Supplementary Material: Joint Feature Learning and Relation Modeling for Tracking: A One-Stream Framework

Botao Ye[1,2], Hong Chang[1,2], Bingpeng Ma[2], Shiguang Shan[1,2], and Xilin Chen[1,2]

[1] Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing, 100190, China
[2] University of Chinese Academy of Sciences, Beijing, 100049, China
botao.ye@vipl.ict.ac.cn, changhong@ict.ac.cn, bpma@ucas.ac.cn, {sgshan, xlchen}@ict.ac.cn

1 More Implementation Details

Training Details. In OSTrack-256, the input sizes of templates and search regions are 128×128 pixels and 256×256 pixels respectively, corresponding to 2^2 and 4^2 times of the target bounding box area. In OSTrack-384, the input sizes of templates and search regions are 192×192 pixels and 384×384 pixels, corresponding to 2^2 and 5^2 times of the target bounding box area. For the GOT-10k test benchmark [9], which requires training the models with only the training split of GOT-10k (one-shot setting), we set the total training epoch to 100 with 60k image pairs per epoch, and we decrease the learning rate by a factor of 10 after 80 epochs. The other settings are kept consistent with the models trained with all datasets.

Classification Loss. We adopt the weighted focal loss [11] for classification. Specifically, for each ground truth target center $\hat{p}$ and its corresponding low-resolution equivalent $\tilde{p} = [\tilde{p}_x, \tilde{p}_y]$, the ground truth heatmap can be generated using a Gaussian kernel as $\hat{\boldsymbol{P}}_{xy} = \exp\left(-\frac{(x-\tilde{p}_x)^2+(y-\tilde{p}_y)^2}{2\sigma_{\tilde{p}}^2}\right)$, where σ is an object size-adaptive standard deviation [11]. The Gaussian weighted focal loss can be formulated as:

$$L_{cls} = -\sum_{xy} \begin{cases} (1-\boldsymbol{P}_{xy})^{\alpha}\log(\boldsymbol{P}_{xy}), & \text{if } \hat{\boldsymbol{P}}_{xy} = 1 \\ (1-\hat{\boldsymbol{P}}_{xy})^{\beta}(\boldsymbol{P}_{xy})^{\alpha}\log(1-\boldsymbol{P}_{xy}), & \text{otherwise} \end{cases} \tag{1}$$

where α and β are hyper-parameters and we set $\alpha = 2$ and $\beta = 4$ as in [11, 22].

Position Embeddings. The length of the position embeddings in the pre-trained ViT is different from the length of the input template and search region embeddings. Therefore, the pre-trained positional embeddings are interpolated (2D bicubic interpolation is adopted) to the sizes of the template and search region embeddings separately, which are further added to the patch embeddings.

Model Details. In Sec.4.3 of the main paper, we compare our OSTrack (without the early candidate elimination module) with aligned two-stream track-

