PURA: Parameter Update-Recovery Test-Time Adaption for RGB-T Tracking

Supplementary Material

1. Summary

In this supplementary material, Section 2 presents the detailed setup of the method, covering the baseline architecture, computational resources, and relevant descriptions of the dataset. Section 3 provides the pseudo-code for the core algorithm. For further analysis of our method, please refer to Section 4.

2. Detailed Settings

Baseline Architecture. Our baseline adopts a one-stream ViT-B [2] backbone using a template update mechanism. Let $I_r^z, I_t^z \in \mathbb{R}^{3\times H_z\times W_z}$ represent the RGB and infrared templates, $I_r^x, I_t^x \in \mathbb{R}^{3\times H_x\times W_x}$ represent the RGB and infrared search regions, and $I_r^d, I_t^d \in \mathbb{R}^{3\times H_z\times W_z}$ represent the RGB and infrared dynamic templates. All input images are first passed through the PatchEmbed [2] layer to convert them into token sequences, yielding $P_r^z, P_t^z, P_r^d, P_t^d \in \mathbb{R}^{N_z\times C}$, $P_r^x, P_t^x \in \mathbb{R}^{N_x\times C}$ where N_z, N_x denote the number of RGB and infrared tokens, respectively. C represents the channel dimension. Subsequently, the token sequences of the two modalities are concatenated to form $\{P_r^z; P_t^z; P_r^x; P_t^x; P_r^d; P_t^d\}$, which is then fed into the backbone for joint feature extraction. In the output stage, the features of the search regions are processed by a convolutional head [15] to generate the final results. For template updates, our baseline utilizes a score prediction head [1] to predict the confidence of objects and performs updates to P_r^d, P_t^d based on a threshold.

Computing Environments. All of our experiments are conducted on a server running Ubuntu 22.04, equipped with an Intel(R) Xeon(R) Platinum 8276 CPU @ 2.20GHz, and accelerated by 10 NVIDIA A40 GPUs, each with 48GB of memory. The software environment is configured with Python 3.8.18, PyTorch 1.11.0, and CUDA 11.3 for our computational tasks.

Implementation Details. The baseline is trained using a two-stage method. In the first stage, four A40 GPUs are used. The batch size is 32, the learning rate is set to 1×10^{-4}, and 60,000 images are sampled for each epoch, totalling 15 training epochs. In order to learn the image-matching ability, all the parameters except the score prediction head are trained during this stage. At the 11th epoch, the learning rate is reduced to one-tenth of its original value. The AdamW [9] optimizer with a 1×10^{-4} weight decay is adopted for model optimization. The size of the search images is 256×256, and the sizes of both the template and online template images are 128×128. In the second stage, the training is only conducted on one A40 GPU. The learning rate is set to 1×10^{-4}, the batch size is 64, and 20,000 images are sampled for each epoch, totalling five epochs. At the third epoch, the learning rate is reduced to one-tenth of its original value, and other settings are the same as those in the first stage. In this stage, only the score prediction head is trained using positive and negative samples to learn the classification ability. Both stages of training use the LasHeR [6] training set. In addition, our parameter decomposition refers to the method in [14]. For parameter selection, we suggest setting its initial value as the BatchNorm layer's momentum in the fast parameter update step, tuning it step-by-step while suspending other strategies. The parameter decomposition recovery step starts with small learning rates η_μ and η_σ and adjusts for optimal performance. For adaptive momentum scaling, base ξ and m_λ on paper-given best values, be cautious with the truncation intervals of r and the final momentum value, and ensure the final momentum value's truncation interval is adjusted when modifying the basic value of momentum m.

Detailed Setup of Dataset Corruption. We evaluate the robustness and adaptability of our method under significant distribution shifts by corrupting images in the dataset. Specifically, we introduce controlled perturbations by shifting the brightness, contrast, saturation, and hue of each frame in the dataset. As shown in Table 1, our dataset corruption includes three different levels. The severity of image corruption progressively increases by applying random perturbations within a specified range for each attribute. Figure 1 shows the visualization of image corruption in the search region at different levels. This approach introduces a high degree of variability, challenging the model's performance under extreme and unpredictable conditions, thereby providing a comprehensive assessment of our method's performance in the face of severe environmental disturbances.

2.1. Datasets

GTOT [4]. As the first benchmark designed explicitly for RGB-T tracking, the GTOT dataset comprises 50 carefully matched pairs of video sequences, each consisting of an infrared video and its corresponding grayscale video. These videos cover various shooting environments, including laboratories, campus roads, and playgrounds, providing a rich array of testing scenarios for various tracking algorithms.

