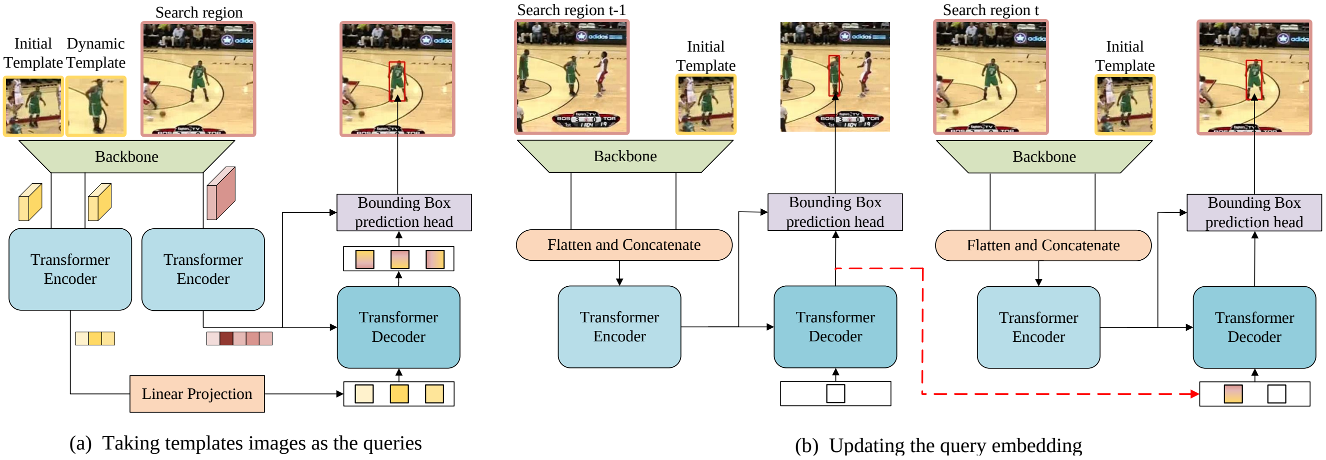
Learning Spatio-Temporal Transformer for Visual Tracking
—— Supplementary Material ——

Bin Yan[1,*], Houwen Peng[2,†], Jianlong Fu[2], Dong Wang[1,†], Huchuan Lu[1]
[1]Dalian University of Technology [2]Microsoft Research Asia



(a) Taking templates images as the queries

(b) Updating the query embedding

Figure 1: Other candidates of spatio-temporal transformer frameworks. Note that both two frameworks adopt the same score head as STARK-ST to control the template or query update. We omit the score head here for brevity.

A. Appendix

In this appendix, we provide the details of other candidate frameworks for spatio-temporal transformer design. Although, in the experiments, these frameworks are inferior to the proposed one presented in the manuscript, their differences in design may inspire future works.

A.1. Template images serve as the queries

Fig. 1(a) presents the framework of taking the template images as the queries. More specifically, the initial and dynamic template images first pass through the backbone network to generate their features. The features are then flatten and concatenated in the encoder. After that, a single-layer fully-connected perceptron is performed to transform the feature sequence to L query embeddings, where $L = 2H_zW_z/s^2$. Here, we use L queries, rather than a single one query, aiming to keep more target information. Meanwhile, the backbone features of current search region are processed by the encoder. For decoder, it receives inputs of the search region features and the L target queries. Finally, the bounding box head transforms L target queries into L box predictions. The final result is obtained by taking average of L boxes' coordinates. Similar to STARK-ST, the update of the dynamic template is controlled by a score head. The success score on LaSOT [1] of this framework is 61.2%, being 5.2% lower than that of STARK-ST50 (66.4%). The proposed STARK-ST framework considers four types of information interaction: template-to-template, template-to-search, search-to-template, and search-to-search. In contrast, this framework lacks the information interaction from the template to the search regions, thus degrading the discriminative power of the search region features.

A.2. Updating the query embedding

Fig. 1(b) demonstrates the framework of updating the query embeddings as TrackFormer [2]. At the first frame, the network takes the initial template and the current search region as the inputs, and outputs one predicted box and one dynamic target query. In subsequent frames, the decoder processes the joint set of the previous dynamic target query and the initial target query. Like TrackFormer [2],

[*] Work performed when Bin Yan was an intern of MSRA.
[†] Corresponding authors: Houwen Peng (houwen.peng@microsoft.com), Dong Wang (wdice@dlut.edu.cn).

we adopt a *track query attention* block to map two types of queries into a unified feature space before feeding the two queries to the decoder. As done by STARK-ST, the update of the dynamic query is controlled by a score head. Different from STARK-ST which introduces dynamic templates, this framework exploits an dynamic target query to capture the temporal information in an autoregressive fashion. The success score of this framework is 64.8%, being 1.6% lower than that of STARK-ST50. The underlying reason might be that the feature dependencies learned through an updatable query embedding is inferior to that by an extra template (1 *v.s.* $\frac{H_z}{s}\frac{W_z}{s}$).

References

[1] 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 *CVPR*, 2019. 1

[2] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. TrackFormer: Multi-object tracking with transformers. *arXiv preprint arXiv:2101.02702*, 2021. 1