

MambaVLT: Time-Evolving Multimodal State Space Model for Vision-Language Tracking

Supplementary Material

A. Additional Implementation Details

A.1. Target Discrimination Head

We proposed the target discrimination head to exploit the discriminative information from the reference feature to locate the target. The head predicts the target box based on the search token with the highest target score. Herein the target score is computed as the product of the template-search similarity and foreground-background classification score. For contrastive learning, we treat the patches inside the target box as positive and others as negative. We will first extract a unified reference token T_{uni} . In the BBOX and NL tasks, T_{uni} is extracted based on the template feature and the language feature, respectively. In the NL&BBOX task, we perform mean pooling on the template and language features to obtain T_{uni} . Subsequently, we apply two separate linear layers to transform T_{uni} into target token T_{tgt} and background token T_{bgd} . T_{tgt} and T_{bgd} are used to compute the similarity with the search region feature, generating the target score and the background score for each search region token. These scores are then utilized to select the final output bounding box. We employ the binary cross-entropy target score map loss \mathcal{L}_{tgt} , whose groundtruth is generated based on the bounding box, as the contrastive learning loss for target discrimination. Finally, the prediction with a target score exceeding the threshold will be used to update the template video clip. The threshold is set to 0.8.

A.2. Training Settings

In the intra-video contrastive learning, we utilize 1 positive sample and 8 negative samples. In the inter-video contrastive learning, we utilize 1 positive sample and 224 negative samples. The multimodal contrastive learning is performed in the last two layers of the preliminary feature extraction stage and every module of the time-evolving multimodal fusion module. For training datasets, the MGIT dataset consists of three subsets: train, val, and test, each containing 105, 15, and 30 videos, respectively. We use its train subset to train our model. Additionally, MambaVLT is trained using mixed reference inputs, including vision-only, language-only, and a combination of both. For GOT-10K, which lacks textual annotations, we use the dataset as training data with only vision references.

B. More Experimental Results

B.1. Efficiency Analysis

We employ the number of model parameters and floating-point operations (FLOPs) to evaluate the model size and computational complexity [22, 34]. Table A presents a detailed comparison with state-of-the-art trackers in parameter count, FLOPs, and FPS. Among the trackers, MambaVLT has the fewest parameters and achieves the lowest FLOPs. Nevertheless, we observe that the lowest FLOPs do not lead to the fastest inference speed for MambaVLT. This is because the core component of our feature extractor and multimodal encoder, *i.e.*, Multi-directional Selective Scan, lacks sufficient parallelization in implementation, leading to longer inference time despite lower FLOPs in these two modules, as shown in Table A. As the search region size increases, the computational complexity of MambaVLT grows slowly, while that of UVLTrack shows a rapid quadratic growth trend. JointNLT reduces computational complexity by introducing the Swin Transformer [35].

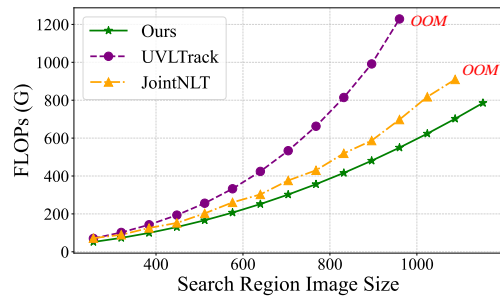


Figure A. Computational complexity comparison with different search region image scales. OOM represents the computation cost is out of memory.

To achieve a balance between tracking accuracy and tracking FPS, we propose a lightweight variant of MambaVLT, termed MambaVLT-light, whose tracking FPS is **50** on RTX 2080 Ti. In contrast to MambaVLT, the MambaVLT-light reduces the depth of the four stages in the vision encoder to 1, 1, 2, and 1, substitutes the modality-guided bidirectional scan with a single bidirectional scan, and removes the Selective Locality Enhancement Block. The performance is shown in Table B. The FLOPs of MambaVLT-light with three templates is **19.28 G**, which is much lower than others.

Table A. Analyses on Computational Performance. ‘# z’ denotes the template number. All models are tested on RTX2080Ti.

Method	# z	Feat. Extraction	Multimodal Encoder	Decoder	Pred. Head	FPS	Params	FLOPs
JointNLT	1	18 ms / 81 G	3 ms / 6 G	5 ms / 1.5 G	2 ms / 1.5 G	35	193 M	90 G
MMTrack	1	23 ms / 105 G	0.5 ms / 0.2 G	4.5 ms / 1 G	-	36	217 M	106 G
UVLTrack-B	1	9 ms / 36 G	6 ms / 31 G	-	3 ms / 4 G	55	169 M	71 G
MambaVLT [†]	1	11 ms / 25 G	14 ms / 21 G	6 ms / 2 G	2 ms / 3 G	30	149 M	51 G
MambaVLT	3	11 ms / 35 G	15 ms / 29 G	6 ms / 3 G	2 ms / 3 G	29	149 M	70 G

Table B. Comparison of MambaVLT-light and other methods in the NL&BBOX task.

