Better, Stronger, Faster: Tackling the Trilemma in MLLM-based Segmentation with Simultaneous Textual Mask Prediction

Jiazhen Liu Mingkuan Feng* Long Chen†
The Hong Kong University of Science and Technology
jliugj@connect.ust.hk, fmk24@mails.tsinghua.edu.cn, longchen@ust.hk
https://github.com/HKUST-LongGroup/STAMP

Abstract

Integrating segmentation into Multimodal Large Language Models (MLLMs) presents a core trilemma: simultaneously preserving dialogue ability, achieving high segmentation performance, and ensuring fast inference. Prevailing paradigms are forced into a compromise. Embedding prediction methods introduce a conflicting pixel-level objective that degrades the MLLM's general dialogue abilities. The alternative, next-token prediction, reframes segmentation as an autoregressive task, which preserves dialogue but forces a trade-off between poor segmentation performance with sparse outputs or prohibitive inference speeds with rich ones. We resolve this trilemma with ***all-mask prediction****, a novel paradigm that decouples autoregressive dialogue generation from non-autoregressive mask prediction. We present* STAMP[1]*, an MLLM that embodies this paradigm. After generating a textual response,* STAMP *predicts an entire segmentation mask in a single forward pass by treating it as a parallel "fill-in-the-blank" task over image patches. This design maintains the MLLM's dialogue ability by avoiding conflicting objectives, enables high segmentation performance by leveraging rich, bidirectional spatial context for all mask tokens, and achieves exceptional speed. Extensive experiments show that* STAMP *significantly outperforms state-of-the-art methods across multiple segmentation benchmarks, providing a solution that excels in dialogue, segmentation, and speed without compromise.*

1. Introduction

The success of Multimodal Large Language Models (MLLMs) has spurred a trend to unify diverse vision tasks within a single instruction-driven framework [4, 14, 20, 50, 53, 55]. For such models to be practical, they must resolve a

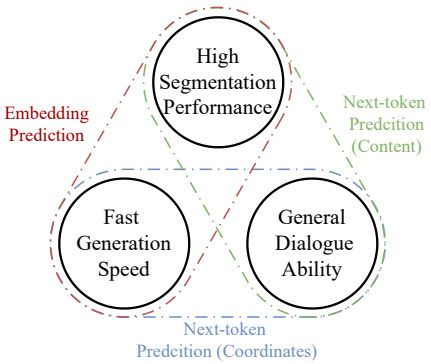


Figure 1. **The trilemma of segmentation in MLLMs.** Embedding prediction may harm dialogue abilities. Next-token prediction methods are either fast with poor segmentation performance or achieve superior performance at the cost of inference speed, particularly when generating rich content (*e.g.*, chain-of-thought or patch-wise classification).

Table 1. **Comparison of MLLM segmentation paradigms.** "Steps" refers to mask generation steps (excluding textual prefix). "Decoder-free" indicates whether an external mask decoder is necessary. "Patch Class." refers to patch-wise classification.

Model	Output	Steps	Token-only Supv.	Decoder-free
Paradigm 1: Embedding Prediction				
LISA (CVPR'24) [18]	Embeddings	$\mathcal{O}(1)$	✗	✗
GSVA (CVPR'24) [48]	Embeddings	$\mathcal{O}(1)$	✗	✗
PixelLM (CVPR'24) [40]	Embeddings	$\mathcal{O}(1)$	✗	✗
M^{2}SA (ICLR'25) [13]	Embeddings	$\mathcal{O}(1)$	✗	✗
READ (CVPR'25) [35]	Embeddings	$\mathcal{O}(1)$	✗	✗
Paradigm 2: Next-token Prediction				
VisionLLM (NIPS'24) [46]	Coordinates	$\mathcal{O}(N_{\text{points}})$	✓	✓
Seg-Zero (arXiv'25) [30]	CoT + Coords.	$\mathcal{O}(N_{\text{CoT}})$	✗	✗
SegAgent (CVPR'25) [56]	CoT + Coords.	$\mathcal{O}(N_{\text{CoT}})$	✓	✗
Text4Seg (ICLR'25) [19]	Patch Class.	$\mathcal{O}(N_{\text{patches}})$	✓	✓
Paradigm 3: All-mask Prediction (Ours)				
STAMP (Ours)	Patch Class.	$\mathcal{O}(1)$	✓	✓

*Work done during his visit at HKUST.
†Corresponding author.
[1] *STAMP*: **S**imultaneous **T**extual **A**ll-**M**ask **P**rediction.

core trilemma: simultaneously (i) preserving dialogue abilities, (ii) achieving high task performance, and (iii) ensuring fast inference. While this has been successful for recognition [20, 55] and detection [2, 50], the trilemma remains unsolved for segmentation. This stems from a fundamental challenge: the MLLM's sequential text-generative nature is ill-suited for producing dense, pixel-level masks [19, 27]. Consequently, current MLLM segmentation paradigms are forced into a trade-off, where they must compromise on one or more of these fronts.

Fig. 1 illustrates how mainstream MLLM segmentation paradigms tackle this trilemma, with their primary differences rooted in the MLLM's role and output format (*cf*. Tab. 1). The first paradigm, *embedding prediction* (*cf*. Fig. 2a), achieves good segmentation performance by fine-tuning the MLLM with a pixel-level mask loss to control an external mask decoder [18, 35, 38, 40, 48]. However, this auxiliary objective degrades general dialogue abilities [19, 27, 47], as seen in models like LISA [18], which may fail to follow a simple instruction like "How many deer are there?" and instead output a segmentation result [27].

The second paradigm, next-token prediction (*cf*. Fig. 2b), avoids this objective conflict by reframing segmentation as a pure language modeling task, where the MLLM autoregressively generates a textual representation of the mask. An early approach was to generate sparse *token-to-coordinates* outputs, such as polygon vertices defining an object's contour [42]. This method, however, proved unreliable due to its high susceptibility to error accumulation, where a single inaccurate point can corrupt the entire mask [19, 46], leading to poor segmentation performance. To mitigate this unreliability, a recent *token-to-content* trend generates richer outputs, like Chain-of-Thought (CoT) reasoning [30, 56] or patch-wise foreground/background classifications (*cf*. Fig. 2b) [19]. While these strategies improve segmentation performance, the autoregressive generation of extremely long token sequences results in prohibitively slow inference speeds.

Evidently, existing paradigms face an inherent trade-off among dialogue ability, segmentation performance, and inference speed. In this work, we break this trilemma by formulating mask generation as a non-autoregressive, patch-wise classification task. First, this approach keeps supervision at the token level by defining each mask token as a textual classification for a corresponding image patch, thus preserving the MLLM's dialogue capabilities. Second, the direct one-to-one correspondence between mask tokens and image patches provides the rich representation required for high segmentation performance. Finally, by fixing the length of mask tokens to match the number of image patches, we reframe mask generation as a fill-in-the-blank task. This enables the simultaneous prediction of all mask tokens over pre-defined placeholders in a single forward pass, achieving exceptional inference speed.

Thus, we introduce **all-mask prediction** (*cf*. Fig. 2c), a novel paradigm that decouples autoregressive dialogue from non-autoregressive mask generation. Beyond resolving the trilemma, this paradigm offers two critical advantages. For one, by generating all mask tokens in a single pass, our method ensures that every token is mutually visible during prediction. This allows the model to leverage a rich, bidirectional 2D spatial context, overcoming the unidirectional limitation inherent in autoregressive approaches. Moreover, by defining placeholders (`[MASK]`) based on the input image, our method inherits a natural compatibility with dynamic resolutions [2, 50, 55].

To validate our paradigm, we implement *STAMP*[1] model. After generating a textual response, the model can emit a special `<SEG>` token. This triggers a single-pass prediction stage where bidirectional attention is applied over all mask placeholders, whose initial embeddings are fused with their corresponding image patch features. The effectiveness of this approach is immediate: on the standard referring segmentation benchmark, *STAMP* establishes a new state-of-the-art, outperforming the strongest prior methods by 4.5%. Crucially, it achieves this with fast inference speeds on par with the embedding prediction methods, all while maintaining exceptional dialogue ability.

In summary, our main contributions are threefold:

- We introduce *all-mask prediction*, a novel prediction paradigm that resolves the segmentation trilemma by avoiding the trade-offs of prior autoregressive and embedding-based methods.
- We propose *STAMP*, the first realization of this paradigm, demonstrating how to seamlessly integrate autoregressive dialogue with non-autoregressive mask generation.
- *STAMP* achieves state-of-the-art results across multiple segmentation benchmarks, significantly advancing performance and speed while preserving the MLLM's core dialogue capabilities.

2. Related Work

Multimodal Large Language Models (MLLMs). The advent of MLLMs, powered by the advanced reasoning of their LLM foundations [9, 15, 34], has established a new frontier for instruction-driven vision tasks, including dense prediction like segmentation. Architectures such as LLaVA [20, 25, 26], InstructBLIP [7], and Qwen-VL [1, 2, 50] typically bridge a visual encoder with a pre-trained LLM, enabling them to ground complex language instructions within an image's spatial context. The central challenge for segmentation, therefore, is how to translate this high-level, spatially-aware understanding into a pixel-level mask. Two primary strategies have emerged to address this: one leverages the MLLM as a powerful vision-language encoder, using its internal embeddings to guide a separate

