictions per image to 1000.

Main Results on LVIS Object Detection. We compare with recent open-vocabulary detectors, especially lightweight or real-time-oriented baselines.

4.1 Ablation Studies

Table 2 evaluates SCN and CTI on LVIS minival, using O-only pre-training for the component rows.

Table 2: LVIS zero-shot ablation.

SCN	CTI	O+G+C	AP	AP_r	AP_c	AP_f
×	×	×	16.6	10.0	13.9	20.3
✓	×	×	17.8	11.1	14.9	21.4
×	✓	×	17.4	10.8	14.6	21.0
✓	✓	×	18.5	11.9	15.8	22.1
✓	✓	✓	23.6	15.5	20.6	27.6

Pre-training Data. With SCN and CTI fixed, additional grounding and caption data improve overall AP from 18.5 to 23.6 and rare-class AP from 11.9 to 15.5.

Spiking Convolutional Neurons (SCN). Table 2 reports the effect of using SCN in the visual backbone. Compared with the baseline variant, SCN improves overall AP from 16.6 to 17.8 (+1.2) and rare-category AP_r from 10.0 to 11.1 (+1.1). These results indicate that SCN provides a modest but consistent improvement in zero-shot LVIS detection, while enabling a spike-based visual representation for the proposed framework.

Cross-scale Text-Image Interaction (CTI). Table 2 evaluates the contribution of the CTI module. Adding CTI to SCN improves AP from 17.8 to 18.5 (+0.7) and AP_r from 11.1 to 11.9 (+0.8). These results indicate that adaptive cross-scale text–image interaction improves open-vocabulary recognition, particularly for challenging or underrepresented categories.

Table 3: COCO component ablation. SF denotes standard text fusion, and Energy denotes the estimated detector-side energy (mJ/image).

SCN	TASA	SF	CTI	AP	AP_{50}	AP_{75}	Energy
×	×	×	×	43.2	60.0	46.8	18.6
✓	×	×	×	44.3	61.2	48.0	18.9
✓	✓	×	×	45.1	62.0	49.0	19.1
✓	✓	✓	×	46.3	63.1	50.4	19.5
✓	✓	×	✓	47.2±0.2	64.0±0.1	51.6±0.2	19.8

4.2 Quantitative Analysis

We further evaluate Omni-SpikeDet under zero-shot and fine-tuning settings on COCO and LVIS to assess both open-vocabulary transfer and downstream detection performance.

Experimental Setup. Omni-SpikeDet is initialized with pre-trained weights and fine-tuned for 80 epochs using AdamW with an initial learning rate of 0.0002. For LVIS, we fine-tune on base categories and evaluate on novel rare categories. When fine-tuned, the CLIP text encoder uses a learning rate scaled by 0.01 relative to the base learning rate. G denotes the grounding data used for region-level vision-language pre-training, including GQA and Flickr30k.

Table 4: Zero-shot evaluation on LVIS minival [85]; Fixed AP is reported. O denotes Objects365, G denotes GQA+Flickr30k and C denotes CC3M-Lite, and Cap denotes Cap4M.

Method	Backbone	Params	Pre-trained Data	AP	AP_r	AP_c	AP_f
MDETR [86]	R-101	169M	G	24.2	20.9	24.3	24.2
GLIP-T [59]	Swin-T	232M	O+G	24.9	17.7	19.5	31.0
GLIP-T [59]	Swin-T	232M	O+G+Cap	26.0	20.8	21.4	31.0
GLIPv2-T [53]	Swin-T	232M	O+G	26.9	-	-	-
GLIPv2-T [53]	Swin-T	232M	O+G+Cap	29.0	-	-	-
BARON [13]	-	-	-	18.0	24.4	28.9	25.1
GroundingDINO-T [60]	Swin-T	172M	O+G	25.6	14.4	19.6	32.2
GroundingDINO-T [60]	Swin-T	172M	O+G+Cap	27.4	18.1	23.3	32.7
YOLO-World-S [27]	S	13M	O+G	26.2	19.1	23.6	29.8
RALF [87]	-	-	-	21.9	26.2	29.1	26.6
ProxyDet [88]	R-50	-	O+G	26.2	33.1	35.6	32.5
SIC-CADS [89]	V8s	13M	O+G	26.5	33.0	35.6	32.5
YOLOE-S [90]	V8s	13M	O+G	27.9	22.3	27.8	29.0
YOLOE-26-N [91]	N	3.9M	O+G	24.7	20.5	24.1	26.1
Omni-SpikeDet	S	4.2M	O+G	27.7	21.9	27.0	29.4
Omni-SpikeDet	S	4.2M	O+G+C	27.9	22.2	26.8	30.4

Table 5: Comparison of results with related closed-set spiking neural network detectors on COCO.

