MixFormerV2: Efficient Fully Transformer Tracking Supplementary Material

Yutao Cui[†] **Tianhui Song**[†] **Gangshan Wu** **Limin Wang**[*]
State Key Laboratory for Novel Software Technology, Nanjing University, China

https://github.com/MCG-NJU/MixFormerV2

Broader Impact

In this paper, we introduce MixFormerV2, a fully transformer tracking approach for efficiently and effectively estimating the state of an arbitrary target in a video. Generic object tracking is one of the fundamental computer vision problems with numerous applications. For example, object tracking (and hence MixFormerV2) could be applied to human-machine interaction, visual surveillance and unmanned vehicles. Our research could be used to improve the tracking performance while maintaining a high running speed. Of particular concern is the use of the tracker by those wishing to position and surveil others illegally. Besides, if the tracker is used in unmanned vehicles, it may be a challenge when facing the complex real-world scenarios. To mitigate the risks associated with using MixFormerV2, we encourage researchers to understand the impacts of using the trackers in particular real-world scenarios.

Limitations

The main limitation lies in the training overhead of MixFormerV2-S, which performs *multiple* model pruning based on the dense-to-sparse distillation and deep-to-shallow distillation. In detail, we first perform distillation from MixViT with 12 layers and plain corner head to MixFormerV2 of 12 layers. The 12-layers MixFormerV2 is pruned to 8-layers and then to 4-layers MixFormerV2 based on the deep-to-shallow distillation. Finally, the MLP-ratio-4.0 4-layers MixFormerV2 is pruned to the MLP-ratio-4.0 4-layers MixFormerV2-S for real-time tracking on CPU. For each step, it requires training for 500 epochs which is time-consuming.

Details of Training Time

The models are trained on 8 Nvidia RTX8000 GPUs. The dense-to-sparse stage takes about 43 hours. The deep-to-shallow stage1 (12-to-8 layers) takes about 42 hours, and stage2 (8-to-4 layers) takes about 35 hours.

S.0 Introduction

In the supplementary material, we first present more results on VOT20 [9] and GOT10k [8]. Then we perform more ablation studies on our MixFormerV2 framework and the model pruning route during the distillation-based model reduction. We also provide some visualization results of the prediction-token-to-search and prediction-token-to-template attention maps.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

	KCF [7]	SiamFC [1]	ATOM [5]	LightTrack [14]	DiMP [2]	STARK [13]	TransT [3]	CSWinTT [12]	MixFormer [4]	**Ours-S**	**Ours-B**
VOT20$_{EAO}$	0.154	0.179	0.271	0.242	0.274	0.280	-	0.304	-	0.258	**0.322**
GOT10k$_{AO}$	0.203	0.348	0.556	-	0.611	0.688	0.723*	0.694	0.726*	0.621*	**0.739***

Table 1: State-of-the-art comparison on VOT2020 [9] and GOT10k [8]. $*$ denotes training with four datasets including LaSOT [6], TrackingNet [11], GOT10k [8] and COCO [10]. The best results are shown in **bold** font.

token num.	MLP num.	AUC
1	4	67.1%
4	4	67.3
4	1	67.5%

(a) **Different prediction designs**. 'token num.' indicates the number of the learnable prediction tokens, 'MLP num.' denotes the number of employed MLP layers for localization. Models are *without* distillation and score prediction.

blocks num.	head	AUC
12	Py-Corner.	69.0%
12	T4	68.9%
8	T4	68.5%

(b) **Model pruning route of MixFormerV2-B***. 'T4' denotes the proposed distribution-based prediction with 4 prediction tokens. We use the MixViT-B as the distillation teacher for this analysis. Models are *without* score prediction.

blocks num.	head	MLP-r	AUC
12	Cor.	4	68.2%
12	T4	4	67.7%
8	T4	4	66.6%
4	T4	4	61.0%
4	T4	1	59.4%

(c) **Model pruning route of MixFormerV2-S**. 'Cor.' represents for the plain corner head, which is used in the initial teacher model. 'MLP-r' denotes the MLP ratio in backbone. Models are *without* score prediction.