Tracker	TNL2K		OTB99	
	AUC	Prec	AUC	Prec
UVLTrack-B [37]	63.1	66.7	69.3	89.9
MambaVLT	66.5	69.9	72.2	94.4
MambaVLT-light	62.5	63.1	70.5	91.2

B.2. More Results of Semi-reference-free Tracking

To analyze the effectiveness of state space memory, we design the semi-reference-free tracking paradigm, in which the reference data (language or initial bounding box) is used by tracker **only** in the first frame. From the second frame, the tracker needs to locate the target without explicitly using the reference data. The main challenge is extracting and memorizing target information based on the reference input in the first frame. We conduct the SRF-based experiments **without retraining**. As shown in Figure C, MambaVLT is able to track the target even without reference data after the first frame, demonstrating the effectiveness of the state space memory in target information retention.

Figure B illustrates the results of SRF on three sequences where the targets undergo both significant appearance changes and occlusions. SRF tracks the targets stably, suggesting that state space memory effectively models the varying target states and facilitates tackling these challenges. We will provide more analyses.

B.3. Analyses of the MS module

The modality weights in MSM are dynamically predicted based on the textual and visual tokens. To analyze its effect, we conducted experiments by directly setting the modality weights to different fixed values. Table C shows that the fixed weight settings lead to performance drops, validating the effectiveness of our dynamic mechanism.

B.4. Extensive Experiments on MGIT

In the MGIT [24] dataset, in addition to the language descriptions of the targets in the first frames, it also provides corresponding natural language specifications of the targets in certain subsequent frames. Therefore, **without retraining** the model, we update the language information during inference with the latest natural language description,

Table C. Comparison between dynamic and fixed modality weight.

Variants	NL&BBOX		
	AUC	PRE	N PRE
$w_l=0.25; w_v=0.75$	64.2	66.3	88.8
$w_l=0.5; w_v=0.5$	65.4	67.8	89.6
$w_l=0.75; w_v=0.25$	65.5	67.8	89.7
Dynamic Modality Adjustment	66.5	69.9	90.9

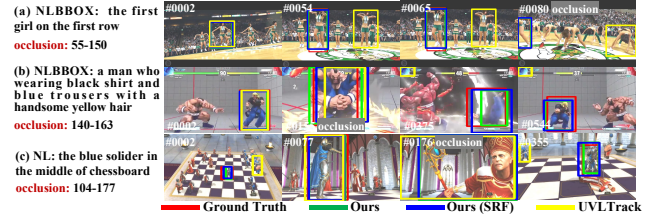


Figure B. Results of SRF in 3 challenging sequences.

Table D. Extensive experiments of natural language updating on MGIT. * denotes the results obtained by updating the language descriptions in the inference process without retraining the model.

Tracker	AUC	Prec	N prec	SR _{IoU}
BBOX				
MambaVLT	65.7	51.6	72.9	60.4
NL				
MambaVLT	64.6	50.3	71.2	58.7
MambaVLT*	65.4	51.5	72.3	60.1
NL&BBOX				
MambaVLT	69.9	58.9	77.9	67.9
MambaVLT*	70.2	59.1	79.0	68.6

to evaluate whether the state space memory can update the target feature based on the new description, thereby improving tracking accuracy. Notably, all the experiments are conducted using the action granularity of the MGIT dataset.

We introduce a new metric, success rate (SR), to align with the official experiments in the MGIT dataset. The prediction with the intersection over union IoU that is higher than the threshold θ_s is regarded as a successful prediction. SR denotes the percentage of successfully tracked frames. According to Table D, in the NL and NL&BBOX tasks, particularly the NL task, performance improves when natural language information is updated with the latest descriptions.