Model	Param (M)	Energy (mJ)	$T \times D$	mAP@50 (%)	mAP@50:95 (%)
Spiking-YOLO [70]	10.2	–	3500×1	–	25.7
EMS-YOLO [30]	33.9	29.0	4×1	50.1	30.1
Meta-SpikeFormer [29]	16.8	70.7	4×1	50.3	–
SpikeYOLO-N [31]	13.2	23.1	1×4	59.2	42.5
SpikeYOLO-S [31]	23.1	34.6	1×4	62.3	45.5
SpikeYOLO-S [31]	23.1	134.7	4×4	63.3	46.3
MSD [32]	7.8	6.4	–	62.0	45.3
SpikePack [92]	4.8	–	6×1	55.7	39.0
SpikeDet-S [75]	22.0	19.6	1×4	62.6	46.2
Omni-SpikeDet	4.2	19.8	4×2	64.0	47.2

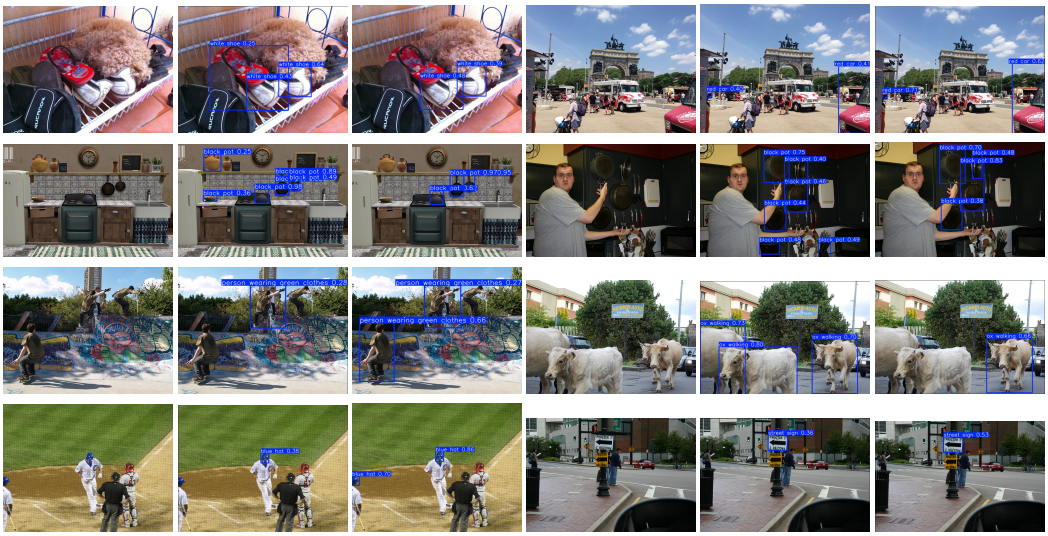
COCO Object Detection.

Table 6 reports COCO results under zero-shot and fine-tuning settings. Without COCO fine-tuning, Omni-SpikeDet obtains 38.5 AP with O+G and 38.9 with O+G+C; after 80 epochs, these become 46.8 and 47.2 AP. Under shared O+G+C pre-training and 80-epoch fine-tuning (Table 3), TASA adds 0.8 AP over LIF-based SCN, CTI adds 0.9 AP over standard text fusion, and the full model gains 4.0 AP over the binary SNN baseline, reaching 47.2 ± 0.2 AP across three seeds.

Table 6: COCO val2017 detection comparison.

Method	Pre.	AP	AP_{50}	AP_{75}	Params (M)
Scratch					
YOLOv6-S [93]	-	43.7	60.8	47.0	18.5
YOLOv7-T [94]	-	37.5	55.8	40.2	36.9
YOLOv8-S [81]	-	44.4	61.2	48.1	11.2
Zero-shot					
YOLO-World-S [27]	O+G	37.6	52.3	40.7	13.0
BARON [13]	-	36.2	55.7	39.1	-
YOLOE-S [90]	O+G	35.6	51.5	38.9	13.0
Omni-SpikeDet	O+G	38.5	53.2	41.5	4.2
Omni-SpikeDet	O+G+C	38.9	54.0	42.2	4.2
Fine-tuned					
YOLO-World-S [27]	O+G	45.9	62.3	50.1	13.0
SIC-CADS [89]	-	46.8	-	-	-
ProxyDet [88]	-	46.8	-	-	-
YOLOE-S [90]	-	45.0	61.6	49.1	13.0
Omni-SpikeDet	O+G	46.8	63.2	50.9	4.2
Omni-SpikeDet	O+G+C	47.2	64.0	51.6	4.2

LVIS Object Detection. Table 4 reports LVIS minival zero-shot results. With O+G pre-training, Omni-SpikeDet reaches 27.7 AP and 21.9 rare-class AP, versus 26.2 and 19.1 for YOLO-World-S. Adding CC3M-Lite to O+G improves COCO AP by 0.4 in both zero-shot and fine-tuned settings (Table 6). On LVIS, AP rises by 0.2 overall, with changes of $+0.3$, -0.2, and $+1.0$ for rare, common, and frequent categories, respectively.



