Dynamic Resolution Routing for Efficient Egocentric Grounding

Huixin Sun[1], **Wangbo Zhao**[2], **Fanyue Wei**[1], **Qiuxia Lin**[3], **Pengzhan Sun**[1], **Angela Yao**[1]

[1]National University of Singapore
[2]The Hong Kong University of Science and Technology
[3]Nanyang Technological University

Abstract

Egocentric visual grounding requires high-resolution inputs to localize small objects. However, scaling Multimodal Large Language Models to this domain is constrained by the excessive cost of visual token processing. We identify that current efficient strategies based on token reduction are unreliable for selecting object-centric spatial evidence. To overcome this, we propose *SmartRes*, a framework that performs efficiency optimization in the pixel space via dynamic resolution routing. SmartRes first encodes a low-resolution view for global context and uses a lightweight router to activate high-resolution patches in object-centric regions and constructs an order-preserving visual sequence. To further enable robust routing under severe foreground-background imbalance, we introduce a margin-regularized routing objective that increases foreground-background logit separation and improves foreground recall. Experiments on Ego4D and EgoIntention show that SmartRes reduces visual tokens by up to 67% while retaining 86.4% of full-resolution performance, and achieves up to $1.66\times$ faster inference than state-of-the-art token reduction methods with higher accuracy. Furthermore, strong performance on small object grounding indicates the effectiveness of *SmartRes* towards egocentric applications. Code is available at `https://github.com/HuixinSun/SmartRes`.

1 Introduction

Egocentric grounding [1] aims to localize target objects based on natural-language queries within first-person video streams. This capability is fundamental for understanding human-environment interactions and is useful for applications in robotics, AR/VR, and long-form video understanding. The latest Multimodal Large Language Models (MLLMs) [2, 3] have impressive visual grounding capabilities, achieving over 90% accuracy on general benchmarks such as RefCOCO [4, 5], yet their accuracy remains limited in egocentric scenarios [6].

A key challenge in egocentric scenes is that target objects are often small and are observed under rapid viewpoint changes during human interactions [7]. As a result, high-resolution inputs are essential for preserving the fine-grained details required for perception [8]. However, given that egocentric data are captured at significantly higher and more diverse native resolutions, scaling state-of-the-art MLLMs to this domain becomes prohibitively expensive. For example, a 3780×1920 frame in Any Resolution encoders [9] can result in 9.3k visual tokens. When generating concise text responses that output bounding boxes, the visual encoding cost becomes dominant, consuming up to 66.5% of the total end-to-end inference budget, as analyzed in Fig. 1 (e). This poses a major bottleneck for inference and constrains deployment in edge devices; such a challenge is still underexplored.

Existing efficient methods reduce computation by pruning or merging tokens in the latent space, using heuristics such as decoder attention [10, 11], feature similarity [12], or activation statistics [13].

Correspondence to: Huixin Sun <huixin.sun@nus.edu.sg>.

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

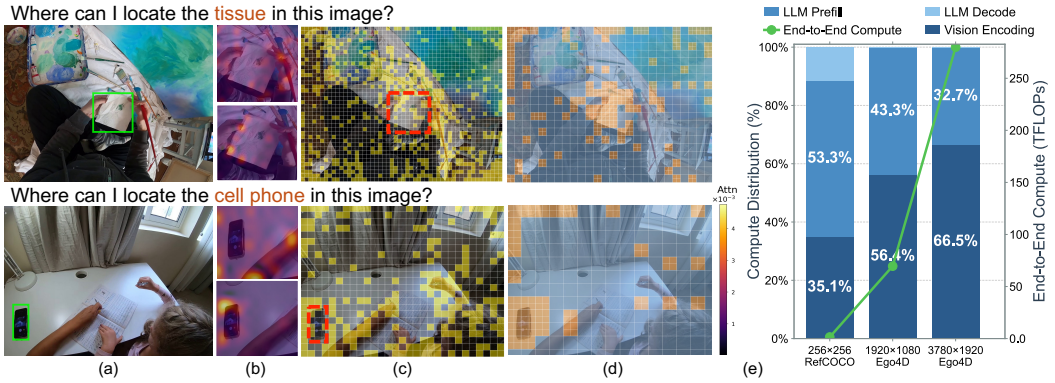

Figure 1: **Left: Pixel-space dynamic routing *vs.* latent-space token pruning.** (a) Input image with the ground-truth target; (b) Vanilla cross-attention (top) drifts to artifacts, whereas our method (bottom) calibrates it to the object. (c) FastV [10] retains tokens (■) mostly from the background, failing to preserve fine-grained object evidence. (d) SmartRes allocates high-resolution computation (■) to object-centric regions and keeps the remaining visual context compact (■). **Right: the visual encoding bottleneck.** (e) End-to-end FLOPs (green) of Qwen2.5-VL-3B-Instruct on RefCOCO and Ego4D frames grow rapidly with input resolution, shifting the compute from LLM prefilling to vision encoding.

However, these heuristics are unreliable for identifying relevant objects in egocentric scenes. As shown in Fig. 1 (b), attention can be affected by noisy context, *e.g.*, high-activation artifacts [14], or drift to semantically salient but irrelevant regions, such as hands. This leads to misguided token selection, as illustrated in Fig. 1 (c), where target object evidence can be discarded while background tokens are retained. Such methods, as we later show, can severely degrade grounding performance even under moderate token retention ratios.

To avoid using noisy statistical proxies, we shift efficiency optimization from post-hoc token pruning to proactive resolution allocation in the pixel space. To this end, we introduce *SmartRes*, a framework for dynamic resolution routing at the patch level. SmartRes utilizes a low-resolution view to provide global spatial guidance, enabling a lightweight router to activate high-resolution patches in object-centric regions, as shown in Fig. 1 (d). This yields a variable-length token sequence that interleaves localized high-resolution evidence with compact low-resolution context. The router is supervised by box-projected token labels to preserve candidate foreground evidence; however, this is challenged by severe foreground–background imbalance in egocentric data. To address this, we introduce a margin-regularized routing objective that enforces separation between foreground and background logit distributions, preserving object integrity under a fixed compute budget. As shown in Fig. 1 (b), *SmartRes* calibrates visual attention toward object-centric areas via robust foreground selection. Meanwhile, *SmartRes* reduces the visual encoding costs at its source, and the shortened visual sequence in turn lowers LLM compute.

Experiments on egocentric benchmarks show that *SmartRes* establishes a new Pareto frontier for efficient egocentric grounding. Specifically, the proposed *SmartRes-Lite* retains 86.4% of full-resolution performance using only 33% of visual tokens, exceeding down-scaling at a comparable 32% token budget by 34.1 percentage points in performance retention. Furthermore, the performance-oriented *SmartRes-Pro* retains 89.9% of full-resolution performance at a 55% token budget. SmartRes-Lite runs 1.66× faster than the 50%-token Dyn-LLaVA baseline [15]. The results demonstrate that SmartRes offers a preferable trade-off between fine-grained perception and inference efficiency for egocentric grounding. Our contributions are summarized as follows:

- We observe that latent-space token reduction guided by heuristic proxies is unreliable for preserving object-centric spatial evidence in egocentric grounding. To address this, we introduce *SmartRes*, which replaces post-hoc latent pruning with proactive pixel-space selection through dynamic resolution routing.
- We introduce a patch-level allocation mechanism where a low-resolution view guides the dynamic activation of high-resolution processing. To ensure robust routing under severe foreground-background imbalance, we introduce a margin-regularized objective that explicitly separates foreground and background logits.
- *SmartRes* reduces visual token consumption by up to 67% while maintaining comparable performance. Notably, it operates 1.66× faster than leading baselines, establishing a new Pareto frontier for efficient grounding on egocentric benchmarks.

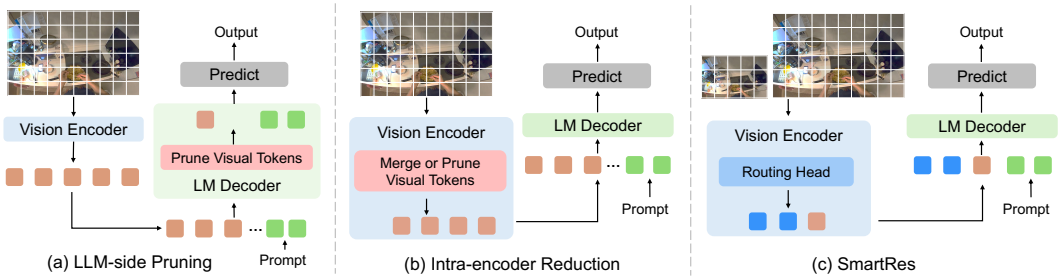

Figure 2: **Visual token reduction paradigms.** Token types: ■ text, ■ LR visual, ■ HR visual (a) LLM-side pruning discards tokens with decoder-side heuristics such as attention scores [10]. It lowers LLM compute but has full-resolution visual encoding costs. (b) Intra-encoder reduction merges or prunes tokens inside the vision encoder with heuristics such as feature cosine similarity [12]. (c) *SmartRes* performs low-resolution guided routing before full high-resolution encoding, and reduces compute in both the vision encoder and the LLM.

2 Related Work

Visual grounding. Visual grounding [4, 16, 17] aims to localize image regions referred to by natural-language expressions. Early works mainly studied referring expression comprehension (REC) [5, 18, 19], focusing on aligning descriptive language with image regions. Recent MLLMs have improved visual grounding through stronger language-conditioned reasoning [20, 21]. However, localizing small objects remains challenging [22] due to inherently limited visual evidence and high sensitivity to localization errors. This issue is especially pronounced in egocentric visual grounding [1, 6, 23, 24], where targets are often small, partially occluded, and surrounded by hands, tools, and cluttered interaction context. Recent benchmarks (*e.g.*, Ego4D [1], RefEgo [23], and EgoIntention [6]) have driven progress in this setting, primarily emphasizing task formulation, data, and localization accuracy. Yet efficient inference for high-resolution egocentric grounding remains less explored, especially for small, resolution-sensitive objects. Our work targets this gap by studying compute-efficient grounding under reduced visual token budgets.

