SAMWISE: Infusing Wisdom in SAM2 for Text-Driven Video Segmentation

Claudia Cuttano[1] Gabriele Trivigno[1] Gabriele Rosi[1,2] Carlo Masone[1,2] Giuseppe Averta[1,2]
[1] Politecnico di Torino [2] Focoos AI
{name.surname}@polito.it {name.surname}@focoos.ai

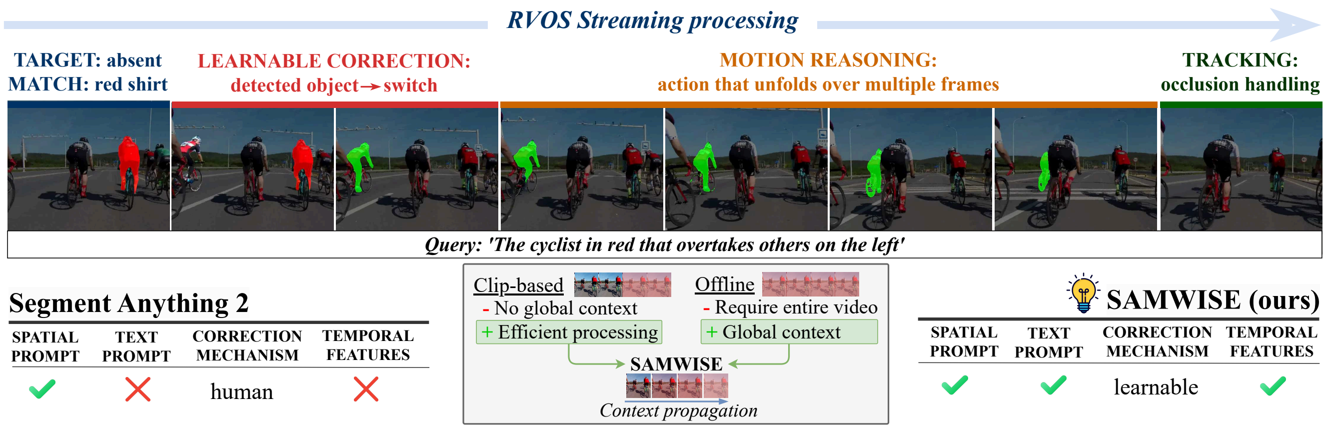


Figure 1. **SAMWISE**. Our approach infuses knowledge about natural language in the Segment-Anything 2 model, adding explicit temporal cues in the feature extraction for the task of streaming-based Referring Video Segmentation (RVOS). We use a learnable mechanism to mitigate the so-called *tracking* bias, *i.e.* SAM2 tendency to overlook a correct object once it becomes identifiable, due to its ongoing tracking of a different object. Our design enables effective streaming processing for RVOS, exploiting the memory from previous frames to propagate past context. The figure shows an example where the target object is not present in the first frame, leading SAM2 to start tracking the wrong one. Afterwards, when the correct object appears, our learnable correction mechanisms guides SAM2 to switch its tracking focus. By adding in its features the notion of temporal evolution, the model is able to recognize that the new object is more aligned with the provided textual query. Finally, we exploit SAM2 tracking skills and robustness to occlusions to keep following the object.

Abstract

Referring Video Object Segmentation (RVOS) relies on natural language expressions to segment an object in a video clip. Existing methods restrict reasoning either to independent short clips, losing global context, or process the entire video offline, impairing their application in a streaming fashion. In this work, we aim to surpass these limitations and design an RVOS method capable of effectively operating in streaming-like scenarios while retaining contextual information from past frames. We build upon the Segment-Anything 2 (SAM2) model, that provides robust segmentation and tracking capabilities and is naturally suited for streaming processing. We make SAM2 *wiser, by empowering it with natural language understanding and explicit temporal modeling at the feature extraction stage, without fine-tuning its weights, and without outsourcing modality interaction to external models. To this end, we introduce a novel adapter module that injects temporal information and multi-modal cues in the feature extraction process. We further reveal the phenomenon of tracking bias in* SAM2 *and propose a learnable module to adjust its tracking focus when the current frame features suggest a new object more aligned with the caption. Our proposed method, SAMWISE, achieves state-of-the-art across various benchmarks, by adding a negligible overhead of less than 5 M parameters. Code is available at* https://github.com/ClaudiaCuttano/SAMWISE.

1. Introduction

Referring video segmentation (RVOS) [10, 18, 25, 39, 44, 49] aims at segmenting and tracking specific objects of interest within video content, guided by natural language expressions [3, 11, 29]. Existing RVOS methods are mostly based on a *divide and conquer* paradigm, where the video is divided into shorter clips that are processed independently [3, 40, 44]. However, as demonstrated by MeViS [8], this solution fails in examples that require taking into account long-term motion and global context. As a workaround to handle this challenge, the state-of-the-art method [12] processes the entire video in an *offline* fashion, first modeling trajectories of all instances throughout the entire clip and then selecting the most appropriate one. Albeit effective, this approach is not applicable when the model has access only to a portion of the video, for example when the data at inference time are presented in a streaming fashion or due to limitations in the computational resources. The trade-off of these two paradigms is schematized in Fig. 1. To this end, OnlineRefer [43] introduced a context propagation scheme for *online* RVOS but relies solely on past context from a single frame, limiting its ability to capture long-term dependencies. In this work, we investigate how to exploit the memory from past frames to design an RVOS method capable of retaining global context while operating within a streaming paradigm, *i.e.*, without requiring access to the whole video at once. This idea is inspired by the recent release of Segment-Anything 2 (SAM2) [36], a foundational model that has shown impressive capabilities in various Video Segmentation tasks thanks to a memory bank that allows to leverage long-range past information. Since SAM2 operates in a streaming fashion, extending this method to enable context-aware streaming processing in RVOS would appear a natural step. However, this entails some non-trivial challenges:

i) Text understanding. SAM2 original design accounts only for *spatial* prompts (e.g. points) and lacks mechanisms to interpret *semantic* prompts like text, which require reasoning over visual and textual modalities. While we are the first to address the challenge of adding textual prompts to SAM2, previous methods have explored this problem for SAM-1 at image-level. These solutions [20, 51] delegate visual-textual interaction to an off-the-shelf large VLM (like BEIT-3 [42], LLaVa [22]), which generates a multi-modal embedding that is used to prompt SAM-1.

ii) Temporal modeling. To segment the referred object throughout the video, it must be first *recognized* and then *tracked*. While the latter requires matching objects visual appearance across adjacent frame, the recognition problem entails modeling temporal evolution to reason over actions that unfold over multiple frames. However, SAM2 extracts frame features independently, lacking such reasoning.

iii) Tracking bias. In RVOS, the target object might be unrecognizable during certain time intervals, due to occlusions, presence of multiple instances or forthcoming actions, as in the first frames of Fig. 1. In such cases, SAM2 may start tracking an incorrect object that partially matches the textual prompt, and persist in following it, leading to what we denote as *tracking bias*. While SAM2 original design allows for a user to manually correct the prediction by providing a new prompt, such a strategy is not applicable in tasks without a human-in-the-loop like RVOS.

In this work, we aim at making SAM2 *wiser*, by addressing these limitations without fine-tuning SAM2 weights, thereby preserving its original capabilities, and without outsourcing modality interaction to external, heavy models. To overcome challenges i) and ii), we design a learnable Adapter [13] module, named Cross-Modal Temporal Adapter (CMT), with two key principles in mind: a) enabling mutual interaction between visual and linguistic modalities; and b) encoding temporal cues into visual features. Then, to generate a prompt, we follow [22, 52] and employ a learnable MLP to project the sentence embedding for the SAM2 Mask Decoder, which then outputs the final segmentation mask. In this way, we can exploit SAM2 tracking capability to segment an object given a textual query across the video. Finally, to mitigate the *tracking bias* problem iii), we introduce a lightweight Conditional Memory Encoder (CME) which detects when a candidate object, aligned with the text, appears in the frame, thus enabling SAM2 to dynamically refocus its tracking to the correct object as it becomes distinguishable.

Summarizing, this paper contributes with the following:

- We present SAMWISE, the first method that integrates natural language knowledge into SAM2 in an end-to-end solution tailored to address the challenges of RVOS. We introduce a novel adapter, namely Cross Modal Temporal (CMT) Adapter, which purposefully models temporal evolution and multi-modal interaction;
- We provide insight into the functioning of SAM2, highlighting the phenomenon of *tracking bias*, and introduce a learnable module (Conditional Memory Encoder) to adjust tracking based on new information;
- Our methods achieves state-of-the-art results both on traditional RVOS benchmarks (Ref-Youtube-VOS [39], Ref-DAVIS [18]), as well as the more challenging MeViS [8], without compromising SAM2 capabilities and adding less than $5M$ learnable parameters.

2. Related works

Referring Video Segmentation. In RVOS, the goal is to segment an object in a clip described with natural language queries [9, 12, 18, 25, 39, 44]. Earlier works proposed adapted image-based methods [2, 10, 18, 49], or used a spatio-temporal memory to attend to masks of previous frames [32, 39]. Subsequent works employ a DETR-like [4] structure to process multiple frames and text embeddings [3, 11, 30, 44]. All these methods process short clips independently, thus losing global context.