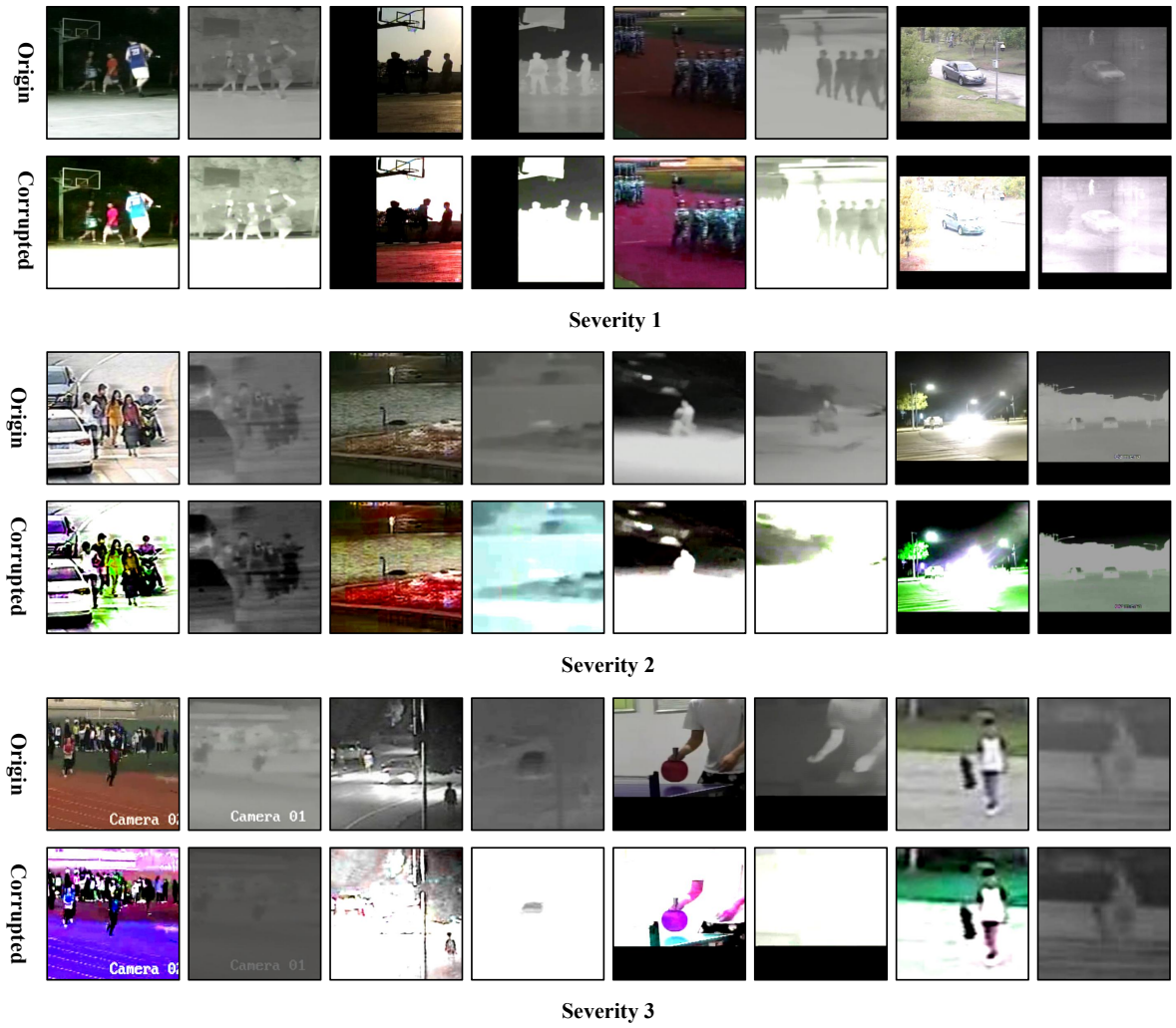

Figure 1. Visualization of image corruption. Our image corruption includes three levels, and as the severity increases, the images exhibit progressively more pronounced distortions.

RGBT210 [7]. The RGBT210 dataset comprises 210 precisely annotated pairs of RGB and infrared videos, offering more than 210,000 frames in total. For each tracked object, the dataset supplies ground truth annotations. Additionally, it highlights 12 unique attributes, including deformation and fast motion, which allow for a detailed performance analysis of tracking algorithms based on specific characteristics.

RGBT234 [5]. The RGBT234 dataset comprises 234 precisely aligned pairs of visible and infrared video sequences and encompasses approximately 234,000 frames, with individual video sequences reaching up to about 8,000 frames in length. Not only does RGBT234 expand the scope of available data, but it also emphasizes the provision of detailed annotation information, covering up to 12 distinct attributes.

