

Tracking and Segmenting Anything in Any Modality

Tianlu Zhang¹, Qiang Zhang², Guiguang Ding³, Jungong Han¹ *

¹Department of Automation, Tsinghua University

²School of Mechano-Electronic Engineering, Xidian University

³School of Software, Tsinghua University.

Abstract

Tracking and segmentation play essential roles in video understanding, providing basic positional information and temporal association of objects within video sequences. Despite their shared objective, existing approaches often tackle these tasks using specialized architectures or modality-specific parameters, limiting their generalization and scalability. Recent efforts have attempted to unify multiple tracking and segmentation subtasks from the perspectives of any modality input or multi-task inference. However, these approaches tend to overlook two critical challenges: the distributional gap across different modalities and the feature representation gap across tasks. These issues hinder effective cross-task and cross-modal knowledge sharing, ultimately constraining the development of a true generalist model. To address these limitations, we propose a universal tracking and segmentation framework named SATA, which unifies a broad spectrum of tracking and segmentation subtasks with any modality input. Specifically, a Decoupled Mixture-of-Expert (DeMoE) mechanism is presented to decouple the unified representation learning task into the modeling process of cross-modal shared knowledge and specific information, thus enabling the model to maintain flexibility while enhancing generalization. Additionally, we introduce a Task-aware Multi-object Tracking (TaMOT) pipeline to unify all the task outputs as a unified set of instances with calibrated ID information, thereby alleviating the degradation of task-specific knowledge during multi-task training. SATA demonstrates superior performance on 18 challenging tracking and segmentation benchmarks, offering a novel perspective for more generalizable video understanding.

Introduction

Video, as a crucial medium for information dissemination, has emerged as a dominant constituent of internet traffic. This prevalence has notably spurred the advancement of video understanding technologies. At present, video understanding has witnessed substantial expansion and now encompasses a wide range of tasks, including action recognition, video segmentation, object tracking, and etc. Among these, object tracking and segmentation are dedicated to establishing instance-level or pixel-level correspondences

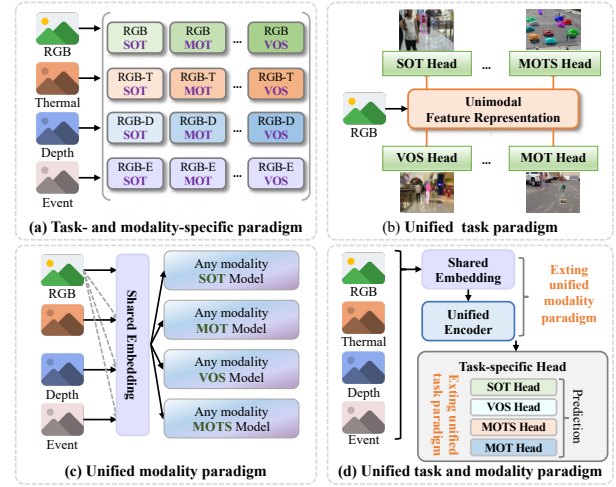


Figure 1: Illustration of existing tracking and segmentation paradigm. (a) Task- and modality-specific paradigm. (b) Unified task paradigm. (c) Unified modality paradigm. (d) Unified task and modality paradigm obtained by combining existing unified task methods and unified modality models.

across frames, thereby laying the groundwork for tackling video tasks.

Over the years, the object tracking and segmentation have developed into multiple subtask branches, including Single Object Tracking (SOT) (Huang et al. 2019), Multiple Object Tracking (MOT) (Li et al. 2024c), Video Object Segmentation (VOS) (Ravi et al. 2024) and Multi-Object Tracking and Segmentation (MOTS) (Yan et al. 2023). In addition to the widely used RGB cameras, various sensors have been introduced to enhance the performance of tracking and segmentation in complex scenarios, e.g., depth, thermal, and event data. For a long period, research in these typical tracking and segmentation subtasks has adopted a task- and modality-specific paradigm, i.e., designing specialized architectures and loss functions to cater to the unique requirements of different subtasks (e.g., SOT, MOT and VOS) and fixed input modalities (e.g., RGB, RGB-T and RGB-D), as shown in Fig. 6 (a). The divergent setups require customized methods with carefully designed architectures and hyper-parameters, leading to complex training and redundant parameters.

*Corresponding author: jghan@tsinghua.edu.cn.

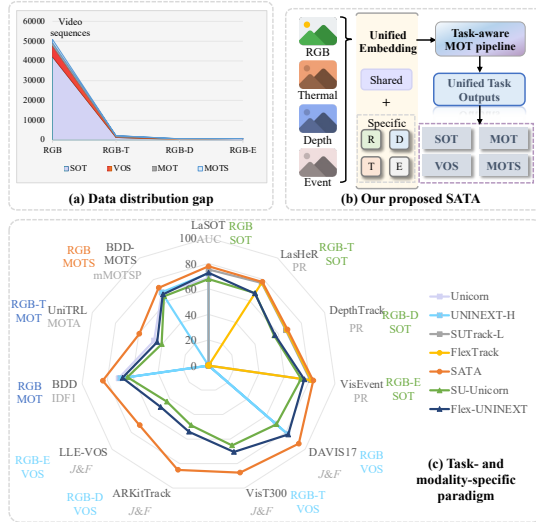


Figure 2: Analysis of data distribution gap and comparison of the proposed SATA with existing strategies. (a) Statistical overview of the data distribution gap during model training. (b) Overview of the proposed SATA framework. (c) Our SATA v.s. the existing methods on 11 challenging benchmarks. Here, SU-Unicorn denotes the combination of SU-Track (Chen et al. 2025) and Unicorn (Yan et al. 2022), Flex-UNINEXT denotes the combination of FlexTrack (Tan et al. 2025a) and UNINEXT (Yan et al. 2023).

To mitigate this issue, some recent works have explored the possibility of establishing general models from two perspectives: unified task paradigm and unified modality paradigm. Specifically, as shown in Fig. 6 (b), the unified task paradigm (Yan et al. 2023; Chen et al. 2025; Wang et al. 2024) breaks through the isolation of task domain knowledge and general knowledge, establishing the implicit collaboration between different tasks. Meanwhile, as shown in Fig. 6 (c), the unified modality methods (Wu et al. 2024; Chen et al. 2025) consolidate the input of any modality using a unified model with shared parameters. The above two methods respectively establish unified modality representation learning and unified task feature space modeling, thereby reducing the complexity of model design and the need for extensive hyper-parameter tuning across various input modalities and sub-task combinations.

Recently, due to the potential to realize Artificial General Intelligence (AGI), these unified models have drawn great attention. A natural idea is: combining the above two paths to achieve a more generalizable video tracking and segmentation model capable of handling any input modality and supporting multi-task inference, as shown in Fig. 6 (d). While this strategy is conceptually straightforward and partially effective, it overlooks the distributional discrepancies across modalities and the representational gaps between tasks—factors that limit the generalist model’s ability to fully leverage cross-modal and cross-task knowledge.

Specifically, the distribution gap in multi-modal data arises not only from discrepancies in cross-modal information but also from differences in their underlying represen-

tations. Most existing unified modality approaches (Chen et al. 2025; Tan et al. 2025a) have attempted to learn a cohesive embedding across diverse input modalities, but the effectiveness of modality-specific clues have not been fully exploited. Similarly, while generic representations can be learned with the unified task approaches, they are still constrained by elaborated-designed task-specific heads and isolated multi-stage training strategies across different downstream tasks (Yan et al. 2022, 2023; Wang et al. 2024). These fragmented architectures and learning paradigms restrict the model’s ability to acquire truly generalizable knowledge. Moreover, as shown in Fig. 7 (a), inconsistencies in the quality and scale of training data across different multi-modal subtasks further exacerbate task and modality biases in unified models.

With this in mind, we present SATA, a unified tracking and segmentation framework that models a unified representation of arbitrary modalities and approaches a broad spectrum of tracking and segmentation subtasks as MOT inference task conditioned on different priors, as shown in Fig. 7 (b). Technically, we introduce a Decoupled Mixture-of-Expert (DeMoE) mechanism to decouple the unified representation learning into parallel modeling of modality-common and modality-specific knowledge, thus achieving a more comprehensive representation of any modality input. In addition, unlike previous unified task models that employ several task-specific heads, we present a Task-aware MOT pipeline to unify all the task outputs (i.e., SOT, VOS, MOT, MOTS) as a unified set of instances with calibrated ID information, thereby alleviating the degradation of task-specific knowledge during multi-task training. With the unified model architecture and totally-shared parameters, SATA can solve 4 type of tracking and segmentation subtasks of 4 combinations of input modality. As shown in Fig. 7 (c), our proposed SATA significantly outperforms the simple combination of existing unified methods on all downstream tasks. We summarize that our work has the following contributions:

- To the best of knowledge, SATA is the first unified framework capable of performing both tracking and segmentation tasks in arbitrary modality input and multi-task joint prediction.
- We propose a Decoupled Mixture-of-Expert (DeMoE) mechanism and a Task-aware MOT pipeline to address the distribution gap in multi-modal data and the feature representation gap across tasks, enabling more effective cross-modality and cross-task knowledge sharing.
- We show superior results of our proposed method on 18 challenging benchmarks from 4 subtasks tasks, all using the same model architecture and parameter set.

Related works

Task- and modality-specific Methods. Over the years, a wide variety of task-specific models have been proposed to continuously improve the tracking and segmenting performance in specific scenarios, e.g., SOT, MOT, VOS and MOTS. SOT (Huang et al. 2019) and VOS (Ravi et al. 2024) specify tracked objects on the first frame of a video using

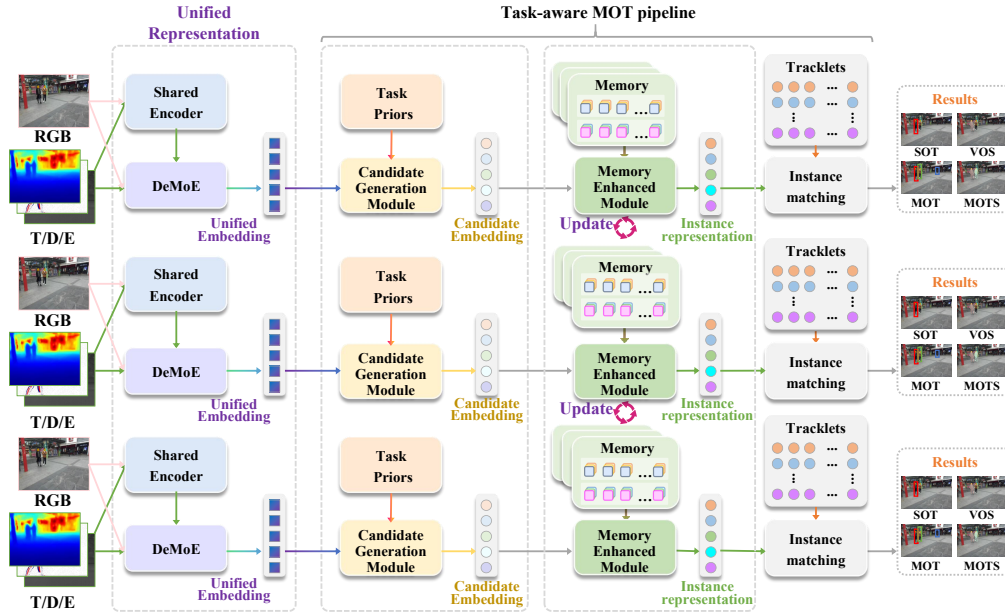


Figure 3: Overview architecture of our SATA, which consists of two core components: the Decoupled Mixture-of-Expert mechanism and the Task-aware MOT pipeline.

boxes or masks, then require algorithms to predict the trajectories of the tracked objects in boxes or masks, respectively. At present, Transformers based methods have mainstream SOT, which can be categorized into classification- and regression-based (Ye et al. 2022; Lin et al. 2024), corner prediction-based (Cui et al. 2022, 2023), and sequence-learning-based trackers (Chen et al. 2023). Meanwhile, memory-based VOS approaches have become the dominant method in VOS, which extend a single vision transformer to jointly process the current frame along with all previous frames and associated predictions (Ravi et al. 2024; Cheng et al. 2022). Different from SOT and VOS, MOT and MOTS aim to find and associate all instances (Yan et al. 2022). The mainstream methods follow the tracking-by-detection paradigm (Yan et al. 2022) and tracking-by-query (Chu et al. 2023) pipeline. With the development of sensor technology, tracking and segmentation tasks have evolved from using only a RGB camera to introduce various auxiliary modalities (Chen et al. 2025; Gao et al. 2022; Zhang et al. 2025b).