Visual token reduction. Recent MLLMs enable high-resolution visual understanding through adaptive resolution strategies [9, 25]. However, the resulting growth in visual tokens introduces substantial memory and latency overhead, motivating visual token reduction methods [26]. Training-free methods insert lightweight pruning or merging operations into frozen backbones, either in the vision encoder [12, 13, 27, 28] or during LLM prefilling [10, 11, 29, 30]. Re-training methods, on the other hand, learn explicit compression modules end-to-end [31–34]. While effective at reducing token sequences, these methods rely on attention, similarity, or learned importance as proxies for spatial utility, which can discard fine-grained spatial details critical for grounding tasks.

Efficient high-resolution visual perception. Another line of work focuses on selective high-resolution processing for efficient perception, including crop- or visual-search-based methods [35, 36], region-selective sampling [37, 38], sample-level resolution request [39], and tool-augmented resolution acquisition [40]. These methods selectively acquire high-resolution evidence through crops, proposed regions, tool invocation, or whole-image resolution requests. For instance, Zoomer [36] builds an efficient image canvas from externally selected crops; TEVA [38] sparsely samples patches inside externally proposed regions; and AdaptVision [40] learns to invoke a crop tool after low-resolution reasoning. In contrast, SmartRes learns task-supervised resolution routing at the patch level, enabling object-centric high-resolution coverage without external agents or multi-stage pipelines.

3 SmartRes: Dynamic Resolution Routing

3.1 Preliminaries

Multimodal Large Language Models (MLLMs). Modern MLLMs [41, 42] typically adopt an encoder-bridge-decoder architecture. Given an input image $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$, a vision encoder maps the image into N_v visual features $\mathbf{V} \in \mathbb{R}^{N_v\times D_v}$. Concurrently, the input text is tokenized and transformed into text embeddings $\mathbf{T} \in \mathbb{R}^{N_t\times D}$ via the language model's embedding layer. To bridge the modality gap, visual features are aligned with the language model's latent dimension D using a

linear layer $\mathrm{Proj} : \mathbb{R}^{D_v} \to \mathbb{R}^{D}$. The resulting visual and textual tokens are concatenated and fed into the language model (LM):

$$\mathbf{y}_t = \begin{cases} \mathrm{LM}_{\mathrm{prefilling}}\left([\mathrm{Proj}(\mathbf{V}); \mathbf{T}]\right), & t = 1, \\ \mathrm{LM}_{\mathrm{decoding}}\left([\mathbf{C}_{t-1}; \mathbf{y}_{1:t-1}]\right), & t > 1. \end{cases} \tag{1}$$

where $\mathbf{y}_t$ denotes the t-th generated token. In the prefilling stage ($t = 1$), the model processes the entire multimodal prefix in parallel to compute the initial KV cache. For $t > 1$, the model enters the auto-regressive decoding stage, where $\mathbf{C}_{t-1}$ represents the cached KV pairs from previous steps.

MLLMs for Egocentric Grounding and Its Compute Bottleneck. Given a first-person image $\mathbf{I}$ and a grounding instruction $\mathbf{T}$, the inference process is formulated as:

$$\mathbf{B} = \mathrm{MLLM}(\mathbf{I}, \mathbf{T}) \tag{2}$$

where $\mathbf{B}$ is a coordinate tuple $[x_{\min}, y_{\min}, x_{\max}, y_{\max}]$ representing the bounding box of the target object. In this work, we identify a *fundamental tension* between the high-resolution requirements of egocentric grounding to preserve fine-grained spatial details and the resolution-dependent cost of visual processing. As shown in Fig. 1 (e), high-resolution inference is bottlenecked by vision encoding and LLM prefilling. Scaling the input from 256×256 to 3780×1920, a resolution commonly encountered in egocentric scenarios, increases the vision-encoding cost from 0.53T to 186.03T FLOPs and the LLM-prefilling cost from 0.81T to 93.70T FLOPs. In contrast, the autoregressive decoding remains lightweight as grounding outputs are short (typically $<$350). Existing token pruning (Fig. 2 (a)) and token merging (Fig. 2 (b)) methods reduce the visual sequence length before or inside the LLM but cannot tackle the dominant visual encoding cost.

3.2 SmartRes Architecture

To resolve the above tension, we propose *SmartRes*, a dynamic resolution routing framework. As shown in Fig. 3, SmartRes encodes a low-resolution view to obtain global spatial guidance, learns a patch-level router to activate high-resolution evidence, and assembles the selected HR features with compact LR context in raster-scan order. This design reduces visual processing cost at its source while preserving localized details needed for accurate grounding. We next describe the image preparation, learnable routing strategy, and visual token assembly.

Image preparation. From the original image $\mathbf{I}$, we derive two aligned views: a low-resolution image $\mathbf{I}_{\mathrm{LR}}$ for routing and a high-resolution image $\mathbf{I}_{\mathrm{HR}}$ for selective encoding. For target token ratios $r_{\mathrm{LR}}, r_{\mathrm{HR}} \in (0, 1]$, the LR and HR views are resized by isotropically scaling both height and width by $\sqrt{r_{\mathrm{LR}}}$ and $\sqrt{r_{\mathrm{HR}}}$, respectively, preserving the original aspect ratio. We then snap the resized side lengths to multiples of the effective visual-token stride $F = PM$, where P is the patch size and M is the spatial merge size. This defines aligned LR and HR routing grids and induces a HR-to-LR token correspondence; see Sec. A of the *Supplementary*. The LR view $\mathbf{I}_{\mathrm{LR}}$ is then passed through the vision encoder, and its layer-ℓ features $\mathbf{F} \in \mathbb{R}^{N_{\mathrm{LR}} \times D_{\mathrm{v}}}$ are used as input to the routing module.

Learnable patch-level resolution routing. High-resolution evidence is essential for grounding small objects, yet uniform high-resolution encoding is computationally prohibitive. SmartRes uses a query-agnostic routing policy based only on LR visual features to preserve object-centric candidates at high resolution. Query-specific disambiguation is performed by the language model using the assembled visual sequence and the referring expression. Specifically, we introduce a lightweight MLP-based routing head to predict a routing saliency map $\mathbf{S}$ from the LR features $\mathbf{F}$:

$$\mathbf{z} = \mathrm{MLP}(\mathbf{F}) \in \mathbb{R}^{N_{\mathrm{LR}}}, \qquad \mathbf{S} = \sigma(\mathbf{z}) \in [0, 1]^{N_{\mathrm{LR}}}, \tag{3}$$

where N_{LR} denotes the number of LR locations, $\mathbf{z} = \{z_i\}_{i=1}^{N_{\mathrm{LR}}}$ are routing logits, and $\sigma(\cdot)$ is the element-wise sigmoid function. This saliency map is converted into a binary routing mask $\mathbf{M}$ by thresholding $\mathbf{S}$ at a hyperparameter τ:

$$\mathbf{M} = \mathrm{STE}(\mathbf{S} > \tau) \in \{0, 1\}^{N_{\mathrm{LR}}}, \tag{4}$$

where $\mathrm{STE}(\cdot)$ denotes the Straight-Through Estimator [43]. During the forward pass, a hard binary threshold is applied; in the backward pass, gradients are directly propagated from $\mathbf{M}$ to $\mathbf{S}$, enabling end-to-end optimization of routing decisions. A selected location, $\mathbf{M}_i = 1$, indicates that the LR evidence at location i is expected to benefit from additional HR detail and activates its corresponding HR patch group; otherwise, the compact LR feature is retained. In this way, SmartRes converts dense high-resolution encoding into a sparse, learnable patch activation procedure, recovering high-resolution local details where needed while keeping the remaining visual context compact.

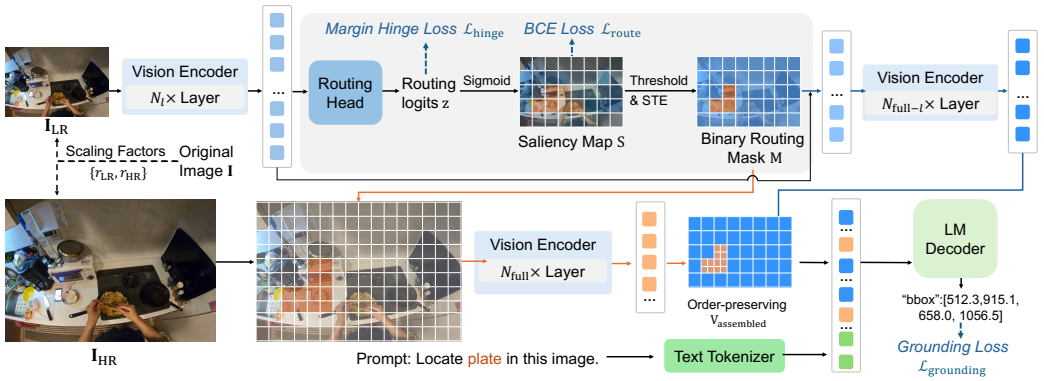

Figure 3: **Overview of the SmartRes framework.** Token types: LR feature $\mathbf{F}$, LR visual feature $\mathbf{V}_{\text{LR}}$, HR visual feature $\mathbf{V}_{\text{HR}}$, textual token. SmartRes first processes a low-resolution input $\mathbf{I}_{\text{LR}}$ to extract routing features $\mathbf{F}$. A lightweight routing head predicts a saliency map from these features, supervised by the routing loss $\mathcal{L}_{\text{route}}$ and the margin hinge loss $\mathcal{L}_{\text{hinge}}$. The Straight-Through Estimator (STE) binarizes the saliency map into a routing mask $\mathbf{M}$, which activates selected object-centric regions from the high-resolution input $\mathbf{I}_{\text{HR}}$. The selected HR features are assembled with the compact LR context in raster-scan order, yielding a routed visual sequence passed to the LM decoder.