Input Baseline Omni-SpikeDet Input Baseline Omni-SpikeDet

Figure 4: Text-prompted detection results on COCO 2017. Each row shows the input image, baseline results, and Omni-SpikeDet results under natural language prompts.

4.3 Qualitative Analysis

As illustrated in Figure. 4, Omni-SpikeDet supports prompt-guided detection with natural language queries. Compared with the baseline, it produces more complete detections in several challenging examples involving occlusion, overlapping objects, and small objects. These qualitative results illustrate that text-guided modulation can help align spiking visual features with textual prompts in open-vocabulary detection.

Failure Cases. Figure. 5 shows representative failure cases. The model can still confuse fine-grained attributes, such as detecting a black horse under the prompt "white horse," and may over-focus on salient prompt terms, such as detecting glasses instead of the whole person. These failures suggest that fine-grained attribute grounding and holistic object-level reasoning remain challenging for the current framework. The cases illustrate distinct errors in text–region alignment: one matches a localized object despite an incompatible color attribute, whereas the other selects a constituent part under a whole-object prompt.



Figure 5: Failure cases of the open-vocabulary detection model, including color confusion and partial detections under text-guided prompts.

5 Limitations and Conclusion

Limitations. The method processes repeated static frames rather than event streams, retains dense text encoding and CTI gating, and remains weak on small, occluded, and attribute-sensitive objects. The 19.8 mJ estimate excludes text encoding, non-convolutional operations, and memory access; no neuromorphic-device energy is measured. Future work should evaluate end-to-end energy and event-based inputs.

Conclusion. Omni-SpikeDet combines spike-based visual inference with text-conditioned multi-scale fusion for open-vocabulary detection. LVIS and COCO results support the contributions of TASA and CTI. The operation-level estimate characterizes detector-side convolutional cost; end-to-end hardware efficiency remains to be established.

Acknowledgments

This work was supported by the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (JYB2025XDXM104); the National Natural Science Foundation of China (52572353, 52441205, and 62501080); the Shaanxi Provincial Natural Science Basic Research Program (2026JC-JCQN-105); the Shaanxi International Science and Technology Cooperation Programme (2024GH-YBXM-24); the Innovation Capability Support Plan of Shaanxi Province (2025RS-CXTD-012); the Fundamental Research Funds for the Central Universities (300102245715); and the Fundamental Research Funds for the Central Universities, Chang'an University (300102406108).

References

[1] Wen, Y., T. Gao, S. Liang, et al. When optimal transport meets photo-realistic image dehazing with unpaired training. *IEEE Transactions on Neural Networks and Learning Systems*, 37(9):4755–4769, 2026.

[2] Zhang, X., Y.-C. Chen. Adaptive domain generalization via online disagreement minimization. *IEEE Transactions on Image Processing*, 32:4247–4258, 2023.

[3] Zhang, X., J. Xie, Y. Yuan, et al. HEAP: Unsupervised object discovery and localization with contrastive grouping. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pages 7323–7331. 2024.

[4] Yang, X., X. Zhang, X. Wang. ERF: A benchmark dataset for robust semantic segmentation under extreme rainfall conditions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, pages 9301–9309. 2025.

[5] He, K., G. Gkioxari, P. Dollár, et al. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969. 2017.

[6] Lin, T.-Y., P. Dollár, R. Girshick, et al. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125. 2017.

[7] Redmon, J., S. Divvala, R. Girshick, et al. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788. 2016.

[8] Ren, S., K. He, R. Girshick, et al. Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28, 2015.

[9] Lin, T.-Y., M. Maire, S. Belongie, et al. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[10] Du, Y., F. Wei, Z. Zhang, et al. Learning to prompt for open-vocabulary object detection with vision-language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14084–14093. 2022.

[11] Gu, X., T.-Y. Lin, W. Kuo, et al. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021.

[12] Shi, C., S. Yang. EdaDet: Open-vocabulary object detection using early dense alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15724–15734. 2023.

[13] Wu, S., W. Zhang, S. Jin, et al. Aligning bag of regions for open-vocabulary object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15254–15264. 2023.

[14] Zareian, A., K. D. Rosa, D. H. Hu, et al. Open-vocabulary object detection using captions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14393–14402. 2021.

[15] Li, Q., X. Wang. Sponge tool attack: Stealthy denial-of-efficiency against tool-augmented agentic reasoning. *arXiv preprint arXiv:2601.17566*, 2026.

[16] Li, Q., R. Yu, X. Wang. Vid-SME: Membership inference attacks against large video understanding models. *Advances in Neural Information Processing Systems*, 38:111572–111596, 2025.

[17] Li, Q., X. Yang, X. Wang. BadWAM: When world-action models dream right but act wrong. *arXiv preprint arXiv:2607.15207*, 2026.

