ATTNSEG: ATTENTION AS THE INTERFACE BETWEEN LANGUAGE MODELS AND SEGMENTERS

Anonymous authors
Paper under double-blind review

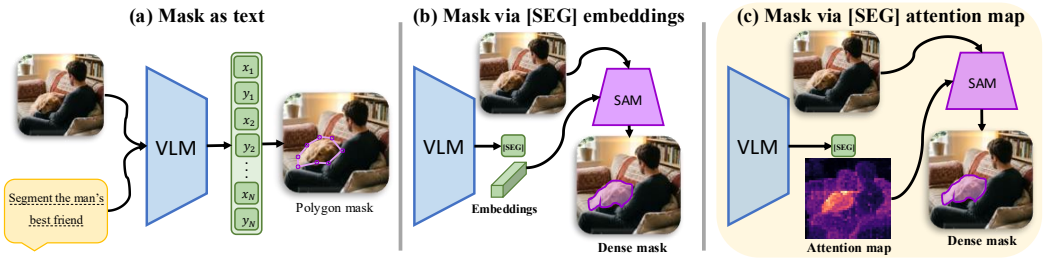


Figure 1: Three ways a vision language model can hand a segmenter its spatial intent. (a) The mask is serialized into the token stream. (b) The dominant design: one d dimensional `[SEG]` embedding, carrying no spatial index, prompts SAM. (c) Ours: the `[SEG]` token's attention is already a distribution on the patch grid, so we forward it instead of re-deriving it.

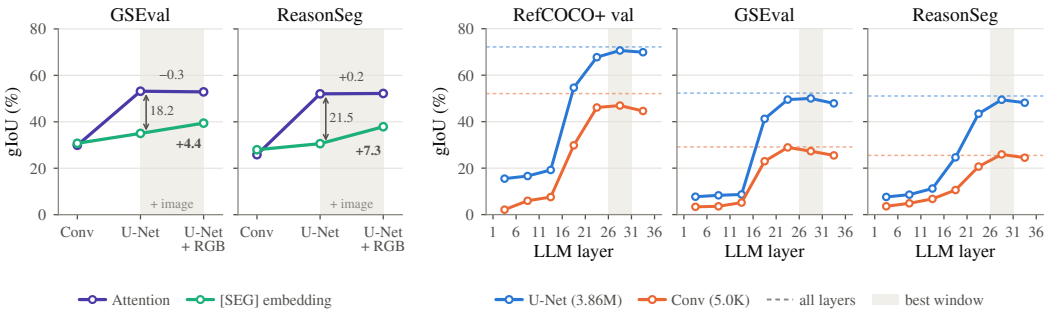
ABSTRACT

Vision-language models for segmentation must communicate *where* a referred object lies to a pixel decoder. Existing approaches do so by either serializing the mask into the token stream or compressing spatial information into the hidden state of a single `[SEG]` token. We argue that both overlook a spatial representation the model has already computed: the `[SEG]` token's attention over image tokens, which is natively defined on the image patch grid and can directly encode the referred region. Probing a frozen segmentation VLM shows that attention contains substantially more directly readable localization information than the `[SEG]` hidden state at matched capacity. This signal captures object shape rather than merely pointing to the referent and is concentrated in a small subset of late attention heads, while standard uniform aggregation obscures much of this spatial structure. Motivated by these findings, we introduce ATTNSEG, an attention-based interface that makes the VLM the primary localizer and leaves the segmentation decoder to refine its prediction. ATTNSEG learns to aggregate the `[SEG]` token's attention maps into a dense spatial prompt and directly supervises it with an attention-guidance objective. A two-stage training strategy further enforces this division of labor by first learning localization in the VLM and then training the decoder for mask refinement. Applied to Sa2VA, ATTNSEG improves ReasonSeg by $+6.4$ gIoU and GS-Eval by $+10.2$ at 4B, and by $+8.8$ and $+11.3$ at 14B, respectively. Gains reach $+18$ on object parts, multiple objects, and stuff regions, where identifying *where* to segment rather than refining boundaries is the primary challenge.

1 INTRODUCTION

Segmenting an object from a language description requires answering two questions: *which* object is being referred to, and *where* exactly its boundary lies. Only one of them is still hard. Promptable segmenters such as SAM (Kirillov et al., 2023), HQ-SAM (Ke et al., 2023) and SAM3 (Carion et al., 2026) have largely closed the boundary problem: given a prompt that points at the right object, a modern decoder returns an excellent mask. Deciding which pixels a phrase such as *the man's best friend* refers to takes language understanding, and that is the vision language model's job rather than the decoder's. This motivates the hypothesis at the centre of our work: **the vision language model should be the load-bearing localizer, and the decoder's role should be reduced to refinement.**

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Which space, and what it carries **(b)** Where in the stack it lives

Figure 2: Probing a frozen Sa2VA-4B. (a) At matched U-Net capacity the attention beats the `[SEG]` embedding by 18.2 and 21.5 gIoU, and adding the RGB image helps the embedding (+4.4, +7.3) but not the attention (−0.3, +0.2). (b) Restricting the probe to a five layer window: early layers carry almost nothing, and the whole stack (dashed) is matched by one late window.

Whether that happens is decided by the **interface**, the representation through which the model hands its spatial intent to whatever produces pixels. Two dominate the field, summarized in Figure 1(a,b).

The first, Figure 1(a), turns the mask into a string, playing to the model's strength rather than around it: segmentation becomes ordinary next token prediction over polygon vertices (Liu et al., 2023; Wang et al., 2023), learned codewords (Beyer et al., 2024; Wang et al., 2025b), or a label for every patch of a coarse grid (Lan et al., 2025b). Geometry must then survive a token bottleneck: the output grid is the language model's own coarse one, a cross entropy over coordinate tokens is blind to how far off a coordinate is, and one mis-predicted early vertex distorts everything the autoregressive string emits after it. The accuracy gap against decoder based systems is wide (Section 2).

The second family, Figure 1(b), fixes precisely that by leaving text space altogether. LISA (Lai et al., 2024) and the many systems that followed (Rasheed et al., 2024; Zhang et al., 2024a; Yuan et al., 2025) add a `[SEG]` token to the vocabulary and pass its final hidden state to SAM as a prompt. The mask never enters the token stream, so error compounding disappears and the boundary is drawn at full resolution by a model built for it. But localization now lives in the model's semantic space and is trained end to end, so the only gradient that ever shapes the `[SEG]` vector is the segmentation loss computed *after* the decoder has produced its mask. The feedback is late and indirect: nothing states what that vector should mean spatially, only that the decoder should end up correct. Nor does anything oblige the vision language model to carry the localization, since the decoder can recover much of the answer by itself, so the division of labour drifts toward it, the opposite of what we argued for above. Compounding both, **the entire spatial message is a single d dimensional vector with no spatial index**.

Where does the localization actually live? Rather than assert a better channel, we look for one inside a model that already works. We freeze Sa2VA (Yuan et al., 2025), among the strongest open source grounding vision language models, and attach a small probe that must produce the mask from a single internal signal. Two are candidates: the `[SEG]` token's hidden state, which is what the dominant interface transmits, and its attention over the image tokens, which is already a distribution on the patch grid. Read by the same architecture at matched capacity, the attention recovers 18.2 and 21.5 more gIoU than the embedding on two out of domain benchmarks (Figure 2a). This is intuitive in hindsight: an attention row is already a two dimensional map, whereas a hidden state must express a region as an unordered vector. To our knowledge the two channels have not been compared head to head inside the same frozen model, and the answer is our first motive.

From observation to hypothesis. If the attention is the stronger representation, the right response is not to read it passively but to train it. An explicit objective that pushes the attention to encode the referent more sharply, and to encode it so that it is easily decodable, should *reinforce* that capability rather than merely reveal it. We write this objective $\mathcal{L}_{\text{att}}$; where the `[SEG]` family applies its only pressure late and through the decoder, $\mathcal{L}_{\text{att}}$ acts on the map directly and from the first step.

Making the language model do the work. Testing that hypothesis means keeping the two contributions apart, so we train in two stages. In **stage one** we LoRA finetune the vision language model *alone* under $\mathcal{L}_{\text{att}}$; the decoder is never constructed and contributes no gradient, so every bit of pressure

lands on the attention. In **stage two** we freeze the vision language model, including the aggregation that forms the map, and train only the decoder, for which the map is an input rather than a target. The separation is not cosmetic: the decoder's gradient into the map is roughly 49 times the size of the attention loss's, so under joint training the map reaching the decoder is mostly the decoder's own doing. The result improves the base model at both scales, by $+6.4$ gIoU on ReasonSeg and $+10.2$ on GS-Eval at 4B and by $+8.8$ and $+11.3$ at 14B, concentrated exactly where the hypothesis predicts: object parts, multiple objects and stuff regions gain between 13 and 18 gIoU, while single salient objects, which the decoder can already find on its own, gain less than 1.5.