LasHeR [6]. The LasHeR dataset is a large-scale RGB-T tracking dataset, encompassing 1,224 pairs of RGB and infrared videos, totalling over 734,800 frame pairs. All frame pairs are spatially aligned and manually annotated with bounding boxes, ensuring high-quality and dense annotations. The dataset covers many object categories, multiple camera perspectives, complex scenes, and varying environmental factors, including different seasons, weather conditions, and day-night variations, showcasing a high degree of diversity.

Table 1. Detailed setup of dataset corruption.

Severity	Attribute			
	Brightness	Contrast	Saturation	Hue
1	[0.50, 2.00]	[0.50, 2.00]	[0.50, 2.00]	[-0.20, 0.20]
2	[0.33, 3.00]	[0.33, 3.00]	[0.33, 3.00]	[-0.30, 0.30]
3	[0.25, 4.00]	[0.25, 4.00]	[0.25, 4.00]	[-0.40, 0.40]

2.2. Evaluation Metrics

Precision Rate. The precision rate (PR) measures the accuracy of the predicted object location by calculating the Euclidean distance between the centers of the tracking bounding box and the ground truth bounding box. The precision rate is the proportion of frames in which the distance between the tracking result and the actual position is less than a given threshold. The formula for the precision rate is:

$$\text{PR} = \frac{1}{N}\sum_{t=1}^{N}\delta(c_t < \xi_{pr}), \tag{1}$$

where c_t is the distance between the predicted and ground truth center locations in frame t, N represents the total number of frames, $\delta(\cdot)$ is the indicator function, and ξ_{pr} denotes the threshold.

Success Rate. The success rate (SR) evaluates the overlap between the predicted and ground truth bounding boxes. It is determined by the percentage of frames in which the Intersection over Union (IoU) exceeds a specified threshold. The formula for success rate can be expressed as:

$$\text{SR} = \frac{1}{N}\sum_{t=1}^{N}\delta(\text{IoU}_t \geq \xi_{sr}), \tag{2}$$

where IoU_t represents the IoU between predicted and ground truth bounding boxes in frame t, and ξ_{sr} indicates the threshold.

Maximum Precision Rate. The maximum precision rate (MPR) addresses alignment errors between RGB and infrared images. It evaluates precision using the smaller distance between the ground truth and predicted bounding box centers of RGB and infrared images.

Maximum Success Rate. The maximum success rate (MSR) is also designed to address alignment errors. It evaluates the success rate using the larger IoU between the ground truth and predicted bounding box of RGB and infrared images.

3. Algorithm

In this section, we provide a detailed introduction to the three components that underpin our research, illustrated with pseudo-code for their specific implementation steps. These components are fast parameter update (Algorithm 1), parameter decomposition recovery (Algorithm 2), and adaptive momentum scaling (Algorithm 3).

Algorithm 1 Fast Parameter Update

Require: Initial training mean $\hat{\mu}_0$ and variance $\hat{\sigma}_0^2$
Require: Momentum parameter m
Require: The number N of elements of each test data
Require: A sequence of sample test data $x_1, x_2, ..., x_T$
  Initialize: $\hat{\mu} \leftarrow \hat{\mu}_0$, $\hat{\sigma}^2 \leftarrow \hat{\sigma}_0^2$
  for $t = 1$ to T **do**
    Calculate the mean and variance for the current data:
      $\mu_t \leftarrow \sum x_t$/N
      $\sigma_t^2 \leftarrow \sum(x_t - \mu)^2$/ (N-1)
    Update the running mean and running var:
      $\hat{\mu} \leftarrow (1-m)\cdot\hat{\mu} + m\cdot\mu_t$
      $\hat{\sigma}^2 \leftarrow (1-m)\cdot\hat{\sigma}^2 + m\cdot\sigma_t^2$
  end for

