electronics

Article

Parameter-Efficient Tuning for Object Tracking by Migrating Pre-Trained Decoders

Ruijuan Zhang [1,2,*], Li Wang [3] and Song Yang [2]

[1] School of Mathematical Science, Huaiyin Normal University, Huai'an 223300, China
[2] College of Information Science and Engineering, Hohai University, Nanjing 211100, China; 150207080003@hhu.edu.cn
[3] School of Computer and Software, Nanjing Vocational University of Industry Technology, Nanjing 210023, China; li1019wang@niit.edu.cn
* Correspondence: zhangruijuan@hytc.edu.cn

Abstract: Video object tracking has taken advantage of pre-trained weights on large-scale datasets. However, most trackers fully fine-tune all the backbone's parameters for adjusting to tracking-specific representations, where the utilization rate of parameter adjustment is inefficient. In this paper, we aim to explore whether a better balance can be achieved between parameter efficiency and tracking performance, and fully utilize the weight advantage of training on large-scale datasets. There are two main differences from a normal tracking paradigm: (i) We freeze the pre-trained weights of the backbone and add a dynamic adapter structure for every transformer block for tuning. (ii) We migrate the pre-trained decoder blocks to the tracking head for better generalization and localization. Extensive experiments are conducted on both mainstream challenging datasets and datasets for special scenarios or targets such as night-time and transparent objects. With the full utilization of pre-training knowledge, we found through experiments that a few tuned parameters can compensate for the gap between the pre-trained representation and the tracking-specific representation, especially for large backbones. Even better performance and generalization can be achieved. For instance, our AdaDe-B256 tracker achieves 49.5 AUC on the $\text{LaSOT}_{\text{ext}}$ which contains 150 sequences.

Keywords: visual object tracking; transformer; parameter-efficient tuning

check for updates

Citation: Zhang, R.; Wang, L.; Yang, S. Parameter-Efficient Tuning for Object Tracking by Migrating Pre-Trained Decoders. *Electronics* **2024**, *13*, 4621. https://doi.org/10.3390/electronics13234621

Academic Editor: Hideaki IIduka

Received: 18 October 2024
Revised: 18 November 2024
Accepted: 19 November 2024
Published: 22 November 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Visual object tracking is a fundamental visual task that tracks the target specified in the initial frame in subsequent frames. In recent years, great improvement has been achieved in the field of tracking, specifically with the further development of pre-trained models [1–3] and the application of transformer structures [4] in visual tasks.

Early backbones for tracking, such as AlexNet [5] and ResNet [6], are usually pre-trained with the ImageNet dataset with a supervised training method. The tracking pipeline can be modeled as a two-stream matching process, between search and template regions. With the introduction of Vision Transformer (ViT), many works benefit from the transformer model in vision tasks [7,8]. Tracking can enjoy one-stream modeling which promotes the extracting representations and relation interactions benefiting from the transformer model. Moreover, various self-supervised training methods have been explored for ViT, prompting generalizing representation learning with richer training data. For instance, the contrastive learning model CLIP [1] and masked image modeling model MAE [2] have shown their excellent performance as the pre-trained weights [9,10]. DropMAE [11] improves MAE pre-training with video data rather than static image data, exploiting prior temporal correspondence information and making pre-trained weights more suitable for transferring to the downstream tracking task. Tracking specified representation is learned further in the masked image modeling method with rich tracking video data in MAT [12].

Although pre-trained weights have a significant impact on tracking performance [10], they only exert their influence on a tracking model as an initialization method. Full parameter fine-tuning is still necessary, which results in very low adjusting efficiency of parameters for the downstream tracking task and consumes more computing resources.

Since pre-trained models may come into contact with richer training data than tracking data, and even multi-modal data [1], especially large models that may benefit more from large amounts of pre-training data, does the learned pre-task representation and tracking specified representation contain only a small fraction of the task differences, such as priors of temporal relations? As parameter-efficient transfer learning (PETL) has attracted attention as an alternative to full fine-tuning, we also resort to that method for exploring that issue. In this paper, we freeze the pre-trained weights of the backbone and add a dynamic adapter structure for every transformer block for tuning. Inspired by [13], we also migrate the pre-trained decoder blocks in MAE pre-training ahead of the tracking head for better generalization. This can be regarded as an extension of the traditional tracking head, which only encoder features of the search region are fed into.

Our significant contributions are summarized below:

- We leverage adapter adjusting parameters to improve transfer learning in tracking. Attached to pre-trained models, with only a few additional parameters trained, tracking performance comparable to full fine-tuning can be achieved, dramatically increasing parameter efficiency.
- We migrate pre-trained transformer decoders in MAE pre-training to enhance the tracking head, increasing the robustness and generalization of tracking.

2. Related Work

Visual Object Tracking. We focus on single-object tracking rather than multiple-object tracking [14–18]. Recently, there have been significant developments in visual object tracking technology, especially with the support of the transformer structure. The transformer structure can promote the relation modeling between the template and search region features such as a neck module [19–22], and its attention modeling mechanism facilitates the one-stream pipeline [9,10] as well. The one-stream tracking pipeline makes tracking-specific representation learning and information interaction between template–search pairs optimized simultaneously, achieving strong discriminative power and a better performance–speed trade-off.

Inspired by development in pre-training methods, some works focus on the optimization of tracking-specified representation learning [11,12]. DropMAE [11] introduces masked autoencoder (MAE) pre-training in videos as an alternative to MAE pre-training in static images, ImageNet, and the latter is regarded as a sub-optimal method for a video object tracking task. MAT [12] can learn to track specified representations through a simple encoder–decoder pipeline via a masked appearance transfer technique. However, these works may require newly designed pre-training or additional fine-tuning of all parameters, consuming more computational resources. Currently, new pre-training methods are constantly emerging [1–3], and pre-trained models can learn more robust and generalized representations from large-scale datasets. Given this situation, in this work, we draw support from the recent parameter-efficient transfer learning method to explore whether it is possible to make slight adjustments on the basis of pre-trained models for obtaining better tracking-specified representations.

