Efficient Joint Detection and Multiple Object Tracking with Spatially Aware Transformer

Siddharth Sagar Nijhawan, Leo Hoshikawa, Atsushi Irie, Masakazu Yoshimura,
Junji Otsuka, Takeshi Ohashi
{Siddharth.Nijhawan, Leo.Hoshikawa, Atsushi.Irie, Masakazu.Yoshimura
Junji.Otsuka, Takeshi.A.Ohashi}@sony.com
Sony Group Corporation

Abstract

We propose a light-weight and highly efficient Joint Detection and Tracking pipeline for the task of Multi-Object Tracking using a fully-transformer architecture. It is a modified version of TransTrack, which overcomes the computational bottleneck associated with its design, and at the same time, achieves state-of-the-art MOTA score of 73.20%. The model design is driven by a transformer based backbone instead of CNN, which is highly scalable with the input resolution. We also propose a drop-in replacement for Feed Forward Network of transformer encoder layer, by using Butterfly Transform Operation to perform channel fusion and depth-wise convolution to learn spatial context within the feature maps, otherwise missing within the attention maps of the transformer. As a result of our modifications, we reduce the overall model size of TransTrack by 58.73% and the complexity by 78.72%. Therefore, we expect our design to provide novel perspectives for architecture optimization in future research related to multi-object tracking.

1. Introduction

The actively studied task of Multi-Object Tracking (MOT) serves a variety of use-cases where the algorithm has to be deployed on embedded systems with a light-weight implementation. Recent works like TransTrack [10] and TransCenter [17] achieve state-of-the-art MOT performance, however, struggle to solve the computational bottleneck, resorting to HW inefficient operations such as Deformable Convolutions [17]. Therefore, there is a huge scope of improving the existing MOT architectures to build efficient models having low deployment cost in terms of size and complexity while maintaining good tracking accuracy.

MOT is computationally heavy because it involves multiple tasks: classification, object detection, and tracking. This bottleneck can be addressed with Joint Detection and Tracking (JDT) paradigm, which performs detection and tracking simultaneously in a single stage [6]. JDT trains each component jointly in a single end-to-end pipeline and achieves a significantly faster model speed, thus, making it as a suitable choice for designing a light-weight MOT architecture.

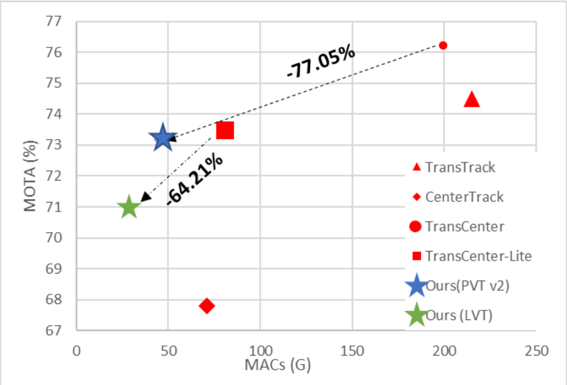

Figure 1. Comparison of our models with state-of-the-art across MACs vs. MOTA.

CNNs have been widely used to perform MOT by operating locally and greedily [15, 16]. Such methods perform pairwise association between newly detected objects to existing set of confirmed tracks through distance metrics. However, several techniques have recently emerged which leverage transformers [12] and the query-key mechanism of JDT to perform MOT. TransTrack [10] introduces a set of learned object queries for performing detection of incoming objects using transformer encoder-decoder modules with CNN backbone and outperforms state-of-the-art architectures in MOT17 challenge [8]. A single training pipeline, fast post-processing, modular architecture, and high tracking performance favors TransTrack [10] to be a good baseline in the family of JDT architectures.

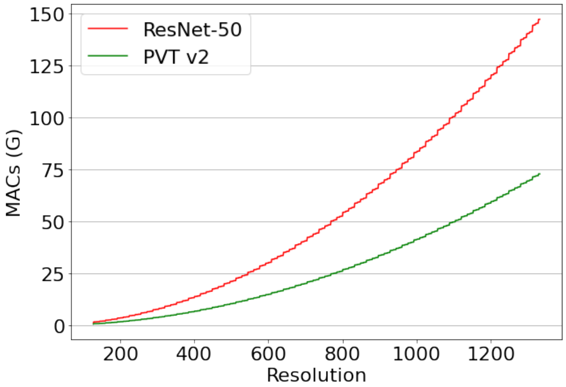

Figure 2. Input Resolution vs. MACs. PVT v2 [14] requires almost half the MACs for a high input resolution of (1333×1333) and the slope gradually rises compared to ResNet-50 [4].

