Drift-Resilient Temporal Priors for Visual Tracking

Supplementary Material

In this supplementary material, we provide more implementation details and visualizations for the proposed DTPTrack.

A. Additional Implementation Details

Our experimental setup largely aligns with the foundational configurations and hyperparameters established in LoRAT [31]. A summary of the essential hyperparameters, including both the training and inference phases, is provided in Tab. 8.

Table 8. Hyper-parameters used in DTPTrack.

Item	Value
template area factor	2
search region area factor	4 (B-224) / 5 (L-378)
scale jitter	0.25
translation jitter	3
horizontal flip	0.5
color jitter	0.4
batch size	128
epochs	170 / 100 (GOT-10k)
optimizer	AdamW
lr	1e-4
weight decay	0.1
drop path	0.0 (B) / 0.1 (L)
clip max norm	1.0
lr_min	5e-6
warmup epochs	2
warmup lr mult	1e-3
BCE loss coef	1.0
GIoU loss coef	1.0
Hann window penalty	0.45
LoRA rank r	64

A.1. Model Configuration

We adopt standard ViT-B/14 and ViT-L/14 architectures [16, 38] as backbones for our B-224 and L-378 models, respectively, initializing them with DINOv2 pre-trained weights [15, 38]. While the backbone parameters remain frozen, we introduce Low-Rank Adaptation (LoRA) [22] with a rank of $r = 64$ to all linear projection matrices within the attention and MLP blocks for efficient fine-tuning. Consistent with the methodology of LoRAT [31], input images are tokenized using a patch size of 14 with shared positional embeddings, and the final prediction is handled by two separate 3-layer Multi-Layer Perceptrons (MLPs) for bounding box regression and classification.

A.2. Training Details

Loss. The overall loss is a sum of a Binary Cross-Entropy (BCE) loss for classification and a GIoU loss [41] for bounding box regression. We assign equal importance to both terms, setting their coefficients to 1.0.

Data Augmentation. We employ the common data augmentation pipeline [31] on the video clips, including random horizontal flipping (probability 0.5), color jittering (e.g., brightness, contrast, saturation), and the 3-Augment strategy [42]. The template images are cropped with a 2× area factor around the initial bounding box or predicted boxes. Search regions are cropped with an area factor of 4 (for B-224) or 5 (for L-378). Scale jitter and translation jitter are also applied to both versions.

Optimization. Models are trained for 170 epochs (100 for GOT-10k) using AdamW [34] with a batch size of 128. We use an initial learning rate of 1×10^{-4}, a weight decay of 0.1 (excluding bias/norm), and gradient clipping at 1.0. The learning rate follows a cosine schedule decaying to 5×10^{-6} after a 2-epoch linear warmup from 1×10^{-7}. For DTPTrack-L, a DropPath rate of 0.1 is applied [28]. To maximize computation efficiency, we utilize Automatic Mixed Precision (AMP) and memory-efficient attention [40].

A.3. Inference Details

During inference, a Hann window penalty is applied to the classification score map. The penalty coefficient is 0.45. We employ optimized fused GPU kernels, such as those provided by FlashAttention [13, 14], to reduce the latency incurred by attention-related memory operations and yield inference speeds (FPS) that more closely align with the theoretical FLOPs.

B. More Experimental Results

In this section, we provide additional experimental results and detailed ablation studies to further demonstrate the effectiveness of our proposed DTPTrack.

B.1. Attribute-wise Performance Analysis

To understand how DTPTrack mitigates tracking drift, we evaluate the performance across different video attributes

on the LaSOT benchmark. As shown in Table 9, DTPTrack achieves consistent improvements across all categories, with the most significant gains observed in *Scale Variation (SV)* (+2.3%) and *Aspect Ratio Change (ARC)* (+1.0%), confirming that our temporal priors effectively handle geometric variations that typically lead to drift.

Table 9. Attribute-wise AUC (%) comparison on the LaSOT.

Method	SV	ARC	ROT	POC	FOC	DEF	BC	LR
Baseline	71.8	71.8	73.0	70.8	65.6	74.7	66.8	66.8
+ DTPTrack	**74.1**	**72.8**	**73.5**	**71.5**	**66.7**	**75.0**	**68.2**	**68.4**

B.2. Drift Rate Error (DRE) on DiDi

We further quantify the drift reduction using the Drift Rate Error (DRE) metric on the DiDi dataset. As reported in Table 10, DTPTrack achieves a lower DRE compared to the baseline, indicating more stable long-term tracking performance.

Table 10. Comparison of tracking stability and DRE on the DiDi dataset.

Method	Qual.	Acc.	Rob.	**DRE** ↓
Baseline	0.668	0.742	0.809	0.100
+ DTPTrack	**0.680**	**0.760**	**0.840**	**0.080**

B.3. Additional Comparisons on VOT Benchmarks

To further validate the generalization and robustness of DTPTrack, we conduct additional experiments on the VOT-STB2022 [26] and VOT2024 [27] benchmarks. We focus on bounding box (BBox) tracking to ensure a fair comparison regarding input resolution, MACs, and supervision level.

Table 11. EAO performance comparison on VOT-STB2022 and VOT2024 benchmarks.

Benchmark	Method	Backbone	EAO ↑
VOT-STB2022	MixFormer-L	ViT-L	0.582
	DTPTrack-B$_{224}$	ViT-B	**0.610**
VOTS2024	LoRAT-g	ViT-L	0.536
	DTPTrack-L$_{378}$	ViT-L	**0.630**

As shown in Table 11, our DTPTrack-B$_{224}$ outperforms recent state-of-the-art methods like MixFormerL on the VOT-STB2022 benchmark with a smaller backbone. Furthermore, on the challenging VOTS2024 benchmark, our approach achieves a competitive EAO of 0.630, significantly surpassing previous BBox-based trackers like LoRAT-g.