Order-preserving visual token assembly. The routing mask $\mathbf{M}$ determines LR retention and HR replacement. After routing at layer ℓ, the LR features $\mathbf{F}$ are propagated through the remaining vision encoder blocks to obtain the low-resolution visual features $\mathbf{V}_{\text{LR}} \in \mathbb{R}^{N_{\text{LR}} \times D_v}$. The routing mask $\mathbf{M}$ is then upsampled to align with the high-resolution image. Based on the HR→LR correspondence from Eq. (14), selected patches from $\mathbf{I}_{\text{HR}}$ are processed through the vision encoder to obtain HR visual features $\mathbf{V}_{\text{HR}} \in \mathbb{R}^{N'_{\text{HR}} \times D_v}$, where N'_{HR} denotes the number of selected HR patches. The assembled visual sequence interleaves LR context with HR details:

$$\mathbf{V}_{\text{assembled}} = \mathcal{G}\left(\mathbf{V}_{\text{LR}}, \mathbf{V}_{\text{HR}}, \mathbf{M}\right). \tag{5}$$

A key property of $\mathcal{G}(\cdot)$ is that it preserves the canonical LR raster-scan order. Unselected LR locations remain as compact patch tokens, while selected LR locations are expanded in place into their corresponding contiguous HR token groups; see Fig. 4. $\mathbf{V}_{\text{assembled}}$ is zero-padded to ensure divisibility by the spatial merge unit M^2, and is then passed through Proj to yield the final visual token sequence $\mathbf{V}_{\text{final}} \in \mathbb{R}^{N_{\text{final}} \times D}$. Thus, $\mathbf{V}_{\text{final}}$ covers the entire image with a variable number of tokens without disrupting the spatial order.

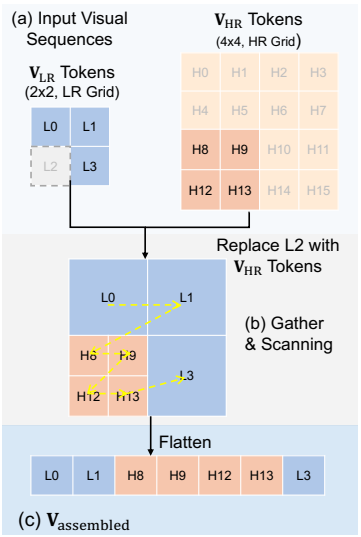

Figure 4: Selected LR locations, *e.g.*, L2, are replaced in place by their corresponding contiguous HR token groups, *e.g.*, H8, H9, H12, and H13, while unselected locations remain as LR context.

3.3 Routing Optimization with Margin Regularization

The routing strategy is trained end-to-end under the primary grounding task while receiving object-centric supervision. Specifically, we encourage high saliency scores on tokens overlapping the target box and suppress task-irrelevant background tokens. Given the LR image $\mathbf{I}_{\text{LR}}$ and the ground-truth bounding box $\hat{\mathbf{B}}$, we rasterize $\hat{\mathbf{B}}$ onto the LR patch grid to obtain token-wise labels $\hat{\mathbf{M}} \in \{0,1\}^{N_{\text{LR}}}$, and supervise the predicted saliency scores $\mathbf{S}$ using a standard binary cross-entropy (BCE) loss:

$$\mathcal{L}_{\text{route}} = \frac{1}{N_{\text{LR}}} \sum_{i=1}^{N_{\text{LR}}} \text{BCE}\left(\mathbf{S}[i], \hat{\mathbf{M}}[i]\right). \tag{6}$$

Margin hinge regularization. However, we observe that supervising the routing head solely with the BCE loss often yields suboptimal saliency maps. This result is primarily attributed to the extreme foreground-background imbalance inherent in egocentric grounding data: target objects typically occupy a minute fraction of the spatial area, while the vast majority of visual tokens represent background context. Consequently, the per-token routing loss is dominated by

an abundance of easily classified background tokens. This imbalance dilutes the gradients essential for learning sparse foreground features, resulting in reduced routing recall and overlapping logit distributions between foreground and background regions, as illustrated in Fig. 5 (a). To address this imbalance, we introduce a margin-based hinge regularizer on the routing logits to encourage class-level separation. Let $\mathcal{P} = \{i \mid \hat{M}_i = 1\}$ and $\mathcal{N} = \{i \mid \hat{M}_i = 0\}$ denote foreground and background token sets. Given routing logits $\mathbf{z} = \{z_i\}_{i=1}^{N_{\text{LR}}}$, we define the class-wise logit means as $\mu_{\mathcal{P}} = \frac{1}{|\mathcal{P}|}\sum_{i\in\mathcal{P}} z_i$ and $\mu_{\mathcal{N}} = \frac{1}{|\mathcal{N}|}\sum_{i\in\mathcal{N}} z_i$, and encourage a margin m between them:

$$\mathcal{L}_{\text{hinge}} = \left[m - (\mu_{\mathcal{P}} - \mu_{\mathcal{N}})\right]_+, \tag{7}$$

where $[\cdot]_+ = \max(0, \cdot)$. As shown in Fig. 5(b), the hinge term enlarges the foreground–background logit gap and improves foreground recall under spatial sparsity. Gradient analysis and comparisons with alternative imbalance-aware objectives, including Dice loss [44], focal loss [45], and Tversky loss [46], are provided in Sec. C of the *Supplementary*.

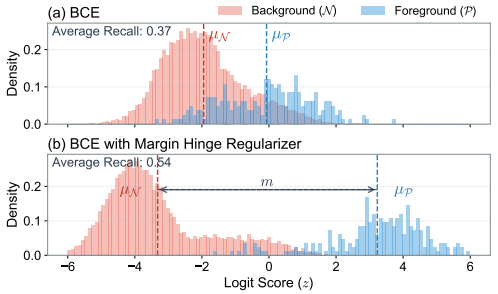

Figure 5: **Routing logit separation.** Routing logits of 3,000 randomly sampled tokens from the Ego4D test set. (a) Token-wise BCE yields highly overlapping foreground/background distributions and low foreground recall ($R = 0.37$). (b) Adding $\mathcal{L}_{\text{hinge}}$ increases logit separation and improves foreground recall ($R = 0.54$).

Optimization objective. The overall training objective combines the primary grounding loss with the routing supervision terms. For the grounding task, we optimize the standard next-token cross-entropy loss over the ground-truth output sequence $\hat{\mathbf{y}} = (\hat{y}_1, \ldots, \hat{y}_L)$, which is tokenized from the target bounding box $\hat{\mathbf{B}}$, using teacher forcing:

$$\mathcal{L}_{\text{grounding}} = -\sum_{t=1}^{L} \log P(\hat{y}_t \mid \mathbf{V}_{\text{final}}, \mathbf{T}, \hat{\mathbf{y}}_{1:t-1}), \tag{8}$$

where $\mathbf{T}$ denotes the grounding instruction, $\mathbf{V}_{\text{final}}$ denotes the routed visual token sequence, and $P(\cdot)$ is the probability distribution predicted by the model. The final objective is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{grounding}} + \lambda_{\text{route}}\mathcal{L}_{\text{route}} + \lambda_{\text{hinge}}\mathcal{L}_{\text{hinge}}, \tag{9}$$

where λ_{route} and λ_{hinge} are weighting hyperparameters.

4 Experiments

4.1 Benchmarks and Implementation Details

Datasets. We evaluate *SmartRes* on public egocentric visual grounding benchmarks, Ego4D [1] and EgoIntention [6]. These egocentric datasets contain diverse high-resolution frames and a larger proportion of small objects which require fine-grained visual details to localize; see Tab. 4. We additionally evaluate *SmartRes* on the standard referring expression comprehension (REC) benchmarks RefCOCO, RefCOCO+, and RefCOCOg [4, 5, 16].

Metrics. Grounding performance is evaluated by Precision at IoU 0.3/0.5 (P@0.3, P@0.5) and mean IoU (mIoU). Following [22], objects are grouped by relative box area S into small ($S < 0.005$), medium ($0.005 \leq S < 0.05$), and large ($S \geq 0.05$), with P_s, P_m, and P_l denoting scale-specific P@0.5. For efficiency, we compute LLM prefilling FLOPs and GPU latency with synchronized `torch.cuda.Event` timing following [11].

Compared Methods. We evaluate in two adaptive configurations: *SmartRes-Lite* (r_{LR}=10%, r_{HR}=50% budget) and *SmartRes-Pro* (r_{LR}=10%, r_{HR}=100% budget). We compare against representative state-of-the-art efficiency strategies: (1) *Token pruning* (VisionZip [13], FastV [10], Dyn-LLaVA [15], VScan [28]); (2) *Token merging* (ToMe [12]); and (3) *Naive down-sampling* baselines, where images are uniformly resized to a target token budget.

Implementation Details. For fair comparison, we reproduce all baselines and comparisons with `Qwen2.5-VL` base model and fine-tune with the same LoRA rank. We fine-tune for 3 epochs with

Table 1: **Main results on egocentric grounding benchmarks.** Vanilla denotes Qwen2.5-VL-3B-Instruct. EgoInt-C/U denote the Context/Uncommon subsets of EgoIntention. Overall is the mean P@0.5 over Ego4D, EgoInt-C, and EgoInt-U. Avg. denotes the mean performance retention rate across all 9 metrics relative to Vanilla. **Bold** marks the best value among efficient methods.

