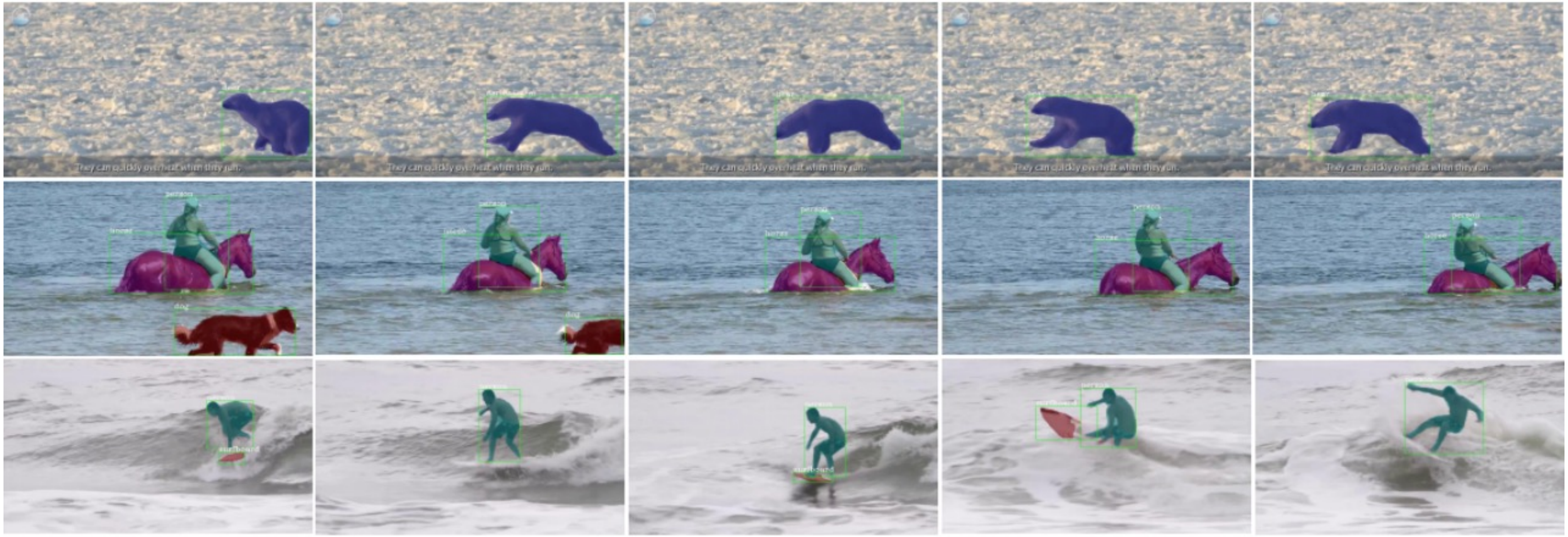
Video Instance Segmentation

- Simultaneous tracking、detection and segmentation
- Evaluation metric : temporal mAP