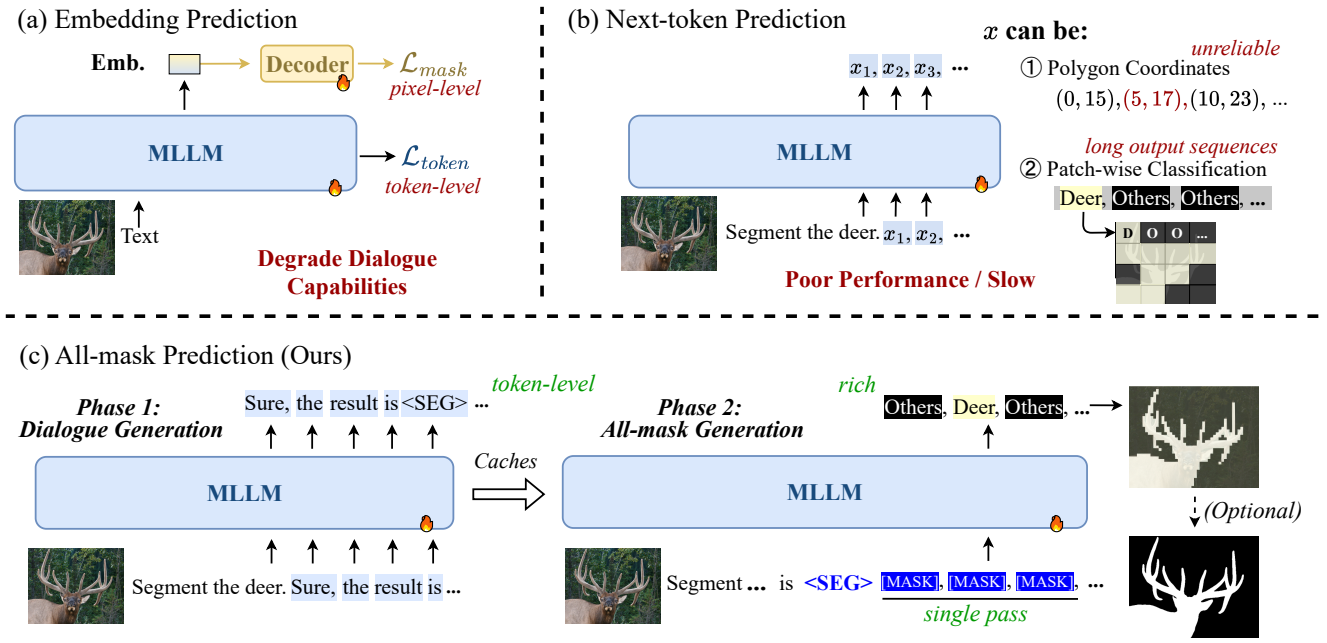


Figure 2. **Comparison of MLLM-based segmentation paradigms. (a) Embedding Prediction**: A conflicting pixel-level objective [18, 40] degrades the MLLM's general dialogue capabilities. **(b) Next-token Prediction**: Generates masks autoregressively [19, 30, 42], forcing a trade-off between poor segmentation performance (for sparse outputs) and slow inference (for rich outputs). **(c) Our All-mask Prediction**: We decouple dialogue generation (autoregressive) from mask generation (non-autoregressive). By simultaneously predicting all mask tokens as patch-wise classifications in a single pass, our paradigm resolves the segmentation trilemma, uniting preserved dialogue abilities, high segmentation performance and fast inference speed.

mask decoder [18, 48]; the other reformulates mask generation as a language task, compelling the MLLM to "describe" the mask through text [19, 30, 46].

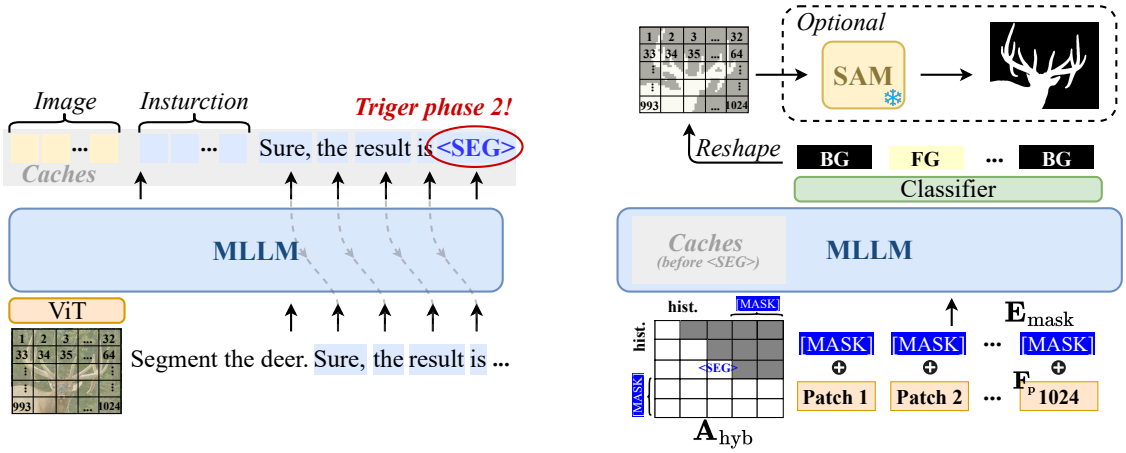
MLLM-based Segmentation. Existing MLLM-based segmentation methods can be categorized into two primary paradigms: *Embedding Prediction* and *Next-token Prediction* (*cf*. Tab. 1). They are distinguished by the MLLM's output format for mask generation. The former employs the MLLM to produce continuous embeddings that guide an external mask decoder. The latter, however, uses the MLLM to generate a sequence of discrete tokens which, in turn, define the segmentation mask.

Embedding Prediction. This paradigm, pioneered by LISA[18], trains an MLLM to output a special token (*e.g.*, `[SEG]`) whose embedding prompts an external, SAM-like decoder [39]. As noted in Tab. 1, this approach, refined by subsequent works [13, 35, 40, 44, 48], yields an efficient $\mathcal{O}(1)$ mask generation step, as the MLLM only needs to generate a single special token to initiate the entire process. However, this design introduces significant drawbacks. The mandatory external module means these methods are not mask decoder-free, requiring structural modifications to the MLLM architecture. Furthermore, training this decoder violates the principle of token-only supervision by necessitating a pixel-level loss, which often compromises the MLLM's general dialogue abilities.

Next-token Prediction. This paradigm avoids objective conflicts by representing the mask as a sequence of discrete tokens. Early works like VisionLLM [42] generated sparse polygon coordinates, a natively decoder-free design. However, this approach proved unreliable due to error accumulation. To improve robustness, methods like Seg-Zero [30] and SegAgent [56] use CoT reasoning to plan keypoints before calling an external SAM decoder. A different strategy, proposed by Text4Seg [19], generates patch-wise classifications to remain decoder-free. Despite these advances, the entire paradigm is fundamentally bottlenecked by its autoregressive nature. As shown in Tab. 1, the number of generation steps scales linearly with the sequence length, making it prohibitively slow for the long outputs required for high-quality segmentation.

3. Method

In this section, we introduce *STAMP*, our framework for realizing the *All-mask Prediction* paradigm. The key challenge is determining *when* to generate the mask and *where* to place its tokens. As depicted in Fig. 3, our solution is a two-phase architecture. In Phase 1 (Dialogue Generation), the MLLM operates autoregressively to generate a textual response, learning to emit a special `<SEG>` to-



Phase 1: Dialogue Generation

Phase 2: All-mask Generation

Figure 3. **The *STAMP* Pipeline. Phase 1 (Dialogue Generation):** The MLLM autoregressively generates a conversational response, emitting a special <SEG> token to trigger mask generation. **Phase 2 (All-mask Generation):** Triggered by the <SEG> token, a sequence of [MASK] placeholders, each fused with its corresponding image patch feature, is processed. A single non-autoregressive forward pass with hybrid attention then predicts all mask tokens simultaneously.

ken. This single token elegantly resolves the core challenge: its presence signals *when* to segment, while its position dictates *where* the mask placeholders will be inserted. The emission of <SEG> triggers Phase 2 (All-mask Generation), which is highly efficient as it leverages the cached activations from the initial phase. In a single, non-autoregressive forward pass, it predicts a classification (*e.g.*, foreground/background) for each image patch based on a sequence of [MASK] placeholders fused with their respective patch features. The complete pipeline is detailed in Algorithm 1, and we elaborate on each phase below.

3.1. Phase 1: Dialogue Generation

In Phase 1, the model generates a contextual textual response and identifies segmentation targets. Given an image I and an instruction T, a ViT first extracts patch features $\mathbf{F_p} \in \mathbb{R}^{N \times D}$, which are then prepended to the text embeddings. The MLLM processes this combined input to autoregressively generate a response R. During this process, the model can emit a special <SEG> token from its vocabulary. Critically, unlike methods such as LISA [18] that tie the [SEG] token's embedding to an external decoder, our <SEG> token is a standard vocabulary item. It functions purely as a learned, in-vocabulary signal that triggers Phase 2, as depicted in Fig. 3 (left).

Caches $\mathcal{C}$. To ensure an efficient transition, intermediate representations from Phase 1 are cached. For each <SEG> token, we define a context-specific tuple $(\text{hist}_i, \text{cache}_i)$, where hist_i is the dialogue history leading to the token, and cache_i stores the pre-computed key-value (KV) states for all preceding tokens. Preserving these states allows *STAMP* to avoid redundant computation when initiating mask prediction for each target.

Algorithm 1: *STAMP* Inference Pipeline

Require: Image I, Instruction T
Ensure : Textual response R, Set of masks $\mathcal{M}$

Phase 1: Dialogue Generation
$\mathbf{F_p} \leftarrow \text{ViT}(I)$
$R, \mathcal{C} \leftarrow \text{GenerateAutoregressive}(T, \mathbf{F_p})$ ▷ Generate response and capture caches
Phase 2: All-mask Generation
$\mathcal{M} \leftarrow \emptyset$
for *each* $(hist_i, cache_i)$ *in* $\mathcal{C}$ **do**
  $\mathbf{E}_{\text{mask}} \leftarrow \text{Embed}(\texttt{[MASK]}_{1..N}) + \mathbf{F_p}$ ▷ Create visually-augmented mask tokens
  $\mathbf{S}_{\text{in}} \leftarrow \text{Concat}(\text{Embed}(\text{hist}_i), \mathbf{E}_{\text{mask}})$
  $\mathbf{A}_{\text{hyb}} \leftarrow \text{ConstructHybridMask}(\text{len}(\text{hist}_i), N)$
  $\mathbf{Z}_{\text{mask}} \leftarrow \text{MLLM.forward}(\mathbf{S}_{\text{in}}, \text{cache} = \text{cache}_i, \text{attn\_mask} = \mathbf{A}_{\text{hyb}})$
  $\text{logits} \leftarrow \text{Classifier}(\mathbf{Z}_{\text{mask}})$
  $\text{mask}_i \leftarrow \text{argmax}(\text{logits})$
  $\text{mask}_i \leftarrow \text{Refine}(\text{mask}_i, \text{SAM})$ ▷ Optional
  $\mathcal{M} \leftarrow \mathcal{M} \cup \{\text{mask}_i\}$
return $R, \mathcal{M}$

3.2. Phase 2: All-mask Generation

Upon the emission of a <SEG> token, Phase 2 generates the entire mask in a single, non-autoregressive forward pass. This begins by preparing a specialized input sequence $\mathbf{S}_{\text{in}}$. We take the dialogue history preceding the <SEG> token

and append N `[MASK]` placeholders, one for each image patch. To provide spatial relationship, the initial embedding of each `[MASK]` token is fused with its corresponding ViT patch feature from $\mathbf{F_p}$, along with the positional embedding that specifies the patch's location in the original image grid. We term these visually-augmented mask embeddings $\mathbf{E}_{\text{mask}}$.

