Supplementary material: A Distractor-Aware Memory for Visual Object Tracking with SAM2

Jovana Videnovic*, Alan Lukezic*, Matej Kristan
Faculty of Computer and Information Science, University of Ljubljana, Slovenia
jovanavidenovic10@gmail.com, {alan.lukezic, matej.kristan}@fri.uni-lj.si

In this supplementary document, we provide additional details and experiments that support the findings presented in the main paper. Section 1 presents the evaluation of the proposed distractor-aware memory (DAM) integrated into different model sizes of SAM2.1 and further extended with results for DAM integrated into SAM2 (Section 2). Section 3 reports results on additional bounding box tracking datasets, while Section 4 extends the evaluation to a real-time tracking benchmark. A parameter sensitivity analysis of DAM4SAM is provided in Section 5, followed by additional details on the DiDi (distractor-distilled) dataset in Section 6. Finally, Section 7 presents qualitative comparisons between SAM2.1 and DAM4SAM.

1. Impact of the model size

The segment anything model 2 (SAM2) [11] was originally developed in four model sizes, denoted by tiny (T), small (S), base (B) and large (L). In Table 1 these four model sizes of unchanged SAM2.1 are compared with our DAM4SAM version, presented in the paper on the new DiDi dataset. Results show a clear and consistent performance improvement across all four model sizes. In particular, the tracking quality improves by approximately 6% or 7%, depending on the model size, mostly due to the improved robustness. These results show that the proposed distractor-aware memory generalizes well over various model sizes, demonstrating that the model has not been tuned to the exact SAM2 model.

2. Impact of the model version

This section compares the performance improvements for the two individual SAM versions, i.e., SAM2 and SAM2.1. The SAM2.1 version improves the initial version in handling small and visually similar objects by introducing additional augmentation techniques in training. It also includes improved occlusion handling by training the model on longer frame sequences.

* The authors contributed equally.

Table 1. Comparison of different model sizes on DiDi dataset. The T, S, B, L denote the tiny, small, base and large Hiera backbone sizes, respectively. *Params* denotes number of parameters, while *Acc.* and *Rob.* denote accuracy and robustness, respectively.

	Params	Quality	Acc.	Rob.
SAM2.1-T	39M	0.600	0.697	0.848
DAM4SAM-T	39M	0.642 ↑7%	0.695	0.907
SAM2.1-S	46M	0.630	0.718	0.866
DAM4SAM-S	46M	0.668 ↑6%	0.709	0.930
SAM2.1-B	81M	0.624	0.721	0.856
DAM4SAM-B	81M	0.664 ↑6%	0.709	0.930
SAM2.1-L	224M	0.649	0.720	0.887
DAM4SAM-L	224M	0.694 ↑7%	0.727	0.944

Table 2. Comparison of two SAM model versions: SAM2 and SAM2.1 on DiDi dataset.

	Quality	Accuracy	Robustness
SAM2	0.627	0.723	0.850
DAM4SAM (2)	0.668 ↑7%	0.710	0.929
SAM2.1	0.649	0.720	0.887
DAM4SAM (2.1)	0.694 ↑7%	0.727	0.944

Results are shown in Table 2. To demonstrate the improvement of the 2.1 model version over version 2, we compare the SAM2 with the SAM2.1 on the DiDi dataset, which results in approximately 3.5% improvement in tracking quality. Next, we compare the original SAM2 and the version with our new memory model. The tracking performance improves by 7%, which is well beyond the performance improvement from SAM2 to SAM2.1 and supports the importance of a high-quality memory model and the memory management regime. A similar performance boost (close to 7%) is observed between SAM2.1 and DAM4SAM, which implies complementarity of the new memory model with the baseline method performance imporvements that come from better training. Similarly as in

Section 1, we conclude that the proposed distractor-aware memory is robust to different model versions, demonstrating a consistent improvements in tracking performance on two SAM2 versions.

3. Results on additional benchmarks

In this section, we report the performance of the tracker on three additional benchmarks: OTB100 [12], NFS [6], and Vasttrack [10]. The results are presented in Table 3.

Table 3. State-of-the-art comparison on three additional benchmarks.