Init. method	LaSOT	LaSOT_ext	UAV123
Tea-fir4	62.9%	45.2%	65.7%
Tea-skip4	64.4%	46.1%	66.6%
PMDP	64.8%	47.1%	67.5%

(d) **Progressive Model Depth Pruning (PMDP).** 'Tea-fir4' denotes using first 4 layers of the teacher for student initialization. 'Tea-skip4' is using 4 skipped layers of the teacher.

Arch	Online	Epoch m	AUC
MixFormerV2-B	no	30	68.3%
MixFormerV2-B	no	40	68.5%
MixFormerV2-B	no	50	68.5%

(e) **Eliminating Epochs.** 'Epoch m' indicates the number of epochs in eliminating process. Models are *without* score prediction.

Table 2: **More ablation studies**. The default choice for our model is colored in gray.

S.1 More Results on VOT2020 and GOT10k

VOT2020. We evaluate our tracker on VOT2020 [9] benchmark, which consists of 60 videos with several challenges including fast motion, occlusion, etc. The results is reported in Table 1, with metrics Expected Average Overlap(EAO) considering both Accuracy(A) and Robustness. Our MixFormerV2-B obtains an EAO score of 0.322 surpassing CSWinTT by 1.8%. Besides, our MixFormerV2-S achieves an EAO score of 0.258, which is higher than the efficient tracker LightTrack, with a real-time running speed on CPU.

GOT10k. GOT10k [8] is a large-scale dataset with over 10000 video segments and has 180 segments for the test set. Apart from generic classes of moving objects and motion patterns, the object classes in the train and test set are zero-overlapped. We evaluate MixFormerV2 trained with the four datasets of LaSOT, TrackingNet, COCO and GOT10k-train on GOT10k-test. We compare it with MixFormer and TransT with the same training datasets for fair comparison. MixFormerV2-B improves MixFormer and TransT by 0.7% and 1.6% on AO respectively with a high running speed of 165 FPS.

S.2 More Ablation Studies

Design of Prediction Tokens. We practice three different designs of prediction tokens for the target localization in Tab. 2a. All the three methods use the formulation of estimating the probability distribution of the four coordinates of the bounding box. The model on the first line denotes using one prediction token and then predicting coordinates distribution with four independent MLP heads.

It can be observed that adopting separate prediction tokens for the four coordinates and a same MLP head retains the best accuracy.

Model Pruning Route. We present the model pruning route from the teacher model to MixFormerV2-B* and MixFormerV2-S in Tab. 2b and Tab. 2c respectively. The models on the first line are corresponding teacher models. We can see that, through the dense-to-sparse distillation, our token-based MixFormerV2-B obtains comparable accuracy with the dense-corner-based MixViT-B with higher running speed. Through the progressive model depth pruning based on the feature and logits distillation, MixFormerV2-B with 8 layers only decreases little accuracy compared to the 12-layers one.

Detailed Structure of Score Heads The Score Head is a simple MLP composed of two linear layers with the hidden dimension of 768. Specifically, firstly we average these four prediction tokens to gather the target information, and then feed the token into the MLP-based Score Head to directly predict the confidence score s which is a real number. Formally, we can represent it as:

$$s = \mathrm{MLP}\left(\mathrm{mean}\left(\mathrm{token}_X\right)\right), X \in \mathcal{T}, \mathcal{L}, \mathcal{B}, \mathcal{R}$$

Computation Loads of Different Localization Head We showcase the FLOPs of different heads as follows. Formally, we denote C_{in} as input feature dimension, C_{out} as output feature dimension, H_{in}, W_{in} as input feature map shape of convolution layer, H_{out}, W_{out} as output feature map shape, and K as the convolution kernel size. The computational complexity of one linear layer is $O(C_{in}C_{out})$, and that of one convolutional layer is $O(C_{in}C_{out}H_{out}W_{out}K^2)$.