A key component of this phase is our hybrid attention mechanism, which uses a custom attention mask $\mathbf{A}_{\text{hyb}}$ to partition the sequence. For the dialogue history, it enforces standard causal attention. For the $\mathbf{E}_{\text{mask}}$, it enables bi-directional attention, allowing each placeholder to attend to the entire context and all other placeholders. This ensures a holistic and context-aware prediction, as shown in Fig. 3.

With this setup, the MLLM performs a single forward pass. The KV states for the dialogue history are efficiently loaded from the Phase 1 cache. The final hidden states Z_{mask} corresponding to the `[MASK]` input are then fed into a linear classifier to produce foreground (FG) or background (BG) logits for each patch. Finally, this patch-level output can be optionally refined into a high-resolution mask by sampling a single keypoint from the prediction to prompt a frozen SAM decoder [19].

3.3. Training

We train *STAMP* end-to-end with a unified token-level prediction objective comprising two components: a text generation loss ($\mathcal{L}_{\text{text}}$) and a mask prediction loss ($\mathcal{L}_{\text{mask}}$).

The text loss $\mathcal{L}_{\text{text}}$ is the standard cross-entropy loss for autoregressive language modeling. For a ground-truth response $Y = (y_1, \ldots, y_L)$, it maximizes the log-likelihood of each token:

$$\mathcal{L}_{\text{text}} = -\sum_{i=1}^{L} \log P(y_i | y_{<i}, I, T). \tag{1}$$

We extend this token-level supervision to the mask generation phase. The mask loss $\mathcal{L}_{\text{mask}}$ is applied to the output logits of the $\mathbf{E}_{\text{mask}}$. Each token performs a binary classification on its corresponding image patch, predicting whether it belongs to the foreground or background. This is supervised using a combination of Binary Cross-Entropy and Dice loss:

$$\mathcal{L}_{\text{mask}} = \mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{Dice}}. \tag{2}$$

The final training objective is a simple sum of the two losses: $\mathcal{L} = \mathcal{L}_{\text{text}} + \mathcal{L}_{\text{mask}}$. By jointly optimizing both components, *STAMP* learns to treat segmentation not as a separate task, but as an integral part of its generative vocabulary.

4. Experiments

4.1. Common Setup

Implementation Details. We built *STAMP* on the 2B and 7B variants of the Qwen2-VL series [2], which provides inherent support for dynamic input resolutions. For the optional mask refinement stage, we employed the huge (-H) variant of SAM. The number of `[MASK]` placeholders was set dynamically between 1024 and 1280 to correspond with the input image resolution.

Training. All models were trained on NVIDIA H800 GPUs. We used a maximum learning rate of 3e-5, which followed a linear decay schedule after a warm-up phase constituting 3% of the total training steps. The weight decay was set to 0, and gradient norms were clipped at 1.0. Following common practice, the 2B models were fully fine-tuned, while the larger 7B models were trained using LoRA [12] for parameter-efficient fine-tuning.

With these implementation and training details established, we now proceed to a comprehensive evaluation of *STAMP*, which starts from referring and reasoning segmentation tasks, extending to more general dialogue abilities, and concluding with a detailed analysis of inference efficiency and key design choices via ablation studies.

4.2. Referring Expression Segmentation

Setup. We first evaluated *STAMP* on the task of Referring Expression Segmentation (RES) to validate its core segmentation performance. Our evaluation covered both single-object RES, using the standard RefCOCO [16], RefCOCO+ [16], and RefCOCOg [32] datasets, as well as the more challenging multi-object and no-object scenarios presented by gRefCOCO [24].

To ensure a fair and direct comparison with prior work, we strictly adhered to the training data and schedule of Text4Seg [19]. Initially, the model was trained for 3 epochs on a combined dataset of 800k samples, comprising the training splits of RefCLEF [16], RefCOCO, RefCOCO+, and RefCOCOg. Following this, we continued training for an additional 2 epochs exclusively on the 419k samples from the gRefCOCO training split. All testing was conducted using the standard evaluation protocols [18, 19].

Baselines. As shown in Tab. 2, we compare *STAMP* against a comprehensive set of recent and influential works. These include state-of-the-art *Specialised Baselines* (*e.g.*, UNINEXT-L [49]) for context, as well as top methods from the two dominant MLLM paradigms: *Embedding Prediction* (*e.g.*, GSVA [48], READ [35]) and *Token Prediction* (*e.g.*, next-token prediction method Text4Seg [19]). For the multi-/no-object task on gRefCOCO, we also include specially designed models like LAVT [51].

Results on Single-Object RES. As presented in Table 2, our *STAMP*-7B model set a new state of the art, achieved an average cIoU of 80.7 and outperformed all competing methods across all evaluation splits. This marked a significant advance over the previous best, Text4Seg-13B (76.2 cIoU). Notably, even our lightweight *STAMP*-2B variant surpassed this prior art with 79.1 cIoU, highlighting the ef-

Table 2. **Comprehensive comparison on the single-object RES task.** Methods are categorized by paradigm and sorted by performance. Our models are highlighted with a gray background, and the top-performing method is shown in bold. "Mask-Decoder-Free" indicates results without SAM-based post-processing.

Method	LLM	RefCOCO			RefCOCO+			RefCOCOg		Avg.
		val	testA	testB	val	testA	testB	val(U)	test(U)	
Specialised Baselines										
HIPIE (NIPS'24) [43]	BERT	78.3	-	-	66.2	-	-	69.8	-	-
ReLA (CVPR'23) [24]	BERT	73.8	76.5	70.2	66.0	71.0	57.7	65.0	66.0	68.3
PolyFormer-L (CVPR'23) [28]	BERT	76.0	78.3	73.3	69.3	74.6	61.9	69.2	70.2	71.6
UNINEXT-L(CVPR'24) [49]	BERT	80.3	82.6	77.8	70.0	74.9	62.6	73.4	73.7	74.4
Paradigm: Embedding Prediction										
PixelLM (CVPR'24) [40]	Vicuna-7B	73.0	76.5	68.2	66.3	71.7	58.3	69.3	70.5	69.2
LISA (CVPR'24) [18]	Vicuna-7B	74.9	79.1	72.3	65.1	70.8	58.1	67.9	70.6	69.9
LISA (CVPR'24) [18]	Vicuna-13B	76.0	78.8	72.9	65.0	70.2	58.1	69.5	70.5	70.1
GSVA (CVPR'24) [48]	Vicuna-7B	77.2	78.9	73.5	65.9	69.6	59.8	72.7	73.3	71.4
READ (CVPR'25) [35]	Vicuna-7B	78.1	80.2	73.2	68.4	73.7	60.4	70.1	71.4	71.9
GSVA (CVPR'24) [48]	Vicuna-13B	78.2	80.4	74.2	67.4	71.5	60.9	74.2	75.6	72.8
Paradigm: Token Prediction										
Mask-Decoder-Free										
Text4Seg (ICLR'25) [19]	Vicuna-13B	74.1	76.4	72.4	68.5	72.8	63.6	69.1	70.1	70.9
Text4Seg (ICLR'25) [19]	InternLM2.5-7B	74.7	77.4	71.6	68.5	73.6	62.9	70.7	71.6	71.4
STAMP	Qwen2-2B	77.7	79.4	76.1	73.4	76.4	69.7	74.9	75.1	75.3
STAMP	Qwen2-7B	**78.1**	**79.2**	**76.8**	**74.7**	**77.6**	**70.9**	**75.7**	**76.2**	**76.2**
With Mask Decoder										
Seg-Zero (arXiv'25) [30]	Qwen2.5-3B	-	79.3	-	-	73.7	-	-	71.5	-
Seg-Zero (arXiv'25) [30]	Qwen2.5-7B	-	80.3	-	-	76.2	-	-	72.6	-
SegLLM (ICLR'25) [44]	Vicuna-7B	80.2	81.5	75.4	70.3	73.0	62.5	72.6	73.6	73.6
Text4Seg (ICLR'25) [19]	Vicuna-7B	79.3	81.9	76.2	72.1	77.6	66.1	72.1	73.9	74.9
Text4Seg (ICLR'25) [19]	InternLM2.5-7B	79.2	81.7	75.6	72.8	77.9	66.5	74.0	75.3	75.4
SegAgent (CVPR'25) [56]	Qwen-7B	79.7	81.4	76.6	72.5	75.8	66.9	75.1	75.2	75.4
Text4Seg (ICLR'25) [19]	Vicuna-13B	80.2	82.7	77.3	73.7	78.6	67.6	74.0	75.1	76.2
STAMP	Qwen2-2B	81.9	83.7	79.5	77.1	80.5	72.7	78.5	78.8	79.1
STAMP	Qwen2-7B	**83.1**	**84.5**	**80.8**	**79.4**	**82.8**	**74.6**	**79.9**	**80.4**	**80.7**

ficiency of our approach. The core strength of our architecture is further evidenced by an unrefined version of *STAMP*-7B, which matched the previous state of the art without any SAM-based post-processing. Crucially, these performance gains were attained even though competitors like Seg-Zero and Text4Seg leveraged more powerful LLMs (Qwen2.5-7B and InternLM2.5-7B vs. our Qwen2), confirming that our improvements originate from a superior architectural design rather than simply a stronger backbone.

Results on Multi/No-Object RES. As presented in Table 3, *STAMP* extended its superior performance to the more challenging gRefCOCO benchmark. Echoing our previous findings, our *STAMP*-7B model without SAM-based refinement already surpassed the strongest fully-equipped baseline, Text4Seg (71.8 vs. 71.5). Our best-performing model established a new state of the art with a score of 74.8, exceeding the strongest competitor by a significant margin of 3.3%. This robust performance underscores the effectiveness and reliability of our architecture for complex segmentation challenges involving multiple or absent objects.

