Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Sec. 5 contains our future work.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] We included potential misuses in Sec. 5.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] We stated the assumptions and motivations of our work in Sec. 1 and Sec. 3.1.
   (b) Did you include complete proofs of all theoretical results? [Yes] In addition to the Big-O complexity in Sec. 3.1, we included actual results in Sec. 4.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We listed the data and instructions in Sec. 4 and Appendix, and the code will be released upon acceptance.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We specified details in Sec. 4 and Appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] In Appendix, we reported standard deviations of the results in Table 3 (a, c, d).
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We included the type of resources used for training and inference in Appendix and Table 2.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We cited the papers and repositories that are used.
   (b) Did you mention the license of the assets? [Yes] We mentioned the licenses at the end.
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A] New assets are not included.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] Not the scope of this paper.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] Not the scope of this paper.
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] Not the scope of this paper.

A Appendix

We provide further details needed for training and inference in the appendix.

A.1 Implementation Details

For the training, we use 8 Tesla V100 GPUs with 16GB memory. As noted in Table 2, we used a single RTX 2080Ti GPU for measuring FPS of the main results (see Sec. 4.1). However, 16GB memory is not sufficient for evaluating the model with full self-attention over space-time inputs. Therefore, we used a single Tesla V100 GPU with 32GB memory for completing the results in Table 3.

We used `detectron2` [34] for our code basis, and hyper-parameters mostly follow the settings of DETR [13] unless specified. We used AdamW [36] optimizer with initial learning rate of 10^{-4} for transformers, and 10^{-5} for backbone. We first pre-train the model for image instance segmentation on COCO [35] by setting our model to $T = 1$. The pre-train procedure follows the shortened training schedule of DETR [13], which runs 300 epochs with a decay of the learning rate by a factor of 10 at 200 epochs. Using the pre-trained weights, the models are trained on targeted dataset using the batch size of 16, each clip composed of $T = 5$ frames downscaled to either 360p or 480p. The models are trained for 8 epochs, and decays the learning rate by 10 at 5th epoch. For the evaluation, the input videos are downscaled to 360p, and we average clip predictions of equal identities for the final results.

To balance the weights of class and mask predictions, we use $\lambda_0 = \lambda_1 = \lambda_2 = 3$. Sigmoid-focal loss uses $\alpha = 0.25, \gamma = 2$ to alleviate foreground-background pixel imbalance. Following CondInst [7], we upscale predicted masks to the stride of 4 with bilinear interpolation for computing mask-related losses. For the number of layers, we use $N_E = 3, N_D = 3$ where each transformer layer is of width 256 with 8 attention heads.

We include extra figure which specifies the details of our network (see Fig. 3). We freeze batch normalization layers [37] of the backbone due to small batch-sized input. In spatial decoder, each convolutional layer is followed by group normalization layer [38] except the last depthwise separable convolutional layer [39].

A.2 Qualitative Comparison

For comparison, we provide visualized outputs of our model in addition to that of MaskTrack R-CNN [1], SipMask [2], and VisTR [11] (see Fig. 4-8). The models are all built on top of ResNet-50, and we used official checkpoints offered from the authors. Moreover, we visualized attention maps of two memory tokens that have tendencies of attending foreground and background respectively. As shown in the results, our model shows superiority over the others under various situations such as: (a) fast instance movement, (b) overlaps between instances (c) instances of similar appearances (d) motion blurs.