Parameter-Efficient Transfer Learning. Due to the increasing size of pre-trained networks, fine-tuning all network parameters in downstream tasks has become challenging in terms of computational resources and storage. Parameter-efficient transfer learning (PETL) is then presented to tackle with that issue in the NLP field [23], and has become popular in the field of computer vision as well. Common tuning methods include prompt tuning, adapter tuning and LoRA tuning. VPT [24] is a typical prompt-tuning work in vision. A set of learnable parameters is added to a pre-trained model, resulting in better performance than full fine-tuning on downstream tasks. Adapter tuning [25–27] always

inserts a lightweight module into the original backbone structure for single or multiple downstream tasks. LoRA [28] tuning further presents low-rank matrices to approximate weight updating.

Recently, parameter-efficient transfer learning has received attention in the visual tracking field as well. By leveraging the power of RGB tracking foundation models, multi-modal tracking methods [29–33] can perform efficient training with a small number of parameters, and achieve comparable or better performance than the fully fine-tuned paradigm. These works demonstrate that RGB tracking models can adapt to multi-modal tracking with only a small number of parameters' adjustments. Since multi-modal tracking representations and RGB-specific tracking representations can be bridged via a few parameters tuned, we are curious: *is there a similar conclusion about RGB-specific tracking representations and pre-trained representations?* We also investigated the effectiveness of PETL under different pre-training tasks, including contrastive learning such as CLIP [1] and masked image modeling such as MAE [2].

3. Proposed Approach

3.1. Preliminary

The one-stream tracking pipeline has shown its strong discriminative power and become the mainstream pipeline applied in many works [9,10,34–36]. With the attention mechanism of the transformer, the template and the search region are embedded in the same space for similarity calculation, and all-layer information interaction can better promote the template-guided feature extraction of search regions.

The one-stream pipeline first embeds the template region $I_z \in \mathbb{R}^{3*H_z*W_z}$ and search region $I_x \in \mathbb{R}^{3*H_x*W_x}$ into a series of patch embeddings $F_{xz} \in \mathbb{R}^{(N_x+N_z)*d}$. H_z is the height of the template region; W_z is the width of the template region. H_x is the height of the search region; W_x is the width of the search region. N_x is the patch number of search region features under the patch size of P; N_z is the patch number of template region features similarly. d is the channel dimension of embedded features. Then, two learnable positional embeddings, $P_x \in \mathbb{R}^{N_x*d}$ and $P_z \in \mathbb{R}^{N_z*d}$, are concatenated and added to the sequence of patch embeddings F_{xz}:

$$F_{xz}^{'} = F_{xz} + concat(P_x, P_z) \tag{1}$$

Finally, patch embeddings $F_{xz}^{'}$ will be fed into the encoder composed of transformer layers for representation learning and relation modeling. The i-th block of the transformer layers can be written as follows:

$$\begin{aligned} F_{xz}^{i} &= Attention(LayerNorm(F_{xz}^{i})) + F_{xz}^{i} \\ F_{xz}^{i+1} &= MLP(LayerNorm(F_{xz}^{i})) + F_{xz}^{i} \end{aligned} \tag{2}$$

Finally, the search region part of the last layer's output of features will be sent to the tracking head, which is used for target bounding box estimation. In general, the tracking head is the only part for random initialization rather than initialization inherited from pre-trained models.

3.2. Adapter Tuning

In most existing trackers, such as OSTrack [10] and KeepTrack [37], the full fine-tuning method is utilized for training. All the parameters are tuned, resulting in high costs during training. The adapter tuning technique can achieve equally effective training to full fine-tuning by training only a small number of parameters and freezing the remaining model parameters. There are many forms of adapter applied [25,26,32]. As illustrated in Figure 1, adapter tuning in our work is a bottleneck module for residual addition after MLP (Multi-layer Perceptron) operation in every transformer block. That bottleneck module includes two light MLP layers, one for downward projection and the other for upward projection. Every light MLP layer contains two linear projections which are connected by

an activation function, GELU. With F as the input, our inserted adapter module can be expressed as follows:

$$\begin{aligned} F_{down} &= GELU(FW_{down})W_1 \\ F_{up} &= GELU(F_{down}W_2)W_{up} \\ F_{adapter} &= F_{up} + F \end{aligned} \tag{3}$$

$W_2 \in \mathbb{R}^{k*k}, W_{up} \in \mathbb{R}^{k*d}, W_1 \in \mathbb{R}^{k*k}, W_{down} \in \mathbb{R}^{d*k}.$

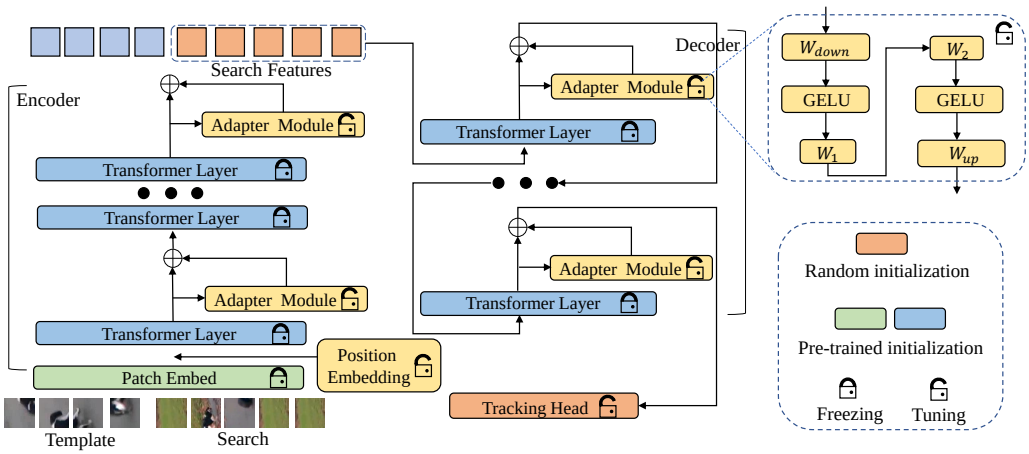

Figure 1. Details of our tracking framework with parameter-efficient transfer learning. "…" represents the several omitted transformer layers which are the same as others.

