UETrack: A Unified and Efficient Framework for Single Object Tracking —— Supplementary Material ——

In this appendix, we provide additional content to complement the main manuscript:

- More details about TAD.
- More implementation details.
- Introduction of benchmarks.
- Additional state-of-the-art comparisons.
- Additional ablation studies.
- Additional visualization results.
- Limitations.

1. More Details about TAD

In this section, we provide the pseudocode of the TAD processing pipeline, as shown in Algorithm 1. The overall procedure is as follows: First, the student network and the teacher network perform forward inference separately to obtain the prediction results and corresponding feature maps. Then, the feature maps are fed into the Adaptive Net to predict a one-hot vector. Based on this one-hot vector, a proxy prediction is carried out. After that, we compute the loss functions for both the student network and the Adaptive Net according to the proxy prediction. Finally, the parameters of the student network and the Adaptive Net are updated using their respective losses. Here, B, L, C_t, C_s, H, and W denote the batch size, sequence length, number of channels in the teacher's features, number of channels in the student's features, height, and width of the search region, respectively.

Algorithm 1 TAD Process

T: the teacher model
S: the student model
/* teacher forward */
T_{pre}, T_{feat} = T(search,template)
/* student forward */
S_{pre}, S_{feat} = S(search,template)
/* Adaptive Net forward */
T_{feat}.shape:[B,L,Ct], S_{feat}.shape:[B,L,Cs]
T_{feat} = Reshape(T_{feat}) # shape:[B,Ct,H,W]
S_{feat} = Reshape(S_{feat}) # shape:[B,Cs,H,W]
T_{avg} = Avg_Pool2d(T_{feat}) # shape:[B,Ct]
S_{avg} = Avg_Pool2d(S_{feat}) # shape:[B,Cs]
F_{cat} = torch.cat([T_{avg},S_{avg}]) # shape:[B,Cs+Ct]
Logit = MLP(F_{cat}) # shape:[B,2]
Logit = Gumbel_Softmax(Logit) # one_hot vector
/* surrogate policy */
O_{dis} = Logit[:,0] # shape:[B]
A_{pre} = (1-O_{dis}) * S_{pre} + O_{dis} * T_{pre}
/* compute student loss*/
feature alignment
S_{feat} = Linear(S_{feat}) # shape:[B,L,Ct]
GT: ground truth
$L_s = L_{cls}(S_{pre}, GT) + \lambda_g L_{giou}(S_{pre}, GT) + \lambda_{l1} L_{l1}(S_{pre}, GT) + L_{task}(S_{pre}, GT)$
L_s += O_{dis} * ($\lambda_{kd} L_{kd}(S_{pre}, T_{pre}) + \lambda_f L_f(S_{pre}, T_{pre})$)
/* compute Adaptive Net loss*/
$L_a = L_{cls}(A_{pre}, GT) + \lambda_g L_{giou}(A_{pre}, GT) + \lambda_{l1} L_{l1}(A_{pre}, GT) + L_{task}(A_{pre}, GT)$
/* back-propagate*/
student update
L_s.backward()
Update(S) # AdamW
Adaptive Net update
L_a.backward()
Update(Adaptive Net) # AdamW

2. More Implementation Details

In this section, we present additional training details of UETrack that were omitted from the main manuscript due to space limitations. We construct the training dataset by mixing data from five modalities. The included datasets are: COCO [28], LaSOT [13], GOT-10k [17], TrackingNet [31], VASTTrack [32], DepthTrack [42], VisEvent [36], LasHeR [25], OTB99 [26], and TNL2K [35]. The mixing ratio among these datasets is $[4:4:4:4:4:2:2:2:1:3]$. For video datasets, we sample inputs from a randomly selected video sequence. For the image dataset COCO, we randomly choose an image and apply data augmentation techniques to generate input samples.

During model initialization, we use Fast-iTPN-T [33] to initialize the backbone weights of UETrack. Fast-iTPN-T takes a three-channel RGB image as input, and its first

downsampling convolution layer in the patch embedding module has weights denoted as $\mathbf{W}_{\mathrm{p}} \in \mathbb{R}^{D\times 3\times P\times P}$. In contrast, UETrack takes six-channel multi-modal composite data as input, and the corresponding convolution weights are $\mathbf{W} \in \mathbb{R}^{D\times 6\times P\times P}$, where $D = 96$ and $P = 4$. Since $\mathbf{W}\mathrm{p}$ and $\mathbf{W}$ have different channel dimensions, directly loading the pre-trained weights would cause a mismatch. To address this, we duplicate $\mathbf{W}\mathrm{p}$ along the channel dimension to obtain $\mathbf{W}\mathrm{p}' \in \mathbb{R}^{D\times 6\times P\times P}$. To keep the value range consistent with the original weights, we divide $\mathbf{W}\mathrm{p}'$ by 2 before loading it into the model for multi-modal input. Besides the patch embedding module, there are also mismatches in the Transformer blocks where TP-MoE is inserted. To align the parameters, we replicate the pre-trained weights of the feed-forward network (FFN) n times, where n is the number of experts, and assign one copy to each expert accordingly.

In addition, we adopt the task-recognition auxiliary training strategy from SUTrack [10] to enhance the model's ability to perceive the input modality. Specifically, we first apply global average pooling to the features extracted by the backbone network to obtain a mean vector V_a. This vector is then fed into a three-layer perceptron to classify the input modality into one of five types: RGB, Depth, Thermal, Event, and Language. The loss for this auxiliary task is computed using the cross-entropy between the predicted results and the ground-truth labels. It is important to note that this auxiliary strategy is only used during training. Its purpose is to help the model better recognize different modalities, thereby improving its adaptability and representation capability across modalities.

