ST-DETR: Spatio-Temporal Object Traces Attention Detection Transformer

Eslam Mohamed
Senior Machine Learning Engineer
Valeo R&D Cairo, Egypt
eslam.mohamed-abdelrahman@valeo.com

Ahmad El Sallab
Senior Expert AI/Senior Chief Engineer
Valeo R&D Cairo, Egypt
ahmad.el-sallab@valeo.com

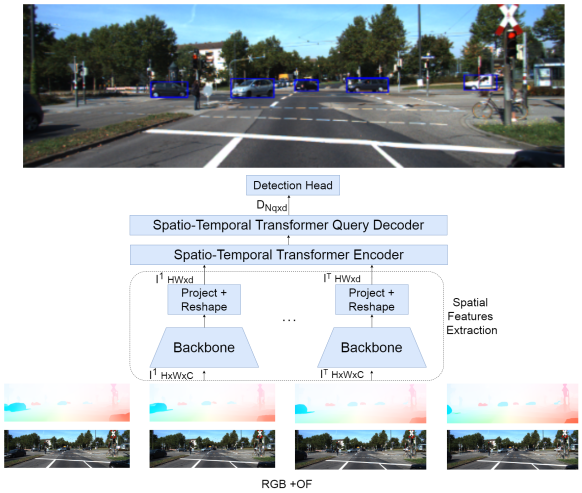

Fig. 1: High level framework of the proposed ST-DETR architecture. Lower part shows 4 time-stamp inputs consists of RGB and OF frames for each time-step. Upper part shows the output of the network.

***Abstract*—We propose ST-DETR, a Spatio-Temporal Transformer-based architecture for object detection from a sequence of temporal frames. We treat the temporal frames as sequences in both space and time and employ the full attention mechanisms to take advantage of the features correlations over both dimensions. This treatment enables us to deal with frames sequence as temporal object features traces over every location in the space. We explore two possible approaches; the early spatial features aggregation over the temporal dimension, and the late temporal aggregation of object query spatial features. Moreover, we propose a novel Temporal Positional Embedding technique to encode the time sequence information. To evaluate our approach, we choose the Moving Object Detection (MOD) task, since it is a perfect candidate to showcase the importance of the temporal dimension. Results show a significant 5% mAP improvement on the KITTI MOD dataset over the 1-step spatial baseline.**

I. Introduction

For many years, ConvNets have been the architecture of choice in computer vision in general, and for performing object detection tasks in particular. Recently, transformers have shown promising results compared to ConvNets, in object detection, [1] ,where the input image is treated as a sequence of spatial features, and full attention mechanisms are employed to extract features interactions. This motivates us to extend DETR to handle the sequence information in both spatial and temporal dimensions.

In the general problem setup, we need to perform a Spatio-temporal sequence-to-sequence mapping. The input Spatio-temporal sequence is formulated as a sequence of frames in the temporal dimension within a certain window of time, each is, in turn, a sequence of features in the spatial dimension. The output Spatio-temporal sequence is also a sequence of temporal outputs, each having a list of objects queries.

In order to transform DETR into a Spatio-Temporal model, we undergo some architectural changes. First, we adopt the classical spatial features extraction, and apply it on each input frame across the temporal window. Then we modify both the transformer encoder and decoder to handle the temporal aggregation. Here we have two options, either 1) early temporal aggregation of the spatial features, resulting in a temporal trace of features at each spatial location, or 2) later temporal objects queries aggregation, where we extract the objects queries per time step, and then stack them, resulting in a trace of objects queries. The output of either architecture is a list of object queries features, which are used to predict the bounding boxes and their corresponding classes. The same Hungarian matching and bi-partite loss [1] are adopted from DETR.