In our work, F represents the output of every transformer block; then, $F_{adapter}$ is the output of every transformer block after adjusting with the adapter module. d is the channel size of the transformer backbone and k is the compressed size, $k < d$. The details are also illustrated in Figure 1. Only a very small number of parameters is introduced in that module. By tuning these parameters, the pre-training representations of the pretext task can be well transferred to the downstream tracking task.

3.3. Migrating Pre-Trained Decoders

Motivation. Recently, some works have pointed out that discarding part of the pre-trained network may cause information loss [38] and taking full advantage of pre-trained models can contribute to improving the generalization of detectors [13] in the MAE pre-training method. To verify whether the pre-trained decoder contributes to downstream tracking tasks, we designed a set of comparative experiments. As illustrated in Figure 2, we insert the decoder module in the pre-training stage for predicting the image pixels between the encoder and tracking head. Only the features of the search region are imported into the decoder, so it can be regarded as an enhanced part of the tracking head. We adopt the corner head [19] without output embedding weighting as the tracking head for all variants. For variants with a decoder structure, the tracking head is reduced to only two convolution layers for the top-left and bottom-right score maps' estimation. We also set a variant of random decoder for comparing the effectiveness of the pre-trained weights, as in Figure 2c. Experiments are conducted on LaSOT [39] and LaSOT$_{ext}$ [40]. Full fine-tuning is employed for all and the training settings are mostly the same as for OSTrack [10]. Details are in Section 4.1.

As shown in Table 1, with the pre-trained decoder as the main part of the tracking head, just two additional convolution layers random initialized can achieve an even better performance than the baseline tracker. However, the decoder which is random initialized shows a decline in performance especially for the LaSOT$_{ext}$ benchmark. We think that the pre-trained decoder module indeed has benefits for tracking.

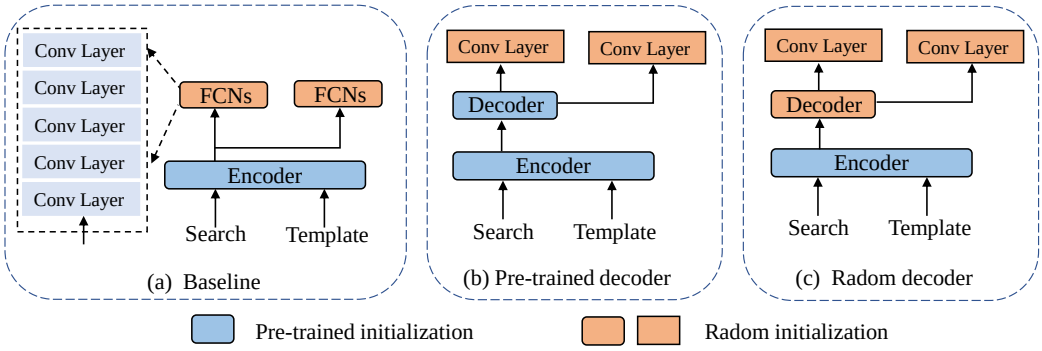

Figure 2. Three variants of trackers for performance effects of pre-trained decoder.

Table 1. Comparison of three variants for exploration of pre-trained decoder. AUC scores are reported.

Tracker	LaSOT	Improvement	$\text{LaSOT}_{\text{ext}}$	Improvement
Baseline	68.9	-	48.7	-
Pre-trained Decoder	69.3	+0.4%	48.5	−0.2%
Random Decoder	68.6	−0.3%	47.5	−1.2%

Decoder with Adapter Tuning. To taking full advantage of the pre-trained models, we apply the adapter tuning in decoder blocks as in Section 3.2 as well. Decoder blocks are expected to enhance the generalization ability and robustness when transferring to tracking.

4. Experiments

4.1. Implementation Details

Our models are trained with the AdamW optimizer [41] on two NVIDIA A100 GPUs (NVIDIA, Santa Clara, CA, USA), with each GPU holding a batchsize of 64. Training data contain training slices of COCO-2017 [42] and LaSOT [39]. Aligned with most trackers, data augmentations contains brightness jitter and horizontal flip. A total of 300 epochs are conducted with 6×10^4 template–search pairs each epoch, and each template–search pair is sampled among all the frames of a sequence. The learning rate decreases by a factor of 0.1 after the first 240 epochs. The learning rate is set as 2×10^{-4} for both the encoder and decoder and 2×10^{-5} for the remaining parameters. The compressed dimension k of the adapter structure is 128.

We present two variants for our model: **B256** and **L256**. For **B256**, we adopt the ViT-Base model pre-trained with MAE [2]; for **L256**, we adopt the ViT-Large model pre-trained with MAE [2]. The ViT-Base model contains 12 layers in the encoder part and 8 layers in the decoder part, with an embedding dimension of 768. The ViT-Large model contains 24 layers in the encoder part and 8 layers in the decoder part, with an embedding dimension of 1024. For the **B256** and **L256** variants, the search region is resized to 256×256 pixels and cropped with 4^2 times the region of the target size; the template region is resized to 128×128 pixels and cropped with 2^2 times the region of the target size. The channel dimension k of the adapter module is 128 for all variants of model.