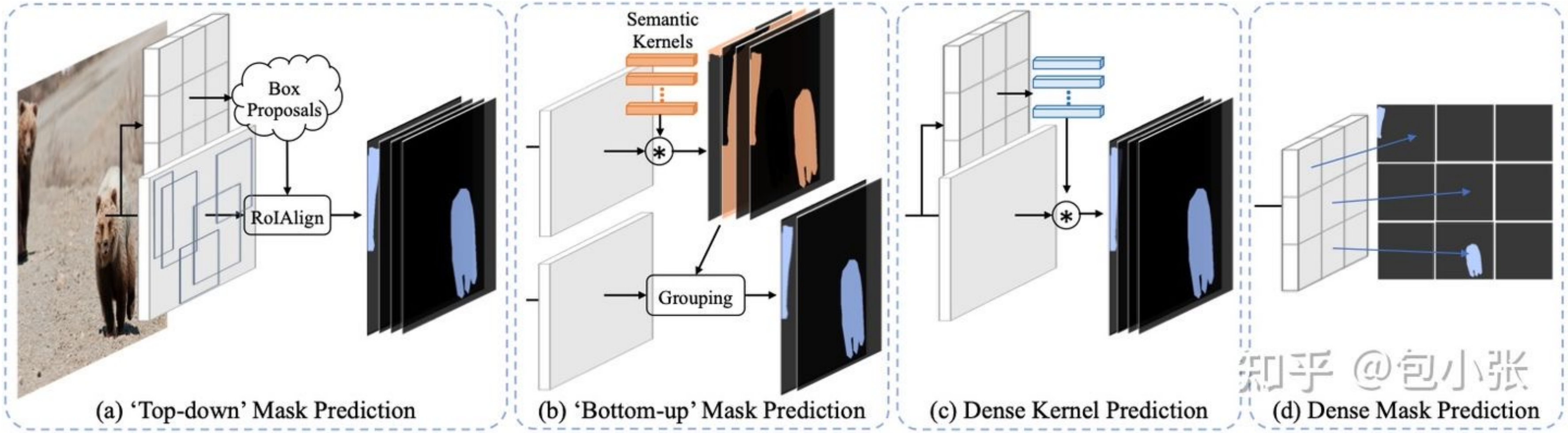
$$\mathbf{IoU}(i,j) = \frac{\Sigma_{t=1}^{T} |\mathbf{m}_t^i \cap \tilde{\mathbf{m}}_t^j|}{\Sigma_{t=1}^{T} |\mathbf{m}_t^i \cup \tilde{\mathbf{m}}_t^j|}$$

- Datasets: YouTube-VIS 2019/2021、OVIS



Past Work Overview

- Based on different instance segmentation method:

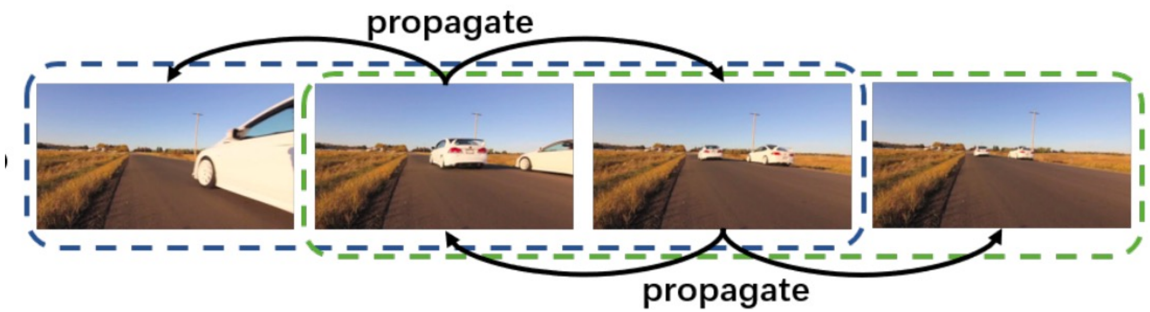


- Compared with instance segmentation method:
  - Need to fuse temporal information
  - Instances association may be required
  - Greater memory consumption

Past Work Overview

- Classified by tracking method:
  - Online (Frame-level) method:

  - Offline (Clip-level) method



Recent Research Trend

- Use temporal query to generate predictions
- Transformer-based more efficient spatiotemporal feature fusion
- More efficient data association / instance identification mechanism

Efficient Video Instance Segmentation via Tracklet Query and Proposal

Jialian Wu[1] Sudhir Yarram[1] Hui Liang[2] Tian Lan[2]
Junsong Yuan[1] Jayan Eledath[2] Gérard Medioni[2]

