UNIMERGETRACK: HIERARCHICAL MODEL MERGING FOR GENERAL RGB-X VISUAL OBJECT TRACKING

Anonymous authors
Paper under double-blind review

ABSTRACT

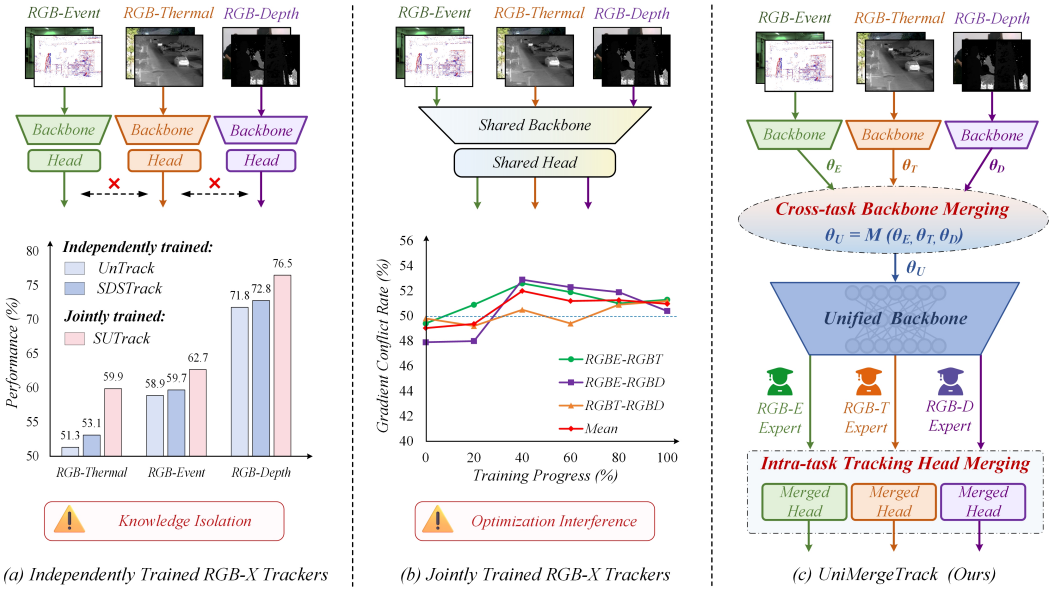
RGB-X tracking leverages complementary modalities, such as thermal infrared, event, or depth, to improve robustness under challenging conditions. Existing general-purpose RGB-X trackers typically follow two paradigms: independent task-specific training and joint training across modality combinations. However, the former suffers from knowledge isolation and model redundancy, while the latter introduces optimization interference across heterogeneous tasks. In this work, we present UniMergeTrack, a new model merging paradigm that preserves task-specific optimization while enabling effective cross-task knowledge integration. UniMergeTrack first trains modality-specific trackers independently and then performs hierarchical model merging. Specifically, transferable backbones are merged across RGB-X tasks via high-pass-filtered merging, and complemented by lightweight task-specific experts that preserve modality-specific cues. Tracking heads are merged within each task using optimal transport-guided channel alignment to mitigate hidden channel mismatch. A subsequent distribution alignment fine-tuning stage further improves compatibility between the unified backbone and task-specific heads. Extensive experiments on the LasHeR, RGBT234, VisEvent, VOT-RGBD22, COESOT, and CDTB benchmarks demonstrate that UniMergeTrack achieves state-of-the-art or competitive performance across RGB-Thermal, RGB-Event, and RGB-Depth tracking tasks, while exhibiting strong generalization to unseen datasets. The source code will be released upon acceptance.

1 INTRODUCTION

Visual object tracking (VOT) aims to continuously localize a target in a video given its initial state, serving as a fundamental component in various vision applications, including autonomous driving, surveillance, and human-computer interaction. Deep learning has driven significant advances in RGB-based trackers (Yan et al., 2021a; Cui et al., 2022; Ye et al., 2022; Chen et al., 2023b). However, RGB cameras remain vulnerable to adverse conditions such as low illumination, motion blur, occlusion, and background clutter, where appearance cues may become ambiguous or unreliable. Auxiliary modalities, including thermal infrared, event, and depth, provide complementary information beyond visible appearance. Multimodal tracking has therefore gained increasing attention, with early methods tailored to specific modality combinations such as RGB-Thermal (Wang et al., 2020; Zhang et al., 2021b), RGB-Event (Zhang et al., 2021a; Wang et al., 2024), and RGB-Depth (Kart et al., 2019; Yan et al., 2021b), but generalizing poorly across different modality combinations.

More recently, general RGB-X (X denotes thermal infrared, event, or depth data) multimodal trackers have been developed to handle multiple modality combinations within a unified framework. Existing approaches mainly follow two paradigms. The first adopts a general tracking architecture but trains it independently for each modality combination (Zhu et al., 2023; Hou et al., 2024; Wu et al., 2024), resulting in multiple task-specific trackers. As shown in Fig. 1 (a), although such independent training preserves task-specific specialization, it leads to model redundancy and knowledge isolation across RGB-X tasks. The second jointly trains a single model on mixed RGB-X data (Chen et al., 2025; Hong et al., 2026; Geng et al., 2026), enabling shared representations and improved cross-task knowledge transfer. However, as illustrated in Fig. 1 (b), joint training suffers from persistent gradient conflicts across tasks (measured by the Gradient Conflict Rate defined based on negative

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Independently Trained RGB-X Trackers (b) Jointly Trained RGB-X Trackers (c) UniMergeTrack (Ours)

Figure 1: **Comparison of three paradigms for RGB-X tracking.** (a) Independent training retains task specialization but suffers from knowledge isolation. (b) Joint training enables knowledge sharing but introduces optimization interference. (c) UniMergeTrack achieves unified tracking via hierarchical model merging with cross-task backbone and intra-task head merging.

gradient cosine similarity (Yu et al., 2020; Gong et al., 2022)), suggesting optimization interference among heterogeneous RGB-X tasks and potentially constraining task-specific specialization.

These complementary limitations motivate us to explore an alternative paradigm that preserves task specialization while enabling effective cross-task knowledge integration. To this end, we introduce **UniMergeTrack**, a model merging paradigm for general RGB-X tracking, as illustrated in Fig. 1 (c). Instead of jointly optimizing heterogeneous modality combinations in a shared parameter space, UniMergeTrack first trains modality-specific trackers independently and then integrates their knowledge through hierarchical model merging. Specifically, we merge backbone parameters across RGB-X tasks via high-pass-filtered merging to construct a unified backbone, complemented by lightweight task-specific experts that retain modality-specific cues. In contrast, tracking heads are more closely coupled to task-specific prediction behavior; therefore, we merge them within each task using optimal transport-guided channel alignment. Finally, distribution alignment fine-tuning is performed to mitigate the distribution shifts introduced by model merging and ensure compatibility between the unified backbone and task-specific tracking heads. In this way, UniMergeTrack retains task-specific specialization while integrating cross-task knowledge, providing an effective alternative to conventional paradigms.

To sum up, the contributions of this work can be summarized as follows: 1) We present UniMergeTrack, the first model merging paradigm for RGB-X tracking, which alleviates knowledge isolation and optimization interference in conventional paradigms. 2) We propose a hierarchical model merging strategy, with high-pass-filtered cross-task merging for backbones and OT-guided intra-task merging for tracking heads. 3) Extensive experiments on multiple RGB-X tracking benchmarks, including LasHeR, RGBT234, VisEvent, VOT-RGBD22, COESOT, and CDTB, demonstrate the effectiveness and generalization ability of the proposed tracker.

2 Related Work

2.1 RGB-X Tracking

RGB-X tracking leverages an auxiliary modality, such as thermal infrared, event, or depth data, to complement RGB observations under challenging conditions. Early multimodal trackers were typi-

cally tailored to a specific modality combination, improving robustness within their target settings. Representative methods include dedicated RGB-Event trackers (Zhang et al., 2021a; Wang et al., 2024; Tang et al., 2025), RGB-Thermal trackers (Wang et al., 2020; Zhang et al., 2021b; Xiao et al., 2022), and RGB-Depth trackers (Kart et al., 2019; Qian et al., 2020; Yan et al., 2021b). However, such modality-specific designs generalize poorly across modality combinations. Recent works have therefore explored general RGB-X tracking. Existing approaches mainly follow two paradigms. The first employs a shared tracking framework while training it separately for each modality combination, as in ViPT (Zhu et al., 2023), SDSTrack (Hou et al., 2024), and OneTracker (Hong et al., 2024). The second jointly trains a single model on mixed data from multiple modality combinations, enabling one parameter set to handle diverse RGB-X tracking tasks, as in SUTrack (Chen et al., 2025), OneTrackerV2 (Hong et al., 2026), and TGTrack (Geng et al., 2026). While both paradigms improve generalization across modality combinations, they exhibit complementary limitations. Independent adaptation leads to model redundancy and limited knowledge sharing across tasks, whereas joint training suffers from optimization interference among heterogeneous modality combinations, potentially constraining task-specific specialization. In contrast, we introduce a new model merging paradigm for RGB-X tracking, which combines the strengths of both approaches while mitigating knowledge isolation and optimization interference.