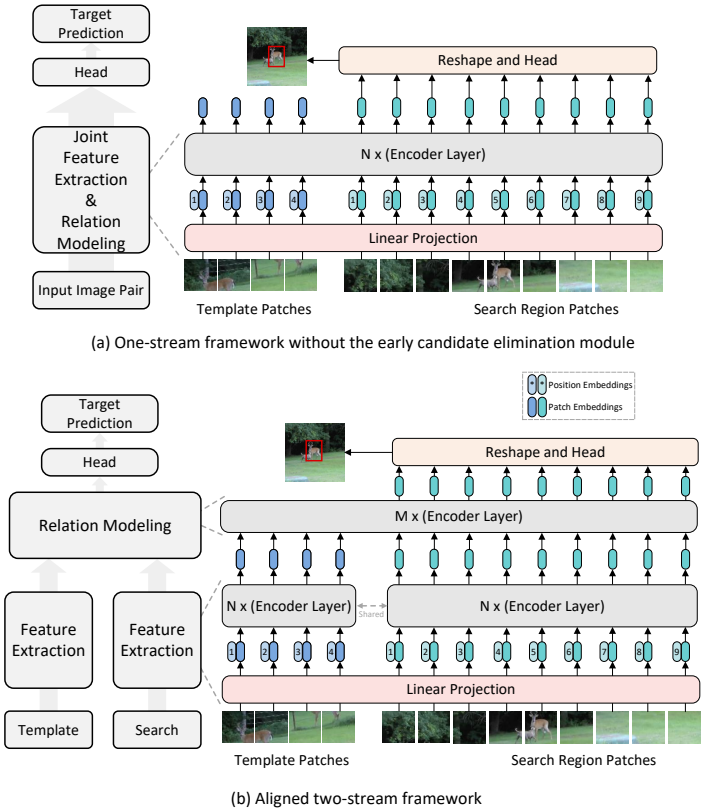

Fig. 1: **(a)** Our proposed one-stream framework without the early candidate elimination module, which combines feature extraction and relation modeling modules. **(b)** The aligned two-stream tracking framework, which extracts features of the template and search region separately and then models the feature relation with extra Transformer encoder layers.

ers (*i.e.*, STARK-aligned and SwinTrack-aligned), and we further present the detailed structures in this section. The proposed one-stream framework, as shown in Fig. 1(a), combines feature extraction and relation modeling modules into a single ViT backbone. The aligned two-stream framework, as shown in Fig. 1(b), first extracts features of the template and the search region separately with the same ViT backbone and then models the feature relation with several extra Transformer encoder layers. As presented in Sec.4.3 of the main paper, this relation modeling module is instantiated with the encoder structure proposed in STARK [19] (STARK-aligned) and SwinTrack [14] (SwinTrack-aligned) separately.

Table 1: Ablation study on different choices of template tokens used to identify candidates belonging to background.

Template Token Selection	LaSOT			TrackingNet			GOT-10k		
	Success	P_{Norm}	P	Success	P_{Norm}	P	AO	$SR_{0.5}$	$SR_{0.75}$
No Early Candidate Elimination	68.7	78.1	74.6	82.9	87.5	81.6	**73.6**	83.0	71.7
All Template Tokens	68.1	77.4	73.5	82.8	87.5	81.6	72.9	82.3	70.1
All Template Tokens within GT Box	68.5	78.1	74.2	**83.1**	**87.8**	81.7	**73.6**	**83.4**	**72.0**
Center 4x4 Template Tokens	68.3	77.6	73.9	82.9	87.7	**82.0**	73.5	83.0	71.5
Center Template Token	**69.1**	**78.7**	**75.2**	**83.1**	**87.8**	**82.0**	**73.6**	82.8	71.4

2 More Ablation Studies

2.1 The Effect of Different Template Token Choices.

As pointed out in Sec.3.2 of the main paper, the goal of the early candidate elimination module is to identify and discard candidates belonging to background regions based on the ranking of similarity between the target and each candidate. However, the input template also contains background regions, which introduces noisy information when calculating the similarity score. Therefore, different choices of template parts (tokens) used for the similarity calculation may influence the candidate elimination results and consequently affect the tracking performance. We compare four different template token choices (the similarity scores of all chosen template tokens are summed up for the final ranking): 1) all template tokens; 2) all template tokens within the ground truth target bounding box; 3) template tokens within a 4x4 region around the center of the template image; 4) the template token corresponding to the center of the template image. The result comparison of these template choices is shown in Tab. 1. The results demonstrate that different template token choices do affect the quality of identifying background candidates. Since the input template contains background regions, directly using "All Template Tokens" clearly degrades the tracking performance compared with the baseline ("No Early Candidate Elimination"), *i.e.*, 0.6% lower in LaSOT AUC. Compared to other choices, using the central template token shows better performance, probably because the central token does not contain any background region and has aggregated the entire target information through self-attention.

2.2 Identity Embeddings and Relative Positional Embeddings

We additionally verify the effect of adding identity embeddings and relative positional embeddings. Specifically, for the identity embeddings, we add learnable identity embeddings (to indicate a token belonging to the template or search region as in BERT [6]) to template tokens and search region tokens separately. For the relative positional embeddings, the same method as in SwinTrack [14] is adopted. The results are presented in Tab. 2, these two components do not bring performance gain compared to the original design, thus not adopted in our model.

Table 2: The effect of adding additional identity embeddings to the template and search region embeddings and adding relative positional embeddings to the OSTrack-256 (without the early candidate elimination module). The results on LaSOT [7], TrackingNet [16] and GOT10k [9] benchmarks are presented.