[1]State University of New York at Buffalo [2]Amazon

https://jialianwu.com/projects/EfficientVIS.html

Efficient VIS

- Clip level offline
- Query Based

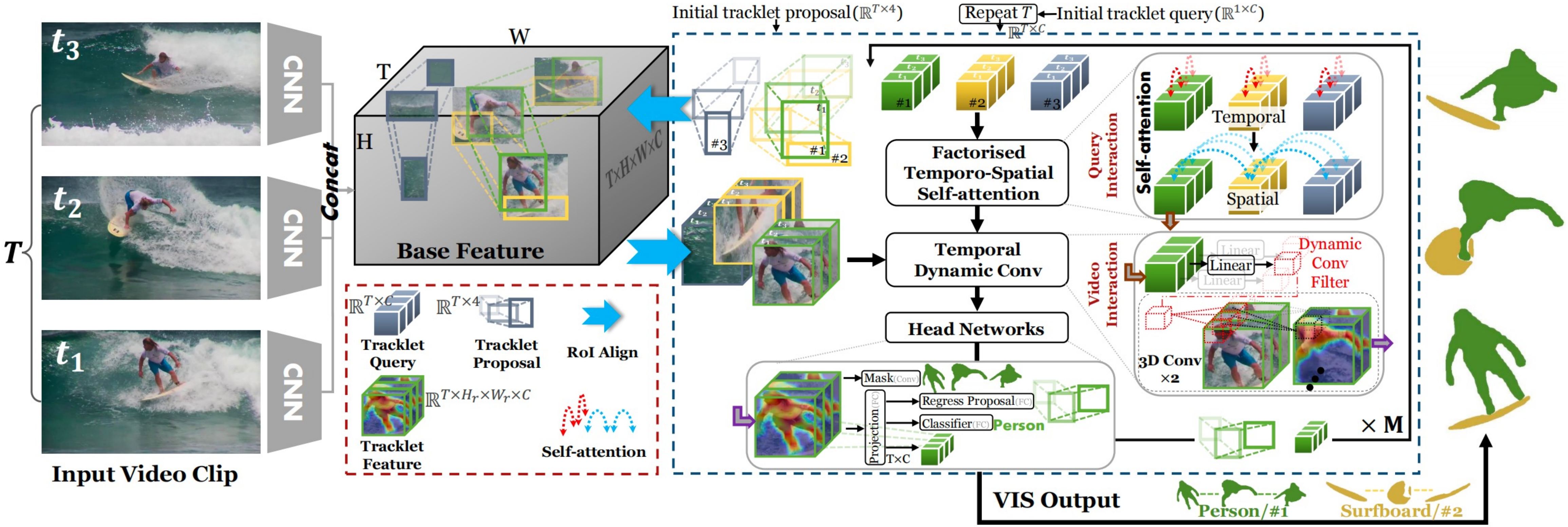
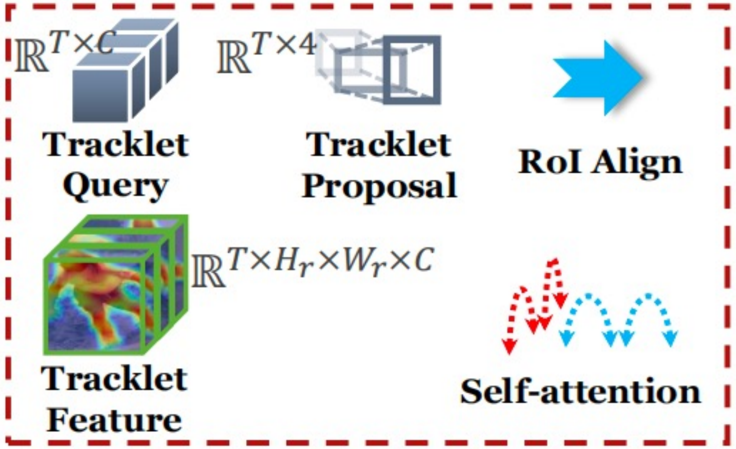