2.2 Model Merging

Model merging consolidates multiple fine-tuned models into a single model, reducing redundancy while improving cross-task generalization. Early studies established weight-space averaging as a simple yet effective strategy. Stochastic Weight Averaging (SWA) (Izmailov et al., 2018) averages checkpoints along an optimization trajectory, while Model Soups (Wortsman et al., 2022) extends this idea to models fine-tuned from a shared pretrained initialization. Recent methods increasingly focus on merging task-specific models derived from a common pretrained model. Fisher Merging (Matena & Raffel, 2022) weights parameters by their estimated importance. Task Arithmetic (Ilharco et al., 2023) represents task knowledge as parameter displacements from the shared initialization and composes them through vector arithmetic. To mitigate redundancy and destructive interference, TIES-Merging (Yadav et al., 2023) trims insignificant updates and resolves sign conflicts, while DARE (Yu et al., 2024) sparsifies task vectors through random dropping and rescaling. FREE-Merging (Zheng & Wang, 2025) further addresses task interference in the frequency domain, whereas RAIN-Merging (Huang et al., 2026) employs null-space projection and module-specific scaling to preserve existing capabilities. Despite these advances, most methods remain architecture-agnostic, applying the same merging rule to transferable backbones and task-specific heads, potentially causing information misalignment. We adopt a hierarchical merging strategy that merges backbones across tasks and tracking heads within each task.

3 Our Proposed Approach

3.1 Preliminary

We build on Task Arithmetic (TA) (Ilharco et al., 2023), which represents task-specific knowledge as parameter shifts from a shared pretrained model. Let $\boldsymbol{\theta}_{pre}$ and $\boldsymbol{\theta}_k$ denote the pretrained and task-k fine-tuned parameters, respectively. The task vector is defined as

$$\boldsymbol{v}_k = \boldsymbol{\theta}_k - \boldsymbol{\theta}_{pre}. \tag{1}$$

Given K task-specific models, their acquired knowledge can be integrated by performing arithmetic operations on the corresponding task vectors:

$$\boldsymbol{\theta}_{\text{merge}} = \boldsymbol{\theta}_{pre} + \sum_{k=1}^{K} \lambda_k \boldsymbol{v}_k, \tag{2}$$

where λ_k controls the contribution of the k-th task vector. This formulation enables knowledge acquired from multiple tasks to be composed without joint training on their combined datasets.

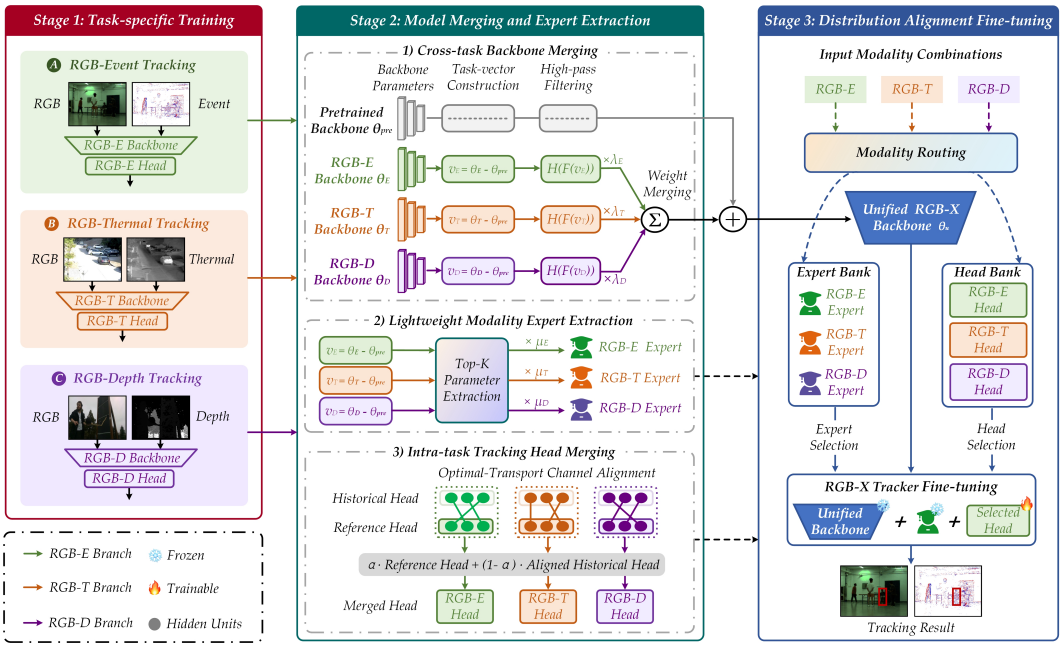


Figure 2: **Overview of the proposed model merging framework for RGB-X tracking.** We first train RGB-E, RGB-T, and RGB-D trackers independently, then perform hierarchical model merging by merging high-pass-filtered backbone task vectors across tasks and OT-aligned tracking heads within each task, while extracting lightweight modality experts. Finally, distribution alignment fine-tuning improves the compatibility between the unified backbone and task-specific tracking heads.

3.2 Overview

We propose UniMergeTrack, a model merging framework for general RGB-X tracking. In the first stage, we train RGB-Event, RGB-Thermal, and RGB-Depth trackers independently. In the second stage, we merge backbone parameters across tasks while extracting lightweight task-specific experts, and merge tracking heads within each task. In the third stage, we freeze the merged backbone and experts, fine-tuning tracking heads and other lightweight parameters to improve the compatibility between the unified backbone and tracking heads. The overall framework is illustrated in Fig. 2.

3.3 Task-specific Training

To support multiple multimodal SOT tasks within a unified one-stream framework, we initialize all task-specific models from the same pretrained weights and independently train them on the corresponding RGB-Event, RGB-Thermal, and RGB-Depth datasets.

Following SUTrack (Chen et al., 2025), we concatenate an input RGB image $\mathbf{I}_{\mathrm{RGB}} \in \mathbb{R}^{H\times W\times 3}$ with its corresponding X-modality image $\mathbf{I}_{\mathrm{X}} \in \mathbb{R}^{H\times W\times 3}$ along the channel dimension, producing a multimodal input $\mathbf{I}_{\mathrm{concat}} \in \mathbb{R}^{H\times W\times 6}$, where X represents an event, thermal infrared, or depth modality. The concatenated input is then partitioned into non-overlapping patches of size $P \times P$, each of which is flattened into a $6P^2$-dimensional vector. Let $\mathbf{F}_p \in \mathbb{R}^{N\times 6P^2}$ denote the resulting sequence of flattened patch vectors, where N is the number of patches. These vectors are mapped into a D-dimensional embedding space through a learnable linear projection:

$$\mathbf{E} = \mathbf{F}_p\mathbf{W}_{\mathrm{proj}} + \mathbf{b}_{\mathrm{proj}}, \tag{3}$$

where $\mathbf{W}_{\mathrm{proj}} \in \mathbb{R}^{6P^2\times D}$ and $\mathbf{b}_{\mathrm{proj}} \in \mathbb{R}^{D}$ denote the projection matrix and bias vector, respectively. This projection produces the multimodal patch embeddings $\mathbf{E} \in \mathbb{R}^{N\times D}$.

The template and search regions are embedded identically, concatenated along the token dimension, and fed into the Transformer backbone for joint feature extraction and cross-modal interaction. The

resulting features are passed to a center-based tracking head for target prediction. We use this pipeline to independently train RGB-Event, RGB-Thermal, and RGB-Depth trackers.

3.4 Model Merging and Expert Extraction

Cross-task Backbone Merging. Given the independently trained backbone parameters $\boldsymbol{\theta}_E$, $\boldsymbol{\theta}_T$, and $\boldsymbol{\theta}_D$ for RGB-Event, RGB-Thermal, and RGB-Depth tracking, respectively, we integrate their knowledge into a unified backbone that supports all three modality combinations. Since these backbones share the same pretrained initialization $\boldsymbol{\theta}_{pre}$, we construct the corresponding task vectors as:

$$\boldsymbol{v}_k = \boldsymbol{\theta}_k - \boldsymbol{\theta}_{pre}, \qquad k \in \{E, T, D\}, \tag{4}$$

where $\boldsymbol{v}_k$ characterizes the parameter displacement induced by task-specific training.