Method	Ratio ↓	Ego4D			EgoInt-C			EgoInt-U			Overall	Avg. ↑	FLOPs ↓	Lat. ↓
		P@0.5	P@0.3	mIoU	P@0.5	P@0.3	mIoU	P@0.5	P@0.3	mIoU	P@0.5	(%)	(T)	(ms)
Vanilla	100%	63.22	68.51	57.56	60.01	65.15	54.62	54.77	59.18	49.98	59.33	100.0	11.32	553.6
Down-scaling	10%	21.34	33.26	21.85	20.65	27.05	20.75	19.50	23.73	19.60	20.50	38.8	**1.55**	**139.6**
	32%	32.76	43.12	30.92	31.83	37.61	31.01	23.10	27.69	22.69	29.23	52.3	3.94	240.8
	50%	52.17	56.08	47.19	50.25	53.77	46.90	42.70	46.02	40.91	48.37	81.8	5.89	323.6
ToMe [12]	70%	22.83	35.66	25.12	20.36	33.99	22.85	17.96	29.65	20.09	20.38	42.5	8.52	408.8
VisionZip [13]	70%	34.18	42.63	31.04	30.73	38.05	27.62	28.02	34.56	25.33	30.98	54.5	8.63	305.7
FastV [10]	70%	51.65	60.29	47.03	50.33	57.58	45.30	43.61	49.83	39.44	48.53	83.3	8.92	495.5
Dyn-LLaVA [15]	70%	54.42	60.30	48.24	52.12	58.20	48.15	44.54	47.65	40.86	50.36	85.1	9.16	506.7
ToMe [12]	50%	11.41	22.62	15.05	10.78	21.36	14.41	9.74	19.40	13.08	10.64	25.7	7.42	365.5
VisionZip [13]	50%	20.28	31.64	19.74	18.87	26.23	18.53	17.22	23.82	16.96	18.79	36.0	7.98	293.4
FastV [10]	50%	45.52	53.00	41.44	47.32	54.68	42.65	35.74	40.90	32.85	42.86	73.6	7.15	410.2
Dyn-LLaVA [15]	50%	52.48	61.15	48.72	50.76	57.38	46.65	39.19	42.87	36.74	47.48	81.4	7.41	419.3
Adaptive: 10% → 50% tokens														
SmartRes-Lite	33%	54.55	61.46	48.42	52.15	58.25	46.39	46.63	51.94	41.71	51.11	86.4	4.05	252.6
Adaptive: 10% → 100% tokens														
SmartRes-Pro	55%	**55.45**	**62.18**	**49.10**	**53.84**	**60.81**	**47.48**	**50.04**	**56.37**	**44.36**	**53.11**	**89.9**	6.58	365.8

learning rate 1×10^{-4} using a cosine schedule. For *SmartRes*, the router utilizes features from layer $l = 30$ and is trained with a learning rate of 5×10^{-4} and threshold $\tau = 0.5$. All experiments are conducted with 2 NVIDIA H100-96GB GPUs for training and one H100-96GB GPU for inference.

4.2 Main Experiments on Egocentric Benchmarks

Tab. 1 summarizes grounding performance and computational efficiency on Ego4D and EgoIntention across methods. As illustrated, *SmartRes* achieves the Pareto-optimal accuracy-efficiency trade-offs.

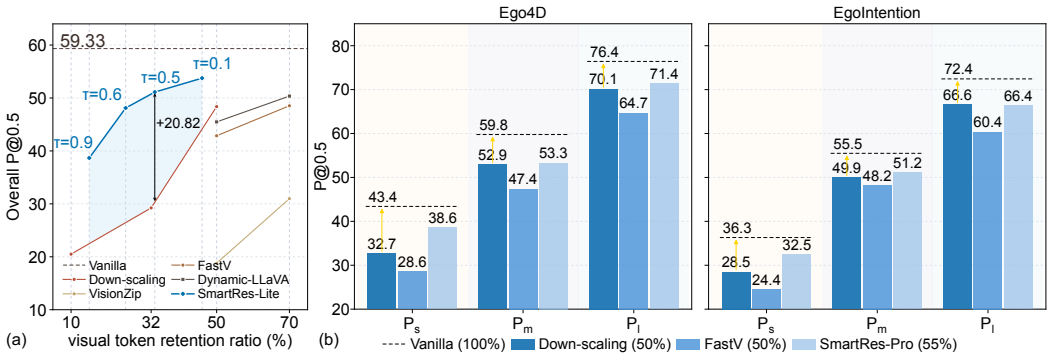

Figure 6: **(a) Adaptive compute allocation during inference.** Varying the routing threshold τ in *SmartRes-Lite* yields different token retention ratios, establishing an adaptive accuracy–efficiency frontier that consistently outperforms uniform down-scaling at comparable token budgets. **(b) Performance by object scale.** *SmartRes-Pro* better preserves small object performance by routing high-resolution computation to object-centric regions.

Grounding Performance. We achieve high grounding accuracy under strong compression. *SmartRes-Lite* retains 86.4% of full-resolution performance using only 33% of visual tokens on average, surpassing uniform down-scaling at a comparable budget by 34.1 percentage points in retention and improving Overall P@0.5 from 29.23 to 51.11. Fig. 6 (a) further shows that *SmartRes-Lite* consistently outperforms uniform down-scaling across different routing thresholds τ, forming an adaptive accuracy–efficiency frontier. *SmartRes-Pro* improves retention to 89.9% at a 55% token budget and achieves the best grounding accuracy among efficient methods. In contrast, token-reduction methods degrade notably under compression. At 70% tokens, ToMe and VisionZip retain only 42.5% and 54.5%, while FastV and Dyn-LLaVA retain 83.3% and 85.1% but require higher

Table 2: **Main results on referring expression comprehension (REC).** Vanilla denotes Qwen2.5-VL-3B-Instruct. We report P@0.5 on RefCOCO, RefCOCO+, and RefCOCOg. Overall is the arithmetic mean over the eight evaluation splits. Avg. denotes mean performance retention over the eight splits relative to Vanilla. **Bold** marks the best value among efficient methods.

Method	Ratio ↓	RefCOCO			RefCOCO+			RefCOCOg		Overall	Avg. ↑
		Val	TestA	TestB	Val	TestA	TestB	Val	Test	P@0.5	(%)
Vanilla	100%	91.19	92.88	86.85	86.15	90.02	80.41	87.96	88.12	87.95	100.0
Down-scaling	50%	66.92	68.74	63.85	63.22	66.62	59.12	64.55	65.22	64.78	73.7
	32%	57.86	56.06	52.42	54.66	54.33	48.53	55.81	53.19	54.11	61.5
	10%	41.47	36.61	38.56	39.18	35.48	35.70	40.00	34.73	37.72	42.9
FastV [10]	50%	61.02	62.71	59.25	55.40	58.62	51.72	58.21	58.42	58.17	66.1
VScan [28]	50%	72.32	**78.51**	64.95	66.70	**74.32**	55.52	67.31	66.72	68.29	77.5
Adaptive: 10% → 100% tokens											
SmartRes-Pro	42%	**74.19**	77.87	**70.34**	**67.68**	71.31	**61.97**	**71.72**	**71.98**	**70.88**	**80.5**

FLOPs and latency. At 50% tokens, the strongest comparison, Dyn-LLaVA, drops to 81.4% retention. These results support our observation that latent-space pruning can discard fine-grained spatial evidence needed for accurate localization. **Efficiency.** The uniform down-sampling baselines have the lowest compute but also suffer from severe performance degradation. SmartRes achieves a better trade-off, incurring negligible overhead over down-sampling while significantly outperforming token pruning. *SmartRes-Lite* reduces FLOPs by 64% and latency by 54% compared to the full-resolution Vanilla model. Crucially, it operates 1.66× faster than the 50%-token Dyn-LLaVA variant while delivering 5.0 percentage points higher retention.

4.3 Evaluation on Small Objects

Small objects are common in egocentric grounding and are especially sensitive to resolution loss. We therefore analyze performance by object scale in Fig. 6 (b). The results show that performance degradation under compression is scale-dependent. Uniform down-scaling to 50% reduces $\mathrm{P_s}$ by 10.7% on Ego4D and 7.8% on EgoIntention, while the corresponding drops for $\mathrm{P_l}$ are only 6.3% and 5.8%, respectively. This indicates that medium and large objects remain well-represented after down-sampling, whereas small objects are more sensitive to resolution loss. Token pruning methods further amplify this imbalance. In contrast, *SmartRes-Pro* better preserves small-object performance under a reduced token budget. At a 55% ratio, it retains 88.9% of the full-resolution $\mathrm{P_s}$ on Ego4D and 89.5% on EgoIntention, while maintaining competitive performance on medium and large objects. These results show that routing high-resolution computation to object-centric regions mitigates small-object resolution loss in egocentric scenes.

4.4 Generalization to Standard REC Benchmarks

We further evaluate SmartRes on standard referring expression comprehension (REC) benchmarks in Tab. 2. SmartRes generalizes well to general visual grounding. Averaged over the eight evaluation splits, SmartRes-Pro achieves 70.88 Overall P@0.5 and 80.5% performance retention while using only 42% of tokens on average. Our results also show that uniform down-scaling provides a competitive low-cost baseline on REC. At a 50% token ratio, it achieves 64.78 Overall P@0.5, which drops to 54.11 and 37.72 at 32% and 10%, respectively. *SmartRes-Pro* improves over Down-scaling-50% and VScan by 6.10 and 2.59 Overall P@0.5, respectively.

4.5 Qualitative Analysis

Fig. 7 shows that FastV discards fine-grained details, causing it to miss small targets like "detergent" in row 1. In contrast, SmartRes routes high-resolution computation to object-centric regions with a low-resolution context. Similarly, the token retention map derived from FastV (Fig. 7(b)) is divergent, which ignores "spoon", while ours focuses on object-aware regions. In addition to the accurate localization results, this further enhances the interpretability of *SmartRes*.

4.6 Ablation Studies

Component analysis. Tab. 3a ablates the main components of SmartRes. Compared with uniform down-sampling, random HR patch selection improves Overall P@0.5 by 4.68 percentage points,

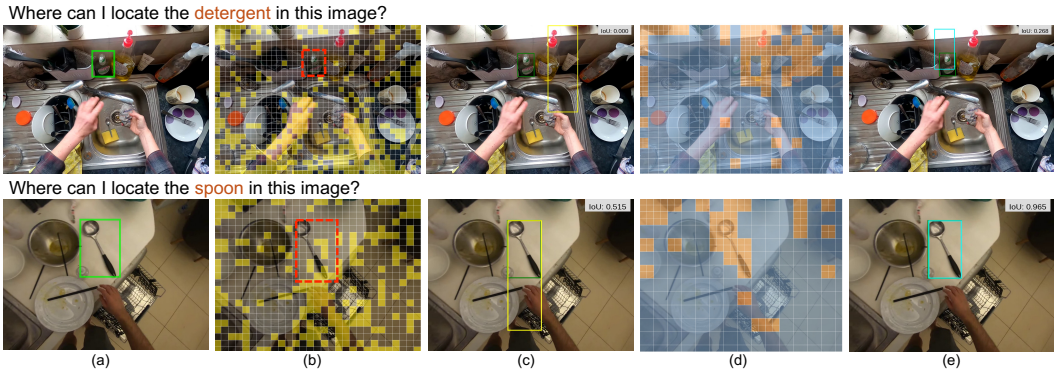

Figure 7: **Visualization of compared efficiency mechanisms.** (a) Input image with query. **FastV's** (b) pruning mask ( ) and (c) prediction. Relying on noisy latent proxies, the baseline often discards fine-grained details, missing small targets like the "detergent" in Row 1. **SmartRes'** (d) routing mask and (e) prediction. By dynamically routing high-resolution computation ( ) to object-centric areas against a low-res background ( ), our method yields precise localization.