	LaSOT			TrackingNet			GOT-10k		
	Success	P_{Norm}	P	Success	P_{Norm}	P	AO	$SR_{0.5}$	$SR_{0.75}$
Ours	**68.7**	**78.1**	**74.6**	82.9	87.5	81.6	73.6	83.0	**71.7**
+ Identity Embeddings	68.0	77.3	73.6	**83.3**	**88.0**	**82.2**	73.6	82.9	**71.7**
+ Relative Positional Embeddings	68.5	77.8	74.1	83.2	87.8	82.0	**73.7**	**83.3**	71.2

Table 3: Add additional relation modeling module to our OSTrack-256 (without the early candidate elimination module).

	LaSOT			TrackingNet			GOT-10k		
	Success	P_{Norm}	P	Success	P_{Norm}	P	AO	$SR_{0.5}$	$SR_{0.75}$
Ours	**68.7**	**78.1**	**74.6**	**82.9**	**87.5**	**81.6**	**73.6**	**83.0**	**71.7**
+ Relation Modeling	68.5	78.0	74.1	**82.9**	87.4	81.5	72.7	82.2	70.5

2.3 Additional Relation Modeling Module

To investigate whether our one-stream framework does not require an extra feature relation module, we add an additional transformer-based feature fusion module proposed in [14], which consists of 4 self-attention layers and 1 cross-attention layer, to further fusion the extracted template and search region features. As the results in Tab. 3 show, adding such a relation modeling module instead degrades the tracking performance, indicating that the output search region features of the ViT backbone have been sufficiently fused with the template features.

2.4 Fewer Relation Modeling Layers

In the implementation of vanilla OSTrack, all encoder layers in ViT-Base (12 layers in total) are used for simultaneous feature extraction and relation modeling. In this subsection, we try to decrease the number of layers used for relation modeling. Specifically, only the last n encoder layers are used for simultaneous feature extraction and relation modeling, and the first $12 - n$ layers are only used for the template and search region feature extraction. n is set to be 6 and 3 separately and the results are presented in Tab. 4. The results show that using fewer encoder layers for simultaneous feature extraction and relation modeling will degrade the tracking performance, showing the necessity of sufficient feature fusion.

2.5 Different Token Drop Rate

We also try to apply a different keeping ratio ρ for the early candidate elimination module. As the results in Tab. 5 show, using $\rho < 0.7$ leads to performance drop

Table 4: Ablation studies on the number of encoder layers used for relation modeling.

	LaSOT			TrackingNet			GOT-10k		
	Success	P_{Norm}	P	Success	P_{Norm}	P	AO	$SR_{0.5}$	$SR_{0.75}$
12 (Ours)	**68.7**	**78.1**	**74.6**	82.9	**87.5**	**81.6**	**73.6**	**83.0**	**71.7**
6	67.9	77.3	73.6	**83.0**	**87.5**	81.5	73.3	82.9	71.4
3	67.8	77.0	73.5	82.7	87.4	80.7	72.8	82.5	70.7

Table 5: Different keeping ratio ρ used in the early candidate elimination module ($\rho = 1$ means the early candidate elimination module is not adopted).

Keeping Ratio	LaSOT			TrackingNet			GOT-10k			MACs (G)
	Success	P_{Norm}	P	Success	P_{Norm}	P	AO	$SR_{0.5}$	$SR_{0.75}$	
1	68.7	78.1	74.6	82.9	87.5	81.6	73.6	83.0	71.7	29.0
0.9	68.7	78.2	74.6	83.2	87.8	82.0	**74.1**	**83.6**	**71.8**	26.2
0.8	68.9	78.4	74.9	**83.3**	**88.0**	**82.3**	73.4	82.7	71.4	23.6
0.7	**69.1**	**78.7**	**75.2**	83.1	87.8	82.0	73.6	82.8	71.4	21.5
0.6	68.4	77.9	74.3	83.1	87.6	81.8	73.5	82.9	71.7	19.6
0.5	67.8	77.2	73.4	82.7	87.4	81.3	71.8	81.3	68.4	**18.0**

on the LaSOT [7] tracking benchmark since small ρ may cause a significant information loss. However, the reduction in computational cost that comes with large ρ is limited. Setting $\rho = 0.7$ shows a decent decrease in computational cost with a slight improvement in tracking performance. Therefore, we use $\rho = 0.7$ in our experiments.