Recently, [8] showed how traditional RVOS benchmarks lack challenging captions that require to disambiguate between instances and their actions, as well as occlusions and dynamic queries, highlighting how they could be solved

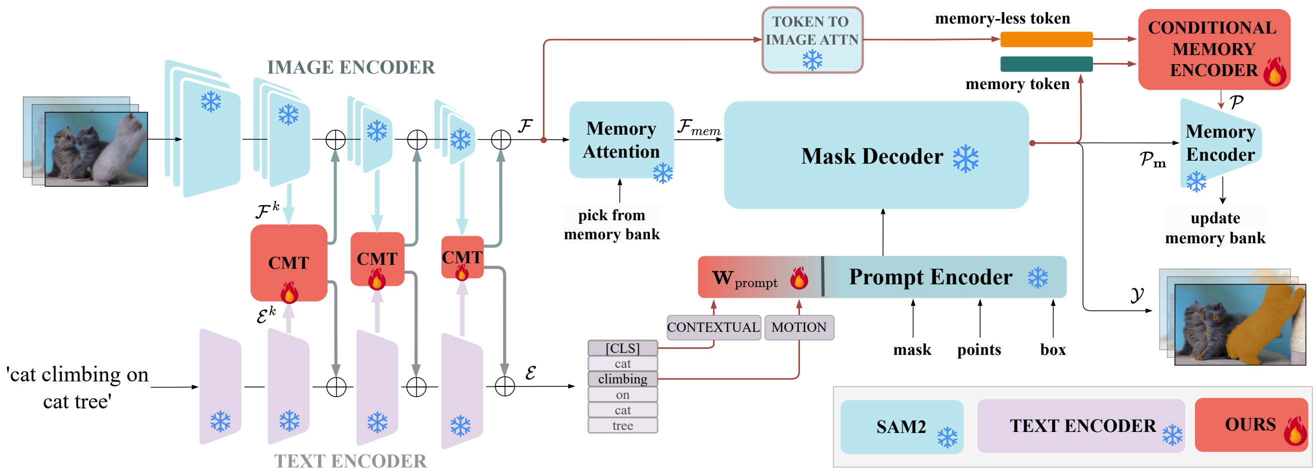


Figure 2. **Overview of SAMWISE.** We build on a frozen SAM2 and a frozen Text Encoder to segment images in video given a textual description. We incorporate the Cross-Modal Temporal Adapter (CMT) into the text and visual encoders at every intermediate layer k to model temporal dynamics within visual features while contaminating each modality with the other. Then, we extract the `[CLS]` and verb embeddings, namely Contextual and Motion prompts, from the adapted textual features and project them through a learnable MLP. The final embedding is used to prompt the Mask Decoder, which outputs the segmentation mask. Finally, the Conditional Memory Encoder detects when a new candidate object, aligned with the caption, appears in the frame, enabling SAM2 to dynamically refocus its tracking.

even with image-based methods. The MeViS dataset [8] targets these scenarios, with challenging examples that previous image or clip-based methods fail to address. To this end, a few works proposed *offline* methods to explicitly model multiple object trajectories [12, 29], with the latter representing the state-of-the-art on MeViS. Concurrently, OnlineRefer [43] proposed a first attempt towards an *online* RVOS setting, with a query propagation scheme. However, its effectiveness is limited as predictions are based on a single frame. Our method builds on this paradigm by leveraging SAM2 memory bank to encode long-range past context.

Text-prompted Segment-Anything. Recent works have provided solutions to adapt SAM-1 for text-prompted segmentation. Grounded SAM [37] employs a two step pipeline where GroundingDINO [26] generates bounding boxes for SAM-1 to produce segmentation masks. Applying such pipeline in RVOS is problematic, as potential errors in the first frame are propagated throughout the whole video. To directly prompt SAM-1, RefSAM [21] exploits a projection layer to map the textual embedding into the prompt space, while [1, 20, 46] resort to large off-the-shelf VLM to generate a multi-modal embedding that is used to prompt SAM-1. Both solutions finetune the Mask Decoder, thereby compromising its capabilities on its original task. In contrast, our work is the first to propose an end-to-end model that incorporates textual knowledge within SAM2 without fine-tuning nor relying on external models.

Pre-Trained Knowledge Transfer. In recent years, the release of powerful pretrained models has sparked interest in the question of how to extend their skills to novel tasks, as full fine-tuning becomes increasingly impractical with growing model sizes [17, 34]. A powerful strategy to address this problem relies on using Adapters [13], small trainable modules that enable efficient adaptation of pre-trained models. Following this paradigm, recent studies have explored adapting CLIP [35] for downstream tasks. At the image level, [45] inserts Transformer Decoder blocks within CLIP encoders, which entail costly Self-Attentions on all tokens. For video tasks, [41] places independent adapter modules within each encoder, whereas [16, 17, 28, 48] rely on a weight-sharing mechanism to project both modalities in a shared sub-space. Nevertheless, as features of each modality are independently extracted, none of these adapters allows explicit feature interaction, unlike our CMT, which also incorporates temporal modeling. Lastly, all these works start from a model that already includes a text encoder (CLIP), whereas ours is the first to propose an adapter for the Segment-Anything 2 model to add textual understanding, achieving robust performances while introducing less than 5 M parameters.

3. SAMWISE

Problem setting. Given an input video $\mathcal{V} = \{I_t\}_{t=1}^{T_V}$ with T_V frames and a referring expression, we aim to predict a set of binary masks $S = \{s_t\}_{t=1}^{T_V}$, $s_t \in \mathbb{R}^{H \times W}$ of the referred object. We tokenize the textual query in a set of L words, $E = \{e_l\}_{l=1}^{L}$, and add a global sentence representation token `[CLS]`. The tokens are then processed using a frozen text encoder to extract language features $\mathcal{E} \in \mathbb{R}^{L \times C_t}$. We process videos in a streaming fashion, collecting clips of T frames as they are available. Throughout the rest of the section, we use T to indicate clip length.

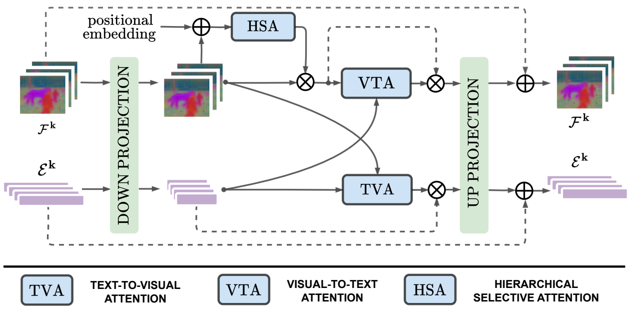


Figure 3. Architecture of our Cross Modal Temporal (CMT) Adapter, made up of Hierarchical Selective Attention (HSA) to model temporal cues, a Visual-to-Text Attention (VTA) and Text-to-Visual Attention (VTA) modules.

Overview. We first provide a brief discussion of the SAM2 model (Sec. 3.1). We then outline the pipeline of our proposed SAMWISE, starting from the prompting strategy in Sec. 3.2. In Sec. 3.3, we detail our novel Cross-Modal Temporal Adapter. Lastly, in Sec. 3.5, we discuss our learnable correction strategy, named Conditional Memory Encoder, to address the issue of *tracking bias*.

3.1. Background: Segment-Anything

The Segment-Anything Model 2 (SAM2) builds upon SAM-1 [19] to tackle the task of Promptable Video Object Segmentation, *i.e.*, tracking an object in a video given a textual prompt. Following SAM-1, it consists of an *image encoder*, a *Prompt Encoder* and a *Mask Decoder*, which combines the image and prompt embeddings to predict segmentation masks. To enable video processing, SAM2 comes with a few modifications: i) the original ViT backbone is replaced by Hiera [38], roughly 3 times faster, which processes frames independently to provide hierarchical visual features. Hereinafter, we refer to them as *memory-less features* $\mathcal{F}$; ii) frame embeddings are not directly fed to the Mask Decoder, but they are first *conditioned* on memories of past predictions from a *Memory Bank*. We refer to these conditioned features as *memory features* $\mathcal{F}_{mem}$. Lastly, iii) once the mask for the current frame is predicted, the *Memory Encoder* updates the Memory Bank. By design, SAM2 handles video frames as they become available, progressively encoding the past in its Memory Bank. We argue that this streaming approach is especially valuable in RVOS, enabling reasoning over a wide temporal horizon.

3.2. Prompting SAM2

To guide the SAM2 decoder, we use a Contextual Prompt, $\mathcal{E}_C \in \mathbb{R}^{1\times C_t}$, which encodes the high-level semantic information for the given text query, emphasizing the essential aspects of the query while downplaying less relevant elements. To this end, we employ the `[CLS]` embedding of text features, $\mathcal{E}$. Furthermore, we also introduce a second prompt, the Motion Prompt $\mathcal{E}_M \in \mathbb{R}^{1\times C_t}$, which captures action-related cues by using verb embeddings from $\mathcal{E}$. These prompts are concatenated and projected through a learnable three-layer MLP:

$$\rho = \mathbf{W}_{\text{prompt}}(\texttt{CAT}[\mathcal{E}_C, \mathcal{E}_M]). \tag{1}$$

In this way, the provided prompts encode both subject-related and motion-based information. Given that in our task the textual prompt is not referred a-priori to any particular frame, we prompt SAM2 at each frame, so that the model has to balance the influence of tracking while also considering the content of each frame. We discuss more in depth this aspect in Sec. 3.5.

3.3. Cross-Modal Temporal Adapter

An adapter consists of a linear down-projection ($\mathbf{W}_{\text{down}}$) to a bottleneck dimensionality, followed by an up-projection back ($\mathbf{W}_{\text{up}}$) in the original space, separated by a non-linear activation function σ. Formally, given an input feature $\mathbf{x} \in \mathbb{R}^{1\times d}$, the adapter function is defined as:

$$Adapter(\mathbf{x}) = \mathbf{x} + \sigma(\mathbf{x}\mathbf{W}_{\text{down}})\mathbf{W}_{\text{up}} \tag{2}$$

We build on this popular Adapter framework [13] and propose a novel Cross-Modal Temporal Adapter (CMT) (see Fig. 3) which models temporal dynamics within visual features while enriching each modality with the other. Formally, given the visual features in a clip $\mathcal{F}^k \in \mathbb{R}^{T\times H_k\times W_k\times C_k}$ and the textual features $\mathcal{E}^k \in \mathbb{R}^{L\times C}$ extracted at layer k of the image and text encoders, respectively, the CMT can be formulated as:

$$\begin{aligned} Adapter(\mathcal{F}^k) &= \mathcal{F}^k + h(\mathcal{F}^k\mathbf{W}_{\text{down,v}}, \mathcal{E}^k\mathbf{W}_{\text{down, t}})\mathbf{W}_{\text{up,v}} \\ Adapter(\mathcal{E}^k) &= \mathcal{E}^k + h(\mathcal{E}^k\mathbf{W}_{\text{down,t}}, \mathcal{F}^k\mathbf{W}_{\text{down,v}})\mathbf{W}_{\text{up,t}} \end{aligned} \tag{3}$$

where $\mathbf{W}_{\text{down,v}}$, $\mathbf{W}_{\text{down,t}}$, $\mathbf{W}_{\text{up,v}}$,$\mathbf{W}_{\text{up,t}}$ are modality specific down- and up-projections weights and h is our proposed adapter function. The adapter output is summed with the original features, allowing the model to retain the original encoding while incorporating temporal and cross-modal reasoning. We integrate the Cross-Modal Temporal Adapter (CMT) into the frozen text and visual encoders at every intermediate layer k. In the following paragraphs we detail the temporal and cross-modal adaptation functions, which are tightly coupled in our Adapter module.