2 Related Work

We organize the literature by interface, since that is the axis along which we move; Appendix A gives the full family by family review.

Interface I: the mask as text. A mask can be serialized as polygon vertices (Liu et al., 2023; Wang et al., 2023; Pramanick et al., 2024; Xiao et al., 2024), as learned codewords from a segmentation VQ-VAE (Beyer et al., 2024; Wang et al., 2025b;a), or as per patch text labels (Lan et al., 2025b). The formulation unifies detection with segmentation under one head, but its pixel accuracy trails decoder based systems by a wide margin, and the most instructive evidence is internal to the family: adding a SAM-H refiner on top of Text4Seg's 16×16 descriptors lifts RefCOCO val from 73.5 to 79.2 cIoU (Lan et al., 2025b), purely by handing the boundary to a pixel space model while the localization decision stays where it was. Drawing was the part a decoder could absorb, which is why the field attached decoders and why localization is the part still worth improving.

Interface II: the mask as a `[SEG]` embedding. LISA (Lai et al., 2024) removes the mask from the token stream: a `[SEG]` token is added to the vocabulary and its final hidden state prompts SAM. The design has since been extended to grounded conversation, multiple and empty targets, unified pixel level tasks and video (Rasheed et al., 2024; Xia et al., 2024; Zhang et al., 2024b;a; Yan et al., 2024; Yuan et al., 2025). What they share, and what we take as our object of study, is the interface itself: one d dimensional vector with no spatial index is the entire channel for *where*. SAM4MLLM (Chen et al., 2024) is the closest existing probe of this question, emitting coordinates as text so that both arms enjoy the decoder's pixels and only the interface varies; the foundational papers never ran that comparison, arguing against serialization on grounds of reasoning ability and training cost rather than with a same backbone ablation. Section 3.1 asks it from the other direction: how much of the model's spatial knowledge does that vector carry?

What a model's attention already encodes. Multimodal models trained without grounding supervision localize referents in their attention well enough for competitive zero shot boxes (Cao et al., 2025), and Kang et al. (2025b) explain why this has nonetheless looked unusable: averaging heads gives sparse, noisy maps, while a few *localization heads* track the referent reliably. Much of the remaining mass sits on sinks at semantically empty tokens (Kang et al., 2025a), and raw single layer attention needs aggregation before it can be read as evidence (Abnar & Zuidema, 2020; Chefer et al., 2021). Which heads and layers are read is therefore not a detail, which is why our aggregation is selective and learned (Section 3.2). Liu et al. (2025) further find that deep layer attention concentrates on the correct region even when the model answers wrongly, which licenses routing the map around the generative path and warns against using map sharpness as a proxy for end task quality.

Interface III: attention as the spatial channel. Several recent systems turn such a map into segmenter prompts, and all pass through a sparse, discretized bottleneck: VL-SAM (Lin et al., 2024) and DecAF (Han et al., 2025) sample *points* training free, and READ (Qian et al., 2025) samples points from an embedding space similarity map. We differ in three ways that compound. We read the `[SEG]` token's own attention row, so our object of study is the dominant paradigm's internal signal rather than an auxiliary query or a similarity map. We keep the map dense and continuous into the decoder's mask channel and never threshold or sample it, which preserves graded confidence and leaves the gradient path back to the attention intact. And we retain the `[SEG]` embedding as the sparse prompt, so we measure what the spatial channel adds on top of the semantic one rather than swapping one for the other.

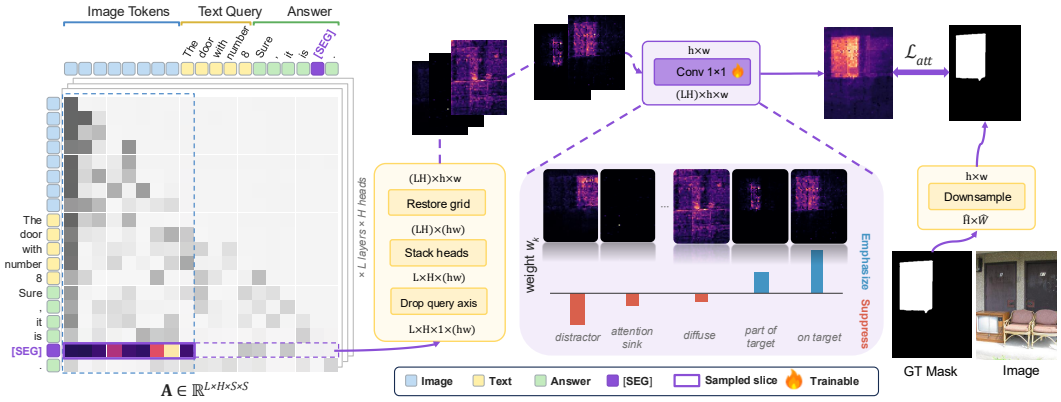


Figure 3: How AttnSeg reads and shapes the spatial channel. The [SEG] token's query row of $\boldsymbol{A}$, restricted to image columns and folded onto the patch grid, gives one map per layer and head; a 1×1 convolution with signed weights combines them and $\mathcal{L}_{\text{att}}$ supervises the result.

3 Methodology

3.1 What the Model Already Knows, and Where It Keeps It

When a [SEG] token is emitted, its attention over the image tokens is already a distribution on the patch grid, whereas its hidden state is a vector with no spatial axis. We therefore hypothesise that the attention, and not the hidden state the dominant interface transmits, is the channel through which a language model should hand a segmenter its spatial intent, and we test that on a released model before designing anything. We freeze Sa2VA-4B (Yuan et al., 2025) entirely, discard the segmenter, and train only a small readout that must predict the mask from one internal signal, so that any score obtained is attributable to the frozen model rather than to the readout. We vary the readout's capacity, which distinguishes grounding that can be read off directly from grounding that must first be reconstructed; the space it reads; and the depth it may see, through a sliding five layer window. The same instrument is applied to the trained model in Section 4.3.

Protocol. From the attention we take, at each of the L=36 layers and H=16 heads, the [SEG] token's query row restricted to the image token columns, folded onto the patch grid and rescaled to spatial mean one, which yields LH=576 maps. Every embedding probe first passes through a shared learned projection onto a 16×16 grid, which alone costs 672K parameters, more than either of the smaller attention probes contains in total, so the comparison is conservative with respect to our own hypothesis. The readouts span four orders of magnitude: a 560 parameter gate, one scalar per layer and head followed by a softmax and a weighted sum, with no convolution, bias or nonlinearity; a 5K convolution; and a 3.86M U-Net. Appendix B gives the details.

The grounding is directly readable, and it lives in the attention. Figure 2a reports both spaces at every capacity. The 560 parameter gate already reaches 26.3 on GS-Eval and 21.6 on ReasonSeg (Lai et al., 2024); having no spatial support, it can produce only shapes an individual head has already drawn, and its parameters do nothing but decide which heads to believe. At U-Net capacity the attention reaches 53.2 and 52.1 gIoU on the two benchmarks, against 35.0 and 30.6 for the [SEG] embedding.

Finding 1. Read by the same architecture, the [SEG] token's attention yields 18.2 and 21.5 more gIoU than its hidden state. The vector the dominant interface transmits is the weaker of the two spatial representations the model computes.

The same figure shows that this advantage is visible only to a readout able to select among the 576 channels: at the 5K convolution level the two spaces are within a few points. That is consistent with existing accounts of attention sinks and of the small number of heads that carry localization at all (Kang et al., 2025b;a), and it constrains the architecture rather than qualifying the claim.

The attention carries structure, the embedding carries a pointer. A representation that merely indicated the correct region would already score well above, so we supply the U-Net probe with

the RGB image as an additional input. A space that only points should allow a probe with access to pixels to grow a boundary around the indicated region, whereas a space that already carries the boundary should render the image redundant. The shaded segment of Figure 2a resolves this: the embedding gains 4.4 and 7.3 gIoU, the attention moves by -0.3 and $+0.2$.

Finding 2. The image helps the `[SEG]` embedding and does not help the `[SEG]` attention. The embedding is a pointer that requires pixels to acquire a shape; the attention already contains the shape.

The signal occupies a narrow late band. Restricting the probe to a sliding five layer window (Figure 2b) shows that the first fifteen layers carry almost nothing and that granting the probe every layer is matched by granting it one late window. Both capacities agree on where the transition occurs, which identifies it as a property of the model and implies that a uniform average over layers and heads spends most of its budget on channels contributing nothing.