Algorithm 2 Parameter Decomposition Recovery

Require: Model recovery interval K
Require: Learning rates η_μ, η_σ
  Initialize pseudo-gradient set $\mathbf{\Theta} \leftarrow \emptyset$
  for each frame $t = 1, 2, 3, \ldots$ **do**
    Obtain the updated $\hat{\mu}$ and $\hat{\sigma}^2$ from Algorithm 1
    Obtain μ_{t-1} and σ^2_{t-1} calculated by Algorithm 1
    Construct pseudo-gradient set:
      $\nabla_p \leftarrow \{\hat{\mu} - \mu_{t-1}, \hat{\sigma}^2 - \sigma^2_{t-1}\}$
      $\mathbf{\Theta} \leftarrow \mathbf{\Theta} \cup \{\nabla_p\}$
    if $t \bmod \mathrm{K} = 0$ **then**
      Perform decomposition for pseudo-gradient set:
        $\bar{\mathbf{\Theta}} \leftarrow \mathbf{\Theta} - \mathrm{mean}(\mathbf{\Theta})$
        $\lambda, z \leftarrow \textbf{SVD}\left(\frac{1}{\mathrm{K}-1}\bar{\mathbf{\Theta}}^\top \bar{\mathbf{\Theta}}\right)$
      Calibrate and accumulate principal directions:
        $\nabla\theta \leftarrow \{\mu_{t-\mathrm{K}} - \mu_t, \sigma^2_{t-\mathrm{K}} - \sigma^2_t\}$
        $\hat{z} \leftarrow \mathcal{N}_2\left(\bar{\mathbf{\Theta}}^\top z\right) \cdot \|\nabla\theta\|_2$
        $\nabla_{\hat{p}} \leftarrow \sum \mathcal{N}_2(\lambda) \cdot \delta(\nabla\theta \cdot \hat{z}) \cdot \hat{z}$
      Recover parameters:
        $\hat{\mu}_t \leftarrow \hat{\mu}_{t-1} + \eta_\mu \cdot \nabla^{(\mu)}_{\hat{p}}$
        $\hat{\sigma}^2_t \leftarrow \hat{\sigma}^2_{t-1} + \eta_\sigma \cdot \nabla^{(\sigma)}_{\hat{p}}$
      Reset pseudo-gradient set
        $\mathbf{\Theta} \leftarrow \emptyset$
    end if
  end for

Algorithm 3 Adaptive Momentum Scaling

Require: Initial momentum m
Require: Momentum parameter m_λ
Require: Model recovery interval K
Require: Threshold parameter ξ
Require: Truncation function $\mathcal{T}(\cdot)$
  Initialize: $\bar{\lambda} \leftarrow 0$
  for each frame $t = 1, 2, 3, \ldots$ **do**
    if $t \bmod \mathrm{K} = 0$ **then**
      Obtain the eigenvalues λ from Algorithm 2
      Compute the sum of eigenvalues:
        $\lambda \leftarrow \sum \lambda$
      Update the exponential moving average of λ:
        $\bar{\lambda} \leftarrow (1 - m_\lambda) \cdot \bar{\lambda} + m_\lambda \cdot \lambda$
      Calculate the scaling factor:
        $\Delta\lambda \leftarrow |\lambda - \bar{\lambda}|$
        $r \leftarrow \mathcal{T}(\xi - \Delta\lambda)$
      Update the momentum:
        $\hat{m} \leftarrow \mathcal{T}(r \cdot m)$
    end if
  end for

Table 2. Attribute-based MPR/MSR scores (%) on the LasHeR→RGBT234 scenario compared with several state-of-the-art trackers. The best results are highlighted in **bold**.

Attribute	ViPT [16]	TBSI [3]	MPLT [10]	No Adap.	PURA
NO	92.4/71.0	96.2/73.9	97.7/74.8	96.6/73.8	**98.0/76.1**
PO	85.4/63.0	88.6/66.1	88.5/65.5	93.7/70.4	**94.0/71.7**
HO	77.6/56.3	83.8/61.9	84.1/61.7	85.2/62.1	**90.6/66.3**
LI	81.0/58.4	88.3/65.2	87.6/64.4	92.5/68.3	**94.6/70.9**
LR	83.1/59.4	85.6/62.2	87.6/62.8	88.3/63.4	**91.2/66.5**
TC	83.0/62.2	86.2/65.1	85.0/64.2	89.9/67.4	**91.1/68.4**
DEF	81.7/62.2	84.6/65.0	85.8/65.4	89.2/67.9	**91.8/70.4**
FM	80.3/58.6	82.3/61.1	83.5/61.8	86.1/63.7	**90.5/67.7**
SV	83.8/63.0	89.3/67.6	89.0/67.2	92.0/69.5	**93.7/71.9**
MB	83.2/62.6	89.2/67.7	86.1/64.8	87.1/65.9	**91.7/69.3**
CM	83.0/62.1	87.0/66.0	86.9/65.3	89.9/67.5	**91.8/69.0**
BC	79.6/55.7	83.9/59.9	84.5/60.4	82.3/58.2	**88.3/63.1**
ALL	83.5/61.7	88.0/65.8	88.4/65.7	90.8/67.6	**93.3/70.3**

Table 3. Attribute-based PR/SR scores (%) on the LasHeR→RGBT210 scenario compared with several state-of-the-art TTA methods. The best results are highlighted in **bold**.

