6. Experiment Details

6.1. Attention Maps in Qwen2-0.5B Decoding

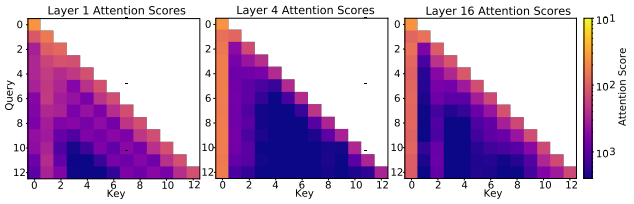

Figure 6. Attention maps during the decoding process of Qwen2-0.5B.

Based on the attention maps during the decoding process of Qwen2-0.5B, We draw the attention scores for different layer as fellow:

$$\text{Attention Score}(Q, K) = \frac{QK^T}{\sqrt{d_k}} \tag{6}$$

Where, Q, K are the Query and Key matrix in the transformer Decoder layer, softmax$(\cdot)$ .as show in Figure 6, we have reached an approximate conclusion [4]. We can observe that in the first layer, attention is distributed relatively smoothly across different types of tokens. In the deeper layers, starting from local attention, attention scores are aggregated onto system prompts, instructions, and output tokens, while attention to image tokens becomes quite sparse. In the deeper layers, there are strong vertical lines (in the system prompts) that dominate most of the attention scores. The presence of these strong vertical lines indicates that certain input tokens remain highly attended to throughout the decoding process. If large-scale, high-intensity attention to image tokens were also maintained in the deeper layers, it would suggest that a significant amount of visual information is still needed for inference at those stages. However, as seen in the visualization, the deeper layers primarily focus on some key text tokens, which precisely indicates that the processing of images has already been completed in the earlier layers, and most of the image information has been "condensed" into the representations.

6.2. Adapter Architecture Detail

Architecture I (Double Cross-Modal Fusion). As show in Figure 7, this design leverages two sequential cross-attention layers to incorporate both textual and visual cues into the Detector. First, the image embeddings $\mathbf{E_{V}}_L$ attend to text embeddings $\mathbf{E_T}$, ensuring high-level semantic alignment via a *text-guided image fusion*. Second, the fused features further guide the Detector through a *prompt-enhanced detection* stage, where Detector queries attend to the fused representation. A zero-initialized gating mechanism adaptively combines these cross-modal features back into the Detector's backbone.

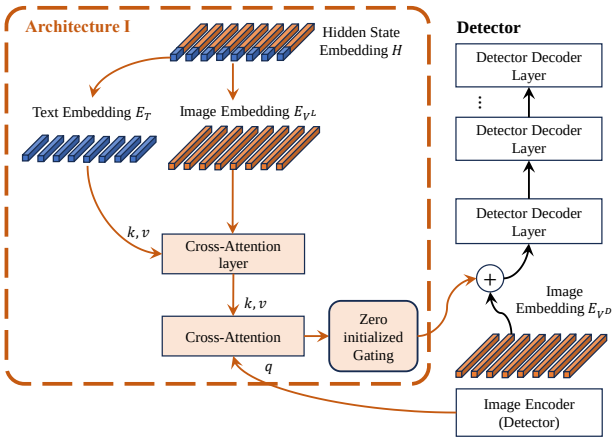

Figure 7. The adapter architecture I of our approach.

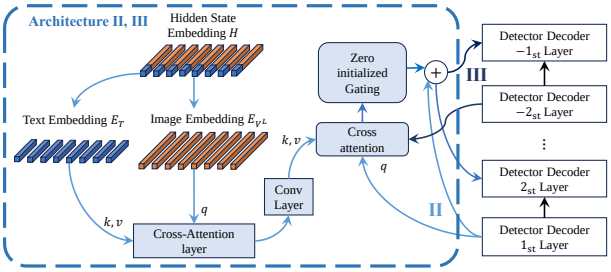

Figure 8. The adapter architecture II, III of our approach.

Architecture II, III **(Late Prompt Projection).** As show in Figure 8, we remove the second cross-attention step to streamline the flow into the Detector. A 3×3 convolution directly transforms the fused adaptation prompts into the compatible feature dimension before feeding them into the final cross-modal decoder.

6.3. Experiment Configuration

All experiments are conducted on eight NVIDIA A100 GPUs. During Stage 1 (pre-training) and Stage 2 (fine-tuning), we train the MLP layer for exactly one epoch each. For pre-training we use a batch size of 128, a learning rate of 1×10^{-3}, weight decay of 0.03, and no warm-up. For fine-tuning, the warm-up ratio is set to 0.03, while the batch size, learning rate, and weight decay are fixed at 128, 4×10^{-5}, and 0, respectively. In Stage 3 we train the adapter for one epoch with a batch size of 64, employing cosine learning-rate scheduling (base learning rate 2×10^{-4}) without warm-up or weight decay. The MLP layer and the LLM are co-trained with the adapter using a separate learning rate of 4×10^{-5}.