3.2 AttnSeg: Attention as the Spatial Interface

The analysis above determines the design rather than merely suggesting one. It identifies the attention as the channel to connect the language model to the segmenter, with the `[SEG]` embedding retained for the semantic answer. It rules out the uniform head average that prior work reads, since the grounding occupies a narrow band and a plain mean spends most of its budget elsewhere. And it argues for learning that combination rather than hard coding the band we measured, both because the band was located for one model and because hard coding it would require repeating the analysis for every new backbone. This subsection makes each of those concrete.

Extracting the spatial channel. For a model of L layers and H heads, self attention over one forward pass is $\boldsymbol{A} \in \mathbb{R}^{L\times H\times S\times S}$, shown at the left of Figure 3. Let r be the `[SEG]` token's position and $\mathcal{I}$ the image token positions, with $|\mathcal{I}| = P = hw$. Exactly one row of $\boldsymbol{A}$ constitutes a spatial statement about the referent, namely row r:

$$\boldsymbol{a}^{(l,h)} \;=\; \big[\, \boldsymbol{A}^{(l,h)}_{r,j} \,\big]_{j\in\mathcal{I}} \;\in\; \mathbb{R}^{P}, \qquad \boldsymbol{a}^{(l,h)} \;\leftarrow\; \frac{P\,\boldsymbol{a}^{(l,h)}}{\sum_{p} \boldsymbol{a}^{(l,h)}_{p}}. \tag{1}$$

Row r is a softmax over the entire sequence, so the mass reaching the image is a small fraction that varies with sequence length and over training; without the rescaling a head would be judged on how much attention it directed at the image rather than on where. Dropping the singleton query axis, stacking the heads and restoring the patch grid gives $\boldsymbol{a} \in \mathbb{R}^{LH\times h\times w}$, whose channels are already expressed in the output's coordinate system, so nothing requires decoding, only combination.

Learning the combination. Those channels disagree with one another, and Section 3.1 shows the disagreement is systematic rather than random. The smallest operator that is selective, learned and able to subtract is a 1×1 convolution from LH channels to one:

$$\hat{\boldsymbol{a}} \;=\; \sum_{k=1}^{LH} w_k\, \boldsymbol{a}_k, \qquad w \in \mathbb{R}^{LH}, \qquad w^{(0)}_k = \tfrac{1}{LH}. \tag{2}$$

It adds 576 parameters for Sa2VA-4B and starts at the uniform value, so at step zero $\hat{\boldsymbol{a}}$ is the plain head mean prior work reads, and training departs from it only where that helps. A softmax gate cannot express the same operation, since its weights lie on the simplex, so a misleading head can be zeroed but never cancelled against a correct one; Appendix C gives a second reason to abandon it.

What the combination learns. The centre of Figure 3 shows the result on a real forward pass. Heads that lock onto the referent or onto part of it receive positive weight, drawn in blue, while a diffuse head, one that has collapsed onto an attention sink and one locked confidently onto a distractor receive negative weight, drawn in red, and are subtracted rather than averaged in. Nothing in the training signal names these categories, and no supervision is applied to w directly; the distinction is acquired from $\mathcal{L}_{\text{att}}$ and the mask loss alone. Across the trained model 240 of the 576 weights are negative, the largest positive weight is $6.2\times$ uniform and the largest negative one $-4.0\times$, and 64% of the positive mass lands in the last twelve layers, so the combination recovers the band the frozen probe identified and additionally subtracts late layer channels the probe could only ignore. Figures 5 and 6 in Appendix E give further examples of the map beside the mask refined from it.

3.3 WHAT THE OBJECTIVE SHOULD ASK OF THE MAP

On the right of Figure 3, $\hat{\boldsymbol{a}}$ is supervised against the ground truth mask, area pooled onto the patch grid as $\boldsymbol{m} \in \{0,1\}^P$. One measurement constrains what that supervision should request before any candidate is considered: the decoder's gradient into $\hat{\boldsymbol{a}}$ is roughly 49 times the magnitude of the attention loss's and roughly orthogonal to it. An objective requesting what the mask loss already requests is therefore both redundant and inaudible, and the useful objective constrains a property the mask loss leaves free.

Which property that is follows from what the decoder can and cannot do. It receives no text, so it cannot reconsider which object was named; it can only denoise the prompt against its own objectness prior, which is strong for large objects and weak for small ones. A poor prompt is therefore recoverable on a large target and decisive on a small one, and of the map's two independent properties, the region it selects and the shape it assigns, only the first lies beyond the decoder's reach.

Three requests, one of which the argument predicts. Region overlap, as a soft Dice term, requests precisely the quantity the mask loss already optimises, at $h \times w$ rather than at 1024px. Pointing, as a peak ratio between the map's strongest responses inside and outside the target (Appendix C), requests selection alone. Between them lies shape agreement, a weighted cosine,

$$\ell^{\cos}(\hat{\boldsymbol{a}}, \boldsymbol{m}) = 1 - \frac{\langle \boldsymbol{c} \odot \hat{\boldsymbol{a}}, \boldsymbol{m} \rangle}{\|\boldsymbol{c} \odot \hat{\boldsymbol{a}}\| \|\boldsymbol{m}\|}, \qquad c_p = \lambda_{\mathrm{fg}} \text{ if } \boldsymbol{m}_p = 1, \;\; \lambda_{\mathrm{bg}} \text{ otherwise,} \tag{3}$$

invariant to the absolute scale Eq. (1) neutralises, and constraining the whole spatial pattern without demanding the calibrated overlap that makes Dice redundant. The observed ordering is the predicted one: Dice yields the map that agrees best with the annotation and is the only variant whose peak lands inside the target less often than for a model trained with no attention loss at all; the peak ratio improves selection and leaves the shape unchanged; cosine improves both, and is the form we adopt.

Two refinements the same mechanism implies. The benefit of correcting the map grows as the target shrinks, as the mechanism above predicts and as we observe across the size quintiles of GS-Eval. A constant foreground weight cannot express that dependence, and the foreground to background imbalance itself varies by an order of magnitude across the mixture, so no single scalar fits it. We therefore weight each supervised row by its target's area. We also exclude rows whose peak already lies inside the target, since the objective does not improve the map monotonically: over a run, it moves some peaks into the target and displaces others, and a displacement costs substantially more than a correction gain. The second refinement becomes necessary once the first is in place, as area weighting concentrates the gradient in exactly the band where displacement occurs. With a_i the area of row i's target and $g_i \in \{0,1\}$ the correctness gate, which is 0 when the peak of $\hat{\boldsymbol{a}}_i$ already falls inside $\boldsymbol{m}_i$ and 1 otherwise,

$$\mathcal{L}_{\mathrm{att}} = \frac{1}{\sum_i g_i s_i} \sum_i g_i \, s_i \, \ell^{\cos}(\hat{\boldsymbol{a}}_i, \boldsymbol{m}_i), \qquad s_i = \mathrm{clamp}\big(a_i^{-\rho},\, s_{\min},\, s_{\max}\big)\,, \tag{4}$$

with $\rho = 0.30$ and the clamp at $[0.25, 12]$. Because g_i enters the normalizer as well as the sum, gating a row redistributes its weight over the rows that are still wrong rather than reducing the objective. The weights are renormalised to mean one over the surviving rows, so only their relative magnitudes change and the scale of the objective, and hence the effective learning rate, is unaffected.

3.4 TRAINING AND IMPLEMENTATION

Training proceeds in two stages, which the gradient ratio above requires: under joint training the map arriving at the decoder is predominantly the decoder's own construction. Stage one finetunes the language model and the aggregation weights under $\mathcal{L}_{\mathrm{att}}$, with the segmenter never constructed, so the map is shaped by $\mathcal{L}_{\mathrm{att}}$ alone. Stage two freezes both, switches $\mathcal{L}_{\mathrm{att}}$ off and trains only the mask branch, leaving the decoder to refine the mask the attention already implies. The map reaches it through SAM's dense mask prompt channel, which the literature leaves at its learned default, while the `[SEG]` embedding remains the sparse prompt. Nothing is thresholded or sampled, so gradients propagate back through Eq. (2) into the language model's queries and keys. Appendix C gives the normalisation, the video case and one detail easily overlooked, namely that freezing the language model does not by itself freeze w.