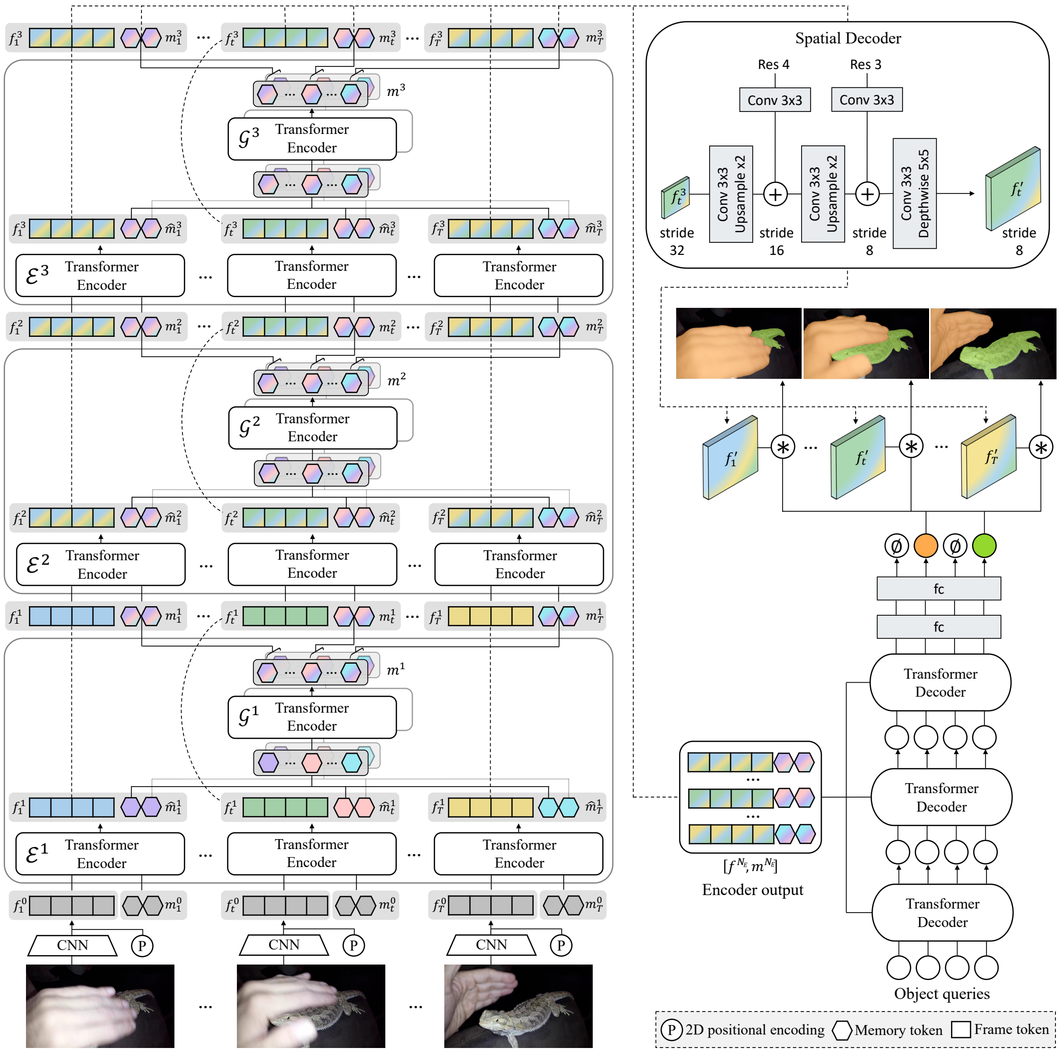

Figure 3: Further specifications of our network.

Table 4: Standard deviations of Table 3 that had to be omitted due to the space limit.

(a) Standard deviations of Table 3 (a)

	T=5		T=10		T=15		T=20	
	AP	AP_{75}	AP	AP_{75}	AP	AP_{75}	AP	AP_{75}
No Comm	1.0	1.5	1.2	1.3	1.1	1.3	1.0	1.2
Full THW	0.4	0.5	0.7	0.5	1.4	1.3	1.4	1.3
Decomp T-HW	1.4	1.4	1.0	1.0	1.0	1.3	1.1	1.3
IFC	1.6	2.1	1.1	1.9	1.0	1.5	1.2	1.8

(b) Standard deviations of Table 3 (c)

	T=5	T=10	T=15	T=20
M=1	1.1	0.9	1.6	1.3
M=2	1.1	0.6	0.6	1.0
M=4	0.8	0.9	0.6	0.7
M=8	1.6	1.1	1.0	1.2
M=16	1.5	0.9	0.4	0.7

(c) Standard deviations of Table 3 (d)