In our situation, for T4, the Localization Head contains four MLP to predict four coordinates. Each MLP contains two linear layer, whose input and output dimensions are all 768. The loads can be calculated as:

$$Load_{T4} = 4 \times (768 \times 768 + 768 \times 72) = 2580480 \sim 2.5M$$

For Py-Corner, totally 24 convolution layers are used. The loads can be calculated as:

$$\begin{aligned} Load_{Py-Corner} = 2 * (&768 * 384 * 18 * 18 * 3 * 3 + \\ &384 * 192 * 18 * 18 * 3 * 3 + \\ &384 * 192 * 18 * 18 * 3 * 3 + \\ &192 * 96 * 36 * 36 * 3 * 3 + \\ &384 * 96 * 18 * 18 * 3 * 3 + \\ &96 * 48 * 72 * 72 * 3 * 3 + \\ &48 * 1 * 72 * 72 * 3 * 3 + \\ &192 * 96 * 18 * 18 * 3 * 3 + \\ &96 * 48 * 18 * 18 * 3 * 3 + \\ &48 * 1 * 18 * 18 * 3 * 3 + \\ &96 * 48 * 36 * 36 * 3 * 3 + \\ &48 * 1 * 36 * 36 * 3 * 3) \\ =&3902587776 \sim 3.9B \end{aligned}$$

For simplicity, we do not include some operations such as bias terms and Layer/Batch-Normalization, which does not affect the overall calculation load level. Besides, the Pyramid Corner Head utilize additional ten interpolation operations. Obviously the calculation load of Py-Corner is still hundreds of times of T4.

More Exploration of PMDP Tea-skip4 is a special initialization method, which chooses the skiped four layers (layer-3/6/9/12) of the teacher (MixViT-B) for initialization. In other words, Tea-skip4 is an extreme case of ours PMDP when the eliminating epoch m equal to 0. So it is reasonable that Tea-skip4 performs better than the baseline Tea-fir4, which employs the first four layers of the teacher (MixViT-B) to initialize the student backbone. In Table 2d, we further evaluate the performance on more benchmarks. It can be seen that ours PMDP surpasses Tea-skip4 by 1.0% on LaSOT_ext, which demonstrate its effectiveness.

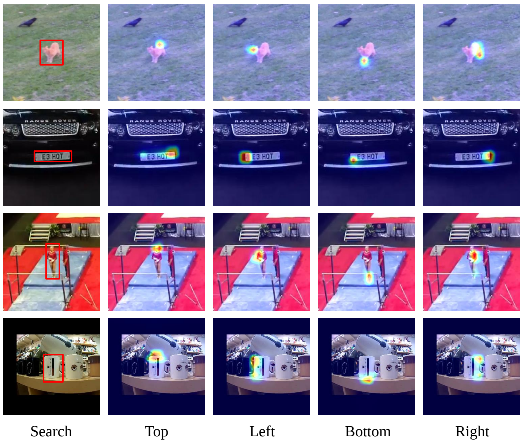


Figure 1: Visualization of prediction-token-to-search attention maps, where the prediction tokens are served as *query* of attention operation.

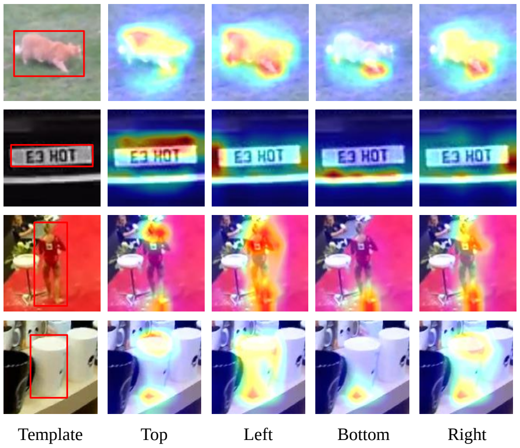


Figure 2: Visualization of prediction-token-to-template attention maps, where the prediction tokens are served as *query* of attention operation.

Determination of Eliminating Epochs As shown in the Table 2e, we find that when the epoch m greater than 40, the choice of m seems hardly affect the performance. So we determine the epoch to be 40.

S.3 Visualization Results

Visualization of Attention Map To explore how the introduced learnable prediction tokens work within the P-MAM, we visualize the attention maps of prediction-token-to-search and prediction-token-to-template in Fig. 1 and Fig. 2, where the prediction tokens are served as *query* and the others as *key*/*val* of the attention operation. From the visualization results, we can arrive that the four prediction tokens are sensitive to corresponding part of the targets and thus yielding a compact object bounding box. We suspect that the performance gap between the dense corner head based MixViT-B and our fully transformer MixFormerV2-B without distillation lies in the lack of holistic target modeling capability. Besides, the prediction tokens tend to extract partial target information in both the template and the search so as to relate the two ones.