The hyper-parameters for Stages 1–3 are summarised in Table 9a. To investigate the impact of model scale, we vary LLM parameter sizes as detailed in Table 9b, with the results reported in Section 4.1.2. Table 9c lists the settings used to evaluate alternative adapter architectures; the corresponding

Vision Decoder Share from Detector	
MLLM	InternVL2-1B
ℓ_{LM}	2
AP to ℓ_D	6
Train Dataset	Object365, COCO, Flickr30k
Stage 1 lr	MLP: $1e^{-3}$
Stage 2 lr	MLP: $2e^{-5}$
Stage 3 lr	Adapter:$2e^{-4}$
Adapter arch.	IV

(a) Experiment configuration details for vision decoder share from detector.

Different MLLMs	
MLLM	InternVL2: 1B, 2B, 8B
ℓ_{LM}	2
AP to ℓ_D	6
Train Dataset	Object365, COCO, Flickr30k
Adapter arch.	IV

(b) Experiment configuration details for different LLMs.

Adapter Architectures	
MLLM	InternVL2-2B
ℓ_{LM}	8
AP to ℓ_D	1, 6
Train Dataset	COCO, Flickr30k
Adapter arch.	I,II,III,IV

(c) Experiment configuration details for different adapter architectures.

Ablations: Decoder Layers	
MLLM	InternVL2-1B
ℓ_{LM}	-1, 2, 4, 8, 24
AP to ℓ_D	6
Train Dataset	Object365, COCO, Flickr30k
Adapter arch.	IV

(d) Ablation details for different decoder layers.

Table 9. Experiment configuration summary.

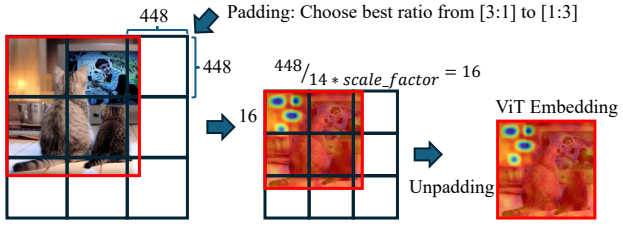

Figure 9. Dynamic Image Processing

OmniLabel scores appear in Table 1. Finally, Table 9d specifies the setup for examining different decoder layers, and the ablation results are presented in Section 4.3.

6.4. Dynamic Image Processing

In Section 4.1.2 and Section 4.3, Since the MLLM and Detector's Image encoder process the same image in a single inference, and considering that the Hidden state cropped from the Vision token positions in LLMs corresponds to the Vision Embedding processed by the detector, we are inspired by LLaVA[21]'s approach to dynamic image processing. The Vision Embedding sent to LLMs is processed as shown in Figure 9. For a given image, based on a 448×448 patch, rectangles are generated from a 1:3 to 3:1 aspect ratio and matched to the image. The image is then padded to the smallest rectangle that can contain the original image and sent to the Vision Transformer (ViT) of LLMs. The size of the patch embedding should be $448/(14 \times \text{scale factor} = 2) = 16$. Finally, the ViT embedding is cropped according to the previous padding ratio to restore the original image's aspect ratio.

MME	InternVL2-1B	Swin-T & Qwen2-0.5B
total	1363.0	1027.0
existence	180.0	158.3
count	118.3	105.0
position	126.7	66.7
color	135.0	128.3
posters	110.2	75.9
celebrity	146.8	131.8
scene	148.5	120.8
landmark	132.5	83.3
artwork	140.0	80.0
OCR	125.0	77.5
total	419.29	273.57
commonsense reasoning	99.29	68.57
numerical calculation	62.50	62.50
text translation	162.50	70.00
code reasoning	95.00	72.50

Table 10. Performance comparison of InternVL2-1B, Swin-T & Qwen2-0.5B on MME Perception and Cognition tasks.