3. Introduction of Benchmarks

In this section, we give a detailed overview of the benchmarks used for evaluation.

3.1. RGB-based Tracking Benchmarks

LaSOT. LaSOT [13] is a large-scale benchmark for single-object tracking. It contains 1,550 videos covering 85 object categories. Its test set includes 280 videos, each with an average length of 2,448 frames. The evaluation metrics include Success (AUC) and Precision (P and $\mathrm{P}_{\mathrm{Norms}}$), with AUC serving as the primary metric.

LaSOT$_{ext}$. As an extension of the LaSOT dataset, LaSOT$_{ext}$ [14] contains 150 video sequences covering 15 new object categories. Each video has an average length of approximately 1,000 frames. The evaluation metrics are consistent with those used for LaSOT.

TrackingNet. TrackingNet [31] is a large-scale single-object visual tracking dataset that includes various real-world scenarios. It contains 30,643 video clips, with 30,132 used for training and 511 for testing. The tracking results of UETrack are evaluated by submitting them to the official online server, which provides Success (AUC) and Precision (P and P_{Norm}) metrics.

GOT-10k. GOT-10k [17] is a large-scale and highly diverse generic object tracking benchmark, containing over 10,000 video clips covering 563 object categories. Its test set includes 180 video sequences. We submit our tracking results to the official evaluation server. The evaluation metrics include Average Overlap (AO) and Success Rates ($\mathrm{SR}_{0.5}$ and $\mathrm{SR}_{0.75}$).

VOT2021 real-time experiment. The VOT2021 real-time [22] experiment is a sub-challenge of VOT2021 that focuses on real-time short-term tracking. It sends images to the tracker at a rate of 20 frames per second. If the tracker fails to respond in time, the last reported bounding box is assumed to be the output for the available frames. The primary evaluation metric is the expected average overlap (EAO), which measures both the accuracy and robustness of the tracker simultaneously.

NFS. The NFS [21] dataset is the first high-frame-rate benchmark for visual object tracking. It contains 100 video clips covering various real-world scenarios, mainly focusing on fast-moving objects. The primary evaluation metric is the Success (AUC) score.

UAV123. UAV123 [30] is a dataset specifically designed for UAV visual object tracking research, comprising 123 low-altitude aerial videos. The main evaluation metric used is the Success (AUC) score.

3.2. RGB-Depth Tracking Benchmarks.

DepthTrack. DepthTrack [42] is an extensive benchmark for long-term RGB-Depth tracking, containing 50 test videos annotated with 15 per-frame attributes. The key evaluation metric is the F-score, a standard measure in long-term tracking research.

VOT-RGBD22. VOT-RGBD2022 [23] is a sub-challenge of VOT2022 focusing on short-term object tracking using RGB and depth (RGB-D) images. It contains 127 RGB-Depth video sequences. The primary performance metric is expected average overlap (EAO).

3.3. RGB-Thermal Tracking Benchmarks

LasHeR. LasHeR [25] is a large and diverse dataset designed for RGB-Thermal tracking. The test set comprises 245 sequences, and the evaluation is based on AUC and Precision (P) scores.

RGBT234. The RGBT234 [24] dataset is a benchmark for RGB-Thermal (RGB-T) object tracking. It contains 234 pairs of precisely aligned RGB and thermal infrared video sequences, totaling approximately 234,000 frames. The longest video pair can reach up to 8,000 frames in length. The evaluation metrics include Maximum Success Rate (MSR) and Maximum Precision Rate (MPR) scores.

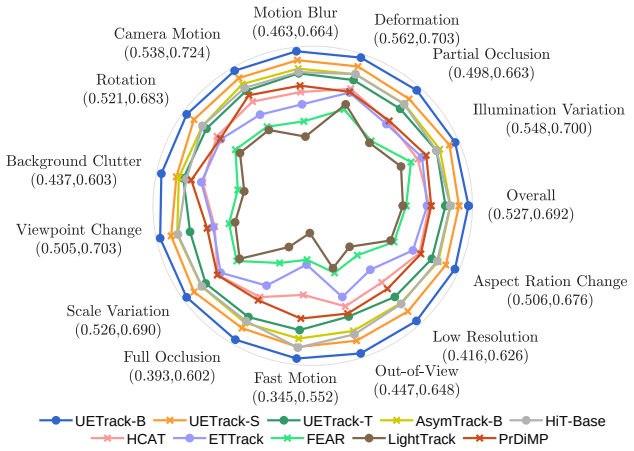

Figure 1. AUC scores of different attributes on LaSOT.

3.4. RGB-Event Tracking Benchmarks

VisEvent. VisEvent [36] is the first large-scale benchmark for object tracking that fuses RGB and event camera data. It contains a total of 820 video pairs, with 500 pairs used for training and 320 pairs for testing. The evaluation metrics include AUC and Precision (P) scores.

3.5. RGB-Language Tracking benchmarks.

TNL2K. TNL2K [35] is a large benchmark for RGB-language tracking, featuring 700 test videos with associated language annotations and target bounding boxes. The tracking performance is assessed using AUC and Precision (P) scores.

OTB99. OTB99 [26] is a small-scale benchmark for RGB-language tracking, created by adding language annotations to the OTB100 [38] dataset. The evaluation metrics include AUC and Precision (P) scores.

4. Additional State-of-the-Art Comparisons

In this section, we provide additional state-of-the-art (SOTA) comparisons, including attribute-based analysis on LaSOT, evaluations on additional datasets and complete comparisons on depth modality.