Visualization of Predicted Probability Distribution We show two good cases and bad cases in Figure 3. In Figure 3.(a) MixFormerV2 deals with occlusion well and locate the bottom edge correctly. As show in Figure 3.(b), the probability distribution of box representation can effectively alleviate issue of ambiguous boundaries. There still exist problems like strong occlusion and similar objects which will lead distribution shift, as demonstrated in Figure 3.(c) and 3.(d).

References

[1] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *Proceedings of the European Conference on Computer Vision, ECCV Workshops*, 2016.

[2] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV*, pages 6182–6191, 2019.

[3] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2021.

[4] Yutao Cui, Cheng Jiang, Limin Wang, and Gangshan Wu. Mixformer: End-to-end tracking with iterative mixed attention. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2022.

[5] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ATOM: accurate tracking by overlap maximization. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2019.

[6] 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 *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2019.

[7] João F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(3):583–596, 2015.

[8] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(5):1562–1577, 2021.

[9] Matej Kristan, Ales Leonardis, and et. al. The eighth visual object tracking VOT2020 challenge results. In Adrien Bartoli and Andrea Fusiello, editors, *Proceedings of the European Conference on Computer Vision, ECCV Workshops*, 2020.

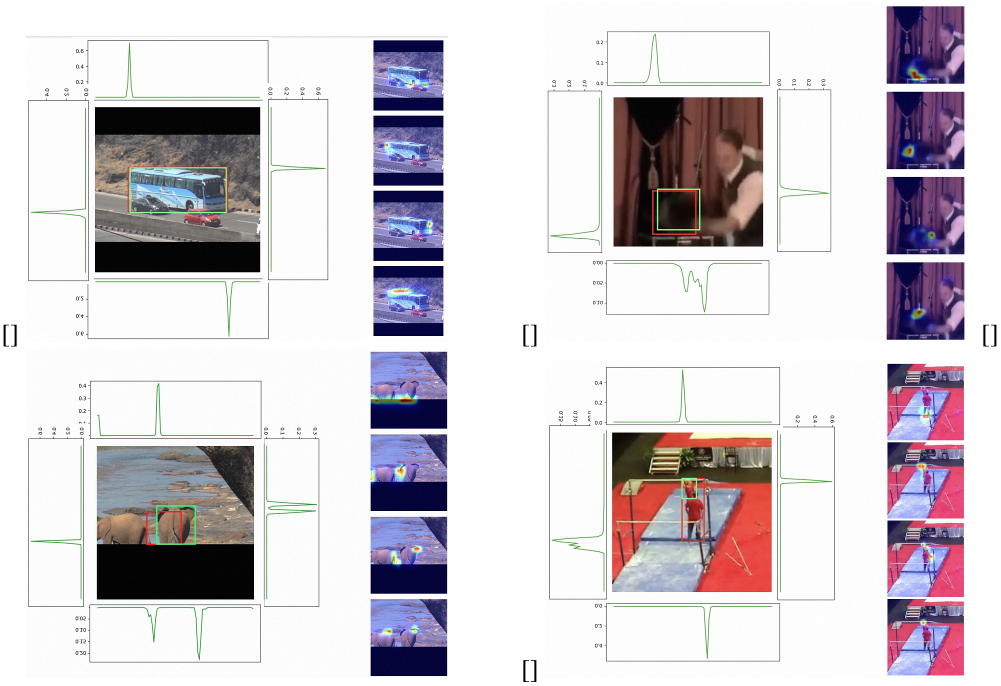
[10] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *Proceedings of the European Conference on Computer Vision, ECCV*, 2014.

[11] Matthias Müller, Adel Bibi, Silvio Giancola, Salman Al-Subaihi, and Bernard Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *Proceedings of the European Conference on Computer Vision, ECCV*, 2018.

[12] Zikai Song, Junqing Yu, Yi-Ping Phoebe Chen, and Wei Yang. Transformer tracking with cyclic shifting window attention. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2022.

[13] Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. Learning spatio-temporal transformer for visual tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV*, 2021.

[14] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15180–15189, 2021.



[]

[]

[]

[]

Figure 3: In each figure, the left one is plot of the probability distribution of bounding box, which demonstrates how our algorithm works. The right one is heatmap visualization of attention weights in the last layer of backbone. The examples are from LaSOT test subset.