However, for the baseline of TransTrack [10], an input of shape (800×1333) requires a total of 46.87M parameters and 215.23G Multiplication Addition Accumulation Operations (MACs), which is a huge computational bottleneck. Therefore, we design a light-weight version of TransTrack [10] which overcomes the bottleneck with minimal to no impact on the tracking performance an achieve a good reduction compared to state-of-the-art Fig. 1.

By performing layer-by-layer profiling of TransTrack [10] in terms of parameters and MACs, we identified 2 key bottlenecks requiring 78% total parameters with 92% MACs: backbone and encoder, as showcased in Fig. 4. For the backbone, we propose PVT v2 [14] as a direct replacement for existing backbone of ResNet-50 [4] in TransTrack [10]. PVT v2 [14], an improved version of PVT [13], reduces the computational complexity to linear and achieves significant improvements on fundamental vision tasks through a progressive shrinking pyramid. Fig. 2 validates the scalability of PVT v2 [14] (b1 configuration) in comparison to ResNet-50 [4] when the input resolution is increased. Therefore, use of PVT v2 [14] backbone leads to a significant reduction in model size and complexity. Another efficient backbone called LVT [18] introduces convolution in self-attention operation to extract low-level features for dense prediction tasks.

For the encoder, we propose a new block based on depth-wise convolution similar to MobileNet [5]. Our block is inspired by the fact that encoder contains self-attention mechanism to find global dependencies among feature map, but lacks the understanding of spatial information contained within these maps due to flattening. We replace the feed forward network (FFN) of the encoder with 3 sub-blocks: 2 channel fusion blocks using Butterfly Transform (BT) [11], and a depth-wise convolution block. BT [11] is a drop-in replacement for channel fusion operation with logarithmic complexity rather than quadratic. The proposed block reduces the complexity significantly by 86.76%. To sum up, the contributions of this paper are as follows:

- We performed layer-by-layer profiling of TransTrack [10] to identify key computational bottlenecks in its architecture.
- Through ablation studies, we showcased that CNN backbone is not computationally efficient for high input resolution and a transformer-based backbone scales better in generating feature maps for dense prediction tasks.
- We solved the computational bottleneck associated with encoder block by proposing a block which requires 74.3% less MACs compared to a standard transformer encoder.
- By applying the above mentioned modifications, our architecture contains 58.73% less parameters and requires 78.72% less MACs in comparison to TransTrack [10], while achieving state-of-the-art MOT score of 73.20%. To our knowledge, this is the first fully-transformer based JDT pipeline. In conclusion, our model is an extremely light-weight version of TransTrack [10] with small deployment costs.

2. Proposed Methodology

The proposed architecture is described in Fig. 3 which is based on TransTrack [10] with several improvements to reduce the computational complexity and model size.

2.1. Problem Formulation

To identify the areas of improvement in TransTrack [10], we passed a random tensor of shape $(3 \times 800 \times 1333)$ and profiled the number of trainable parameters per layer and total MACs required for a single forward pass. Fig. 4 shows the percentage distribution of these metrics across 5 key layers: backbone, feature aggregation, encoder, decoder, and output. We conclude that model size is majorly dependent on ResNet-50 [4] backbone with 23.23M parameters consisting of 65% of the total size. In terms of computational complexity, the key bottleneck is the encoder which requires 49% of total MACs. Therefore, our architecture optimizes two key components of TransTrack [10] to make it light-weight; backbone and encoder. Next section describes each of the components of our proposed architecture in detail.

2.2. Architecture Components

This section describes each component for extracting spatial feature maps (backbone), performing object detection (encoder), generating tracking boxes (detection), and finally associating them to form final set of output boxes (tracking and association).