Base model	ReasonSeg (gIoU)			GS-Eval (gIoU)				
	Sentence	Phrase	Overall	Single obj.	Object parts	Multi obj.	Stuff	Overall
LISA (finetuned) 7B	47.49	39.08	45.47	80.34	21.25	71.95	42.62	56.34
GLaMM (FullScope)	51.68	42.94	49.58	91.38	24.37	78.50	54.11	65.66
Sa2VA-Qwen3-4B-SAM2	59.42	56.28	58.67	91.41	22.79	76.91	54.77	65.41
Sa2VA-Qwen3-4B-SAM3	65.46	61.47	64.50	89.09	17.06	70.23	48.74	60.29
VISA-7B	50.73	43.50	49.00	85.30	25.39	75.70	50.48	62.10
SAMTok-Qwen3VL-4B	59.41	58.03	59.08	87.64	31.01	56.77	56.28	61.82
VideoLISA-3.8B	45.59	33.61	42.72	86.99	23.40	73.54	42.52	58.70
GSVA-13B	38.73	36.75	38.26	77.34	22.84	55.46	38.30	50.58
PixelLM-7B	46.12	37.02	43.94	41.29	11.12	49.96	30.08	34.93
Sa2VA-InternVL2.5-4B	59.51	52.04	57.70	90.80	17.80	73.33	49.80	61.91
+ AttnSeg (ours)	**64.80**	**61.93**	**64.11**	**91.20**	**35.29**	**86.37**	**62.97**	**72.11**
Sa2VA-InternVL3-14B	61.44	59.11	60.88	89.45	26.97	69.66	52.67	63.05
+ AttnSeg (ours)	**69.55**	**70.23**	**69.71**	**90.85**	**45.01**	**85.08**	**66.43**	**74.31**

Table 1: Comparison with prior work. Each highlighted row is AttnSeg on the base model directly above it, so both comparisons are row adjacent. The gain is narrow on single objects and reaches 13 to 18 gIoU on object parts, multiple objects and stuff.

4 Experiments

4.1 Implementation Details

We build AttnSeg on two base models, Sa2VA with InternVL2.5-4B and with InternVL3-14B, so that every claim is tested at a small and at a large scale. Both stages train on the thirteen dataset mixture released with Sa2VA (Yuan et al., 2025), with LoRA at rank 128 on the language model, AdamW at 4×10^{-5} and a global batch of 48 for one epoch per stage; Appendix D gives the mixture and the remaining settings. We report gIoU on two benchmarks, each challenging for a different reason. ReasonSeg (Lai et al., 2024) names the target by its function or its role in the scene rather than by its appearance, as a short phrase or a full sentence, so locating it requires reasoning over the image. GS-Eval (Hu et al., 2025) varies the target's granularity instead, over single objects, multiple instances of one category, object parts and unbounded stuff regions.

4.2 Benchmarking

Main results. Table 1 compares AttnSeg against the base model it is built on and against SoTA methods. On the 4B model, the released checkpoint reaches 57.70 gIoU on ReasonSeg and 61.91 on GS-Eval; AttnSeg reaches 64.11 and 72.11, improvements of 6.41 and 10.20 gIoU. On ReasonSeg the gain is larger on short phrases, 52.04 to 61.93, than on full sentences, 59.51 to 64.80, consistent with the short form offering less context and leaving more of the work to grounding.

The GS-Eval splits show where that improvement is earned. On single objects the base model already reaches 90.80 and AttnSeg adds 0.40, a narrow margin that every competent method in the table shares: when the image contains one salient instance, finding it is not the difficulty and the decoder alone is sufficient. The three remaining splits behave differently. Object parts improve from 17.80 to 35.29, multiple objects from 73.33 to 86.37 and stuff from 49.80 to 62.97, gains of 17.49, 13.04 and 13.17 gIoU. These are the cases where the target must be singled out rather than merely outlined: a part has no boundary of its own to snap to, several instances of one category are indistinguishable except through the expression, and a stuff region has no instance structure.

This distribution is what our premise predicts. Improvement originating in the decoder would appear wherever boundaries are hard and be indifferent to whether the referent was easy to find. We observe the opposite: the gain is negligible where localization is trivial and largest where it is the entire problem, which is what it means for the vision language model to carry the localization and for the decoder to refine what it is handed.

The improvement does not diminish with a stronger base. An auxiliary objective usually benefits a small model most and fades as capacity grows, on the reading that it supplies something a larger model would have acquired by itself. That is not the behaviour here. The 14B model is larger and built on the newer InternVL3 backbone, and on it AttnSeg improves ReasonSeg from 60.88 to

Finetuning	Interface	$\mathcal{L}_{\text{att}}$	ReasonSeg	GS-Eval
✗	Embeddings	✗	57.70	61.91
✓	Embeddings	✗	59.92	62.55
✓	Attention	✗	60.18	63.64
✓	Attention	✓	**64.11**	**72.11**

Table 2: Isolating the source of the gain on Sa2VA-InternVL2.5-4B, gIoU. One factor changes per row, from the released checkpoint (row 1) to AttnSeg (row 4). $\mathcal{L}_{\text{att}}$ accounts for most of the improvement.

	Gating probe (560)		Conv probe		U-Net probe		U-Net probe + image	
	ReasonSeg	GS-Eval	ReasonSeg	GS-Eval	ReasonSeg	GS-Eval	ReasonSeg	GS-Eval
Sa2VA-InternVL2.5-4B	24.00	26.82	25.84	29.89	52.05	53.18	52.21	52.89
+ AttnSeg (ours)	**42.04**	**44.68**	**31.27**	**34.67**	**53.96**	**57.13**	**54.10**	**57.43**

Table 3: Does $\mathcal{L}_{\text{att}}$ make the attention itself easier to read? The probe ladder of Section 3.1 re-run on the released Sa2VA-InternVL2.5-4B and on AttnSeg built from it, in gIoU. Each readout is trained on the frozen model in its row. The gain is largest for the smallest readout.



Figure 4: The attention map is the prediction, not a diagnostic. For each query we show the aggregated `[SEG]` attention $\hat{\boldsymbol{a}}$ and the mask the decoder refines from it. More examples in Appendix E.

69.71 and GS-Eval from 63.05 to 74.31, gains of 8.83 and 11.26. Both are wider than at 4B, and the per split pattern is unchanged: object parts rise from 26.97 to 45.01, multiple objects from 69.66 to 85.08 and stuff from 52.67 to 66.43, while single objects move by 1.40. The objective is therefore not compensating for capacity the base model lacks; it supplies something neither scale obtains from the mask loss alone.

The comparison with recent work makes the same point from the other direction. AttnSeg on InternVL2.5-4B, a backbone two generations old, exceeds SAMTok (Zhou et al., 2026) on a Qwen3-VL-4B backbone by 5.03 gIoU on ReasonSeg and 10.29 on GS-Eval. Strengthening the segmenter instead behaves differently: under the same 4B base, replacing SAM2 with SAM3 (Carion et al., 2026) raises ReasonSeg from 58.67 to 64.50 but lowers GS-Eval from 65.41 to 60.29. A better decoder trades one benchmark against the other, whereas AttnSeg improves both over its own base at both scales, as expected if the headroom that remains is localization rather than boundary quality.

4.3 Analysis

Where the improvement originates. Table 2 varies one factor at a time on the 4B model. Finetuning alone, at the original embedding interface, is worth +2.22 gIoU on ReasonSeg and +0.64 on GS-Eval. Substituting the attention map for that embedding adds a further +0.26 and +1.09, close to negligible on ReasonSeg. The remaining factor is $\mathcal{L}_{\text{att}}$, the only difference between rows 3 and 4, and it contributes +3.93 and +8.47 gIoU, or 61% and 83% of the total.

The small effect of the interface is worth a comment, since Section 3.1 measured the opposite: there the same substitution was worth 18 to 22 gIoU. The difference is what may adapt. A probe is a small readout on a frozen model, so it recovers only what is already laid out plainly and measures accessibility. Here the readout is SAM2, far larger and trained jointly with the language model, so it learns to decode what no probe could reach and the embedding stops being hard to read. We therefore treat the interface not as a source of improvement but as the precondition for one: only the attention map is an object a spatial objective can act on, since a d dimensional embedding has no spatial index.

The attention does not just get better, it gets easier to read. The benchmark gains of Table 1 could in principle be produced by the decoder alone learning to cope with a map that has not changed. To separate the two we repeat the procedure of Section 3.1 on the trained model: the language model is frozen again, and the same three readouts, spanning hundreds, thousands and millions of parameters, are trained from scratch on its attention. Table 3 sets the outcome beside the released checkpoint measured the same way, so both rows are readouts of a frozen model and differ only in whether that model was trained with ATTNSEG.

Every readout improves, but the improvements are not distributed evenly, and that is the informative part. The 3.86M parameter U-Net gains 1.91 gIoU on ReasonSeg and 3.95 on GS-Eval. The 560 parameter gate, which has no convolution, no nonlinearity and no spatial support of any kind, gains 18.04 and 17.86, rising from 24.00 to 42.04 and from 26.82 to 44.68. A readout that can only take a weighted combination of maps the model has already drawn now recovers close to four fifths of what a network five thousand times its size recovers, where before it recovered under half. The gain is largest exactly where capacity is smallest, the signature of a representation that has been reorganised rather than merely enriched. $\mathcal{L}_{\text{att}}$ does not add grounding a readout must then work to extract; it moves grounding into a form a trivial readout can take directly. The two stage recipe depends on that property, since the decoder in stage 2 trains against a frozen map and can only refine what it can read.