II. Related Work

Transformer based methods: For object detection in the DEtection TRansformer (DETR) [1] the input image is treated as sequence of spatial features. This enables the extension of the traditional transformer, previously used in NLP [14], in computer vision problems. Full attention mechanisms are employed to extract feature interactions in an end-to-end architecture, followed by bi-partite matching that enables the replacement of the complex post-processing pipeline in the corresponding ConvNet architectures during training. The ground truth to prediction matching is treated as an association problem and solved using the Hungarian algorithm, producing one-one mapping that can be used to calculate the loss. While Panoptic segmentation is possible directly in DETR [1], using object queries, semantic masks require different architecture

changes. Recent works extend the encoder-decoder architecture using transformers. In SETR [16], the encoder is kept convolution based same as in FCN [8], while the decoder is based on the transformer decoder architecture, with the learnable queries using progressive upsampling. The same idea is used in TransUNET [2], following the UNet architecture with skip connections between the encoder and decoder. In [15], a full transformer encoder-decoder architecture is used, which is the closest to the architecure used for MOSeg in this paper. However, in [15], the decoded segmentation mask is taken as the decoder attention weights directly, while in our case, we keep the Multi-Head attention query-key-value structure to decode the final segmentation mask.

Spatio-Temporal methods Like ConvNets in spatial computer vision, Recurrent models have been the architecture of choice for sequence models, especially in NLP. In computer vision, ConvNets and LSTM mixed architectures, like ConvLSTM have been used to handle both the spatial and temporal nature of videos, like in Moving Object Detection (MOD) [12, 13], and Instance Moving Object Segmentation [10] tasks. Recently full attention transformers [14] [9] are replacing RNN, LSTM, and GRU in NLP, taking advantage of the parallel encoding process, which removes the sequential nature of recurrent models. This motivates our work here to extend the DETR to handle also the temporal dimension, and to replace the ConvLSTM models to take advantage of the fast nature of the transformer architectures.

III. PROPOSED METHOD

A. Spatio-Temporal Detection Transformer

To transform the vanilla 1-step DETR to deal with temporal sequences, firstly, we have to first deal with multiple streams across T time steps, each having a spatial feature $I_{HW\times d}$, resulting in $I_{HW\times Td}$ streams. Then using Spatio-Temporal Transformer Encoder (ST-TE) which performs self-attention over the spatial HW dimension, resulting in $E \in \mathbb{R}^{HW\times Td}$. Finally, by exploiting the Spatio-Temporal Query Transformer Decoder (ST-TD), which performs the query-to-spatial multi-head attention transformation, resulting in $D \in \mathbb{R}^{N_q\times d_{final}}$, where d_{final} is the final dimension after spatio-temporal queries aggregation. The rest of the components remains the same as in vanilla transfomer.

One can think of two alternatives of temporal features aggregation in both the ST-TE and ST-TD, where we can early aggregate the spatial features over the temporal dimension in the ST-TE, or defer the temporal aggregation to the ST-TD to be done late over the object queries.

1) Early Temporal Aggregation: In this alternative, the list of T spatial features I_{HWxd} are aggregated and flattened into I_{HWxTD}. This aggregated tensor I_{HWxTD} can thought of as a spatial map of T temporal traces of spatial features, each of dimension d, mapped to the spatial locations $H \times W$. This is visualized in Figure 2. The ST-TE will then perform multi-head self-attention over the spatio-temporal map of object features traces. In this case, we have $Q = V = K = I_{HW\times Td}$. The spatio-temporal features traces attention map $W_{HW\times HW}$=Softmax(QK^T) is then used to obtain the spatio-temporal features $E_{HW\times Td} = W_{HW\times HW} I_{HW\times Td}$.

The ST-TD will perform multi-head query-to-spatio-temporal features traces attention, where $Q \in \mathbb{R}^{N_q\times Td}$ and $V = K = E_{HW\times Td}$. The query-spatio-temporal features traces attention map will be $W_{N_q\times HW}$ =Softmax(QK^T), resulting in $D_{N_q\times Td} = W_{N_q\times HW} E_{HW\times Td}$. This represents the final object queries spatio-temporal features, where $d_{final} = Td$ in this case.

2) Late Temporal Aggregation: We could also defer the temporal aggregation until the object queries are obtained per each time step. In this case, the resulting list of T spatial features each of $I_{HW\times d}$ dimension are not stacked and flattened as in the early aggregation. The ST-TE is formed on T Spatial Transformer Encoders, same as in the vanilla DETR, each performing multi-head self-attention, resulting also in a list of T spatial features E_{HWxd}. Finally, the ST-TD is formed of two levels of decoders; spatial and temporal query decoders.

Spatial Query Decoders which are a list of T decoders, each performing multi-head attention, resulting in a list of T query features each is $D_{N_q\times d}$, which are then reshaped into an aggregated tensor over the temporal dimension to be $D_{T\times N_q d}$. Those represent the Spatio-temporal queries traces.