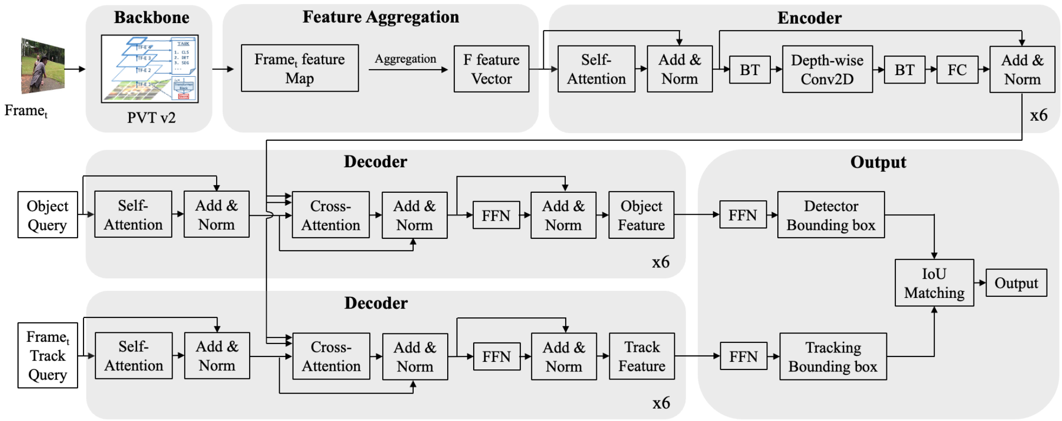

Figure 3. Architecture of proposed Multi-object Tracker.

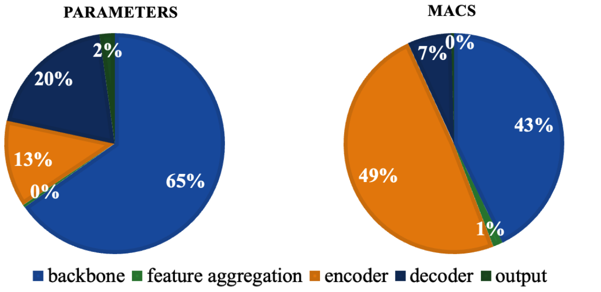

Figure 4. Layer-by-layer profiling of TransTrack [10] architecture in terms of Trainable Parameters and MACs required for a single forward pass.

Backbone. PVT v2 [14] backbone of our model incorporates a progressive shrinking pyramid by uniformly reducing spatial dimensions at each stage which reduces the length of input embedding sequence as we go deeper into the network, therefore, reducing the computational cost associated with it. To optimize the computation of feature map extraction, we temporarily save the extracted maps of current frame to be used for the next frame.

Encoder. We propose a modification in transformer encoder structure of TransTrack [10]. The FFN of encoder is computationally very expensive because it expands the dimensions by a factor of 8. Also, the attention mechanism of encoder lacks the understanding of spatial context among image patch tokens. Therefore, we replace the first fully connected (FC) layer with a structure shown in Fig. 5 containing three sub-blocks: two BT operations and a single 3×3 depth-wise convolution block. After generating a set of tokens from previous Multi-head Self Attention (MHSA) block, the patch tokens are restored back to spatial dimensions with shapes relative to four feature maps generated by the backbone. Then we perform depth-wise separable convolution [1], however, performing channel fusion with BT blocks rather than 1×1 point-wise convolution. We use a 3×3 kernel for performing depth-wise convolution, enhancing the correlation among neighboring 8 pixels for spatial representation. After fusing spatial context, we flatten these tokens into original sequence of shape $B \times N \times C$, where B is the batch size, N is the sequence length, and C is the total number of channels. Finally, we add a linear projection layer without expanding the channel dimensions.

Here, we use BT [11] operation to replace channel fusion in depth-wise separable convolution block to reduce the computational complexity of 1×1 point-wise convolution which is considerably higher than spatial fusion operation [5]. BT reduces the space and time complexity from $O(N^2WH)$ to $O((NlogN)WH)$, where N is the number of input and output channels. Therefore, our block combines the advantages of both CNN and transformer into a single encoder module by extracting local information and keeping the self-attention portion unchanged. It also reduces the number of trainable parameters in each encoder block from 4.54M in TransTrack [10] to 602.11K. To perform a single forward pass, the encoder stack in TransTrack [10] required 100.49G MACs, whereas, our block requires 13.30G MACs which is a huge 86.76% reduction.