4.1. Attribute-based Comparisons on LaSOT

Figure 1 presents the attribute-based performance comparison between UETrack and SOTA real-time trackers on the LaSOT dataset. It can be observed that both UETrack-B and UETrack-S consistently achieve top-two performance across all attributes. Notably, UETrack shows outstanding results on three representative and challenging attributes: Background Clutter, Full Occlusion, and Rotation. These attributes typically require stronger appearance modeling capabilities. The strong performance of UETrack under these conditions highlights its effective feature modeling. Even in complex or extreme tracking scenarios, UETrack maintains stable and robust performance, demonstrating its practicality and reliability in real-world applications.

Table 1. SOTA comparisons on NFS and UAV123 benchmarks in AUC score.

	Method	NFS	UAV123
Real-time	UETrack-B (Ours)	**68.4**	**70.2**
	UETrack-S (Ours)	66.6	69.5
	UETrack-T (Ours)	63.9	67.6
	AsymTrack-B [48]	64.4	66.5
	DyHiT [20]	61.6	64.9
	HiT-Base [18]	63.6	65.6
	MixFormerV2-S [11]	-	65.8
	FEAR [4]	61.4	-
	HCAT [7]	63.5	62.7
	E.T.Track [3]	59.0	62.3
	LightTrack [41]	55.3	62.5
	ATOM [12]	58.4	64.2
Non-real-time	ARTrackV2-L384 [1]	68.4	71.7
	LoRAT-L378 [27]	66.7	72.5
	ARTrack-L384 [37]	67.9	71.2
	SeqTrack-L384 [8]	66.2	68.5
	OSTrack [44]	66.5	70.7
	SimTrack [5]	-	71.2
	STARK [40]	66.2	68.2
	TransT [6]	65.7	69.1
	TrDiMP [34]	66.5	67.5
	DiMP [2]	61.8	64.3
	Ocean [45]	49.4	57.4

4.2. Comparisons on NFS and UAV123

As shown in Table 1, we compare UETrack with SOTA methods on two additional datasets: NFS [21] and UAV123 [30]. It can be seen that both UETrack-B and UETrack-S achieve the top two results on both datasets. Specifically, compared to the previous best real-time tracker AsymTrack-B [48], UETrack-B improves the AUC score by 4.0% on NFS and 3.7% on UAV123.

4.3. Comprehensive Evaluation on Depth Modality

Due to space limits in the main manuscript, we omitted the robustness metric in the comparison on the VOT-RGBD22 dataset, and we omitted the Pr metric in the comparison on the DepthTrack dataset. In this section, we provide the complete comparisons on both datasets, as shown in Table 2 and Table 3.

5. Additional Ablation Studies

In this section, we present additional ablation studies, as shown in Table 4. Models #1 to #8 are all trained without TAD. The baseline model (#1) is UETrack-B, which

Table 2. SOTA comparisons on VOT-RGBD22.

	Method	VOT-RGBD22			Speed (*fps*)		
		EAO	Acc.	Rob.	GPU	CPU	AGX
Real-time	UETrack-B (Ours)	68.3	80.8	83.8	**163**	**56**	**60**
	UETrack-S (Ours)	66.5	79.9	80.6	**183**	**68**	**67**
	UETrack-T (Ours)	62.5	77.4	76.5	**221**	**83**	**77**
	SUTrack-T [10]	68.1	**81.0**	83.9	100	23	34
	EMTrack [29]	**69.7**	80.6	**84.4**	109	29	36
	ViPT-Tiny [29]	68.5	80.4	83.5	56	22	20
Non-real-time	SeqTrackv2 [9]	74.4	81.5	91.0	23	2	5
	OneTracker [15]	72.7	81.9	87.2	-	-	-
	SDSTrack [16]	72.8	81.2	88.3	42	3	7
	Un-Track [39]	72.1	82.0	86.9	22	4	5
	ViPT [47]	72.1	81.5	87.1	55	6	13
	ProTrack [43]	65.1	80.1	80.2	-	-	-
	OSTrack [44]	67.6	80.3	83.3	105	11	19
	SPT [49]	65.1	79.8	85.1	25	-	-
	DeT [42]	65.7	76.0	84.5	37	-	-

Table 3. SOTA comparisons on DepthTrack.

	Method	DepthTrack			Speed (*fps*)		
		F-score	Re	Pr	GPU	CPU	AGX
Real-time	UETrack-B (Ours)	60.6	61.0	60.2	**163**	**56**	**60**
	UETrack-S (Ours)	58.9	58.0	59.7	**183**	**68**	**67**
	UETrack-T (Ours)	55.7	54.8	56.6	**221**	**83**	**77**
	SUTrack-T [10]	**61.7**	**62.1**	**61.2**	100	23	34
	EMTrack [29]	58.3	58.5	58.0	109	29	36
	ViPT-Tiny [29]	53.9	53.7	54.1	56	22	20
Non-real-time	SeqTrackv2 [9]	63.2	63.4	62.9	23	2	5
	OneTracker [15]	60.9	60.4	60.7	-	-	-
	SDSTrack [16]	61.9	60.9	61.4	42	3	7
	Un-Track [39]	61.0	60.8	61.1	22	4	5
	ViPT [47]	59.4	59.6	59.2	55	6	13
	ProTrack [43]	57.8	57.3	58.3	-	-	-
	OSTrack [44]	52.9	52.2	53.6	105	11	19
	SPT [49]	53.8	54.9	52.7	25	-	-
	DeT [42]	53.2	50.6	56.0	37	-	-

includes TP-MoE but does not use TAD. Model #9 represents the complete version of UETrack-B, which uses both TP-MoE and TAD.