Unified Modality Methods. Despite the success of previous multi-modal methods tailored for modality-specific input, the inherent multi-parameter set paradigm limits the flexibility of these models in practical application scenarios, and generally requires the deployment of multiple models for different situations, greatly increasing the training and deployment costs. To echo this problem, some works (Zhu et al. 2023; Hong et al. 2024; Hou et al. 2024; Feng et al. 2025) adopt a architecture-shared design for RGB-X tracking, only activating the modality-specific parameters according to the input modality. Besides, some unified methods (Wu et al. 2024; Chen et al. 2025; Tan et al. 2025b,a) have been proposed to learn the common latent space of any modality input, aiming at achieving more complete unification. *However, the overlook of distribution gap and the lack*

of multi-task inference capabilities remains incomplete for a powerful unified tracking and segmentation model.

Unified Task Methods. Meanwhile researchers have undertaken prominent efforts to unify tracking and segmentation tasks within specific modality. Existing unified task methods can be categorized into two main branches: the prompt based methods and the detection based methods. The prompt based methods (Ma et al. 2022; Yan et al. 2022; Wang et al. 2024) apply a shared appearance model for unified representation learning, and employ the delicately designed target prior to solve multiple subtasks. Besides, the detection based methods (Li et al. 2024c; Yan et al. 2023; Wang et al. 2025) introduce an external detectors to predict objects in various frames independently, and then establish the associations for tracking and segmentation. *However, these methods pay less attention to the multi-task gap, result in the degradation of task-specific knowledge during multi-task training.*

Method

Overall Pipeline

The proposed framework, termed SATA, comprises two core components: the Decoupled Mixture-of-Expert (DeMoE) mechanism and the Task-aware MOT (TaMOT) pipeline, as shown in Fig. 8. The DeMoE, combined with a powerful Transformer-based encoder, first yields the unified representation. Subsequently, the designed Task-aware MOT pipeline first generates tracking candidates based on task-specific prior knowledge, then maintains tracking of all candidates, thereby unifying all tracking and segmentation subtasks under the MOT paradigm.

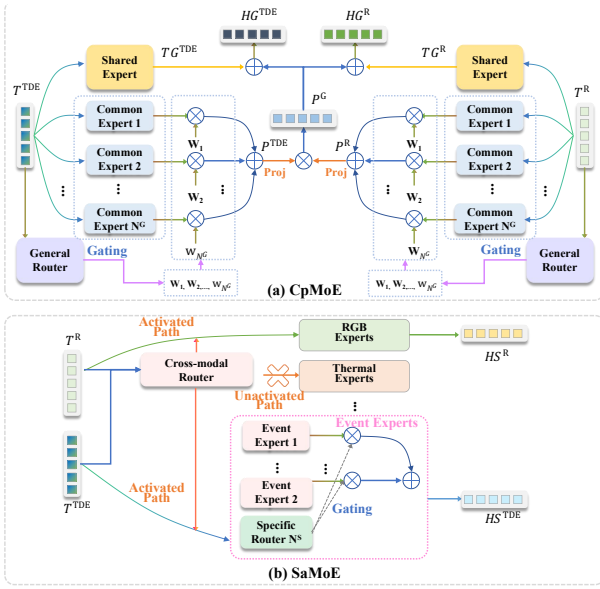


Figure 4: Overview architecture of our CpMoE and SaMoE. (a) CpMoE. (b) SaMoE.

Decoupled Mixture-of-Expert

Given the input video frames of the RGB (R) modality and their corresponding auxiliary modalities (thermal, depth, or event modalities, collectively referred to as TDE), a weight-shared Transformer-based encoder (Zhang et al. 2022b) is employed to generate RGB tokens and TDE tokens. To enable the unified model to handle diverse input modalities, the proposed DeMoE is used to replace the feed-forward network (FFN) in each Transformer encoder layer, thereby converting different modality combinations into a unified token embedding format. DeMoE consists of three primary components: a Common-prompt Mixture of Expert (CpMoE), a Specific-activated Mixture of Expert (SaMoE), and Decoupled Learning.

Common-prompt Mixture of Expert. At the l -th Transformer encoder layer, our CpMoE takes the RGB token T_l^R and TDE token T_l^{TDE} from the output of the Multi-head Attention (MSA) block as input. It comprises a general router network $\mathcal{R}^G$, a shared expert g^S , and N^G modality-common experts $\mathcal{G}^G = \{g_1^C, \dots, g_{N^G}^C\}$. Specifically, as shown in Fig. 9 (a), the shared expert g^S directly copies weights from the corresponding FFN layer of the encoder and remains frozen during training to retain pre-trained general knowledge, yielding TG_l^G and TG_l^{TDE} , i.e.,

$$TG_l^G = g^S(T_l^R), \quad TG_l^{TDE} = g^S(T_l^{TDE}). \quad (1)$$

Then, the modality-common experts g_n^C ($n = 1, \dots, N^G$) are employed to generate modality-common prompts, transforming the general knowledge to be more suitable for the current input modalities. Each modality-common expert g_n^C consists of two linear layers and a GELU activation layer. The generated RGB prompt P_l^G and TDE prompt P_l^{TDE} are weighted sums of outputs from the top-K activated experts,

i.e.,

$$P_l^R = \sum_{n=1}^{N^G} p_n^R g_n^C(T_l^R), \quad P_l^{TDE} = \sum_{n=1}^{N^G} p_n^{TDE} g_n^C(T_l^{TDE}), \quad (2)$$

where p_n^R and p_n^{TDE} denote the gating values derived from the function $\mathcal{R}^G$.

Subsequently, the modality-common prompt is obtained by performing element-wise multiplication on the RGB prompt and TDE prompt, which can be formulated as:

$$P_l^G = \text{proj}(P_l^R) \otimes \text{proj}(P_l^{TDE}), \quad (3)$$

where $\otimes$ denotes the element-wise multiplication operation, and $\text{proj}(\cdot)$ is the projection layer.

Finally, the modality-common prompt is employed to map TG_l^G and TG_l^{TDE} into a cohesive token representation through element-wise addition, i.e.,

$$HG_l^R = P_l^G \oplus TG_l^R, \quad HG_l^{TDE} = P_l^G \oplus TG_l^{TDE}, \quad (4)$$

where HG_l^R and HG_l^{TDE} refer to the prompted tokens for the RGB and TDE modalities, respectively, and $\oplus$ denotes the element-wise addition operation.

Specific-activated Mixture of Expert. SaMoE is designed to capture modality-specific clues within multi-modal data, comprising intra-modal router networks ($\mathcal{R}^R$ and $\mathcal{R}^{TDE}$), a cross-modal router network $\mathcal{R}^{CM}$, and several modality-specific experts ($\mathcal{G}^X = \{g_1^X, \dots, g_{N^S}^X\}$ ($X = R, TDE$), where N^S denotes the number of modality-specific experts).

As shown in Fig. 9 (b), the cross-modal router network $\mathcal{R}^{CM}$ activates the specific modality branch, thereby associating the current input modality with the corresponding modality experts. The modality-specific RGB tokens HS_l^R and TDE tokens HS_l^{TDE} are weighted sums of outputs from the top-K activated experts for each modality, i.e.,

$$HS_l^X = \sum_{n=1}^{N^S} s_n^X g_n^X(T_l^X), \quad X = R, TDE, \quad (5)$$

where s_n^X represents the gating values derived from the function $\mathcal{R}^X$.

Unified Representation. The RGB feature representation F_l^R and TDE feature representation F_l^{TDE} are obtained by aggregating these two types of embeddings from CpMoE and SaMoE with residual connections:

$$F_l^X = HG_l^X \oplus HS_l^X \oplus T_l^X, \quad X = R, TDE. \quad (6)$$

In the last layer (L -th layer) of the encoder, the final unified feature representation F^U is obtained as:

$$F^U = F_L^R \oplus F_L^{TDE}. \quad (7)$$

Decoupling Learning. To comprehensively model multi-modal representations, the learning processes of CpMoE and SaMoE adhere to two key principles: (a) promoting mutual complementarity among cross-modal experts; (b) avoiding information overlap between specific experts and common experts. Accordingly, we introduce two critical loss functions that act on the DeMoE.

Cross-modal complementary learning: To fully explore the complementary information across modalities, we randomly mask patches of one modality by assigning their values to a learnable token vector, generating masked features $\hat{H}G_l^G$ and $\hat{H}G_l^{TDE}$. The cross-modal complementary learning loss is then defined as:

$$\mathcal{L}_{CM} = \sum_{l=1}^L \text{MSE}(\hat{H}G_l^R, HG_l^R) + \sum_{l=1}^L \text{MSE}(\hat{H}G_l^{TDE}, HG_l^{TDE}), \quad (8)$$

where $\text{MSE}(\cdot)$ denotes the mean squared error.

Cross-expert orthogonal learning: Additionally, we introduce an orthogonal loss to encourage independence between common and specific expert representations. This cross-expert orthogonal loss is defined as:

$$\mathcal{L}_{CE} = \sum_{l=1}^L \text{OPL}(\sum_{j=1}^{N^S} \sum_{i=1}^{N^G} g_i^C(T_l^X), g_j^X(T_l^X)), X = R, TDE. \quad (9)$$

Here, $g_i^C(\cdot)$ and $g_j^X(\cdot)$ represents the outputs of experts from $\mathcal{G}^G$ and $\mathcal{G}^X$, respectively, $\text{OPL}(\cdot)$ denotes the orthogonal loss (Ranasinghe et al. 2021).

Task-aware MOT pipeline

To achieve the grand unification of tracking and segmentation while mitigating the representation gap across task domains, the proposed Task-aware MOT pipeline integrates all subtasks into the localization and association of multiple objects. Firstly, a Candidates Generation Module (CGM) is utilized to generate potential proposals based on task-specific prior information. Subsequently, the proposed Memory-enhanced Module (MEM) refines the historical features of each candidate, enabling efficient and effective modeling of temporal information. Finally, a simple bi-softmax nearest neighbor search is employed to achieve accurate matching between candidates and trajectories.

Candidates Generation Module. Instead of introducing additional detectors to detect all potential targets in each frame (Yan et al. 2023; Wang et al. 2025), we adopt a modified SAM2 (Ravi et al. 2024) as our foundation model to predict candidates based on the prior information of various tasks. Specifically, for SOT and VOS tasks, the initial target information (bounding box or mask) from the first frame can be used to generate the initial prompt token. In subsequent frames, masks predicted by the mask decoder with an affinity score exceeding a preset threshold are regarded as distractors; both the target and these distractors are defined as candidates. For MOT and MOTs tasks, we append a detection head to the foundation model to predict all potential objects in each frame. The box predictions are fed as multi-object prompts to the mask decoder, generating masks for all candidates. Once the positions of candidates in each frame are determined, we extract instance-level features by applying RoI Align. Specifically, given the unified embedding F_t^U of t -th frame and candidate coordinate B_t^m corresponding to the m -th tracklet, the corresponding candidate embedding can be acquired via $a_t^m = \text{RoIAlign}(F_t^U, B_t^m)$.

Memory-enhanced Module. The memory-enhanced module comprises two components: (a) generating fine-grained instance embeddings; (b) effectively modeling spatiotemporal relationship, as shown in Fig. 10.