Directly merging task vectors can induce cross-task interference (Yadav et al., 2023) due to conflicting updates across different modality combinations. As low-frequency components are particularly susceptible to such interference (Zheng & Wang, 2025), we apply high-pass filtering to each task vector before merging to suppress these conflict-prone components.

For the l-th parameter tensor, we first transform each task-vector slice into the frequency domain:

$$\widehat{\boldsymbol{v}}_k^{\,l} = \mathcal{F}\big(\boldsymbol{v}_k^{\,l}\big)\,, \tag{5}$$

where $\mathcal{F}(\cdot)$ denotes the discrete Fourier transform. We then apply a radial high-pass mask $\mathbf{H}_k^l$:

$$\mathbf{H}_k^l(u,v) = \begin{cases} 1, & A_l(u,v) \geq \rho_l, \\ 0, & A_l(u,v) < \rho_l, \end{cases} \qquad \rho_l = \eta \cdot \min(H_l, W_l). \tag{6}$$

where $A_l(u,v)$ is the radial distance from frequency coordinate (u,v) to the spectral center, ρ_l is the cutoff radius, and H_l and W_l are the spatial dimensions of the frequency representation. We set the filtering ratio η to 0.7 by default. The filtered task vector is then obtained as

$$\widetilde{\boldsymbol{v}}_k^{\,l} = \mathcal{F}^{-1}\big(\mathbf{H}_k^l \odot \widehat{\boldsymbol{v}}_k^{\,l}\big)\,, \tag{7}$$

where $\odot$ denotes element-wise multiplication and $\mathcal{F}^{-1}(\cdot)$ is the inverse Fourier transform. This filtering suppresses low-frequency interference while preserving high-frequency variations.

Finally, the unified RGB-X backbone is constructed by merging the filtered task vectors into the shared pretrained backbone:

$$\boldsymbol{\theta}_U = \boldsymbol{\theta}_{pre} + \sum_{k \in \{E,T,D\}} \lambda_k \widetilde{\boldsymbol{v}}_k, \tag{8}$$

where λ_k denotes the merging coefficient, which is set to 0.25 by default.

Lightweight Modality Expert Extraction. Given a task vector $\boldsymbol{v}_k$, where $k \in \{E, T, D\}$, the magnitude of fine-tuning-induced parameter updates provides a natural measure of task specialization, with larger updates typically reflecting stronger task-specific adaptation. To extract a sparse modality expert, $R_\gamma(\cdot)$ first removes the top γ fraction of entries by magnitude, and $M_d(\cdot)$ then retains the top d fraction of the remaining entries. The sparse task expert is defined as

$$\hat{\boldsymbol{E}}_k = M_d\left(R_\gamma\left(\boldsymbol{v}_k\right)\right), \tag{9}$$

where $d \ll 1$ controls the sparsity of the expert. When $\gamma = 0$, this formulation reduces to standard magnitude-based Top-d selection.

Although sparse extraction reduces parameter overhead, it alters the statistics of the original task vector and may weaken the expert contribution when combined with the unified backbone. To compensate, we further introduce a task-wise rescaling strategy. Let $\bar{\boldsymbol{v}}_k$ denote the shared task vector used in backbone merging, and $\mathcal{E}(\cdot)$ denote the mean absolute parameter value averaged across parameter tensors. The rescaling factor and the final task expert are defined as

$$\mu_k = -\frac{\mathcal{E}\left(\hat{\boldsymbol{E}}_k\right) \log d}{d\,\mathcal{E}(\bar{\boldsymbol{v}}_k)}, \qquad \boldsymbol{E}_k = \mu_k \hat{\boldsymbol{E}}_k. \tag{10}$$

This rescaling compensates for the magnitude shift caused by sparse extraction, preserving the contribution of retained updates under high sparsity. Thus, each RGB-X task requires only a lightweight expert rather than a separate backbone. The resulting expert bank, $\mathcal{B}_E = \{\boldsymbol{E}_E, \boldsymbol{E}_T, \boldsymbol{E}_D\}$, enables task-specific adaptation to the unified backbone with minimal storage overhead.

Intra-task Tracking Head Merging. Unlike the backbone, which learns transferable representations and benefits from cross-task merging, the tracking head is closely tied to task-specific prediction distributions. Cross-task head merging may therefore introduce interference across heterogeneous objectives. We instead retain a separate head for each task and merge checkpoints within the same task to consolidate knowledge from different optimization stages.

However, even heads trained for the same task may develop inconsistent hidden-channel organizations, making direct parameter merging unreliable. For each task k, we align a historical head $\boldsymbol{h}_k^{\text{his}}$ with the final reference head $\boldsymbol{h}_k^{\text{ref}}$ using optimal transport (OT) (Cuturi, 2013). Each output channel is represented by a neuron descriptor constructed from normalized incoming weights, and the pairwise channel cost is defined as

$$\mathbf{C}_{ij} = \left\| \boldsymbol{d}_i^{\text{his}} - \boldsymbol{d}_j^{\text{ref}} \right\|_2^2, \tag{11}$$

where $\boldsymbol{d}_i^{\text{his}}$ and $\boldsymbol{d}_j^{\text{ref}}$ denote the descriptors of the i-th historical and j-th reference channels, respectively. We then solve the entropy-regularized OT problem

$$\mathbf{P}^* = \arg\min_{\mathbf{P}\geq 0} \langle \mathbf{P}, \mathbf{C} \rangle_F - \epsilon H(\mathbf{P}), \quad \text{s.t. } \mathbf{P}\mathbf{1} = \frac{1}{B_s}\mathbf{1}, \quad \mathbf{P}^\top \mathbf{1} = \frac{1}{B_t}\mathbf{1}, \tag{12}$$

where $\mathbf{P}^* \in \mathbb{R}^{B_s \times B_t}$ is the optimal transport plan, B_s and B_t are the numbers of historical and reference channels, and $H(\mathbf{P}) = -\sum_{i,j} P_{ij} \log P_{ij}$ is the entropy term with regularization coefficient $\epsilon > 0$. We obtain $\mathbf{P}^*$ via Sinkhorn iterations (Cuturi, 2013) and propagate the resulting correspondence layer by layer to align the historical head with the reference head before merging. After that, the aligned historical head is merged with the reference head through weighted parameter averaging:

$$\boldsymbol{H}_k = \alpha \boldsymbol{h}_k^{\text{ref}} + (1-\alpha)\widetilde{\boldsymbol{h}}_k^{\text{his}}, \tag{13}$$

where $\widetilde{\boldsymbol{h}}_k^{\text{his}}$ denotes the OT-aligned historical head and α controls the contribution of the reference head. The resulting task-specific heads form the head bank $\mathcal{B}_H = \{\boldsymbol{H}_E, \boldsymbol{H}_T, \boldsymbol{H}_D\}$.

3.5 Distribution Alignment Fine-tuning

Although model merging integrates shared and modality-specific knowledge, the recomposed components may not be fully compatible. In particular, replacing the original task-specific backbone with the unified backbone shifts the feature distribution seen by the tracking head. We therefore perform lightweight post-merging fine-tuning to improve their compatibility.

For each modality $k \in \{E, T, D\}$, we compose the corresponding lightweight expert $\boldsymbol{E}_k$ with the unified backbone $\boldsymbol{\theta}_U$ through sparse parameter overriding, yielding a task-specialized backbone:

$$\left(\boldsymbol{\theta}_U^k\right)_i = \begin{cases} \left(\boldsymbol{E}_k\right)_i, & i \in \mathcal{I}_k, \\ \left(\boldsymbol{\theta}_U\right)_i, & i \notin \mathcal{I}_k, \end{cases} \tag{14}$$

where $\mathcal{I}_k$ denotes the sparse parameter indices retained by the task expert. The resulting backbone integrates the cross-task knowledge retained by the unified backbone with the complementary task-specific knowledge encoded by the lightweight experts.

We pair each backbone $\boldsymbol{\theta}_U^k$ with its corresponding tracking head $\boldsymbol{H}_k$, and independently fine-tune the resulting tracker on the task-specific training distribution $\mathcal{D}_k$. The optimization objective is

$$\min_{\boldsymbol{\phi}_k} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}_k} \left[\mathcal{L}_{\text{total}} \left(f_k(\mathbf{x}; \boldsymbol{\theta}_U^k, \boldsymbol{\phi}_k), \mathbf{y} \right) \right], \qquad \boldsymbol{\theta}_U^k \text{ frozen}, \quad k \in \{E, T, D\}, \tag{15}$$

where $\boldsymbol{\phi}_k$ comprises the trainable tracking head and other lightweight task-specific parameters, while the unified backbone remains frozen to prevent its consolidated parameters from being overwritten. This stage enhances the compatibility between the backbone and its task-specific head.

