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

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

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.

1. Training protocol

Following [9], 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:

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

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:

$$\mathcal{Y}_l[t] = \mathcal{D}_{dec}\left(\mathcal{F}, \rho\right) > 0, t = 1..T \tag{2}$$

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

$$y_t = \begin{cases} 1 & if \quad \mathcal{Y}_l[t] \cap \mathcal{Y}_m[t] = 0 \\ 0 & \quad otherwise \end{cases} \tag{3}$$

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:

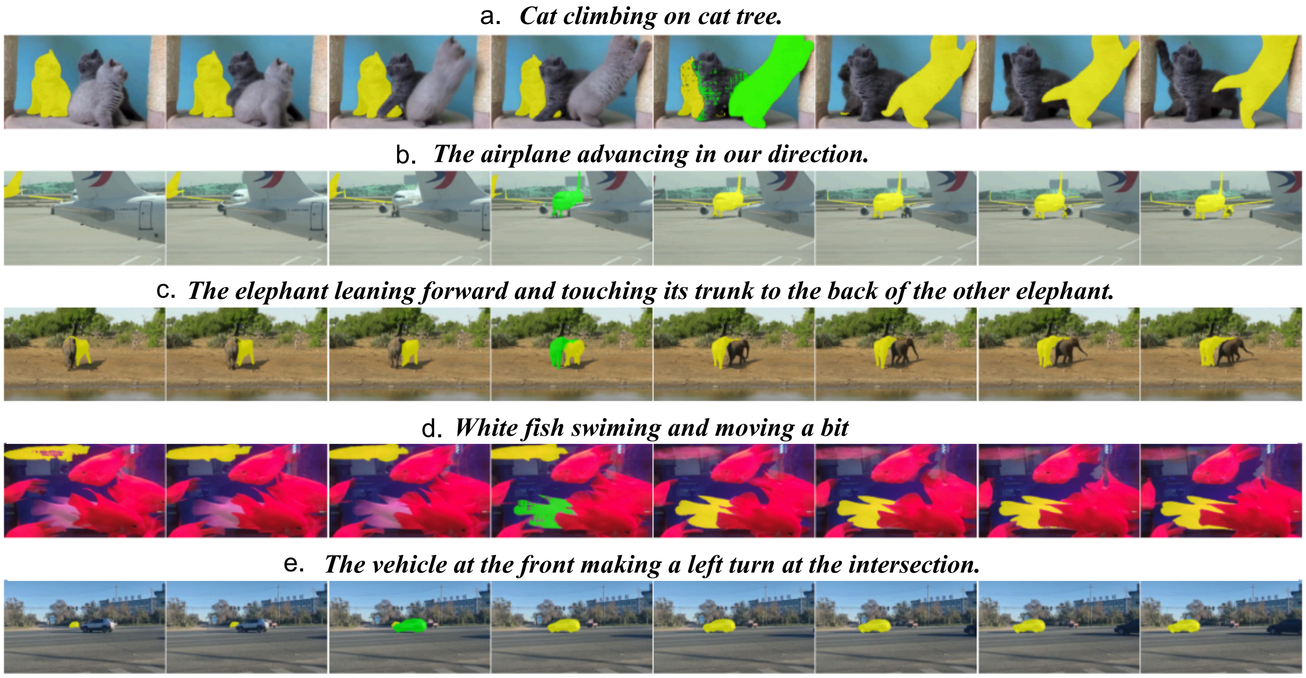
$$\mathcal{L}_{CME} = -\frac{1}{T}\sum_{t=1}^{T} [y_t log(p_{detect}) + (1 - y_t) log(1 - p_{detect})], \tag{4}$$

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

2. CME: Qualitative impact

In this section, we analyze the impact of the Conditional Memory Encoder (CME) within SAMWISE. In Fig. 1 and Fig. 2, 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. 1, 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