	T=5	T=10	T=15	T=20
Unified	0.5	0.6	0.5	0.6
Decomp	1.6	1.1	1.0	1.2

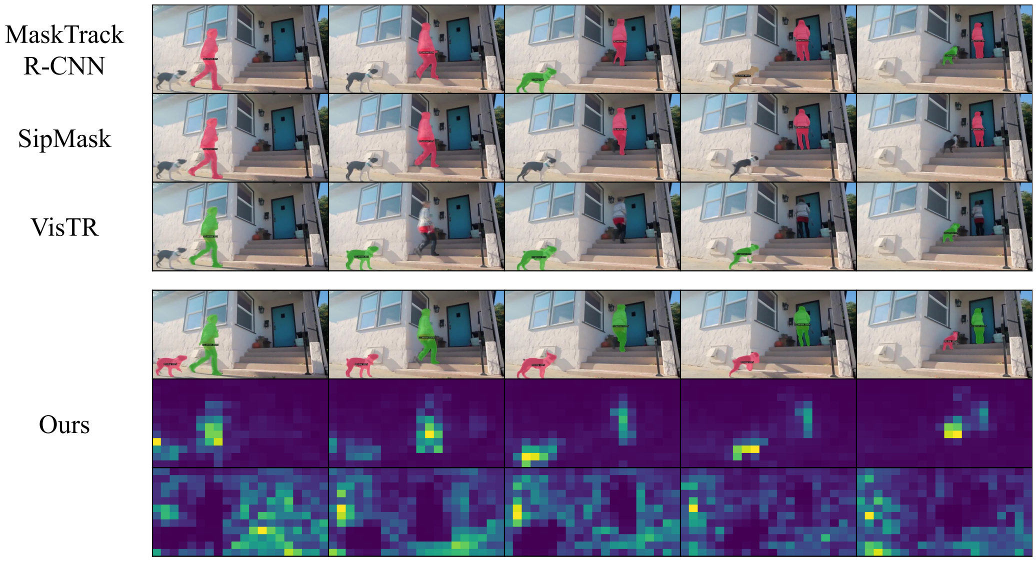

Figure 4: Qualitative results. Best viewed on screen.

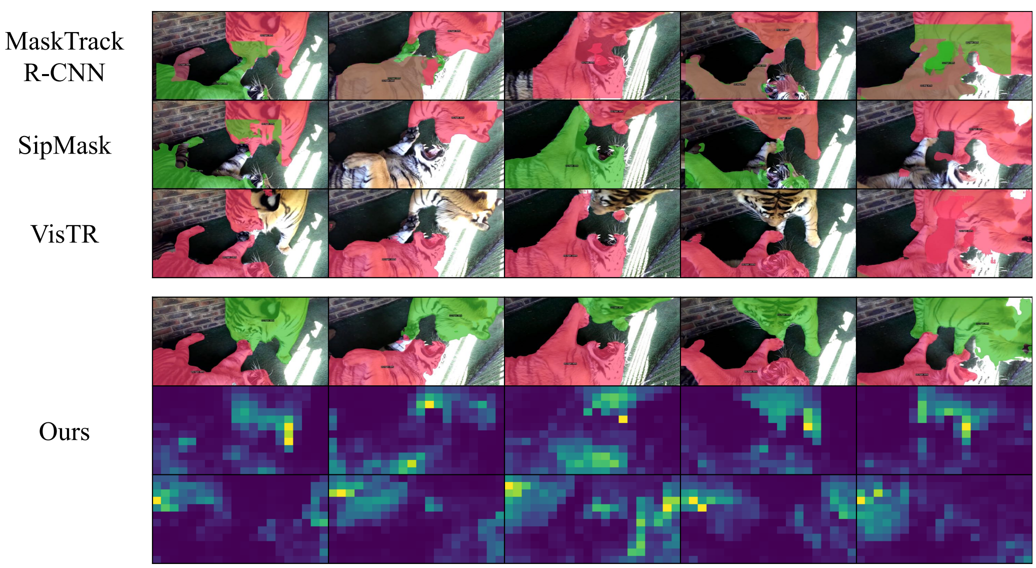

Figure 5: Qualitative results. Best viewed on screen.

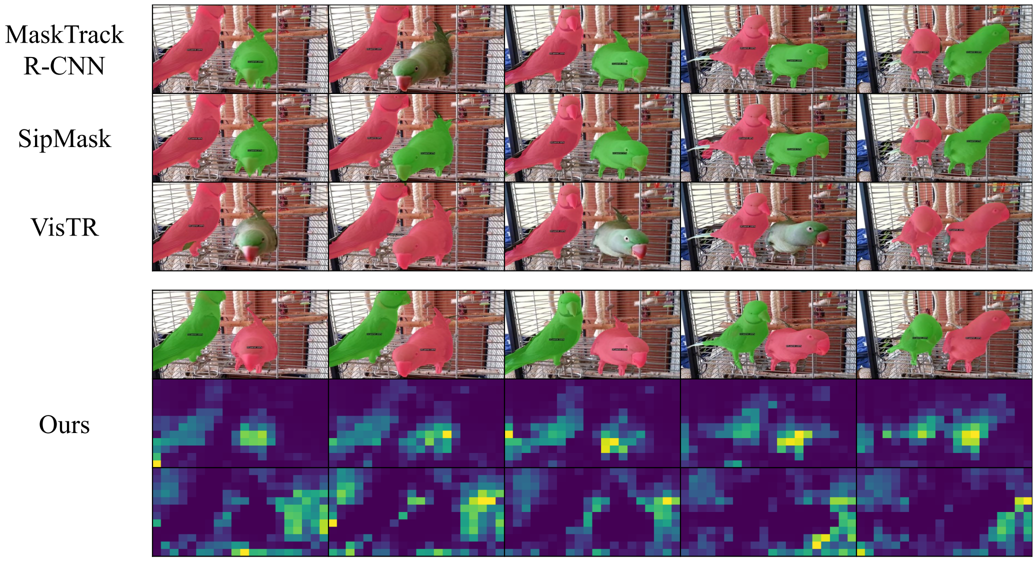

Figure 6: Qualitative results. Best viewed on screen.