Temporal Adaptation. Our approach aims to embed motion cues directly into the frame-level features of SAM2. Previous works based on Adapters either perform self-attention (SA) over all tokens in a clip [17], which is costly, or restrict the attention to the temporal axis for each pixel [23, 24]. We observe that, within a video, object motion across adjacent frames typically spans a localized region of the image [33]. Consequently, a given element of the

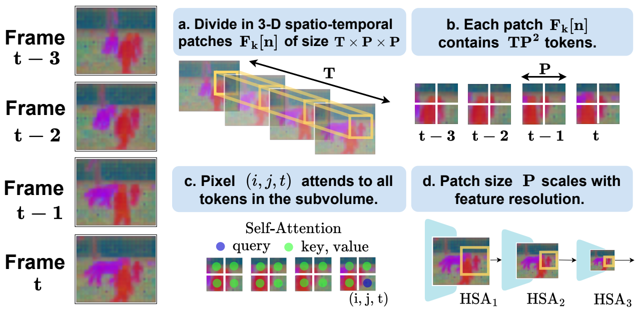


Figure 4. Scheme of our Hierarchical Selective Attention (HSA), modeling temporal evolution of features in our adapter.

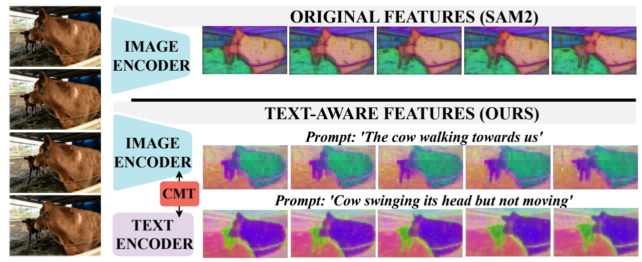


Figure 5. **Cross Modal Temporal Adapter**: we show via PCA that our CMT provides contextualized visual features based on the given textual prompt, compared to SAM2 original ones.

feature map primarily benefits from interactions with its spatial and temporal neighbors, rather than requiring long-range connections across the entire feature map. Building on this intuition, we introduce a Hierarchical Selective Attention (HSA) mechanism, illustrated in Fig. 4. By modeling interactions among spatially and temporally proximal regions, HSA reduces unnecessary computations while capturing motion-based context.

Formally, at layer k, given the set of feature maps for a T-frames clip: $\mathcal{F}^k \in \mathbb{R}^{T\times H_k\times W_k\times C_k}$, we decompose this feature volume into non-overlapping, 3-D spatio-temporal patches of size $T \times P \times P$, obtaining $N = H_k W_k / P^2$ sub-volumes. These sub-volumes, considered pixel-wise, can be represented as a set of tokens $F^k[n] = \left\{x^{k,n}_{i,j,t} \in \mathbb{R}^{C_k} : i \in 1, \ldots, P, j \in 1, \ldots, P, t \in 1, \ldots, T\right\}$. To encode spatio-temporal positioning, to each vector we add a spatial ($e[i, j]$) and a temporal ($e[t]$) sinusoidal positional embeddings, in 2-D and 1-D formats, respectively. Specifically: $x^{k,n}_{i,j,t} = x^{k,n}_{i,j,t} + e[i, j] + e[t]$. Each sub-volume contains $M = P^2 * T$ tokens, on which we perform self-attention as follows:

$$\mathbf{x}^{k,n}_{i,j,t} := SA\left(\left\{\mathbf{x}^{k,n}_{i',j',t'}\right\}_{\substack{i'=1..P\\ j'=1..P\\ t'=1..T}}\right). \tag{4}$$

At each layer k of the feature extraction process, the patch size P is progressively scaled, as depicted in Fig. 4-d. This scaling adapts the sub-volume to the hierarchy of feature resolution, encoding information at multiple scales.

Cross-Modal Adaptation. To unify text and visual representations, we encourage modality interaction from early stages of the feature extraction process through two symmetric operations: Visual-to-Text Attention (VTA) and Text-to-Visual Attention (TVA).

Within the former, each visual feature, already enriched with temporal information through the HSA, attends to the full textual expression, allowing the model to identify candidate regions within the image based on both categorical details (*e.g.*, the subject described in the text) and motion cues (*e.g.*, actions), facilitating early alignment with the prompt, as visible in Fig. 5.

Formally, at layer k, we consider the feature of each frame in the clip, *i.e.* $\mathcal{F}^k[t] \in \mathbb{R}^{H_k\times W_k\times C_k}, t = 1, \ldots, T$, and the set of textual embeddings $\mathcal{E}^k \in \mathbb{R}^{L\times C}$ to compute:

$$\mathcal{F}^k[t] := \mathcal{F}^k[t] * CA(\mathcal{F}^k[t],\ \mathcal{E}^k). \tag{5}$$

In parallel, as the meaning of a caption can shift significantly depending on the visual content of the associated image [7], we aim at contextualizing the textual query with the semantics provided by the visual modality. To this end, the TVA progressively enriches the linguistic tokens $\mathcal{E}^k \in \mathbb{R}^{L\times C}$ with information from the visual feature maps, averaged over the video clip $\mathcal{F}^k_{avg}$:

$$\mathcal{E}^k := \mathcal{E}^k * CA(\mathcal{E}^k,\ \mathcal{F}^k_{avg}). \tag{6}$$

3.4. Mask prediction

At the end of the feature extraction process, we obtain the adapted visual and linguistic features, respectively $\mathcal{E}$ and $\mathcal{F}$. To perform the final prediction, we extract the prompt ρ as in Eq. (1), while the Memory Attention module generates the *memory features* $\mathcal{F}_{mem}$ by conditioning the visual features $\mathcal{F}$ on past predictions from the Memory Bank. The prompt ρ is fed into the frozen Mask Decoder $\mathcal{D}_{dec}$, which generates the output mask $\mathcal{P}_M \in \mathbb{R}^{1\times H\times W}$ and the mask token $\tau_m \in \mathbb{R}^{1\times d}$, *i.e.* an embedding representing the segmented object. Formally:

$$\begin{gathered}\tau_m, \mathcal{P}_M = \mathcal{D}_{dec}(\mathcal{F}_{mem}, \rho),\\ \mathcal{Y} = \mathcal{P}_M > 0,\end{gathered} \tag{7}$$

where $\mathcal{Y} \in \mathbb{R}^{1\times H\times W}$ denotes the output binary segmentation mask. Finally, the Memory Encoder updates the memory bank with $\mathcal{P}_M$.

3.5. Conditional Memory Encoder

We identify as *tracking bias* the phenomenon of SAM2 tracking the wrong object when the correct one is not yet identifiable in the video, and persist in following it. This

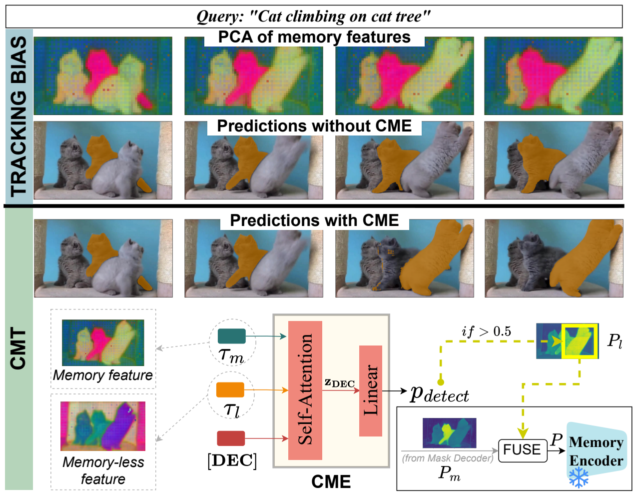


Figure 6. Effect of our **Conditional Memory Encoder**. The caption above requires disambiguating multiple instances of the same class (e.g., "cat") by identifying a specific action (e.g., "climbing"). Since none of the instances perform this action initially, the model begins tracking the wrong instance and fails to correct itself once the target object performs the action. The top section visualizes the effect of this *tracking bias* in the *memory features* of SAM2. Our CME detects that the cat that starts *climbing* is more aligned with the caption, and encodes its presence in the memory bank, allowing SAM2 to switch its focus.