4.3. Reasoning Segmentation

Setup. To evaluate the model's complex reasoning capabilities, we tested *STAMP* on the reasoning segmentation benchmark, following the protocol of LISA [18]. We further fine-tuned our single-object RES model (2B) for one epoch on a mixture of the RefCOCO* datasets and the 239 training images from the ReasonSeg training split [18]. No additional training data was used. Evaluation adhered to the established benchmark protocol from LISA. We include OVSeg [21] as a traditional baseline to highlight the difficulty of the ReasonSeg benchmark, as its challenges can typically only be addressed by MLLM-based models.

Results. As shown in Table 4, our *STAMP*-2B model outperformed all existing methods, achieving an average score of 63.2. This result significantly surpassed the previous leading method, READ [35] (61.1), which utilized a larger training dataset [47]. Notably, this superior performance was achieved without explicit chain-of-thought reasoning and with a substantially smaller backbone (Qwen2-VL-2B) compared to competitors like Seg-Zero (Qwen2.5-VL-7B) and Text4Seg (Qwen2-VL-7B). This provides strong evi-

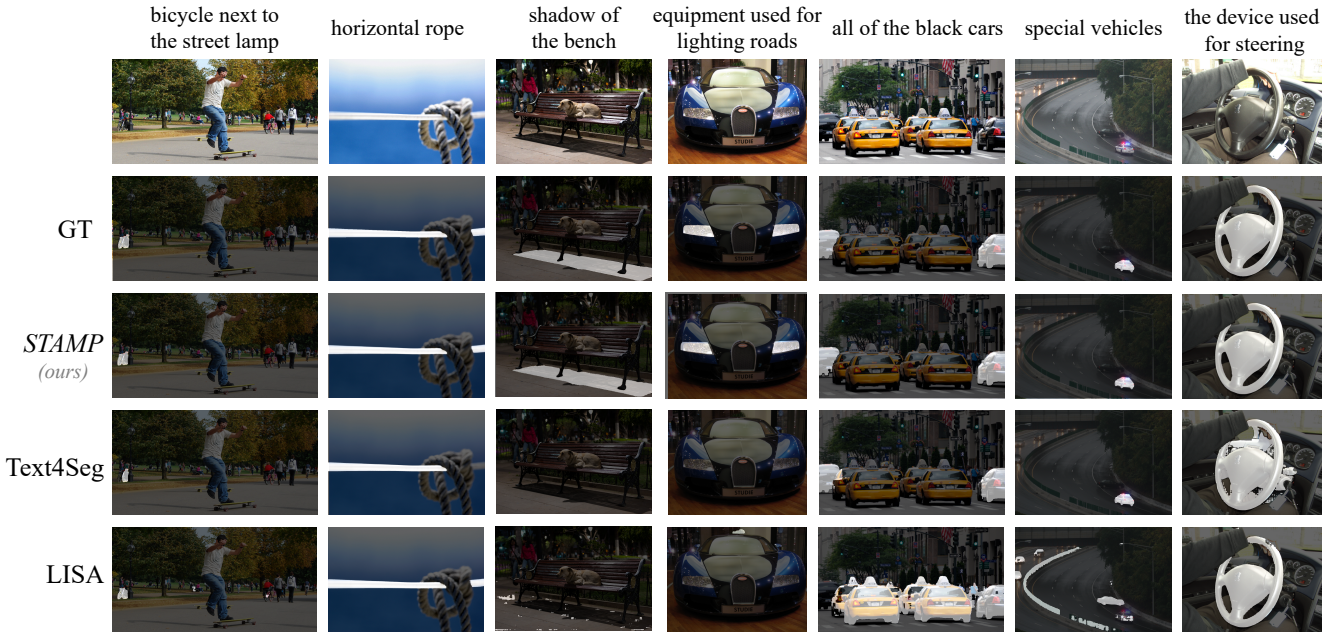


Figure 4. **Showcase of *STAMP* against competing methods.**

Table 3. **Comparison on the multi/no object RES task.** Our models are highlighted with a gray background, and the top-performing method is shown in bold. "Mask-Decoder-Free" indicates results without SAM-based post-processing.

Method	LLM	Val Set		Test Set A		Test Set B		Avg.
		gIoU	cIoU	gIoU	cIoU	gIoU	cIoU	
Mask-Decoder-Free								
Text4Seg	InternLM2.5-7B	70.0	66.1	69.4	70.9	63.1	64.1	67.3
Text4Seg	Vicuna-13B	70.3	66.9	69.8	71.4	63.8	64.4	67.8
STAMP	Qwen2-2B	73.9	70.3	73.6	73.7	67.5	68.1	71.2
STAMP	Qwen2-7B	**74.4**	**70.9**	**73.8**	**74.7**	**68.1**	**69.1**	**71.8**
With Mask Decoder								
LAVT	BERT	58.4	57.6	65.9	65.3	55.8	55.0	59.7
LISA	Vicuna-7B	61.6	61.8	66.3	68.5	58.8	60.6	62.9
ReLA	BERT	63.6	62.4	70.0	69.3	61.0	59.9	64.4
LISA	Vicuna-13B	63.5	63.0	68.2	69.7	61.8	62.2	64.7
GSVA	Vicuna-7B	66.5	63.3	71.1	69.9	62.2	60.5	65.6
Text4Seg	InternLM2.5-7B	74.4	69.1	75.1	73.8	67.3	66.6	71.1
Text4Seg	Vicuna-13B	74.8	69.8	75.1	74.3	68.0	67.1	71.5
STAMP	Qwen2-2B	76.4	72.2	76.5	75.7	70.0	69.8	73.4
STAMP	Qwen2-7B	**77.6**	**73.6**	**77.6**	**77.2**	**71.4**	**71.6**	**74.8**

dence that our all-mask prediction paradigm not only enhances segmentation but also more effectively leverages the MLLM's intrinsic reasoning capabilities. The qualitative results in Fig. 4 further substantiate this, showcasing accurate segmentation from complex instructions.

4.4. Visual Understanding

Setup. To verify that *STAMP* retains its general dialogue abilities, we evaluated its comprehensive performance on general vision-language tasks. We strictly followed the protocol established by Text4Seg [19]. Specifically, our model was fine-tuned from a base model for two epochs on a composite dataset comprising 800k samples from the RefCOCO* suite and 665k vision-language instruction samples from LLaVA. For evaluation, we used the same referring segmentation benchmarks as Text4Seg to ensure a direct comparison. To assess visual understanding, we used a diverse range of VQA datasets, including MMMU [52], MMBench [29], MMStar [5], ScienceQA [31] and TextVQA [41]. Baseline methods are built upon the LLaVA; *STAMP* utilizes the Qwen backbone.

Table 4. **Comparison on the reasoning segmentation task.**

Method	LLM	Val		Test		Avg.
		gIoU	cIoU	gIoU	cIoU	
OVSeg	Vicuna-7B	28.5	18.6	26.1	20.8	23.5
LISA	Vicuna-7B	53.6	52.3	48.7	48.8	50.9
SegLLM	Vicuna-7B	57.2	54.3	52.4	48.4	53.1
Text4Seg	Qwen2-7B	59.1	49.5	57.1	52.1	54.5
Seg-Zero	Qwen2.5-7B	62.6	62.0	57.5	52.0	58.5
READ	Vicuna-7B	59.8	**67.6**	58.5	58.6	61.1
STAMP	Qwen2-2B	**65.1**	63.9	**62.7**	**60.9**	**63.2**

Results. As shown in Table 5, training on the mixed dataset allowed *STAMP* to significantly outperform all baselines in both general-purpose visual understanding and segmentation simultaneously. Crucially, its visual understanding performance remained on par with the Qwen2-VL-2B baseline trained exclusively on VQA data. This demonstrates that our all-mask paradigm does not degrade the model's gen-

Table 5. **Comparison of joint visual understanding and segmentation performance.** We include LLaVA-1.5-7B and Qwen2-VL-2B fine-tuned on the LLaVA-665k set (VQA) as reference models. The training data consists of segmentation data (Seg.), the VQA set, or a mixture (Mix). Our models are highlighted with a gray background; Performance of reference models is shown in gray text. The results show that *STAMP* maintains its general dialogue abilities after mixed training, while its segmentation performance is further improved.

Methods	Training Data	VQA					RES (val)		
		MMMU	MMBench	MMStar	ScienceQA	TextVQA	RefC	RefC+	RefCg
LLaVA-1.5-7B	VQA	35.7	66.5	33.1	68.4	55.0	n.a.	n.a.	n.a.
Qwen2-VL-2B	VQA	**38.3**	66.5	42.1	70.2	**70.7**	n.a.	n.a.	n.a.
LISA-7B	Mix	0	0	0	0	0	74.9	65.1	67.9
READ-7B	Mix	1.1	0	14.4	23.2	22.6	78.1	68.4	70.1
Text4Seg-7B	Mix	34.0	54.8	33.4	68.1	55.0	79.3	72.1	72.1
STAMP-2B	Seg.	n.a.	n.a.	n.a.	n.a.	n.a.	81.9	77.1	78.5
STAMP-2B	Mix	37.8	**68.7**	**42.4**	**72.6**	69.7	**82.2**	**77.3**	**79.0**

eral dialogue abilities, effectively addressing the segmentation trilemma. Interestingly, this co-training strategy also boosted segmentation performance compared to training on segmentation-only data. This suggests that the all-mask prediction paradigm effectively allows higher-level understanding to enhance its segmentation ability, demonstrating a promising potential for further scaling.

4.5. Ablation Study

In this section, we conducted ablation studies to validate our architectural components and analyze the model's efficiency. We first ablated our key designs and evaluated the model's support for dynamic resolutions. We then benchmarked its inference speed against other paradigms to highlight high performance without sacrificing efficiency.

Ablation of Core Components. As shown in Tab. 6, our two primary architectural designs are critical for performance. Removing either the visually-enhanced mask embedding ($\mathbf{E}_{\text{mask}}$) or the hybrid attention mechanism ($\mathbf{A}_{\text{hyb}}$) led to a distinct degradation in segmentation accuracy, confirming the contribution of each component.