3.6 Tracking Head and Loss Function

Following SUTrack (Chen et al., 2025), we adopt a center-based tracking head to predict the target location and bounding box from the search features. The tracking objective consists of a weighted focal loss $\mathcal{L}_{\text{cls}}$ for target classification, together with the generalized IoU loss $\mathcal{L}_{\text{GIoU}}$ and ℓ_1 loss $\mathcal{L}_{\ell_1}$ for bounding-box regression. We further employ a cross-entropy loss $\mathcal{L}_{\text{task}}$ for task recognition. The overall objective is

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{cls}} + \lambda_{\text{GIoU}}\mathcal{L}_{\text{GIoU}} + \lambda_{\ell_1}\mathcal{L}_{\ell_1} + \lambda_{\text{task}}\mathcal{L}_{\text{task}}, \tag{16}$$

where $\lambda_{\text{GIoU}} = 2$, $\lambda_{\ell_1} = 5$, and $\lambda_{\text{task}} = 1$. The same objective and loss weights are used for both initial training and post-merging fine-tuning.

4 Experiments

4.1 Implementation Details

Model Architecture. UniMergeTrack is built upon the SUTrack (Chen et al., 2025) architecture and adopts HiViT-base (Zhang et al., 2023) for patch embedding and backbone feature extraction. The model is initialized with weights (Chen et al., 2026) pretrained on LaSOT (Fan et al., 2019), TrackingNet (Muller et al., 2018), COCO (Lin et al., 2014), VastTrack (Peng et al., 2024), and GOT-10k (Huang et al., 2021). The model is implemented in Python 3.8 with PyTorch 1.11. All experiments are conducted on a single NVIDIA RTX 4090 GPU.

Training Setup. *For the task-specific training stage*, we independently train RGB-Thermal, RGB-Event, and RGB-Depth models on LasHeR (Li et al., 2022), VisEvent (Wang et al., 2024), and DepthTrack (Yan et al., 2021b), respectively. To ensure a fair comparison with existing RGB-X trackers, we additionally include RGB data from TNL2K (Wang et al., 2021) during this stage. The template and search regions are resized to 112×112 and 224×224, respectively. Each task is trained for 30 epochs, with 100,000 training pairs sampled per epoch. We adopt AdamW (Loshchilov & Hutter, 2019) as the optimizer, with initial learning rates of 1×10^{-5} for the Transformer encoder and 1×10^{-4} for the remaining parameters. The learning rates are decayed by a factor of 10 after 24 epochs, with a weight decay of 1×10^{-4}. *During the distribution alignment fine-tuning stage*, the unified backbone is frozen, while the tracking heads are separately fine-tuned on their corresponding RGB-X tasks. TNL2K is excluded at this stage, and all other training settings remain unchanged.

4.2 Comparison on Public Benchmark Datasets

• **Results on LasHeR Dataset.** As shown in Table 1, we compare UniMergeTrack with existing mainstream trackers on the LasHeR dataset. UniMergeTrack achieves an AUC of 61.8% and a PR of 77.2%, attaining the best AUC among all compared methods. Compared with the independently trained XTrack and the jointly trained SUTrack, our method improves PR by 8.1 and 2.7 percentage points, respectively. These results demonstrate that our model merging paradigm effectively integrates cross-task knowledge while preserving task-specific tracking capability.

• **Results on RGBT234 Dataset.** On the RGBT234 dataset, UniMergeTrack achieves 70.6% MSR and 94.0% MPR, ranking first on both metrics. Compared with XTrack, it improves MSR and MPR by 5.7 and 6.6 percentage points, respectively, and outperforms SUTrack by 1.1 and 1.8 points. These consistent gains further validate the effectiveness of our method for RGB-Thermal tracking.

• **Results on VisEvent Dataset.** As illustrated in Table 1, we also report results on the RGB-Event tracking task. UniMergeTrack achieves 64.1% AUC and 81.5% PR on VisEvent, establishing new state-of-the-art performance on both metrics. It surpasses the jointly trained SUTrack by 1.6 percentage points in PR and the recent OneTrackerV2 by 1.7 percentage points. These results further demonstrate the effectiveness of UniMergeTrack in exploiting motion cues from event data.

• **Results on VOT-RGBD22 Dataset.** We further evaluate UniMergeTrack on VOT-RGBD22 for RGB-Depth tracking. As shown in Table 1, it achieves 76.8% EAO, 83.1% Accuracy, and 92.3% Robustness, obtaining the best Accuracy and competitive overall performance. These results further validate the effectiveness of UniMergeTrack for RGB-Depth tracking.

Table 1: Comparison results on LasHeR, RGBT234, VisEvent, and VOT-RGBD22 datasets. The best and second-best results are highlighted in **bold** and underlined, respectively.

Method	Source	LasHeR		RGBT234		VisEvent		VOT-RGBD22		
		AUC	PR	MSR	MPR	AUC	PR	EAO	Acc.	Rob.
OSTrack (Ye et al., 2022)	ECCV22	41.2	51.5	54.9	72.9	53.4	69.5	67.6	80.3	83.3
ProTrack (Yang et al., 2022)	ACM MM22	42.0	53.8	59.9	79.5	47.1	63.2	65.1	80.1	80.2
ViPT (Zhu et al., 2023)	CVPR23	52.5	65.1	61.7	83.5	59.2	75.8	72.1	81.5	87.1
SeqTrackv2 (Chen et al., 2023a)	ArXiv23	55.8	70.4	64.7	88.0	61.2	78.2	74.4	81.5	91.0
OneTracker (Hong et al., 2024)	CVPR24	53.8	67.2	64.2	85.7	60.8	76.7	72.7	81.9	87.2
SDSTrack (Hou et al., 2024)	CVPR24	53.1	66.5	62.5	84.8	59.7	76.7	72.8	81.2	88.3
UnTrack (Wu et al., 2024)	CVPR24	51.3	64.6	62.5	84.2	58.9	75.5	71.8	82.0	86.4
CMDTrack (Zhang et al., 2025)	TPAMI25	56.6	68.8	61.8	85.9	61.3	75.8	–	–	–
STTrack (Hu et al., 2025)	AAAI25	60.3	76.0	66.7	89.8	61.9	78.6	**77.6**	82.5	**93.7**
SMSTracker (Chan et al., 2025)	ICCV25	56.0	70.3	64.5	86.9	60.4	76.3	74.8	82.2	89.7
XTrack (Tan et al., 2025)	ICCV25	55.7	69.1	64.9	87.4	60.9	77.5	74.0	82.8	88.9
SUTrack (Chen et al., 2025)	AAAI25	59.9	74.5	69.5	92.2	62.7	79.9	76.5	82.8	91.8
UniSOT (Ma et al., 2026)	TPAMI26	54.0	67.4	64.1	86.6	60.7	78.0	73.2	81.8	88.8
SEATrack (Su et al., 2026)	CVPR26	57.3	71.6	63.9	87.8	60.3	77.1	73.6	82.1	88.4
UETrack (Dong et al., 2026)	CVPR26	55.5	69.1	64.2	86.7	59.2	76.2	68.3	80.8	–
UTPTrack (Wu et al., 2026)	CVPR26	59.2	73.9	70.4	92.9	62.6	80.0	76.4	82.7	92.1
TGTrack (Geng et al., 2026)	CVPR26	61.7	**77.4**	69.9	93.7	63.9	81.4	77.4	82.9	93.0
OneTrackerV2 (Hong et al., 2026)	ICML26	60.6	75.5	69.5	91.3	63.0	79.8	–	–	–
UniMergeTrack	Ours	**61.8**	77.2	**70.6**	**94.0**	**64.1**	**81.5**	76.8	**83.1**	92.3

Table 2: Generalization comparison on COESOT and CDTB datasets.

Dataset	Metric	SMSTracker	XTrack	SUTrack	SEATrack	UETrack	UTPTrack	UniMergeTrack
COESOT (RGB-Event)	SR	63.6	65.2	64.6	64.8	61.5	64.7	**66.3**
	PR	76.8	78.8	77.7	78.9	74.7	78.2	**80.5**
CDTB (RGB-Depth)	F-score	69.7	69.8	70.8	68.5	67.7	71.5	**72.0**
	Precision	70.5	68.5	69.8	69.4	68.2	71.4	**72.0**

• **Generalization Performance on COESOT (Tang et al., 2025) and CDTB (Lukezic et al., 2019).** To assess generalization, we directly evaluate UniMergeTrack and recent open-source trackers on COESOT and CDTB without any additional training, as shown in Table 2. UniMergeTrack achieves 66.3% SR and 80.5% PR on COESOT, outperforming SUTrack by 1.7 and 2.8 percentage points, respectively. On CDTB, UniMergeTrack also achieves the best performance in terms of both F-score and Precision, further demonstrating its robust generalization ability to unseen datasets.