	OTB100 (AUC)	NFS (AUC)	Vasttrack (AUC)
MixFormer [3]	70.7 ③	-	39.5
SeqTrack [2]	68.3	66.2 ③	39.6 ③
LORAT [9]	**72.0** ①	66.7 ②	44.0 ②
DAM4SAM	71.7 ②	**68.6** ①	**59.9** ①

4. Real-time performance

We further evaluate the proposed DAM4SAM under real-time tracking constraints. We thus evaluate the tracker on VOT2022-RT [8] challenge[1], which was specifically designed for real-time evaluation. Specifically, VOT challenges are run by VOT toolkits, which manage the real-time constraints. A frame is sent to the tracker, which needs to process it and report the target position at 20FPS frame rate. If the tracker is not able to process the frame in time, the prediction from the previous frame is used as the estimate for the current frame and next frame is sent to the tracker. Such a setup simulates actual real-time scenario, which is much more realistic than reporting just the average tracking speed. The tracking performance is measured using standard VOT2022 measures [8]: the primary measure expected average overlap (EAO), and two auxiliary measures, i.e., accuracy and robustness.

The results in Table 4 show that the proposed DAM4SAM (L model size) outperforms all trackers that participated in VOT2022-RT challenge. In particular, it outperforms the challenge winner by 4% in EAO demonstrating excellent real-time performance. These results show that the proposed distractor-aware memory adds only a small computational overhead, yet bringing remarkable robustness capabilities and making it useful for real applications.

[1]SAM2.1 and DAM4SAM were evaluated on the machine with the AMD EPYC 7763 64-Core 2.45 GHz CPU and Nvidia A100 40GB GPU.

Table 4. Real-time performance on VOT2022-RT challenge. The challenge winner is marked by 🏆.

	EAO	Accuracy	Robustness
MS_AOT 🏆	0.610 ③	0.751 ②	0.921 ③
OSTrackSTS	0.569	0.766 ①	0.860
SRATransTS	0.547	0.743 ③	0.866
SAM2.1	0.614 ②	0.722	0.922 ②
DAM4SAM	0.635 ①	0.717	0.942 ①

5. Sensitivity to threshold values

We analyze the sensitivity of the proposed DAM4SAM to the exact value of the manually determined parameters. In particular, we focus on thresholds defined in Distractor-resolving memory (DRM, Section 3.2.2). Experiments were conducted on the VOT2022 [8] dataset using the unsupervised experiment to ensure fast execution and compute the average overlap (AO) as the performance measure.

Firstly, we examined the impact of different memory splits, specifically 2 frames in DRM and 4 frames in RAM, and vice versa. The results showed no significant changes, with performance drops of up to 0.3%, indicating robustness to memory allocation variations.

Next, we analyzed the influence of the threshold for the ratio between the target and alternative predicted masks, i.e. $\Theta_{anc} = 0.7$. This ratio is used for preemptive update upon distractor detection and is thus crucial for our distractor-aware memory. As shown in Figure 1, tracking performance remains highly stable for a wide range of $\Theta_{anc} \in [0.6, 0.9]$ demonstrating the robust design of the tracker.

Similarly, we evaluated the IoU score threshold ($\Theta_{Iou} = 0.8$), which determines if a predicted mask is reliable for distractor testing. Results in Figure 1 show that tracking performance remains consistent for a wide range of parameters ($\Theta_{IoU} \in [0.5, 0.8]$) scoring almost identical AO.

The mask area threshold was also tested, i.e. $\Theta_{area} = 0.2$. This threshold is used to determine the tracking stability and is together with the Θ_{IoU} a necessary condition to trigger the DRM update.

Finally, we tested the impact of the window size used in the median calculation, starting with an initial value of $N_M = 10$ and exploring a range of values three times larger and smaller. The performance variations were up to 1.5% AO, indicating stability.

Sensitivity analysis for parameters Θ_{anc}, Θ_{IoU}, Θ_{area} and N_M across a wide range of thresholds is summarized in Figure 1. DAM4SAM demonstrates stable tracking performance, confirming that it is not sensitive to the exact value of these parameters.