Temporal Query Decoder which transforms the Spatio-temporal queries traces into the final query features, using multi-head attention. The Spatio-temporal queries traces are first flattened such that $V = K = D_{TN_q\times d}$. The attention learn-able object queries will be $Q \in \mathbb{N}_{||} \times$, resulting in an attention map of dimensions $W_{N_q\times TN_q}$=Softmax(QK^T). This is illustrated in Figure 2. The TNq dimension represents the flattened late T object queries features, each of dimension d. This can be thought of as the temporal traces of objects queries as opposed to the objects features traces in the early aggregation alternative. while the N_q dimension represents the final object queries of the last time step, which are to be learned from attending to all the T times steps objects queries. Thus, the final object query features are then $D_{N_q\times d} = W_{N_q\times TN_q} D_{TN_q\times d}$.

B. Sequence-to-sequence prediction

One can notice from Figure 2 that the temporal attention takes place between the last time ($t = T$) step queries; $Q_{N_q xd}$ and the temporal traces of object queries over all the previous steps; $D_{TN_q xd}$. The reason is that we predict the objects in the last frame, given all the features of the previous frames. However, it is straightforward to modify the architecture to obtain a sequence of temporal predictions of N_q object queries per each time step T. Simply, in the Temporal Query Decoder,

TABLE I: Detailed comparisons on the effect of the motion features.

Method	mAP_{Total}	AP_{50}	AP_{75}
RGB-only	23%	42.2%	23.7%
RGB+RGB	25.3%	47.2%	24.5%
RGB + OF	**33.9%**	**59.3%**	**37.2%**

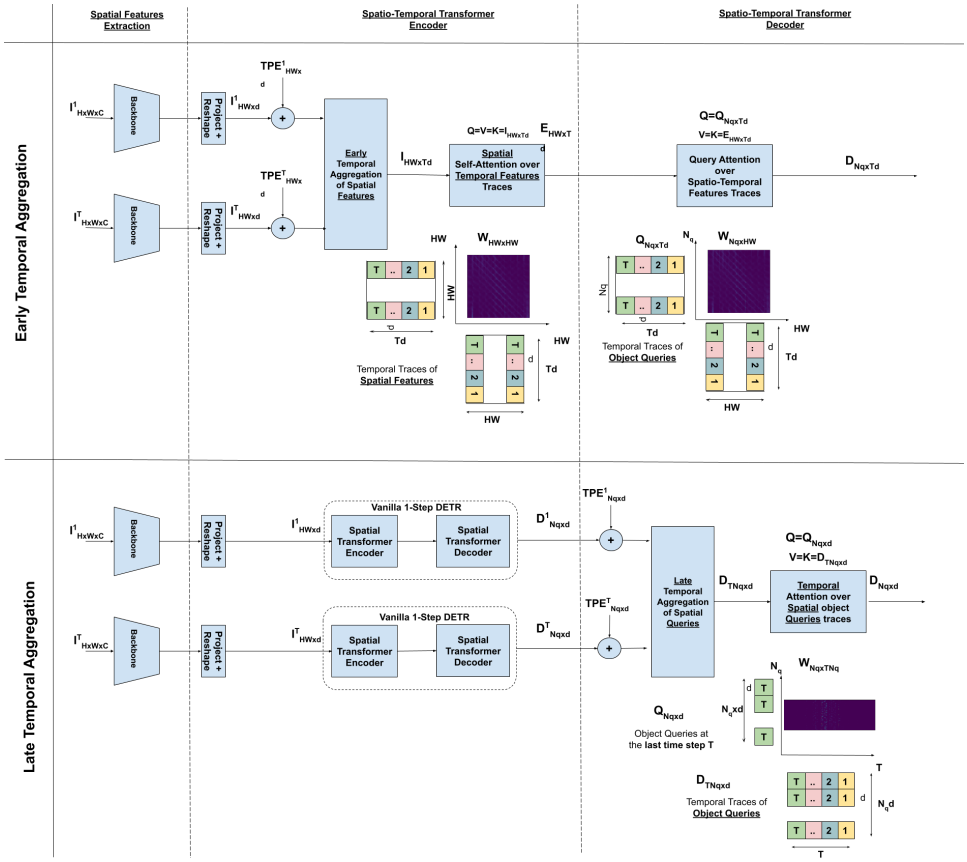

Fig. 2: Architectural details of the Early vs. Late Temporal Aggregation variants.