Table 3: Ablation of key components on EgoIntention. Overall denotes the mean P@0.5 over the two EgoIntention splits. Let $\mathcal{G}$ be the ground-truth foreground token set and $\mathcal{R}$ be the set of tokens routed to HR refinement. FG-Recall denotes $|\mathcal{R} \cap \mathcal{G}|/|\mathcal{G}|$, measuring foreground coverage, while FG-Precision denotes $|\mathcal{R} \cap \mathcal{G}|/|\mathcal{R}|$.

Method	**Ratio**	**Overall** P@0.5	**FG-Recall** (%)	**FG-Precision** (%)
Uniform Down-sampling	32.14%	27.47	–	–
SmartRes-Random	33.00%	32.15	25.6	19.1
Attn2HR	32.96%	36.12	18.5	22.3
SmartRes w/o $\mathcal{L}_{\text{hinge}}$	32.94%	45.35	37.0	24.7
SmartRes (Full)	33.05%	**49.39**	54.0	31.1
Oracle (GT-guided)	18.97%	58.45	100.0	100.0

(a) Routing strategy and supervision.

Setting	**Value**	**Overall** P@0.5	**FG-Recall** (%)	**FLOPs** (T)
Layer ℓ	2nd	47.82	47.1	3.82
	15th	48.95	51.8	4.01
	30th	49.39	54.2	4.05
Threshold τ	0.3	**49.41**	**55.6**	4.82
	0.5	49.39	54.2	4.05
	0.7	46.20	42.5	**3.10**

(b) Routing layer and threshold.

showing that preserving local high-resolution details is already beneficial. Using the learned router raises Overall P@0.5 to 45.35, and adding the margin-regularized loss $\mathcal{L}_{\text{hinge}}$ further improves it to 49.39. This objective also boosts FG-Recall from 37.0% to 54.0% and FG-Precision from 24.7% to 31.1%, effectively mitigating foreground–background imbalance. The result of Oracle (GT-guided) (58.45) indicates remaining potential for precise patch selection.

Attn2HR. We compare against *Attn2HR*, a fixed attention-guided selection baseline detailed in Sec. D of the *Supplementary*. The results in Tab. 3a show that it improves over uniform down-sampling and random routing, raising Overall P@0.5 from 32.15% to 36.12%. Thus, visual self-attention provides useful saliency cues for high-resolution selection. However, it remains substantially below the learned router under a comparable token budget, indicating that fixed attention heuristics are insufficient for guiding patch selection and highlighting the need for learnable patch-level routing.

Router hyperparameters. Tab. 3b studies the routing layer ℓ and threshold τ. Using deeper features improves routing quality. Increasing ℓ from 2nd to 30th raises Overall P@0.5 from 47.82 to 49.39 and FG-Recall from 47.1% to 54.2%. Besides, the smaller τ=0.3 activates more object-aware regions, thus increasing the performance of Overall P@0.5 from 49.39 to 49.41 at the cost of efficiency. Conversely, a variant with a larger τ=0.7 improves efficiency while degrading performance.

5 Conclusion

We introduce SmartRes, a framework for efficient egocentric grounding through dynamic resolution routing in pixel space. A low-resolution view guides selective high-resolution processing, preserving local detail with compact global context. A margin-regularized objective improves routing under foreground–background imbalance. SmartRes-Lite retains 86.4% of full-resolution performance with 33% of visual tokens and runs 1.66× faster than the 50%-token Dyn-LLaVA baseline. Future work will extend resolution routing to egocentric video.

Acknowledgments

This research is supported by the Ministry of Education, Singapore, under its MOE Academic Research Fund Tier 2 (MOE-T2EP20125-0037). We would like to acknowledge that computational work involved in this research is partially supported by NUS IT's Research Computing group using grant number NUSREC-HPC-00001.

References

[1] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022.

[2] Hao Zhang, Hongyang Li, Feng Li, Tianhe Ren, Xueyan Zou, Shilong Liu, Shijia Huang, Jianfeng Gao, Lei Zhang, Chunyuan Li, et al. Llava-grounding: Grounded visual chat with large multimodal models. In *European Conference on Computer Vision*, pages 19–35. Springer, 2024.

[3] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 1(2):3, 2023.

[4] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 69–85. Springer, 2016.

[5] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. ReferItGame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798, 2014.

[6] Pengzhan Sun, Junbin Xiao, Tze Ho Elden Tse, Yicong Li, Arjun Akula, and Angela Yao. Visual intention grounding for egocentric assistants. *ICCV*, 2025.

[7] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision. *International Journal of Computer Vision*, 130(1):33–55, 2022.

[8] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5693–5703, 2019.

[9] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[10] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pages 19–35. Springer, 2024.

[11] Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*, 2024.

[12] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022.

[13] Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. VisionZip: Longer is better but not necessary in vision language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19792–19802, 2025.

[14] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. *arXiv preprint arXiv:2309.16588*, 2023.

[15] Wenxuan Huang, Zijie Zhai, Yunhang Shen, Shaosheng Cao, Fei Zhao, Xiangfeng Xu, Zheyu Ye, Yao Hu, and Shaohui Lin. Dynamic-LLaVA: Efficient multimodal large language models via dynamic vision-language context sparsification. *arXiv preprint arXiv:2412.00876*, 2024.

[16] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20, 2016.

[17] Varun K Nagaraja, Vlad I Morariu, and Larry S Davis. Modeling context between objects for referring expression understanding. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 792–807. Springer, 2016.

[18] Margaret Mitchell, Kees Van Deemter, and Ehud Reiter. Generating expressions that refer to visible objects. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1174–1184, 2013.

[19] Nicholas FitzGerald, Yoav Artzi, and Luke Zettlemoyer. Learning distributions over logical forms for referring expression generation. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1914–1925, 2013.

[20] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[21] Yongyi Su, Haojie Zhang, Shijie Li, Nanqing Liu, Jingyi Liao, Junyi Pan, Yuan Liu, Xiaofen Xing, Chong Sun, Chen Li, Nancy F. Chen, Shuicheng Yan, Xulei Yang, and Xun Xu. Patch-as-decodable-token: Towards unified multi-modal vision tasks in mllms. In *International Conference on Learning Representations (ICLR)*, 2026.

[22] Jiarui Zhang, Mahyar Khayatkhoei, Prateek Chhikara, and Filip Ilievski. Mllms know where to look: Training-free perception of small visual details with multimodal llms. *ICLR*, 2025.

[23] Shuhei Kurita, Naoki Katsura, and Eri Onami. RefEgo: Referring expression comprehension dataset from first-person perception of ego4d. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15214–15224, 2023.

[24] Vignesh Ramanathan, Anmol Kalia, Vladan Petrovic, Yi Wen, Baixue Zheng, Baishan Guo, Rui Wang, Aaron Marquez, Rama Kovvuri, Abhishek Kadian, et al. Paco: Parts and attributes of common objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7141–7151, 2023.

[25] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. LLaVA-NeXT: Improved reasoning, ocr, and world knowledge. *arXiv preprint*, 2024.

[26] Linli Yao, Long Xing, Yang Shi, Sida Li, Yuanxin Liu, Yuhao Dong, Yi-Fan Zhang, Lei Li, Qingxiu Dong, Xiaoyi Dong, et al. Towards efficient multimodal large language models: A survey on token compression. *Authorea Preprints*, 2026.

[27] Xin Zou, Di Lu, Yizhou Wang, Yibo Yan, Yuanhuiyi Lyu, Xu Zheng, Linfeng Zhang, and Xuming Hu. Don't just chase "highlighted tokens" in MLLMs: Revisiting visual holistic context retention. *arXiv preprint arXiv:2510.02912*, 2025.

[28] Ce Zhang, Kaixin Ma, Tianqing Fang, Wenhao Yu, Hongming Zhang, Zhisong Zhang, Haitao Mi, and Dong Yu. VScan: Rethinking visual token reduction for efficient large vision-language models. *Transactions on Machine Learning Research*, 2026.

[29] Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, et al. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *arXiv preprint arXiv:2410.17247*, 2024.

[30] Bizhe Bai, Jianjian Cao, Yadan Luo, and Tao Chen. Local information matters: Inference acceleration for grounded conversation generation models through adaptive local-aware token pruning. *arXiv preprint arXiv:2503.23959*, 2025. doi: 10.48550/arXiv.2503.23959.

[31] Wentong Li, Yuqian Yuan, Jian Liu, Dongqi Tang, Song Wang, Jie Qin, Jianke Zhu, and Lei Zhang. TokenPacker: Efficient visual projector for multimodal llm. *International Journal of Computer Vision*, pages 1–19, 2025.

[32] Runhui Huang, Xinpeng Ding, Chunwei Wang, Jianhua Han, Yulong Liu, Hengshuang Zhao, Hang Xu, Lu Hou, Wei Zhang, and Xiaodan Liang. HiRes-LLaVA: Restoring fragmentation input in high-resolution large vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29814–29824, 2025.

[33] Xubing Ye, Yukang Gan, Yixiao Ge, Xiao-Ping Zhang, and Yansong Tang. ATP-LLaVA: Adaptive token pruning for large vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[34] Mingyu Ouyang, Kevin Qinghong Lin, Mike Zheng Shou, and Hwee Tou Ng. FocusUI: Efficient UI grounding via position-preserving visual token selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

[35] Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal LLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13084–13094, 2024.