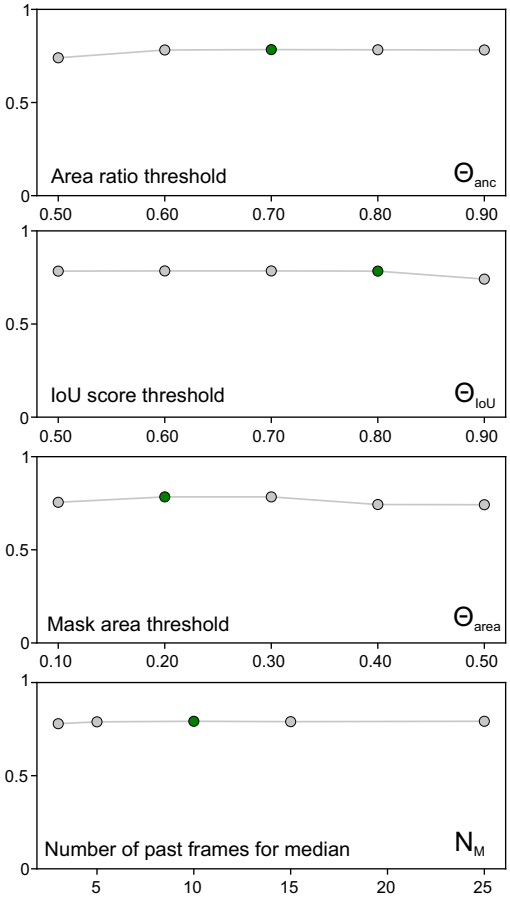

Figure 1. Sensitivity of the DAM4SAM to different values of parameters: Θ_{anc}, Θ_{IoU} and Θ_{area} and N_M. Experiments were done on VOT2022 using average overlap as the performance measure. The selected parameter value is marked by a green circle.

6. DiDi dataset statistics

In this section we provide additional information about the distractor-distilled dataset DiDi construction. In particular, number of sequences from each source dataset is given in the following:

- LaSoT [4]: 86 sequences
- UTB-180 [1]: 56 sequences
- VOT2022-ST [8]: 20 sequences
- VOT2022-LT [8]: 7 sequences
- VOT2020-LT [7]: 6 sequences
- GoT10k [5]: 4 sequences
- VOT2020-ST [7]: 1 sequence

- Total: 180 sequences (274,882 frames)

7. Qualitative analysis

Figure 2 presents a qualitative comparison between the baseline SAM2.1 and the proposed DAM4SAM on four video sequences. In the first row a zebra is tracked with other zebras in its vicinity. When the zebra is partially occluded, SAM2.1 drifts to the wrong zebra and starts to track it, while DAM4SAM tracks only the visible part of the target during occlusion and stays on the selected zebra until the end of the sequence.

In the second row of Figure 2, the baseline SAM2.1 tracker successfully tracks the bus until the full occlusion and fails to re-detect it after the re-appearance. This failure occurs due to the too frequent memory updates when target is occluded and is successfully addressed with the proposed memory update in DAM4SAM.

The third row in Figure 2 shows tracking of a flamingo's head. The baseline SAM2.1 tends to jump on the bird's beak or extend to the whole body, since it prefers to segment the regions with so-called high objectness (i.e., regions with well-defined edges). The proposed DAM4SAM successfully tracks the flamingo's head even if the edge between the head and the neck is not clearly visible. In this case, part of the neck is segmented by an alternative mask and thus detected as a distractor. Updating the distractor resolving memory (DRM) using such *critical* frames results in a more stable and accurate tracking.

A similar effect is demonstrated in the fourth row of Figure 2, where a fish similar to the tracked fish occludes it and causes SAM2.1 to jump to it. On the other hand, DAM4SAM successfully detects such critical frames, updates the DRM and avoids the tracking failure.

References

[1] Basit Alawode, Yuhang Guo, Mehnaz Ummar, Naoufel Werghi, Jorge Dias, Ajmal Mian, and Sajid Javed. Utb180: A high-quality benchmark for underwater tracking. In *Proc. Asian Conf. Computer Vision*, pages 3326–3342, 2022. 3

[2] Xin Chen, Houwen Peng, Dong Wang, Huchuan Lu, and Han Hu. Seqtrack: Sequence to sequence learning for visual object tracking. In *Comp. Vis. Patt. Recognition*, pages 14572–14581, 2023. 2

[3] Yutao Cui, Cheng Jiang, Limin Wang, and Gangshan Wu. Mixformer: End-to-end tracking with iterative mixed attention. In *Comp. Vis. Patt. Recognition*, pages 13608–13618, 2022. 2

[4] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. Lasot: A high-quality benchmark for large-scale single object tracking. In *Comp. Vis. Patt. Recognition*, 2019. 3

[5] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. In *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 1562–1577, 2018. 3

[6] Hamed Kiani Galoogahi, Ashton Fagg, Chen Huang, Deva Ramanan, and Simon Lucey. Need for speed: A benchmark for higher frame rate object tracking. In *Int. Conf. Computer Vision*, pages 1125–1134, 2017. 2