bias, as exemplified in Fig. 6, is encoded in the memory features, which are propagated to subsequent frames through the Memory Encoder. On the other hand, we observe that the memory-less features: i) contain an *unbiased* representation of the current frames, ii) are aligned with the textual prompt via our CMT (*cf.* Fig. 5), and iii) can thus be used to propose candidate instances that match the prompt without being biased by past predictions. Building on these intuitions, we derive a *memory-less token* τ_l from a cross-attention between the unbiased feature maps and the prompt. Such token represents a summary of the visual features that match the prompt. The idea is to compare it with the mask token τ_m generated by the Mask Decoder, to detect when they represent different objects, *i.e.*, to detect when SAM2 is tracking an object that is not the one *currently* most aligned with the caption. Formally:

$$\tau_l = CA\left(\mathcal{F}, \rho\right). \tag{8}$$

We note that we initialize (and keep frozen) the weights of the cross-attention with those from SAM2 Mask Decoder. We introduce a small learnable module, named Conditional Memory Encoder (CME), to detect such situations. When a new object is detected, a naive solution would be to compute its mask and use it to re-prompt the model at the given frame, just like a user would do, forcing SAM2 to switch its prediction. However, since the prediction computed on the *memory-less* features does not have access to past video context, it might generate false positives. Thus, we propose a *soft assignment*, obtained by encoding the masks of both objects in the memory bank. Essentially, the CME allows SAM2 to 'see' other objects beyond the currently tracked one, and balance the influence of past context with new information, to select the one that fits the prompt the most. In detail, our CME, illustrated in Fig. 6-bottom, concatenates the two tokens τ_m, τ_l with a learnable *decision token* `[DEC]`, and performs a self-attention followed by a Linear classifier:

$$\begin{aligned} [z_{DEC}, z_{MT}, z_{ML}] &= SA\left(\left[[\texttt{DEC}], \tau_m[t], \tau_l[t]\right]\right), \\ p_{detect} &= \phi(z_{DEC}), \end{aligned} \tag{9}$$

where ϕ is a linear function $\mathbb{R}^d \rightarrow \mathbb{R}^1$. When detecting a candidate text-aligned object, (*i.e.*, $p_{detect} > 0.5$), instead of directly feeding the predicted output mask $\mathcal{P}_m$ to the Memory Encoder, our module computes the unbiased output mask, namely $\mathcal{P}_L \in \mathbb{R}^{1\times H\times W}$, to fuse it with $\mathcal{P}_m$:

$$\begin{aligned} \mathcal{P}_l &= \mathcal{D}_{dec}\left(\mathcal{F}, \rho\right), \\ \mathcal{M}(h,w) &= \mathbb{1}(h,w)\left[h,w : \mathcal{P}_l > 0\right], \\ \mathcal{P} &= \lambda * \mathcal{P}_l \circ \mathcal{M} + \mathcal{P}_m \circ (1-\mathcal{M}), \end{aligned} \tag{10}$$

where $\mathcal{M}(h,w)$ is a binary mask whose value is zero except for the pixels corresponding to the object, and λ is an hyperparameter weighing the influence of the memory-less prediction. The resulting mask $\mathcal{P}$ is fed to the Memory Encoder. We train the CME via self-supervision with a standard Cross-Entropy loss, by providing examples where the *memory-less* features highlight different objects w.r.t the one currently tracked. We discuss in detail our training protocol in the Supp. Mat..

4. Experimental results

Dataset. We evaluate our method on MeVis [8], Ref-Youtube-VOS [2] and Ref-Davis [18]. MeViS includes 2,006 videos and features a total of 28K annotations that capture various aspects of motion. Ref-Youtube-VOS enhances the original YouTube-VOS benchmark by incorporating textual descriptions. It contains a total of 3,978 videos and approximately 15K language expressions. Ref-DAVIS17 builds upon DAVIS17 dataset, adding more than 1.5K linguistic annotations to 90 videos.

Evaluation Metrics. We utilize standard evaluation metrics, region similarity ($\mathcal{J}$), contour accuracy ($\mathcal{F}$), and their average ($\mathcal{J}$ & $\mathcal{F}$). For MeViS and Ref-Youtube-VOS we conduct the evaluation using the official challenge servers; for Ref-DAVIS17, we used the official evaluation code.

Implementation Details. We employ Hiera-B [38] as visual extractor. As text encoder, we experiment with

Method	Visual Encoder	Text Encoder	Total Params	MeViS $\mathcal{J}\&\mathcal{F}$	MeViS $\mathcal{J}$	MeViS $\mathcal{F}$	Ref-YouTube-VOS $\mathcal{J}\&\mathcal{F}$	Ref-YouTube-VOS $\mathcal{J}$	Ref-YouTube-VOS $\mathcal{F}$	Ref-DAVIS17 $\mathcal{J}\&\mathcal{F}$	Ref-DAVIS17 $\mathcal{J}$	Ref-DAVIS17 $\mathcal{F}$
***Large VLM** based*												
LISA [20] [CVPR'24]	ViT-H	LLaVa	7 B	37.2	35.1	39.4	53.9	53.4	54.3	64.8	62.2	67.3
VISA [46] [ECCV'24]	ViT-H	Chat-UniVi	7 B	43.5	40.7	46.3	61.5	59.8	63.2	69.4	66.3	72.5
One-Token-Seg-All [1] [NIPS'24]	ViT-H	Phi-3	3.8 B	42.3	39.4	45.2	61.7	60.2	63.3	67.7	63.8	71.5
MTTR [3] [CVPR'22]	V-Swin-T	RoBERTa	-	30.0	28.8	31.2	55.3	54.0	56.6	-	-	-
TCE-RVOS [15] [WACV'24]	ResNet-50	RoBERTa	-	-	-	-	59.6	58.3	60.8	59.4	56.5	62.4
ReferFormer [44] [CVPR'22]	V-Swin-B	RoBERTa	237 M	31.0	29.8	32.2	62.9	61.3	64.6	61.1	58.1	64.1
SOC [29] [NIPS'23]	V-Swin-B	RoBERTa	220 M	-	-	-	66.0	64.1	67.9	64.2	61.0	67.4
OnlineRefer [43] [ICCV'23]	Swin-L	RoBERTa	232 M	32.3	31.5	33.1	63.5	61.6	65.5	64.8	61.6	67.7
LMPM [8] [ICCV'23]	Swin-T	RoBERTa	195 M	37.2	34.2	40.2	-	-	-	-	-	-
RefSAM [21] [arXiv]	ViT-B	T5	3 B	-	-	-	58.4	57.4	59.4	62.1	59.0	65.3
DsHmp [12] [CVPR'24]	V-Swin B	RoBERTa	339 M	-	-	-	67.1	65.0	69.1	64.9	61.7	68.1
DsHmp [12] [CVPR'24]	Swin-T	RoBERTa	272 M	46.4	43.0	49.8	-	-	-	-	-	-
MUTR [47] [AAAI'24]	V-Swin-B	RoBERTa	250 M	-	-	-	67.5	65.4	69.6	66.4	62.8	70.0
GroundingDINO [26]+SAM2	Hiera-B	BERT	240 M	37.7	34.9	40.5	57.5	55.6	59.5	66.4	62.8	69.9
SAMWISE **(ours)**	Hiera-B	CLIP-B	150 M	48.3	45.4	51.2	67.2	65.2	69.3	68.5	65.6	71.5
SAMWISE **(ours)**	Hiera-B	RoBERTa	202 M	**49.5**	**46.6**	**52.4**	**69.2**	**67.8**	**70.6**	**70.6**	**67.4**	**74.5**

Table 1. Comparison of **SAMWISE** against state-of-the-art RVOS methods on MeViS, Ref-Youtube-VOS and Ref-DAVIS datasets. We further include methods based on large VLMs for comparison. **Bold** and underline indicate the two top results.

CLIP [35] and RoBERTa [27]. We note that the text encoder and SAM2 weights are entirely frozen and we train only the Adapters and the CME module (4.2M parameters when using CLIP and 4.9M with RoBERTa). Following [11, 21, 29, 43, 44], we undergo pre-training for 6 epochs on RefCOCO/+/g [31, 50] with a learning rate at 1e-4 and fine-tune on Ref-Youtube-VOS [39] for 4 epochs with a learning rate of 1e-5, using the Adam optimizer. The model trained on the Ref-YouTube-VOS is directly evaluated on DAVIS-17 [18]. On MeViS [8], we train for 1 epoch. We set $T = 8$.

4.1. Main Results

In this section, we compare against existing works in the literature, and ablate our contributions. In the Supp. Mat. we report additional qualitative results and ablations.

Baselines. To asses the validity of our approach, we divide the experimental comparison in the following categories:

- **Standard RVOS** methods: we compare against recent relevant works in RVOS. The main comparison is w.r.t. the previous state-of-the-art, namely DsHmp [12];
- Methods with **Context propagation**: OnlineRefer [43] was the first to propose this setting. RefSAM [21] relies on SAM1 to provide frame-level masks, and then propagates the mask token to subsequent frames. A baseline that we propose is GroundingDINO + SAM2, where we use the popular grounded detector to provide boxes for the first frame, and let SAM2 track the object;
- **Large VLM based**: Although these methods [1, 20, 46] are not comparable to ours, or previous ones, in terms of model size, we include them in the table to provide an interesting reference of performance.

Comparison with standard RVOS methods. Traditional RVOS methods, such as ReferFormer [44] and MTTR [3], suffer a significant performance drop on the MeViS benchmark, as they are unable to solve queries which require to model long-term context. An exception is represented by LMPM and its follow up work DsHmp, which represents the state-of-the-art: these methods process the entire video clip at once, modeling multiple trajectories for all the instances in the video to select the one that fits the prompt the most. Despite this, SAMWISE outperforms DsHmp [12] on all three datasets, improving $\mathcal{J}\&\mathcal{F}$ of +3.1%, +2.1%, and +5.7%, while utilizing a smaller model in terms of total parameters. Notably, we achieve this by training only 4.9 M parameters out of 202 M. This result is particularly impressive, as offline methods exploit information from the entire video to handle challenges such as late-appearing objects or motion-dependent disambiguation, as opposed to our streaming approach. With respect to other methods, we outperform them significantly on MeViS, whereas the gap is smaller on Ref-YouTube-VOS and Ref-DAVIS, which contain more descriptive captions, and object-centric videos. Lastly, we also experiment with the text encoder of CLIP, which achieves state-of-the-art results on MeViS, and competitive performance on other benchmarks, while providing a more compact model with just 150 M params.