Detection. Trajectory Query Decoder performs object detection to generate detection bounding boxes based on the architecture proposed in Deformable DETR [22] without any modifications.

Tracking and Association. This stage takes the input as detected object in the previous frame and passes them as object features on to the next frame. Decoder takes both spatial features along with location based features of objects in the previous frame, and looks up the coordinates of these objects in the current frame to output tracking bounding boxes. Finally, we perform pairwise mapping of the boxes generated from both decoder stages using IoU mapping and generate the output bounding boxes. The boxes which have no match are initialized as new object tracks.

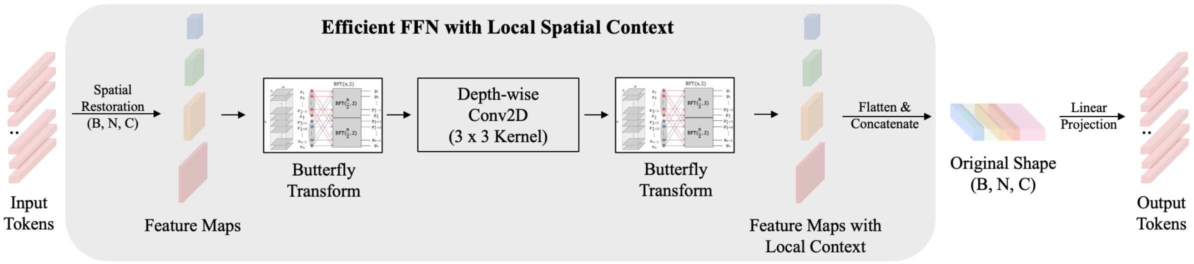

Figure 5. Proposed replacement for encoder's FFN.

Table 1. Quantitative Evaluation on MOT17 private test set.

Model	MOTA↑	FP↓	FN↓	IDS↓	Params↓	MACs↓
TransTrack [10]	74.50	28,323	112,137	3,663	46.87M (-0%)	215.23G (-0%)
GTR [21]	75.30	36,231	93,150	**2,346**	43.80M	-[1]
CenterTrack [20]	67.80	**18,489**	160,332	3,039	19.32M	70.88G
FairMOT [19][2]	73.70	27,507	117,477	3,303	19.71M	84.98G
TransCenter [17]	**76.20**	40.101	**88.827**	5.394	35.1M	199.59G
TransCenter-Lite [17]	73.50	-	-	-	8.1M	80.53G
Ours (PVT v2 [14])	73.20	28,341	118,689	4,218	19.34M (-58.73%)	45.80G (-78.72%)
Ours (LVT [18])	71.00	32,730	125,274	5,757	**6.07M (-87.04%)**	**28.82G (-86.60%)**

3. Experiments

3.1. Training Setup

Datasets and Training Pipeline. Training pipeline utilizes 2 datasets - CrowdHuman [9] and MOT17 [8]. We also use a subset of MOT17 data called 'MOT17 half-half' split for hyper-parameter tuning which uses sequences of a single detector having first half of video as training and second half as validation data. The training pipeline is similar to TransTrack [10] - pre-training the backbone; pre-training encoder and detector on CrowdHuman [9]; and finally, training end-to-end MOT model on a mix of CrowdHuman and MOT17. The evaluation is performed using MOT17's private test set.

3.2. MOT17 Benchmark

Table 1 contains the evaluation results on MOT17 private test set using MOTA score for the proposed model along with several state-of-the-art MOT architectures. Our model with PVT v2 [14] backbone achieves performance comparable to state-of-the-art methods with considerably lower model size and computational complexity. It has 58.73% less trainable parameters and requires 78.72% less MACs for a single forward pass in comparison to the baseline of TransTrack [10] with only 1.3% drop in MOTA score. Compared to FairMOT, our proposal do not require heavy post-processing. Compared to TransCenter [17], a transformer-based model, our model obtain a slightly inferior accuracy but at much lower params and MACs. We also test an extended version of our model which contains LVT [18] backbone instead of PVT v2 [14] and it further reduces the overall parameter size to 6.07M with 28.82G MACs. Both of our models outperform CNN based CenterTrack [20] with a relative improvement of 5.4% and 3.2% MOTA score for PVT v2 [14] and LVT [18] backbones respectively. We also achieve excellent FP and FN denoting that most of the objects within sequences were successfully detected. In terms of IDS, our models under-perform, however, it is a future direction of research to improvise the performance of our JDT architecture.