The image still adds nothing, before or after training. Section 3.1 showed on the released model that the attention carries the object's structure and not only a pointer to it, and the same test survives training. Two U-Net variants appear in Table 3: one reads the attention alone, the other reads the attention concatenated with the RGB image. After training with $\mathcal{L}_{\text{att}}$ the image is worth 0.14 gIoU on ReasonSeg and 0.30 on GS-Eval, which is nothing. Were the map only a pointer, a readout handed the image as well should be able to turn that pointer into a boundary and gain substantially, which is exactly what the `[SEG]` embedding does in Figure 2a, where the image is worth 4.4 and 7.3 gIoU. The attention gains nothing from it in either model, so the contours it is scored on were already in the map.

Qualitative results. Figure 4 shows the aggregated `[SEG]` attention $\hat{\boldsymbol{a}}$ beside the mask the decoder refines from it. Because the map is the prompt and not a diagnostic, the two can be read against each other directly. Further examples on GS-Eval and ReasonSeg are in Appendix E, Figures 5 and 6.

4.4 LIMITATIONS

The attention has to exist to be read. Our approach needs the attention matrix, and modern kernels are fast precisely because they never build one: FlashAttention style implementations fuse the softmax into the matrix product and keep only the layer's output. An eager fallback forms the matrix and retains it for the backward pass, which on Sa2VA-InternVL3-14B is 48 layers by 40 heads by seq^2, roughly 44 GB in bfloat16 at a typical length of 3400, and does not fit on an 80 GB device beside the model. The resolution follows from what Eq. (1) reads: one row, the one the `[SEG]` token queries, and only its image columns. We keep the fused kernel for the layer output and recompute just those rows beside it, over the full key length and in full precision, so each is what eager would have produced. The retained tensor shrinks by a factor of seq/R for R the number of `[SEG]` queries, about $3400\times$ for a single object, at the cost of one extra matrix product per layer, linear rather than quadratic in sequence length.

5 CONCLUSION

Modern decoders have made drawing the boundary the easy half of segmenting from language, which makes the interface through which a vision language model passes its spatial intent decisive. Both dominant interfaces construct that answer on the way out, as a string or as a vector with no spatial index, and neither is necessary: probing a frozen segmenter shows the `[SEG]` attention carries far more grounding than the hidden state that interface transmits, and the object's shape rather than a pointer to it. ATTNSEG forwards it instead, through a learned signed combination, a dense prompt that is never thresholded, and an objective asking for the property the mask loss leaves free. The gains are largest where the referent must be found and nearly absent where the decoder finds it unaided, as predicted. Two limitations bound the claim: the probing study covers one model family, and every variant of $\mathcal{L}_{\text{att}}$ supervises the map against the decoder's own ground truth.

AI USE STATEMENT

We used generative AI tools as writing assistants, under author direction and on material whose substance the authors had already fixed. Their use was confined to drafting and polishing: redrafting passages from author supplied content and findings, correcting grammar and typography, and condensing sections to the page limit.

We did not use generative AI to generate research ideas, to design experiments, to write or debug the experimental code, to run or analyse experiments, or to produce any number reported in this paper. All results come from the authors' own runs. No AI system is an author. We have reviewed all AI-assisted text and take full responsibility for the final content of this work, including every claim, number and artifact.

ETHICS STATEMENT

This work studies the interface between a vision language model and a promptable mask decoder. It involves no human subjects and no new data collection: every dataset and benchmark we use is public and is used as released, under its own licence and evaluation protocol, and both models we build on are publicly released checkpoints.

Two risks are worth stating plainly. First, our method deliberately moves the localization burden onto the language model, which means the system's errors become the language model's errors. It inherits whatever biases the base model carries in deciding which object a description refers to, and a sharper, more confident attention map can make a confidently wrong choice of referent harder for the decoder to correct rather than easier. An improvement in gIoU on the benchmarks we report is not evidence that the model resolves ambiguous or contested descriptions correctly, and we would not deploy it in a safety critical setting without task specific validation. Second, a system that segments arbitrary objects from free form language lowers the barrier to fine grained visual search, and carries the dual use considerations that come with it.

We have tried to report negative outcomes as carefully as positive ones. Section 4.3 reports that the interface substitution on which the paper is built contributes almost nothing to the end task on its own, Section 3.3 reports that the objective which improves the attention map most is the one that degrades the property the decoder cannot supply, and Section 4.4 states the computational cost our method adds before describing how we reduce it.

REPRODUCIBILITY STATEMENT

Section 3 specifies the method in full: Eq. (1) the extraction, Eq. (2) the aggregation, Eq. (3) and Eq. (4) the objective together with its two refinements, and Section 3.4 the two training stages. Appendix C gives the injection equation, the video case, the pointing variant we compare against, and the one implementation detail that is easy to get wrong, namely that freezing the language model does not by itself freeze the aggregation weights.

Section 4.1 and Appendix D state the complete training configuration, which is identical at both model scales and in both stages: the thirteen dataset mixture released with Sa2VA together with the repetition factor applied to each, LoRA at rank 128 with $\alpha = 256$ and dropout 0.05 on the language model alone, AdamW at 4×10^{-5} with weight decay 0.05, a cosine schedule preceded by a 5% linear warmup, gradient clipping at 1.0, bfloat16 mixed precision, a maximum sequence length of 8192, a global batch of 48 and one epoch per stage. Nothing is tuned per benchmark, so there are no per dataset hyperparameters to recover, and every run uses the same seed. Appendix B gives the probing protocol, including each readout's architecture, parameter count, input and training budget.

One numerical caveat matters for exact replication. Reading the attention requires materialising it, which a fused attention kernel does not do, so we keep the fused kernel for each layer's output and recompute only the `[SEG]` query rows over the full key length with the softmax in full precision (Section 4.4). Those rows are computed exactly as an eager implementation computes them, but the layer outputs come from the fused path, so results are not bitwise identical to a run using eager attention throughout.

Our code, together with the training configuration for every arm reported here, will be made publicly available upon acceptance.

REFERENCES

Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. In *ACL*, 2020.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. PaliGemma: A versatile 3B VLM for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

Shengcao Cao, Liang-Yan Gui, and Yu-Xiong Wang. Emergent visual grounding in large multimodal models without grounding supervision. In *ICCV Findings*, 2025.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. SAM 3: Segment anything with concepts. In *International Conference on Learning Representations*, volume 2026, pp. 138846–138923, 2026.

Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *CVPR*, 2021.

Xi Chen, Xiao Wang, Lucas Beyer, Alexander Kolesnikov, Jialin Wu, Paul Voigtlaender, Basil Mustafa, Sebastian Goodman, Ibrahim Alabdulmohsin, Piotr Padlewski, et al. PaLI-3 vision language models: Smaller, faster, stronger. *arXiv preprint arXiv:2310.09199*, 2023.

Yi-Chia Chen, Wei-Hua Li, Cheng Sun, Yu-Chiang Frank Wang, and Chu-Song Chen. SAM4MLLM: Enhance multi-modal large language model for referring expression segmentation. In *ECCV*, 2024.

Su Ho Han, Jeongseok Hyun, Pilhyeon Lee, Minho Shim, Dongyoon Wee, and Seon Joo Kim. Decomposed attention fusion in MLLMs for training-free video reasoning segmentation. *arXiv preprint arXiv:2510.19592*, 2025.

Rui Hu, Lianghui Zhu, Yuxuan Zhang, Tianheng Cheng, Lei Liu, Heng Liu, Longjin Ran, Xiaoxin Chen, Wenyu Liu, and Xinggang Wang. GroundingSuite: Measuring complex multi-granular pixel grounding. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 23105–23114. IEEE, 2025.

Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. See what you are told: Visual attention sink in large multimodal models. In *ICLR*, 2025a.

Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. Your large vision-language model only needs a few attention heads for visual grounding. *arXiv preprint arXiv:2503.06287*, 2025b.

Lei Ke, Mingqiao Ye, Martin Danelljan, Yifan Liu, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Segment anything in high quality. In *NeurIPS*, 2023.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *ICCV*, 2023.

Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. LISA: Reasoning segmentation via large language model. In *CVPR*, 2024.

Mengcheng Lan, Chaofeng Chen, Jiaxing Xu, Zongrui Li, Yiping Ke, Xudong Jiang, Yingchen Yu, Yunqing Zhao, and Song Bai. Text4Seg++: Advancing image segmentation via generative language modeling. *arXiv preprint arXiv:2509.06321*, 2025a.

Mengcheng Lan, Chaofeng Chen, Yue Zhou, Jiaxing Xu, Yiping Ke, Xinjiang Wang, Litong Feng, and Wayne Zhang. Text4Seg: Reimagining image segmentation as text generation. In *ICLR*, 2025b.