Comparison with Context Propagation methods. Our proposed baseline GroundingDINO [26]+SAM2, while obviously flawed, being forced to predict the desired instance based on the first frame only, achieves acceptable results on Ref-DAVIS, whereas on MeViS and Ref-Youtube-VOS its performance drops of 11.8% and 11.7%, respectively. Differently, SAMWISE, demonstrates excellent performance in both motion-dependent and static scenarios. Specifically,

MLP-only	Text-to-Visual	Visual-to-Text	HSA	CME	$\mathcal{J}\&\mathcal{F}$
✓					45.2
✓	✓				47.5
✓		✓			48.3
✓	✓	✓			50.3
✓	✓	✓	✓		54.2
✓	✓	✓	✓	✓	55.5

Table 2. Ablation of our **Cross-Modal Temporal Adapter** (CMT). We show the effect of not using CMT (*i.e. MLP-only* to prompt SAM2), vs. adding one at a time its core components. Lastly, the **Conditional Memory Encoder** (CME) is added.

	HSA Patch Size				
	Fixed Size			Hierarchical	
	1	4	8	8 / 4 / 2	16 / 8 / 4
$\mathcal{J}\&\mathcal{F}$	49.7	52.3	53.1	**54.2**	53.8

	CME vs Random choice			
	Never	Always	1 every 4	CME
$\mathcal{J}\&\mathcal{F}$	54.2	50.7	52.4	**55.5**

Table 3. Top: Ablation of the Patch size in our **HSA**, with the effect of a fixed size vs Hierarchical. Bottom: Effect of not predicting detections (*Never*), predicting them at every frame (*Always*), randomly (*1 in 4*) vs. using predictions of our CME.

on MeViS, we outperform OnlineRefer [43] by +17.2%. On Ref-YouTube-VOS and Ref-DAVIS, the gap is of +5.7%, and +5.8%, respectively.

Comparison with Large-VLM based. While comparisons with Large-VLM based approaches are not standard in RVOS evaluations, we include them in this work to provide additional context. The VLM-based solutions [1, 20, 46] are designed to leverage the extensive reasoning capabilities of VLMs to address complex textual instructions and implicit descriptions that require world knowledge [20]. This leads to improved performance in tasks like MeViS, where reasoning over motion patterns is required. However, delegating cross-modal reasoning to these VLMs incurs in significant computational overhead, whereas SAMWISE incorporates visual-text interaction directly at the feature level. Notably, SAMWISE outperforms VISA, the best VLM-based competitor by a substantial margin, respectively +6%, +7.7%, +2.9% on the three benchmarks.

4.2. Ablation Studies

We conduct our ablations on MeViS, as it embodies the core challenges of *online* RVOS. We report results on the 'valid_u' set [8], employing CLIP-B as text encoder.

Making SAM2 Wiser. We start by showing, in Tab. 2, how each of the core components of our CMT Adapter progressively injects *wisdom* (*i.e*. knowledge about language and temporal context) into SAM2. The first line reports the result using the 'naive' solution of aligning the textual prompt to the visual features using a single learnable MLP [52]. While effective to some extent, the results show that allowing early interaction of the two modalities grants a substantial boost (+5.1% with both adapters). Adding explicit temporal feature modeling provides an additional improvements of +3.9%. These results sustain our intuition that adding frame-level alignment through a MLP is not enough to obtain robust performances, and that it is essential to couple the text and visual semantics, as well as modeling temporal context. Lastly, the table shows how adding our CME is effective in mitigating SAM2 *tracking bias*.

Hierarchical Selective Attention. In the top section of Tab. 3, we study the effect of the spatial patch size in our HSA module, which models the temporal evolution of over a spatial patch of size P across the temporal axis. Using $P = 1$ is equivalent to processing each pixel independently across frames. The table shows that including spatial context, up to 8 pixels, is beneficial. Using a hierarchical patch size that scales with the feature map resolution yields a gain of +1.1% over the fixed sized alternative.

Conditional Memory Encoder. The bottom section of Tab. 3 provides insight into our CME module. The CME, essentially, detects whenever an object in the *unbiased* feature maps of the current frame displays higher alignment with the textual prompts w.r.t. the currently tracked one, but SAM2 fails in noticing it due to the *tracking bias* (Fig. 6). The table compares the effect of *Never* applying such strategy (*i.e.*, not using CME), doing it *Always* (*i.e.*, at every frame), or once every 4 frames. The results show that increasing the frequency of *artificial* detection worsens performances, adding noise to the tracking, whereas the predictions of our CME are beneficial, with a boost of +1.3%.

Adaptation strategy. In Tab. 4 we compare different adaptation strategies, including Full Fine-Tuning (FT), LoRa [14], AdaptFormer [5], and the proposed CMT.

	Full-FT	LoRa [14]	AdaptFormer [5]	CMT (ours)
MeViS $\mathcal{J}\&\mathcal{F}$	43.1	44.2	43.9	**48.3**

Table 4. Baselines: prior adapters and full-finetune, with CLIP-B.

5. Conclusion

In this work, we introduced SAMWISE, a novel approach for RVOS that builds upon the SAM2 model by incorporating i) natural language understanding, ii) temporal feature modeling, and iii) a learnable strategy to adjusts tracking focus according to visual cues that emerge over time. SAMWISE achieves SOTA performance across benchmarks while adding less than 5M parameters, without modifying SAM2 weights or using external models for visual-text alignment. We obtain an effective pipeline for applications of streaming video segmentation, addressing limitations of existing RVOS approaches, which either lack long-term context or rely on single-frame context propagation.

Acknowledgements. Claudia Cuttano was supported by the Sustainable Mobility Center (CNMS) which received funding from the European Union Next Generation EU (Piano Nazionale di Ripresa e Resilienza (PNRR), Missione 4 Componente 2 Investimento 1.4 "Potenziamento strutture di ricerca e creazione di "campioni nazionali di R&S" su alcune Key Enabling Technologies") with grant agreement no. CN_00000023.

Gabriele Trivigno, Giuseppe Averta and Carlo Masone were supported by FAIR - Future Artificial Intelligence Research which received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3 – D.D. 1555 11/10/2022, PE00000013). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

We acknowledge the CINECA award under the ISCRA initiative, for the availability of high per- formance computing resources.

References

[1] Zechen Bai, Tong He, Haiyang Mei, Pichao Wang, Ziteng Gao, Joya Chen, Lei Liu, Zheng Zhang, and Mike Zheng Shou. One token to seg them all: Language instructed reasoning segmentation in videos. *arXiv preprint arXiv:2409.19603*, 2024. 3, 7, 8

[2] Miriam Bellver, Carles Ventura, Carina Silberer, Ioannis Kazakos, Jordi Torres, and Xavier Giro-i Nieto. A closer look at referring expressions for video object segmentation. *Multimedia Tools and Applications*, 82(3):4419–4438, 2023. 2, 6, 5

[3] Adam Botach, Evgenii Zheltonozhskii, and Chaim Baskin. End-to-end referring video object segmentation with multimodal transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4985–4995, 2022. 1, 2, 7

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 2

[5] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678, 2022. 8

[6] Yong Xien Chng, Henry Zheng, Yizeng Han, Xuchong Qiu, and Gao Huang. Mask grounding for referring image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26573–26583, 2024. 4

[7] Henghui Ding, Chang Liu, Suchen Wang, and Xudong Jiang. Vlt: Vision-language transformer and query generation for referring segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):7900–7916, 2022. 5

[8] Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, and Chen Change Loy. MeViS: A large-scale benchmark for video segmentation with motion expressions. In *ICCV*, 2023. 1, 2, 3, 6, 7, 8, 5

[9] Zihan Ding, Tianrui Hui, Shaofei Huang, Si Liu, Xuan Luo, Junshi Huang, and Xiaoming Wei. Progressive multimodal interaction network for referring video object segmentation. *The 3rd Large-scale Video Object Segmentation Challenge*, 8(10), 2021. 2

[10] Kirill Gavrilyuk, Amir Ghodrati, Zhenyang Li, and Cees GM Snoek. Actor and action video segmentation from a sentence. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5958–5966, 2018. 1, 2

[11] Mingfei Han, Yali Wang, Zhihui Li, Lina Yao, Xiaojun Chang, and Yu Qiao. Html: Hybrid temporal-scale multimodal learning framework for referring video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13414–13423, 2023. 1, 2, 7

[12] Shuting He and Henghui Ding. Decoupling static and hierarchical motion perception for referring video segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13332–13341, 2024. 2, 3, 7

[13] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR, 2019. 2, 3, 4

[14] 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. *ICLR*, 1(2):3, 2022. 8

[15] Xiao Hu, Basavaraj Hampiholi, Heiko Neumann, and Jochen Lang. Temporal context enhanced referring video object segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5574–5583, 2024. 7, 3

[16] Haojun Jiang, Jianke Zhang, Rui Huang, Chunjiang Ge, Zanlin Ni, Jiwen Lu, Jie Zhou, Shiji Song, and Gao Huang. Cross-modal adapter for text-video retrieval. *arXiv preprint arXiv:2211.09623*, 2022. 3

[17] Xiaojie Jin, Bowen Zhang, Weibo Gong, Kai Xu, Xueqing Deng, Peng Wang, Zhao Zhang, Xiaohui Shen, and Jiashi Feng. Mv-adapter: Multimodal video transfer learning for video text retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27144–27153, 2024. 3, 4

[18] Anna Khoreva, Anna Rohrbach, and Bernt Schiele. Video object segmentation with language referring expressions. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part IV 14*, pages 123–141. Springer, 2019. 1, 2, 6, 7, 5

[19] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 4

[20] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation

via large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9579–9589, 2024. 2, 3, 7, 8

[21] Yonglin Li, Jing Zhang, Xiao Teng, Long Lan, and Xinwang Liu. Refsam: Efficiently adapting segmenting anything model for referring video object segmentation, 2024. 3, 7

[22] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 2

[23] Ruyang Liu, Jingjia Huang, Ge Li, Jiashi Feng, Xinglong Wu, and Thomas H Li. Revisiting temporal modeling for clip-based image-to-video knowledge transferring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6555–6564, 2023. 4

[24] Ruyang Liu, Chen Li, Yixiao Ge, Thomas H. Li, Ying Shan, and Ge Li. Bt-adapter: Video conversation is feasible without video instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13658–13667, 2024. 4