Number of Experts. Based on the main manuscript, we further investigate the effect of varying the number of experts in TP-MoE, as shown in entries #2 and #3 of Table 4. Specifically, entries #2 and #3 adopt 64 and 247 experts, respectively, while the baseline model uses 8 experts by default. The results show that although the number of experts increases substantially, the average performance decreases by 0.4% and 0.3%, and the inference speed drops by 3 FPS and 6 FPS, respectively. These findings indicate that simply adding more experts does not yield better performance; instead, it introduces redundancy that leads to slower inference and reduced accuracy.

Table 4. Ablation Study. Δ denotes the performance change (averaged over benchmarks) compared with the baseline. The speed is measured on the AGX.

#	Method	LaSOT	DepthTrack	RGBT234	VisEvent	TNL2K	Speed	Δ
1	Baseline	68.5	59.4	63.1	57.9	57.1	60	–
2	64 Experts	68.0	59.2	62.3	56.8	57.5	57	**-0.4**
3	247 Experts	68.3	59.0	62.0	57.1	57.9	54	**-0.3**
4	Last 4 Layers	66.8	57.4	62.2	56.8	57.1	46	**-1.1**
5	Last 5 Layers	66.9	56.4	62.0	56.4	57.3	43	**-1.4**
6	All Layers	65.6	56.3	60.2	54.7	56.4	39	**-2.6**
7	First Layer	67.6	57.8	62.0	57.6	57.6	60	**-0.7**
8	First 3 Layers	66.8	57.1	61.8	56.5	57.5	48	**-1.3**
9	UETrack-B	69.2	60.6	64.2	59.2	58.0	60	**+1.0**
10	+ All Features	69.0	58.8	63.3	58.7	58.9	60	**+0.5**
11	+ Head Feat.	68.6	60.8	62.8	57.9	59.4	60	**+0.7**
12	+ SUTrack-S	67.9	59.6	62.9	58.2	58.5	60	**+0.2**

Insertion Layer of TP-MoE. Building on the main manuscript, we further investigate the effect of inserting TP-MoE at different positions in the backbone. As shown in entries #4 to #8 in Table 4, we evaluate the impact of inserting TP-MoE into the last 4 layers, last 5 layers, all layers, the first layer, and the first 3 layers. By default, the baseline inserts TP-MoE only in the last layer. The results show that all of these entries lead to a noticeable drop in average performance, with decreases of 1.1%, 1.4%, 2.6%, 0.7%, and 1.3%, respectively. This further supports our analysis in the main manuscript (*Ablation and Analysis* section): the semantic features in deeper layers are more stable, making them better suited for expert specialization. In contrast, inserting TP-MoE into earlier layers may interfere with low-level feature learning, resulting in degraded performance.

More Ablation Studies on TAD We further present ablation studies on TAD in Table 4, including entries #10, #11, and #12. Entry #9 is the default UETrack-B, which uses SUTrack-B [10] as the teacher model. It only uses the target distribution map and features from the last layer of the backbone for supervision. In other settings, #10 adds features from all backbone layers for supervision. #11 further includes the size map and offset map from the center head [44] as additional supervision signals. #12 changes the teacher model to SUTrack-S. The results show that compared with #9, the performance of #10, #11, and #12 drops by 0.5%, 0.3%, and 0.8%, respectively. This suggests that introducing too many supervision signals or changing the teacher model may bring redundant information, which weakens model performance instead of improving it.

6. Additional Visualization Results

Due to space limitations in the main manuscript, we provide additional visualization results in this section for qualitative comparison. These include visualizations of the attention

distributions of TP-MoE experts, visualizations of the distillation decisions made by TAD, visualizations of the feature maps output by TP-MoE, and visual comparisons with SOTA real-time trackers on the LaSOT dataset.

6.1. TP-MoE Expert Attention Visualization

We provide additional visualizations of the attention distributions of TP-MoE experts in Figure 2. It can be observed that each expert attends to a distinct region, and they work collaboratively to achieve precise target localization. Specifically, Expert 1 attends to the object center, Expert 5, Expert 6 and Expert 8 focus on the background, while Expert 7 concentrates on the object contour. Such collaboration and clear division of attention enable experts to learn complementary representations, thereby enhancing the model's feature modeling capability.

6.2. TAD Distillation Decision Visualization

We also provide additional visualizations of the distillation decisions made by TAD, as shown in Figure 3. It can be observed that in challenging tracking scenarios—such as fast motion, occlusion, deformation, illumination changes, blur, or similar distractor objects—TAD tends to disable distillation. This is because, under these conditions, the features and target distributions from the teacher model are often unreliable. Blindly applying distillation in such cases may negatively affect the student network. These results intuitively verify the effectiveness of the TAD decision mechanism. It can selectively suppress redundant or misleading supervision signals, thereby improving the stability of distillation and the overall training performance.

6.3. TP-MoE Feature Map Visualization

As shown in Figure 4, we visualize the feature maps produced by the TP-MoE module. It can be observed that the model focuses more on the target's contour after processing with TP-MoE. At the same time, it effectively suppresses surrounding distracting objects. This ability to focus on relevant features helps improve the accuracy and robustness of target localization.

6.4. LaSOT Visualization Comparison

As shown in Figure 5, we compare UETrack with state-of-the-art real-time trackers on the LaSOT dataset, including AsymTrack [48], HiT [18], FEAR [4], HCAT [7], and E.T.Track [3]. It can be observed that UETrack performs more robustly in challenging scenarios such as similar objects, small targets, fast motion, occlusion, and severe deformation. These results indicate that UETrack remains stable in complex real-world situations, highlighting its strong practical value.