3 Results on VOT2020

VOT2020 [10] is a challenging short-term tracking benchmark that is evaluated by target segmentation results. To evaluate OSTrack on VOT2020, we use AlphaRefine [20] to generate segmentation masks, and the results are shown in Tab. 6. Since the wide existence of distractors in VOT2020, updating the template during the tracking process becomes a common practice to avoid tracking drift, which can bring significant performance gain (*e.g.*, STARK-ST50 citestark raises the EAO from 0.462 to 0.505 by simply adding a dynamic template). OSTrack-256 obtains an EAO of 0.518, which already outperforms the STARK-ST50 with an online template updating mechanism. This demonstrates the great potential of OSTrack which serves as a neat and strong baseline model.

4 Results on ITB

ITB [13] benchmark is a newly collected benchmark with 9 representative scenarios and 180 diverse videos, which contains more informative tracking sequences. Tab. 7 shows the results of OSTrack compared with other SOTA tackers. Our OSTrack-384 achieves 64.8% in mIoU, surpassing the previous best tracker STARK [19] by a large margin (7.2%).

Table 6: Comparison on VOT2020 benchmark. The left part of the trackers adopt an online template update mechanism, while the right part of the trackers do not. The best two results are shown in **red** and **blue** fonts.

	Ocean [21]	ATOM [4]	D3S [15]	AlphaRef [20]	STARK-ST50 [19]	STARK-ST101 [19]	SiamMask [18]	STARK-S50 [19]	OSTrack-256	OSTrack-384
EAO (↑)	0.43	0.271	0.439	0.482	0.505	0.497	0.321	0.462	**0.518**	**0.524**
Accuracy (↑)	0.693	0.462	0.699	0.754	0.759	**0.763**	0.624	0.761	0.762	**0.767**
Robustness (↑)	0.754	0.734	0.769	0.777	0.817	0.789	0.648	0.749	**0.814**	**0.816**

Table 7: Comparison with state-of-the-arts on ITB [13] benchmark. mIoU(%) scores are reported. The best two results are shown in **red** and **blue** fonts.

	SiamRPN++ [12]	Ocean [21]	GAT [8]	ATOM [4]	DiMP [1]	PrDiMP [5]	KYS [2]	TrDiMP [17]	TransT [3]	STARK [19]	OSTrack -246	OSTrack -384
mIoU	44.1	47.7	44.9	47.2	53.7	54.4	52.0	56.1	54.7	57.6	**61.2**	**64.8**

5 More Visualization

We first provide more visualization results for attention weights of the search region corresponding to the center part of the template (which can be seen as the target) in Fig. 2. The results show that the model attends to the foreground objects at an early stage (see "Layer 4" in Fig. 2) and finally shows great discriminative power between the target and distractors (see "Layer 12" in Fig. 2). These phenomenons demonstrate that the proposed OSTrack can extract target-oriented features with strong target-distractor discriminability.

In Fig. 3, more visualization results of the early candidate elimination module are presented. The results validate that the proposed method can effectively identify and discard background regions under various target categories and challenge scenarios (*e.g.*, target deformation, occlusion, motion blur, *etc.*).