[25] Si Liu, Tianrui Hui, Shaofei Huang, Yunchao Wei, Bo Li, and Guanbin Li. Cross-modal progressive comprehension for referring segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):4761–4775, 2021. 1, 2

[26] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 3, 7, 4

[27] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 7

[28] Haoyu Lu, Yuqi Huo, Guoxing Yang, Zhiwu Lu, Wei Zhan, Masayoshi Tomizuka, and Mingyu Ding. Uniadapter: Unified parameter-efficient transfer learning for cross-modal modeling. *arXiv preprint arXiv:2302.06605*, 2023. 3

[29] Zhuoyan Luo, Yicheng Xiao, Yong Liu, Shuyan Li, Yitong Wang, Yansong Tang, Xiu Li, and Yujiu Yang. Soc: Semantic-assisted object cluster for referring video object segmentation. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 3, 7

[30] Bo Miao, Mohammed Bennamoun, Yongsheng Gao, and Ajmal Mian. Spectrum-guided multi-granularity referring video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 920–930, 2023. 2

[31] Varun K Nagaraja, Vlad I Morariu, and Larry S Davis. Modeling context between objects for referring expression understanding. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 792–807. Springer, 2016. 7

[32] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9226–9235, 2019. 2

[33] Mandela Patrick, Dylan Campbell, Yuki Asano, Ishan Misra, Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, and Joao F Henriques. Keeping your eye on the ball: Trajectory attention in video transformers. *Advances in neural information processing systems*, 34:12493–12506, 2021. 4

[34] Matthew E Peters, Sebastian Ruder, and Noah A Smith. To tune or not to tune? adapting pretrained representations to diverse tasks. *arXiv preprint arXiv:1903.05987*, 2019. 3

[35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3, 7

[36] 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. 2, 4

[37] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024. 3

[38] Chaitanya Ryali, Yuan-Ting Hu, Daniel Bolya, Chen Wei, Haoqi Fan, Po-Yao Huang, Vaibhav Aggarwal, Arkabandhu Chowdhury, Omid Poursaeed, Judy Hoffman, et al. Hiera: A hierarchical vision transformer without the bells-and-whistles. In *International Conference on Machine Learning*, pages 29441–29454. PMLR, 2023. 4, 6

[39] Seonguk Seo, Joon-Young Lee, and Bohyung Han. Urvos: Unified referring video object segmentation network with a large-scale benchmark. In *European Conference on Computer Vision*, 2020. 1, 2, 7

[40] Jiajin Tang, Ge Zheng, and Sibei Yang. Temporal collection and distribution for referring video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15466–15476, 2023. 1

[41] Mengmeng Wang, Jiazheng Xing, Boyuan Jiang, Jun Chen, Jianbiao Mei, Xingxing Zuo, Guang Dai, Jingdong Wang, and Yong Liu. A multimodal, multi-task adapting framework for video action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5517–5525, 2024. 3

[42] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*, 2022. 2

[43] Dongming Wu, Tiancai Wang, Yuang Zhang, Xiangyu Zhang, and Jianbing Shen. OnlineRefer: A simple online baseline for referring video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2761–2770, 2023. 2, 3, 7, 8

[44] Jiannan Wu, Yi Jiang, Peize Sun, Zehuan Yuan, and Ping Luo. Language as queries for referring video object segmentation. In *Proceedings of the IEEE/CVF Conference*

on Computer Vision and Pattern Recognition, pages 4974–4984, 2022. 1, 2, 7, 3

[45] Zunnan Xu, Zhihong Chen, Yong Zhang, Yibing Song, Xiang Wan, and Guanbin Li. Bridging vision and language encoders: Parameter-efficient tuning for referring image segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17503–17512, 2023. 3

[46] 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. *arXiv preprint arXiv:2407.11325*, 2024. 3, 7, 8

[47] Shilin Yan, Renrui Zhang, Ziyu Guo, Wenchao Chen, Wei Zhang, Hongyang Li, Yu Qiao, Hao Dong, Zhongjiang He, and Peng Gao. Referred by multi-modality: A unified temporal transformer for video object segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6449–6457, 2024. 7, 3

[48] Lingxiao Yang, Ru-Yuan Zhang, Yanchen Wang, and Xiaohua Xie. Mma: Multi-modal adapter for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23826–23837, 2024. 3

[49] Linwei Ye, Mrigank Rochan, Zhi Liu, and Yang Wang. Cross-modal self-attention network for referring image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10502–10511, 2019. 1, 2

[50] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 69–85. Springer, 2016. 7

[51] Yuxuan Zhang, Tianheng Cheng, Rui Hu, Heng Liu, Longjin Ran, Xiaoxin Chen, Wenyu Liu, Xinggang Wang, et al. Evf-sam: Early vision-language fusion for text-prompted segment anything model. *arXiv preprint arXiv:2406.20076*, 2024. 2

[52] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 2, 8

SAMWISE: Infusing Wisdom in SAM2 for Text-Driven Video Segmentation

Supplementary Material

Supplementary

In this supplementary material we discuss:

- the training protocol;
- further insights on the functioning of the Conditional Memory Encoder (CME), our learnable correction mechanism to adjust SAM2 tracking focus;
- additional ablations: on our Cross-Modal Temporal (CMT) Adapter, on inference window size, comparison with smaller backbones, and experiments on Referring Image Segmentation;
- comparison with SAM2-based baselines;
- qualitative examples from MeViS to assess the effectiveness of SAMWISE on challenging scenarios.

6. Training protocol

Following [44], we train our model with a combination of DICE loss and binary mask focal loss. We train our **Conditional Memory Encoder (CME)** via **self-supervision**. For each video clip, given the prompt ρ we compute the predicted masks using SAM2 Mask Decoder:

$$Y_m[t] = \mathcal{D}_{dec}(\mathcal{F}_{mem}, \rho) > 0, t = 1..T. \tag{11}$$

The predicted masks $Y_m[t]$ represent the standard output of SAM2 Mask Decoder, *i.e.* the masks computed given the memory features F_{mem}. As we aim at detecting when the *memory-less* features highlight different object w.r.t. the one currently tracked, we further compute the unbiased output mask. By employing the unbiased *memory-less* features, which do not take into account the previous tracking context encoded in the Memory Bank, the prediction is based solely on the object currently more aligned to the caption in the given clip. Formally:

$$\mathcal{Y}_l[t] = \mathcal{D}_{dec}\left(\mathcal{F}, \rho\right) > 0, t = 1..T. \tag{12}$$

Given each pair of the binary masks at frame t, we define the detection label as:

$$y_t = \begin{cases} 1 & if \quad \mathcal{Y}_l[t] \cap \mathcal{Y}_m[t] = 0 \\ 0 & \quad otherwise \end{cases} \tag{13}$$

The label is 1 if the intersection of the two masks is null, *i.e.* the masks segment different objects. We supervise our CME with a standard Cross-Entropy loss:

$$\mathcal{L}_{CME} = -\frac{1}{T}\sum_{t=1}^{T}\left[y_t log(p_{detect}) + (1-y_t) log(1-p_{detect})\right], \tag{14}$$

where p_{detect} is computed as in eq. 9 of the main paper.

7. CME: Qualitative impact

In this section, we analyze the impact of the Conditional Memory Encoder (CME) within SAMWISE. In Fig. 7 and Fig. 8, the model is tasked to segment the correct object in the video based on the provided referring expression. We use yellow masks to represent the output predictions generated by SAMWISE. Generally, the model tracks the object that appears most relevant according to the information available up to that point. However, due to the phenomenon of *tracking bias*, *i.e.* the tendency to continue tracking an initially detected object, the correct object might not be selected when it appears. Our CME addresses this challenge by detecting when an object aligned with the text prompt becomes visible. Upon detection, the CME computes the corresponding mask and encodes it into the Memory Bank. To highlight the CME role, we show the candidate masks it proposes in green or red, reflecting whether the proposed mask denotes a correct or incorrect detected object. For clarity, these masks are not predicted as final output but are temporary representations stored in the Memory Bank. By encoding these candidate masks, the CME enables SAMWISE to adjust its tracking dynamically, balancing the influence of previously tracked objects with newly detected ones.

Correct Object Detection by CME. In Fig. 7, we showcase examples in which the CME successfully identifies the correct object. These examples highlight various challenging scenarios. In some cases, all potential objects are present in the scene from the beginning, but the discriminative action that distinguishes between them only occurs later in the video. For example, in case (a), the target cat starts *climbing* only at a specific point in the sequence, and similarly, in case (c), the elephant *touches its trunk to the back of the other elephant* at a later moment. In other scenarios, the action itself remains ambiguous until a key point. For instance, in example (e), the action of *turning left* only becomes identifiable after a certain frame, at which point the CME detects the correct car and informs SAMWISE, allowing it to shift focus to the correct instance. Similarly, in (d), the model faces a challenging scenario, where several instances are visible in the video and the action of *moving a bit* remains ambiguous during the first frames. In other situations, like case (b), the target object is not visible at the start. Here, SAMWISE starts tracking a different object (an incorrect airplane) until the target appears in the scene.