Table 6. **Ablation study.** We report performance on the single-object RES task and the corresponding inference time.

Method	Input Size	cIoU	Time (s)
STAMP-2B	896 × 896	79.1	1.3
w/o $\mathbf{A}_{\text{hyb}}$	896 × 896	76.2	1.3
w/o $\mathbf{E}_{\text{mask}}$	896 × 896	73.3	1.3
w/o SAM	896 × 896	75.3	0.9
STAMP-2B	726 × 726	78.6	1.1
STAMP-2B	504 × 504	77.1	0.9
STAMP-7B	896 × 896	80.7	2.4

Dynamic Resolution Support. The results in Tab. 6 also demonstrate our model's adaptability. Although trained at an 896 resolution, *STAMP* could process lower-resolution inputs, trading a marginal drop in performance for a substantial gain in inference speed. This versatility makes it suitable for diverse deployment scenarios.

Efficiency Comparison. As shown in Fig. 5, *STAMP* excels in both segmentation performance and speed. The inference speed is on par with the embedding approaches, clearly distinguishing it from slower next-token approaches.

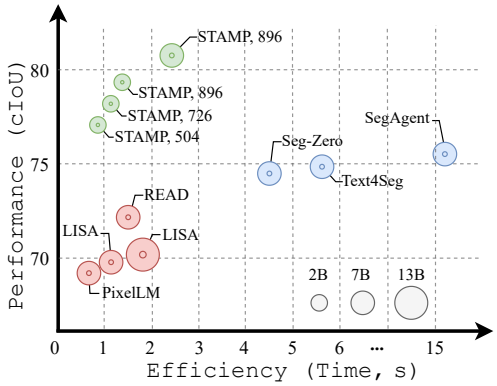


Figure 5. **Efficiency comparison.** Methods from the same paradigm are grouped by color. Numbers following the model names denote the input resolution.

5. Conclusion

In this work, we addressed the fundamental trilemma in MLLM-based segmentation, where models have been forced to trade off between general dialogue abilities, segmentation performance, and inference speed. We introduced all-mask prediction, a novel paradigm that resolves this conflict by decoupling autoregressive dialogue generation from non-autoregressive mask prediction. Our implementation, *STAMP*, is the first effective realization of this paradigm. Through extensive experiments, we demonstrated that *STAMP* not only establishes a new state of the art across multiple challenging benchmarks, including referring and reasoning segmentation, but also achieves this with inference speeds comparable to the efficient prior paradigms, all while fully preserving its general dialogue abilities. By successfully resolving the segmentation trilemma, our work removes the barrier to developing more practical, efficient, and truly unified visual models.

References

[1] 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*, 2023. 2

[2] 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. 2, 5, 1

[3] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. COCO-Stuff: Thing and stuff classes in context. In *CVPR*, 2018. 4

[4] Gongwei Chen, Leyang Shen, Rui Shao, Xiang Deng, and Liqiang Nie. LION: Empowering multimodal large language model with dual-level visual knowledge. In *CVPR*, 2024. 1

[5] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *Advances in Neural Information Processing Systems*, 37:27056–27087, 2024. 7

[6] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1971–1978, 2014. 4

[7] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In *NIPS*, 2023. 2

[8] Daan De Geus, Panagiotis Meletis, Chenyang Lu, Xiaoxiao Wen, and Gijs Dubbelman. Part-aware panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5485–5494, 2021. 4

[9] Google. Google Gemini 2.5 Pro, 2025. https://deepmind.google/technologies/gemini/pro/. 2

[10] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5356–5364, 2019. 4

[11] Ju He, Shuo Yang, Shaokang Yang, Adam Kortylewski, Xiaoding Yuan, Jie-Neng Chen, Shuai Liu, Cheng Yang, Qihang Yu, and Alan Yuille. Partimagenet: A large, high-quality dataset of parts. In *European Conference on Computer Vision*, pages 128–145. Springer, 2022. 4

[12] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022. 5, 1

[13] Donggon Jang, Yucheol Cho, Suin Lee, Taehyeon Kim, and Daeshik Kim. MMR: A large-scale benchmark dataset for multi-target and multi-granularity reasoning segmentation. In *ICLR*, 2025. 1, 3

[14] Qing Jiang, Junan Huo, Xingyu Chen, Yuda Xiong, Zhaoyang Zeng, Yihao Chen, Tianhe Ren, Junzhi Yu, and Lei Zhang. Detect anything via next point prediction. *arXiv preprint arXiv:2510.12798*, 2025. 1

[15] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 2

[16] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. ReferItGame: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014. 5, 3, 4

[17] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017. 4

[18] 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. 1, 2, 3, 4, 5, 6

[19] 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*, 2025. 1, 2, 3, 5, 6, 7, 4

[20] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 1, 2

[21] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. In *CVPR*, 2023. 6

[22] Jun Hao Liew, Scott Cohen, Brian Price, Long Mai, and Jiashi Feng. Deep interactive thin object selection. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 305–314, 2021. 4

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 4

[24] Chang Liu, Henghui Ding, and Xudong Jiang. GRES: Generalized referring expression segmentation. In *CVPR*, 2023. 5, 6, 3, 4

[25] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023. 2, 3, 4

[26] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, 2024. 2

[27] Jiazhen Liu and Long Chen. Segmentation as a plug-and-play capability for frozen multimodal llms. *arXiv preprint arXiv:2510.16785*, 2025. 2

[28] Jiang Liu, Hui Ding, Zhaowei Cai, Yuting Zhang, Ravi Kumar Satzoda, Vijay Mahadevan, and R Manmatha. Poly-

Former: Referring image segmentation as sequential polygon generation. In *CVPR*, 2023. 6

[29] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. MMBench: Is your multi-modal model an all-around player? In *ECCV*, 2024. 7

[30] Yuqi Liu, Bohao Peng, Zhisheng Zhong, Zihao Yue, Fanbin Lu, Bei Yu, and Jiaya Jia. Seg-Zero: Reasoning-chain guided segmentation via cognitive reinforcement. *arXiv preprint arXiv:2503.06520*, 2025. 1, 2, 3, 6, 4

[31] Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *NeurIPS*, 2022. 7

[32] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, 2016. 5, 3, 4

[33] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE international conference on computer vision*, pages 4990–4999, 2017. 4

[34] OpenAI. Hello GPT-4o, 2024. https://openai.com/index/hello-gpt-4o/. 2

[35] Rui Qian, Xin Yin, and Dejing Dou. Reasoning to attend: Try to understand how <SEG> token works. In *CVPR*, 2025. 1, 2, 3, 5, 6, 4

[36] Xuebin Qin, Hang Dai, Xiaobin Hu, Deng-Ping Fan, Ling Shao, and Luc Van Gool. Highly accurate dichotomous image segmentation. In *European Conference on Computer Vision*, pages 38–56. Springer, 2022. 4

[37] 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 *CVPR*, 2023. 4

[38] 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. 2

[39] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. SAM 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 3

[40] 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. 1, 2, 3, 6, 4

[41] Amanpreet Singh, Vivek Natarjan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *CVPR*, 2019. 7

[42] Wenhai Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, et al. VisionLLM: Large language model is also an open-ended decoder for vision-centric tasks. *Advances in Neural Information Processing Systems*, 36:61501–61513, 2023. 2, 3

[43] Xudong Wang, Shufan Li, Konstantinos Kallidromitis, Yusuke Kato, Kazuki Kozuka, and Trevor Darrell. Hierarchical open-vocabulary universal image segmentation. *Advances in Neural Information Processing Systems*, 36: 21429–21453, 2023. 6

[44] XuDong Wang, Shaolun Zhang, Shufan Li, Kehan Li, Konstantinos Kallidromitis, Yusuke Kato, Kazuki Kozuka, and Trevor Darrell. SegLLM: Multi-round reasoning segmentation with large language models. In *ICLR*, 2025. 3, 6, 4

[45] Jianzong Wu, Xiangtai Li, Xia Li, Henghui Ding, Yunhai Tong, and Dacheng Tao. Toward robust referring image segmentation. *IEEE Transactions on Image Processing*, 33: 1782–1794, 2024. 4

[46] Jiannan Wu, Muyan Zhong, Sen Xing, Zeqiang Lai, Zhaoyang Liu, Zhe Chen, Wenhai Wang, Xizhou Zhu, Lewei Lu, Tong Lu, et al. VisionLLM v2: An end-to-end generalist multimodal large language model for hundreds of vision-language tasks. *Advances in Neural Information Processing Systems*, 37:69925–69975, 2024. 1, 2, 3

[47] Tsung-Han Wu, Giscard Biamby, David Chan, Lisa Dunlap, Ritwik Gupta, Xudong Wang, Joseph E Gonzalez, and Trevor Darrell. See say and segment: Teaching LMMs to overcome false premises. In *CVPR*, 2024. 2, 6, 4

[48] Zhuofan Xia, Dongchen Han, Yizeng Han, Xuran Pan, Shiji Song, and Gao Huang. GSVA: Generalized segmentation via multimodal large language models. In *CVPR*, 2024. 1, 2, 3, 5, 6, 4

[49] Bin Yan, Yi Jiang, Jiannan Wu, Dong Wang, Ping Luo, Zehuan Yuan, and Huchuan Lu. Universal instance perception as object discovery and retrieval. In *CVPR*, 2023. 5, 6

[50] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. 1, 2

[51] Zhao Yang, Jiaqi Wang, Yansong Tang, Kai Chen, Hengshuang Zhao, and Philip HS Torr. LAVT: Language-aware vision transformer for referring image segmentation. In *CVPR*, 2022. 5

[52] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *CVPR*, 2024. 7

[53] Jiarui Zhang, Mahyar Khayatkhoei, Prateek Chhikara, and Filip Ilievski. MLLMs know where to look: Training-free perception of small visual details with multimodal llms. In *ICLR*, 2025. 1

[54] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ADE20K dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019. 4

[55] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie

Shao, et al. InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025. 1, 2

[56] Muzhi Zhu, Yuzhuo Tian, Hao Chen, Chunluan Zhou, Qingpei Guo, Yang Liu, Ming Yang, and Chunhua Shen. SegAgent: Exploring pixel understanding capabilities in mllms by imitating human annotator trajectories. In *CVPR*, 2025. 1, 2, 3, 6, 4

Better, Stronger, Faster: Tackling the Trilemma in MLLM-based Segmentation with Simultaneous Textual Mask Prediction

Supplementary Material

In the supplementary materials, we report:

- **Implementation Details.** We provide comprehensive implementation details, including the training parameters and datasets for each experimental stage, the methodology for constructing instruction-following data, and a comparison of datasets used by our method versus baseline models (Sec. S1).
- **Additional Experiments.** We report detailed analyses of our model's efficiency, including inference time, GPU memory consumption (Sec. S2).
- **More Qualitative Showcases.** We present detailed qualitative showcases of *STAMP*, covering standard referring segmentation(in the main text), reasoning segmentation, visual question answering and advanced capabilities such as muti-round dialogue, multi-round segmentation and unified dialogue-segmentation examples (Sec. S3).
- **Code and Documentation.** We provide the source code for training and execution, along with a user guide and setup instructions (Sec. S4).