6.5. MLLM Alignment (Swin-T & Qwen2-0.5B)

We evaluated the alignment of Swin-T & Qwen2-0.5B on MME and compared it with the original method of InternVL2-1B (InternVL Vit & Qwen2-0.5B) as show in Table 10. To evaluate the impact of replacing the vision encoder and aligning the LLM on MLLM performance, we compared the score gaps between InternVL2-1B and Swin-T & Qwen2-0.5B on the MME benchmark. Compared to InternVL2-1B, Swin-T & Qwen2-0.5B scored 1027 on perception, a decrease of 335 (InternVL2-1B scored 1362.97), and 273.57 on cognition, a decrease of 145.72 (InternVL2-1B scored 419.29). It is worth noting that, although more advanced alignment strategies may exist, our work does not aim to achieve maximum alignment between an extremely lightweight vision encoder (Swin-T) and the LLM, particularly without employing strategies such as Dynamic Tiling.

It is worth noting that, the shared Swin-T vision encoder output undergoes a pixel shuffle operation with scaling factor 4, reducing spatial dimensions (h, w) by a factor of 4 while expanding the channels by a factor of 16. This transformed

output is repeated and then truncated to match the MLP projection layer dimension (4096), aligning the feature spaces through pretrained MLP layers.

7. Why MLLM Hidden States Cannot Replace Visual Detectors

One might be tempted to ask: *"Can we simply replace a specialized visual detector with a LLM, MLLM or VLM and still achieve comparable performance?"* To answer this question, Table 11 presents a head-to-head comparison between our detector-based pipeline and an end-to-end LLM approach across multiple benchmarks. As the results make clear, relying solely on the LLM without a dedicated object detector leads to a substantial drop in accuracy and consistency.

Table 11. **Detector vs. MLLM Hidden-State Features on COCO-2017 *val*.** Baselines are two GroundingDINO variants and vision-only SwinTiny-BERT; substitutions plug frozen decoder states from LLaVA-1.5 (L8) or InternVL2-2B (L2/L8).

Method	$AP_{50:95}$	AP_{50}	AP_{75}
G-DINO-PRE	31.23	44.37	34.09
G-DINO-COCO SFT	57.23	73.27	63.18
SWINTINY-BERT	41.97	57.45	45.85
LLAVA-L8	24.19	39.47	24.94
INTERNVL2-L2	38.28	55.25	41.18
INTERNVL2-L8	42.64	63.77	44.92

GROUNDINGDINO-PRE establishes a detector base on Open-GroundingDINO [23] that has *never* been exposed to COCO, being trained only on Objects365, GoldG, and Cap4M; GROUNDINGDINO-COCO SFT fine-tunes the same weights on COCO. SWINTINY-BERT serves as a *vision-only* control whose Swin-T backbone and BERT text branch use only same framework as Open-GroundingDINO [23] is identical to that used in the multi-modal runs yet, like the MLLMs, has never seen COCO or any object-detection data. The three substitution variants, LLAVA-L8, IVL2-L2, and IVL2-L8, replace the Swin feature map with hidden states taken from decoder layers 2 or 8 of LLaVA-1.5 or InternVL-2B, enabling a direct test of whether raw MLLM representations can stand in for detector-oriented visual embeddings.

As show int the Table 11, fine-tuning the vision backbone on COCO (GDINO-COCO) improves $AP_{50:95}$ by **+26** points over its out-of-domain counterpart, underscoring the importance of spatial alignment learned from explicit detection supervision. By contrast, treating MLLM activations as a drop-in surrogate for visual tokens is ineffective: the strongest variant (IVL2-L8) performance is similar to the fully visual SWINTINY-BERT but with a parameter count far larger than SWINTINY-BERT, and it remains -14.6 points below the COCO-tuned baseline. The gap widens to -33 points when substituting LLAVA-L8, revealing that decoder-layer semantics alone carry minimal localisation cues.

This discrepancy arises because MLLM hidden states are produced after heavy token mixing and rotary positional encoding optimised for caption generation; they lack the multi-scale geometry and inductive biases embedded in convolutional or ViT backbones. Without an explicit fusion mechanism that jointly conditions on language and vision, the detector cannot recover accurate object coordinates, leading to systematic localisation failure. The comparison between SWINTINY-BERT (whose Swin weights, like the MLLMs, have *never* seen COCO) and the substitution runs highlights the central claim of this study: **knowledge fusion is indispensable—naive replacement of visual embeddings with isolated MLLM features is not a viable route to open-vocabulary detection.**

8. Case Study

We compared test results on OmniLabel using InternVL2-1B for adaptation prompts against directly applying GroundingDINO on selected samples. We excluded all category-based targets and focused solely on the detection performance of descriptive targets. As shown in Figure 10, GroundingDINO struggles to understand target descriptions involving quantities, states, ages, colors, or spatial relationships. However, when the LLM provides semantic-level understanding, the detector's ability to recognize such targets improves significantly.