4.2. Mainstream Benchmarks

LaSOT [39]. A total of 280 videos constitute the LaSOT testing dataset. That benchmark is popular with its large scale and relatively long average duration. As reported in Table 2, our AdaDe-B256 achieves a 68.7 AUC score, which does not reflect a performance improvement or advantage, while AdaDe-L256 achieves a 71.2 AUC score, which exceeds the corresponding baseline tracker with a 3.1% performance increase. For larger pre-trained models, adapter tuning shows better performance advantages compared with full fine-tuning. That phenomenon may indicate that larger models are more prone to weaken in terms of feature representation ability during downstream fine-tuning training compared to adapter tuning.

LaSOT$_{ext}$ [40]. The LaSOT$_{ext}$ benchmark is an expansion of LaSOT, and consists of 150 videos among 15 categories. It adopts the one-shot evaluation protocol, which ensures that there is no intersection between categories in the testing dataset with the categories in the LaSOT's training dataset. Rather than the MATLAB evaluation toolkit, we take the Python toolkit available from OSTrack [10] for a fair comparison with other trackers' reported results. Our AdaDe-B256 achieves a 49.5 AUC score and AdaDe-L256 achieves a 50.3 AUC score. Compared with the baseline tracker, our trackers also outperform by 0.8% and 1.3% performance increases, respectively.

TrackingNet [43]. The TrackingNet benchmark is a massive short-term tracking benchmark that includes 511 videos for evaluation. From Table 2, we can also see that our tracker works well compared with other methods. On the basis of the adapter tuning of the pre-trained backbones, we also migrate pre-trained transformer decoders into MAE pre-training to enhance the tracking head. This technique can enhance the expressive power of the model to a certain extent, resulting in improved tracking accuracy. The proposed method can work well in both the long-term and short-term tracking datasets.

UAV123 [44], NFS [45] and TNL2K [46]. UAV123 and NFS are two small benchmarks with 123 videos captured by drones and 100 videos, respectively. TNL2K is a recently launched massive tracking benchmark containing 700 linguistically labeled sequences. For TNL2K, we provide the evaluation results of the Python toolkit. The experimental results on these challenging benchmarks show that the proposed method achieves a competitive performance compared to many state-of-the-art algorithms (shown in Table 3).

Table 2. Comparison with the state of the art on the LaSOT [39], LaSOT$_{ext}$ [40] and TrackingNet [43] benchmarks. The best performance is in bold.

Tracker	LaSOT			LaSOT_ext			TrackingNet		
	AUC	P_norm	P	AUC	P_norm	P	AUC	P_norm	P
AdaDe-L256	71.2 (+3.1)	80.9	78.6	**50.3** (+1.3)	60.6	57.3	84.3	88.8	83.8
Baseline-L256	68.1	76.5	73.8	49.0	58.6	56.1	83.8	88.2	83.3
AdaDe-B256	68.7 (−0.2)	78.5	74.6	49.5 (+0.8)	59.9	56.0	82.0	86.6	80.1
Baseline-B256	68.9	78.1	74.5	48.7	58.7	55.3	82.8	87.2	81.2
LoRAT-B-224 [47]	**72.4**	81.6	77.9	48.5	61.7	55.3	83.7	88.2	82.2
GRM-L320 [34]	71.4	81.2	77.9	-	-	-	**84.4**	88.9	84.0
GRM-B256 [34]	69.9	79.3	75.8	-	-	-	84.0	88.7	83.3
OSTrack-B384 [10]	71.1	81.1	77.6	50.5	61.3	57.6	83.9	88.5	83.2
OSTrack-B256 [10]	69.1	78.7	75.2	47.4	57.3	53.3	83.1	87.8	82.0
SimTrack-L [9]	70.5	79.7	-	-	-	-	83.4	87.4	-
SimTrack-B [9]	69.3	78.5	-	-	-	-	82.3	86.5	-
SwinTrack-B [22]	69.6	78.6	74.1	47.6	58.2	54.1	82.5	87.0	80.4
Mixformer-L [48]	70.1	79.9	76.3	-	-	-	83.9	88.9	
MixFormer-22k [48]	69.2	78.7	74.7	-	-	-	83.1	88.1	81.6
AiATrack [49]	69.0	79.4	73.8	47.7	55.6	55.4	82.7	87.8	80.4
ToMP-101 [50]	68.5	-	-	45.9	-	-	81.5	86.4	78.9
GTELT [51]	67.7	-	73.2	45.0	54.2	52.4	82.5	86.7	81.6
KeepTrack [37]	67.1	77.2	70.2	48.2	58.1	56.4	-	-	-
STARK-101 [19]	67.1	77.0	-	-	-	-	82.0	86.9	-
TransT [20]	64.9	73.8	69.0	-	-	-	81.4	86.7	80.3
SiamR-CNN [52]	64.8	72.2	-	-	-	-	81.2	85.4	80.0
TrDiMP [21]	63.9	-	61.4	-	-	-	78.4	83.3	73.1
LTMU [53]	57.2	-	57.2	41.4	49.9	47.3	-	-	-
DiMP [54]	56.9	65.0	56.7	39.2	47.6	45.1	74.0	80.1	68.7
SiamPRN++ [55]	49.6	56.9	49.1	34.0	41.6	39.6	73.3	80.0	69.4
SiamFC [56]	33.6	42.0	33.9	23.0	31.1	26.9	57.1	66.3	53.3

Table 3. Comparison with the state-of-the-art trackers on the UAV123, NFS and TNL2K benchmarks in AUC scores.