[18] Li, Q., B. Yin, W. Huang, et al. Vision-language-action safety: Threats, challenges, evaluations, and mechanisms. *arXiv preprint arXiv:2604.23775*, 2026.

[19] Li, Q., X. Wang. ViMU: Benchmarking video metaphorical understanding. *arXiv preprint arXiv:2605.14607*, 2026.

[20] Zhang, X., H. Wang, Y. Zhou, et al. Actor as its own critic: Unifying region understanding and localization via CycleGRPO. In *European Conference on Computer Vision*, pages 586–604. Springer, 2026.

[21] Zhang, X., R. T. Tan. Mamba as a bridge: Where vision foundation models meet vision language models for domain-generalized semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14527–14537. 2025.

[22] Devlin, J., M.-W. Chang, K. Lee, et al. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. 2019.

[23] Radford, A., J. W. Kim, C. Hallacy, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[24] Tan, L., J. Xia, W. Liu, et al. Occluded person re-identification via saliency-guided patch transfer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pages 5070–5078. 2024.

[25] Tan, L., Y. Zhang, K. Han, et al. RLE: A unified perspective of data augmentation for cross-spectral re-identification. *Advances in Neural Information Processing Systems*, 37:126977–126996, 2024.

[26] Feng, Y., J. Li, J. Hu, et al. MDReID: Modality-decoupled learning for any-to-any multi-modal object re-identification. *Advances in Neural Information Processing Systems*, 38:42584–42605, 2025.

[27] Cheng, T., L. Song, Y. Ge, et al. YOLO-World: Real-time open-vocabulary object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16901–16911. 2024.

[28] Xi, X., Y. Huang, R. Luo, et al. OW-OVD: Unified open world and open vocabulary object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 25454–25464. 2025.

[29] Yao, M., J. Hu, T. Hu, et al. Spike-driven transformer v2: Meta spiking neural network architecture inspiring the design of next-generation neuromorphic chips. In *International Conference on Learning Representations*. 2024.

[30] Su, Q., Y. Chou, Y. Hu, et al. Deep directly-trained spiking neural networks for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6555–6565. 2023.

[31] Luo, X., M. Yao, Y. Chou, et al. Integer-valued training and spike-driven inference spiking neural network for high-performance and energy-efficient object detection. In *Computer Vision – ECCV 2024*, vol. 15090 of *Lecture Notes in Computer Science*, pages 253–272. Springer Nature Switzerland, 2025.

[32] Li, Z., T. Gao, Y. An, et al. Brain-inspired spiking neural networks for energy-efficient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3552–3562. 2025.

[33] Shao, S., Z. Li, T. Zhang, et al. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8430–8439. 2019.

[34] Tian, Z., C. Shen, H. Chen, et al. FCOS: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9627–9636. 2019.

[35] Zhang, S., C. Chi, Y. Yao, et al. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9759–9768. 2020.

[36] Carion, N., F. Massa, G. Synnaeve, et al. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.

[37] Zhu, X., W. Su, L. Lu, et al. Deformable DETR: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

[38] Redmon, J., A. Farhadi. YOLO9000: Better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7263–7271. 2017.

[39] Redmon, J., A. Farhadi. YOLOv3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[40] Liu, S., L. Qi, H. Qin, et al. Path aggregation network for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8759–8768. 2018.

[41] Wang, C.-Y., H.-Y. M. Liao, Y.-H. Wu, et al. CSPNet: A new backbone that can enhance learning capability of CNN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 390–391. 2020.

[42] Ding, X., X. Zhang, N. Ma, et al. RepVGG: Making VGG-style convnets great again. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13733–13742. 2021.

[43] Yang, G., J. Lei, H. Tian, et al. Asymptotic feature pyramid network for labeling pixels and regions. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(9):7820–7829, 2024.

[44] Gao, T., Z. Li, Y. Wen, et al. Attention-free global multiscale fusion network for remote sensing object detection. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–14, 2024.

[45] Li, Z., T. Gao, S. Li, et al. Frequency-oriented adaptive real-time object detector for cluttered traffic scenes. *Nature Communications*, 17:9787, 2026.

[46] Sandler, M., A. Howard, M. Zhu, et al. MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520. 2018.

[47] Tan, M., Q. V. Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, pages 6105–6114. 2019.

[48] Zang, Y., W. Li, K. Zhou, et al. Open-vocabulary DETR with conditional matching. In *European Conference on Computer Vision*, pages 106–122. Springer, 2022.

[49] Zhou, X., R. Girdhar, A. Joulin, et al. Detecting twenty-thousand classes using image-level supervision. In *European Conference on Computer Vision*, pages 350–368. Springer, 2022.

[50] Jia, C., Y. Yang, Y. Xia, et al. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021.

[51] Yao, L., J. Han, Y. Wen, et al. DetCLIP: Dictionary-enriched visual-concept paralleled pre-training for open-world detection. *Advances in Neural Information Processing Systems*, 35:9125–9138, 2022.