Attribute	Tent [12]	ETA [11]	CoTTA [13]	AdaBN [8]	PURA
NO	94.9/73.8	94.6/73.8	**95.4/74.2**	94.8/73.7	94.9/73.8
PO	91.1/68.4	91.4/68.5	90.8/67.7	91.2/68.2	**91.5/68.6**
HO	86.5/62.2	85.6/61.2	82.9/59.5	84.9/60.6	**87.1/62.0**
LI	**91.7/67.2**	89.2/65.5	88.8/65.4	89.1/65.0	91.5/67.1
LR	76.3/52.7	81.9/55.9	**84.6/57.7**	78.4/53.9	83.1/56.6
TC	85.6/64.3	87.4/65.3	84.5/63.3	87.0/65.0	**89.0/66.4**
DEF	88.3/66.9	88.8/66.8	88.3/66.6	88.0/66.4	**89.5/67.2**
FM	83.7/61.2	87.1/63.3	87.6/63.2	86.1/63.2	**88.9/64.0**
SV	90.6/69.1	90.5/69.1	89.4/68.0	89.9/68.4	**91.1/69.2**
MB	84.7/62.9	85.2/62.9	85.6/62.8	83.9/62.0	**85.7/63.2**
CM	86.4/63.9	87.8/64.4	85.6/62.8	88.0/64.5	**88.1/64.6**
BC	82.9/57.9	82.3/57.0	79.2/54.9	80.2/55.4	**84.3/58.2**
ALL	89.9/66.8	89.6/66.4	88.4/65.4	89.2/66.0	**90.3/70.3**

Table 4. Attribute-based MPR/MSR scores (%) on the LasHeR→RGBT234 scenario compared with several state-of-the-art TTA methods. The best results are highlighted in **bold**.

Attribute	Tent [12]	ETA [11]	CoTTA [13]	AdaBN [8]	PURA
NO	97.9/76.1	96.6/75.2	97.9/**76.2**	96.6/74.9	**98.0**/76.1
PO	92.0/70.4	93.5/71.5	92.9/70.7	**94.0/71.7**	**94.0/71.7**
HO	88.8/65.2	89.0/65.3	86.5/63.1	87.1/63.8	**90.6/66.3**
LI	91.9/69.1	93.1/70.1	90.6/68.7	92.0/69.1	**94.6/70.9**
LR	89.1/65.0	88.7/64.9	90.1/65.4	90.2/65.6	**91.2/66.5**
TC	88.7/67.2	88.6/67.3	87.3/66.0	88.3/67.0	**91.1/68.4**
DEF	90.1/69.2	91.2/70.0	88.3/67.8	90.0/69.2	**91.8/70.4**
FM	87.1/65.1	86.1/64.8	89.4/66.6	86.2/64.6	**90.5/67.7**
SV	91.7/70.6	93.3/71.6	92.5/70.9	91.4/70.3	**93.7/71.9**
MB	89.4/67.3	89.1/67.8	89.8/67.6	87.8/66.8	**91.7/69.3**
CM	89.3/67.0	89.7/67.6	89.7/67.3	89.8/67.7	**91.8/69.0**
BC	84.4/60.7	86.1/61.8	84.6/59.8	84.9/60.6	**88.3/63.1**
ALL	91.8/69.3	92.3/69.7	91.2/68.6	91.6/69.1	**93.3/70.3**

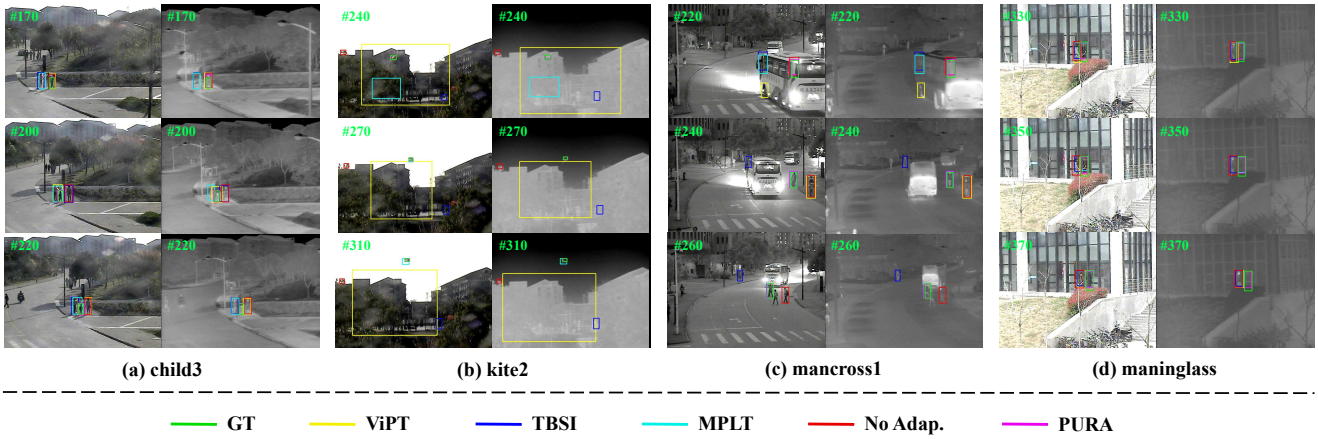

Figure 2. Qualitative comparison of our method with other RGB-T trackers across four representative sequences in the RGBT234 dataset.