7. Limitations

Although UETrack achieves a good balance between speed and accuracy and shows strong practicality and versatility, its performance on the language modality is still relatively limited. This is mainly because UETrack focuses on unified multi-modal modeling and overall efficiency, and does not include language-specific modules or targeted optimization for this modality. Future work may consider designing specialized components or strategies based on language characteristics to further improve its performance in the language modality. In addition, several studies [19, 46] show that explicitly using temporal information is important for improving robustness in complex scenes. However, UETrack currently does not model or use the temporal dimension explicitly. Therefore, another future direction is to explore how to integrate temporal information into UETrack in an effective and lightweight way while maintaining efficiency, which could further enhance its stability and overall performance.

Figure 2. Additional visualizations of the attention distributions of TP-MoE experts. Bright regions indicate attended areas, and each expert focuses on distinct spatial regions.

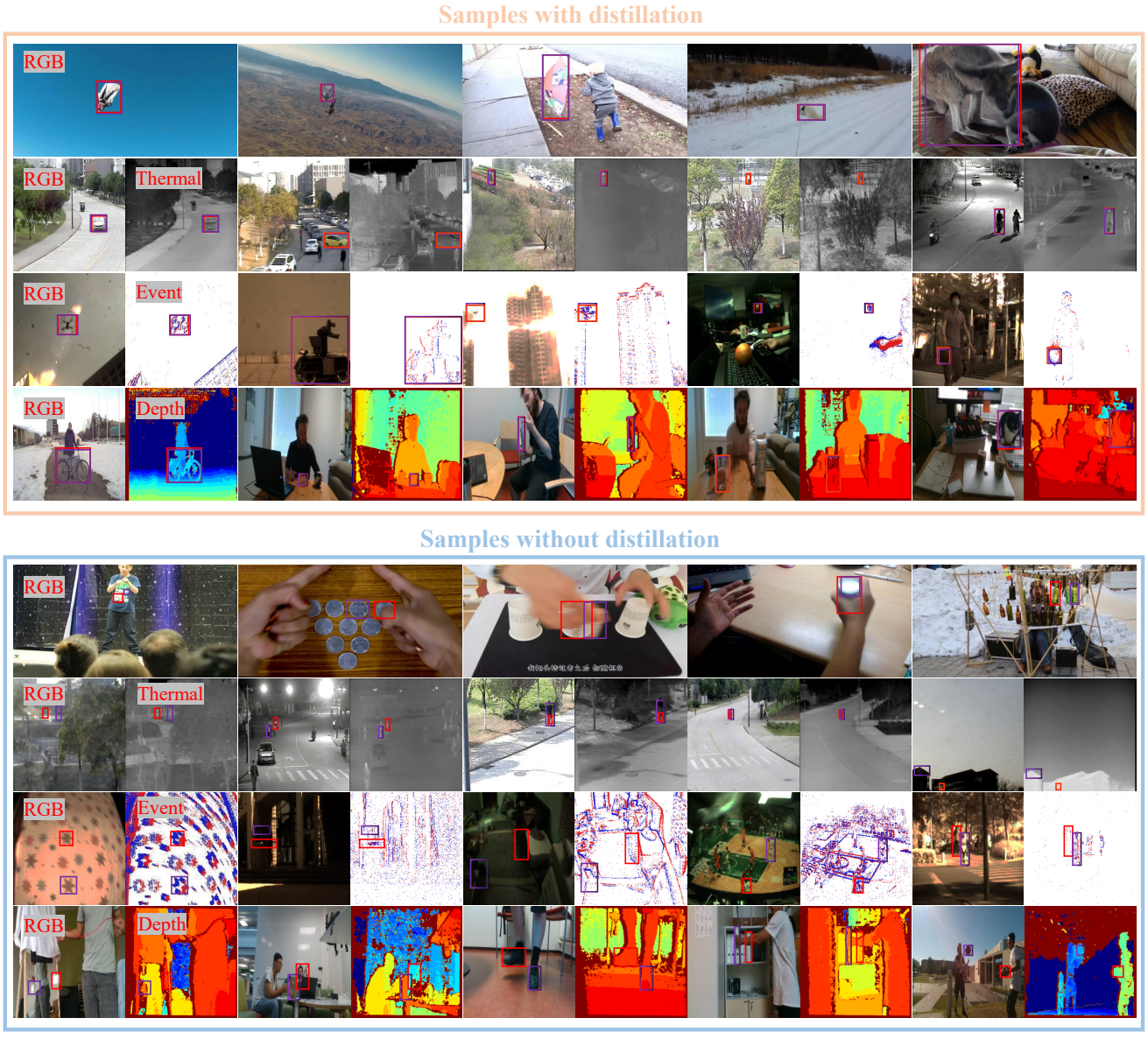

Figure 3. Additional visualizations of adaptive distillation decisions made by TAD across different modalities.