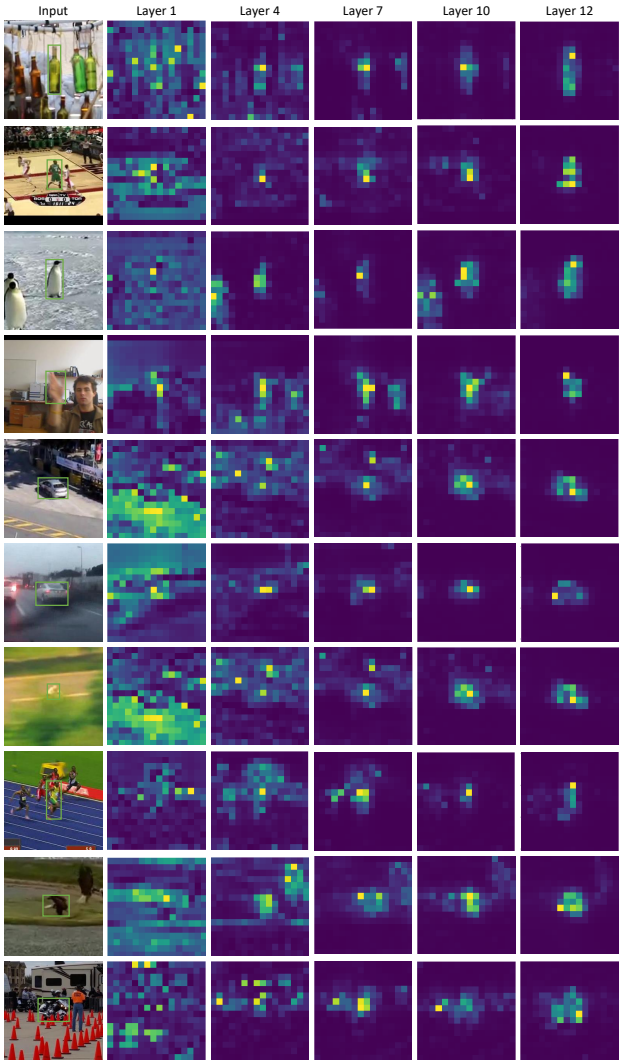

Fig. 2: Extended visualization for attention weights of the search region corresponding to the center part of template after different ViT layers, the green rectangles indicate target objects. The results show that our one-stream framework is able to distinguish between targets and distractors and progressively focus on targets.

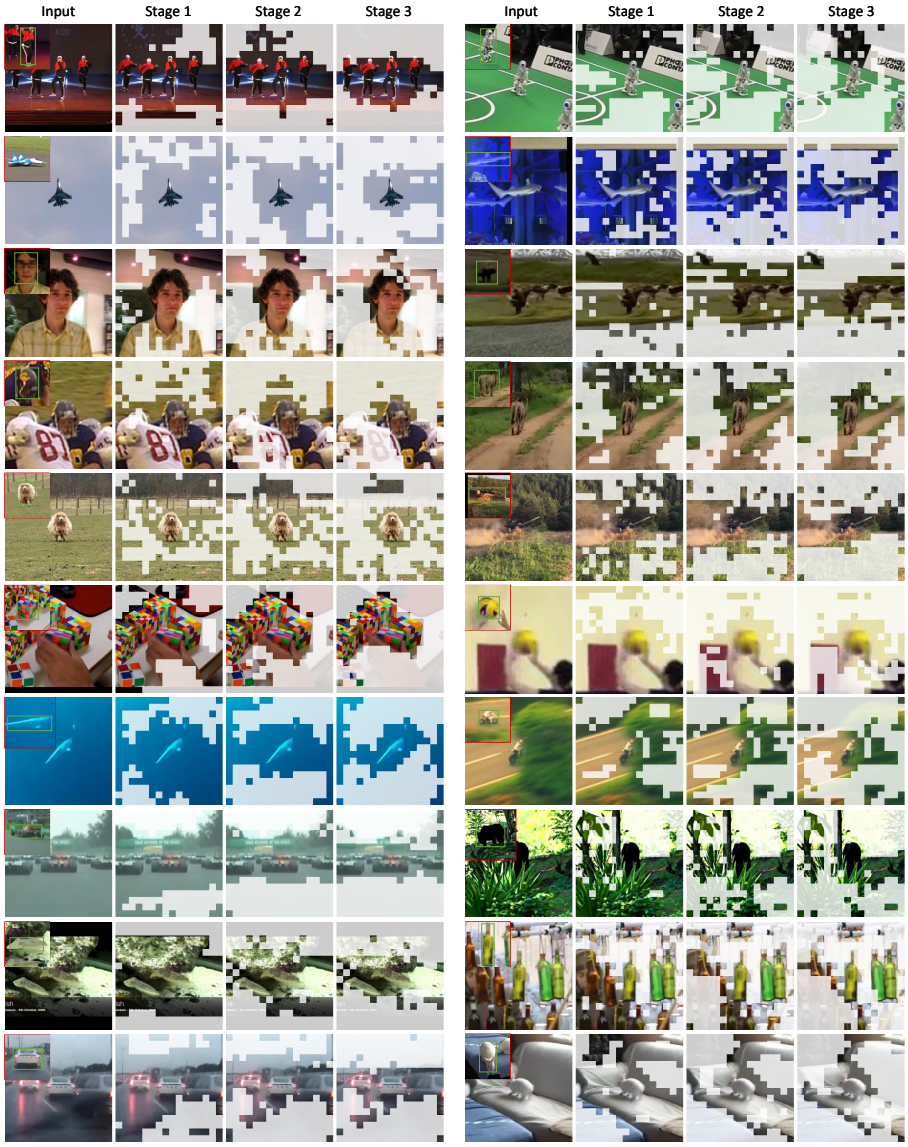

Fig. 3: Extended visualization results of the progressive candidate elimination process. The main body of "Input" is the search region image, and the upper left corner shows the corresponding template image. The Green rectangles indicate target objects and the masked regions represent the discarded tokens. Our early candidate elimination module can effectively dealing with different tracking target and scenarios.