Tracker	UAV123	NFS	TNL2K
AdaDe-L256	69.9	67.5	59.1
AdaDe-B256	68.5	67.3	56.2
GRM-L320 [34]	72.2	66.0	-
GRM-B256 [34]	70.2	65.6	-
OSTrack-384 [10]	70.7	66.5	55.9
OSTrack-256 [10]	68.3	64.7	54.3
MixFormer-22k [48]	70.4	-	-
KeepTrack [37]	69.7	66.4	-
STARK-101 [19]	68.2	66.2	-
TransT [20]	68.1	65.3	50.7
TrDiMP [21]	66.4	66.2	-
SiamR-CNN [52]	64.9	63.9	-
SiamPRN++ [55]	59.3	57.1	-

4.3. Other Benchmarks

To further evaluate the robustness and generalization of the proposed tracker, we select four challenging benchmarks focusing on special and challenging scenarios or targets: AVisT [57] with adverse visibility scenarios, UAVDark135 [58] and DarkTrack2021 [59] with night-time scenarios, and TOTB [60] with transparent objects. These scenarios or targets rarely emerge in most tracking datasets, leading to domain gaps and difficulty in tracking.

AVisT [57] is a recently released tracking benchmark with diverse scenarios and adverse visibility, covering severe weather conditions, obstruction effects such as water splashing and unfavorable imaging effects such as low light, distractor objects or camouflage. That benchmark consist of 120 sequences and presents great tracking challenges under complicated conditions.

UAVDark135 [58] and DarkTrack2021 [59] are two night-time tracking benchmarks, containing 135 and 115 videos, respectively. The results of all comparisons are reported in Tables 4 and 5.

For the adverse visibility tracking dataset AVisT, our AdaDe-B256's performance only outperforms GRM-B256 [34] with a 0.1% increase, while AdaDe-L256 outperforms AdaDe-B256 with a 5.3% performance increase. Compared to other state-of-the-art trackers, AdaDe-L256 achieves a 2.3% gain over OSTrack-B384 [10] and 4.8% gain over GRM-L320 [34]. This shows the excellent performance of our trackers.

For two night-time datasets, our AdaDe-B256's performance is basically on par with OSTrack-B256 [10]. Similar to previous conclusions on the AVisT benchmark, performance on these night-time datasets benefits a lot from larger pre-trained weights, indicating that a larger model can indeed bring stronger generalization ability. The AdaDe-L256 tracker exceeds the AdaDe-B256 tracker with a 4.8% increase on the UAVDark135 dataset and a 4.1% increase on the DarkTrack2021 dataset. Although no specific night-time data are utilized for training additionally in our method, we also compare our tracker with DCPT [61] and DiMP-SCT [59] which employ night-time datasets for training. DCPT [61] is a recent night tracker which is designed with prompt tuning with night-time data of BDD100K [62] and SHIFT [63]. DiMP-SCT [59] learns a lowlight enhancer module with paired low/normal light images from the LOL [64] dataset. The performance of the AdaDe-L256 tracker also exceeds DCPT [61] by 2.0% on the UAVDark135 dataset and is only 0.7% behind on the DarkTrack2021 dataset. These comparative results show our better generalization ability in tracking. More detailed analysis on the generalization ability will be unfolded in ablation studies.

Table 4. Comparison with the state-of-the-art trackers on AVisT.

Tracker	AVisT		
	AUC	OP50	OP75
AdaDe-L256	59.9	69.6	52.1
AdaDe-B256	54.6	63.3	44.5
OSTrack-B384 [10]	58.1	67.9	48.6
OSTrack-B256 [10]	56.2	65.3	46.5
MixFormerL-22k [48]	56.0	65.9	46.3
MixFormer-22k [48]	53.7	63.0	43.0
GRM-L320 [34]	55.1	63.8	46.9
GRM-B256 [34]	54.5	63.1	45.2
STARK-101 [19]	50.5	58.23	39.0
KeepTrack [37]	49.4	56.3	37.2
TransT [20]	49.0	56.4	37.2
TrDiMP [21]	48.1	55.3	33.8
SiamPRN++ [55]	39.0	43.5	21.2
DiMP [54]	38.6	41.5	22.2

Table 5. Comparison with the state-of-the-art trackers on the UAVDark135 and DarkTrack2021 benchmarks. The symbol * means that training of the tracker involves additional night-time data. We re-evaluate the performance of DCPT [61] with a Python toolkit.

Tracker	UAVDark135		DarkTrack2021	
	AUC	P_norm	AUC	P_norm
AdaDe-L256	59.7	72.0	53.3	63.9
AdaDe-B256	54.9	67.3	49.2	58.5
DCPT * [61]	57.7	70.1	54.0	64.6
DiMP-SCT * [59]	56.2	71.7	52.1	67.7
OSTrack-B256 [10]	55.1	66.2	49.1	60.1
PrDiMP [65]	50.7	63.8	46.4	58.0
DiMP18 [54]	49.4	63.9	47.1	62.0

4.4. Visualizations

We visualize some cases of our tracker and the state-of-art trackers for comparison, as shown in Figure 3. Our tracker performs better than OSTrack-B256 [10] in some cases.