Yuqi Lin, Hengjia Li, Wenqi Shao, Zheng Yang, Jun Zhao, Xiaofei He, Ping Luo, and Kaipeng Zhang. SAMRefiner: Taming segment anything model for universal mask refinement. In *ICLR*, 2025.

Zhiwei Lin, Yongtao Wang, and Zhi Tang. Training-free open-ended object detection and segmentation via attention as prompts. In *NeurIPS*, 2024.

Jiang Liu, Hui Ding, Zhaowei Cai, Yuting Zhang, Ravi Kumar Satzoda, Vijay Mahadevan, and R. Manmatha. PolyFormer: Referring image segmentation as sequential polygon generation. In *CVPR*, pp. 18653–18663, 2023.

Zhining Liu, Ziyi Chen, Hui Liu, Chen Luo, Xianfeng Tang, Suhang Wang, Joy Zeng, Zhenwei Dai, Zhan Shi, Tianxin Wei, Benoit Dumoulin, and Hanghang Tong. Seeing but not believing: Probing the disconnect between visual attention and answer correctness in VLMs. *arXiv preprint arXiv:2510.17771*, 2025.

Shraman Pramanick, Guangxing Han, Rui Hou, Sayan Nag, Ser-Nam Lim, Nicolas Ballas, Qifan Wang, Rama Chellappa, and Amjad Almahairi. Jack of all tasks, master of many: Designing general-purpose coarse-to-fine vision-language model. In *CVPR*, 2024.

Rui Qian, Xin Yin, and Dejing Dou. Reasoning to attend: Try to understand how <SEG> token works. In *CVPR*, 2025.

Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M. Anwer, Eric Xing, Ming-Hsuan Yang, and Fahad S. Khan. GLaMM: Pixel grounding large multimodal model. In *CVPR*, 2024.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. In *ICLR*, 2025.

Zhongwei Ren, Zhicheng Huang, Yunchao Wei, Yao Zhao, Dongmei Fu, Jiashi Feng, and Xiaojie Jin. PixelLM: Pixel reasoning with large multimodal model. In *CVPR*, 2024.

Haiyang Wang, Hao Tang, Li Jiang, Shaoshuai Shi, Muhammad Ferjad Naeem, Hongsheng Li, Bernt Schiele, and Liwei Wang. GiT: Towards generalist vision transformer through universal language interface. In *ECCV*, 2024.

Lingfeng Wang, Hualing Lin, Senda Chen, Tao Wang, Changxu Cheng, Yangyang Zhong, Dong Zheng, and Wuyue Zhao. ALTo: Adaptive-length tokenizer for autoregressive mask generation. *arXiv preprint arXiv:2505.16495*, 2025a.

Tao Wang, Changxu Cheng, Lingfeng Wang, Senda Chen, and Wuyue Zhao. HiMTok: Learning hierarchical mask tokens for image segmentation with large multimodal model. In *ICCV*, 2025b.

Wenhai Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, and Jifeng Dai. VisionLLM: Large language model is also an open-ended decoder for vision-centric tasks. In *NeurIPS*, 2023.

Jiannan Wu, Muyan Zhong, Sen Xing, Zeqiang Lai, Zhaoyang Liu, Zhe Chen, Wenhai Wang, Xizhou Zhu, Lewei Lu, Tong Lu, Ping Luo, Yu Qiao, and Jifeng Dai. VisionLLM v2: An end-to-end generalist multimodal large language model for hundreds of vision-language tasks. In *NeurIPS*, 2024.

Zhuofan Xia, Dongchen Han, Yizeng Han, Xuran Pan, Shiji Song, and Gao Huang. GSVA: Generalized segmentation via multimodal large language models. In *CVPR*, 2024.

Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. Florence-2: Advancing a unified representation for a variety of vision tasks. In *CVPR*, 2024.

Cilin Yan, Haochen Wang, Shilin Yan, Xiaolong Jiang, Yao Hu, Guoliang Kang, Weidi Xie, and Efstratios Gavves. VISA: Reasoning video object segmentation via large language models. In *ECCV*, 2024.

Senqiao Yang, Tianyuan Qu, Xin Lai, Zhuotao Tian, Bohao Peng, Shu Liu, and Jiaya Jia. LISA++: An improved baseline for reasoning segmentation with large language model. *arXiv preprint arXiv:2312.17240*, 2023.

Haobo Yuan, Xiangtai Li, Tao Zhang, Yueyi Sun, Zilong Huang, Shilin Xu, Shunping Ji, Yunhai Tong, Lu Qi, Jiashi Feng, and Ming-Hsuan Yang. Sa2VA: Marrying SAM2 with MLLM for dense grounded understanding of images and videos. *arXiv preprint arXiv:2501.04001*, 2025.

Tao Zhang, Xiangtai Li, Hao Fei, Haobo Yuan, Shengqiong Wu, Shunping Ji, Chen Change Loy, and Shuicheng Yan. OMG-LLaVA: Bridging image-level, object-level, pixel-level reasoning and understanding. In *NeurIPS*, 2024a.

Zheng Zhang, Yeyao Ma, Enming Zhang, and Xiang Bai. PSALM: Pixelwise segmentation with large multi-modal model. In *ECCV*, 2024b.

Yikang Zhou, Tao Zhang, Dengxian Gong, Yuanzheng Wu, Ye Tian, Haochen Wang, Haobo Yuan, Jiacong Wang, Lu Qi, Hao Fei, et al. SAMTok: Representing any mask with two words. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 37852–37863, 2026.

A Extended Related Work

This appendix expands Section 2 family by family. The main text keeps only the evidence the argument turns on.

Promptable segmentation, and its unused channel. SAM (Kirillov et al., 2023) turns a prompt into a mask at 1024px and SAM2 (Ravi et al., 2025) extends it to video with a memory bank that propagates a mask from a few conditioning frames. Its prompt encoder treats its three prompt types in two ways: points and boxes are encoded *sparsely*, as positional embeddings concatenated to the decoder's query tokens, whereas a mask is encoded *densely*, downscaled by a small convolutional stack and added elementwise to the image embedding. The dense channel is the least used of the three. Almost every multimodal segmenter leaves it at SAM's learned `no_mask_embed` default; where it is used, it is used as a coarse prior to be cleaned up. SAMRefiner (Lin et al., 2025) mines Gaussian style mask prompts from noisy masks precisely so SAM can repair them, and HQ-SAM (Ke et al., 2023) quantifies how much boundary quality remains recoverable downstream of a given prompt. The property we exploit follows from that elementwise addition: the dense channel accepts a soft, full field spatial distribution without requiring it to be a mask first, which is exactly the shape of an attention map, and is why our interface needs no thresholding step.

Contours as coordinate sequences. PolyFormer (Liu et al., 2023) established the formulation before large language models entered the field, casting referring segmentation as sequential polygon generation and regressing floating point vertices to avoid quantization error. VisionLLM (Wang et al., 2023) carried it into the LLM era, parsing masks as polygon sequences so segmentation becomes ordinary text prediction. VistaLLM (Pramanick et al., 2024) improved the serialization with gradient aware adaptive sampling, placing contour points where the boundary actually bends. Florence-2 (Xiao et al., 2024) adds 1000 location tokens to the vocabulary and emits vertices in clockwise order with no mask decoder, and GiT (Wang et al., 2024) routes dense prediction through a single universal language interface. Tellingly, VisionLLM's own successor (Wu et al., 2024) reintroduces task specific decoders.

Masks as learned discrete tokens. PaLI-3 and PaliGemma (Chen et al., 2023; Beyer et al., 2024) extend the tokenizer with 128 codewords from a lightweight segmentation VQ-VAE, so a mask is four box tokens followed by sixteen codebook indices. HiMTok (Wang et al., 2025b) makes the vocabulary hierarchical, representing a mask in up to 32 coarse to fine tokens and, unlike the `[SEG]` route, needing no access to the image at de-tokenization time; ALTo (Wang et al., 2025a) makes the token budget adaptive to object complexity. These are a genuine middle ground: discrete like text, but with an inductive bias the language model can exploit, since neighbouring codewords denote similar shapes.

Masks as dense patch labels. Text4Seg (Lan et al., 2025b) abandons geometry and labels every patch of a coarse grid with a text descriptor, compressing the string with a row wise run length encoding that cuts sequence length by 74% and speeds up inference roughly threefold; Text4Seg++ (Lan et al., 2025a) extends the recipe. It is the strongest evidence that the text interface is not inherently weak: with an InternVL2-8B backbone it reaches 75.4 average cIoU on the RefCOCO series against 69.9 for LISA and 75.6 for GLaMM.