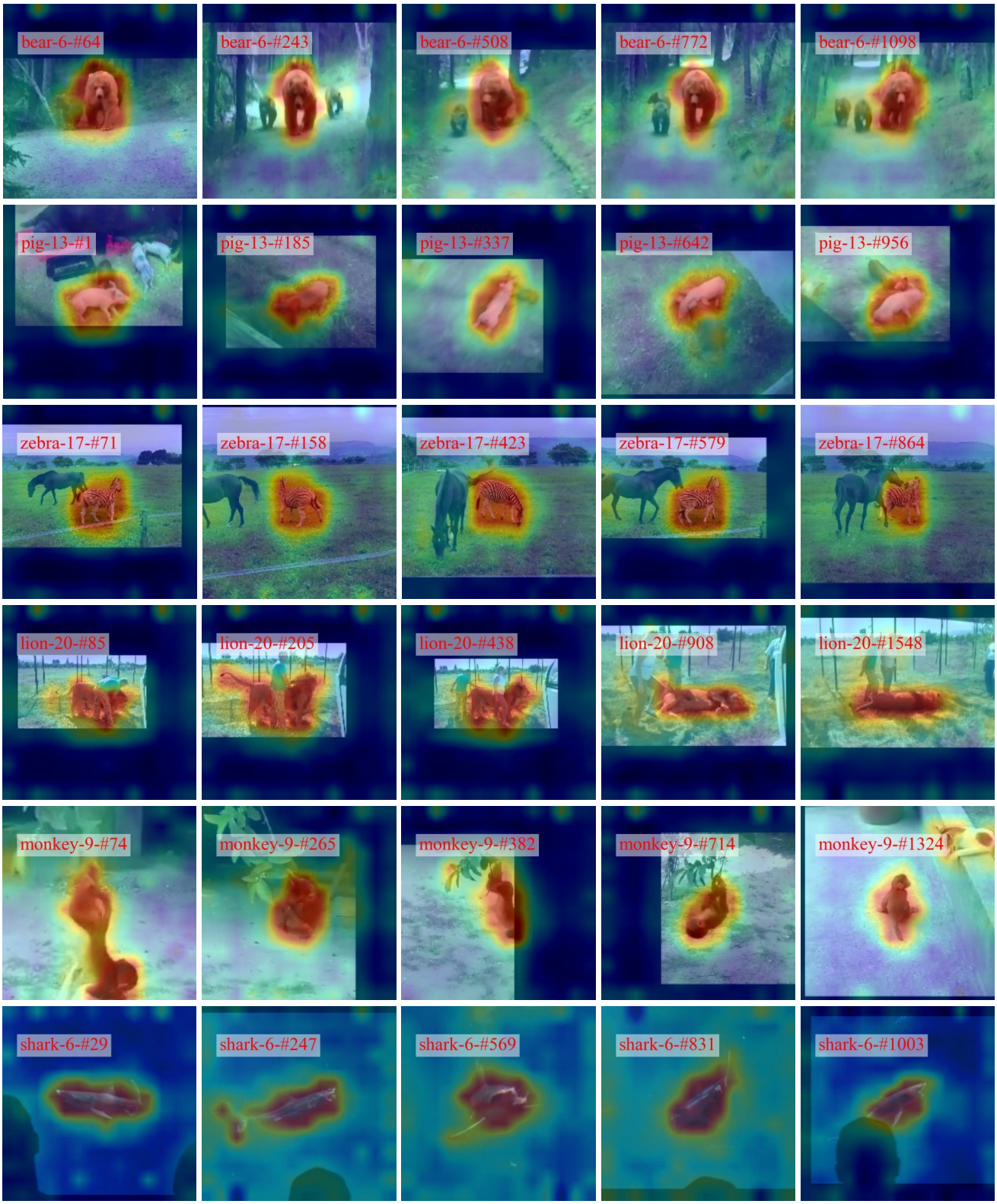

Figure 4. Visualization results of output features from TP-MoE.