[36] Jiaxu Qian, Chendong Wang, Yifan Yang, Chaoyun Zhang, Huiqiang Jiang, Xufang Luo, Yu Kang, Qingwei Lin, Anlan Zhang, Shiqi Jiang, Ting Cao, Tianjun Mao, Suman Banerjee, Guyue Liu, Saravan Rajmohan, Dongmei Zhang, Yuqing Yang, Qi Zhang, and Lili Qiu. Zoomer: Adaptive image focus optimization for black-box mllm. *arXiv preprint arXiv:2505.00742*, 2025.

[37] Baifeng Shi, Boyi Li, Han Cai, Yao Lu, Sifei Liu, Marco Pavone, Jan Kautz, Song Han, Trevor Darrell, Pavlo Molchanov, and Hongxu Yin. Scaling vision pre-training to 4K resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9631–9640, June 2025.

[38] Yitong Jiang, Jinwei Gu, Tianfan Xue, Ka Chun Cheung, Pavlo Molchanov, Hongxu Yin, and Sifei Liu. Token-efficient vlm: High-resolution image understanding via dynamic region proposal. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 24147–24158, 2025.

[39] Senqiao Yang, Junyi Li, Xin Lai, Jinming Wu, Wei Li, Zejun Ma, Bei Yu, Hengshuang Zhao, and Jiaya Jia. VisionThink: Smart and efficient vision language model via reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

[40] Zichuan Lin, Yicheng Liu, Yang Yang, Lvfang Tao, and Deheng Ye. AdaptVision: Efficient vision-language models via adaptive visual acquisition. *arXiv preprint arXiv:2512.03794*, 2025.

[41] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023.

[42] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[43] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

[44] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-Net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571. IEEE, 2016. doi: 10.1109/3DV.2016.79.

[45] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. doi: 10.1109/ICCV.2017.324.

[46] Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, and Ali Gholipour. Tversky loss function for image segmentation using 3d fully convolutional deep networks. In Qian Wang, Yinghuan Shi, Heung-Il Suk, and Kenji Suzuki, editors, *Machine Learning in Medical Imaging*, volume 10541 of *Lecture Notes in Computer Science*, pages 379–387, Cham, 2017. Springer. doi: 10.1007/978-3-319-67389-9_44.

[47] Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

[48] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9268–9277, 2019.

A Image Preparation and Token Correspondence

Given an input image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ and target token ratios r_{LR} and r_{HR}, we first compute the target spatial resolution for each branch. For branch $b \in \{\mathrm{LR}, \mathrm{HR}\}$ with ratio r_b, we set

$$\widetilde{H}_b = H\sqrt{r_b}, \qquad \widetilde{W}_b = W\sqrt{r_b}. \tag{10}$$

The isotropic scaling preserves the original aspect ratio while approximately matching the desired token ratio. To satisfy the patching and spatial-merging constraints of the vision encoder, we snap each dimension to a multiple of the effective visual-token stride $F = PM$:

$$H'_b = \mathrm{snap}_F(\widetilde{H}_b), \qquad W'_b = \mathrm{snap}_F(\widetilde{W}_b), \qquad b \in \{\mathrm{LR}, \mathrm{HR}\}. \tag{11}$$

Here,

$$\mathrm{snap}_F(s) = F \cdot \max\left(1, \left\lfloor \frac{s}{F} + \frac{1}{2} \right\rfloor\right) \tag{12}$$

rounds s to the nearest positive multiple of F. For the Qwen2.5-VL [9] example, the patch size is $P = 14$ and the spatial merge size is $M = 2$, so $F = 28$. We resize $\mathbf{I}$ to $(H'_{\mathrm{LR}}, W'_{\mathrm{LR}})$ and $(H'_{\mathrm{HR}}, W'_{\mathrm{HR}})$ to obtain $\mathbf{I}_{\mathrm{LR}}$ and $\mathbf{I}_{\mathrm{HR}}$, respectively. The snapped dimensions induce routing grids:

$$G_b^h = \frac{H'_b}{F}, \qquad G_b^w = \frac{W'_b}{F}, \qquad N_b = G_b^h G_b^w, \tag{13}$$

where G_b^h and G_b^w denote the height and width of the routing grid for branch b, and N_b is the corresponding feature length used by the router. Because of snapping, the realized token ratio may differ slightly from the target ratio. The LR and HR routing grids are aligned by a deterministic coordinate mapping. Each HR location at grid coordinate $(y_{\mathrm{HR}}, x_{\mathrm{HR}})$ is assigned to a unique LR parent:

$$\phi(y_{\mathrm{HR}}, x_{\mathrm{HR}}) = \left(\left\lfloor \frac{y_{\mathrm{HR}} G_{\mathrm{LR}}^h}{G_{\mathrm{HR}}^h} \right\rfloor, \left\lfloor \frac{x_{\mathrm{HR}} G_{\mathrm{LR}}^w}{G_{\mathrm{HR}}^w} \right\rfloor \right). \tag{14}$$

Here,

$$\phi : \{0, \ldots, G_{\mathrm{HR}}^h - 1\} \times \{0, \ldots, G_{\mathrm{HR}}^w - 1\} \to \{0, \ldots, G_{\mathrm{LR}}^h - 1\} \times \{0, \ldots, G_{\mathrm{LR}}^w - 1\}. \tag{15}$$

This mapping provides the HR-to-LR correspondence used to upsample the routing mask and select HR patches.

B Dataset Statistics

Table 4: **Dataset statistics for egocentric grounding and standard REC benchmarks.** RefCOCO family denotes the combined statistics of RefCOCO, RefCOCO+, and RefCOCOg. Avg. Res. reports the mean image resolution.

Metric	Egocentric Grounding		Standard REC
	Ego4D	**EgoIntention**	**RefCOCO family**
# Images	26.4K	35.6K	28.2K
# Boxes	58.4K	51.1K	321.3K
Avg. Res.	1825×1270	1844×1297	554×486
Object scale distribution			
Small	17.07%	10.07%	0.00%
Medium	54.95%	44.92%	3.31%
Large	27.99%	45.01%	96.69%

Tab. 4 highlights the resolution and object-scale differences between egocentric grounding benchmarks and standard referring expression comprehension (REC) datasets. Egocentric datasets have substantially higher native resolutions, with average image areas more than $8\times$ larger than those of the RefCOCO family. They also exhibit a markedly different object-scale distribution. Small and medium objects account for 72.02% of instances in Ego4D and 54.99% in EgoIntention, whereas the RefCOCO family is dominated by large objects. These statistics motivate resolution-aware processing for egocentric grounding, where many targets occupy limited spatial extent despite high-resolution inputs.

C Gradient Analysis of Imbalance-Aware Routing Losses

We compare the proposed margin hinge regularizer in Eq. (7) with imbalance-aware objectives for binary routing: vanilla BCE, focal loss [45], and soft Dice/Tversky losses [44, 46]. Let $N = N_{\mathrm{LR}}$, z_i be the routing logit for LR token i, $p_i = \sigma(z_i)$, and $\hat{M}_i \in \{0, 1\}$ be the routing label. We denote the foreground and background token sets by $\mathcal{P} = \{i : \hat{M}_i = 1\}$ and $\mathcal{N} = \{i : \hat{M}_i = 0\}$, with $|\mathcal{P}| \ll |\mathcal{N}|$ in egocentric grounding. For each loss, we inspect the logit gradient $\partial\mathcal{L}/\partial z_i$ and the class-level gradient masses

$$G_{\mathcal{P}} = \sum_{i\in\mathcal{P}} \left|\frac{\partial\mathcal{L}}{\partial z_i}\right|, \qquad G_{\mathcal{N}} = \sum_{i\in\mathcal{N}} \left|\frac{\partial\mathcal{L}}{\partial z_i}\right|. \tag{16}$$

These quantities measure the aggregate update pressure applied to foreground and background tokens.

BCE. The vanilla BCE loss is

$$\mathcal{L}_{\mathrm{BCE}} = -\frac{1}{N}\sum_i \left[\hat{M}_i \log p_i + (1-\hat{M}_i)\log(1-p_i)\right], \tag{17}$$

with gradient

$$\frac{\partial\mathcal{L}_{\mathrm{BCE}}}{\partial z_i} = \frac{1}{N}(p_i - \hat{M}_i). \tag{18}$$

Thus,

$$G_{\mathcal{P}} = \frac{1}{N}\sum_{i\in\mathcal{P}}(1-p_i), \qquad G_{\mathcal{N}} = \frac{1}{N}\sum_{i\in\mathcal{N}} p_i. \tag{19}$$

BCE has no class-level normalization. When the average foreground and background errors are comparable, the ratio $G_{\mathcal{P}}/G_{\mathcal{N}}$ scales with $|\mathcal{P}|/|\mathcal{N}|$, so the aggregate update can be dominated by the background tokens.

Class-weighted BCE. A stronger BCE baseline is class-weighted BCE, a standard strategy for imbalanced classification [47, 48], which reweights foreground and background tokens:

$$\mathcal{L}_{\mathrm{wBCE}} = -\frac{1}{N}\sum_i \left[w_{\mathcal{P}}\hat{M}_i \log p_i + w_{\mathcal{N}}(1-\hat{M}_i)\log(1-p_i)\right]. \tag{20}$$

Its logit gradient is

$$\frac{\partial\mathcal{L}_{\mathrm{wBCE}}}{\partial z_i} = \begin{cases} -\dfrac{w_{\mathcal{P}}}{N}(1-p_i), & i\in\mathcal{P}, \\ +\dfrac{w_{\mathcal{N}}}{N}p_i, & i\in\mathcal{N}. \end{cases} \tag{21}$$

We use inverse-frequency class balancing, $w_{\mathcal{P}} = N/(2|\mathcal{P}|)$ and $w_{\mathcal{N}} = N/(2|\mathcal{N}|)$, yielding

$$G_{\mathcal{P}} = \frac{1}{2|\mathcal{P}|}\sum_{i\in\mathcal{P}}(1-p_i), \qquad G_{\mathcal{N}} = \frac{1}{2|\mathcal{N}|}\sum_{i\in\mathcal{N}} p_i. \tag{22}$$

Thus, class-weighted BCE is a stronger token-wise baseline than vanilla BCE because it statically normalizes the foreground/background token counts. However, its aggregate update still depends on the current prediction probabilities and does not explicitly enforce a margin between foreground and background logits. In contrast, the proposed hinge term directly regularizes class-level logit means and provides balanced aggregate gradient mass when the margin is active.