Notably, the adaptation prompt functions as expected, guiding the detector rather than allowing the adapter to dominate the grounding task. As seen in the recognition results in Figure 10, GroundingDINO often fails to locate individuals, instead detecting entire objects, as in the case of "The computer monitors on the desk that are turned on." Additionally, it struggles to interpret positional relationships in text descriptions, such as "the sandwiches that are each on the same plate with the other." With the adaptation prompt, the detector retains an overall understanding while incorporating individual positional relationships. Furthermore, it corrects errors in spatial logic relationships.

9. Implementation Details

9.1. Dataset Introduction

Table 12 provides the attributes, sizes, and functional descriptions of all datasets used in our stages 1, 2, 3 and evaluation, to illustrate the support they offer during the process. All datasets are distributed under licences that permit non-commercial academic research.

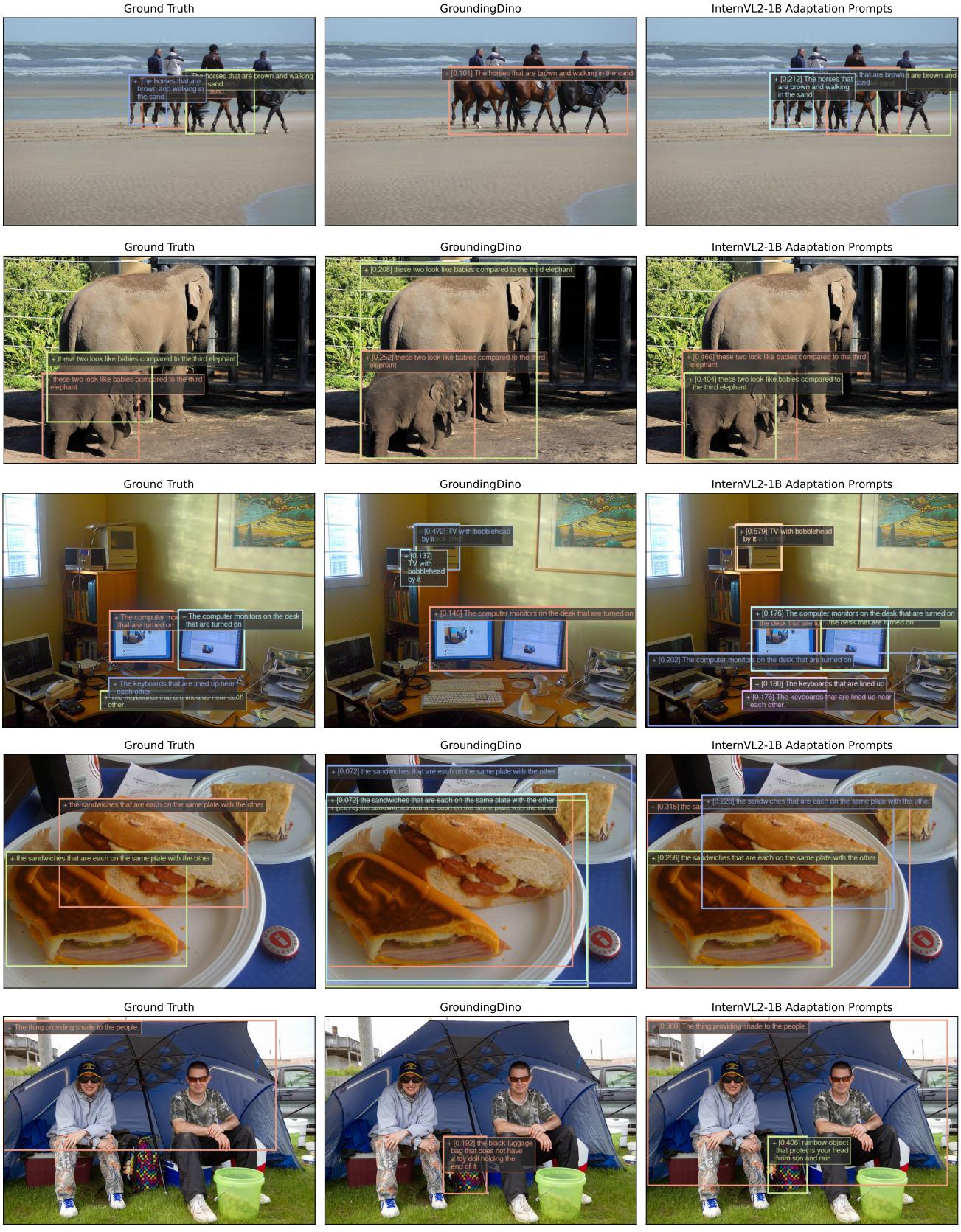

Figure 10. Case Study for OmniLabel using InternVL2-1B to provide adaptation prompts versus directly using GroundingDINO.