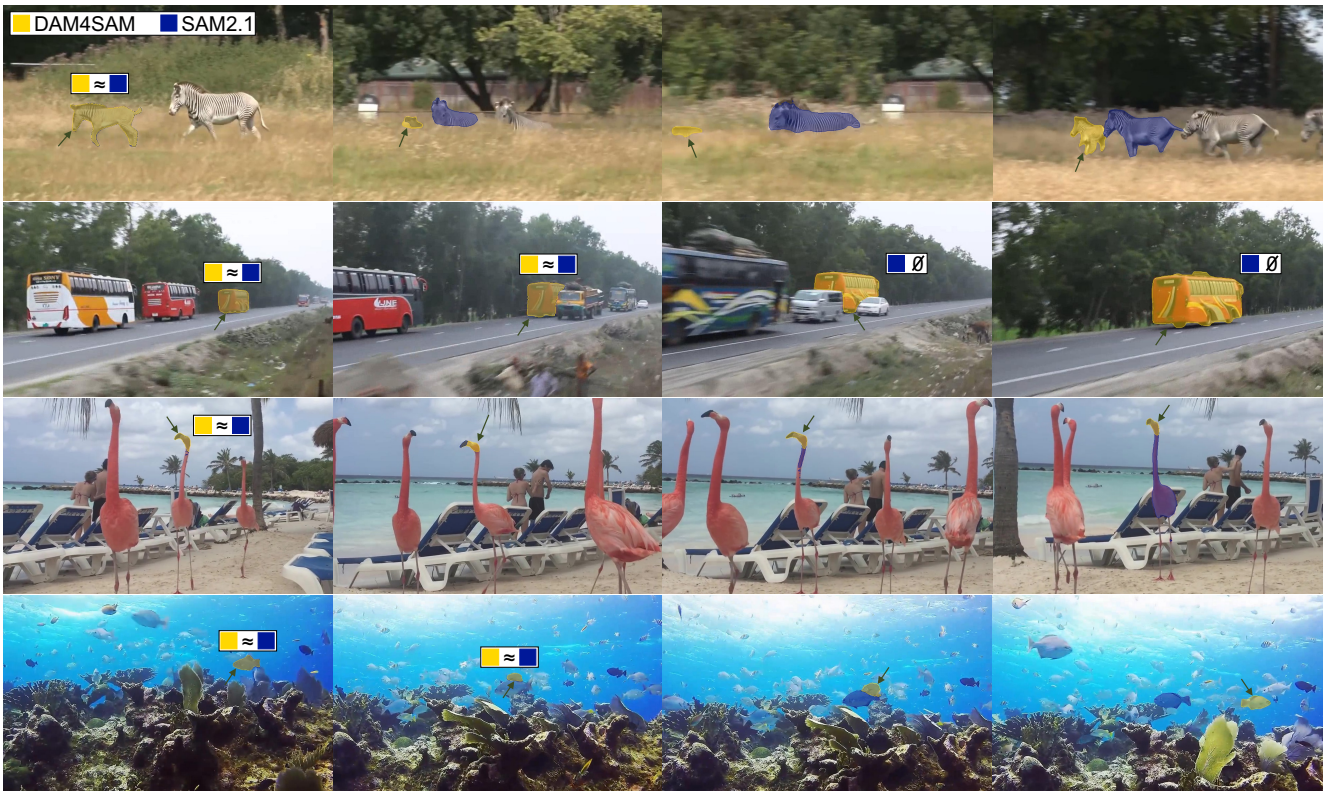

Figure 2. Qualitative comparison of the baseline SAM2.1 (blue) and the proposed DAM4SAM (yellow). The symbol ≈ denotes approximately identical outputs and ∅ denotes an empty prediction (i.e, mask with all-zeros). Tracked object is denoted with a green arrow.

[7] Matej Kristan, Aleš Leonardis, Jiří Matas, Michael Felsberg, Roman Pflugfelder, Joni-Kristian Kämäräinen, Martin Danelljan, Luka Čehovin Zajc, Alan Lukežič, Ondrej Drbohlav, et al. The eighth visual object tracking VOT2020 challenge results. In *Proc. European Conf. Computer Vision Workshops*, pages 547–601, 2020. 3

[8] Matej Kristan, Aleš Leonardis, Jiří Matas, Michael Felsberg, Roman Pflugfelder, Joni-Kristian Kämäräinen, Hyung Jin Chang, Martin Danelljan, Luka Čehovin Zajc, Alan Lukežič, et al. The tenth visual object tracking vot2022 challenge results. In *Proc. European Conf. Computer Vision Workshops*, pages 431–460, 2022. 2, 3

[9] Liting Lin, Heng Fan, Zhipeng Zhang, Yaowei Wang, Yong Xu, and Haibin Ling. Tracking meets lora: Faster training, larger model, stronger performance. In *Proc. European Conf. Computer Vision*, pages 300–318, 2025. 2

[10] Liang Peng, Junyuan Gao, Xinran Liu, Weihong Li, Shaohua Dong, Zhipeng Zhang, Heng Fan, and Libo Zhang. Vasttrack: Vast category visual object tracking. In *Neural Inf. Proc. Systems Datasets and Benchmarks Track*, 2024. 2

[11] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. In *Proc. Int. Conf. Learning Representations*, 2025. 1

[12] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. In *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 1834–1848, 2015. 2