Focal loss. The standard binary focal loss [45] is:

$$\mathcal{L}_{\mathrm{focal}} = -\frac{1}{N}\sum_i \left[\alpha\,\hat{M}_i(1-p_i)^{\gamma}\log p_i + (1-\alpha)(1-\hat{M}_i)p_i^{\gamma}\log(1-p_i)\right], \tag{23}$$

where $\alpha \in (0, 1)$ is a static foreground weighting coefficient and γ controls the strength of hard-example modulation. Its logit gradient is

$$\frac{\partial\mathcal{L}_{\mathrm{focal}}}{\partial z_i} = \begin{cases} -\dfrac{\alpha}{N}(1-p_i)^{\gamma}\left[(1-p_i) - \gamma p_i \log p_i\right], & i\in\mathcal{P}, \\ +\dfrac{1-\alpha}{N}p_i^{\gamma}\left[p_i - \gamma(1-p_i)\log(1-p_i)\right], & i\in\mathcal{N}. \end{cases} \tag{24}$$

The α term provides static foreground–background reweighting, and the focal factors $(1-p_i)^\gamma$ and p_i^γ suppress easy tokens. However, focal loss is still applied independently to each token and does not impose sample-wise class-level normalization. Its aggregate class-level gradient masses remain

$$G_{\mathcal{P}} = \frac{\alpha}{N}\sum_{i\in\mathcal{P}}(1-p_i)^\gamma\left[(1-p_i)-\gamma p_i\log p_i\right], \tag{25}$$

$$G_{\mathcal{N}} = \frac{1-\alpha}{N}\sum_{i\in\mathcal{N}}p_i^\gamma\left[p_i-\gamma(1-p_i)\log(1-p_i)\right]. \tag{26}$$

Thus, the ratio $G_{\mathcal{P}}/G_{\mathcal{N}}$ still depends on the class sizes, current prediction errors, and the fixed choice of α. A constant α can reduce imbalance on average, but it cannot guarantee balanced foreground/background update pressure across images or training stages.

Soft Dice and Tversky. The soft Dice loss [44] is defined as

$$\mathcal{L}_{\text{Dice}} = 1-\frac{2A}{B}, \qquad A=\sum_i p_i\hat{M}_i, \qquad B=\sum_i p_i+\sum_i \hat{M}_i. \tag{27}$$

Here, $p_i=\sigma(z_i)$ is the predicted routing probability, $\hat{M}_i\in\{0,1\}$ is the rasterized foreground label, A is the soft foreground overlap, and B is the Dice normalization term. Its gradient with respect to the routing logit z_i is

$$\frac{\partial\mathcal{L}_{\text{Dice}}}{\partial z_i} = -\frac{2}{B^2}\left(\hat{M}_iB-A\right)p_i(1-p_i) = \begin{cases} -\dfrac{2(B-A)}{B^2}p_i(1-p_i), & i\in\mathcal{P}, \\ +\dfrac{2A}{B^2}p_i(1-p_i), & i\in\mathcal{N}, \end{cases} \tag{28}$$

where $\mathcal{P}=\{i\mid\hat{M}_i=1\}$ and $\mathcal{N}=\{i\mid\hat{M}_i=0\}$ denote foreground and background token sets, respectively. This reveals two mechanisms that can weaken routing supervision. First, the gradient magnitude is coupled to the global overlap statistics A and B, which can be small or unstable when the foreground is sparse. Second, every token is multiplied by the sigmoid factor $p_i(1-p_i)$, so saturated logits receive little correction. For example, with $|\mathcal{P}|=12$, $|\mathcal{N}|=244$, and uniform $p_i=0.1$, we have $A=1.2$ and $B=37.6$. Eq. (28) then gives

$$G_{\mathcal{P}}=\sum_{i\in\mathcal{P}}\left|\frac{\partial\mathcal{L}_{\text{Dice}}}{\partial z_i}\right|\approx 0.056, \qquad G_{\mathcal{N}}=\sum_{i\in\mathcal{N}}\left|\frac{\partial\mathcal{L}_{\text{Dice}}}{\partial z_i}\right|\approx 0.037, \tag{29}$$

which is more than an order of magnitude smaller than the unit class-level gradient mass provided by the hinge loss below. The Tversky loss [46] generalizes Dice by reweighting false positives and false negatives:

$$\mathcal{L}_{\text{Tversky}} = 1-\frac{A}{A+\alpha_T\sum_i p_i(1-\hat{M}_i)+\beta_T\sum_i(1-p_i)\hat{M}_i}, \tag{30}$$

where α_T and β_T control the penalties on false positives and false negatives, respectively. Although this reweighting changes the relative cost of the two error types, the loss still inherits the global-denominator coupling and the $p_i(1-p_i)$ saturation factor.

Margin hinge. Our margin hinge regularizer operates directly on class-level logit means:

$$\mu_{\mathcal{P}}=\frac{1}{|\mathcal{P}|}\sum_{i\in\mathcal{P}}z_i, \qquad \mu_{\mathcal{N}}=\frac{1}{|\mathcal{N}|}\sum_{i\in\mathcal{N}}z_i, \tag{31}$$

$$\mathcal{L}_{\text{hinge}}=\left[m-(\mu_{\mathcal{P}}-\mu_{\mathcal{N}})\right]_+. \tag{32}$$

When the margin is violated, i.e., $\mu_{\mathcal{P}}-\mu_{\mathcal{N}}<m$, its gradient is

$$\frac{\partial\mathcal{L}_{\text{hinge}}}{\partial z_i} = \begin{cases} -1/|\mathcal{P}|, & i\in\mathcal{P}, \\ +1/|\mathcal{N}|, & i\in\mathcal{N}, \end{cases} \tag{33}$$

and is zero otherwise. Therefore, when active,

$$G_{\mathcal{P}}=G_{\mathcal{N}}=1, \tag{34}$$

independent of class size, current probability values, and training stage. During optimization, the hinge term pushes foreground logits upward and background logits downward with equal aggregate magnitude, directly enlarging the foreground–background logit gap. For numerical stability, the implementation computes the class means with denominators $|\mathcal{P}| + \epsilon$ and $|\mathcal{N}| + \epsilon$, where $\epsilon = 10^{-6}$. If either set is empty, its summation is zero and no gradient is assigned to that empty set. In our grounding benchmarks, each training sample contains an annotated target box, so $\mathcal{P}$ is normally non-empty after rasterization; this fallback is used only as a safeguard.

Table 5: **Routing supervision under different imbalance losses.** Ratio denotes average HR-token retention. Let $\mathcal{G}$ be the ground-truth foreground token set and $\mathcal{R}$ be the set of tokens routed to HR refinement. FG-Recall denotes $|\mathcal{R} \cap \mathcal{G}|/|\mathcal{G}|$, measuring foreground coverage, while FG-Precision denotes $|\mathcal{R} \cap \mathcal{G}|/|\mathcal{R}|$.

Loss	Ratio	Overall ↑	FG-Recall ↑	FG-Precision ↑
Dice [44]	33.25%	46.85	0.42	0.240
Focal [45] ($\alpha = 0.25, \gamma = 2$)	32.86%	47.82	0.45	0.260
Tversky [46]	33.38%	48.15	0.49	0.279
Margin hinge (ours)	33.05%	**49.39**	**0.54**	**0.311**

Empirical comparison. We retrain SmartRes-Lite with each imbalance objective while keeping all other components fixed. Tab. 5 reports the average HR-token retention ratio, overall grounding score, foreground recall, and foreground precision. All methods use a comparable token budget of roughly 33% HR-token retention. The margin hinge achieves the best routing quality and grounding score: compared with the strongest non-hinge baseline, Tversky, it improves Overall by $+1.24$, FG-Recall by $+0.05$, and FG-Precision by $+0.032$.

D Attn2HR

We introduce a simple attention-guided high-resolution selection baseline, denoted as *Attn2HR*, to test whether a training-free attention heuristic is sufficient for selecting regions that require high-resolution processing. Given the low-resolution input, *Attn2HR* runs the vision encoder and extracts visual self-attention from a late visual layer. In our experiments, we use layer $\ell = 30$. Let $\mathbf{A}^{(\ell,h)} \in \mathbb{R}^{N_{\mathrm{LR}} \times N_{\mathrm{LR}}}$ denote the visual self-attention matrix of head h at layer ℓ, where N_{LR} is the number of valid LR visual tokens. We compute the attention saliency score of LR token i by aggregating the attention it receives from all visual tokens and heads:

$$s_i^{\mathrm{attn}} = \frac{1}{H N_{\mathrm{LR}}} \sum_{h=1}^{H} \sum_{j=1}^{N_{\mathrm{LR}}} \mathbf{A}_{j,i}^{(\ell,h)}, \tag{35}$$

where H is the number of attention heads and $\mathbf{A}_{j,i}^{(\ell,h)}$ denotes the attention from query token j to key token i. We then select the indices of the top-K LR parent tokens according to s_i^{attn}:

$$\mathcal{S}_K = \mathrm{TopKIdx}\left(\{s_i^{\mathrm{attn}}\}_{i=1}^{N_{\mathrm{LR}}}, K\right). \tag{36}$$

Each selected LR parent token is mapped to its corresponding HR child patches using the parent-child correspondence defined in Eq. 14. The selected HR patches are then processed by the same HR re-encoding module and assembled using the same order-preserving sequence construction as Eq. 5. By keeping the HR re-encoding module and sequence assembly identical to SmartRes, *Attn2HR* isolates the effect of the selection policy: it replaces learned task-supervised routing with a fixed attention-based heuristic. In Tab. 3a, we set $K = 25$, corresponding to approximately 25.5% of LR parent tokens and yielding a final token ratio of 32.96%.