Handling Incorrect Candidate Detection. In Fig. 8, we demonstrate the robustness of SAMWISE against incorrect candidate proposals generated by the CME. While our CME generates masks that align with the text prompt at clip-level, these proposals may not align correctly at a global level. This occurs because the CME reasons locally within the scope of the current clip, potentially leading to plausible but ultimately incorrect proposals. Interestingly, SAMWISE is able to reason about past predictions and determine which object better aligns with the referring query, by relying on the broader context encoded in the Memory Bank. Therefore, the model is able to assess whether the candidate object is more aligned to the tracked object. We show this through a number of representative examples. For instance, in case (a), the CME proposes a novel plausible car (red mask). However, the previously tracked object was already *traveling in a straight line*, and

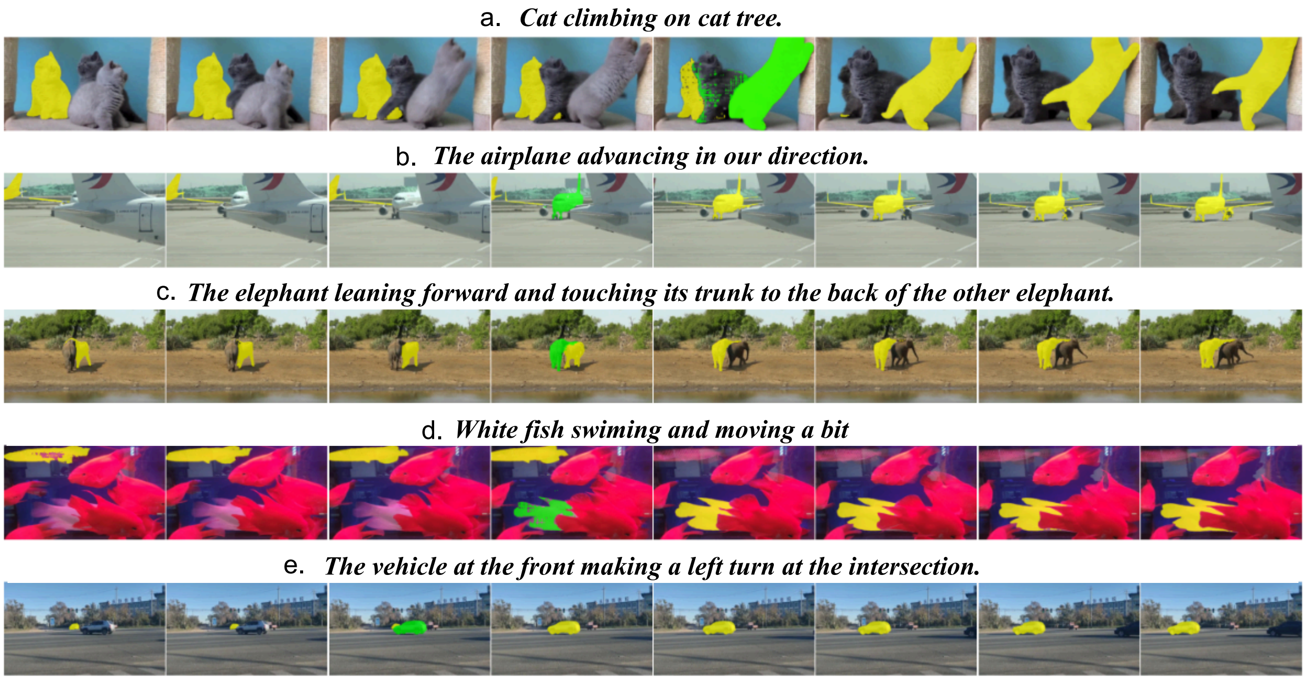


Figure 7. **Correct CME detections.** The plot shows examples where our CME correctly identifies (green masks) the referred object when the action starts unfolding. SAMWISE recognizes that the newly proposed object is more aligned with the query and thus switches its tracking focus in the subsequent frames.

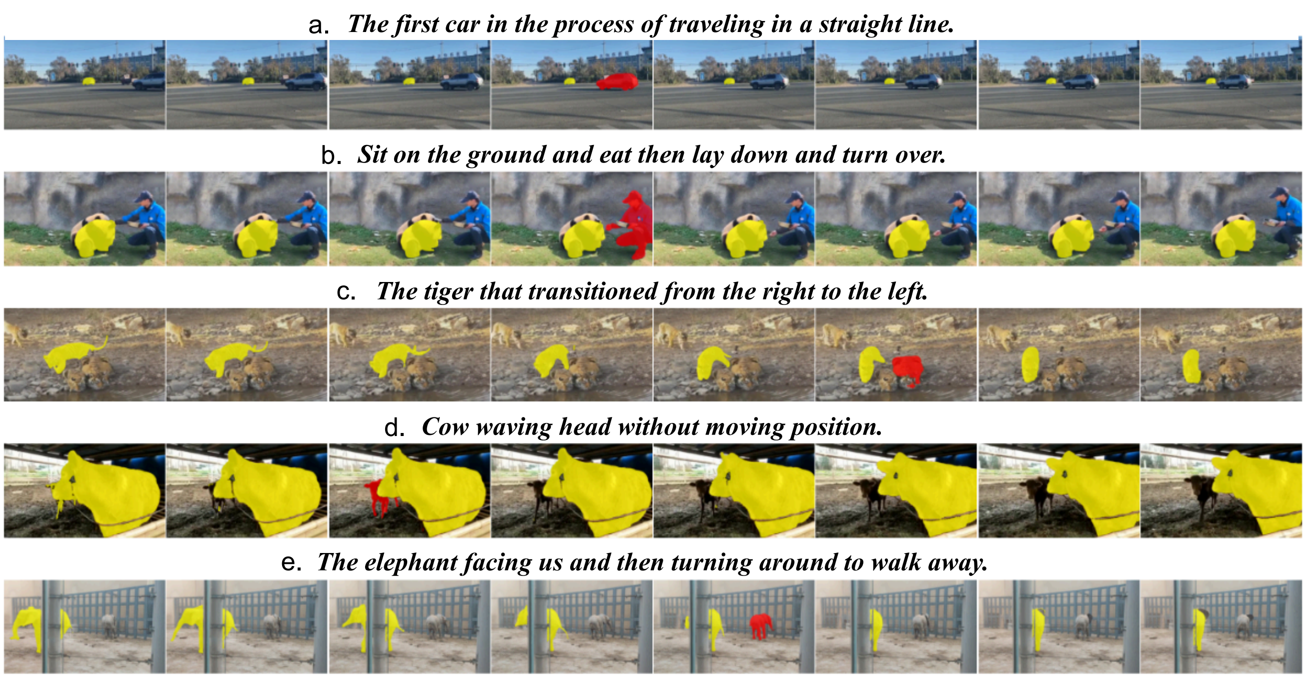
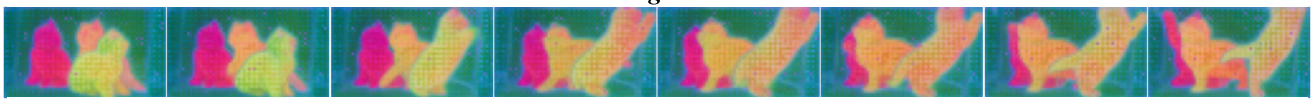


Figure 8. **Incorrect CME detections.**The plot shows examples where our CME provides wrong object proposals (red masks) due to lack of contextual information. In these examples, SAMWISE determines that, when tacking into account past video context, the previously object is more aligned with the query and therefore does not switches its tracking focus.

SAMWISE, by balancing this contextual information with the new proposal, is able to correctly determine that the correct object is the one already subject to tracking. Similarly, in case (d), the CME proposes a different cow, but SAMWISE correctly interprets that *waving head* describes more the foreground cow rather than the new one. In case (b), the referring expression is more ambiguous and lacks a specific subject, leading the CME to propose the human as the target object rather than the panda. However, SAM-

Cat climbing on cat tree.



The airplane advancing in our direction.

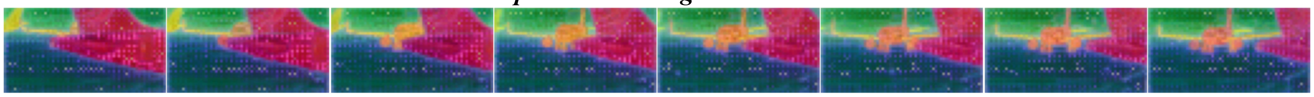


Figure 9. Effect of **Tracking bias.** The figure shows how *memory features* (PCA) reinforce the initial choice, leading to tracking bias and preventing focus to more semantically aligned objects. In the first row, the model fails to shift attention when the correct object begins the relevant action; in the second, it misses the correct object when it appears later in the scene.

Adapter layers				
Layer 1	Layer 2	Layer 3	Params	$\mathcal{J}\&\mathcal{F}$
			0.3 M	45.2
		✓	2.2 M	50.3
	✓	✓	3.5 M	52.1
✓	✓	✓	4.2 M	54.2
Hidden dimensionality				
	64	128	256	384
$\mathcal{J}\&\mathcal{F}$	48.0	52.1	54.2	52.5
Params	1.0 M	2.1 M	4.2 M	8.8 M

Table 5. Top: Ablation on the **Number of Adapters**. *Layer i* indicates the intermediate layer of the Hiera backbone to which we add our CMT modules. Bottom: Effect of **hidden dimensionality** used inside our Cross-Modal Temporal Adapter. All numbers are reported without using our CME module, and CLIP-B as text encoder.

Window	4	6	8	12
$\mathcal{J}\&\mathcal{F}$	51.8	53.9	54.2	54.3

Table 6. **Effect of Window Size**. Ablation on the effect of window size (*i.e.* number of frames processed together in each clip) in our online framework. Numbers combuted on MeViS *valid-u* set, using CLIP-B as text encoder, without CME module.

Method	Visual Encoder	Total Params	MeViS $\mathcal{J}\&\mathcal{F}$	YT-VOS $\mathcal{J}\&\mathcal{F}$	DAVIS $\mathcal{J}\&\mathcal{F}$
TCE-RVOS [15] [WACV'24]	ResNet-50	-	-	59.6	59.4
ReferFormer [44] [CVPR'22]	ResNet-50	176 M	-	58.7	-
OnlineRefer [43] [ICCV'23]	ResNet-50	176 M	-	59.3	57.3
MUTR [47] [AAAI'24]	ResNet-50	190 M	-	61.9	65.3
SAMWISE (w/ CLIP-B)	Hiera-B	150 M	<u>48.3</u>	<u>67.2</u>	<u>68.5</u>
SAMWISE	Hiera-B	202 M	**49.5**	**69.2**	**70.6**

Table 7. Comparison of **SAMWISE** against state-of-the-art RVOS methods on MeViS, Ref-Youtube-VOS and Ref-DAVIS datasets using smaller backbones. **Bold** and <u>underline</u> indicate the two top results.