4.3 Ablation Study

• **Component Analysis.** As shown in Table 3, we conduct a component analysis on both LasHeR and VisEvent to evaluate the contribution of each module. Specifically, when only the first-stage task-specific training is performed without model merging, the average AUC and PR decrease by 2.5 and 3.1 percentage points, respectively, demonstrating the benefit of integrating knowledge across RGB-X tasks. In addition, removing distribution alignment fine-tuning causes significant performance degradation, highlighting the importance of alleviating the distribution shift induced by model merging to ensure effective collaboration between the unified backbone and tracking heads.

For the hierarchical merging design, removing high-pass filtering, the lightweight modality expert, or intra-task tracking head merging results in average PR drops of 1.8, 1.5, and 1.4 percentage points, respectively, validating their crucial roles in preserving transferable and task-specific knowledge. Finally, removing optimal transport-guided channel alignment reduces average PR by 0.9 percentage points, demonstrating its benefit for reliable intra-task tracking head merging.

• **Analysis of Different Training Paradigms.** As shown in Table 4, joint training consistently outperforms independent training, confirming the benefit of cross-task knowledge sharing. Compared with joint training, the model merging paradigm further improves PR by 2.7 and 1.6 points on LasHeR and VisEvent, respectively, achieving the best overall performance.

• **Analysis of Different Merging Strategies.** As shown in Table 5, we compare our hierarchical merging strategy with representative model merging methods. Our method achieves the best perfor-

Table 3: Component Analysis of UniMergeTrack on LasHeR and VisEvent. Δ denotes the average performance change relative to the full model.

No.	Variant	LasHeR		VisEvent		Avg. Change	
		AUC	PR	AUC	PR	ΔAUC	ΔPR
1	**UniMergeTrack (Ours)**	**61.8**	**77.2**	**64.1**	**81.5**	–	–
2	w/o Model Merging (Baseline)	58.8	73.4	62.2	79.2	-2.5	-3.1
3	w/o Distribution Alignment Fine-Tuning	57.7	73.2	60.9	78.6	-3.7	-3.5
4	w/o High-pass Filtering	60.1	75.2	62.9	80.0	-1.5	-1.8
5	w/o Lightweight Modality Expert	60.3	75.3	63.0	80.4	-1.3	-1.5
6	w/o Intra-Task Tracking Head Merging	60.4	75.6	63.2	80.3	-1.2	-1.4
7	w/o Optimal Transport Channel Alignment	61.0	76.2	63.5	80.8	-0.7	-0.9

Table 4: Analysis of Different Training Paradigms on LasHeR and VisEvent.

Paradigm	LasHeR		VisEvent	
	AUC	PR	AUC	PR
Independent Training	58.8	73.4	62.2	79.2
Joint Training	59.9	74.5	62.7	79.9
Model Merging (Ours)	**61.8**	**77.2**	**64.1**	**81.5**

Table 5: Analysis of Different Merging Strategies on LasHeR and VisEvent.

Merging Strategy	LasHeR		VisEvent	
	AUC	PR	AUC	PR
Task Arithmetic (Ilharco et al., 2023)	57.6	72.4	60.6	77.7
TIES-Merging (Yadav et al., 2023)	59.2	73.6	62.5	79.2
FREE-Merging (Zheng & Wang, 2025)	60.4	75.6	63.2	80.3
Hierarchical Merging (Ours)	**61.8**	**77.2**	**64.1**	**81.5**

mance on both datasets, outperforming FREE-Merging by 1.6 and 1.2 PR on LasHeR and VisEvent, respectively. These results validate the effectiveness of our hierarchical merging strategy.

4.4 Visualization

In this section, we present qualitative visualizations to provide a more intuitive understanding of our method. Figure 3 (a) presents comparisons of tracking results across RGB-Thermal, RGB-Event, and RGB-Depth tracking tasks. UniMergeTrack exhibits more accurate and stable target localization across these diverse multimodal tracking scenarios. We further visualize the attention maps in Figure 3 (b), which mainly focus on target regions while suppressing irrelevant backgrounds. These visualizations suggest that the proposed model achieves robust target localization in challenging scenarios while effectively capturing target-relevant information.

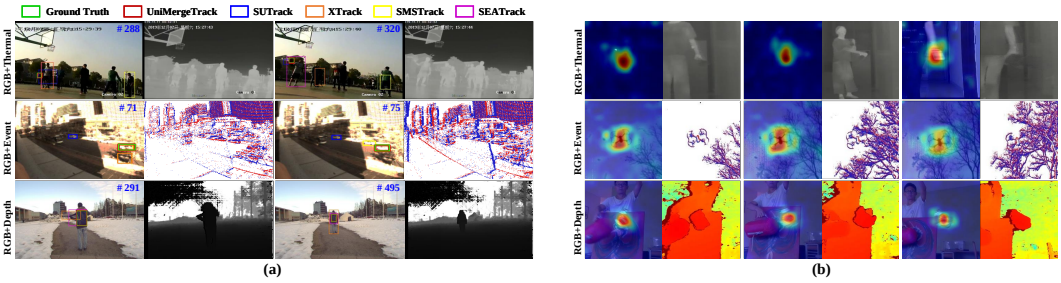


Figure 3: Visualization of (a) tracking results and (b) attention maps produced by UniMergeTrack.

5 Conclusion

In this work, we propose UniMergeTrack, the first model merging paradigm for RGB-X tracking. UniMergeTrack combines task-specific optimization with cross-task knowledge integration through hierarchical model merging, where backbones are merged across tasks and tracking heads are merged within each task. A subsequent distribution alignment stage further establishes compatibility between the unified backbone and task-specific heads. Extensive experiments across RGB-Thermal, RGB-Event, and RGB-Depth tracking demonstrate superior performance and strong generalization to unseen data. Our results highlight model merging as a promising paradigm for building unified RGB-X trackers while mitigating knowledge isolation and optimization interference.

LLM USAGE

We used LLMs to improve the grammar, phrasing, and readability of this manuscript. We also used LLMs as a search tool to help us find related works, but we independently reviewed, interpreted, and summarized all identified works.

REFERENCES

Pankaj Agarwal, Sharath Raghvendra, Pouyan Shirzadian, and Keegan Yao. Efficient algorithms for robust and partial semi-discrete optimal transport. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the monge–kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000.

Sixian Chan, Zedong Li, Wenhao Li, Shijian Lu, Chunhua Shen, and Xiaoqin Zhang. SMSTracker: Tri-path score mask sigma fusion for multi-modal tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4766–4775, 2025.

Xin Chen, Ben Kang, Jiawen Zhu, Dong Wang, Houwen Peng, and Huchuan Lu. Unified sequence-to-sequence learning for single- and multi-modal visual object tracking. *arXiv preprint arXiv:2304.14394*, 2023a.

Xin Chen, Houwen Peng, Dong Wang, Huchuan Lu, and Han Hu. Seqtrack: Sequence to sequence learning for visual object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14572–14581, 2023b.

Xin Chen, Ben Kang, Wanting Geng, Jiawen Zhu, Yi Liu, Dong Wang, and Huchuan Lu. SUTrack: Towards simple and unified single object tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 2239–2247, 2025.

Xin Chen, Chuanyu Sun, Jiao Xu, Houwen Peng, Dong Wang, Huchuan Lu, and Kede Ma. RELO: Reinforcement learning to localize for visual object tracking. In *International Conference on Machine Learning*, 2026.

Yutao Cui, Cheng Jiang, Limin Wang, and Gangshan Wu. Mixformer: End-to-end tracking with iterative mixed attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13608–13618, 2022.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

Hao Dong, Yujin Liu, Haoyue Liu, Zhenyu Wang, Shihan Peng, Zhiwei Shi, Yi Chang, and Luxin Yan. Tracking through severe occlusion via event-derived transient cues. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29526–29536, June 2026.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5374–5383, 2019.

Aude Genevay, Marco Cuturi, Gabriel Peyré, and Francis Bach. Stochastic optimization for large-scale optimal transport. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Wanting Geng, Xin Chen, Chuanyu Sun, Jie Zhao, Ben Kang, Dong Wang, and Huchuan Lu. Tg-track: Temporal generative learning for unified single object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28134–28144, 2026.