S1. Implementation Details

In this section, we detail our training strategies, data construction methodology, and the experimental setup for a fair comparison against baseline methods.

S1.1. Training Strategy

The specific training hyperparameters for the experiments in each section are detailed in Tab. S1, aligning with standard practices. Our experiments utilize the Qwen2-VL-2B and Qwen2-VL-7B models [2]. We maintain consistent hyperparameters across experiments wherever possible to isolate the performance gains from our proposed methods, rather than from extensive hyperparameter tuning.

LoRA Training. We employ LoRA [12] for training our 7B model. This decision aligns with the training strategies of baseline methods such as LISA [18] and Text4Seg [19], ensuring a direct and fair comparison. In contrast, our 2B model is fully fine-tuned, as its smaller size renders the efficiency benefits of LoRA [12] less significant.

S1.2. Training Dataset

For the training data in each sub-experiment, we strictly adhere to the settings of our baselines, Text4Seg [19] and LISA [18], conducting experiments on the same minimal datasets (detailed in Tab. S2). This approach is intentional: we aim to demonstrate that our performance gains stem from our paradigm innovation, rather than from simply using more data. Consequently, our current training is focused on single-object grounding and segmentation tasks. While many contemporary works train on larger, mixed datasets, we are confident that *STAMP* can be scaled to these data sources in the future, presenting a clear path to even greater performance leads.

S1.3. Instruction Construction

The instruction used for training *STAMP* is included in Prompt S1 and S2.

```
Question:
o "Please segment only the [class_name] in the image.",
o "Can you segment the [class_name] in the image?",
o "Where is the [class_name] in this picture? Please respond with segmentation mask.",
o "Where is [class_name] in this image? Please output segmentation mask.",
o "Could you provide the segmentation mask for [class_name] in this image?",
o "Please segment the image and highlight [class_name]."

Answer:
* "Sure, here is the segmentation mask for [class_name] <|seg|>:",
* "Here is the segmentation mask focusing on the [class_name] <|seg|>:",
* "Here is the segmentation mask highlighting the [class_name] <|seg|>:",
* "The segmentation map for [class_name] <|seg|> is:",
* "The segmentation mask for [class_name] <|seg|> is shown below:",
* "Sure, Here's the segmentation mask for [class_name] <|seg|>:",
* "Sure, the segmented output for [class_name] <|seg|> is:",
* "Certainly, the segmentation map for [class_name] <|seg|> is:",
* "Certainly, here is the segmentation mask for [class_name]:",
* "The segmentation mask for [class_name] <|seg|> is shown below:"
```

Prompt S1. Instructions for RES task.

```
% --- 1. Segmentation Task ---
Question:
    o "Can you segment the [class_name] in
       this image?"
    o "Please segment the [class_name] in
       this image."
    o "What is [class_name] in this image?
       Please output segmentation mask."
    o "{sent} Please respond with
       segmentation mask."

Answer:
    * "It is <|seg|>."
    * "Sure, <|seg|>."
    * "Sure, the segmentation result is <|
       seg|>."
    * "<|seg|>."

% --- 2. Explanation Task ---
Question:
o "Explain [class_name] in this image."
o "Could you describe the [class_name] in
    the image?"
o "Tell me more about the [class_name]."

Answer:
* "[reasoning]"
```

Prompt S2. Instructions for Reasoning Segmentation task.

S2. Additional Experiments.

In this section, we list the specific design and detailed test results for evaluating *STAMP*'s efficiency.

S2.1. Implementation Setting.

For the inference latency discussed in §4.5, all methods were rigorously evaluated on a single A800 80GB GPU. Crucially, tests were conducted in isolation, ensuring only one program ran on the card during each measurement. Furthermore, all methods employed the identical test case, which is illustrated in Fig. S1.

S2.2. Results.

As shown in Tab. S3, *STAMP* excels in both accuracy and speed. *STAMP*-7B achieves the highest cIoU of 80.7, significantly outperforming SegAgent-7B(75.4) [56] and READ-7B(71.9) [35]. Despite high input resolution (896×896), *STAMP* remains remarkably fast: *STAMP*-2B variants (0.9s–1.3s) are comparable to embedding-based methods (e.g., LISA, PixelLM [40]) and far faster than next-token approaches like SegAgent (15.9s). Ablation studies further confirm this efficiency; even without the mask decoder, *STAMP*-2B maintains competitive performance (75.3 cIoU) with minimal VRAM usage (10.8 GB), demonstrating superior resource efficiency.

Prompt: Please segment the trees in the image.

Figure S1. **Inference time comparison example**. This example illustrates a representative test case used to measure the inference latency across all compared methods. The consistent input allows for a fair comparison of the computational efficiency between the models discussed in Section 4.5.

S3. More Qualitative Showcases.

To comprehensively validate the capabilities of our *STAMP* model, we designed a suite of test cases targeting its performance across multiple areas, including reasoning segmentation, VQA (Visual Question Answering), multi-round conversation, multi-round segmentation, and dialogue-segmentation. For all subsequent cases, the *STAMP* model used is the *STAMP*-2B variant. This model is configured with the SAM decoder and processes inputs at an 896×896 resolution.

S3.1. Reasoning Segmentation

Reasoning Segmentation requires transcending simple recognition to integrate functional and abstract reasoning. As shown in Fig. S2, *STAMP* excels in this task by explicitly decoupling reasoning from segmentation. It first articulates inferred properties (e.g., nutritional role) before executing precise segmentation. This two-step process confirms *STAMP*'s ability to ground high-level semantic understanding directly onto pixel space, offering a distinct advantage over models that treat reasoning merely as a textual side effect.

S3.2. Visual Question Answering

Crucially, *STAMP* integrates vision-language capabilities without sacrificing the core language proficiency of the underlying MLLM. This contrasts with methods like LISA that often compromise language performance for segmen-

Table S1. **Comprehensive hyperparameter settings for all SFT stages.** We detail the training configurations for our main experiments: Referring Expression Segmentation (§4.2), Reasoning Segmentation (§4.3), and Visual Understanding (§4.4). RefCOCO* means RefCOCO [16], RefCOCO+ [16], RefCOCOg [32] and RefCLEF [16].

Category	Hyperparameter	Referring Seg.		Reasoning Seg.	Dialogue & Seg.
		Qwen2-2B	Qwen2-7B	Qwen2-2B	Qwen2-2B
Training Details	Section §	§4.2	§4.2	§4.3	§4.4
	Datasets	RefCOCO*, gRefCOCO [24] (stage 2)	RefCOCO*, gRefCOCO [24] (stage 2)	RefCOCO*, ReasonSeg [18]	RefCOCO*, LLaVA-665k [25]
	Epochs	3 + 2	3 + 2	1	2
	Batch Size	96	96	96	96
	ZeRO Stage	2	2	2	2
Optimizer (AdamW)	Learning Rate	3×10^{-5}	2×10^{-5}	3×10^{-5}	3×10^{-5}
	Weight Decay	0	0	0	0
	Betas (β_1, β_2)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)
	Grad Norm Clip	1.0	1.0	1.0	1.0
	Scheduler	Linear	Linear	Linear	Linear
	Warmup Ratio	0.03	0.03	0.03	0.03
LoRA Configuration	Method	Full fine-tuning (LoRA not applied)	LoRA	Full fine-tuning (LoRA not applied)	Full fine-tuning (LoRA not applied)
	Rank		64		
	Alpha (α)		128		
	Dropout		0.05		
	Target Modules		All		
Model Parameters	Total Parameters	2B	7B	2B	2B

tation. As shown in Fig. S3, *STAMP* remains robust across diverse VQA tasks, ranging from abstract reasoning to fine-grained grounding. This demonstrates that *STAMP* retains superior language mastery, serving as a versatile multi-modal assistant beyond its primary segmentation role.

S3.3. Muti-round Dialogue

The multi-round dialogue scenario rigorously tests *STAMP*'s contextual retention and linguistic integrity. As illustrated in Fig. S4, the model's seamless performance across a six-round exchange confirms its capability as a unified conversational agent. Specifically, *STAMP* demonstrates zero-shot immunity to common MLLM failures: it successfully grounds complex temporary aliases and maintains resilience against conversational distractions. This confirms that *STAMP* preserves the critical memory and reasoning capabilities essential for conversational AI, despite the integration of segmentation.

S3.4. Muti-round Segmentation

The Multi-round Dialogue scenario evaluates *STAMP*'s contextual retention and linguistic integrity. As shown in Fig. S4, its seamless performance across a six-round exchange confirms its capability as a unified conversational agent. Specifically, *STAMP* demonstrates zero-shot robustness against common failures, successfully grounding temporary aliases and resisting conversational distractions. This sustained coherence validates that integrating segmentation preserves the critical memory and reasoning foundations required for advanced conversational AI.

S3.5. Dialogue Segmentation Interaction

The Dialogue Segmentation Interaction scenario validates *STAMP* as a unified multimodal assistant. As shown in Fig. S6, the model seamlessly transitions between linguistic responses and pixel-level masks within the same flow. A key highlight is the memory-based segmentation in Round 6, where *STAMP* grounds an object (the milk bottle) based solely on context established in Rounds 4-5. This confirms *STAMP* utilizes dialogue history as a dynamic context for visual tasks.

S4. Code and Documentation.

