
LoRATv2: Enabling Low-Cost Temporal Modeling in One-Stream Trackers

Liting Lin

Pengcheng Laboratory
lt.lin@outlook.com

Heng Fan

University of North Texas
heng.fan@unt.edu

Zhipeng Zhang

Shanghai Jiao Tong University
Anyverse Intelligence
zhipeng.zhang.cv@outlook.com

Yuqing Huang

Pengcheng Laboratory
domaingreen2@gmail.com

Yaowei Wang*

Harbin Institute of Technology, Shenzhen
Pengcheng Laboratory
wangyaowei@hit.edu.cn

Yong Xu

South China University of Technology
yxu@scut.edu.cn

Haibin Ling*

Westlake University
linghaibin@westlake.edu.cn

Appendix

A Additional Implementation Details

This section provides further details on the implementation of the proposed LoRATv2 tracker. Many foundational settings, including most hyperparameters, are identical to those used in the original LoRAT [10]. Key hyperparameters for training and inference are summarized in Tab. 1.

A.1 Model Configuration

Input Embeddings. Input images (template and search regions) are tokenized with a patch size of 14. These patch embeddings are then summed with shared, frozen positional embeddings and stream-specific token type embeddings, following the methodology of LoRAT [10].

Backbone. We use standard Vision Transformer (ViT) configurations: ViT-B/14 for base models and ViT-L/14 for large models, corresponding to the details in [14, 5]. Backbones are initialized with DINOv2 pre-trained weights [14, 4] and remain frozen throughout training; only the LoRA modules and prediction heads are trainable.

LoRA. All Low-Rank Adaptation (LoRA) [7] modules utilize a rank of $r = 64$. LoRA is applied to all linear projection matrices within the attention and MLP blocks of the ViT backbone.

Prediction Head. The prediction head consists of two separate 3-layer Multi-Layer Perceptrons (MLPs): one for regressing bounding box coordinates and another for predicting the target classification score.

*Corresponding author

Table 1: Hyper-parameters used in our models.

Item	Value
template area factor	2
search region area factor	4 (-224) / 5 (-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(-224) / 0.40(-378)
LoRA rank r	64

A.2 Training Details

Loss. The overall loss is a sum of a Binary Cross-Entropy (BCE) loss for classification and a GIoU loss [17] for bounding box regression. Both loss components are equally weighted with a coefficient of 1.0.

Data Augmentation. We employ the common data augmentation pipeline [10] on the image pairs/triplets, including random horizontal flipping (probability 0.5), color jittering (e.g., brightness, contrast, saturation, with a factor of 0.4) and the 3-Augment strategy [18]. The template image is cropped with an area factor of 2 around the initial bounding box. Search regions are cropped with an area factor of 4 (for 224×224 inputs x^1) or 5 (for 378×378 inputs x^2). Scale jitter (factor 0.25) and translation jitter (factor 3) are also applied on both x^1 and x^2 .

Optimization. Models are typically trained for 170 epochs. When training exclusively on the GOT-10k training split for its specific evaluation protocol, the training duration is 100 epochs. All models are trained using the AdamW optimizer [11] with a global batch size of 128. The initial learning rate is set to 1×10^{-4} , and a weight decay of 0.1 is applied (0 for bias and norm). A cosine learning rate schedule is used, decaying the learning rate to a minimum of 5×10^{-6} . A 2-epoch linear warmup phase precedes the main training schedule, where the learning rate gradually increases from 1×10^{-7} ($0.001 \times$ initial LR) to 1×10^{-4} . Gradients are clipped at a maximum L2 norm of 1.0. For LoRA_{v2}-L variants, DropPath [9] is applied with a rate of 0.1. Automatic Mixed Precision (AMP) is used throughout training to accelerate computation and reduce memory footprint. Further GPU memory optimization and training speedup are achieved by leveraging “torch.compile” and memory-efficient attention mechanisms [16].

A.3 Inference Details

During inference, a Hann window penalty is applied to the classification score map. The penalty coefficient is 0.45 for -224 variants and 0.40 for -378 variants.

We utilize optimized fused GPU kernels, such as those provided by FlashAttention [3, 2], during both performance and efficiency evaluations. This helps to minimize latency introduced by memory operations involving the KV caches, thereby achieving inference speeds (FPS) more commensurate with the theoretical FLOPs.