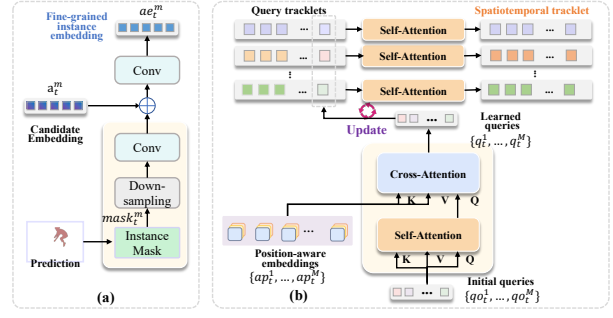


Figure 5: Illustration of the fine-grained instance embeddings and spatiotemporal relationship modeling. (a) Fine-grained instance embeddings. (b) Spatiotemporal relationship modeling.

Fine-grained instance embeddings: Candidate embeddings typically contain both foreground and background information at the edges, which leads to inaccurate instance matching when target objects cannot be clearly distinguished from the background. To echo this problem, given the candidate embedding a_t^m , candidate coordinate B_t^m , and their corresponding mask prediction $mask_t^m$, the adjusted mask $\hat{mask}_t^m$ is first generated by down-sampling the predicted mask $mask_t^m$. Then, the fine-grained instance embedding ae_t^m is generated by:

$$ae_t^m = \text{ConV}(\text{ConV}(\hat{mask}_t^m) \oplus a_t^m), \quad (10)$$

where $\text{ConV}(\cdot, \theta_n)$ denotes the convolutional layer.

Spatiotemporal relationship modeling: We first employ several MLP layers on the normalized coordinate $\hat{B}_t^m$ to generate the positional embedding p_t^m . The position-aware embedding can be represented as $ap_t^m = p_t^m \oplus a_t^m$. Then, we introduce the learnable queries q_t^m to capture video spatiotemporal information via the same architecture as the Querying Transformer (Q-Former) (Li et al. 2023). Specifically, the initial queries $\{qo_t^1, qo_t^2, \dots, qo_t^M\}$ are fed into the self-attention layer to model spatial interactions. After that, the interacted queries and the position-aware embeddings $\{ap_t^1, ap_t^2, \dots, ap_t^M\}$ are fed into the cross-attention layer, obtaining the t -th learned query $\{q_t^1, q_t^2, \dots, q_t^M\}$. Finally, we construct the temporal correlations on each query tracklet $\{q_{t-1}^m, q_{t-2}^m, \dots, q_{t-T}^m\} (m = 1, \dots, M)$ via a single self-attention layer, generating the spatiotemporal tracklet $\{qe_{t-1}^m, qe_{t-2}^m, \dots, qe_{t-T}^m\}$.

Finally, the comprehensive representation f_t^M of the M -th instances in t -th frame can be transformed by concatenating the fine-grained instance embedding ae_t^M and the learned query q_t^M , and the comprehensive tracklet can be represented by $\Gamma^n = \{f_{t-1}^n, f_{t-2}^n, \dots, f_{t-T}^n\} (n = 1, \dots, N)$.

Instance matching. Given the instance f_t^m and the tracklet Γ^n generated by the candidate generation module and the memory-enhanced module, the similarity computation can

RGB SOT Method	GOT10K			LaSOT			RGB-T SOT Method				RGB-D SOT Method			
	AO	SR _{0.5}	SR _{0.75}	AUC	P _{Norm}	P	LasHeR		RGBT234		DepthTrack		VOTRGBD	
Task-specific methods							PR	SR	PR	SR	PR	Re	EAO	Acc
GRM (Gao et al. 2023)	73.4	82.9	70.4	69.9	79.3	75.8								
LORAT-G (Lin et al. 2024)	78.9	87.8	80.7	76.2	85.3	83.5								
ODTrack-B (Zheng et al. 2024)	77.0	87.9	75.1	73.2	83.2	80.6								
ARTrackV2-L (Bai et al. 2024)	77.5	86.0	75.5	73.0	82.0	79.6								
LMTrack (Xu et al. 2025)	80.1	91.5	79.0	73.2	83.4	81.0								
TemTrack (Xie et al. 2025)	76.1	84.9	74.4	73.1	83.0	80.7								
MambaLCT (Li et al. 2025)	76.2	86.7	74.3	73.6	84.1	81.6								
SPMTrack-L (Cai et al. 2025)	80.0	89.4	79.9	76.8	85.9	84.0								
MCITTrack-B (Kang et al. 2025)	77.9	88.2	76.8	75.3	85.6	83.3								
DreamTrack-L (Guo et al. 2025)	79.9	88.5	80.3	76.6	85.6	83.1								
SAM2.1++ (Videnovic et al. 2025)	81.1	-	-	75.1	-	-								
Unified modality methods														
SUTrack-L (Chen et al. 2025)	81.5	89.5	83.3	75.2	84.9	83.2								
Unified task methods														
UTT (Ma et al. 2022)	67.2	76.3	60.5	64.6	-	67.2								
Unicorn (Yan et al. 2022)	-	-	-	68.5	76.6	74.1								
UNINEXT-H (Yan et al. 2023)	-	-	-	72.2	80.7	79.4								
OmniViD (Wang et al. 2024)	-	-	-	70.8	79.6	76.9								
SAM2.1 (Ravi et al. 2024)	80.7	-	-	70.5	-	-								
OmniTracker-L (Wang et al. 2025)	-	-	-	69.1	77.3	75.4								
SATA	81.3	91.4	83.7	77.3	85.7	84.6								

RGB-T SOT Method	LasHeR		RGBT234	
	PR	SR	PR	SR
Modality-specific methods				
TBSI+ (Li et al. 2024a)	75.5	59.6	91.0	67.0
TransMMSTC (Zhang et al. 2024c)	72.3	57.4	88.6	65.7
PURA (Shao et al. 2025)	72.0	57.2	89.4	66.4
AI-Net (Lu et al. 2025)	-	-	93.3	70.3
CAFormer (Xiao et al. 2025)	74.2	59.1	89.2	67.3
AEFormer (Zhu et al. 2025)	70.0	55.6	88.3	66.4
DMD (Hu et al. 2025b)	74.7	59.6	91.6	68.8
Architecture-shared methods				
ViPT (Zhu et al. 2023)	65.1	52.5	83.5	61.7
SDSTrack (Hou et al. 2024)	66.5	53.1	84.8	62.5
OneTrack (Hong et al. 2024)	67.2	53.8	85.7	64.2
GMMT (Tang et al. 2024)	70.7	56.6	87.9	64.7
CSTTrack (Feng et al. 2025)	75.6	60.8	94.0	70.9
STTrack (Hu et al. 2025a)	76.0	60.3	89.8	66.7
Unified modality methods				
UnTrack (Wu et al. 2024)	-	-	84.2	62.5
FlexTrack (Tan et al. 2025a)	77.3	-	92.7	69.9
XTrack (Tan et al. 2025b)	73.1	58.7	87.8	65.4
SUTrack-L (Chen et al. 2025)	76.9	-	93.7	70.3
SATA	77.8	61.7	94.3	71.5

RGB-E SOT Method	VisEvent		COESOT	
	PR	AUC	PR	SR
CSAM (Zhang et al. 2024a)	81.6	-	76.7	68.3
STTrack (Hu et al. 2025a)	78.6	61.9	-	-
CSTTrack (Feng et al. 2025)	82.4	65.2	77.4	83.3
SDSTrack (Hou et al. 2024)	76.7	59.7	-	-
OneTrack (Hong et al. 2024)	76.7	60.8	-	-
FlexTrack (Tan et al. 2025a)	81.4	64.1	-	-
XTrack-L (Tan et al. 2025b)	80.5	63.3	-	-
SUTrack-L (Chen et al. 2025)	80.5	63.8	-	-
UnTrack (Wu et al. 2024)	76.3	58.7	-	-
SATA	82.8	66.7	80.4	71.6

(a) SOTA comparisons on RGB SOT.

(b) SOTA comparisons on RGB-T SOT.

(c) SOTA comparisons on RGB-D SOT.

RGB-E SOT Method	VisEvent		COESOT	
	PR	AUC	PR	SR
CSAM (Zhang et al. 2024a)	81.6	-	76.7	68.3
STTrack (Hu et al. 2025a)	78.6	61.9	-	-
CSTTrack (Feng et al. 2025)	82.4	65.2	77.4	83.3
SDSTrack (Hou et al. 2024)	76.7	59.7	-	-
OneTrack (Hong et al. 2024)	76.7	60.8	-	-
FlexTrack (Tan et al. 2025a)	81.4	64.1	-	-
XTrack-L (Tan et al. 2025b)	80.5	63.3	-	-
SUTrack-L (Chen et al. 2025)	80.5	63.8	-	-
UnTrack (Wu et al. 2024)	76.3	58.7	-	-
SATA	82.8	66.7	80.4	71.6

(d) SOTA comparisons on RGB-E SOT.

Table 1: SOTA comparisons on RGB, RGB-Depth, RGB-Thermal, and RGB-Event SOT tasks.

RGB MOT Method	BDD			DanceTrack		
	IDF1	mMOTA	MOTA	IDF1	HOTA	MOTA
Task-specific methods						
DiffMOT (Lv et al. 2024)	-	-	-	63.0	62.3	92.8
Hybrid-SORT (Yang et al. 2024)	-	-	-	67.4	65.7	91.8
Unified task methods						
Unicorn (Yan et al. 2022)	71.3	41.2	66.6	-	-	-
UNINEXT-H (Yan et al. 2022)	69.9	44.2	67.1	-	-	-
MASA-SAM-H (Li et al. 2024c)	71.7	44.5	-	-	-	-
SAM2MOT-Co (Jiang et al. 2025)	70.8	-	57.5	83.4	75.5	89.2
SATA	73.2	46.3	67.8	83.7	76.1	90.5

RGB-T MOT Method	HOTA	DetA	AssA	MOTA	IDF1
Task-specific methods					
Bytetrack (Zhang et al. 2022c)	53.4	58.2	49.6	72.4	64.9
MOTRv2 (Zhang et al. 2023)	53.1	51.2	55.8	63.3	64.4
HGT-Track (Xu et al. 2024)	54.0	61.3	48.1	71.1	60.9
Unified task methods					
Unitrack (Wang et al. 2021)	46.9	46.1	48.4	57.1	57.4
Unicorn (Yan et al. 2022)	49.4	48.3	51.2	59.0	60.6
UnisMOT (Zhang et al. 2025a)	54.2	59.5	49.7	65.7	60.3
SATA	59.7	63.6	53.7	75.1	65.4

(a) SOTA comparisons on RGB MOT

(b) SOTA comparisons on RGB-T MOT

Table 2: SOTA comparisons on RGB, and RGB-Thermal MOT tasks.

be represented by:

$$\begin{aligned}
s(f_t^m, \Gamma^n) &= \frac{1}{2}(s_1(f_t^m, \Gamma^n) + s_2(f_t^m, \Gamma^n)), \\
s_1(f_t^m, \Gamma^n) &= \frac{1}{2} \left[\frac{\exp(f_t^m \cdot \Gamma^n)}{\sum_{m=1}^M \exp(f_t^m \cdot \Gamma^n)} + \frac{\exp(f_t^m \cdot \Gamma^n)}{\sum_{n=1}^N \exp(f_t^m \cdot \Gamma^n)} \right], \\
s_2(f_t^m, \Gamma^n) &= \frac{f_t^m \cdot \Gamma^n}{\|f_t^m\| \|\Gamma^n\|}.
\end{aligned} \tag{11}$$

With the above similarity function, we can obtain the final assignment matrix $A \in \mathbb{R}^{N \times M}$. Finally, we match the ϵ -th tracklet with m -th instances via $\text{argmax}(s(f_t^m, \Gamma^1), \dots, s(f_t^m, \Gamma^N)) = s(f_t^m, \Gamma^\epsilon)$.

Experiments

Implementation Details

The SATA model is developed using Python 3.8 and PyTorch 1.11. The training process leverages 8 NVIDIA A100 GPUs, while the inference speed is evaluated on a single NVIDIA 3090TI GPU.