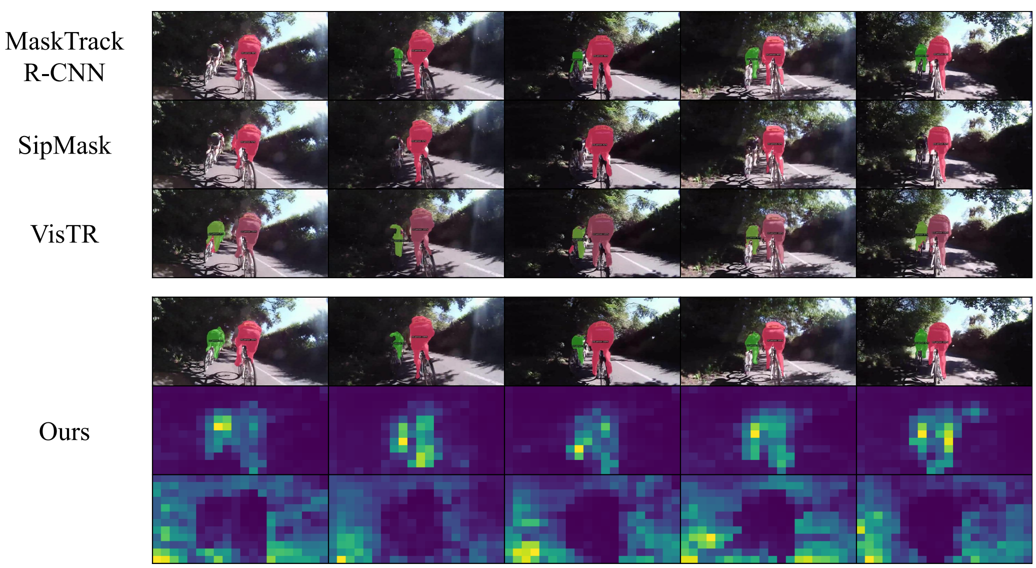

Figure 7: Qualitative results. Best viewed on screen.

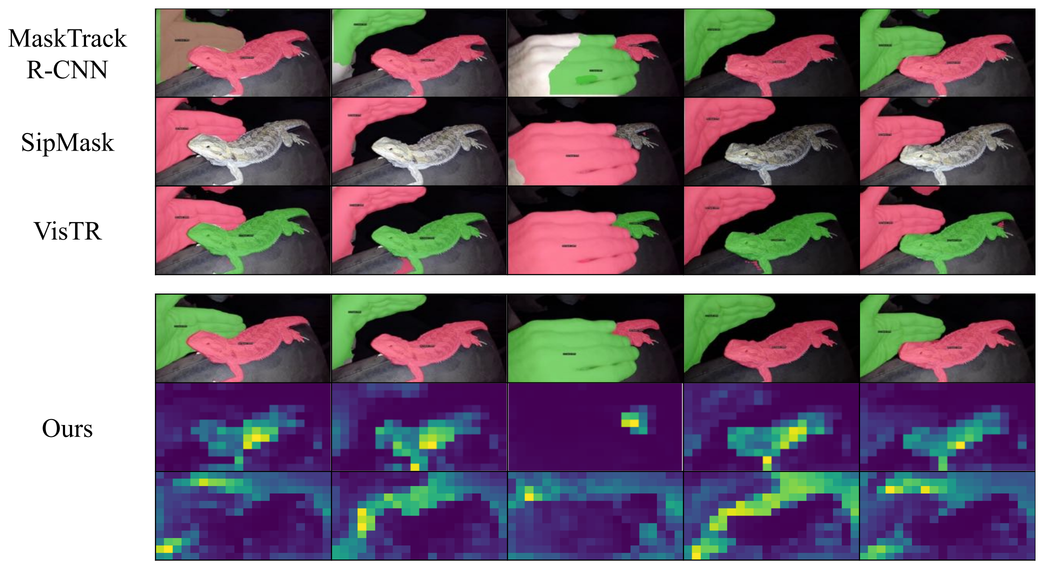

Figure 8: Qualitative results. Best viewed on screen.