Table 2: Performance on four additional benchmarks. Scores reported are Success (SUC) for OTB100, NFS, UAV123, and TrackingNet.

	OTB100 [19]	NFS [8]	UAV123 [12]	TrackingNet [13]
LoRATv2-B ₂₂₄	71.0	66.1	71.3	84.1
LoRATv2-B ₃₇₈	71.7	66.5	71.6	84.8
LoRATv2-L ₂₂₄	72.4	65.7	72.8	85.4
LoRATv2-L ₃₇₈	72.2	67.9	72.6	85.7

Table 3: Effect of template size on performance and efficiency for LoRATv2 and LoRAT [10]. Results are Success (SUC) on LaSOT [10] and VastTrack [15], along with speed (FPS) and MACs.

Method	Template Size	Success (SUC, %)		Efficiency (fps / G)	
		LaSOT	VastTrack	Speed	MACs
LoRAT-B ₂₂₄	112	71.7	38.7	546	30
LoRAT-B ₂₂₄	224	71.9	39.0	467	49
LoRATv2-B ₂₂₄	112	71.3	37.4	714	24
LoRATv2-B ₂₂₄	224	72.0	39.1	713	25

B Performance on Additional Benchmarks

In addition to the aforementioned datasets, we evaluate LoRATv2 on four widely-used benchmarks: OTB100 [19], NFS [8], UAV123 [12], and TrackingNet [13]. The results is presented in Tab. 2.

C Supplementary Ablation Studies

This section presents supplementary ablation studies that further validate the design choices of LoRATv2. Unless otherwise specified, experiments were conducted using the ViT-Base backbone, with performance reported on the LaSOT [6] and VastTrack [15] benchmarks.

C.1 Impact of Template Resolution

We first investigate the influence of template resolution on tracking accuracy and computational cost, comparing LoRAT and LoRATv2. The results are presented in Tab. 3.

As shown in Tab. 3, for LoRAT, increasing the template size from 112×112 to 224×224 marginally improves SUC on LaSOT by +0.2% but substantially increases MACs (30G to 49G) and reduces speed (546 to 467 FPS). In contrast, LoRATv2 with a 224×224 template achieves higher SUC on both LaSOT (72.0%) and VastTrack (39.1%) compared to its 112×112 counterpart, while maintaining high speed (701 FPS) and incurring only a minimal MACs increase (24G to 25G). This demonstrates that frame-wise causal attention and KV caching in LoRATv2 effectively mitigate the computational overhead of larger templates, allowing the model to benefit from richer context. The slightly lower performance of LoRATv2-B-224 with a 112×112 template compared to LoRAT-B-224 with the same template size might suggest that LoRATv2’s architecture is better optimized to leverage richer contextual information provided by larger inputs.

C.2 Effectiveness of Multi-Frame Temporal Modeling

We investigate whether the performance gains of our multi-frame model, LoRATv2-B-378, stem simply from processing more pixels or from explicit temporal modeling. To this end, we train a single-search-region model with an enlarged search region of 378×378 (variant b). As shown in Table 4, while this larger single-frame model outperforms the baseline 224×224 model (②), it falls short of our multi-frame model (③) on both LaSOT (73.3 vs. 74.3 SUC) and VastTrack (39.5 vs. 40.6 SUC). This comparison is particularly insightful as both models have comparable computational costs (77 vs. 81G MACs). The results confirm that explicitly modeling temporal context with an additional search frame via our proposed framework is more beneficial than merely increasing the spatial extent of a single search view.

Table 4: Additional ablation studies comparing our multi-frame approach with a larger single search region and the S-MAM attention mechanism. All models use the ViT-Base backbone.

Variants		LaSOT [6]		VastTrack [15]		FPS	MACs (G)
		SUC	P	SUC	P		
<i>Phase 1 models:</i>							
①	LoRATv2-B ₂₂₄	72.0	77.9	39.1	38.7	713	25
②	+ search region 378	73.3	79.7	39.5	39.2	503	77
<i>Phase 2 models:</i>							
③	LoRATv2-B ₃₇₈	74.3	80.9	40.6	40.8	425	81
④	+ S-MAM [1]	74.0	80.5	40.2	40.5	420	81

C.3 Comparison with S-MAM

We also compare frame-wise causal attention (FWCA) against the S-MAM from MixViT [1], another method for improving attention efficiency. We replace the FWCA in our LoRATv2-B-378 model with S-MAM (④). Table 4 shows that our FWCA-based model (③) achieves slightly better performance (74.3 vs. 74.0 SUC on LaSOT) with nearly identical efficiency. This suggests superior performance of FWCA compared to S-MAM.