Why the text interface loses on geometry. Four reasons, all structural rather than incidental. The resolution is the language model's own coarse patch grid. A token cross entropy penalizes a vertex off by one bin exactly as it penalizes one off by a hundred, absent an explicit number aware objective. Under contour or run length encodings a one pixel shift rewrites many tokens, so small spatial errors become large, incoherent token errors. And a single mis-predicted early vertex distorts the remainder of the contour.

Extensions of the `[SEG]` interface. Beyond those named in Section 2, PixelLM (Ren et al., 2024) replaces SAM with a lightweight decoder and LISA++ (Yang et al., 2023) adds instance level output. LISA also argues the efficiency case directly against VisionLLM, contrasting 4×8 A100-80G GPUs for 50 epochs with under three days on eight 3090s, which is an argument about training cost rather than about the interface.

Offline versus learned head selection. Kang et al. (2025b) isolate localization heads with two criteria: an attention sum that keeps heads dominantly attending to the image, and a spatial entropy over the binarized map's connected components that keeps heads whose mass forms one compact cluster. They take the ten lowest entropy heads per sample, repeat over 1,000 image text pairs, and

Readout	Input	Params	What it can do
Gate	attention, $(LH) \times h \times w$	560	one scalar per (layer, head), softmax, weighted sum
	`[SEG]` embedding, $L \times d$	672 036	one scalar per layer, then the shared projection
Conv	attention	5 041	a single 3×3 convolution, no norm, no nonlinearity
	`[SEG]` embedding	672 325	the shared projection, then the same 3×3 convolution
U-Net	attention	3.86 M	three stages, base width 64, skip connections
	`[SEG]` embedding	4.23 M	the shared projection, then the same U-Net

Table 4: The probe ladder. Three readouts of steeply increasing capacity, applied to both spaces with the vision language model frozen. The gate has no convolution, bias, nonlinearity or spatial support, so it can only choose among maps the model already drew.

rank by selection frequency. That selection is offline and then frozen. Because our map sits on a differentiable path to the mask loss, Eq. (2) instead lets the task choose, and it additionally admits negative weights, which a frequency ranking cannot express.

B Probing Protocol

Table 4 gives the three readouts and both input spaces. The attention input is $(560, h, w)$, one channel per (layer, head) over LLM layers $1..35 \times 16$ heads, each channel rescaled to spatial mean 1.0. The embedding input is $(36, 2048)$, one `[SEG]` vector per layer; all three embedding probes share one projection, a per layer LayerNorm affine ($147{,}456$ parameters) plus one $\mathrm{Linear}(2048 \to 16 \times 16)$ shared across layers ($524{,}544$), reshaped to a 16×16 grid and bilinearly resized to the sample's canvas.

Architectures. The gate is one scalar per (layer, head) with a softmax over all 560 at $T{=}1.0$, output as their weighted sum, with no conv, bias, nonlinearity or norm. The conv is a single 3×3 convolution, $560 \to 1$, with no norm and no nonlinearity. The U-Net has three stages at base width 64: a 3×3 conv $560 \to 64$, an encoder $64 \to 128 \to 256$ with $\mathrm{MaxPool2d}(2)$, a 256 channel bottleneck, a decoder via $\mathrm{ConvTranspose2d}(2, 2)$ with skip concatenation back to $128 \to 64$, and a 1×1 conv to one channel; every block is two repetitions of 3×3 conv, $\mathrm{GroupNorm}(1)$ and ReLU.

Training and scoring. Three epochs on a fixed training subset, scored at full image resolution with the vision language model frozen throughout and the segmenter unused.

One measurement caveat. The gate's output is a convex combination of non-negative maps, so after min-max normalisation essentially every pixel exceeds 0.5 and a fixed threshold makes its mask all foreground by construction. Gate numbers are therefore reported at a swept threshold while every other row is fixed at 0.5, which flatters the gate. We use the gate only for the qualitative accessibility claim (Section 3.1) and never in a head to head comparison.

C Additional Method Details

Why we abandoned the softmax gate. Section 3.2 gives the structural reason: a softmax puts the weights on the simplex, so a misleading head can be driven toward zero but never cancelled against a correct one. There is a second, practical reason. With an unbounded logit z and a saturating map, nothing holds the gate back from collapsing onto a handful of channels, and the temperature T only changes *when* that collapse happens rather than whether it happens. The 1×1 convolution in Eq. (2) has no softmax, no temperature and no normalisation, so the failure mode does not exist on that path.

The pointing objective, and why it is a ratio and not a margin. With π the sum normalised map rescaled by its own detached maximum and $\pi_{\mathrm{fg}}, \pi_{\mathrm{bg}}$ the top k means inside and outside $\boldsymbol{m}$, the pointing form of Section 3.3 is

$$\mathcal{L}_{\mathrm{att}}^{\mathrm{peak}} = -\log \frac{\pi_{\mathrm{fg}} + \epsilon}{\pi_{\mathrm{fg}} + \pi_{\mathrm{bg}} + 2\epsilon}. \tag{5}$$

The natural way to write such an objective is instead a margin, $\max(0, \tau - (\pi_{\mathrm{fg}} - \pi_{\mathrm{bg}}))$, and it does not work. The peak rescaling in Section 3.3 compresses both quantities toward one, so the achievable gap saturates and any fixed τ below that ceiling is eventually satisfied on every sample. The term

Task	Dataset	Repetition factor
Semantic segmentation	COCO-Stuff	0.5
	ADE20K	1
	Pascal-Part	4
Referring segmentation	RefCOCO	2
	RefCOCO+	2
	RefCOCOg	2
Referring video segmentation	MeViS	4
	ReVOS	10
	Ref-YouTube-VOS	4
Grounded conversation	GranD-f	10
	OpenPSG	1
	RefCOCOg-GCG	0.5
	Flickr-30k Entities	0.25

Table 5: Training mixture and sampling weights, the thirteen datasets released with Sa2VA (Yuan et al., 2025), identical in both stages and at both model scales. A factor above one oversamples a dataset within the epoch and a factor below one is a seeded subsample.

then emits no gradient at all and the run silently decays into the baseline. Measured with $\tau = 0.2$, the fraction of samples still producing a gradient fell from 0.34 to zero within the first fifth of training, so most of the run carried no pointing signal. Eq. (5) is strictly positive for any finite gap and decays smoothly as the map improves, which keeps pressure on bad maps without dominating once they are good.

What each stage trains. Stage one finetunes the language model with LoRA together with the LH aggregation weights, under $\mathcal{L}_{\text{att}}$ and the usual language modelling loss. The segmenter is never constructed, never called and contributes no gradient, so the map is shaped by $\mathcal{L}_{\text{att}}$ alone and a recipe that trained both at once could not establish that the language model performs the localization. Stage two freezes the language model completely, base weights and LoRA adapters alike, and freezes w with it, so $\hat{\boldsymbol{a}}$ becomes a fixed function of the input. $\mathcal{L}_{\text{att}}$ is switched off, the map now being an input rather than a target, and only the mask branch trains.

Injecting the map into the dense prompt channel. We normalise $\hat{\boldsymbol{a}}$ per sample, resize it to the prompt encoder's resolution and return it to logit scale,

$$\tilde{\boldsymbol{a}} = \frac{\hat{\boldsymbol{a}} - \min \hat{\boldsymbol{a}}}{\max \hat{\boldsymbol{a}} - \min \hat{\boldsymbol{a}} + \epsilon}, \qquad \boldsymbol{M} = \operatorname{logit}\big(\operatorname{clamp}\big(\operatorname{resize}_{256}(\tilde{\boldsymbol{a}}), \epsilon, 1-\epsilon\big)\big)\,, \tag{6}$$

after which $\boldsymbol{M}$ is the dense prompt and the `[SEG]` embedding $\boldsymbol{h}_{\text{SEG}}$ the sparse one, so the two channels carry which object and where it is respectively. Because nothing along this path is thresholded or sampled, gradients propagate through Eq. (6) and Eq. (2) into the language model's queries and keys. On video, Eq. (1) yields one map per object per conditioning frame, each frame is prompted with its own map, and the memory bank of SAM2 propagates the result through the clip.

Freezing the aggregation in stage 2. Freezing the language model is not by itself enough to freeze the map. The LH weights w of Eq. (2) are registered outside the language model, so a freeze that covers only the language model's parameters leaves w still taking gradient from the decoder. The map would then go on adapting to the decoder while the configuration claims the language model is frozen, and stage 2 would silently become a slower version of joint training. Both w and the language model's LoRA adapters must be frozen explicitly.

D More Experiment Details

Training mixture. Table 5 lists the thirteen datasets of the training mixture and the repetition factor applied to each. The mixture, the factors and the seed are the same in both stages and at both model scales, so a comparison between a released checkpoint and AttnSeg differs only in the interface and the objective.