Table 12. Introduction of datasets used in stage 1, stage 2, and stage 3.

Dataset	Introduction
Pretraining Data for Stage 1	
LAION-CC-SBU 558K	The LAION-CC-SBU dataset is a curated subset of the LAION, Conceptual Captions (CC), and SBU datasets. It comprises 558K image-caption pairs for the pretraining stage for feature alignment in visual instruction tuning. [21]
Instruction Tuning Data for Stage 2	
ShareGPT4V	ShareGPT4V dataset is curated from LAION, CC, SBU, SAM, COCO, web-landmark, web-celebrity, wikiart, etc, resulting in total 102K high-quality image-text pairs with the help of powerful GPT4-Vision [3].
SFT	SFT dataset comprises approximately 665K multimodal instruction-following samples, facilitating improved alignment of visual-language models to human instructions [3].
ChartQA	ChartQA is a domain-specific visual question-answering dataset containing 18K samples designed explicitly for interpreting various types of charts, including bar graphs, pie charts, and line plots. [26]
AI2D	AI2D includes approximately 12K annotated diagrams paired with structured question-answer data, particularly aimed at evaluating multimodal reasoning over scientific diagrams. [15]
DocVQA	DocVQA contains 10K samples that involve complex question-answering tasks over visually rich document images, emphasizing text recognition, layout analysis, and semantic comprehension. [27]
Grounding Data for Stage 3	
Objects365	Objects365 is a large-scale object detection dataset featuring over 1.7 million images with dense annotations covering 365 common object categories, enhancing general object detection capabilities. [33]
COCO2017	COCO2017 provides around 118K training images annotated for object detection, and captioning tasks, widely used as a benchmark in computer vision research.
Flickr30k	Flickr30k is a standard multimodal dataset with 31K images, each annotated with five descriptive captions, commonly utilized for improving image captioning and cross-modal retrieval tasks. [43]
Evaluation Data	
RefCOCO/+/g	The RefCOCO/+/g datasets consist of referring expression comprehension tasks, where models must identify objects in images based on natural language descriptions. REFCOCO has approximately 142K referring expressions, while RefFCOCO+ and RefCOCOg provide more challenging and generalized scenarios. [44]
Omnilabel	Omnilabel is a multimodal benchmark dataset containing diverse visual-language tasks designed to comprehensively evaluate the generalization and zero-shot capabilities of visual-language models across various tasks and domains. [32]

9.2. Model Introduction

To foster clarity and reproducibility, we first outline the *functional role* of every third-party model or code base incorporated into LED (Table 13).

10. Broader Impacts

Potential benefits. By enabling *lightweight open-vocabulary grounding* with minor extra FLOPs, LED can (i) improve on-device perception for assistive robotics and smart prosthetics; (ii) lower the computational barrier for researchers in low-resource regions to experiment with vision–language models; and (iii) accelerate scientific discovery in ecology, astronomy, and digital humanities, where long-tail object categories are common.

Potential harms. Hallucinating an object that is not present—could propagate misinformation through downstream captioning or retrieval systems. Biases inherited from pre-training corpora may yield disparate false-positive rates across demographic groups, disproportionately affecting already-marginalised communities.

Mitigation strategies. **More advanced models and algorithms.** As the LLM field continues to advance, the Hallucinating problem will continue to be overcome and optimized.

Table 13. Concise introductions for third-party models and code bases.

Model / Code	Introduction
Qwen2-0.5B	A 0.5-billion-parameter decoder-only LLM trained on ∼2 T multilingual tokens; we tap its early hidden states for knowledge fusion in Stage 3 of LED [37].
InternVL2 (1 B / 2 B / 8 B)	A family of vision–language foundation models pretrained on 20 M image–text pairs; the 1 B and 2 B variants are used as alternative language decoders in our ablations [18, 28, 39, 49, 50].
LLaVA-1.5	An open-source multimodal chat model aligning a CLIP vision encoder with a Vicuna language head via instruction tuning; included as an external baseline for grounding performance [22].
Open-GroundingDINO	An open-vocabulary detector that couples a Swin-T backbone with a text encoder; serves as the base detector into which our LED adapters are inserted [23].
Swin-T	A hierarchical vision encoder pretrained on ImageNet-22K; provides four-stage feature maps consumed by the detector backbone [24].

Research-only weight release. Pre-trained weights will initially be released under an academic non-commercial licence; commercial use will require a separate agreement that enforces compliance with a no-surveillance clause. **Transparent data lineage.** All training datasets are listed with licences and provenance (Section 9.1); no proprietary or private images were used, reducing privacy concerns at source.