a. *Cat climbing on cat tree.*

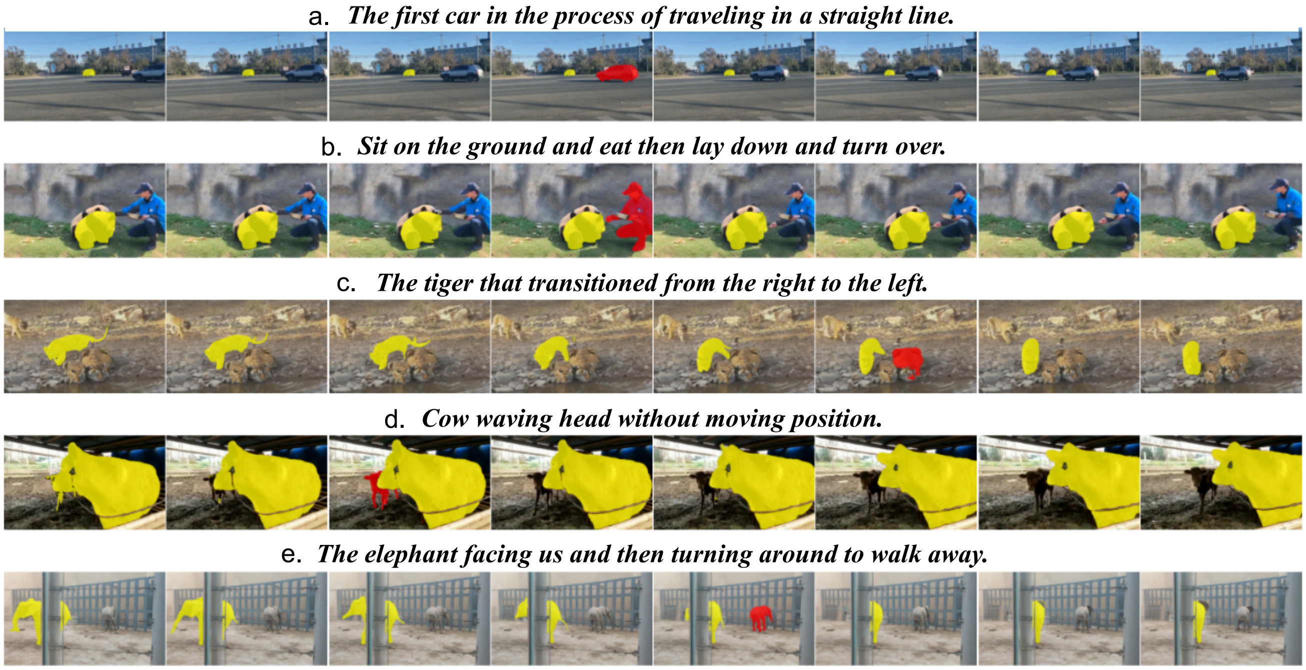
b. *The airplane advancing in our direction.*

c. *The elephant leaning forward and touching its trunk to the back of the other elephant.*

d. *White fish swiming and moving a bit*

e. *The vehicle at the front making a left turn at the intersection.*

Figure 1. **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.



a. *The first car in the process of traveling in a straight line.*

b. *Sit on the ground and eat then lay down and turn over.*

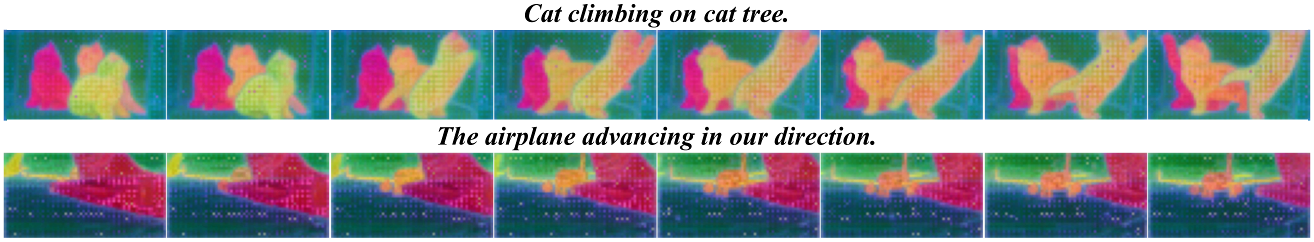
c. *The tiger that transitioned from the right to the left.*

d. *Cow waving head without moving position.*

e. *The elephant facing us and then turning around to walk away.*

Figure 2. **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.

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 be-



Cat climbing on cat tree.

The airplane advancing in our direction.

Figure 3. 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.

comes 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. 2, 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 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, SAMWISE 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.

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 1. 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.

3. Tracking Bias

We provide additional qualitative examples to further exemplify the effect of *tracking bias*, as visualized in Fig. 3, 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.

4. Additional Ablations

Number of CMT adapters. In Tab. 1-top we assess how the number of adapters influences performance. Without

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

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

Table 3. **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.

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. 1-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).

Window size. In Tab. 3 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. 4 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. 5 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	Visual Encoder	Total Params	MeViS	YT-VOS	DAVIS
			$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
TCE-RVOS [4] [WACV'24]	ResNet-50	-	-	59.6	59.4
ReferFormer [9] [CVPR'22]	ResNet-50	176 M	-	58.7	-
OnlineRefer [8] [ICCV'23]	ResNet-50	176 M	-	59.3	57.3
MUTR [10] [AAAI'24]	ResNet-50	190 M	-	61.9	65.3
SAMWISE (w/ CLIP-B)	Hiera-B	150 M	48.3	67.2	68.5
SAMWISE	Hiera-B	202 M	**49.5**	**69.2**	**70.6**

Table 4. Comparison of **SAMWISE** against state-of-the-art RVOS methods on MeViS, Ref-Youtube-VOS and Ref-DAVIS datasets using smaller backbones. **Bold** and underline indicate the two top results.