Architecture. Our model uses HiViT-L (Zhang et al. 2022b) as the transformer encoder, and we select SAM2 (Ravi et al. 2024) as our foundation model in the Task-aware MOT pipeline. The transformer encoder and the foundation model are initialized with the pre-trained parameters of SAM2 (Ravi et al. 2024).

Training. During the stage of candidate generation, the affinity mask scores, object predictions, and IoU predictions of the mask decoder are optimized by MAE loss, cross-entropy loss, and L_1 loss, respectively (Ravi et al. 2024). In addition, the decoupled MoE loss $L_{\text{MoE}} = \mu L_{\text{CM}} + \lambda L_{\text{CE}}$ in the DeMoE is employed to promote comprehensive learning

of the unified embedding. In the stage of instance matching, the partial supervision loss and self-supervised loss in KeepTrack (Mayer et al. 2021) are employed to supervise the assignment matrix A for SOT and VOS datasets, and the cross-entropy loss is applied to optimize SATA for MOT and MOTs datasets.

State-of-the-Art Comparisons

We compare SATA with state-of-the-art methods on 18 large-scale benchmarks with 4 types of input (i.e., RGB, RGB-T, RGB-D, RGB-E) and 4 subtasks (i.e., SOT, VOS, MOT and MOTs).

SOT. The results of RGB SOT are presented in Tab. 1a. Our model achieves 81.3% AO and 77.3% AUC on GOT10K (Huang et al. 2019) and LaSOT (Fan et al. 2019), respectively, surpassing the recent RGB tracker SAM2.1++ (Videnovic et al. 2025), SUTrack (Chen et al. 2025), and LMTrack (Xu et al. 2025) by 0.2%/2.2%, 0.3%/2.1%, and 1.2%/4.1%, respectively. In addition, compared with existing unified task models, e.g., UTT (Ma et al. 2022), Unicorn (Yan et al. 2022), and OmniTracker (Wang et al. 2025), SATA achieves performance gains of 12.7%, 8.8%, and 7.9% in AUC on LaSOT, respectively. Besides, SATA sets a new state-of-the-art on 6 multi-modal SOT benchmarks, as illustrated in Tab. 1b, Tab. 1c, and Tab. 1d. On LasHeR (Li et al. 2022), DepthTrack (Yan et al. 2021b), and VisEvent (Wang et al. 2023), SATA surpassing the recent best unified modality trackers, XTrack (Tan et al. 2025b), SUTrack (Chen et al. 2025) and FlexTrack (Tan et al. 2025a), by 4.7%/2.5%/2.3%, 0.9%/1.4%/2.3%, and

RGB VOS Method	DAVIS 2016 val			DAVIS 2017 val		
	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{F}$
Task-specific methods						
XMem (Cheng et al. 2022)	92.0	90.7	93.2	87.7	84.0	91.4
DEVOS (Fedyunyak et al. 2024)	92.8	91.8	93.8	88.2	84.5	91.9
OneVOS (Li et al. 2024d)	92.7	91.0	94.3	88.5	84.6	92.4
M3-VOS (Chen et al. 2024)	-	-	-	-	86.0	-
SAM2 (Ravi et al. 2024)	-	-	-	88.9	-	-
TAM-S (Ravi et al. 2024)	-	-	-	89.2	-	-
Unified task methods						
UniTrack (Yan et al. 2022)	-	-	-	-	58.4	-
Unicorn-T (Yan et al. 2022)	83.2	83.0	83.4	64.5	62.7	66.3
Unicorn-ConvL (Yan et al. 2022)	87.4	86.5	88.2	69.2	65.2	73.2
UNINEXT-R50 (Yan et al. 2023)	-	-	-	74.5	71.3	77.6
UNINEXT-L (Yan et al. 2023)	-	-	-	77.2	73.2	81.2
UNINEXT-H (Yan et al. 2023)	-	-	-	81.8	77.7	85.8
OmniTracker-T (Wang et al. 2025)	84.7	84.1	85.3	66.2	64.9	67.5
OmniTracker-L (Wang et al. 2025)	88.5	87.3	89.7	71.0	66.8	75.2
SATA	93.4	91.6	95.2	89.7	86.1	93.0

(a) Comparisons on RGB VOS.

RGB-T VOS Method	VisT300			VTUAV		
	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{F}$
Modality-specific methods						
STM (Oh et al. 2019)	60.4	57.9	62.8	-	-	-
STCN (Cheng et al. 2021)	71.4	74.4	73.8	65.5	61.0	69.9
STCN-T (Cheng et al. 2021)	72.3	-	-	-	-	-
TBD (Cho et al. 2022)	70.5	68.3	72.6	-	-	-
CFBI+ (Yang et al. 2021a)	74.1	71.8	76.4	-	-	-
D3S (Lukezic et al. 2020)	-	-	-	57.0	53.4	60.7
AlpharRefine (Yan et al. 2021a)	-	-	-	65.9	59.9	71.9
AOT (Yang et al. 2021b)	76.8	74.0	79.6	-	-	-
AOT-B (Yang et al. 2021b)	-	-	-	81.8	77.7	85.8
AOT-L (Yang et al. 2021b)	-	-	-	82.0	77.5	86.5
XMem (Cheng et al. 2022)	75.7	73.3	78.0	69.1	65.1	73.1
XMem-T (Cheng et al. 2022)	77.9	-	-	-	-	-
ViTNet (Yang et al. 2023)	81.9	79.2	84.5	76.7	72.9	80.8
Architecture-shared methods						
X-Prompt (Guo et al. 2024)	84.2	81.7	86.7	87.3	82.8	91.8
SATA	87.4	84.5	90.3	88.5	84.4	92.6

(b) Comparisons on RGB-T VOS.

RGB-D VOS Method	ARKitTrack		
	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{F}$
STCN-D (Cheng et al. 2021)	53.7	49.8	57.5
ARKitVOS (Zhao et al. 2023)	66.2	62.5	69.8
XMem (Cheng et al. 2022)	71.6	68.5	74.6
DeAOT (Yang et al. 2022)	72.6	70.0	75.3
AOT-L-Swin (Yang et al. 2021b)	77.8	75.0	80.7
X-Prompt (Guo et al. 2024)	82.1	79.4	84.9
SATA	85.1	82.8	87.4

(c) Comparisons on RGB-D VOS.

RGB-E VOS Method	LLE-VOS		
	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{F}$
STCN (Cheng et al. 2021)	47.4	45.0	49.8
XMem (Cheng et al. 2022)	54.2	59.0	49.4
AOT-B (Yang et al. 2021b)	62.3	64.9	59.6
DeAOT (Yang et al. 2022)	63.3	65.3	61.4
LLE-VOS (Li et al. 2024b)	67.8	70.2	65.4
SATA	71.4	73.6	69.1

(d) Comparisons on RGB-E VOS.

Table 3: SOTA comparisons on RGB, RGB-Depth, RGB-Thermal, and RGB-Event VOS tasks.

RGB MOTS Method	mMOTSA	mMOTSP	mIDF1	ID Sw
Task-specific methods				
MaskTrackRCNN (Fu et al. 2021)	12.3	59.9	26.2	9116
STEm-Seg (Athar et al. 2020)	12.2	58.2	25.4	8732
QDTrack-mots (Huang et al. 2023)	22.5	59.6	40.8	1340
PCAN (Ke et al. 2021)	27.4	66.7	45.1	876
VMOT (Ke et al. 2022)	28.7	67.3	45.7	825
MASA-B (Li et al. 2024c)	35.2	-	49.2	-
MASA-H (Li et al. 2024c)	35.8	-	49.7	-
Unified task methods				
Unicorn (Yan et al. 2022)	29.6	67.7	44.2	1731
UNINEXT-L (Yan et al. 2023)	32.0	60.2	45.4	1634
UNINEXT-H (Yan et al. 2023)	35.7	68.1	48.5	1776
SATA	38.1	72.3	52.4	721

Table 4: SOTA comparisons on RGB MOTS tasks.

0.5%/0.8%/1.4% in PR, respectively.

MOT. We report the results of RGB MOT in Tab. 2a. Our SATA achieves the best MOTA, HOTA, and IDF1 scores on DanceTrack (Sun et al. 2022) and BDD (Yu et al. 2020). Specifically, it outperforms all previous unified task methods, achieving an mMOTA score of 67.8% on BDD, surpassing Unicorn (Yan et al. 2022), UNINEXT-H (Yan et al. 2023) and SAM2MOT-Co (Jiang et al. 2025) by 1.2%, 0.7%, and 10.3%, respectively. In addition, as shown in Tab. 2b, SATA achieves state-of-the-art performance on UniRTL (Zhang et al. 2025a) with RGB-T input, outperforming the previous best method UnisMOT (Zhang et al. 2025a) by 5.3% in HOTA.

VOS. We present a comparison with recent advanced RGB VOS methods in Table 3a, reporting accuracy using standard protocols. SATA shows significant improvement over the best existing methods, achieving the highest scores of 93.4% and 89.7% $\mathcal{J}$ & $\mathcal{F}$ on these DAVIS datasets (Pont-Tuset et al. 2017). Besides, as presented in Table 3b, Table 3c, and Table 3d, SATA secures the top positions on the RGB-T (Yang et al. 2023; Zhang et al. 2022a), RGB-D (Zhao et al. 2023), and RGB-E VOS (Li et al. 2024b) benchmarks.

MOTS. We evaluate SATA’s MOTS capability on BDD MOTS (Yu et al. 2020) As shown in Table 4, our approach outperforms existing advanced unified models, i.e., Unicorn (Yan et al. 2022) and UNINEXT-H (Yan et al. 2023) by noteworthy margins of 8.5% and 2.4% in mMOTSA, respectively.

Type	Method	SOT GOT10K (AO)	SOT LasHeR (PR)	VOS LLE-VOS ($\mathcal{J}$ & $\mathcal{F}$)	MOT UniRTL HOTA	MOTS BDD MOTS MOTSA
SATA		81.3	77.8	71.4	59.7	38.1
DeMoE	W/o CpMoE	80.8	75.8	70.7	56.2	37.9
	W/o SaMoE	81.3	75.3	69.3	55.3	37.7
	W/o CpMoE & SaMoE	80.8	74.7	67.9	56.1	37.7
	W/o L_{MoE}	80.7	75.2	70.2	58.4	36.9
TaMOT	W/o CGM	78.5	74.3	68.7	-	-
	W/o fine-grained memory	80.7	75.3	69.2	56.7	34.1
	W/o spatiotemporal memory	79.7	75.8	67.4	57.5	30.7
	W/o MEM	79.5	74.3	67.0	54.2	29.1

Table 5: Ablation studies on DeMoE and TaMOT.

Ablation Studies

In this section, we conduct component-wise analysis for a better understanding of our method. The methods are evaluated on 5 benchmarks (GOT10K (Huang et al. 2019), LasHeR (Li et al. 2022), LLE-VOS (Li et al. 2024b), UniRTL (Zhang et al. 2025a), and BDD MOTS (Yu et al. 2020)) from 4 subtasks (SOT, VOS, MOT, MOTS).

Decoupled Mixture-of-Expert. To investigate the impact of our proposed DeMoE, several versions of our proposed method are provided, including ①: Removing the CpMoE sub-module in DeMoE. ②: Removing the SaMoE sub-module in DeMoE. ③: Removing the CpMoE and SaMoE sub-modules in DeMoE. ④: Removing the MoE loss L_{MoE} in DeMoE. As can be seen in Table 9, the performance degrades significantly after removing CpMoE or SaMoE sub-modules in multi-modal subtasks, which demonstrates the effectiveness of the proposed DeMoE in handling any modality input.

Task-aware MOT pipeline. To further verify the effectiveness of the proposed TaMOT, several variants are designed, including ①: Removing the CGM. ②: Removing the fine-grained memory. ③: Removing the spatiotemporal memory. ④: Removing the MEM. As shown in Table 9, comparative results indicate that keeping track of all potential objects can further improve the robustness of SOT and VOS tasks. Besides, the tracking performance experiences a significant decline upon the ablation of fine-grained memory or spatiotemporal memory, which confirms the necessity of the comprehensive trajectory information incorporated in TaMOT for accurate instance matching.