4. Further Analysis

Analysis Across Various Attributes. We conduct a more comprehensive performance analysis of various methods across two datasets: RGBT210 [7] and RGBT234 [5]. Table 2 presents the detailed attribute-based performance of PURA in the LasHeR→RGBT234 scenario. Table 3 and Table 4 respectively show the detailed attribute-based performance of PURA compared to other TTA methods in the LasHeR→RGBT210 and LasHeR→RGBT234 scenarios. The experimental results indicate that PURA outperforms existing methods in most attributes. Particularly in challenging scenarios, our method demonstrates consistent superiority, with its robustness and adaptability being especially prominent.

Qualitative Comparison. We conduct a qualitative comparison of our method with other RGB-T trackers. As shown in Figure 2, we select four representative sequences from the RGBT234 dataset that encompass various challenges, including scale variations, occlusions, and fast motion, to evaluate the performance of different methods. For instance, in the second sequence, despite the rapid movement of the target leading to swift changes in the background, our method still demonstrates outstanding stability and accuracy. Additionally, our approach successfully addresses several common challenges in the other sequences. These results indicate that our proposed method, PURA, exhibits superior adaptability. To visually emphasize the advantages of our method over existing approaches in challenging attributes, we provide additional visualization results. Figure 3 shows the tracking results of the target under low light conditions. Figure 4 shows the tracking results of the target under partial occlusion conditions. Figure 5 shows the tracking results of the target under motion blur conditions.

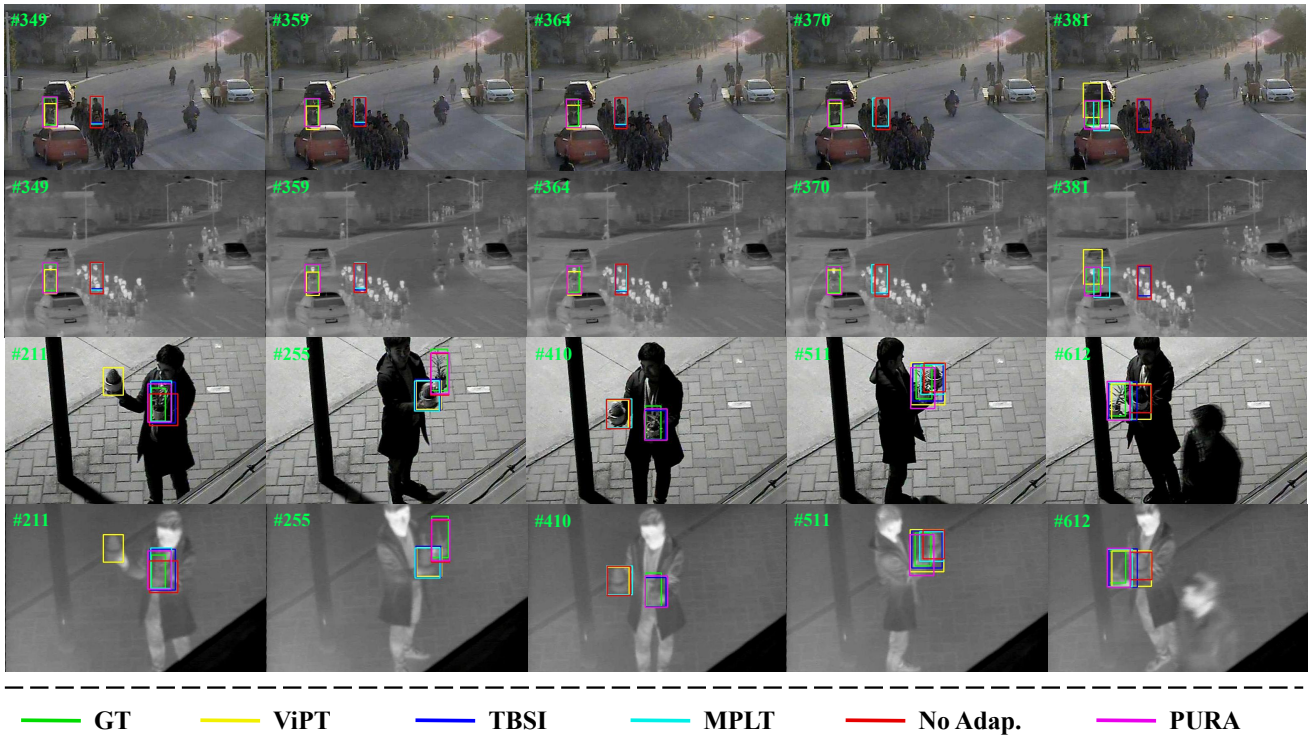

Figure 3. Qualitative comparison of various methods under low light conditions in the LasHeR → RGBT234 scenario.