[52] Yao, L., J. Han, X. Liang, et al. DetCLIPv2: Scalable open-vocabulary object detection pre-training via word-region alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23497–23506. 2023.

[53] Zhang, H., P. Zhang, X. Hu, et al. GLIPv2: Unifying localization and vision-language understanding. *Advances in Neural Information Processing Systems*, 35:36067–36080, 2022.

[54] Zhong, Y., J. Yang, P. Zhang, et al. RegionCLIP: Region-based language-image pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16793–16803. 2022.

[55] Lin, C., P. Sun, Y. Jiang, et al. Learning object-language alignments for open-vocabulary object detection. *arXiv preprint arXiv:2211.14843*, 2022.

[56] Tan, L., W. Li, P. Dai, et al. Knowing where to focus: Attention-guided alignment for text-based person search. *International Journal of Computer Vision*, 134:72, 2026.

[57] Minderer, M., A. Gritsenko, A. Stone, et al. Simple open-vocabulary object detection. In *European Conference on Computer Vision*, pages 728–755. Springer, 2022.

[58] Minderer, M., A. Gritsenko, N. Houlsby. Scaling open-vocabulary object detection. *Advances in Neural Information Processing Systems*, 36:72983–73007, 2023.

[59] Li, L. H., P. Zhang, H. Zhang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975. 2022.

[60] Liu, S., Z. Zeng, T. Ren, et al. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pages 38–55. Springer, 2024.

[61] Liu, Z., Y. Lin, Y. Cao, et al. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022. 2021.

[62] Zhang, H., F. Li, S. Liu, et al. DINO: DETR with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022.

[63] Cao, Y., Y. Chen, D. Khosla. Spiking deep convolutional neural networks for energy-efficient object recognition. *International Journal of Computer Vision*, 113:54–66, 2015.

[64] Diehl, P. U., D. Neil, J. Binas, et al. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2015.

[65] Neftci, E. O., H. Mostafa, F. Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019.

[66] Kaiser, J., H. Mostafa, E. Neftci. Synaptic plasticity dynamics for deep continuous local learning (DECOLLE). *Frontiers in Neuroscience*, 14:424, 2020.

[67] Zheng, H., Y. Wu, L. Deng, et al. Going deeper with directly-trained larger spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pages 11062–11070. 2021.

[68] Hu, Y., L. Deng, Y. Wu, et al. Advancing spiking neural networks toward deep residual learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36(2):2353–2367, 2025.

[69] Fang, W., Z. Yu, Y. Chen, et al. Deep residual learning in spiking neural networks. *Advances in Neural Information Processing Systems*, 34:21056–21069, 2021.

[70] Kim, S., S. Park, B. Na, et al. Spiking-YOLO: Spiking neural network for energy-efficient object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pages 11270–11277. 2020.

[71] Li, Y., X. He, Y. Dong, et al. Spike calibration: Fast and accurate conversion of spiking neural network for object detection and segmentation. *arXiv preprint arXiv:2207.02702*, 2022.

[72] Lien, H.-H., T.-S. Chang. Sparse compressed spiking neural network accelerator for object detection. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 69(5):2060–2069, 2022.

[73] Johansson, O. *Training of object detection spiking neural networks for event-based vision*. Master's thesis, Luleå University of Technology, 2021.

[74] Liu, B.-C., Q. Yu, J.-W. Gao, et al. Spiking neuron networks based energy-efficient object detection for mobile robot. In *2021 China Automation Congress (CAC)*, pages 3224–3229. IEEE, 2021.

[75] Fan, Y., C. Liu, M. Li, et al. SpikeDet: Better firing patterns for accurate and energy-efficient object detection with spiking neural networks. *arXiv preprint arXiv:2501.15151*, 2025.

[76] Kim, Y., H. Park, A. Moitra, et al. Rate coding or direct coding: Which one is better for accurate, robust, and energy-efficient spiking neural networks? In *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 71–75. IEEE, 2022.

[77] Wu, Y., L. Deng, G. Li, et al. Direct training for spiking neural networks: Faster, larger, better. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pages 1311–1318. 2019.

[78] Xiao, Z., X. Wang. Event-guided online video super-resolution. *IEEE Transactions on Image Processing*, 35:8678–8690, 2026.

[79] Zhao, R., R. Xiong, D. Wang, et al. Spike camera optical flow estimation based on continuous spike streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(4):4756–4770, 2026.

[80] Li, Z., J. Liao, C. Tang, et al. USTC-TD: A test dataset and benchmark for image and video coding in 2020s. *IEEE Transactions on Multimedia*, 28:269–284, 2026.

[81] Sapkota, R., M. Karkee. Ultralytics YOLO evolution: An overview of YOLO26, YOLO11, YOLOv8 and YOLOv5 object detectors for computer vision and pattern recognition. *arXiv preprint arXiv:2510.09653*, 2025.