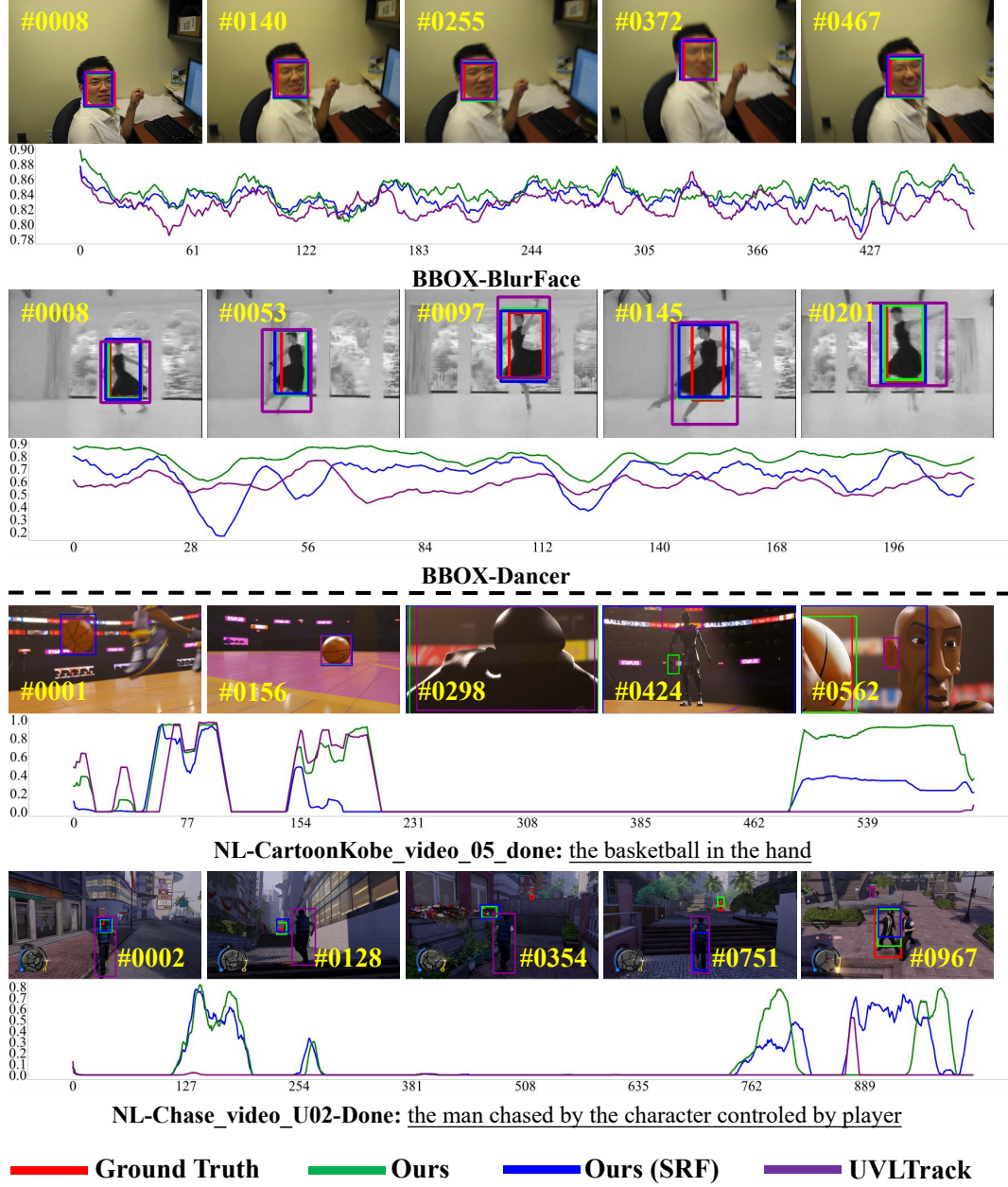


Figure C. Effectiveness analysis of the time-evolving state space memory in BBOX and NL tasks. Under the **semi-reference-free** setting, the state space memory can still effectively extract and retain target features for accurate target localization compared to MambaVLT and UVLTrack using the standard tracking settings, validating the effectiveness of the state space memory.

This demonstrates that the state space memory is capable of modeling varying information.

B.5. Qualitative Results

Table E presents the detailed results and the corresponding reference data of several sequences for robustness evaluation. The results indicate that MambaVLT has strong robustness against interference from the initial reference information, because our model can adaptively update the

reference features and dynamically weigh multimodal information for modality selection. In Figure D, we evaluate MambaVLT on six sequences characterized by drastic target variations. MambaVLT can still track the targets accurately, which demonstrates the introduction of time-evolving state space memory can help the model to retain long-term target features to update reference features for modeling long-term target variations adaptively.