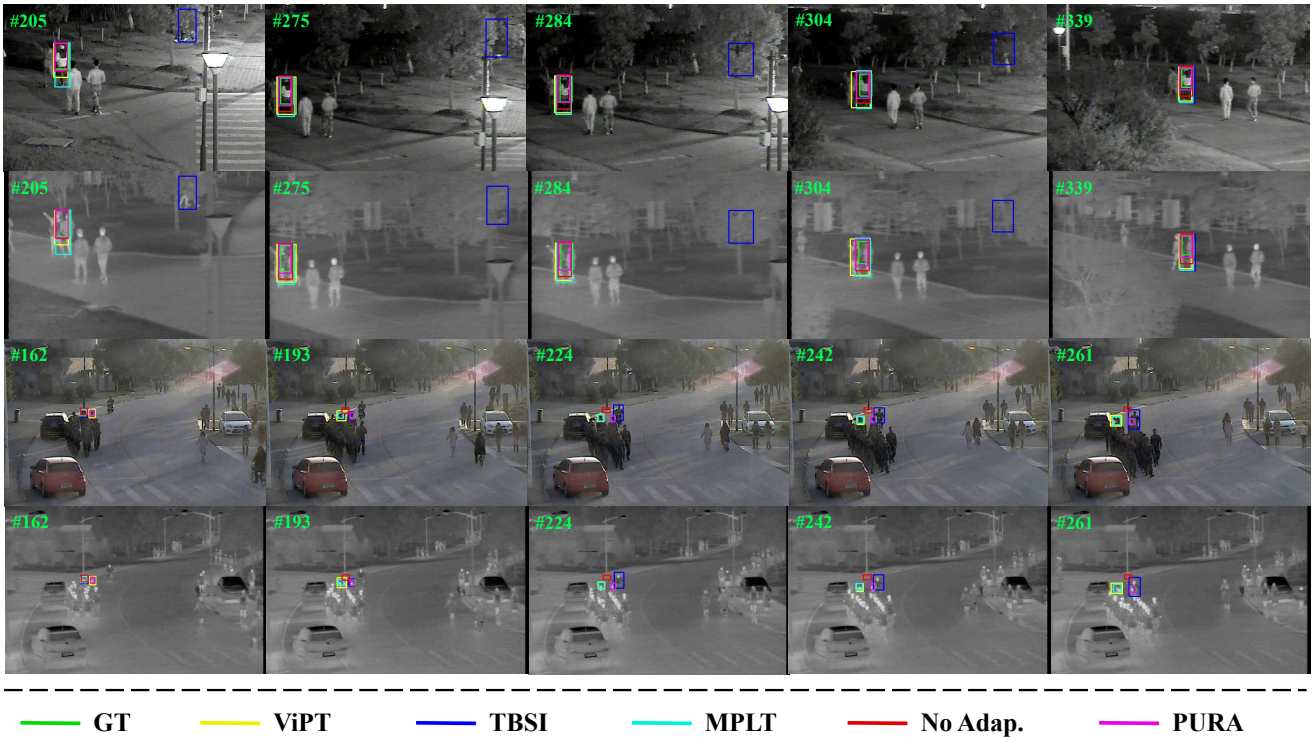

Figure 4. Qualitative comparison of various methods under partial occlusion conditions in the LasHeR → RGBT234 scenario.