[82] Hudson, D. A., C. D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6700–6709. 2019.

[83] Plummer, B. A., L. Wang, C. M. Cervantes, et al. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2641–2649. 2015.

[84] Sharma, P., N. Ding, S. Goodman, et al. 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)*, pages 2556–2565. 2018.

[85] Gupta, A., P. Dollar, R. Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5356–5364. 2019.

[86] Kamath, A., M. Singh, Y. LeCun, et al. MDETR: Modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1780–1790. 2021.

[87] Kim, J., E. Cho, S. Kim, et al. Retrieval-augmented open-vocabulary object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17427–17436. 2024.

[88] Jeong, J., G. Park, J. Yoo, et al. ProxyDet: Synthesizing proxy novel classes via classwise mixup for open-vocabulary object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pages 2462–2470. 2024.

[89] Fang, R., G. Pang, X. Bai. Simple image-level classification improves open-vocabulary object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pages 1716–1725. 2024.

[90] Wang, A., L. Liu, H. Chen, et al. YOLOE: Real-time seeing anything. *arXiv preprint arXiv:2503.07465*, 2025.

[91] Jocher, G., J. Qiu, M. Liu, et al. Ultralytics YOLO26: Unified real-time end-to-end vision models. *arXiv preprint arXiv:2606.03748*, 2026.

[92] Shen, G., J. Li, T. Li, et al. SpikePack: Enhanced information flow in spiking neural networks with high hardware compatibility. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23385–23395. 2025.

[93] Li, C., L. Li, H. Jiang, et al. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv preprint arXiv:2209.02976*, 2022.

[94] Wang, C.-Y., A. Bochkovskiy, H.-Y. M. Liao. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7464–7475. 2023.

[95] Horowitz, M. Computing's energy problem (and what we can do about it). In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, pages 10–14. 2014.

[96] Chen, Z., L. Deng, B. Wang, et al. A comprehensive and modularized statistical framework for gradient norm equality in deep neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):13–31, 2022.

Appendix

A Detector-Side Operation Accounting

Operation-level energy estimation. Detector-side convolutional energy is estimated from per-image multiply–accumulate (MAC) and spike-triggered accumulation (AC) counts across the backbone, CTI convolutional pathway, and inference head. Using 45-nm, 32-bit reference energies of 4.6 pJ/MAC and 0.9 pJ/AC [95], we obtain

$$E_{\mathrm{det}} = \frac{4.6N_{\mathrm{MAC}} + 0.9N_{\mathrm{AC}}}{10^9} \quad \text{mJ/image.} \tag{13}$$

For inference convolution l, the dense operation count is $M_l = H_l W_l C_{\mathrm{out},l}(C_{\mathrm{in},l}/g_l)K_{h,l}K_{w,l}$ per image and timestep. For integer-valued spike input $x_l \in [0, D]$, the effective AC count is $N_{\mathrm{AC},l} = M_l\mu_l$, where $\mu_l = \sum x_l / \operatorname{numel}(x_l)$ is the mean spike multiplicity; dense-input convolutions instead contribute $N_{\mathrm{MAC},l} = M_l$. Counts are summed over deployed timesteps and averaged per image. The one-to-one inference head is included; the training-only branch is excluded. For ANN comparators, MACs are estimated as $N_{\mathrm{MAC}} = \mathrm{FLOPs}/2$. The estimate excludes text encoding, non-convolutional CTI and similarity operations, memory access, and execution overhead; it does not represent measured device energy.

Table 7: LVIS zero-shot accuracy and estimated detector-side convolutional energy per image. Text embeddings are cached.

Method	Params	Energy (mJ)	LVIS AP	Pre-training
YOLO-World-S [27]	13.0M	117.3	26.2	O+G
YOLOE-26-N [91]	3.9M	14.0	24.7	O+G
YOLOE-26-S [91]	10.7M	50.4	30.8	O+G
Omni-SpikeDet	4.2M	19.8	27.7	O+G
Omni-SpikeDet	4.2M	19.8	27.9	O+G+C

Omni-SpikeDet's estimated energy is 19.8 mJ/image, compared with 14.0 mJ/image for YOLOE-26-N and 50.4 mJ/image for YOLOE-26-S. Measured A800 latency is 9.1 ms for Omni-SpikeDet and 8.4 ms for YOLO-World-S (Table 8).

B Interpretability Analysis of Gradient Stability in Omni-SpikeDet.

This section analyzes the theoretical feasibility of deep training in Omni-SpikeDet and explains why it achieves stable gradient propagation. Using Gradient Norm Equality (GNE) theory [96], we analyze how SCN in Omni-SpikeDet may help mitigate gradient vanishing and explosion.