Conclusion

In this paper, we propose a universal tracking and segmentation framework, referred to as SATA, which is capable of processing inputs from any modality and predicting results for a wide range of tracking and segmentation subtasks with a fully shared network architecture, model weights, and inference pipeline. Extensive experiments on 18 challenging benchmarks demonstrate that SATA achieves superior performance across these tasks. We hope that SATA can lay a solid foundation for future research on AGI.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China under Grant No.62441235, and part by the Beijing Natural Science Foundation under Grant No.L257005.

References

- Athar, A.; Mahadevan, S.; Osep, A.; Leal-Taixé, L.; and Leibe, B. 2020. Stem-seg: Spatio-temporal embeddings for instance segmentation in videos. In *ECCV*, 158–177. Springer.
- Bai, Y.; Zhao, Z.; Gong, Y.; et al. 2024. Artrackv2: Prompting autoregressive tracker where to look and how to describe. In *CVPR*, 19048–19057.
- Cai, W.; Liu, Q.; and Wang, Y. 2025. SPMTrack: Spatio-Temporal Parameter-Efficient Fine-Tuning with Mixture of Experts for Scalable Visual Tracking. In *CVPR*, 16871–16881.
- Chen, X.; Kang, B.; et al. 2025. SUTrack: Towards simple and unified single object tracking. In *AAAI*, volume 39, 2239–2247.
- Chen, X.; Peng, H.; et al. 2023. Seqtrack: Sequence to sequence learning for visual object tracking. In *CVPR*, 14572–14581.
- Chen, Z.; Li, J.; et al. 2024. M3-VOS: Multi-Phase, Multi-Transition, and Multi-Scenery Video Object Segmentation. *arXiv:2412.13803*.
- Cheng, H. K.; and Schwing, A. G. 2022. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *ECCV*, 640–658. Springer.
- Cheng, H. K.; Tai, Y.-W.; and Tang, C.-K. 2021. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. *NeurIPS*, 34: 11781–11794.
- Cho, S.; Lee, H.; et al. 2022. Tackling background distraction in video object segmentation. In *ECCV*, 446–462. Springer.
- Chu, P.; Wang, J.; et al. 2023. Transmot: Spatial-temporal graph transformer for multiple object tracking. In *WACV*, 4870–4880.
- Cui, Y.; Jiang, C.; et al. 2022. Mixformer: End-to-end tracking with iterative mixed attention. In *CVPR*, 13608–13618.
- Cui, Y.; Song, T.; Wu, G.; et al. 2023. Mixformerv2: Efficient fully transformer tracking. *NeurIPS*, 36: 58736–58751.
- Ding, H.; Liu, C.; He, S.; Jiang, X.; Torr, P. H.; and Bai, S. 2023. MOSE: A new dataset for video object segmentation in complex scenes. In *ICCV*, 20224–20234.
- Fan, H.; Lin, L.; et al. 2019. Lasot: A high-quality benchmark for large-scale single object tracking. In *CVPR*, 5374–5383.
- Fedynyak, V.; Romanus, Y.; et al. 2024. Devos: flow-guided deformable transformer for video object segmentation. In *WACV*, 240–249.
- Feng, X.; Zhang, D.; et al. 2025. CSTrack: Enhancing RGB-X Tracking via Compact Spatiotemporal Features. In *ICML*.
- Fu, Y.; Liu, S.; et al. 2021. Learning to track instances without video annotations. In *CVPR*, 8680–8689.
- Gao, S.; Yang, J.; Li, Z.; Zheng, F.; Leonardis, A.; and Song, J. 2022. Learning dual-fused modality-aware representations for rgbd tracking. In *ECCV*, 478–494.
- Gao, S.; Zhou, C.; and Zhang, J. 2023. Generalized relation modeling for transformer tracking. In *CVPR*, 18686–18695.
- Guo, M.; Tan, W.; Ran, W.; Jing, L.; and Zhang, Z. 2025. DreamTrack: Dreaming the Future for Multimodal Visual Object Tracking. In *CVPR*, 7201–7210.
- Guo, P.; Li, W.; et al. 2024. X-prompt: Multi-modal visual prompt for video object segmentation. In *ACM MM*, 5151–5160.
- Hong, L.; Yan, S.; et al. 2024. OneTracker: Unifying Visual Object Tracking with Foundation Models and Efficient Tuning. In *CVPR*.
- Hou, X.; Xing, J.; et al. 2024. SDSTrack: Self-Distillation Symmetric Adapter Learning for Multi-Modal Visual Object Tracking. In *CVPR*.
- Hu, X.; Tai, Y.; Zhao, X.; Zhao, C.; Zhang, Z.; Li, J.; Zhong, B.; and Yang, J. 2025a. Exploiting multimodal spatial-temporal patterns for video object tracking. In *AAAI*, volume 39, 3581–3589.
- Hu, Y.; Shao, Z.; Fan, B.; and Liu, H. 2025b. Dual-level Modality De-biasing for RGB-T Tracking. *IEEE TIP*.
- Huang, K.; Lertniphonphan, K.; Chen, F.; Li, J.; and Wang, Z. 2023. Multi-object tracking by self-supervised learning appearance model. In *CVPR*, 3163–3169.
- Huang, L.; Zhao, X.; and Huang, K. 2019. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE TPAMI*, 43(5): 1562–1577.
- Jiang, J.; Wang, Z.; Zhao, M.; Li, Y.; and Jiang, D. 2025. SAM2MOT: A Novel Paradigm of Multi-Object Tracking by Segmentation. *arXiv:2504.04519*.
- Kang, B.; Chen, X.; et al. 2025. Exploring enhanced contextual information for video-level object tracking. In *AAAI*, volume 39, 4194–4202.
- Ke, L.; Ding, H.; Danelljan, M.; Tai, Y.-W.; Tang, C.-K.; and Yu, F. 2022. Video mask transfiner for high-quality video instance segmentation. In *ECCV*, 731–747. Springer.
- Ke, L.; Li, X.; Danelljan, M.; Tai, Y.-W.; Tang, C.-K.; and Yu, F. 2021. Prototypical cross-attention networks for multiple object tracking and segmentation. *NeurIPS*, 34: 1192–1203.
- Li, B.; Peng, F.; Hui, T.; Wei, X.; Wei, X.; Zhang, L.; Shi, H.; and Liu, S. 2024a. RGB-T tracking with template-bridged search interaction and target-preserved template updating. *IEEE TPAMI*.
- Li, C.; Xue, W.; Jia, Y.; Qu, Z.; Luo, B.; Tang, J.; and Sun, D. 2022. LasHeR: A Large-Scale High-Diversity Benchmark for RGBT Tracking. *IEEE TIP*, 31: 392–404.
- Li, H.; Wang, J.; Yuan, J.; Li, Y.; Weng, W.; Peng, Y.; Zhang, Y.; Xiong, Z.; and Sun, X. 2024b. Event-assisted low-light video object segmentation. In *CVPR*, 3250–3259.
- Li, J.; Li, D.; Savarese, S.; and Hoi, S. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 19730–19742. PMLR.
- Li, S.; Ke, L.; et al. 2024c. Matching anything by segmenting anything. In *CVPR*, 18963–18973.
- Li, W.; Guo, P.; et al. 2024d. Onevos: unifying video object segmentation with all-in-one transformer framework. In *ECCV*, 20–40.
- Li, X.; Zhong, B.; Liang, Q.; Li, G.; Mo, Z.; and Song, S. 2025. MambaLCT: Boosting Tracking via Long-term Context State Space Model. In *AAAI*, volume 39, 4986–4994.

- Lin, L.; Fan, H.; et al. 2024. Tracking meets lora: Faster training, larger model, stronger performance. In *ECCV*, 300–318.
- Lin, T. Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *Proceedings of the ECCV*, 740–755.
- Lu, A.; Wang, W.; Li, C.; Tang, J.; and Luo, B. 2025. Rgbt tracking via all-layer multimodal interactions with progressive fusion mamba. In *AAAI*, volume 39, 5793–5801.
- Lukezic, A.; Matas, J.; and Kristan, M. 2020. D3s-a discriminative single shot segmentation tracker. In *CVPR*, 7133–7142.
- Lv, W.; Huang, Y.; Zhang, N.; Lin, R.-S.; Han, M.; and Zeng, D. 2024. Diffmot: A real-time diffusion-based multiple object tracker with non-linear prediction. In *CVPR*, 19321–19330.
- Ma, F.; Shou, M. Z.; et al. 2022. Unified transformer tracker for object tracking. In *CVPR*, 8781–8790.
- Mayer, C.; Danelljan, M.; Paudel, D. P.; and Gool, L. V. 2021. Learning Target Candidate Association to Keep Track of What Not to Track. In *ICCV*, 13424–13434.
- Muller, M.; Bibi, A.; Giancola, S.; Alsubaihi, S.; and Ghanem, B. 2018. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *Proceedings of the ECCV*, 300–317.
- Oh, S. W.; Lee, J.-Y.; et al. 2019. Video object segmentation using space-time memory networks. In *ICCV*, 9226–9235.
- Pont-Tuset, J.; Perazzi, F.; Caelles, S.; Arbeláez, P.; Sorkine-Hornung, A.; and Van Gool, L. 2017. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*.
- Ranasinghe, K.; Naseer, M.; Hayat, M.; Khan, S.; and Khan, F. S. 2021. Orthogonal projection loss. In *ICCV*, 12333–12343.
- Ravi, N.; Gabeur, V.; et al. 2024. Sam 2: Segment anything in images and videos. *arXiv:2408.00714*.
- Ryali, C.; Hu, Y.-T.; Bolya, D.; Wei, C.; Fan, H.; Huang, P.-Y.; Aggarwal, V.; Chowdhury, A.; Poursaeed, O.; Hoffman, J.; et al. 2023. Hiera: A hierarchical vision transformer without the bells-and-whistles. In *International conference on machine learning*, 29441–29454. PMLR.
- Shao, Z.; Hu, Y.; Fan, B.; and Liu, H. 2025. PURA: Parameter Update-Recovery Test-Time Adaption for RGB-T Tracking. In *CVPR*, 22089–22098.
- Sun, P.; Cao, J.; et al. 2022. Dancetrack: Multi-object tracking in uniform appearance and diverse motion. In *CVPR*, 20993–21002.
- Tan, Y.; Shao, J.; et al. 2025a. What You Have is What You Track: Adaptive and Robust Multimodal Tracking. In *ICCV*, 3455–3465.
- Tan, Y.; Wu, Z.; et al. 2025b. XTrack: Multimodal Training Boosts RGB-X Video Object Trackers. *arXiv:2405.17773*.
- Tang, Z.; Xu, T.; Wu, X.; Zhu, X.-F.; and Kittler, J. 2024. Generative-based fusion mechanism for multi-modal tracking. In *AAAI*, volume 38, 5189–5197.
- Videnovic, J.; Lukezic, A.; and Kristan, M. 2025. A distractor-aware memory for visual object tracking with sam2. In *CVPR*, 24255–24264.
- Wang, J.; Chen, D.; Luo, C.; He, B.; Yuan, L.; Wu, Z.; and Jiang, Y.-G. 2024. Omnivid: A generative framework for universal video understanding. In *CVPR*, 18209–18220.
- Wang, J.; Wu, Z.; Chen, D.; Luo, C.; Dai, X.; Yuan, L.; and Jiang, Y.-G. 2025. OmniTracker: Unifying Visual Object Tracking by Tracking-with-Detection. *IEEE TPAMI*.
- Wang, X.; Li, J.; et al. 2023. Visevent: Reliable object tracking via collaboration of frame and event flows. *IEEE TCYB*, 54(3): 1997–2010.
- Wang, Z.; Zhao, H.; Li, Y.-L.; Wang, S.; Torr, P.; and Bertinetto, L. 2021. Do different tracking tasks require different appearance models? *NeurIPS*, 34: 726–738.
- Wu, Z.; Zheng, J.; et al. 2024. Single-Model and Any-Modality for Video Object Tracking. In *CVPR*.
- Xiao, Y.; Zhao, J.; Lu, A.; Li, C.; Yin, B.; Lin, Y.; and Liu, C. 2025. Cross-modulated Attention Transformer for RGBT Tracking. In *AAAI*, volume 39, 8682–8690.
- Xie, J.; Zhong, B.; Liang, Q.; Li, N.; Mo, Z.; and Song, S. 2025. Robust tracking via mamba-based context-aware token learning. In *AAAI*, volume 39, 8727–8735.
- Xu, C.; Zhong, B.; et al. 2025. Less is more: Token context-aware learning for object tracking. In *AAAI*, volume 39, 8824–8832.
- Xu, N.; Yang, L.; Fan, Y.; Yang, J.; Yue, D.; Liang, Y.; Price, B.; Cohen, S.; and Huang, T. 2018. Youtube-vos: Sequence-to-sequence video object segmentation. In *Proceedings of the ECCV (ECCV)*, 585–601.
- Xu, Q.; Wang, L.; et al. 2024. Heterogeneous graph transformer for multiple tiny object tracking in RGB-T videos. *IEEE TMM*.
- Yan, B.; Jiang, Y.; et al. 2022. Towards grand unification of object tracking. In *ECCV*, 733–751.
- Yan, B.; Jiang, Y.; et al. 2023. Universal instance perception as object discovery and retrieval. In *CVPR*, 15325–15336.
- Yan, B.; Zhang, X.; Wang, D.; Lu, H.; and Yang, X. 2021a. Alpha-refine: Boosting tracking performance by precise bounding box estimation. In *CVPR*, 5289–5298.
- Yan, S.; Yang, J.; Käpylä, J.; Zheng, F.; Leonardis, A.; and Kämäräinen, J.-K. 2021b. Depthtrack: Unveiling the power of RGBD tracking. In *ICCV*, 10725–10733.
- Yang, J.; Gao, M.; Cong, R.; Wang, C.; Zheng, F.; and Leonardis, A. 2023. Unveiling the Power of Visible-Thermal Video Object Segmentation. *IEEE TCSVT*, 34(7): 5376–5388.
- Yang, M.; Han, G.; Yan, B.; Zhang, W.; Qi, J.; Lu, H.; and Wang, D. 2024. Hybrid-sort: Weak cues matter for online multi-object tracking. In *AAAI*, volume 38, 6504–6512.
- Yang, Z.; Wei, Y.; and Yang, Y. 2021a. Collaborative video object segmentation by multi-scale foreground-background integration. *IEEE TPAMI*, 44(9): 4701–4712.
- Yang, Z.; Wei, Y.; et al. 2021b. Associating objects with transformers for video object segmentation. *NeurIPS*, 34: 2491–2502.
- Yang, Z.; and Yang, Y. 2022. Decoupling features in hierarchical propagation for video object segmentation. *NeurIPS*, 35: 36324–36336.
- Ye, B.; Chang, H.; Ma, B.; Shan, S.; and Chen, X. 2022. Joint feature learning and relation modeling for tracking: A one-stream framework. In *ECCV*, 341–357.
- Yu, F.; Chen, H.; Wang, X.; Xian, W.; Chen, Y.; Liu, F.; Madhavan, V.; and Darrell, T. 2020. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*, 2636–2645.
- Zhang, L.; Wang, L.; Wu, Y.; Chen, M.; Zheng, D.; Cao, L.; Zeng, B.; and Cai, Y. 2025a. UniRTL: A universal RGBT and low-light benchmark for object tracking. *Pattern Recognition*, 158: 110984.
- Zhang, P.; Zhao, J.; Wang, D.; Lu, H.; and Ruan, X. 2022a. Visible-thermal UAV tracking: A large-scale benchmark and new baseline. In *CVPR*, 8886–8895.
- Zhang, T.; Debattista, K.; Zhang, Q.; Han, J.; et al. 2024a. Revisiting motion information for RGB-Event tracking with MOT philosophy. *NeurIPS*, 37: 89346–89372.