E Qualitative Analysis

Additional examples of SmartRes routing are provided in Fig. 8. As shown, SmartRes consistently allocates HR computation to object-centric regions while keeping the surrounding scene as LR context. This is especially beneficial for small objects, such as the *pliers*, *knife*, and *bottle*, where

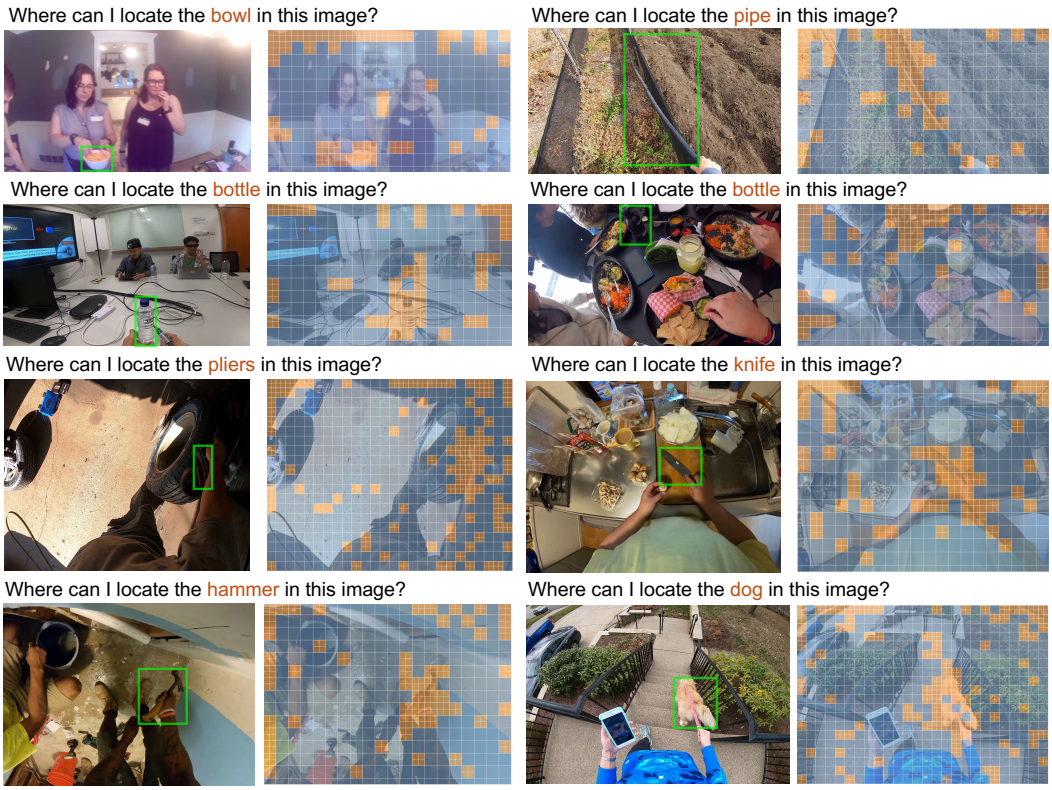

Figure 8: Visualizations of SmartRes' routing mask. SmartRes dynamically routes high-resolution computation (■) to object-centric areas against a low-resolution background (■).

fine-grained visual evidence is easily lost under uniform down-sampling or latent token pruning. The routed regions also often cover nearby hands, tools, or manipulated objects, which provide useful spatial and interaction cues for egocentric grounding. These examples further demonstrate that SmartRes preserves global context while selectively enhancing fine-grained details needed for accurate localization.

F Routing Quality Across Object Scales

We evaluate SmartRes-Lite on 1,000 Ego4D instances to examine whether high-resolution routing covers small targets. Table 6 reports ground-truth (GT) area and routed foreground area as percentages of the image area. Routed foreground is the portion of the GT box selected for high-resolution processing. FG recall measures the fraction of foreground tokens routed to high resolution, whereas background (BG) activation measures the fraction of background tokens routed to high resolution.

Table 6: **Routing quality by object scale on 1,000 Ego4D instances.** All values are percentages. GT area and routed foreground area are normalized by image area; FG recall and BG activation are normalized by foreground and background token counts, respectively.

Object scale	GT area	Routed FG area	FG recall	BG activation
Small	0.47	0.28	59.96	12.59
Medium	3.11	2.01	64.65	28.81
Large	35.21	21.75	61.78	31.45

For small targets, whose average GT area is only 0.47% of the image, the router retains 59.96% of foreground tokens while activating 12.59% of background tokens. Foreground coverage remains comparable across the three scale groups, while background activation increases with object scale. These measurements complement the grounding results by directly quantifying the high-resolution

coverage of target regions. They also show that routing is imperfect: some target evidence remains at low resolution, and some background is processed at high resolution. The routing threshold controls this coverage–cost trade-off.

G Latency Analysis

Latency across object scales. We profile 50 small, 50 medium, and 50 large EgoIntention instances. Table 7 reports visual sequence length, token ratio, and inference latency for SmartRes-Lite, alongside FastV at a 50% token budget and the full-resolution backbone. SmartRes-Lite is faster in each group, although the advantage narrows for large objects: its latency increases from 391.9 ms for small objects to 512.8 ms for large objects, compared with 541.6 ms for FastV and 576.2 ms for the full-resolution backbone in the large-object group. Thus, the efficiency gain depends on the amount of high-resolution computation activated for each image.

Table 7: **Latency by target scale on EgoIntention.** Each group contains 50 instances. Sequence length and token ratio are reported for SmartRes-Lite; latency is in milliseconds.

Scale	Visual seq. length	Token ratio	SmartRes-Lite	FastV-50%	Vanilla
Small	305.3	27.9%	391.9	504.6	536.8
Medium	299.5	32.8%	429.1	491.9	523.3
Large	321.1	58.3%	512.8	541.6	576.2

H Cross-Dataset Router Transfer

We evaluate the transferability of the learned routing policy from Ego4D to EgoIntention using SmartRes-Lite. Table 8 compares an EgoIntention-trained router with an Ego4D-trained router at a reported token ratio of 33%. This comparison concerns the source of the routing policy; it is not an evaluation of zero-shot transfer of the entire grounding model. For the transferred condition, we replace only the router with the Ego4D-trained router and keep its parameters frozen. The vision encoder and language model, including their LoRA parameters, retain the EgoIntention configuration used for the in-domain model.

Table 8: **Cross-dataset router transfer to EgoIntention.** Scores are P@0.5 (%). EgoInt. mean is the arithmetic mean of the Context and Uncommon splits.

Model	Router source	Token ratio	Context	Uncommon	EgoInt. mean
SmartRes-Lite	EgoIntention	33%	52.15	46.63	49.39
SmartRes-Lite	Ego4D	33%	47.48	43.60	45.54

The transferred router is 4.67 and 3.03 percentage points below the in-domain router on Context and Uncommon, respectively. These results indicate that the routing policy retains useful object-centric cues across the two egocentric datasets, while the remaining gap shows that the source-domain training distribution still affects downstream grounding performance.

I Limitations

The current formulation of SmartRes focuses on image-level grounding and does not explicitly model temporal dependencies in egocentric video. Extending resolution routing to video offers opportunities to exploit temporal coherence and reuse visual features across frames. Further investigation of language-conditioned routing could enable query-specific allocation of high-resolution computation, while hardware-aware optimization could align routing policies with device-specific latency constraints. These directions may support the extension of SmartRes to streaming perception and embodied applications.

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 Section 1 state the SmartRes routing idea and empirical claims; Section 4 reports results aligned with those claims.

2. **Limitations**

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

   Answer: [Yes]

   Justification: Section I notes the current image-level scope and outlines future directions in temporal routing, query-adaptive resolution allocation, and hardware-aware computation.

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: [N/A]

   Justification: The paper does not present formal theorems or proofs; contributions are algorithmic and empirical.

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: Section 4 and the supplementary appendix specify datasets, metrics, training hyperparameters, and baseline reproduction; implementation details outline how to instantiate the routing module.

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: [Yes]

   Justification: Code, configuration files, and instructions for installation, training, and evaluation are publicly available at `https://github.com/HuixinSun/SmartRes`. The experiments use the public benchmarks cited in the paper; access to the original data remains subject to the respective providers' terms.

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) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4 reports optimizer, schedule, LoRA configuration, routing hyperparameters, and hardware; the supplement adds baseline and training-free configuration details where space is limited.

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: Main tables report single deterministic runs on published splits without confidence intervals or multi-seed aggregates, owing to the cost of end-to-end VLM training and evaluation.

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: [Yes]

   Justification: Section 4 states NVIDIA H100 GPUs for training and inference; exploratory hyperparameter and ablation runs required additional but comparable compute and are described qualitatively.

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: We reviewed the NeurIPS Code of Ethics and the Main Track policies; experiments rely on public artifacts used according to their documented research terms.

10. **Broader impacts**

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

    Answer: [No]

    Justification: The paper discusses potential applications in assistive technologies and robotics. It does not provide a dedicated discussion of both positive and negative societal impacts.

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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We do not newly release a scraped web-scale dataset or standalone high-risk generative checkpoint; we build on existing public benchmarks and foundation models under their access rules.

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: The paper cites the public datasets, pretrained MLLM backbones, and baseline methods used in the experiments.

13. **New assets**

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

    Answer: [Yes]

    Justification: The public SmartRes repository provides implementation code, configuration files, a SmartRes-Lite checkpoint, and documentation for installation, training, and evaluation: `https://github.com/HuixinSun/SmartRes`.

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: The project does not run new crowdsourcing or interactive human-subjects studies beyond using existing public benchmarks.

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: No IRB-supervised human-subjects protocol is introduced; experiments use public vision–language benchmarks.

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 rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The study uses a standard instruction-tuned MLLM backbone as an existing component; we do not introduce a novel LLM training recipe as a scientific contribution. Any editorial assistance did not affect methodology or results.