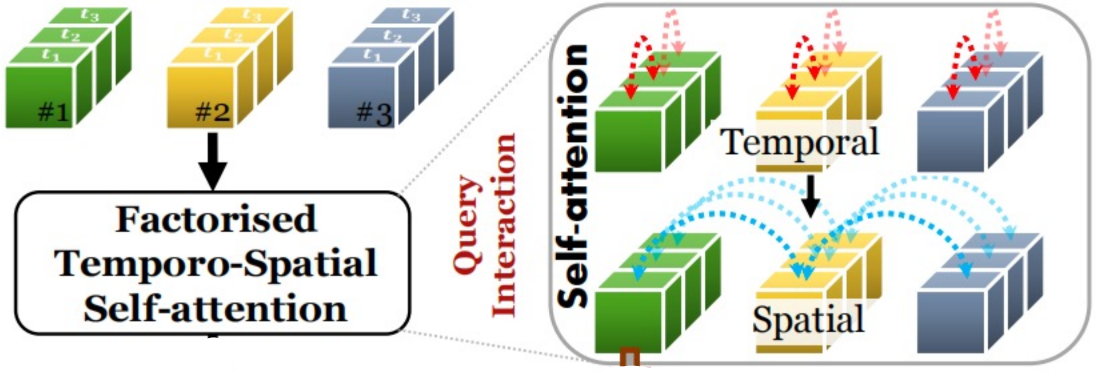
Figure 2. **EfficientVIS architecture.** EfficientVIS performs VIS clip-by-clip where the above figure illustrates how it works in one clip.

Efficient VIS

- Tracklet Query and Proposal Query Based：
  - tracklet queries ： $\{\boldsymbol{q}_i\}_{i=1}^{N}$ $\boldsymbol{q}_i \in \mathbb{R}^{T\times C}$
  - tracklet proposals： $\{\boldsymbol{b}_i\}_{i=1}^{N}$



- Factorised Temporo-Spatial Self-Attention (FTSA)：
  - separately perform temporal and spatial Multi-Head Self-Attention
  - saves more computation and more effective



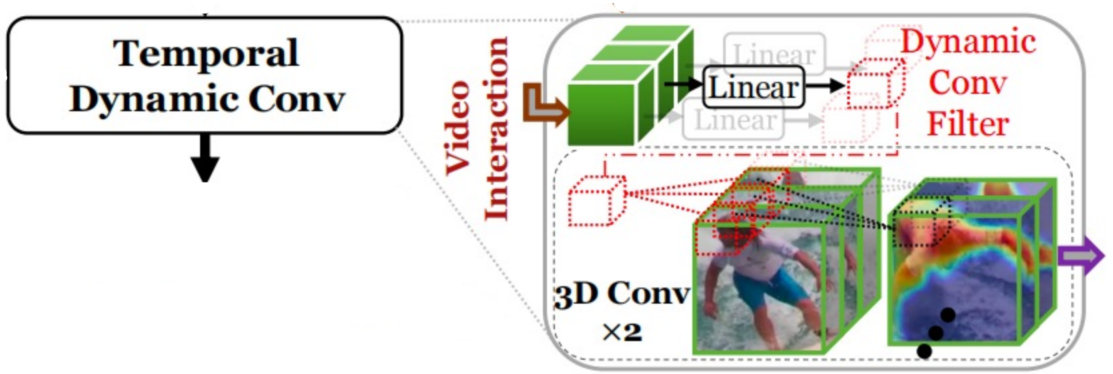
Efficient VIS

- Temporal Dynamic Convolution (TDC):
  - perform 3D dynamic convolution on an RoI region of the base feature

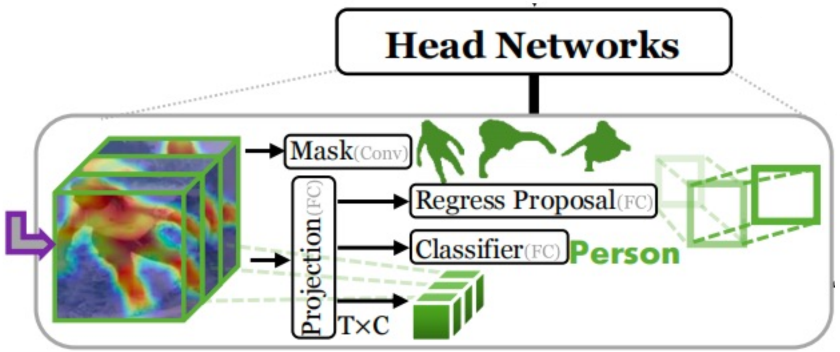
$$\boldsymbol{o}_i^t = \sum_{t'=t-1}^{t+1} \boldsymbol{a}_{i,(t,t')} \circ \mathrm{conv2d}(\boldsymbol{w}_i^t, \phi(\boldsymbol{f}^{t'}, \boldsymbol{b}_i^{t'}))$$