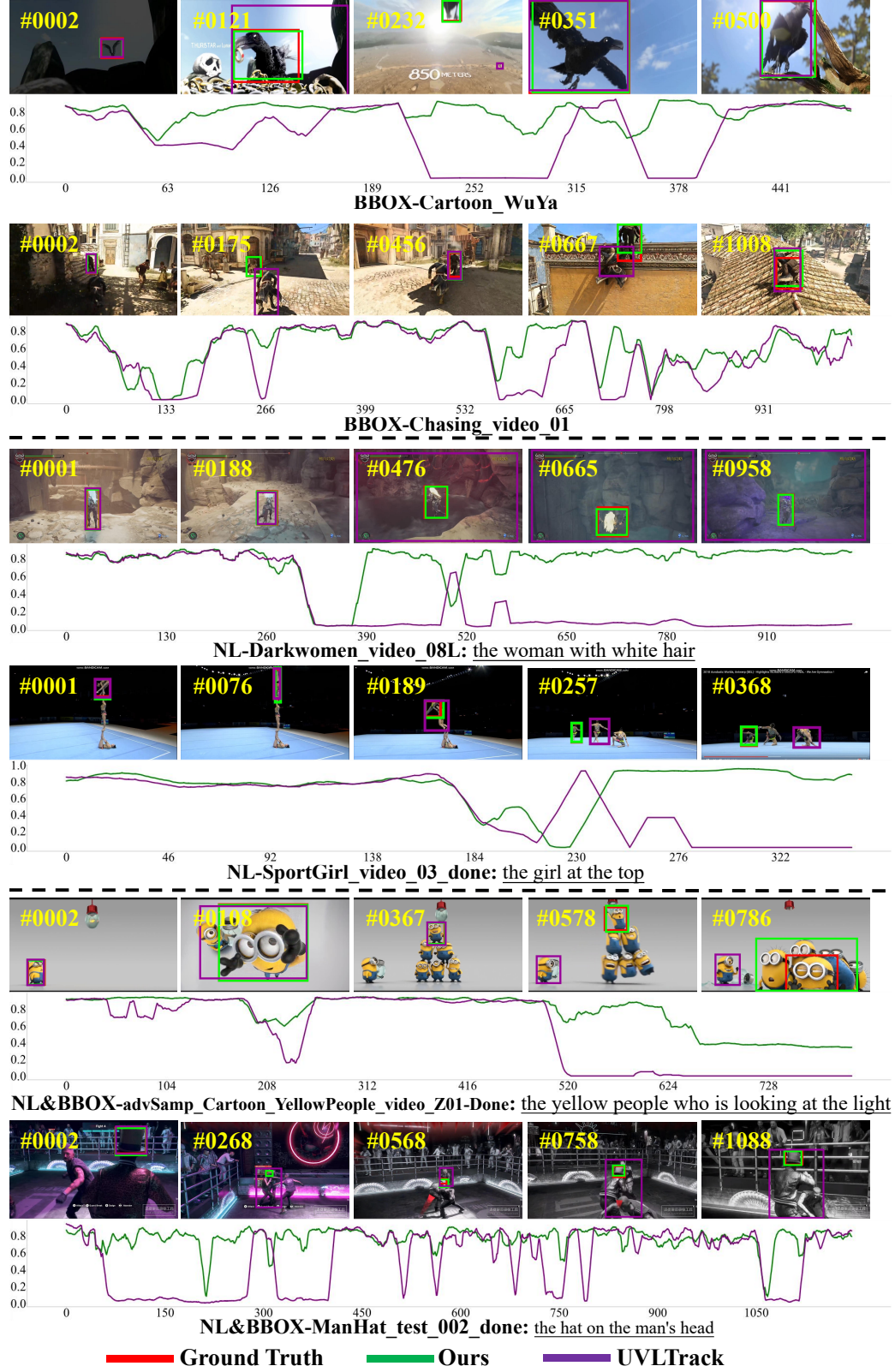


Figure D. Visualized results of the MambaVLT and the UVLTrack method on six challenging sequences with **drastic changes**. Our MambaVLT performs well with the aid of the time-evolving state space memory for long-term target feature retention and adaptive reference feature update, while the UVLTrack with discrete context prompts struggles with these sequences.

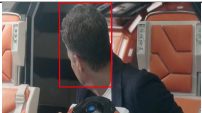



Task	Initial Frame	Language Description	Interference	UVLTrack	MambaVLT
BBOX		-	Distractor	56.1%	70.2%
BBOX		-	Viewpoint Change	64.2%	73.4%
BBOX		-	Occlusion	52.9%	66.1%
NL		we want to track a man holding an umbrella under street lamp	Low Light	1.1%	66.3%
NL		the fourth fish from right to left	Distractor	16.7%	42.5%
NL		the player wears white suit with twenty-three on this back	Distractor	4.0%	64.4%
NL&BBOX		the rightmost pedestrian in white	Distractor	8.5%	75.4%
NL&BBOX		the man on the bottom right corner	Low Light	16.3%	72.7%
NL&BBOX		the person on the corridor	Occlusion	13.4%	41.1%

Table E. Robustness evaluation in terms of AUC score on several challenging sequences. It demonstrates that the MambaVLT significantly improves vision-language tracking performance by updating reference features adaptively in cases where distractions exist between the target and reference information.