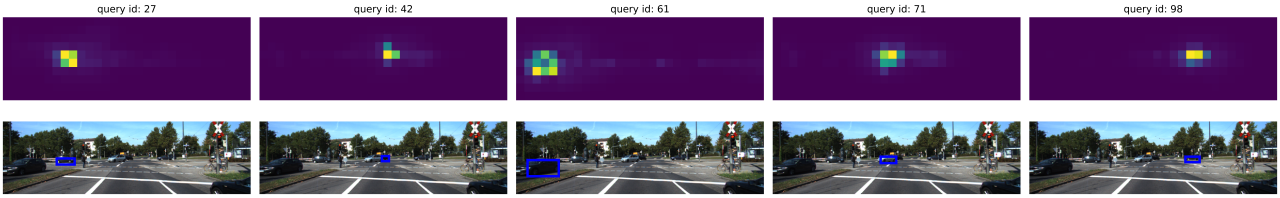

Fig. 3: Attention maps for each quires and the corresponding output bounding box .

we need to set the queries to $Q_{TNq\times d}$, and thus we have a temporal attention map $W_{TN_q\times TNq}$. This can be thought of as a sequence-to-sequence prediction problem, similar to the Neural Machine Translation (NMT) setup in [14], where we have an input sequence of the spatial feature over time, and we predict another sequence of object bounding boxes and classes that corresponds to those inputs.

TABLE II: Comparing the Early architecture, Late architecture, and vanilla one step DETR.

Method	mAP_{Total}	AP_{50}	AP_{75}
1-Step MODETR [9]	33.9%	59.3%	37.2%
Early	**38.7%**	**63.1%**	**44.6%**
Late	34%	61.1%	36.1%

C. Temporal Positional Encoding (TPE)

Transformers are originally presented as a replacement to recurrent models, due to their fast parallel encoding nature [14]. However, this comes at the cost of losing the sequential information of the input. To overcome that, positional encoding embedding was proposed in [14]. Following on that, the vanilla 1-step DETR [1] treats the input features as being sequential in the spatial dimension HW, which leads to the proposal of Spatial Positional Encoding (SPE). In ST-DETR, a similar encoding is needed to distinguish the temporal sequential information of frames. Hence, we propose a Temporal Positional Encoding (TPE), which is added just before the temporal aggregation takes place, being it early

TABLE III: Quantitative comparison results showing the effect of temporal window size

Method	mAP_{Total}	AP_{50}	AP_{75}
Early	36%	62.3%	43.4%
Early+TPE	**38.7%**	**63.1%**	**44.6%**

TABLE IV: Quantitative comparison results showing the effect of the temporal window size T

T	mAP_{Total}	AP_{50}	AP_{75}
1-Step	33.9%	59.3%	37.2%
2-Steps	**38.7%**	63.1%	**44.6%**
4-Steps	**38.7%**	**64.8%**	43%

across the spatial features traces TPE_{HWxd} or late across the object queries traces TPE_{Nqxd}, see Figure 2.

IV. EXPERIMENTS AND RESULTS

In this section, we first describe the used datasets. After that, we specify the experimental setup, including all hyper-parameters, and hardware specifications. Finally, We design our experiments to evaluate each of our contributions, in the form of an ablation study to evaluate the impact of each one.

A. Dataset

We use the extended version [11] of the publicly available KittiMoSeg dataset [13], that consists of 12919 frames which are split into 80% for training, and 20% for testing. The image resolution is 1242×375, and the labels determine whether the object is moving or static, includes the object bounding box and the motion mask.

B. Experimental Setup

We initialize our backbone networks with the weights pre-trained on ImageNet [3], then train the whole network for 30 epochs on COCO dataset [7] while freezing the backbone during the first 10 epochs. In all our experiments, ResNet-50 [4] was used. Our network is trained with Adam optimizer [6] with a scheduled learning rate that is decreased from $1e^{-3}$ to $1e^{-5}$, the whole network is end-to-end trained with learning rate exponentially decayed. We train a total of 200 epochs, using a warm-up learning rate of $1e^{-3}$ to $5e^{-3}$ in the first 5 epochs. 460×140 resolution images have been used across all the experiments. Our approach is implemented in Python using PyTorch on a PC with Intel Xeon(R) 4108 1.8GHz CPU, 64G RAM, Nvidia Titan-XP.