4. Conclusion

We proposed a JDT MOT pipeline which is a lightweight and highly efficient version of TransTrack [10] architecture. It uses a transformer backbone to extract multi-scale feature maps, a novel replacement of FFN network which performs channel fusion with logarithmic complexity, and learns the spatial context within feature maps. The proposed architecture achieves 73.20% MOTA on MOT17 dataset with 58.73% less trainable parameters and 78.72% less MACs in comparison to TransTrack [10]. Our method is the first work solving MOT with a fully-transformer based light-weight JDT pipeline and we expect it to provide future direction for research in domain of architecture optimization for several dense prediction tasks.

References

[1] François Chollet. Xception: Deep learning with depthwise separable convolutions, 2016. 3, 6

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 6

[3] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. 6

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 2, 6

[5] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017. 2, 3

[6] Hilke Kieritz, Wolfgang Hübner, and Michael Arens. Joint detection and online multi-object tracking. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1540–15408, 2018. 1

[7] Youngkeun Lee, Sang-ha Lee, Jisang Yoo, and Soonchul Kwon. Efficient single-shot multi-object tracking for vehicles in traffic scenarios. *Sensors*, 21(19), 2021. 6

[8] Anton Milan, Laura Leal-Taixe, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking, 2016. 1, 4

[9] Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. Crowdhuman: A benchmark for detecting human in a crowd. *arXiv preprint arXiv:1805.00123*, 2018. 4, 6

[10] Peize Sun, Jinkun Cao, Yi Jiang, Rufeng Zhang, Enze Xie, Zehuan Yuan, Changhu Wang, and Ping Luo. Transtrack: Multiple object tracking with transformer, 2020. 1, 2, 3, 4, 6, 7

[11] Keivan Alizadeh Vahid, Anish Prabhu, Ali Farhadi, and Mohammad Rastegari. Butterfly transform: An efficient fft based neural architecture design, 2019. 2, 3

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 1

[13] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions, 2021. 2

[14] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. PVT v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, mar 2022. 2, 3, 4, 6, 7

[15] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards real-time multi-object tracking, 2019. 1

[16] Nicolai Wojke and Alex Bewley. Deep cosine metric learning for person re-identification. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, mar 2018. 1

[17] Yihong Xu, Yutong Ban, Guillaume Delorme, Chuang Gan, Daniela Rus, and Xavier Alameda-Pineda. Transcenter: Transformers with dense representations for multiple-object tracking, 2021. 1, 4

[18] Chenglin Yang, Yilin Wang, Jianming Zhang, He Zhang, Zijun Wei, Zhe Lin, and Alan Yuille. Lite vision transformer with enhanced self-attention, 2021. 2, 4, 6, 7

[19] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. FairMOT: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, 129(11):3069–3087, sep 2021. 4

[20] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points, 2020. 4

[21] Xingyi Zhou, Tianwei Yin, Vladlen Koltun, and Philipp Krähenbühl. Global tracking transformers, 2022. 4

[22] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection, 2020. 3

Table 2. Ablation study on architecture backbone.

Backbone	MOTA↑	Params↓	MACs↓
ResNet-50 [4]	**66.30**	30.12M	80.04G
PVT v2 [14]	65.70	19.34M	45.80G
LVT [18]	64.60	**6.07M**	**28.82G**

Table 3. Ablation study on number of encoders.

Encoders	MOTA↑	Params↓	MACs↓
1	49.10	19.15M	37.76G
3	49.80	19.43M	49.80G
6	67.00	19.84M	62.18G

Table 4. Ablation study on number of decoders.