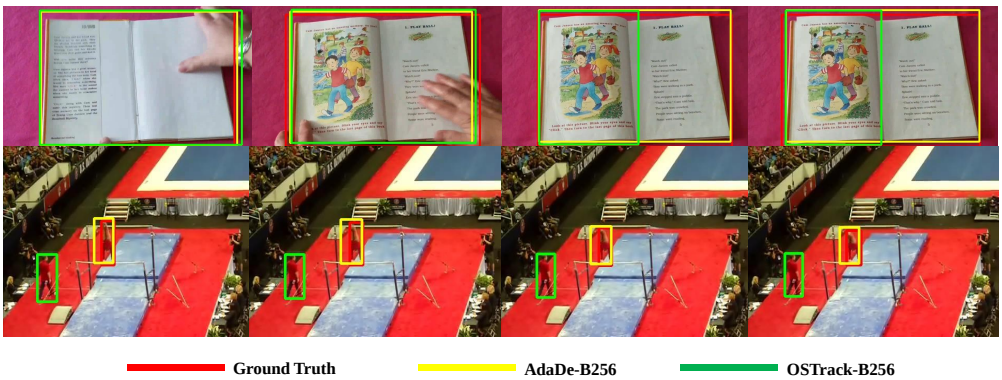

Figure 3. Visualization cases of our AdaDe-B256 tracker and other trackers.

4.5. Effect of Decoder

We evaluate the effects of the decoder part in Table 6. For the B256 variant, adding the decoder brings a performance improvement on almost all tracking benchmarks. The performance improves by 0.4% on LaSOT and 0.8% on $\text{LaSOT}_{\text{ext}}$. For the night-time tracking benchmarks, a 1.7% performance increase is achieved on UAVDark135 and a 1.9% performance increase is achieved on DarkTrack2021. For the L256 variant, there is no significant

fluctuation in performance on LaSOT and $\text{LaSOT}_{\text{ext}}$ with the addition of the decoder, while a 1.5% performance increase is achieved on AVisT and a 1.6% performance increase is achieved on UAVDark135. These results are consistent with the B256 variant. The significant performance increase on these challenging benchmarks indicates that the decoder promotes the robustness and generalization of tracking.

Table 6. Ablation studies of generalization analysis. AUC scores are reported. The uparrow represents the improvement and the downarrow represents the decrease.

Variant	Method	LaSOT	$\text{LaSOT}_{\text{ext}}$	AVisT	UAVDark135	DarkTrack2021
L256	+ Adapter Module	71.2	50.6	58.4	58.1	53.8
	+ Adapter Module + Decoder	71.2 (0.0 ↑)	50.3 (0.3 ↓)	59.9 (1.5 ↑)	59.7 (1.6 ↑)	53.3 (0.5 ↓)
B256	+ Adapter Module	68.3	48.7	54.9	53.2	47.3
	+ Adapter Module + Decoder	68.7 (0.4 ↑)	49.5 (0.8 ↑)	54.6 (0.3 ↓)	54.9 (1.7 ↑)	49.2 (1.9 ↑)

4.6. Limitations

Although the proposed method achieves a comparable performance on most tracking benchmarks with the state-of-the-art trackers, it still has some limitations. First, although there may be different impacts due to the distribution of different datasets, the introduction of the pre-trained decoder does not bring a completely stable improvement on all datasets. Secondly, the introduction of the adapter tuning brings more computation costs during inference. An inference-friendly parameter-efficient tuning may be considered and explored.

5. Conclusions

In this work, we present a novel approach to parameter efficiency tuning by migrating pre-trained decoders, to design an effective tracking method. First, we freeze the pre-trained weights of the backbone and then add a dynamic adapter structure for every transformer block for tuning, which makes online fine-tuning effective. Second, we migrate the pre-trained decoder blocks to the tracking head, which is very suitable for object localization. The experimental results demonstrate that the presented tracking algorithm achieves better properties than other competing methods on many challenging tracking benchmarks.

Author Contributions: Conceptualization, R.Z. and S.Y.; Methodology, R.Z.; Software, R.Z.; Validation, L.W.; Investigation, R.Z.; Data curation, R.Z., L.W. and S.Y.; Writing—original draft, R.Z. and S.Y.; Writing—review & editing, R.Z., L.W. and S.Y.; Funding acquisition, R.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Science Foundation of Jiang Su Higher Education Institutions, grant number 24KJD510005 and Jiang Su Province Industry-University Research Project, grant number BY20230694.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning transferable visual models from natural language supervision. In Proceedings of the 38th International Conference on Machine Learning (ICML 2021), Virtual, 18–24 July 2021; pp. 8748–8763.
2. He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; Girshick, R. Masked autoencoders are scalable vision learners. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 15979–15988.
3. Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A.; et al. DINOv2: Learning Robust Visual Features without Supervision. *arXiv* **2023**, arXiv:2304.07193.
4. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.

5. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1106–1114.
6. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
7. Yan, W.; Sun, Y.; Yue, G.; Zhou, W.; Liu, H. FVIFormer: Flow-Guided Global-Local Aggregation Transformer Network for Video Inpainting. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2024**, *14*, 235–244. [CrossRef]
8. Marin, D.; Chang, J.R.; Ranjan, A.; Prabhu, A.; Rastegari, M.; Tuzel, O. Token Pooling in Vision Transformers for Image Classification. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 2–7 January 2023; pp. 12–21.
9. Chen, B.; Li, P.; Bai, L.; Qiao, L.; Shen, Q.; Li, B.; Gan, W.; Wu, W.; Ouyang, W. Backbone is All Your Need: A Simplified Architecture for Visual Object Tracking. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; pp. 375–392.
10. Ye, B.; Chang, H.; Ma, B.; Shan, S.; Chen, X. Joint feature learning and relation modeling for tracking: A one-stream framework. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; pp. 341–357.
11. Wu, Q.; Yang, T.; Liu, Z.; Wu, B.; Shan, Y.; Chan, A.B. Dropmae: Masked autoencoders with spatial-attention dropout for tracking tasks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 14561–14571.
12. Zhao, H.; Wang, D.; Lu, H. Representation Learning for Visual Object Tracking by Masked Appearance Transfer. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 18696–18705.
13. Liu, F.; Zhang, X.; Peng, Z.; Guo, Z.; Wan, F.; Ji, X.; Ye, Q. Integrally Migrating Pre-trained Transformer Encoder-decoders for Visual Object Detection. In Proceedings of the IEEE International Conference on Computer Vision, Paris, France, 1–6 October 2023; pp. 6802–6811.
14. Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.T.; Upcroft, B. Simple online and realtime tracking. In Proceedings of the IEEE International Conference on Image Processing, Phoenix, AZ, USA, 25–28 September 2016; pp. 3464–3468.
15. Ma, L.V.; Nguyen, T.T.D.; Shim, C.; Kim, D.Y.; Ha, N.; Jeon, M. Visual multi-object tracking with re-identification and occlusion handling using labeled random finite sets. *Pattern Recognit.* **2024**, *156*, 110785.
16. Zhu, T.; Hiller, M.; Ehsanpour, M.; Ma, R.; Drummond, T.; Reid, I.D.; Rezatofighi, H. Looking Beyond Two Frames: End-to-End Multi-Object Tracking Using Spatial and Temporal Transformers. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 12783–12797. [CrossRef] [PubMed]
17. Zhang, Y.; Sun, P.; Jiang, Y.; Yu, D.; Yuan, Z.; Luo, P.; Liu, W.; Wang, X. ByteTrack: Multi-Object Tracking by Associating Every Detection Box. *arXiv* **2021**, arXiv:2110.06864.
18. Luo, W.; Xing, J.; Milan, A.; Zhang, X.; Liu, W.; Kim, T. Multiple object tracking: A literature review. *Artif. Intell.* **2021**, *293*, 103448. [CrossRef]
19. Yan, B.; Peng, H.; Fu, J.; Wang, D.; Lu, H. Learning spatio-temporal transformer for visual tracking. In Proceedings of the IEEE International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 10448–10457.
20. Chen, X.; Yan, B.; Zhu, J.; Wang, D.; Yang, X.; Lu, H. Transformer tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 8126–8135.
21. Wang, N.; Zhou, W.; Wang, J.; Li, H. Transformer Meets Tracker: Exploiting Temporal Context for Robust Visual Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 1571–1580.
22. Lin, L.; Fan, H.; Xu, Y.; Ling, H. Swintrack: A simple and strong baseline for transformer tracking. *arXiv* **2021**, arXiv:2112.00995.
23. Lester, B.; Al-Rfou, R.; Constant, N. The Power of Scale for Parameter-Efficient Prompt Tuning. In Proceedings of the EMNLP, Punta Cana, Dominican Republic, 7–11 November 2021; pp. 3045–3059.
24. Jia, M.; Tang, L.; Chen, B.; Cardie, C.; Belongie, S.J.; Hariharan, B.; Lim, S. Visual Prompt Tuning. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; Volume 13693, pp. 709–727.
25. Chen, S.; Ge, C.; Tong, Z.; Wang, J.; Song, Y.; Wang, J.; Luo, P. AdaptFormer: Adapting Vision Transformers for Scalable Visual Recognition. In Proceedings of the Advances in Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022.
26. Pfeiffer, J.; Kamath, A.; Rücklé, A.; Cho, K.; Gurevych, I. AdapterFusion: Non-Destructive Task Composition for Transfer Learning. In Proceedings of the EACL, Online, 19–23 April 2021; pp. 487–503.
27. Xin, Y.; Du, J.; Wang, Q.; Lin, Z.; Yan, K. VMT-Adapter: Parameter-Efficient Transfer Learning for Multi-Task Dense Scene Understanding. *arXiv* **2023**, arXiv:2312.08733. [CrossRef]
28. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. In Proceedings of the International Conference on Learning Representation, Online, 25–29 April 2022.
29. Yang, J.; Li, Z.; Zheng, F.; Leonardis, A.; Song, J. Prompting for multi-modal tracking. In Proceedings of the ACMMM, Lisbon, Portugal, 10–14 October 2022; pp. 3492–3500.
30. Zhu, J.; Lai, S.; Chen, X.; Wang, D.; Lu, H. Visual Prompt Multi-Modal Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 9516–9526.