B.1 Lemma 1: Block Dynamical Isometry

A neural network can be viewed as a series of blocks, where the Jacobian of block j is J_j. If $\forall j$, $\phi(J_j, J_j^{\mathsf{T}}) \approx 1$ and $\varphi(J_j, J_j^{\mathsf{T}}) \approx 0$, the network achieves "Block Dynamical Isometry," which ensures stable gradient flow without vanishing or explosion.

B.2 Lemma 2: Multiplication Principle

Given $J = \prod_{j=L}^{1} J_j$ as a product of independent block Jacobians, if JJ^{T} is unitarily invariant at the first moment, then

$$\phi(JJ^{\mathsf{T}}) = \prod_{j=L}^{1} \phi(J_j, J_j^{\mathsf{T}}). \tag{14}$$

B.3 Lemma 3: Addition Principle

If at most one J_j is not centered, then

$$\phi(JJ^{\mathsf{T}}) = \sum_j \phi(J_j, J_j^{\mathsf{T}}). \tag{15}$$

B.4 SCN as General Linear Transformations

Pooling, upsampling, and concatenation layers can be expressed as generalized linear transforms. Batch normalization and convolution layers satisfy similar properties [96]. For this interpretability analysis, we follow prior SNN training analyses and approximate the local Jacobian of TASA under surrogate gradients [67]. For SCN, the block Jacobian satisfies:

$$\phi(J_j, J_j^{\mathsf{T}}) = \frac{2}{\alpha_2^{j-1}}, \tag{16}$$

where α_2^{j-1} is the second moment of the input to the block. By controlling input moments, Omni-SpikeDet maintains $\phi(J, J^{\mathsf{T}}) \approx 1$, ensuring stable gradients throughout the network.

B.5 Gradient Norm Equality Proof for Omni-SpikeDet

For SCNs, the residual and shortcut paths have Jacobians J_{res} and J_{sc}. Using the addition and multiplication principles, the overall block Jacobian satisfies:

$$\phi(J_{SN}, J_{SN}^{\mathsf{T}}) = \phi(J_{res}, J_{res}^{\mathsf{T}}) + \phi(J_{sc}, J_{sc}^{\mathsf{T}}) = \frac{2}{\alpha_2^{l-1}} \approx 1, \tag{17}$$

ensuring "Block Dynamical Isometry." This analysis provides an interpretation of why the SCN-based design can help maintain stable gradient propagation in practice.

C Complexity and Energy Analysis of CTI

Given multi-scale feature maps $\{X^{(n)}\}_{n=1}^{3}$, where $X^{(n)} \in \mathbb{R}^{H_n \times W_n \times D_n}$, we first reshape them into flattened representations:

$$X^{(n)} \in \mathbb{R}^{N_n \times D_n}, \quad N_n = H_n W_n. \tag{18}$$

C.1 Cross-scale Feature Normalization

To enable cross-scale interaction, CTI first aligns all feature maps into a unified spatial and channel space. This is achieved via a set of upsampling and downsampling operators:

$$\tilde{X}^{(n)} = f_n(X^{(n)}), \quad \tilde{X}^{(n)} \in \mathbb{R}^{N_l \times D}, \tag{19}$$

where $f_n(\cdot)$ denotes the scale alignment function implemented by SCN up/downsampling, and N_l is the target spatial resolution.

The complexity of this stage is

$$OP_{\text{norm}}^{(l)} = \sum_{n=1}^{3} O(N_l D_n k_s^2), \tag{20}$$

where k_s is the kernel size of the spiking convolution.

C.2 Visual Feature Transformation

Each normalized feature is further processed by a spiking convolution neuron (SCN):

$$V^{(n)} = \text{SCN}(\tilde{X}^{(n)}), \quad V^{(n)} \in \mathbb{R}^{N_l \times D}. \tag{21}$$

The associated cost is

$$OP_{\text{vis}}^{(l)} = \sum_{n=1}^{3} O(N_l D k_s^2). \tag{22}$$

C.3 Text-conditioned Feature Injection

Let $W \in \mathbb{R}^{D_t}$ denote the text embedding. It is first transformed into a visual-compatible representation:

$$t = h(W) \in \mathbb{R}^D, \tag{23}$$

where $h(\cdot)$ is implemented by a lightweight SCN block.

To match the spatial dimension, the text embedding is broadcast:

$$T_l = \mathbf{1}_{N_l} t^\top \in \mathbb{R}^{N_l \times D}. \tag{24}$$

This introduces a negligible cost:

$$OP_{\text{text}}^{(l)} = O(D_t D). \tag{25}$$

C.4 Cross-scale Routing and Gating

The visual features and text embedding are concatenated:

$$Z = \text{Concat}\left(V^{(1)}, V^{(2)}, V^{(3)}, T_l\right) \in \mathbb{R}^{N_l \times 4D}. \tag{26}$$

A lightweight modulation branch $\psi(\cdot)$, composed of SCN and max-sigmoid operations, generates routing logits:

$$G = \psi(Z) \in \mathbb{R}^{N_l \times 3}. \tag{27}$$

The normalized routing weights are obtained via softmax:

$$[\alpha, \beta, \gamma] = \text{Softmax}(G), \quad \alpha + \beta + \gamma = 1. \tag{28}$$

The complexity of the gating branch is

$$OP_{\text{gate}}^{(l)} = O(N_l D k_c^2), \tag{29}$$

where k_c is the kernel size of the routing convolution.