Decoders	MOTA↑	Params↓	MACs↓
1	55.10	15.53M	42.82G
3	64.45	17.25M	44.01G
6	67.00	19.84M	62.18G

A. Implementation details

We initialize the backbone with weights obtained through pretraining on ImageNet [2], shared publicly by PVT v2 [14]. For the remaining portion of the architecture including detector and tracker, we use Xavier-init [3] to initialize the weights, and fine-tune the model using AdamW [1] optimizer. The initial learning rate is set to 2e-4. Similar to TransTrack [10], we apply various data augmentation techniques of random cropping, scaling, and re-sizing the inputs ranging from (480×800) to (800×1333) pixels. The learning rate scheduler is set to drop by a factor of 10 at 100th epoch, with a total of 150 epochs for training. Following the TransTrack [10] procedure, the end-to-end model is first pre-trained on CrowdHuman [9] and finally, fine-tuned on MOT17.

B. Design choice experiments

This section covers various ablation studies we performed on our key design choices. All the experiments are conducted on MOT17 half-half split with the setup described in 3.1.

B.1. Backbone

We ablate the effect of changing the backbone in our architecture with 3 design choices: ResNet-50 [4], PVT v2 [14] and LVT [18]. The quantitative results are shown in Table 2. LVT [18] backbone is the most light-weight and efficient out of our backbone choices, with only 6.07M parameters and 28.82G MACs, however, achieves 64.60% MOTA on MOT17 half-half test set. PVT v2 [14] slightly performs better, however, consumes more than 50% parameters and MACs. Choosing the most widely used backbone of ResNet-50 [4] is highly inefficient, with 80.04G MACs and 30.12M parameters, proving our hypothesis that transformer backbones scale better with resolution as compared to CNN backbones.

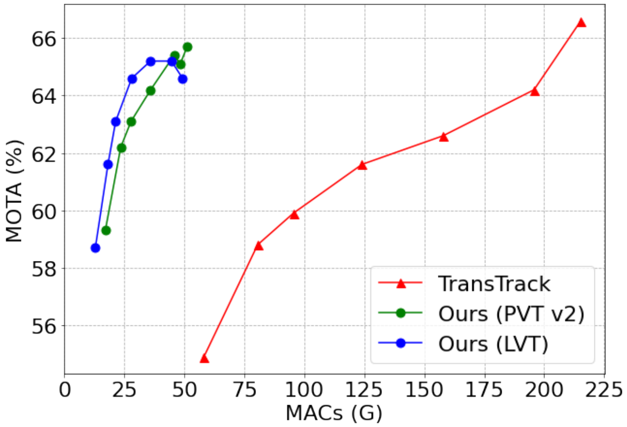

Figure 6. Ablation study on impact of input resolution on MOTA and MACs required for a single forward pass.

B.2. Number of encoders and decoders

Using the proposed architecture with PVT v2 [14] backbone, we study the impact of number of encoders and decoders on MOTA, model size and efficiency in Table 3 and Table 4 respectively. Using a single encoder reduces the performance significantly, achieving only 49.10% MOTA instead of 67.00% for a total of 6 encoders. Similar trend is observed for decoders, where a single decoder achieves 55.10% MOTA with a total of 42.82G MACs. To maximize the performance in terms of MOTA, we choose 6 encoders and 6 decoders as default for our experiments.

B.3. Input Resolution

The computational complexity of the model is largely dependent on the input resolution. Also, larger the input resolution, the better the tracking performance [7]. To validate this, we ablate the dimensions of input across 7 choices between (400×666) and (800×1333).,Dimension (800×1333) is the inference resolution used by TransTrack [10]. Fig. 6 shows the relationship between the tracking performance and MACs for chosen input resolutions. Both our models are highly efficient across all the choices of resolutions compared to TransTrack [10]. For an input resolution

[1] Due to incompatibilities in the implementation, we could not compute the MACs for GTR.

[2] FairMOT does not include post-processing computation

of (800×1333), our model with PVT v2 [14] backbone requires 51.23G MACs and with LVT [18] backbone requires 49.34G MACs. On the other hand, TransTrack [10] needs approximately 4 times more resources at this resolution, with 215.23G MACs. In terms of MOTA score, TransTrack [10] performs slightly better for higher resolutions of (760×1266) and (800×1333), however, is computationally very heavy. For smaller resolutions, both our models outperform TransTrack [10] when tested on MOT17 half-half set and achieve very close MOTA score for the highest resolution. Therefore, our architecture is highly scalable across all the tested resolutions in comparison to TransTrack [10].