Figure 3. Visualization of the gating mechanism. The model assigns lower scores (indicated by darker colors or lower magnitude) to frames with artifacts or occlusions.

C. Visualization

C.1. Visualization of Gating Mechanism

Figure 3 visualizes the learned gating values. It demonstrates that the model effectively assigns lower weights to frames with occlusion or low-quality visual features, successfully preventing noise from corrupting the temporal prior.

C.2. Visualization of Challenging Sequences

In Fig. 4, we provide qualitative comparisons between DTPTrack and three strong baselines (OSTrack, ODTrack, and LoRAT) on four challenging sequences. Green boxes denote ground-truth, while red, cyan, orange, and purple boxes represent the predictions of DTPTrack, OSTrack, ODTrack, and LoRAT, respectively. Across all examples, existing methods frequently drift to appearance-similar distractors, fall behind fast-moving targets, or lock onto background structures, resulting in fragmented or unstable trajectories.

In contrast, DTPTrack consistently stays tightly aligned with the target, even for small target sizes, heavy occlusion, strong background clutter, and dense distractor configurations. For instance, when multiple similar penguins appear in the scene, competing trackers gradually switch identities, whereas our tracker persists in tracking the correct instance. A similar pattern is observed in the bird (second row) and chameleon (third row) sequences, where high-textured background regions easily attract other methods, but DTPTrack maintains a stable prediction. In the coin-hand sequence with many near-duplicate objects, baselines often jump between coins or to the hand, while our predictions remain identity-consistent. These results visually confirm that the Temporal Reliability Calibrator and Temporal Guidance Synthesizer in DTPTrack effectively filter unreliable temporal cues and provide robust guidance to the backbone, leading to markedly improved long-term tracking stability.

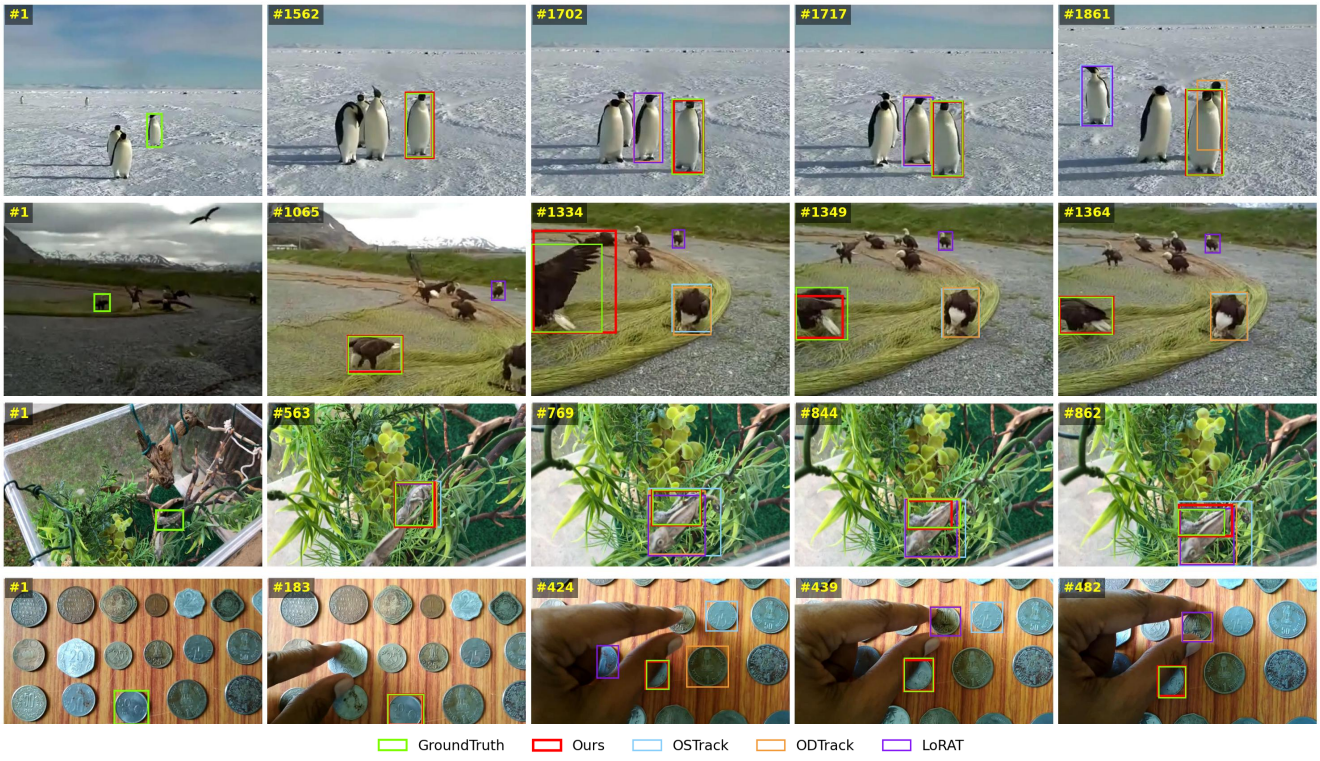

Figure 4. **Qualitative comparison with state-of-the-art trackers on challenging scenarios.**