Chengyue Gong, Dilin Wang, Meng Li, Xinlei Chen, Zhicheng Yan, Yuandong Tian, Qiang Liu, and Vikas Chandra. NASViT: Neural architecture search for efficient vision transformers with gradient conflict aware supernet training. In *International Conference on Learning Representations*, 2022.

Lingyi Hong, Shilin Yan, Renrui Zhang, Wanyun Li, Xinyu Zhou, Pinxue Guo, Kaixun Jiang, Yiting Chen, Jinglun Li, Zhaoyu Chen, and Wenqiang Zhang. OneTracker: Unifying visual object tracking with foundation models and efficient tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19079–19091, 2024.

Lingyi Hong, Jinglun Li, Xinyu Zhou, Kaixun Jiang, Pinxue Guo, Zhaoyu Chen, Runze Li, Xingdong Sheng, and Wenqiang Zhang. Unified multimodal visual tracking with dual mixture-of-experts. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306. PMLR, 2026.

Xiaojun Hou, Jiazheng Xing, Yijie Qian, Yaowei Guo, Shuo Xin, Junhao Chen, Kai Tang, Mengmeng Wang, Zhengkai Jiang, Liang Liu, and Yong Liu. SDSTrack: Self-distillation symmetric adapter learning for multi-modal visual object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26551–26561, 2024.

Xiantao Hu, Ying Tai, Xu Zhao, Chen Zhao, Zhenyu Zhang, Jun Li, Bineng Zhong, and Jian Yang. Exploiting multimodal spatial-temporal patterns for video object tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 3581–3589, 2025.

Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(5):1562–1577, 2021.

Zhehao Huang, Yuhang Liu, Baijiong Lin, Yixin Lou, Zhengbao He, Hanling Tian, Tao Li, and Xiaolin Huang. RAIN-Merging: A gradient-free method to enhance instruction following in large reasoning models with preserved thinking format. In *International Conference on Learning Representations*, 2026.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023.

Jiin Im, Sisung Liu, and Je Hyeong Hong. Shape-of-you: Fused gromov-wasserstein optimal transport for semantic correspondence in-the-wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27730–27739, 2026.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pp. 876–885, 2018.

Leonid V. Kantorovich. On the translocation of masses. *Doklady Akademii Nauk SSSR*, 37(7–8): 199–201, 1942.

Ugur Kart, Alan Lukezic, Matej Kristan, Joni-Kristian Kamarainen, and Jiri Matas. Object tracking by reconstruction with view-specific discriminative correlation filters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1339–1348, 2019.

Chenglong Li, Wanlin Xue, Yaqing Jia, Zhichen Qu, Bin Luo, and Jin Tang. Lasher: A large-scale high-diversity benchmark for rgbt tracking. *IEEE Transactions on Image Processing*, 31: 392–404, 2022.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pp. 740–755, 2014.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Alan Lukezic, Ugur Kart, Jani Kapyla, Ahmed Durmush, Joni-Kristian Kamarainen, Jiri Matas, and Matej Kristan. Cdtb: A color and depth visual object tracking dataset and benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10013–10022, October 2019.

Yinchao Ma, Yuyang Tang, Wenfei Yang, Tianzhu Zhang, Xu Zhou, and Feng Wu. UniSOT: A unified framework for multi-modality single object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(2):1554–1572, 2026.

Michael S. Matena and Colin A. Raffel. Merging models with fisher-weighted averaging. In *Advances in Neural Information Processing Systems*, volume 35, pp. 17703–17716, 2022.

Facundo Mémoli. Gromov–wasserstein distances and the metric approach to object matching. *Foundations of Computational Mathematics*, 11(4):417–487, 2011.

Matthias Muller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *Proceedings of the European Conference on Computer Vision*, pp. 300–317, 2018.

Sloan Nietert and Ziv Goldfeld. Estimation of stochastic optimal transport maps. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Liang Peng, Junyuan Gao, Xinran Liu, Weihong Li, Shaohua Dong, Zhipeng Zhang, Heng Fan, and Libo Zhang. Vasttrack: Vast category visual object tracking. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Gabriel Peyré and Marco Cuturi. Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5–6):355–607, 2019.

Yanlin Qian, Song Yan, Alan Lukezic, Matej Kristan, Joni-Kristian Kamarainen, and Jiri Matas. DAL: A deep depth-aware long-term tracker. In *Proceedings of the International Conference on Pattern Recognition*, pp. 7825–7832, 2020.

Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

Junbin Su, Ziteng Xue, Shihui Zhang, Kun Chen, Weiming Hu, and Zhipeng Zhang. SEATrack: Simple, efficient, and adaptive multimodal tracker. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28679–28689, 2026.

Yuhao Sun, Zhenyi Zhang, Zihan Wang, Tiejun Li, and Peijie Zhou. Variational regularized unbalanced optimal transport: Single network, least action. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Yuedong Tan, Zongwei Wu, Yuqian Fu, Zhuyun Zhou, Guolei Sun, Eduard Zamfir, Chao Ma, Danda Paudel, Luc Van Gool, and Radu Timofte. XTrack: Multimodal training boosts RGB-X video object trackers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5734–5744, 2025.

Chuanming Tang, Xiao Wang, Ju Huang, Bo Jiang, Lin Zhu, Shifeng Chen, Jianlin Zhang, Yaowei Wang, and Yonghong Tian. Revisiting color-event based tracking: A unified network, dataset, and metric. *Pattern Recognition*, pp. 112718, 2025.

Chaoqun Wang, Chunyan Xu, Zhen Cui, Ling Zhou, Tong Zhang, Xiaoya Zhang, and Jian Yang. Cross-modal pattern-propagation for RGB-T tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7064–7073, 2020.

Xiao Wang, Xiujun Shu, Zhipeng Zhang, Bo Jiang, Yaowei Wang, Yonghong Tian, and Feng Wu. Towards more flexible and accurate object tracking with natural language: Algorithms and benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13763–13773, 2021.

Xiao Wang, Jianing Li, Lin Zhu, Zhipeng Zhang, Zhe Chen, Xin Li, Yaowei Wang, Yonghong Tian, and Feng Wu. Visevent: Reliable object tracking via collaboration of frame and event flows. *IEEE Transactions on Cybernetics*, 54(3):1997–2010, 2024. doi: 10.1109/TCYB.2023.3318601.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 23965–23998, 2022.

Hao Wu, Xudong Wang, Jialiang Zhang, Junlong Tong, Xinghao Chen, Junyan Lin, Yunpu Ma, and Xiaoyu Shen. Utptrack: Towards simple and unified token pruning for visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20963–20972, June 2026.

Zongwei Wu, Jilai Zheng, Xiangxuan Ren, Florin-Alexandru Vasluianu, Chao Ma, Danda Pani Paudel, Luc Van Gool, and Radu Timofte. Single-model and any-modality for video object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19156–19166, 2024.

Yun Xiao, Mengmeng Yang, Chenglong Li, Lei Liu, and Jin Tang. Attribute-based progressive fusion network for RGBT tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 2831–2838, 2022.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A. Raffel, and Mohit Bansal. TIES-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 7093–7115, 2023.

Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. Learning spatio-temporal transformer for visual tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10448–10457, 2021a.

Song Yan, Jinyu Yang, Jani Käpylä, Feng Zheng, Aleš Leonardis, and Joni-Kristian Kämäräinen. Depthtrack: Unveiling the power of rgbd tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10725–10733, 2021b.

Jinyu Yang, Zhe Li, Feng Zheng, Aleš Leonardis, and Jingkuan Song. Prompting for multi-modal tracking. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 3492–3500, 2022.

Botao Ye, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Joint feature learning and relation modeling for tracking: A one-stream framework. In *European Conference on Computer Vision*, pp. 341–357. Springer, 2022.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 57755–57775, 2024.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5824–5836, 2020.

Jiqing Zhang, Xin Yang, Yingkai Fu, Xiaopeng Wei, Baocai Yin, and Bo Dong. Object tracking by jointly exploiting frame and event domain. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13043–13052, 2021a.

Pengyu Zhang, Jie Zhao, Chunjuan Bo, Dong Wang, Huchuan Lu, and Xiaoyun Yang. Jointly modeling motion and appearance cues for robust RGB-T tracking. *IEEE Transactions on Image Processing*, 30:3335–3347, 2021b.

Tianlu Zhang, Qiang Zhang, Kurt Debattista, and Jungong Han. Cross-modality distillation for multi-modal tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(7): 5847–5865, 2025.