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 [2] [CVPR'24]	BERT	75.2	66.2	65.4
Ours	RoBERTa	**76.8**	**67.1**	**67.3**

Table 5. Comparison with SOTA for RIS. Results on the val set of the RefCOCO series dataset in terms of mIoU.

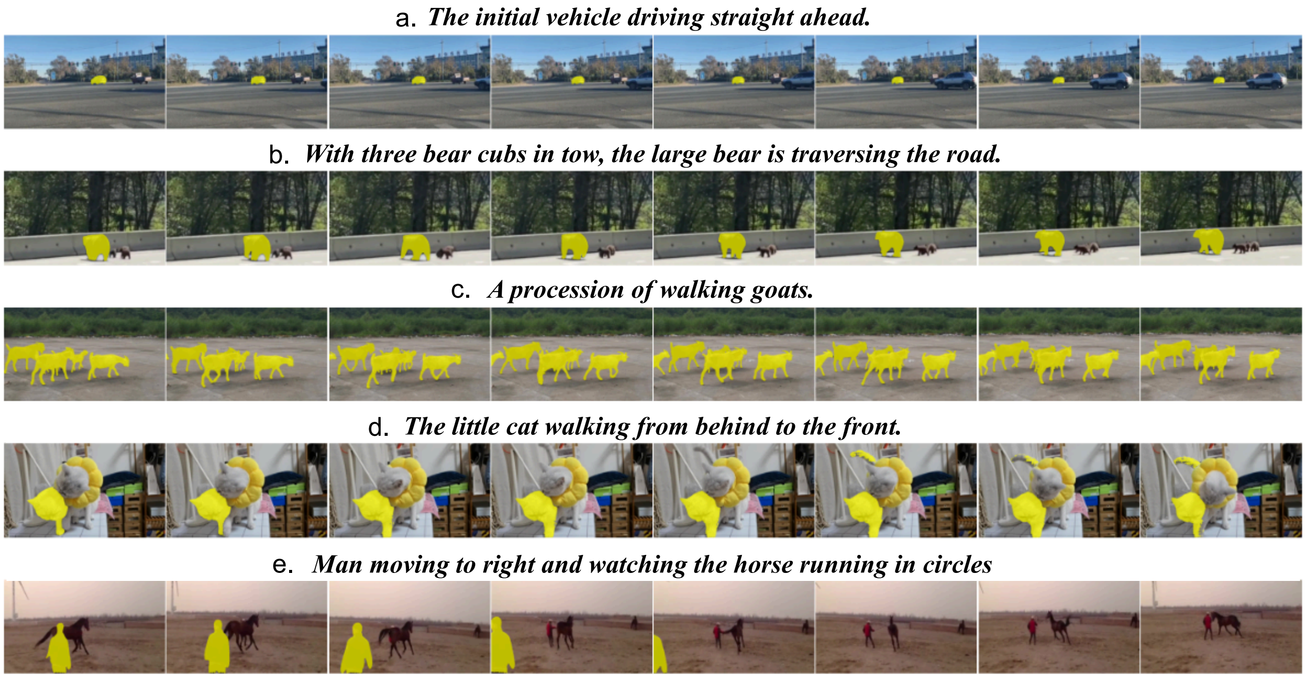
Training time. Training our pipeline requires roughly 150 GB of GPU memory. In our setup, this translates in training on 2 A100 for 18 hours for finetuning on MeViS. Full finetuning of SAM2, *i.e.* without our adapters, requires roughly 3 times more GPU memory. For the experiment on full-finetuning (Tab.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

5. SAMWISE vs naive baselines with SAM2

In Tab. 2, we compare SAMWISE with two baselines utilizing SAM2:

- **GroundingDINO + SAM2 *1st frame***: This approach employs GroundingDINO [6] to identify the referred object in the first frame based on the textual query. The resulting bounding box is then used to prompt SAM2 [7], which tracks the object across the video.
- **GroundingDINO + SAM2 *All frames***: In this baseline, GroundingDINO [6] detects the referred object in each frame using the textual query. The bounding box is then used to prompt SAM2 [7] 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 [3] and Ref-Youtube-



a. *The initial vehicle driving straight ahead.*

b. *With three bear cubs in tow, the large bear is traversing the road.*

c. *A procession of walking goats.*

d. *The little cat walking from behind to the front.*

e. *Man moving to right and watching the horse running in circles*

Figure 4. **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.

VOS [1], and by 2% and 7% on Ref-DAVIS [5], 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 [5], 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.

6. Qualitative results

In Fig. 4, 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.

References

[1] 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. 5

[2] 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

[3] 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. 4

[4] 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. 4

[5] 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. 5

[6] 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. 4

[7] 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. 4

[8] 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. 4

[9] 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, 4

[10] 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. 4