Zhang, T.; He, X.; Luo, Y.; Zhang, Q.; and Han, J. 2024b. Exploring target-related information with reliable global pixel relationships for robust RGB-T tracking. *Pattern Recognition*, 110707.

Zhang, T.; Jiao, Q.; et al. 2024c. Exploring Multi-modal Spatial-Temporal Contexts for High-performance RGB-T Tracking. *IEEE TIP*, 1–1.

Zhang, T.; Zhang, Q.; Debattista, K.; and Han, J. 2025b. Cross-Modality Distillation for Multi-modal Tracking. *IEEE TPAMI*.

Zhang, X.; Tian, Y.; et al. 2022b. Hivit: Hierarchical vision transformer meets masked image modeling. *arXiv:2205.14949*.

Zhang, Y.; Sun, P.; et al. 2022c. Bytetrack: Multi-object tracking by associating every detection box. In *ECCV*, 1–21. Springer.

Zhang, Y.; Wang, T.; and Zhang, X. 2023. Motrv2: Bootstrapping end-to-end multi-object tracking by pretrained object detectors. In *CVPR*, 22056–22065.

Zhao, H.; Chen, J.; Wang, L.; and Lu, H. 2023. ARKitTrack: A New Diverse Dataset for Tracking Using Mobile RGB-D Data. In *CVPR*, 5126–5135.

Zheng, Y.; Zhong, B.; et al. 2024. Odtrack: Online dense temporal token learning for visual tracking. In *AAAI*, volume 38, 7588–7596.

Zhu, J.; Lai, S.; Chen, X.; Wang, D.; and Lu, H. 2023. Visual prompt multi-modal tracking. In *CVPR*, 9516–9526.

Zhu, Z.; Zhong, B.; et al. 2025. Adaptive Expert Decision for RGB-T Tracking. *IEEE TCSVT*.

Appendix

In this supplementary material, we first elaborate on the detailed specifications of the model. Subsequently, we outline the training procedure of the proposed SATA framework. Following this, we present more details about the loss functions. Finally, we present more experimental results along with further in-depth analyses of our SATA approach.

Model details

Here, we further elaborate on the architectural details of the proposed SATA, encompassing the Backbone Network, Decoupled Mixture-of-Experts, Candidates Generation Module, Memory-enhanced Module, and Instance Matching.

Backbone Network

We selected Hiera-L (Ryali et al. 2023) as our backbone network, which comprises four stages for both low-level and high-level visual modeling, as presented in Table 6. Each stage consists of multiple Hiera blocks, which are analogous to the typical Transformer block. Each hiera block incorporates normalization layers, a multi-head attention block, a feed-forward network, and several residual connections. To fully leverage the open-world general knowledge acquired from large-scale training data, we employ the pretrained weights from SAM2 (Ravi et al. 2024) for parameter initialization.

Decoupled Mixture-of-Expert

DeMoE consists of two primary components: a Common-prompt Mixture of Expert (CpMoE) and Specific-activated Mixture of Expert (SaMoE).

CpMoE. Each modality-common expert g_n^C consists of two linear layers and a GELU activation layer: the first linear layer projects the input to a lower dimension, and the second layer projects it back to the original dimension. In the experiments, we project each input token with channel c to the significantly lower dimensional latent space k ($k = c/8$). Each CpMoE contains 4

Model	Channels	Blocks	Heads	Param
Hiera-T	[96-192-384-768]	[1-2-7-2]	[1-2-4-8]	28M
Hiera-B+	[112-224-448-896]	[2-3-16-3]	[2-4-8-16]	70M
Hiera-L	[144-288-576-1152]	[2-6-36-4]	[2-4-8-16]	214M

Table 6: Configuration for Hiera variants. Channels, Blocks and Heads specify the channel width, number of Hiera blocks and heads in each block for the four stages, respectively. The stride of the four stages are [4, 8, 16, 32]. Our SATA employ the Hiera-L (Ryali et al. 2023) as our backbone network.

modality-common experts, and we activate the top-2 experts during inference.

SaMoE. In the designed SaMoE, the modality-specific experts shared the same architecture as that in the modality-common experts. SaMoE contains 4 modality-specific experts in each modality, and we also activate the top-2 experts in each modality during inference.

Candidates Generation Module

The designed Candidates Generation Module adopt a modified SAM2 (Videnovic et al. 2025) our foundation model to predict candidates based on the prior information of various tasks. In the following content, we first briefly revisit the employed foundation model-SAM2. Then, we introduce the process of generating potential candidates for SOT&VOS and MOT&MOTS, respectively.

Revisiting SAM2. SAM2 consists of four main components: (i) image encoder, (ii) prompt encoder, (iii) memory bank, and (iv) mask decoder.

Image encoder applies ViT Hiera backbone to embed the input image. Our SATA uses the unified embedding generated by the proposed DeMoE.

The prompt encoder takes two types of prompts, including sparse (e.g., points, bounding boxes) and dense (e.g., masks). The prompt tokens output by the prompt encoder can be represented as x_{prompt} .

The mask decoder is designed to take the memory-conditioned image embeddings produced by the memory attention layer along with the prompt tokens from the prompt encoder as its inputs. It can then generate a set of predicted masks $\{mask_1, \dots, mask_N\}$, along with the corresponding mask affinity score $\{s_{mask,1}, \dots, s_{mask,N}\}$, and a single occlusion score s_{occ} for each frame. The final output of the mask decoder is selected based on the highest affinity score.

The memory bank consists of the encoded initialization frame with a user-provided segmentation mask and six recent frames with segmentation masks generated by the tracking output. After the mask decoder generates output masks, the output mask is passed through a memory encoder to obtain a memory embedding. A new memory is created after each frame is processed. Please refer (Ravi et al. 2024) for more details.

Candidates Generation for SOT&VOS. For SOT and VOS tasks, initial target information is used to generate the initial prompt token and predict candidates with affinity scores, as shown in Fig. 6. Given the predicted masks $\{mask_1, \dots, mask_{N'}\}$ and their corresponding mask affinity scores $\{s_{mask,1}, \dots, s_{mask,N'}\}$, we select candidates only when $s_{occ} > 0$ and s_{mask} exceeds the preset threshold τ_{mask} . We then generate M candidates per frame. In our experiments, τ_{mask} is set to 0.7. The masks $\{mask_t^1, \dots, mask_t^M\}$ are used to generate bounding boxes $\{B_t^1, \dots, B_t^M\}$.

Candidates Generation for MOT&MOTS. For MOT and MOTS tasks, we append an additional detection head to the foundation

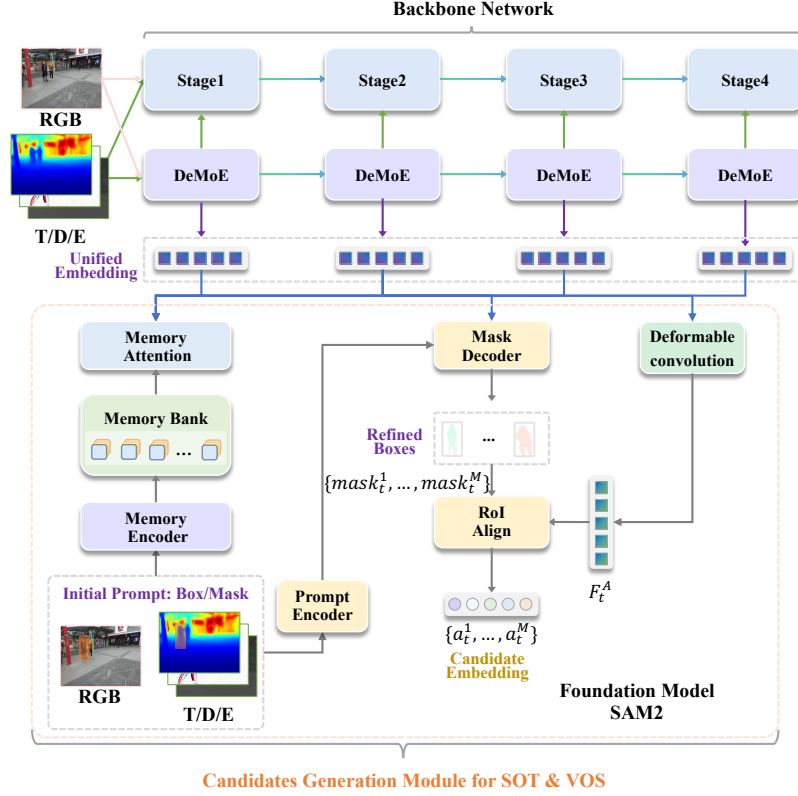


Figure 6: Details of our Candidates Generation Module (CGM) for SOT&VOS.

model to predict all potential objects in each frame, as shown in Fig. 7.