We provide a detailed *README.md* as the primary reference for the codebase, covering essential aspects including environment setup, detailed usage protocols for training and evaluation, and an explanation of the modular project structure. Due to upload size constraints, raw image datasets are not included in this submission; however, the documentation strictly defines the expected data hierarchy. To facilitate reproducibility, we also include specific scripts designed to generate all required annotations from standard data sources.

Table S2. **Training data comparison across CPT and SFT stages.** This table contrasts the datasets used during the Continued Pre-Training (CPT) stage and the Supervised Fine-Tuning (SFT) stage for Referring Expression Segmentation. † Some methods incorporate RefCLEF during CPT, while others (like Ours, Text4Seg, *STAMP*) use it in the SFT stage.

Method	**Continued Pre-Training (CPT) Datasets**	**Referring Seg. SFT Datasets**
LISA [18]	ADE20K [54], COCO-Stuff [3], PACO-LVIS [37], PartImageNet [11], PASCAL-Part [6], RefCLEF† [16], RefCOCO [16], RefCOCO+ [16], RefCOCOg [32], LLaVA-665k [25]	RefCOCO [16], RefCOCO+ [16], RefCOCOg [32]
PixelLM [40]	ADE20K [54], COCO-Stuff [3], PACO-LVIS [37], RefCLEF† [16], RefCOCO [16], RefCOCO+ [16], RefCOCOg [32], LLAVA-150k [25], MUSE [40]	**None**
GSVA [48]	ADE20K [54], COCO-Stuff [3], PACO-LVIS [37], Mapillary Vistas [33], PASCAL-Part [6], RefCLEF† [16], RefCOCO [16], RefCOCO+ [16], RefCOCOg [32], gRefCOCO [24], LLaVA-Instruct-150K [25], ReasonSeg [18]	RefCOCO [16], RefCOCO+ [16], RefCOCOg [32]
READ [35]	ADE20K [54], COCO-Stuff [3], PACO-LVIS [37], PASCAL-Part [6], RefCLEF† [16], RefCOCO [16], RefCOCO+ [16], RefCOCOg [32], LLaVA-Instruct-150K [25]	R-RefCOCO [45], FP-RefCOCO+ [47], FP-RefCOCOg [47]
Seg-Zero [30]	**None**	RefCOCOg [47]
SegLLM [44]	RefCOCO [16], Visual Genome [17], PACO-LVIS [37], LVIS [10], Pascal Panoptic Part [8], ADE20K [54], COCO-Stuff [3], Attributes-COCO [23], ReasonSeg [18], MRSeg (hard) [44]	**None**
SegAgent [56]	**None**	DIS5K [36], ThinObject5K [22]
Text4Seg [19]	**None**	RefCOCO [16], RefCOCO+ [16], RefCOCOg [32], RefCLEF† [16]
STAMP	**None**	RefCOCO [16], RefCOCO+ [16], RefCOCOg [32], RefCLEF† [16]

Table S3. **Ablation and efficiency study on the single-object RES task.** We compare performance on the RES task (cIoU), inference time, and decoder dependency across paradigms. Methods are grouped by their core segmentation paradigm as defined in Table 1. All inference metrics (Time and VRAM) are benchmarked on an A800 GPU.

Method	**Input Size**	**cIoU**	**Time (s)**	**Decoder Status**	**Infer VRAM (GB)**
Paradigm 1: Embedding Prediction					
LISA-7B (CVPR'24) [18]	336 × 336	69.9	1.1	Required SAM	15.5
LISA-13B (CVPR'24) [18]	336 × 336	70.1	1.8	Required SAM	27.6
READ-7B (CVPR'25) [35]	336 × 336	71.9	1.5	Required SAM	15.7
PixelLM-7B (CVPR'24) [40]	448 × 448	69.2	0.7	Required Custom Dec.	14.1
Paradigm 2: Next-token Prediction					
Seg-Zero-7B (arXiv'25) [30]	840 × 840	74.1	4.5	Required SAM2	18.0
SegAgent-7B (CVPR'25) [56]	448 × 448	75.4	15.9	Required SAM	20.7
Text4Seg-7B (ICLR'25) [19]	336 × 336	74.9	4.0	Optional	20.4
W/O Mask Decoder	336 × 336	70.9	3.9	*SAM*	20.4
Paradigm 3: All-mask Prediction (Ours)					
STAMP-2B	504 × 504	77.1	0.9	Optional	10.8
STAMP-2B	726 × 726	78.6	1.1	Optional	10.8
STAMP-2B	896 × 896	79.1	1.3	Optional	10.8
W/O Mask Decoder	896 × 896	75.3	0.9	*SAM*	10.8
STAMP-7B	896 × 896	80.7	2.4	Optional	25.6
W/O Mask Decoder	896 × 896	76.2	2.1	*SAM*	25.6

User

In our daily lives, we need to consume ***food to replenish vitamins, dietary fiber, water, etc***. What objects in the picture, when ingested, can do this?

STAMP *Please explain.*

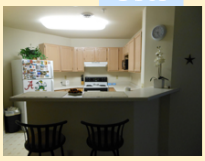
The object in the picture that can do this is ***a bowl of fruit***, such as the one containing bananas, limes, and other fruits. ***Consuming a variety of fruits provides essential nutrients and vitamins that are vital for maintaining good health and overall well-being***. The bowl of fruit in the image serves as a convenient and visually appealing way to incorporate these nutrients into one's diet.

STAMP *Please segment it.*

Sure, the segmentation result is<|seg|>.

User

Hey, in this kitchen photo, there are a bunch of different items. Could you tell me which object is the one that is used for ***telling time*** and its location? Please segment it.



STAMP *Please explain.*

The object used for ***telling time in this kitchen is the clock***, and it is located on the wall.

STAMP *Please segment it.*

Sure, the segmentation result is<|seg|>.

User

Somthing that can be used to eat things. Please segment it.

STAMP *Please explain.*

The image shows a fork resting on a glass bowl filled with a mixture of ingredients. The ***fork is used to scoop the mixture***, which appears to be a mixture of fish and vegetables. The bowl is placed on a surface, and the fork is positioned on top of the mixture, ready to be used to eat the contents. ***The fork is likely designed for serving and eating purposes, as it is designed to hold and manipulate food items.***

STAMP *Please segment it.*

Sure, the segmentation result is<|seg|>.

Figure S2. **Reasoning segmentation cases.** The left column presents the model's detailed explanations of inferred object properties, such as the nutritional role of fruit, the time-keeping function of a clock, and the utility of a fork. Correspondingly, the right column displays the successful segmentation results derived from this understanding."

Input image

User

What sport are the people in the picture doing? Please provide your reasoning and conclusion.

STAMP

The people in the picture are playing ***baseball***. This can be concluded from ***the presence of a baseball bat, a baseball glove, and the players' uniforms***, which are typical equipment for the sport. The players are standing on a baseball field, and the scene appears to be taking place during a game.

User

What is the athlete's number?

STAMP

The athlete's number is 13.

User

Who is that person squatting on the ground wearing red and black protective gear?

STAMP

The person squatting on the ground wearing red and black protective gear is the catcher.

User

Who is that person squatting on the ground wearing red and black protective gear?

STAMP

The player holding a baseball bat with his back to the camera is the batter.

Figure S3. **Visual question answering (VQA) cases.** This example showcases *STAMP*'s performance across various single-turn VQA scenarios. Each exchange, separated by a dashed line, is treated as an independent question-answering task. The model demonstrates comprehensive visual understanding, covering: 1) complex scene reasoning (identifying the sport), 2) precise detail extraction (athlete's number), and 3) specific referential grounding (identifying the catcher and batter).