C.5 Feature Fusion

The final fused representation is computed as:

$$Y = \alpha \odot V^{(1)} + \beta \odot V^{(2)} + \gamma \odot V^{(3)}. \tag{30}$$

A final SCN produces the output:

$$C_l = \text{SCN}(Y). \tag{31}$$

The fusion cost is

$$OP_{\text{fuse}}^{(l)} = O(3N_l D). \tag{32}$$

C.6 Total Computational Complexity

Summing all components, the total complexity of CTI at level l is:

$$OP_{\text{CTI}}^{(l)} = OP_{\text{norm}}^{(l)} + OP_{\text{vis}}^{(l)} \tag{33}$$

$$+ OP_{\text{text}}^{(l)} + OP_{\text{gate}}^{(l)} + OP_{\text{fuse}}^{(l)}. \tag{34}$$

Ignoring lower-order terms, we obtain:

$$OP_{\text{CTI}}^{(l)} = O\left(N_l \sum_{n=1}^{3} D_n k_s^2 + N_l D k_c^2 + N_l D\right). \tag{35}$$

Discussion. Unlike dense cross-modal attention, CTI avoids explicit pairwise interaction across all spatial locations and all text tokens in the main fusion pathway. Although the initial text-conditioned modulation involves the prompt vocabulary, the subsequent routing and cross-scale fusion have bounded overhead with respect to the number of scales.

C.7 Energy Analysis

The complexity expressions above include all CTI sub-operations, whereas the 19.8 mJ estimate in Table 7 counts CTI convolutions but omits dense gating, text similarity, and memory movement.

D Additional Experimental Results

D.1 Closed-Set ANN Comparison

Published compact closed-set models include YOLO11-S (47.0 COCO AP, 9.4M parameters) and YOLO26-S (48.6 AP, 9.5M parameters) [91].

D.2 Efficiency Comparison

Table 8: Comparison of efficiency-related metrics. Energy is operation-based; latency is measured on A800.

Methods	FLOPs (G)	Energy (mJ)	Latency (ms)	Params (M)	AP
YOLO-World-S	51.0	117.3	8.4	13.0	37.6
Omni-SpikeDet	117.3	19.8	9.1	4.2	38.9

D.3 Binary-Inference Activity

Table 9: Binary-inference activity. Firing rate is normalized; spike count is the absolute per-image total.

Setting	Firing rate	Spikes/image
Binary SNN baseline	49.6%	34.1M
Omni-SpikeDet	45.5%	62.7M

Omni-SpikeDet has lower normalized firing (45.5% vs. 49.6%) but more spike events (62.7M vs. 34.1M), reflecting its larger feature volume.

E Sensitivity Analysis

E.1 Effect of Timestep Number and Integer Range

Table 10: Controlled sensitivity analysis of timestep number T and integer range D.

$T \times D$	Energy (mJ)	mAP@50
1×1	4.2	60.0
1×4	4.1	65.8
4×1	15.4	66.6
4×2	13.5	67.9

Table 10 reports a controlled $T \times D$ study with energy estimates specific to this configuration.

At $T = 1$, increasing D from 1 to 4 raises AP_{50} from 60.0 to 65.8 with similar estimated energy; at $D = 1$, increasing T from 1 to 4 raises AP_{50} to 66.6 at higher cost. Energy is non-monotone because the estimate uses observed spike activity.

F Implementation Details and Reproducibility

F.1 Training Configuration

Unless stated otherwise, we use $T = 4$ binary inference steps and $D = 2$ as the training-time response cap. COCO ablations share backbone scale, O+G+C pre-training, 80-epoch fine-tuning, and evaluation. The full variant uses three seeds; other ablation variants and large-scale pre-training results are single-run.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction of this paper accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of this work are clearly discussed. Detailed descriptions are provided in Section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We have introduced all the details in our theoretical result.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: This paper has provided detailed information for reproducing the experimental results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Code is not included with this version; baseline construction, configurations, and evaluation scripts remain pending release.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: This paper has specified all the training and test details, such as dataset and hyperparameters.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: The full COCO model reports 47.2 ± 0.2 AP over three seeds; other controlled variants and large-scale OVD pre-training comparisons are single-run, so uncertainty is not reported for every result.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: GPU type is reported, but per-run GPU-hours, memory use, and total project compute are not yet available.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This paper meets the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This paper has no societal impact to the best of our knowledge.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper poses no such risks since it does not release data or models that have risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: This paper has properly cited the original paper that produced the code package or dataset.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: This paper only uses LLMs for writing and editing.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.