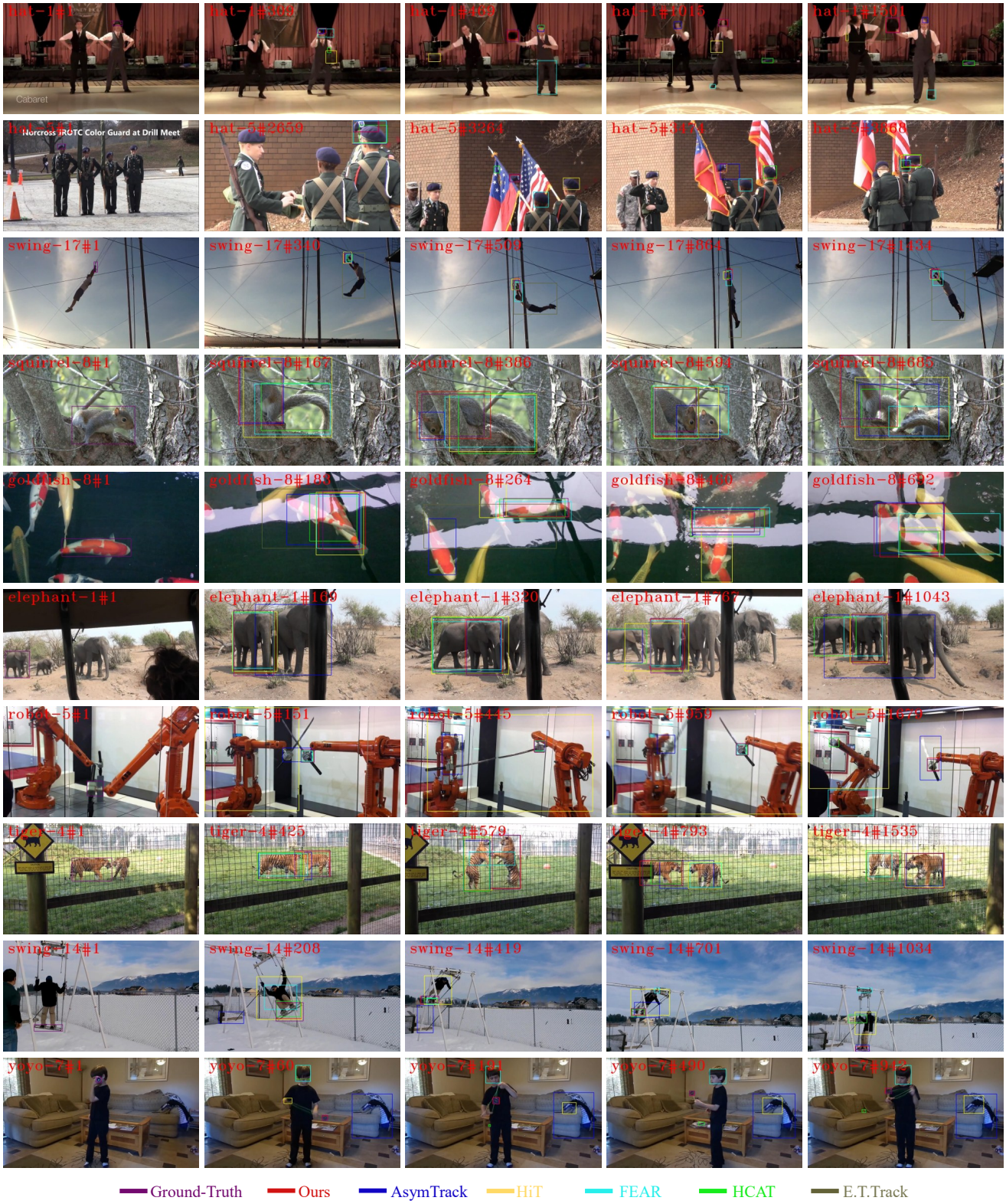

Figure 5. Comparison of visualization results on the LaSOT dataset with state-of-the-art trackers.

References

[1] Yifan Bai, Zeyang Zhao, Yihong Gong, and Xing Wei. ARTrackV2: Prompting autoregressive tracker where to look and how to describe. In *CVPR*, pages 19048–19057, 2024. 3

[2] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *ICCV*, pages 6182–6191, 2019. 3

[3] Philippe Blatter, Menelaos Kanakis, Martin Danelljan, and Luc Van Gool. Efficient Visual Tracking with Exemplar Transformers. In *WACV*, pages 1571–1581, 2023. 3, 5

[4] Vasyl Borsuk, Roman Vei, Orest Kupyn, Tetiana Martyniuk, Igor Krashenyi, and Jiři Matas. FEAR: Fast, Efficient, Accurate and Robust Visual Tracker. In *ECCV*, pages 644–663, 2022. 3, 5

[5] Boyu Chen, Peixia Li, Lei Bai, Lei Qiao, Qiuhong Shen, Bo Li, Weihao Gan, Wei Wu, and Wanli Ouyang. Backbone is all your need: A simplified architecture for visual object tracking. In *ECCV*, pages 375–392, 2022. 3

[6] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In *CVPR*, pages 8126–8135, 2021. 3

[7] Xin Chen, Ben Kang, Dong Wang, Dongdong Li, and Huchuan Lu. Efficient Visual Tracking via Hierarchical Cross-Attention Transformer. In *ECCVW*, pages 461–477, 2022. 3, 5

[8] Xin Chen, Houwen Peng, Dong Wang, Huchuan Lu, and Han Hu. Seqtrack: Sequence to sequence learning for visual object tracking. In *CVPR*, pages 14572–14581, 2023. 3

[9] 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*, 2024. 4

[10] Xin Chen, Ben Kang, Wanting Geng, Jiawen Zhu, Yi Liu, Dong Wang, and Huchuan Lu. Sutrack: Towards simple and unified single object tracking. In *AAAI*, pages 2239–2247, 2025. 2, 4

[11] Yutao Cui, Tianhui Song, Gangshan Wu, and Limin Wang. Mixformerv2: Efficient fully transformer tracking. In *NeurIPS*, pages 58736–58751, 2023. 3

[12] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ATOM: Accurate tracking by overlap maximization. In *CVPR*, pages 4660–4669, 2019. 3

[13] 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 *CVPR*, pages 5374–5383, 2019. 1, 2

[14] Heng Fan, Hexin Bai, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Mingzhen Huang, Juehuan Liu, Yong Xu, et al. LaSOT: A high-quality large-scale single object tracking benchmark. *IJCV*, pages 439–461, 2021. 2

[15] 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 *CVPR*, pages 19079–19091, 2024. 4

[16] 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 *CVPR*, pages 26551–26561, 2024. 4

[17] Lianghua Huang, Xin Zhao, and Kaiqi Huang. GOT-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE TPAMI*, pages 1562–1577, 2019. 1, 2

[18] Ben Kang, Xin Chen, Dong Wang, Houwen Peng, and Huchuan Lu. Exploring lightweight hierarchical vision transformers for efficient visual tracking. In *ICCV*, pages 9612–9621, 2023. 3, 5

[19] Ben Kang, Xin Chen, Simiao Lai, Yang Liu, Yi Liu, and Dong Wang. Exploring enhanced contextual information for video-level object tracking. In *AAAI*, pages 4194–4202, 2025. 5

[20] Ben Kang, Xin Chen, Jie Zhao, Chunjuan Bo, Dong Wang, and Huchuan Lu. Exploiting lightweight hierarchical vit and dynamic framework for efficient visual tracking. *IJCV*, pages 1–23, 2025. 3

[21] Hamed Kiani Galoogahi, Ashton Fagg, Chen Huang, Deva Ramanan, and Simon Lucey. Need for speed: A benchmark for higher frame rate object tracking. In *ICCV*, pages 1125–1134, 2017. 2, 3

[22] Matej Kristan, Jiří Matas, Aleš Leonardis, Michael Felsberg, Roman Pflugfelder, Joni-Kristian Kämäräinen, Hyung Jin Chang, Martin Danelljan, Luka Cehovin, Alan Lukežič, et al. The ninth visual object tracking vot2021 challenge results. In *ICCVW*, pages 2711–2738, 2021. 2