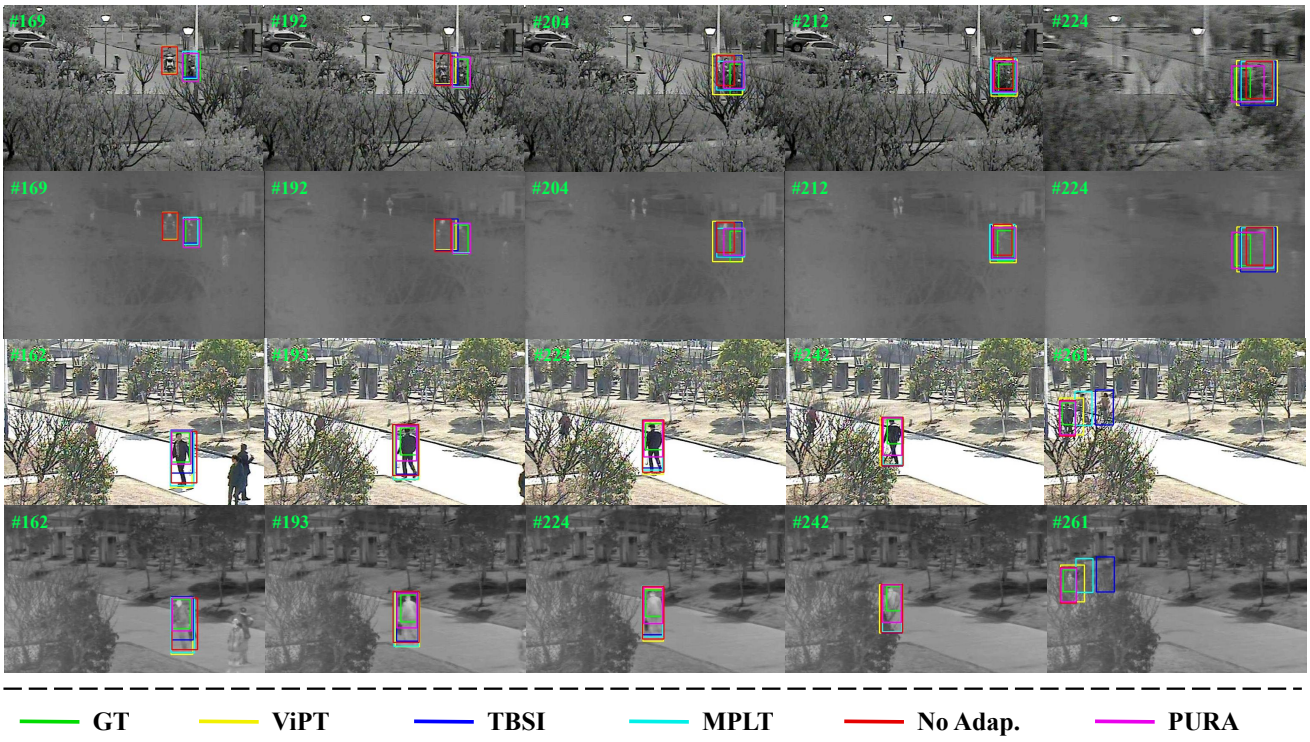

Figure 5. Qualitative comparison of various methods under motion blur conditions in the LasHeR → RGBT234 scenario.

References

[1] 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*, pages 13608–13618, 2022. 1

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1

[3] Tianrui Hui, Zizheng Xun, Fengguang Peng, Junshi Huang, Xiaoming Wei, Xiaolin Wei, Jiao Dai, Jizhong Han, and Si Liu. Bridging search region interaction with template for rgb-t tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13630–13639, 2023. 5

[4] Chenglong Li, Hui Cheng, Shiyi Hu, Xiaobai Liu, Jin Tang, and Liang Lin. Learning collaborative sparse representation for grayscale-thermal tracking. *IEEE Transactions on Image Processing*, 25(12):5743–5756, 2016. 1

[5] Chenglong Li, Xinyan Liang, Yijuan Lu, Nan Zhao, and Jin Tang. Rgb-t object tracking: Benchmark and baseline. *Pattern Recognition*, 96:106977, 2019. 2, 6

[6] 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 Transactions on Image Processing*, 31:392–404, 2021. 1, 2

[7] Chenglong Li, Nan Zhao, Yijuan Lu, Chengli Zhu, and Jin Tang. Weighted sparse representation regularized graph learning for rgb-t object tracking. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1856–1864, 2017. 2, 6

[8] Yanghao Li, Naiyan Wang, Jianping Shi, Xiaodi Hou, and Jiaying Liu. Adaptive batch normalization for practical domain adaptation. *Pattern Recognition*, 80:109–117, 2018. 5

[9] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 1

[10] Yang Luo, Xiqing Guo, Hui Feng, and Lei Ao. Rgb-t tracking via multi-modal mutual prompt learning. *arXiv preprint arXiv:2308.16386*, 2023. 5

[11] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*, pages 16888–16905. PMLR, 2022. 5

[12] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*. 5

[13] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7201–7211, 2022. 5

[14] Zhe Wang, Jake Grigsby, and Yanjun Qi. Pgrad: Learning principal gradients for domain generalization. In *The Eleventh International Conference on Learning Representations*. 1

[15] 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*, pages 341–357. Springer, 2022. 1

[16] 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*, pages 9516–9526, 2023. 5