References

1. Bhat, G., Danelljan, M., Gool, L.V., Timofte, R.: Learning discriminative model prediction for tracking. In: ICCV. pp. 6182–6191 (2019) 6
2. Bhat, G., Danelljan, M., Gool, L.V., Timofte, R.: Know your surroundings: Exploiting scene information for object tracking. In: ECCV. pp. 205–221 (2020) 6
3. Chen, X., Yan, B., Zhu, J., Wang, D., Yang, X., Lu, H.: Transformer tracking. In: CVPR. pp. 8126–8135 (2021) 6
4. Danelljan, M., Bhat, G., Khan, F.S., Felsberg, M.: Atom: Accurate tracking by overlap maximization. In: CVPR. pp. 4660–4669 (2019) 6
5. Danelljan, M., Gool, L.V., Timofte, R.: Probabilistic regression for visual tracking. In: CVPR. pp. 7183–7192 (2020) 6
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: NAACL. pp. 4171–4186 (2019) 3
7. 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: CVPR. pp. 5374–5383 (2019) 4, 5
8. Guo, D., Shao, Y., Cui, Y., Wang, Z., Zhang, L., Shen, C.: Graph attention tracking. In: CVPR. pp. 9543–9552 (2021) 6
9. Huang, L., Zhao, X., Huang, K.: Got-10k: A large high-diversity benchmark for generic object tracking in the wild. TPAMI **43**(5), 1562–1577 (2019) 1, 4
10. Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Kämäräinen, J.K., Danelljan, M., Zajc, L.Č., Lukežič, A., Drbohlav, O., et al.: The eighth visual object tracking vot2020 challenge results. In: ECCV. pp. 547–601 (2020) 5
11. Law, H., Deng, J.: Cornernet: Detecting objects as paired keypoints. In: ECCV. pp. 734–750 (2018) 1
12. Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., Yan, J.: SiamRPN++: Evolution of siamese visual tracking with very deep networks. In: CVPR. pp. 4282–4291 (2019) 6
13. Li, X., Liu, Q., Pei, W., Shen, Q., Wang, Y., Lu, H., Yang, M.H.: An informative tracking benchmark. arXiv preprint arXiv:2112.06467 (2021) 5, 6
14. Lin, L., Fan, H., Xu, Y., Ling, H.: Swintrack: A simple and strong baseline for transformer tracking. arXiv preprint arXiv:2112.00995 (2021) 2, 3, 4
15. Lukezic, A., Matas, J., Kristan, M.: D3s-a discriminative single shot segmentation tracker. In: CVPR. pp. 7133–7142 (2020) 6
16. Muller, M., Bibi, A., Giancola, S., Alsubaihi, S., Ghanem, B.: Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In: ECCV. pp. 300–317 (2018) 4
17. Wang, N., Zhou, W., Wang, J., Li, H.: Transformer meets tracker: Exploiting temporal context for robust visual tracking. In: CVPR. pp. 1571–1580 (2021) 6
18. Wang, Q., Zhang, L., Bertinetto, L., Hu, W., Torr, P.H.: Fast online object tracking and segmentation: A unifying approach. In: CVPR. pp. 1328–1338 (2019) 6
19. Yan, B., Peng, H., Fu, J., Wang, D., Lu, H.: Learning spatio-temporal transformer for visual tracking. In: ICCV. pp. 10448–10457 (2021) 2, 5, 6
20. Yan, B., Zhang, X., Wang, D., Lu, H., Yang, X.: Alpha-refine: Boosting tracking performance by precise bounding box estimation. In: CVPR. pp. 5289–5298 (2021) 5, 6
21. Zhang, Z., Peng, H., Fu, J., Li, B., Hu, W.: Ocean: Object-aware anchor-free tracking. In: ECCV. pp. 771–787 (2020) 6

22. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. arXiv preprint arXiv:1904.07850 (2019) 1