[23] Matej Kristan, Aleš Leonardis, Jiří Matas, Michael Felsberg, Roman Pflugfelder, Joni-Kristian Kämäräinen, Hyung Jin Chang, Martin Danelljan, Luka Čehovin Zajc, Alan Lukežič, et al. The tenth visual object tracking vot2022 challenge results. In *ECCVW*, pages 431–460, 2023. 2

[24] Chenglong Li, Xinyan Liang, Yijuan Lu, Nan Zhao, and Jin Tang. RGB-T object tracking: Benchmark and baseline. *PR*, page 106977, 2019. 2

[25] Chenglong Li, Wanlin Xue, Yaqing Jia, Zhichen Qu, Bin Luo, Jin Tang, and Dengdi Sun. LasHeR: A large-scale high-diversity benchmark for RGBT tracking. *IEEE TIP*, pages 392–404, 2021. 1, 2

[26] Zhenyang Li, Ran Tao, Efstratios Gavves, Cees GM Snoek, and Arnold WM Smeulders. Tracking by natural language specification. In *CVPR*, pages 6495–6503, 2017. 1, 3

[27] Liting Lin, Heng Fan, Zhipeng Zhang, Yaowei Wang, Yong Xu, and Haibin Ling. Tracking meets lora: Faster training, larger model, stronger performance. In *ECCV*, pages 300–318, 2024. 3

[28] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, pages 740–755, 2014. 1

[29] Chang Liu, Ziqi Guan, Simiao Lai, Yang Liu, Huchuan Lu, and Dong Wang. Emtrack: Efficient multimodal object tracking. *IEEE TCSVT*, pages 2202–2214, 2024. 4

[30] Matthias Mueller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for UAV tracking. In *ECCV*, pages 445–461, 2016. 2, 3

[31] 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 *ECCV*, pages 300–317, 2018. 1, 2

[32] Liang Peng, Junyuan Gao, Xinran Liu, Weihong Li, Shaohua Dong, Zhipeng Zhang, Heng Fan, and Libo Zhang. Vasttrack: Vast category visual object tracking. In *NeurIPS*, pages 130797–130818, 2024. 1

[33] Yunjie Tian, Lingxi Xie, Jihao Qiu, Jianbin Jiao, Yaowei Wang, Qi Tian, and Qixiang Ye. Fast-itpn: Integrally pre-trained transformer pyramid network with token migration. *IEEE TPAMI*, pages 1–15, 2024. 1

[34] Ning Wang, Wengang Zhou, Jie Wang, and Houqiang Li. Transformer meets tracker: Exploiting temporal context for robust visual tracking. In *CVPR*, pages 1571–1580, 2021. 3

[35] 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 *CVPR*, pages 13763–13773, 2021. 1, 3

[36] 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 TCYB*, pages 1997–2010, 2024. 1, 3

[37] Xing Wei, Yifan Bai, Yongchao Zheng, Dahu Shi, and Yihong Gong. Autoregressive visual tracking. In *CVPR*, pages 9697–9706, 2023. 3

[38] Yi Wu, Jongwoo Lim, and Ming Hsuan Yang. Object tracking benchmark. *IEEE TPAMI*, pages 2411–2418, 2013. 3

[39] 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 *CVPR*, pages 19156–19166, 2024. 4

[40] Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. Learning spatio-temporal transformer for visual tracking. In *ICCV*, pages 10448–10457, 2021. 3

[41] Bin Yan, Houwen Peng, Kan Wu, Dong Wang, Jianlong Fu, and Huchuan Lu. LightTrack: Finding Lightweight Neural Networks for Object Tracking via One-Shot Architecture Search. In *CVPR*, pages 15180–15189, 2021. 3

[42] Song Yan, Jinyu Yang, Jani Käpylä, Feng Zheng, Aleš Leonardis, and Joni-Kristian Kämäräinen. DepthTrack: Unveiling the power of RGBD tracking. In *ICCV*, pages 10725–10733, 2021. 1, 2, 4

[43] Jinyu Yang, Zhe Li, Feng Zheng, Ales Leonardis, and Jingkuan Song. Prompting for multi-modal tracking. In *ACMMM*, pages 3492–3500, 2022. 4

[44] Botao Ye, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Joint feature learning and relation modeling for tracking: A one-stream framework. In *ECCV*, pages 341–357, 2022. 3, 4

[45] Zhipeng Zhang, Houwen Peng, Jianlong Fu, Bing Li, and Weiming Hu. Ocean: Object-aware anchor-free tracking. In *ECCV*, pages 771–787, 2020. 3

[46] Yaozong Zheng, Bineng Zhong, Qihua Liang, Zhiyi Mo, Shengping Zhang, and Xianxian Li. ODtrack: Online dense temporal token learning for visual tracking. In *AAAI*, pages 7588–7596, 2024. 5

[47] Jiawen Zhu, Simiao Lai, Xin Chen, Dong Wang, and Huchuan Lu. Visual prompt multi-modal tracking. In *CVPR*, pages 9516–9526, 2023. 4

[48] Jiawen Zhu, Huayi Tang, Xin Chen, Xinying Wang, Dong Wang, and Huchuan Lu. Two-stream beats one-stream: Asymmetric siamese network for efficient visual tracking. In *AAAI*, pages 10959–10967, 2025. 3, 5

[49] Xue-Feng Zhu, Tianyang Xu, Zhangyong Tang, Zucheng Wu, Haodong Liu, Xiao Yang, Xiao-Jun Wu, and Josef Kittler. RGBD1K: A large-scale dataset and benchmark for RGB-D object tracking. In *AAAI*, pages 3870–3878, 2023. 4