C. Motion Features

Previous works on MOD [12, 13] indicates that input features can have a strong impact on the results. In particular, features holding motion cues can be of high impact. Thus, we evaluate the best input features at each time step, where we compare RGB, RGB+RGB, and RGB+OF options. In this setup, we use the vanilla 1-step DETR architecture. Optical flow is generated using FlowNet 2.0 [5]. The results are infavor of the RGB+OF setup as shown in Table I.

D. Early vs Late temporal aggregation

In this setup, we evaluate the two architectural alternatives in Figure 2. For the sake of comparison, we fix the time window $T = 2$, the number of queries $N_q = 100$ and the transformer hidden dimension $d = 256$. Results are shown in Table II. Both results of early and late architectures improve over the 1-step baseline. However, the early architecture provides a significant improvement of 5% mAP.

E. Effect of TPE

In this experiment, we evaluate the addition of TPE. Building on the results of early temporal aggregation in Table II, we perform this comparison on the early temporal setup as shown in Figure 2. As expected, results in Table III, show 2% mAP improvement over the variant without TPE.

F. Effect of the temporal window size T

In this setup, we evaluate the effect of the increased window size, for $T = 1, 2, 4$. Results in Table IV show increased performance with the increase of T. However, a saturation barrier is hit at $T = 4$.

V. CONCLUSION

In this work, we extend the vanilla DETR architecture, into a Spatio-Temporal model to deal with video inputs. We explore various design choices in our endeavor; the early vs. late temporal aggregation setups. Results are in favor of the early architecture which deals with temporal traces of spatial motion features. Our analysis of the 1-step motion features suggests that the best option is to feed the RGB+OF frames of the input 1-step scene, which is also in line with previous works. We also propose an extra Temporal Positional Embedding (TPE) step, to enable the temporal differentiation of features. Results show improved performance with TPE introduced to the architecture. The new ST-DETR architecture achieves 5% mAP improvement on the KITTI MOD dataset.

REFERENCES

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *arXiv preprint arXiv:2005.12872*, 2020.

[2] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L Yuille, and Yuyin Zhou. Transunet: Transformers make strong encoders for medical image segmentation. *arXiv preprint arXiv:2102.04306*, 2021.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. Ieee, 2009.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 770–778, 2016.

[5] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017.

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[7] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[8] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[9] Eslam Mohamed and Ahmad El-Sallab. Modetr: Moving object detection with transformers. *arXiv preprint arXiv:2106.11422*, 2021.

[10] Eslam Mohamed, Mahmoud Ewaisha, Mennatullah Siam, Hazem Rashed, Senthil Yogamani, and Ahmad El-Sallab. Instancemotseg: Real-time instance motion segmentation for autonomous driving. *arXiv preprint arXiv:2008.07008*, 2020.

[11] Hazem Rashed, Mohamed Ramzy, Victor Vaquero, Ahmad El Sallab, Ganesh Sistu, and Senthil Yogamani. Fusemodnet: Real-time camera and lidar based moving object detection for robust low-light autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[12] Mennatullah Siam, Heba Mahgoub, Mohamed Zahran, Senthil Yogamani, Martin Jagersand, and Ahmad El-Sallab. ModNET: Moving object detection network with motion and appearance for autonomous driving. *arXiv preprint arXiv:1709.04821*, 2017.

[13] Mennatullah Siam, Heba Mahgoub, Mohamed Zahran, Senthil Yogamani, Martin Jagersand, and Ahmad El-Sallab. MODNet: Motion and appearance based moving object detection network for autonomous driving. In *Proceedings of the 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2859–2864, 2018.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[15] Enze Xie, Wenjia Wang, Wenhai Wang, Peize Sun, Hang Xu, Ding Liang, and Ping Luo. Segmenting transparent object in the wild with transformer. *arXiv preprint arXiv:2101.08461*, 2021.

[16] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. *arXiv preprint arXiv:2012.15840*, 2020.