Xiaosong Zhang, Yunjie Tian, Lingxi Xie, Wei Huang, Qi Dai, Qixiang Ye, and Qi Tian. Hivit: A simpler and more efficient design of hierarchical vision transformer. In *International Conference on Learning Representations*, 2023.

Shenghe Zheng and Hongzhi Wang. Free-merging: Fourier transform for efficient model merging. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3863–3873. IEEE, 2025.

Jiawen Zhu, Simiao Lai, Xin Chen, Dong Wang, and Huchuan Lu. Visual prompt multi-modal tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9516–9526, 2023.

APPENDIX

A GRADIENT CONFLICT RATE CALCULATION

Multi-task learning facilitates knowledge sharing across tasks, but joint optimization can also introduce interference due to competing task objectives. To quantify such interference caused by joint training in RGB-X tracking, we measure pairwise gradient conflicts using the Gradient Conflict Rate (GCR), defined based on the cosine similarity between task-specific gradients (Yu et al., 2020; Gong et al., 2022).

Given a task i, we compute its gradient with respect to the trainable parameters θ of the shared encoder as

$$g_i = \nabla_\theta \mathcal{L}_i, \tag{17}$$

where $\mathcal{L}_i$ denotes the training loss for task i. For a pair of tasks i and j, the cosine similarity between their gradients is defined as

$$C_{ij} = \frac{g_i^\top g_j}{\|g_i\|_2 \|g_j\|_2}. \tag{18}$$

We consider the two tasks to exhibit a gradient conflict when $C_{ij} < 0$, indicating opposing optimization directions on the shared encoder.

We evaluate gradient conflicts every 20 epochs throughout joint training. At each evaluation checkpoint, we construct $M = 1000$ batch triplets, each consisting of independently sampled batches from the RGB-T, RGB-D, and RGB-E training sets. For the m-th triplet, we separately compute the encoder gradients induced by the three tasks using the same model checkpoint, without updating the model parameters. We then obtain the following three pairwise cosine similarities:

$$C_{TD}^{(m)}, \qquad C_{TE}^{(m)}, \qquad C_{DE}^{(m)}. \tag{19}$$

For each task pair (i, j), the Gradient Conflict Rate is defined as the proportion of sampled batch pairs that yield a negative gradient cosine similarity:

$$\mathrm{GCR}_{ij} = \frac{1}{M} \sum_{m=1}^{M} \mathbb{I}\left(C_{ij}^{(m)} < 0\right) \times 100\%, \tag{20}$$

where $\mathbb{I}(\cdot)$ denotes the indicator function and $M = 1000$ in our experiments. We report the GCR separately for the RGBT-RGBD, RGBT-RGBE, and RGBD-RGBE task pairs. Their average is computed as

$$\mathrm{GCR}_{\mathrm{mean}} = \frac{1}{3}\left(\mathrm{GCR}_{TD} + \mathrm{GCR}_{TE} + \mathrm{GCR}_{DE}\right). \tag{21}$$

A higher GCR indicates more frequent conflicts between task-specific optimization directions on the shared encoder, suggesting stronger cross-task interference during joint training.

B RELATED WORK

B.1 OPTIMAL TRANSPORT

Optimal transport (OT) provides a principled framework for comparing probability distributions through minimum-cost transport. Kantorovich (Kantorovich, 1942) relaxed the original formulation into a tractable linear program, laying the foundation of modern OT, while Benamou and

Brenier (Benamou & Brenier, 2000) introduced a dynamic formulation that interprets transport as continuous mass flow. A major computational advance came from entropy-regularized OT (Cuturi, 2013), which enables efficient optimization via the Sinkhorn algorithm (Sinkhorn & Knopp, 1967). Subsequent work improved scalability through stochastic optimization (Genevay et al., 2016) and extended OT to structured spaces via Gromov-Wasserstein distances (Mémoli, 2011). More recent studies further improved robustness and scalability through partial and unbalanced formulations (Agarwal et al., 2025; Sun et al., 2025), as well as stochastic and structure-aware transport (Nietert & Goldfeld, 2025; Im et al., 2026). These advances have established OT as a versatile tool for distribution alignment in modern machine learning (Peyré & Cuturi, 2019). During the training of trackers, the channels of the tracking head may drift in parameter space, leading to inconsistent channel correspondences across historical checkpoints. We resort to OT for channel alignment before merging, ensuring more reliable parameter merging.

C Implementation Details of OT-Based Tracking Head Merging

For each task, we use the tracking heads saved at epochs 24 and 30 as the historical head $\boldsymbol{h}_k^{\mathrm{his}}$ and the reference head $\boldsymbol{h}_k^{\mathrm{ref}}$, respectively. The reference head defines the target channel basis. OT alignment is applied independently to the three branches of the center-based head:

$$\mathcal{B} = \{\mathrm{ctr}, \mathrm{offset}, \mathrm{size}\}, \tag{22}$$

where each branch contains five convolutional layers. The following operations are applied independently to each branch.

For layer $l > 1$, the historical input channels are first expressed in the reference basis using the transport plan obtained from the preceding layer:

$$\overline{\mathbf{W}}_l^{\mathrm{his}} = \mathbf{W}_l^{\mathrm{his}} \times_{\mathrm{in}} \overline{\mathbf{P}}_{l-1}, \tag{23}$$

where $\mathbf{W}_l^{\mathrm{his}}$ and $\overline{\mathbf{W}}_l^{\mathrm{his}}$ are the historical weights before and after input-channel alignment, respectively, and $\overline{\mathbf{P}}_{l-1}$ is the normalized transport plan. For the first layer, we set

$$\overline{\mathbf{W}}_1^{\mathrm{his}} = \mathbf{W}_1^{\mathrm{his}}. \tag{24}$$

Each output channel is represented by its normalized incoming weights, augmented with the standardized convolutional bias and BatchNorm statistics when available:

$$\mathbf{d}_{l,i}^{\mathrm{his}} = \mathrm{Norm}_2\left(\left[\mathrm{Norm}_2\left(\mathrm{vec}(\overline{\mathbf{W}}_{l,i}^{\mathrm{his}})\right); \beta \widehat{\mathbf{s}}_{l,i}^{\mathrm{his}}\right]\right), \tag{25}$$

$$\mathbf{d}_{l,j}^{\mathrm{ref}} = \mathrm{Norm}_2\left(\left[\mathrm{Norm}_2\left(\mathrm{vec}(\mathbf{W}_{l,j}^{\mathrm{ref}})\right); \beta \widehat{\mathbf{s}}_{l,j}^{\mathrm{ref}}\right]\right). \tag{26}$$

where $\mathbf{W}_{l,i}$ is the kernel of the i-th output channel in layer l, $\mathrm{vec}(\cdot)$ flattens the kernel, and $\mathrm{Norm}_2(\cdot)$ denotes ℓ_2 normalization. The vector $\widehat{\mathbf{s}}_{l,i}$ contains the corresponding standardized bias and BatchNorm statistics, while $[\cdot\,;\cdot]$ denotes vector concatenation. The coefficient β controls the contribution of these auxiliary statistics, and we set $\beta = 0.1$.

The pairwise alignment cost between the historical channel i and the reference channel j is defined as

$$C_l^{ij} = \left\|\mathbf{d}_{l,i}^{\mathrm{his}} - \mathbf{d}_{l,j}^{\mathrm{ref}}\right\|_2^2. \tag{27}$$

We obtain the channel correspondence by solving the entropy-regularized optimal transport problem:

$$\mathbf{P}_l^* = \arg\min_{\mathbf{P}\geq 0} \langle \mathbf{P}, \mathbf{C}_l \rangle_F - \epsilon \mathcal{H}(\mathbf{P}), \tag{28}$$

subject to

$$\mathbf{P}\mathbf{1} = \frac{1}{B_s}\mathbf{1}, \qquad \mathbf{P}^\top \mathbf{1} = \frac{1}{B_t}\mathbf{1}, \tag{29}$$

where B_s and B_t denote the numbers of historical and reference channels, respectively. We use Sinkhorn iterations (Cuturi, 2013) with $\epsilon = 0.05$, at most 300 iterations, and a stopping tolerance of

10^{-6}. Because the two heads originate from the same task and training trajectory, an identity prior of 0.02 is added to the off-diagonal transport costs to discourage unnecessary channel permutations.

After column normalization, we denote the normalized transport plan by $\overline{\mathbf{P}}_l$, which maps the historical output channels to the reference basis:

$$\widetilde{\mathbf{W}}_l^{\text{his}} = \overline{\mathbf{P}}_l^{\top} \overline{\mathbf{W}}_l^{\text{his}}. \tag{30}$$

The same mapping is applied to the corresponding bias and BatchNorm parameters. The resulting transport plan $\overline{\mathbf{P}}_l$ is then propagated to the input channels of layer $l+1$.