Detection head. Given unified embeddings generated by DeMoE with scale ratios of $\frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}$ (from Stage 1 to Stage 4 of the backbone network), we first use deformable convolution (Li et al. 2024c) to dynamically fuse multi-scale features, producing unified image embeddings F_t^A for the t -th frame. We then employ 4 stacked blocks (incorporating task-aware and scale-aware attention from Dynamic Head (Li et al. 2024c)), a classification predictor, and a box regressor to predict all potential instances.

Boxes refinement. As illustrated in our manuscript, predicted boxes from the detection head are fed as multi-object prompts to the mask decoder, generating masks $\{\text{mask}_t^1, \dots, \text{mask}_t^M\}$ for all candidates in the t -th frame. Leveraging SAM2’s strong capability in predicting fine-grained instance masks, we use these masks to correct the bounding boxes generated by the additional detection head, yielding refined boxes $\{B_t^1, \dots, B_t^M\}$.

Instance embedding. Using the refined boxes $\{B_t^1, \dots, B_t^M\}$, we extract instance embeddings $\{a_t^1, \dots, a_t^M\}$ via RoI-Align applied to the unified image embeddings F_t^A .

Memory-enhanced Module

Instance queries. In the proposed MEM module, we introduce learnable queries $q_t^m \in \mathbb{R}^{D \times C}$ ($m = 1, \dots, M$) for each candidate to capture video spatiotemporal information, employing the same architecture as the Querying Transformer (Q-Former) (Li et al. 2023). Here, D denotes the number of learned queries, and C represents the channel dimension. In our experiments, the Q-Former outputs 8 tokens for each instance to enable more efficient

storage of historical information.

Memory updating. Assuming that the Candidates Generation Module (CGM) generates M candidates for the t -th frame, while the Memory-enhanced Module stores N tracklets $\Gamma^n = \{f_{t-1}^n, f_{t-2}^n, \dots, f_{t-T}^n\}$ ($n = 1, \dots, N$). Candidates that fail to match any tracklet are assigned new IDs, and tracklets that do not match any instances over T^E consecutive frames are terminated, where T^E is experimentally set to 50. Historical tracklets are updated only after candidates are matched with them, and spatiotemporal relationships are calculated to facilitate instance matching in the next frame. The default length T of historical tracklets is set to 15. The memory and its management mechanism are visualized in Fig. 8.

Instance matching

This sub-section details the algorithm for instance matching association. Our proposed SATA employs a simple bi-softmax nearest neighbor search to generate the affinity matrix A . To avoid ambiguous candidate-tracklet matching, only pairs with affinity exceeding $\tau_{th} = 0.75$ can be associated. Subsequently, the Hungarian algorithm (Mayer et al. 2021) is applied to generate the tracking output.

Training Process

The overall training process consists of two stages: Detection Training and Mixture Training. We use the AdamW optimizer with a weight decay of 0.05. Detailed training hyperparameters are provided in Table 7. The model is trained on 8 A100 GPUs with a base learning rate of $2e^{-4}$. A StepLR learning rate scheduler is employed in each stage, where the learning rate is reduced by a

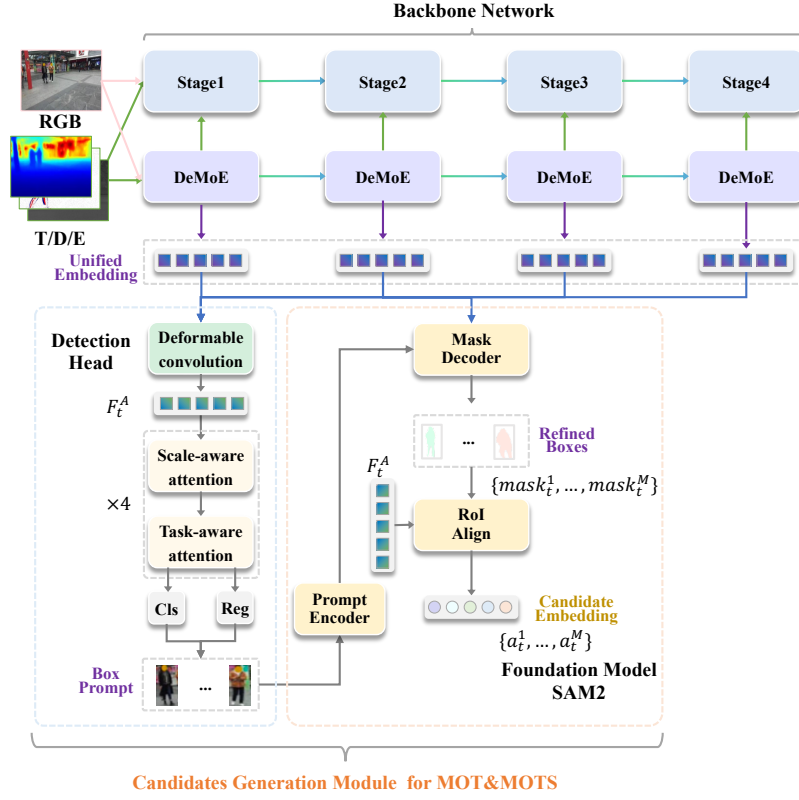


Figure 7: Details of our Candidates Generation Module (CGM) for MOT&MOTS.

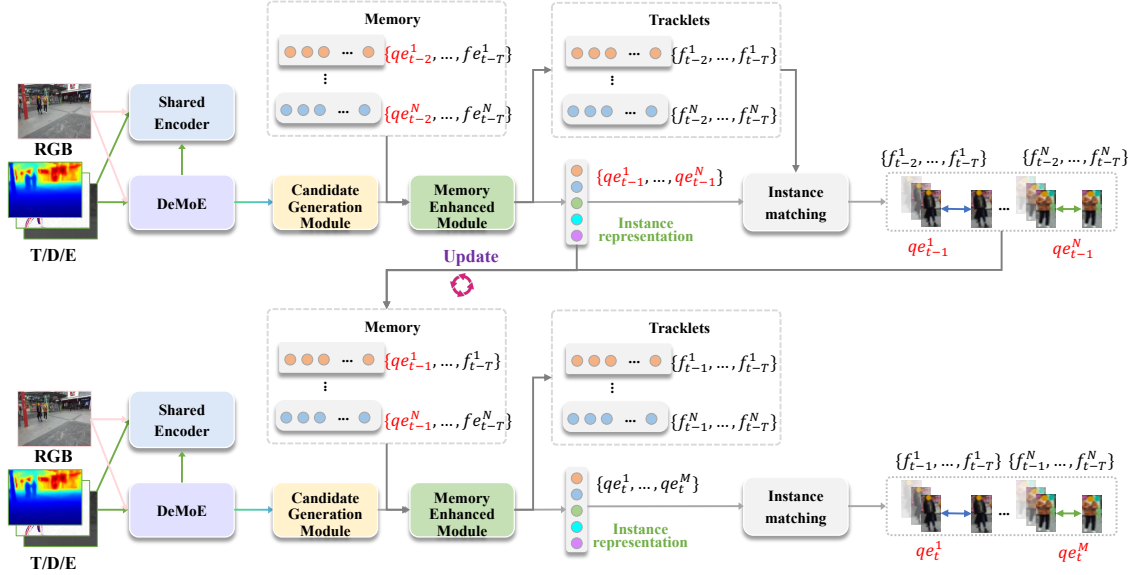


Figure 8: Illustration of memory updating pipeline.

factor of 10 after specified steps. Multi-scale training is adopted: original images are resized such that the shortest side ranges between 480 and 800 pixels, while the longest side does not exceed 1333 pixels (Wang et al. 2025).

Detection Training

In the first stage, we initialize with the official SAM2 weights (Ravi et al. 2024) and keep all SAM2 components frozen during training. Only the parameters of the additional detection head in the CGM

Stage	Task	Dataset	Sampling Weight	Batch Size	Num GPU	Lr	Step
I	RGB OD	COCO (Lin et al. 2014)	0.9	16	8	$2e^{-4}$	180K
	RGB-T OD	UniTRL (Zhang et al. 2025a)	0.1				
II	RGB SOT	LaSOT (Lin et al. 2014)	0.2	32	8	$3e^{-4}$	360K
		GOT10K (Huang et al. 2019)	0.2				
		TrackingNet (Muller et al. 2018)	0.2				
		COCO (Lin et al. 2014)	0.2				
	RGB-T SOT	LasHeR (Li et al. 2022)	0.33				
	RGB-D SOT	DepthTrack (Yan et al. 2021b)	0.33				
	RGB-E SOT	VisEvent (Wang et al. 2023)	0.33				
	RGB VOS	DAVIS (Pont-Tuset et al. 2017)	0.33				
		YouTube (Xu et al. 2018)	0.33				
		MOSE (Ding et al. 2023)	0.33				
	RGB-T VOS	VisT300 (Wang et al. 2023)	0.25				
	RGB-D VOS	ArKitTrack (Zhao et al. 2023)	0.25				
	RGB-E VOS	DepthTrack (Yan et al. 2021b)	0.25				
	RGB MOT&MOTS	LLE-VOS (Li et al. 2024b)	0.25				
	RGB MOT	BDD (Yu et al. 2020)	0.8				
		UniTRL (Zhang et al. 2025a)	0.2				

Table 7: Details in training. 'OD' denotes object detection.

are trainable. The model is trained on the COCO (Lin et al. 2014) and UniTRL (Zhang et al. 2025a) datasets with a batch size of 16 for 180K iterations.

Mixture Training

In the second stage, parameters of DeMoE, CGM, and MEM are set to be trainable. Our training data includes commonly used datasets for SOT and VOS tasks: COCO (Lin et al. 2014), LaSOT (Fan et al. 2019), GOT-10k (Huang et al. 2019), TrackingNet (Muller et al. 2018), DepthTrack (Yan et al. 2021b), VisEvent (Wang et al. 2023), LasHeR (Li et al. 2022), DAVIS (Pont-Tuset et al. 2017), YouTube (Xu et al. 2018), MOSE (Ding et al. 2023), VisT300 (Yang et al. 2023), ARKitTrack (Zhao et al. 2023), and LLE-VOS (Li et al. 2024b); as well as datasets for MOT and MOTS tasks: BDD (Yu et al. 2020) and UniRTL (Zhang et al. 2025a). Training is conducted with a batch size of 32 for 360K iterations. For SOT&VOS, each batch contains mixed data sampled from these datasets, with RGB data sampled at twice the rate of multi-modal data, following the approach in SUTrack (Chen et al. 2025). To ensure balanced performance across various benchmarks, we set the data sampling ratios as (SOT&VOS):(MOT&MOTS) = 0.6:0.4.

Loss functions

First, the original loss function L_{ORI} employed in SAM2 are used across all stages. Please find more details in SAM2 (Ravi et al. 2024). Secondly, to promote the unified representation learning, we employ the MoE loss L_{MoE} proposed in DeMoE. Thirdly, to associate instances on different frames, since only the targets' locations are provided in the existing SOT&VOS dataset, the partial supervision loss and self-supervised loss in KeepTrack (Mayer et al. 2021) are employed to supervise the affinity matrix A generated by TaMOT. For MOT&MOTS tasks, the cross-entropy loss is applied to optimize SATA.

MoE supervision

Cross-modal complementary learning loss. In order to promote the mutual complementarity among cross-modal experts, as illustrated

in our manuscript, the cross-modal complementary learning loss is defined as:

$$\mathcal{L}_{CM} = \sum_{l=1}^L \text{MSE}(\hat{H}G_l^R, HG_l^R) + \sum_{l=1}^L \text{MSE}(\hat{H}G_l^{TDE}, HG_l^{TDE}), \quad (12)$$

where $\text{MSE}(\cdot)$ denotes the mean squared error.