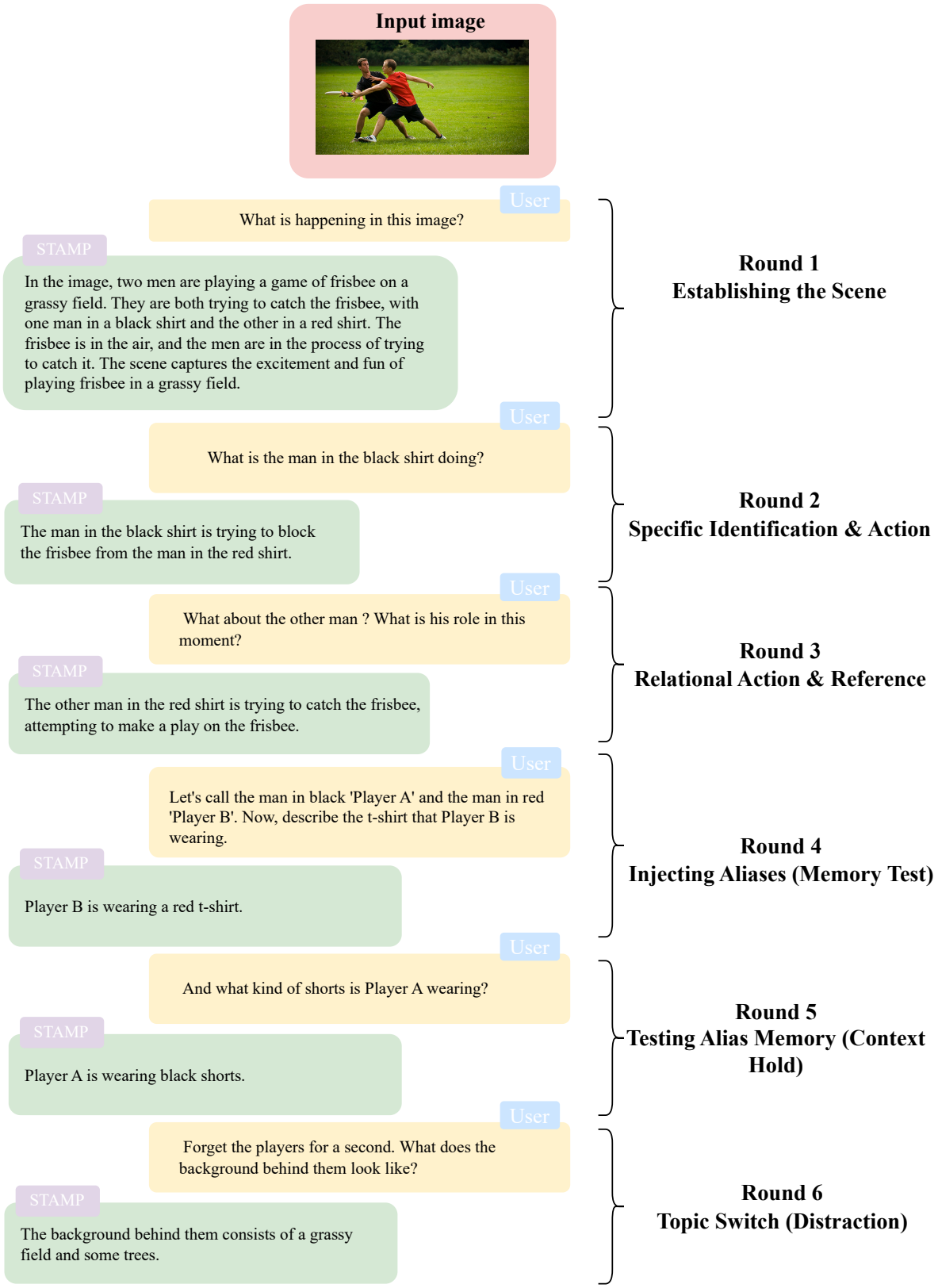


Figure S4. **Multi-round conversational capabilities.** This figure illustrates *STAMP*'s performance in a complex, multi-turn conversation about a single image. The interaction is structured to test the model's ability to maintain context, including: establishing the scene (Round 1), specific identification (Round 2), relational referencing (Round 3), memory injection/testing via aliases (Rounds 4-5), and topic switching with distraction (Round 6). Successful completion of these rounds validates *STAMP*'s robust contextual memory and dialogue coherence.

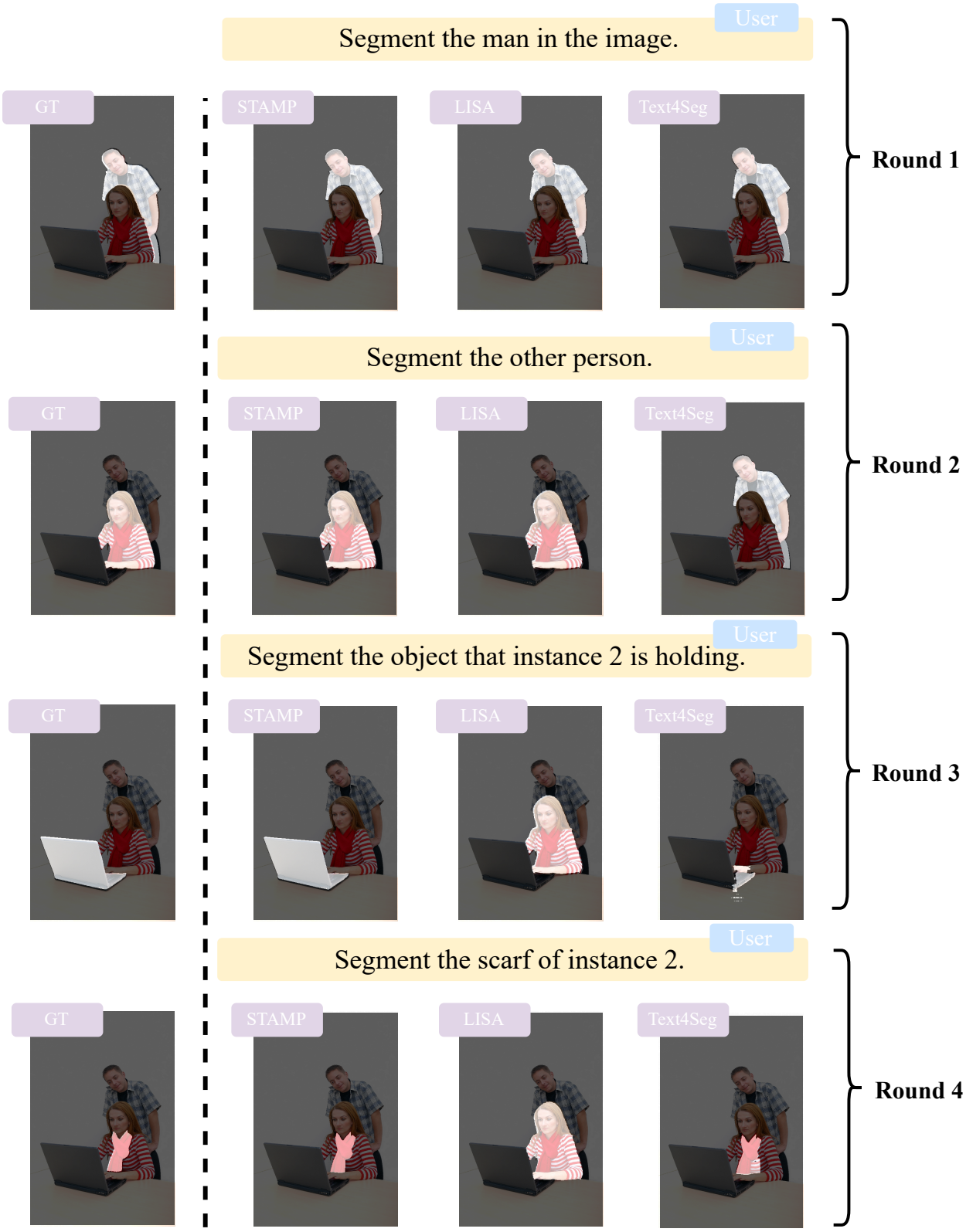


Figure S5. **Multi-round segmentation cases.** This figure illustrates *STAMP*'s performance in multi-turn interactive segmentation, compared against LISA and Text4Seg. The sequential rounds (Round 1 to Round 4) require the model to maintain segmentation context and handle complex referential queries. *STAMP* consistently delivers accurate segmentation even as the queries become more nuanced (e.g., referencing "the other person" in Round 2, segmenting an object held by a previously identified instance in Round 3, and fine-grained attribute segmentation like "scarf of instance 2" in Round 4), showcasing its superior contextual understanding compared to other methods.

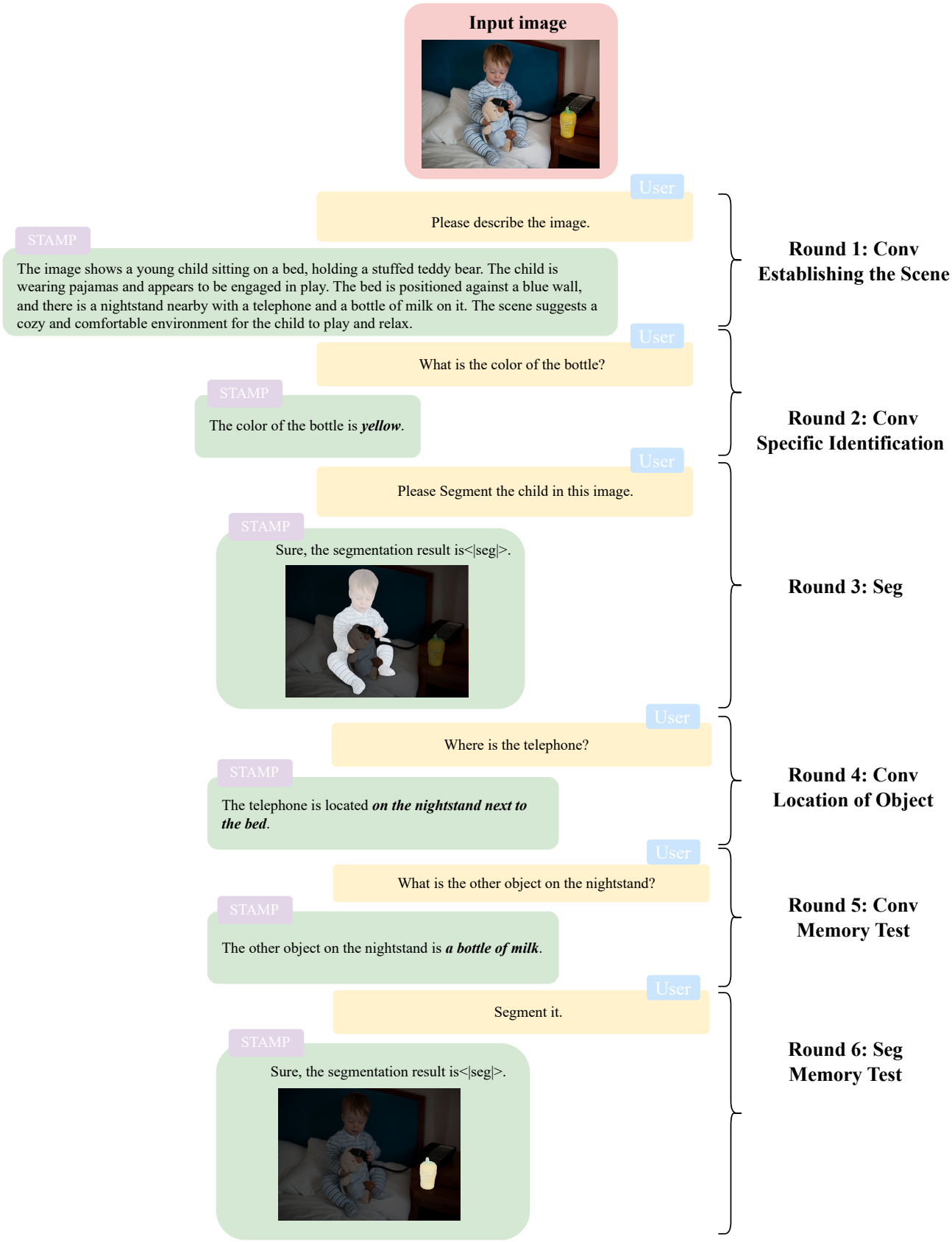


Figure S6. **Seamless interleaving of segmentation and dialogue.** It shows *STAMP*'s unique ability to flexibly interleave Conv and Seg tasks in one dialogue thread. This structure demonstrates context-dependent segmentation, culminating in memory-based segmentation (Round 6). The successful segmentation in Round 6 confirms *STAMP* accurately grounds an object (bottle of milk) into pixel space using only the dialogue history (Rounds 4-5), without requiring a new descriptive prompt.