OT-based output-channel alignment is applied to the first four layers. For the fifth layer, only the input channels are aligned because its output channels have fixed semantics: objectness for the `ctr` branch, $(\Delta x, \Delta y)$ for the `offset` branch, and (w, h) for the `size` branch.

After alignment, the historical and reference heads are merged by weighted averaging:

$$\boldsymbol{H}_k = \alpha \boldsymbol{h}_k^{\text{ref}} + (1-\alpha)\widetilde{\boldsymbol{h}}_k^{\text{his}}. \tag{31}$$

We set α to 0.5, 0.7, and 0.5 for RGB-Thermal, RGB-Event, and RGB-Depth tracking, respectively.

D More Experiments and Analysis

D.1 Datasets and Evaluation Metric

RGB-Thermal Tracking. For RGB-Thermal tracking, we adopt LasHeR and RGBT234. LasHeR is a large-scale RGB-Thermal tracking benchmark containing 1,224 aligned RGB-Thermal video pairs with diverse object categories and environmental conditions. RGBT234 consists of 234 RGB-Thermal sequences, which is widely used for evaluating the robustness of RGBT trackers. Following the standard evaluation protocols, we report AUC (Area Under the Success Curve), PR (Precision Rate), MSR (Mean Success Rate), and MPR (Mean Precision Rate). Specifically, AUC measures the area under the success plot and reflects the overall tracking performance across different overlap thresholds. PR evaluates localization accuracy by measuring the proportion of frames with accurate target center predictions. MSR and MPR denote the average success rate and precision rate over all sequences. We train UniMergeTrack on LasHeR and evaluate it on both LasHeR and RGBT234.

RGB-Event Tracking. For RGB-Event tracking, we use VisEvent and COESOT. VisEvent contains 820 RGB-Event video pairs collected under challenging conditions, such as low illumination, fast motion, and background clutter. COESOT is a large-scale RGB-Event tracking benchmark containing diverse scenarios. We follow the standard protocols and report AUC and PR for evaluation. UniMergeTrack is trained on VisEvent and evaluated on both VisEvent and COESOT to assess its generalization ability across RGB-Event datasets.

RGB-Depth Tracking. For RGB-Depth tracking, we adopt DepthTrack, VOT-RGBD22, and CDTB. DepthTrack is a large-scale RGB-Depth tracking benchmark used for training RGB-Depth trackers, while VOT-RGBD22 and CDTB provide challenging evaluation scenarios with diverse RGB-Depth sequences. VOT-RGBD22 follows the VOT evaluation protocol and reports EAO (Expected Average Overlap), Accuracy (Acc.), and Robustness (Rob.), which measure overall tracking performance, localization accuracy, and tracking stability, respectively. CDTB evaluates RGB-Depth trackers using Precision and F-score, where Precision reflects localization accuracy and F-score provides a balanced evaluation by combining Precision and Recall. We train UniMergeTrack on DepthTrack and evaluate it on VOT-RGBD22 and CDTB. We do not report results on the DepthTrack test set because its evaluation toolkit has been updated and the original version is no longer available, preventing the reproduction of existing methods and making fair comparisons infeasible.

D.2 Supplementary Experiments

• **Analysis of Hierarchical Merging Strategy.** As shown in Table 6, cross-task backbone merging significantly improves the baseline by transferring shared knowledge across tasks. In contrast, intra-task head merging outperforms cross-task head merging, achieving the best results on both datasets. These results support our design of cross-task backbone merging and intra-task head merging.

Table 6: Analysis of Hierarchical Merging Strategy on LasHeR and VisEvent.

No.	Merging Strategy		LasHeR		VisEvent	
	Backbone	Tracking Head	AUC	PR	AUC	PR
1	–	–	58.8	73.4	62.2	79.2
2	Cross-task	–	60.4	75.6	63.2	80.3
3	Cross-task	Cross-task	60.7	76.0	63.4	80.6
4	**Cross-task**	**Intra-task**	**61.8**	**77.2**	**64.1**	**81.5**

Table 7: Ablation Study of Different Channel Alignment Strategies.

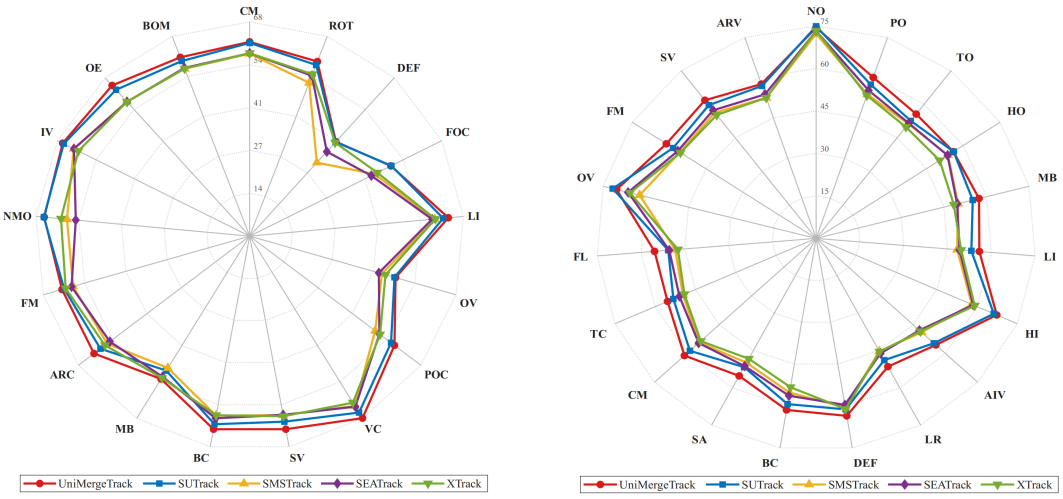
Merging Strategy	LasHeR		VisEvent	
	AUC	PR	AUC	PR
w/o Alignment	61.0	76.2	63.5	80.8
Hungarian Alignment	61.2	76.5	63.5	81.0
OT Alignment (Ours)	**61.8**	**77.2**	**64.1**	**81.5**

Table 8: Hyperparameter settings for different tasks.

Hyperparameter	RGB-Thermal	RGB-Event	RGB-Depth
Expert sparsity (d)	0.01	0.02	0.02
Reference head contribution (α)	0.50	0.70	0.50

• **Ablation Study of Different Channel Alignment Strategies.** Table 7 compares different channel alignment strategies for intra-task tracking head merging. Direct parameter averaging suffers from hidden-channel inconsistency, while Hungarian alignment is limited by hard matching. OT-based alignment achieves the best performance on both datasets.

• **Attribute Evaluation on the VisEvent and LasHeR Datasets.** As shown in Fig. 4, we present the attribute evaluation on VisEvent and LasHeR. UniMergeTrack achieves consistently competitive performance across diverse challenging attributes, including Scale Variation, Viewpoint Change, Low Illumination, and Background Clutter. These improvements demonstrate that our model merging strategy preserves task-specific knowledge while promoting cross-task integration, enabling robust generalization across diverse tracking conditions.



(a) Attribute evaluation of AUC on VisEvent dataset

(b) Attribute evaluation of AUC on LasHeR dataset

Figure 4: Attribute evaluation on the VisEvent and LasHeR test sets.

D.3 Hyperparameter Settings

As shown in Table 8, we use task-specific hyperparameters for different RGB-X tracking scenarios. The expert sparsity d is set to 0.01/0.02/0.02, and the reference head contribution coefficient α is set to 0.5/0.7/0.5 for RGB-Thermal, RGB-Event, and RGB-Depth tracking, respectively.

D.4 Efficiency Analysis

UniMergeTrack operates entirely in the parameter space without altering the network architecture, thereby preserving the original model complexity and inference efficiency at 89M parameters, 23

GFLOPs, and 55 FPS. In terms of storage, it shares the merged backbone across tasks while retaining lightweight task-specific components, reducing the storage requirement from 3×340 MB for independently trained models to 379 MB, only 39 MB more than joint training (340 MB).

E LIMITATION ANALYSIS

Although UniMergeTrack achieves effective multimodal tracking, there also remains room for further improvement. First, the current framework does not incorporate the text modality. Text-guided multimodal tracking, potentially integrated with multimodal large language models, represents a promising direction for enhancing the model's target perception. Second, the framework does not explicitly account for missing or degraded modalities in challenging scenarios. Developing quality-aware and adaptive multimodal fusion mechanisms could further improve tracking robustness to incomplete or unreliable multimodal inputs.