Cross-expert orthogonal loss. Furthermore, there exists a risk that multiple experts may converge to similar functions, which could undermine the overall effectiveness of Mixture-of-Expert. To address this issue, we design a cross-expert orthogonal loss that encourages each expert to specialize in distinct aspects of the data. This loss enhances the ability of experts to learn diverse information from multi-modal data and is defined as follows:

$$\mathcal{L}_{CE} = \sum_{l=1}^L \text{OPL}(\sum_{j=1}^{N^S} \sum_{i=1}^{N^G} g_i^C(T_i^X), g_j^X(T_i^X)), X = R, T, D, E \quad (13)$$

Here, $g_i^C(\cdot)$ and $g_j^X(\cdot)$ represents the outputs of experts from $\mathcal{G}^G$ and $\mathcal{G}^X$, respectively, $\text{OPL}(\cdot)$ denotes the orthogonal loss (Ranasinghe et al. 2021), which can be defined as:

$$\text{OHL} = \sum_{j=1}^{N^G} \sum_{k \neq j}^{N^S} \frac{\langle \tilde{x}_j, \tilde{x}_k \rangle}{\langle \tilde{x}_k, \tilde{x}_k \rangle} \tilde{x}_k, \quad (14)$$

where $\langle \cdot \rangle$ denotes the inner product between two vector, $\tilde{x}_j$ denotes the output of expert $g_j^C(\cdot)$ for input x , $\tilde{x}_k$ denotes the output of expert $g_k^X(\cdot)$ for input x .

Modality-aware loss. In addition, to supervise the cross-modal router and ensure accurate prediction of the input modality, we use the predicted probabilities to compute a cross-entropy loss against the true task labels y^{task} :

$$L_{TASK} = \sum_{n=1}^L \sum_{j=1}^K \left(-y_{task}(j) \cdot \log(s_n^{CM}(j)) \right), \quad (15)$$

where K denotes the number of tasks, $y_{task}(j)$ represents the ground-truth label for task j , and $s_n^{CM}(j)$ denotes the predicted probability for task j .

SOT/VOS supervision

In existing SOT/VOS datasets, only target objects and their corresponding locations are provided for training. To effectively super-

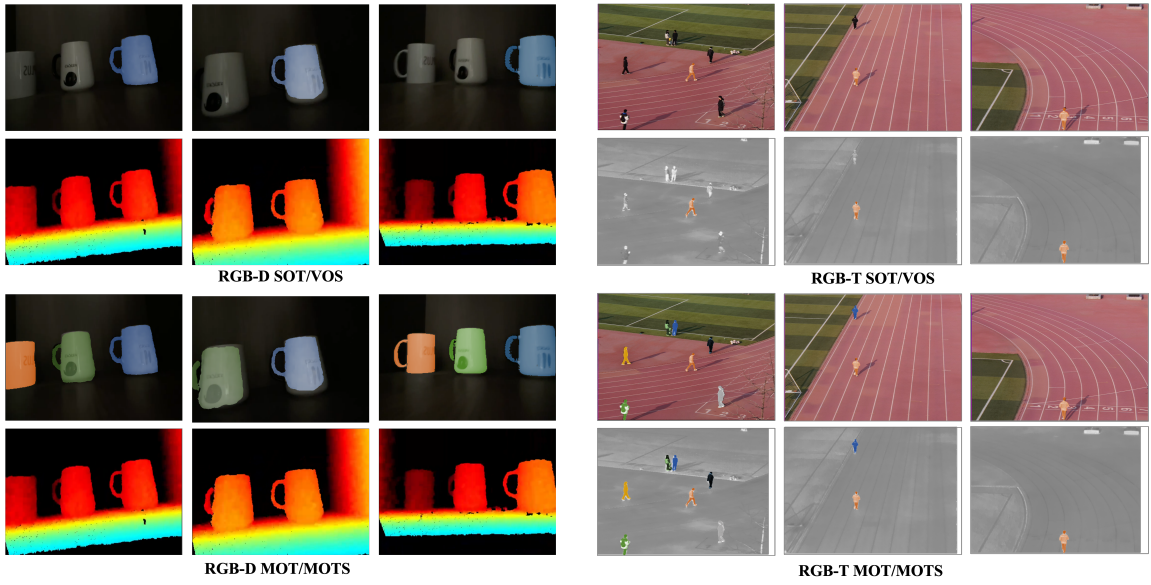


Figure 9: Visualizations of tracking results predicted by SATA.

vises the learning of TaMOT, we employ both the partial supervision loss and self-supervision loss from KeepTrack (Mayer et al. 2021).

Partially Supervised Loss. We formulate the problem of associating instances across two consecutive frames as obtaining an affinity matrix A between the two candidate sets. For candidate v_i^t in frame t that corresponds to candidate v_j^{t-1} in frame $t-1$, we set $A_{ij} = 1$; otherwise, $A_{ij} = 0$. For each consecutive frame pair in a video sequence, we retrieve the single candidate corresponding to the annotated target. For the candidates $\{v_i^{t-T}, \dots, v_i^t\}$, the assignment matrix A generated by our proposed TaMOT reflects the association between v_i^t and v_i^{t-1} . The supervised loss is then defined as the negative log-likelihood of the assignment probability:

$$L_{\text{sup}} = -\log A_{i,i}. \quad (16)$$

Self-Supervised Loss. To enhance the robustness of TaMOT, given a predicted candidate set $\mathcal{V}^t$ and its corresponding ground-truth association set $\mathcal{C} = \{(i, i)\}_{i=1}^N$, we generate a series of candidate sets $\{\mathcal{V}^{t-T}, \dots, \mathcal{V}^{t-1}\}$ from $\mathcal{V}^t$ via feature augmentation. This augmentation involves randomly translating candidate locations, randomly adjusting candidate affinity scores, and transforming the input image before extracting features. The self-supervised loss is formulated as:

$$L_{\text{self}} = \sum_{(i,j) \in \mathcal{C}} -\log A_{i,j}. \quad (17)$$

Finally, we combine the self-supervised loss and partially supervised loss as $L_{\text{TaMOT-S}} = L_{\text{self}} + L_{\text{sup}}$.

MOT supervision

For MOT/MOTS tasks, given the ground-truth assignment matrix A^g , we formulate the prediction of the assignment matrix A as a binary classification problem. Accordingly, we employ cross-entropy loss to optimize the network:

$$\mathcal{L}_{\text{TaMOT-M}} = \frac{-1}{MN} \sum_{i=1}^N \sum_{j=1}^M [A_{i,j}^g \log(A_{i,j}) + (1 - A_{i,j}^g) \log(1 - A_{i,j})], \quad (18)$$

where $A_{i,j}$ and $A_{i,j}^g$ denote the (i, j) -th elements of the predicted assignment matrix A and the ground-truth assignment matrix A^g , respectively.

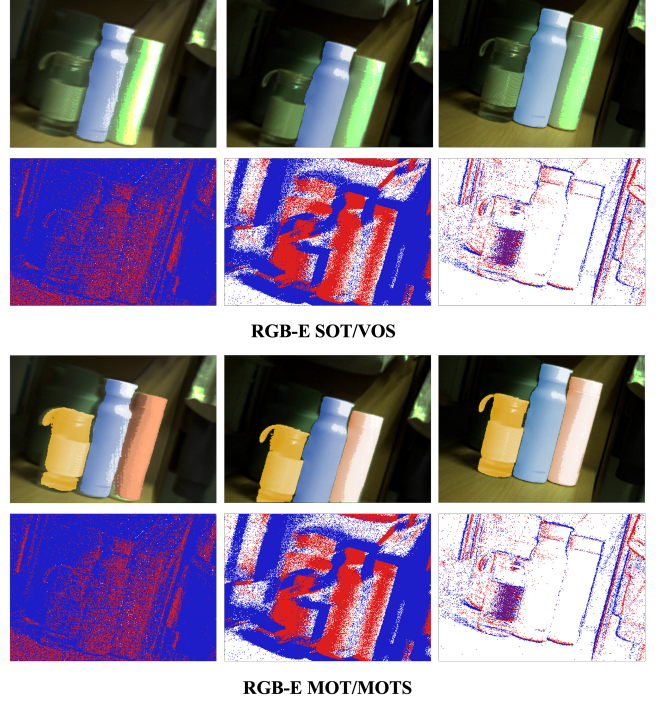


Figure 10: Visualizations of tracking results predicted by SATA.

Additional Experiments

Visualizations

As illustrated in Fig. 9 and Fig. 10, we visualize the tracking results of SATA across different tasks. SATA can accurately locate targets based on their state in the initial frame. Additionally, with the designed TaMOT pipeline, our SATA is capable of tracking multiple objects belonging to specific categories. By consistently tracking multiple potential objects, SATA effectively alleviates task conflicts and bridges the data distribution gap between various downstream tasks. Notably, even in the absence of RGB-D and RGB-E training data for MOT&MOTS tasks, SATA can still precisely predict object positions and establish accurate associations.

Ablation studies

Unified Embedding. To investigate the effectiveness of the proposed DeMoE, we implement several variants by replacing DeMoE with existing methods, including: ①: Replacing DeMoE with the Shared Embedding from UnTrack (Wu et al. 2024). ②: Replacing DeMoE with the MeMoE from XTrack (Tan et al. 2025b). ③: Replacing DeMoE with HMOE-Fuse from FlexTrack (Tan et al. 2025a). ④: Replacing DeMoE with the Unified Modality Representation from SUTrack (Chen et al. 2025).

We conduct ablation studies on RGB-X SOT tasks, evaluating all models on LasHeR (Li et al. 2022) (PR score), DepthTrack (Yan et al. 2021b) (PR score), and VisEvent (Wang et al. 2023) (PR score). As shown in Table 8, the proposed DeMoE outperforms all these variants.

Method	RGB-T SOT	RGB-D SOT	RGB-E SOT
	LasHeR (PR)	DepthTrack (PR)	VisEvent (PR)
SATA	77.8	67.9	82.8
Shared Embedding	74.2	63.7	77.5
MeMoE	75.8	65.2	79.4
HMOE-Fuse	73.2	65.9	78.7
Unified Modality Representation	75.3	64.8	80.1

Table 8: Ablation studies on the unified embedding

Method	SOT	SOT	VOS	MOT	MOTS
	GOT10K (AO)	LasHeR (PR)	LLE-VOS ($\mathcal{J}\&\mathcal{F}$)	UniRTL HOTA	BDD MOTS MOTSA
SATA	81.3	77.8	71.4	59.7	38.1
Separate models	78.5	74.3	68.7	56.4	36.3
Task-specific heads	80.8	75.4	67.5	54.2	35.5
Multi-stage training	79.5	75.2	68.4	57.3	37.6

Table 9: Ablation studies on the training strategy.

Training Strategy To investigate the effectiveness of the proposed TaMOT and our unified training strategy, we implement several comparative variants using existing methods, including: ①: Training separate models for the SOT&VOS task and MOT&MOTS task, respectively. ②: Employing task-specific heads. ③: Adopting the multi-stage training strategy from (Yan et al. 2022). These methods are evaluated across 5 benchmarks (GOT10K (Huang et al. 2019), LasHeR (Li et al. 2022), LLE-VOS (Li et al. 2024b), UniRTL (Zhang et al. 2025a), and BDD MOTS (Yu et al. 2020)) covering 4 subtasks (SOT, VOS, MOT, MOTS).

Compared with our unified model, single-task models exhibit inferior performance across all tasks. Experimental results confirm that the proposed TaMOT can leverage expanded training data and further enhance the model’s generalization capability through multi-task joint training.

Limitation

The current SATA is dedicated to constructing an effective framework for tracking and segmentation with any modality input, but it pay less attention on the efficiency of the proposed model. When tracking and segmenting multiple targets, the proposed method adopts a strategy of sharing unified embedding but tracking and segmenting these objects separately. Such a manner lacks interaction between objects and affects the efficiency of the model to some extent.