Benchmarks. ReasonSeg (Lai et al., 2024) reports separately on the two forms its referring expressions take, a short phrase and a full sentence, and on their union; Table 1 follows that split. GS-Eval (Hu et al., 2025) reports four splits by the granularity of the target: a single object, multiple instances of one category, an object part, and an unbounded stuff region.

Optimisation. Nothing about the data or its sampling changes between the two stages or between the two model scales. Both stages use AdamW at a learning rate of 4×10^{-5} with weight decay 0.05, a cosine schedule preceded by a linear warmup over the first 5% of the budget, gradient clipping at 1.0, bfloat16 mixed precision and a maximum sequence length of 8192. Training runs on a single node of H100 80 GB GPUs with a global batch of 48, and each stage runs for one epoch. The language model is adapted with LoRA at rank 128 and $\alpha = 256$ with dropout 0.05; the vision encoder, the token embedding, the language modelling head and the vision language projector remain frozen throughout. The same settings are used at both model scales.

E Additional Qualitative Results

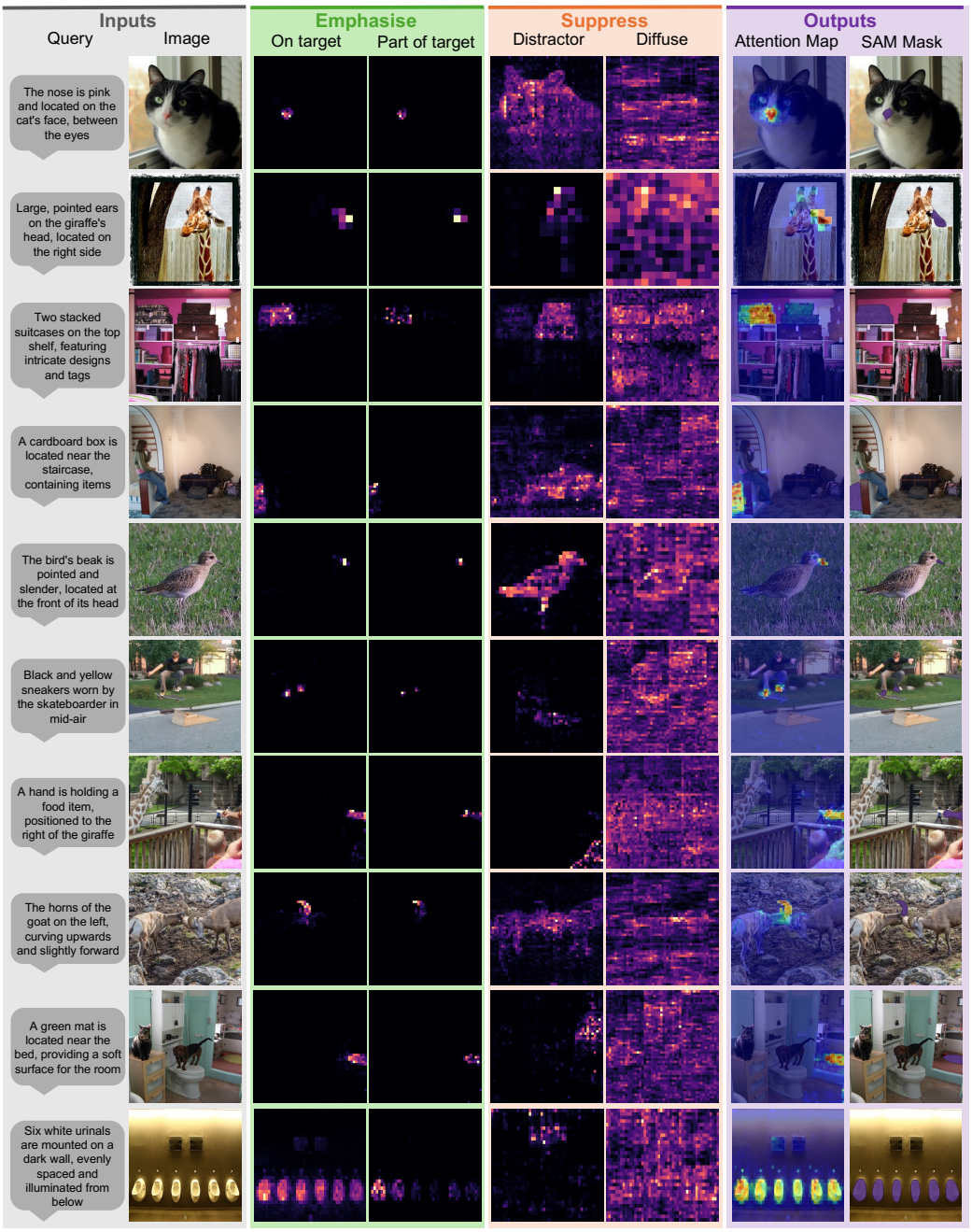


Figure 5: Additional qualitative results on GS-Eval. Query, aggregated `[SEG]` attention map, and the refined mask.

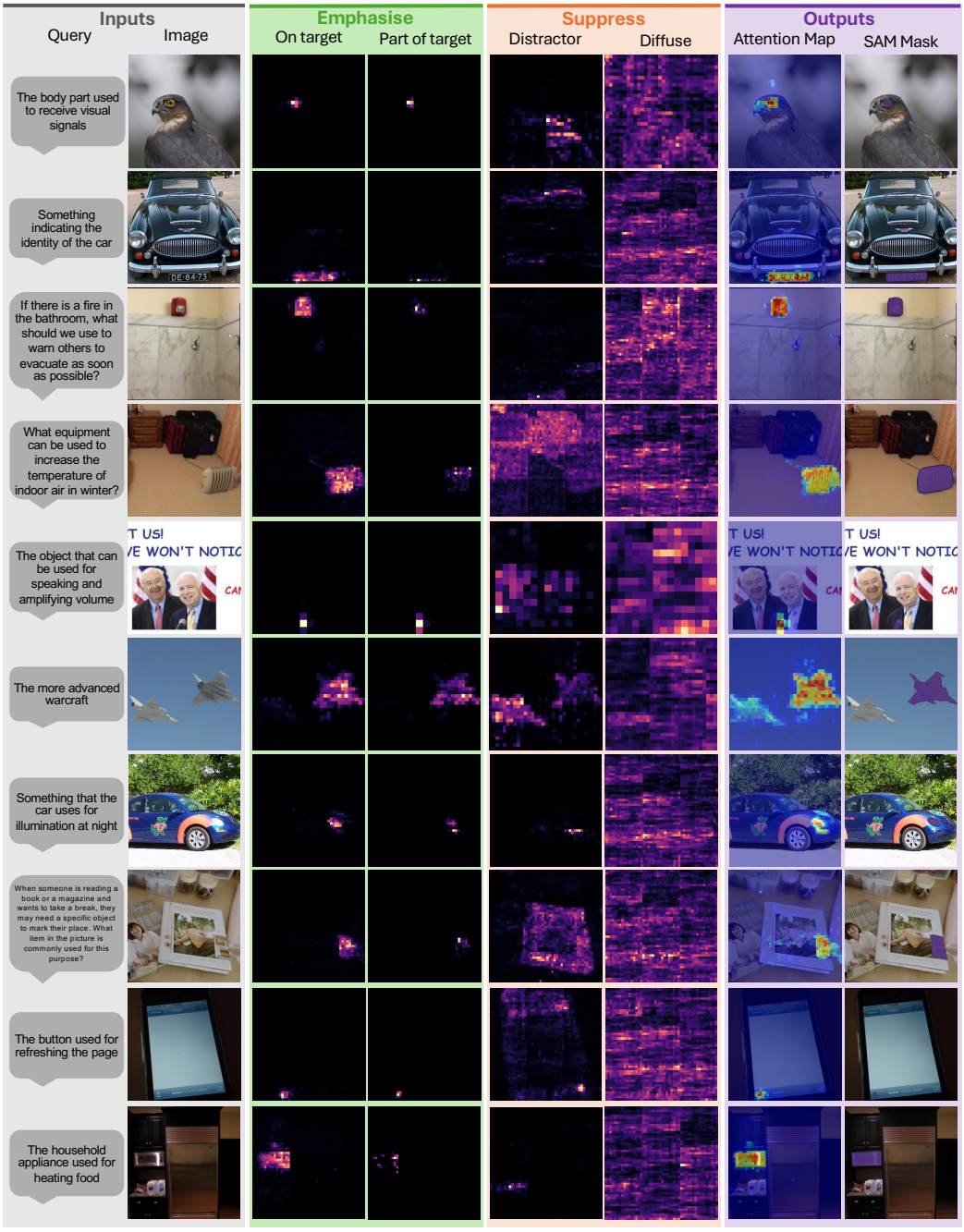


Figure 6: Additional qualitative results on ReasonSeg. Query, aggregated `[SEG]` attention map, and the refined mask.