While S-MAM prunes target-to-search cross-attention and attention among target templates, effectively treating multiple templates as a reference bank for localization, FWCA establishes a genuine autoregressive history. By sequentially processing frames, it captures the target’s evolving appearance and state, which strengthens tracking robustness and paves the way for future advancements in motions-aware or occlusions-aware trackers.

D Limitations

While LoRATv2 demonstrates strong performance and efficiency, we identify two main limitations:

1. **Hyperparameter Sensitivity:** Empirically, the causal attention mechanism (as in LoRATv2) exhibited greater sensitivity to hyperparameter configurations compared to traditional non-causal (fully self-attention based) models, potentially necessitating more careful tuning.
2. **Training Pipeline Complexity:** The beneficial multi-phase progressive training strategy inherently introduces additional complexity to the overall training pipeline when compared to simpler single-stage training methods.

References

- [1] Cui, Y., Jiang, C., Wu, G., Wang, L., 2024. MixFormer: End-to-end tracking with iterative mixed attention. IEEE Transactions on Pattern Analysis and Machine Intelligence , 1–18.
- [2] Dao, T., 2024. FlashAttention-2: Faster attention with better parallelism and work partitioning, in: ICLR.
- [3] Dao, T., Fu, D.Y., Ermon, S., Rudra, A., Ré, C., 2022. FlashAttention: Fast and memory-efficient exact attention with IO-awareness, in: NeurIPS.
- [4] Darcet, T., Oquab, M., Mairal, J., Bojanowski, P., 2024. Vision transformers need registers, in: ICLR.
- [5] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2021. An image is worth 16x16 words: Transformers for image recognition at scale, in: ICLR.
- [6] Fan, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., Bai, H., Xu, Y., Liao, C., Ling, H., 2019. LaSOT: A high-quality benchmark for large-scale single object tracking, in: CVPR.
- [7] Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., 2022. LoRA: Low-rank adaptation of large language models, in: ICLR.
- [8] Kiani Galoogahi, H., Fagg, A., Huang, C., Ramanan, D., Lucey, S., 2017. Need for speed: A benchmark for higher frame rate object tracking, in: ICCV.

- [9] Larsson, G., Maire, M., Shakhnarovich, G., 2016. FractalNet: Ultra-deep neural networks without residuals, in: ICLR.
- [10] Lin, L., Fan, H., Zhang, Z., Wang, Y., Xu, Y., Ling, H., 2024. Tracking meets lora: Faster training, larger model, stronger performance, in: ECCV.
- [11] Loshchilov, I., Hutter, F., 2019. Decoupled weight decay regularization, in: ICLR.
- [12] Mueller, M., Smith, N., Ghanem, B., 2016. A benchmark and simulator for uav tracking, in: ECCV.
- [13] Muller, M., Bibi, A., Giancola, S., Alsubaihi, S., Ghanem, B., 2018. TrackingNet: A large-scale dataset and benchmark for object tracking in the wild, in: ECCV.
- [14] Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al., 2024. DINOv2: Learning robust visual features without supervision, in: TMLR.
- [15] Peng, L., Gao, J., Liu, X., Li, W., Dong, S., Zhang, Z., Fan, H., Zhang, L., 2024. Vasttrack: Vast category visual object tracking, in: NeurIPS.
- [16] Rabe, M.N., Staats, C., 2021. Self-attention does not need $o(n^2)$ memory. arXiv preprint arXiv:2112.05682 .
- [17] Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S., 2019. Generalized intersection over union: A metric and a loss for bounding box regression, in: CVPR.
- [18] Touvron, H., Cord, M., Jégou, H., 2022. DeiT III: Revenge of the ViT, in: ECCV, Springer.
- [19] Wu, Y., Lim, J., Yang, M.H., 2015. Object tracking benchmark. IEEE Transactions on Pattern Analysis and Machine Intelligence 37, 1834–1848.