WISE correctly identifies the panda as the object that aligns best with the query, as it is both *sitting on the ground* and *eating*. In example (e), the CME proposes the wrong elephant, but SAMWISE, by reasoning over the frames, understands that the candidate object does not match the query, which describes an elephant *turning around to walk away*. Finally, in case (c), the described action has occurred in the past. The CME proposes a candidate tiger; however, SAMWISE, by remembering which object actually *transitioned from the right to the left*, refrains from switching its focus.

8. Tracking Bias

We provide additional qualitative examples to further exemplify the effect of *tracking bias*, as visualized in Fig. 9, where we plot the *memory features*. Tracking bias occurs when the model mistakenly focuses on an incorrect object, failing to transition its attention to another, more relevant object once it emerges. This issue is particularly evident in scenarios where the target object becomes distinguishable only after performing a specific action. As shown in the examples, the model initial focus on an object causes it to overlook the presence of another, more semantically aligned instance, even when the latter matches the caption. This behavior stems from biased memory features, which reinforce the initial selection instead of adapting to new cues.

9. Additional Ablations

Number of CMT adapters. In Tab. 5-top we assess how the number of adapters influences performance. Without any adapter (*i.e.* relying only on a learnable MLP to project text prompts), the model achieves a modest $\mathcal{J}\&\mathcal{F}$ of 45.2%. Adding a single adapter at the final layer, *i.e.* on $\mathcal{F}^3$, provides a significant boost of 5.1%. Adding a second adapter, on $\mathcal{F}^2$, further improves performance by $+1.8\%$. Our chosen configuration, with three adapters across the last three layers of feature extractors, achieves the highest performance with a $\mathcal{J}\&\mathcal{F}$ of 54.2%, indicating that multi-layer integration enhances feature refinement, thereby improving segmentation accuracy.

Adapter hidden dimensionality. In Tab. 5-bottom, we evaluate the performance of our CMT adapter with varying hidden dimensionalities. Our configuration, with a channel dimension of 256, achieves strong performance (54.2 $\mathcal{J}\&\mathcal{F}$) while maintaining a lightweight model with only 4.2M trainable parameters. Reducing the channel dimension to 64 or 128 results in a significant drop in performance, with a reduction in $\mathcal{J}\&\mathcal{F}$ of 6.2 and 2.1, respectively. Increasing the hidden dimensionality to 384 leads to a marginal performance drop of -1.7 $\mathcal{J}\&\mathcal{F}$, while doubling the number of trainable parameters (8.8 M).

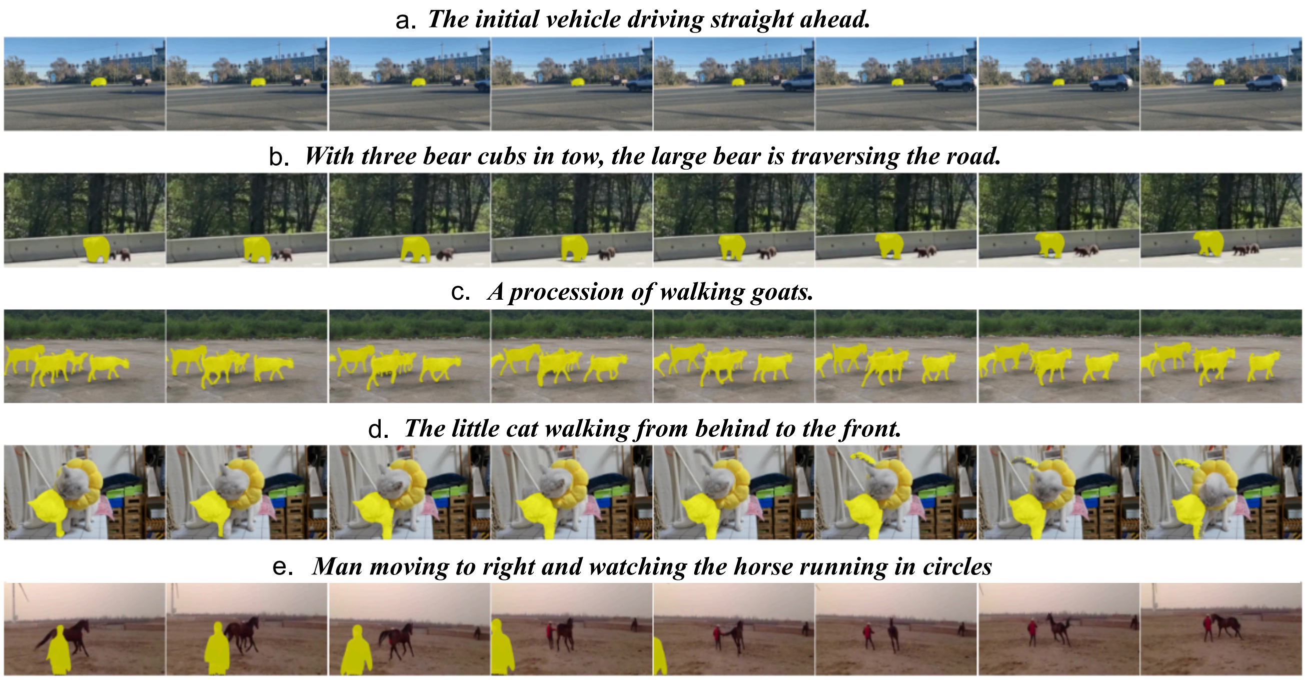


Figure 10. **Qualitative examples from MeViS.** The figure highlights SAMWISE ability to handle challenging RVOS scenarios, including occlusions, multiple instances, and distinguishing between similar objects based on actions and descriptive attributes.

Method	Text Encoder	Referring Image Segmentation		
		RefCOCO	RefCOCO+	RefCOCOg
Large VLM based				
VISA [CVPR'24]	ChatUnivi	72.4	59.8	65.5
RIS Specialist				
MagNet [6] [CVPR'24]	BERT	75.2	66.2	65.4
Ours	RoBERTa	**76.8**	**67.1**	**67.3**

Table 8. Comparison with SOTA for RIS. Results on the val set of the RefCOCO series dataset in terms of mIoU.

Window size. In Tab. 6 we evaluate how the number of frames in each processed clip affects performances. Performances increase with the number of frames, as a larger window allows to better model temporal evolution. Since increasing the window size from 8 to 12 only yields marginal gains, we chose to keep 8 as clip length to better suit an online framework.

Comparison with smaller backbones. In Tab. 7 we compare against previous methods using a smaller backbone, namely a ResNet-50. In this setting we obtain comparable model sizes and higher perfoemance gap.

Referring Image Segmentation. Among our contributions, the design of the HSA and the CME module are tailored to address challenges of referring segmentation in videos. However, the fundamental value of our CMT adapter is that it enables prompting SAM2 with referring expressions, which can be thus easily applied for image-level tasks. In Tab. 8 we evaluate SAMWISE on Referring Image Segmentation benchmarks, comparing against state-of-the-art specialist models, and Large-VLM based. Remarkably, we find that our SAMWISE achieves competitive results also in image-level tasks, showcasing its versatility.

Method	MeViS	Ref-YT-VOS	Ref-DAVIS
	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
G.DINO+SAM2 *1st frame*	37.7	57.5	66.4
G.DINO+SAM2 *All frames*	36.8	56.9	61.2
SAMWISE **(ours)**	**48.3**	**67.2**	**68.5**

Table 9. Comparison of **SAMWISE** against baselines that employ an off-the-shelf grounded detector (GroundingDino) to provide box prompts.

Training time. Training our pipeline requires roughly 150 GB of GPU memory. In our setup, this translates in training on 2 A100 for 18 hours for finetuning on MeViS. Full fine-tuning of SAM2, *i.e.* without our adapters, requires roughly 3 times more GPU memory. For the experiment on full-finetuning (Tab. 4 of main paper), we used 8 A100 for 26 hours.

10. SAMWISE vs naive baselines with SAM2

In Tab. 9, we compare SAMWISE with two baselines utilizing SAM2:

- **GroundingDINO + SAM2 *1st frame***: This approach employs GroundingDINO [26] to identify the referred object in the first frame based on the textual query. The resulting bounding box is then used to prompt SAM2 [36], which tracks the object across the video.
- **GroundingDINO + SAM2 *All frames***: In this baseline, GroundingDINO [26] detects the referred object in each frame using the textual query. The bounding box is then used to prompt SAM2 [36] independently on each frame.

Results indicate that SAMWISE consistently outperforms both baselines. Specifically, it surpasses them by approximately 10%

in $\mathcal{J}\&\mathcal{F}$ on both MeViS [8] and Ref-Youtube-VOS [2], and by 2% and 7% on Ref-DAVIS [18], respectively. GroundingDINO + SAM *1st Frame* baseline heavily relies on the accuracy of the initial bounding box proposal since the object is identified solely in the first frame and then tracked. This dependency leads to suboptimal results, especially when the target object cannot be clearly identified in the first frame, either because the object appears later or the relevant action unfolds as the video progresses. However, this baseline performs relatively well on Ref-DAVIS [18], which contains more static, object-centric videos. The second row shows the results for GroundingDINO + SAM *All Frames*. Although this method allows for frame-by-frame object detection, it does not leverage SAM2 tracking capabilities, leading to poor masks quality. Additionally, limiting reasoning to individual frames causes the model to overlook temporal consistency, often resulting in shifts between objects across frames. In contrast, SAMWISE explicitly models temporal evolution within its features and integrates textual cues without relying on external bounding box proposals. This design enables consistent localization, segmentation, and tracking of the target object.

11. Qualitative results

In Fig. 10, we present qualitative examples from the MeViS dataset that highlight the effectiveness of SAMWISE. These examples cover a range of challenges typical in RVOS. SAMWISE shows strong robustness in dealing with occlusions (case e.), accurately tracking target objects even when they are partially or fully obscured. It also handles situations with multiple instances (case c.), correctly segmenting all relevant objects. Additionally, SAMWISE excels at disambiguating between similar objects by reasoning over both actions (cases a. and b.) and descriptive attributes (case b.), ensuring precise identification of the correct targets based on their behavior and characteristics in the scene.