31. Wu, Z.; Zheng, J.; Ren, X.; Vasluianu, F.; Ma, C.; Paudel, D.P.; Gool, L.V.; Timofte, R. Single-Model and Any-Modality for Video Object Tracking. *arXiv* **2023**, arXiv:2311.15851.
32. Cao, B.; Guo, J.; Zhu, P.; Hu, Q. Bi-directional Adapter for Multi-modal Tracking. *arXiv* **2023**, arXiv:2312.10611.
33. Hou, X.; Xing, J.; Qian, Y.; Guo, Y.; Xin, S.; Chen, J.; Tang, K.; Wang, M.; Jiang, Z.; Liu, L.; et al. SDSTrack: Self-Distillation Symmetric Adapter Learning for Multi-Modal Visual Object Tracking. *arXiv* **2024**, arXiv:2403.16002.
34. Gao, S.; Zhou, C.; Zhang, J. Generalized relation modeling for transformer tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 18686–18695.
35. Chen, X.; Peng, H.; Wang, D.; Lu, H.; Hu, H. SeqTrack: Sequence to Sequence Learning for Visual Object Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 14572–14581.
36. Song, Z.; Luo, R.; Yu, J.; Chen, Y.P.P.; Yang, W. Compact transformer tracker with correlative masked modeling. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 2–14 February 2023; Volume 37, pp. 2321–2329.
37. Mayer, C.; Danelljan, M.; Paudel, D.P.; Van Gool, L. Learning target candidate association to keep track of what not to track. In Proceedings of the IEEE International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 13444–13454.
38. Han, Q.; Cai, Y.; Zhang, X. RevColV2: Exploring Disentangled Representations in Masked Image Modeling. In Proceedings of the Advances in Neural Information Processing Systems, New Orleans, LA, USA, 10–16 December 2023;
39. Fan, H.; Lin, L.; Yang, F.; Chu, P.; Deng, G.; Yu, S.; Bai, H.; Xu, Y.; Liao, C.; Ling, H. LaSOT: A high-quality benchmark for large-scale single object tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5374–5383.
40. Fan, H.; Bai, H.; Lin, L.; Yang, F.; Ling, H. LaSOT: A High-quality Large-scale Single Object Tracking Benchmark. *Int. J. Comput. Vis.* **2021**, *129*, 439–461. [CrossRef]
41. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv* **2017**, arXiv:1711.05101.
42. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
43. Muller, M.; Bibi, A.; Giancola, S.; Alsubaihi, S.; Ghanem, B. TrackingNet: A large-scale dataset and benchmark for object tracking in the wild. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 300–317.
44. Mueller, M.; Smith, N.; Ghanem, B. A benchmark and simulator for UAV tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 445–461.
45. Galoogahi, H.K.; Fagg, A.; Huang, C.; Ramanan, D.; Lucey, S. Need for Speed: A Benchmark for Higher Frame Rate Object Tracking. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1134–1143.
46. Wang, X.; Shu, X.; Zhang, Z.; Jiang, B.; Wang, Y.; Tian, Y.; Wu, F. Towards More Flexible and Accurate Object Tracking With Natural Language: Algorithms and Benchmark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13763–13773.
47. Lin, L.; Fan, H.; Zhang, Z.; Wang, Y.; Xu, Y.; Ling, H. Tracking Meets LoRA: Faster Training, Larger Model, Stronger Performance. In Proceedings of the European Conference on Computer Vision, Milan, Italy, 29 September–4 October 2024; Volume 15059, pp. 300–318.
48. Cui, Y.; Jiang, C.; Wang, L.; Wu, G. MixFormer: End-to-End Tracking with Iterative Mixed Attention. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 13608–13618.
49. Gao, S.; Zhou, C.; Ma, C.; Wang, X.; Yuan, J. Aiatrack: Attention in attention for transformer visual tracking. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; pp. 146–164.
50. Mayer, C.; Danelljan, M.; Bhat, G.; Paul, M.; Paudel, D.P.; Yu, F.; Van Gool, L. Transforming model prediction for tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 8731–8740.
51. Zhou, Z.; Chen, J.; Pei, W.; Mao, K.; Wang, H.; He, Z. Global tracking via ensemble of local trackers. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 8761–8770.
52. Voigtlaender, P.; Luiten, J.; Torr, P.H.; Leibe, B. Siam R-CNN: Visual Tracking by Re-Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6578–6588.
53. Dai, K.; Zhang, Y.; Wang, D.; Li, J.; Lu, H.; Yang, X. High-Performance Long-Term Tracking With Meta-Updater. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6298–6307.
54. Bhat, G.; Danelljan, M.; Gool, L.V.; Timofte, R. Learning discriminative model prediction for tracking. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6182–6191.
55. Li, B.; Wu, W.; Wang, Q.; Zhang, F.; Xing, J.; Yan, J. SiamRPN++: Evolution of siamese visual tracking with very deep networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4282–4291.
56. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H.S. Fully-Convolutional Siamese Networks for Object Tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 850–865.

57. Noman, M.; Ghallabi, W.A.; Najiha, D.; Mayer, C.; Dudhane, A.; Danelljan, M.; Cholakkal, H.; Khan, S.; Gool, L.V.; Khan, F.S. AVisT: A Benchmark for Visual Object Tracking in Adverse Visibility. In Proceedings of the British Machine Vision Conference, London, UK, 21–24 November 2022.
58. Li, B.; Fu, C.; Ding, F.; Ye, J.; Lin, F. All-Day Object Tracking for Unmanned Aerial Vehicle. *IEEE Trans. Mob. Comput.* **2023**, *22*, 4515–4529. [CrossRef]
59. Ye, J.; Fu, C.; Cao, Z.; An, S.; Zheng, G.; Li, B. Tracker Meets Night: A Transformer Enhancer for UAV Tracking. *IEEE Robot. Autom. Lett.* **2022**, *7*, 3866–3873. [CrossRef]
60. Fan, H.; Miththanthaya, H.A.; Harshit; Rajan, S.R.; Liu, X.; Zou, Z.; Lin, Y.; Ling, H. Transparent Object Tracking Benchmark. In Proceedings of the IEEE International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 10714–10723.
61. Zhu, J.; Tang, H.; Cheng, Z.; He, J.; Luo, B.; Qiu, S.; Li, S.; Lu, H. DCPT: Darkness Clue-Prompted Tracking in Nighttime UAVs. *arXiv* **2023**, arXiv:2309.10491.
62. Yu, F.; Chen, H.; Wang, X.; Xian, W.; Chen, Y.; Liu, F.; Madhavan, V.; Darrell, T. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 3–19 June 2020; pp. 2633–2642.
63. Sun, T.; Segù, M.; Postels, J.; Wang, Y.; Gool, L.V.; Schiele, B.; Tombari, F.; Yu, F. SHIFT: A Synthetic Driving Dataset for Continuous Multi-Task Domain Adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 21339–21350.
64. Wei, C.; Wang, W.; Yang, W.; Liu, J. Deep Retinex Decomposition for Low-Light Enhancement. In Proceedings of the British Machine Vision Conference, Newcastle, UK, 3–6 September 2018; p. 155.
65. Danelljan, M.; Gool, L.V.; Timofte, R. Probabilistic Regression for Visual Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 7183–7192.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.