$$\{\boldsymbol{o}_i^t\}_{t=1}^T \in \mathbb{R}^{T \times H_r \times W_r \times C}$$

$$\boldsymbol{a}_{i,(t,t')} \in \mathbb{R}^{H_r \times W_r}$$



- Head Networks:



Efficient VIS

- YTVIS 2019:

Method	Publication	Augmentations	Backbone	FPS	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
MaskTrack R-CNN [28]	ICCV'19	✗	ResNet-50	33	30.3	51.1	32.6	31.0	35.5
SipMask [6]	ECCV'20	✗	ResNet-50	34	32.5	53.0	33.3	33.5	38.9
CompFeat [11]	AAAI'21	✗	ResNet-50	<33	35.3	56.0	38.6	33.1	40.3
TraDeS [26]	CVPR'21	✗	ResNet-50	26	32.6	52.6	32.8	29.1	36.6
QueryInst [10]	ICCV'21	✗	ResNet-50	32	34.6	55.8	36.5	35.4	42.4
CrossVIS [29]	ICCV'21	✗	ResNet-50	40	34.8	54.6	37.9	34.0	39.0
VisSTG [24]	ICCV'21	✗	ResNet-50	22	35.2	55.7	38.0	33.6	38.5
EfficientVIS (Ours)	CVPR'22	✗	ResNet-50	36	**37.0**	**59.6**	**40.0**	**39.3**	**46.3**
STMask [14]	CVPR'21	DCN backbone [9]	ResNet-50	29	33.5	52.1	36.9	31.1	39.2
SG-Net [16]	CVPR'21	multi-scale training	ResNet-50	23	34.8	56.1	36.8	35.8	40.8
VisTR [25]	CVPR'21	random crop training	ResNet-50	30	35.6	56.8	37.0	35.2	40.2
QueryInst [10]	ICCV'21	multi-scale training	ResNet-50	32	36.2	56.7	39.7	36.1	42.9
CrossVIS [29]	ICCV'21	multi-scale training	ResNet-50	40	36.3	56.8	38.9	35.6	40.7
VisSTG [24]	ICCV'21	multi-scale training	ResNet-50	22	36.5	58.6	39.0	35.5	40.8
EfficientVIS (Ours)	CVPR'22	multi-scale training	ResNet-50	36	**37.9**	**59.7**	**43.0**	**40.3**	**46.6**
MaskTrack R-CNN [28]	ICCV'19	✗	ResNet-101	29	31.9	53.7	32.3	32.5	37.7
SRNet [30]	ACMMM'21	✗	ResNet-101	35	32.3	50.2	34.8	32.3	40.1
CrossVIS [29]	ICCV'21	✗	ResNet-101	36	36.6	57.3	39.7	36.0	42.0
EfficientVIS (Ours)	CVPR'22	✗	ResNet-101	32	**38.7**	**61.3**	**44.0**	**40.6**	**47.7**
SipMask [6]	ECCV'20	multi-scale training	ResNet-101	24	35.8	56.0	39.0	35.4	42.4
STMask [14]	CVPR'21	DCN backbone [9]	ResNet-101	23	36.8	56.8	38.0	34.8	41.8
SG-Net [16]	CVPR'21	multi-scale training	ResNet-101	20	36.3	57.1	39.6	35.9	43.0
VisTR [25]	CVPR'21	random crop training	ResNet-101	28	38.6	61.3	42.3	37.6	44.2
EfficientVIS (Ours)	CVPR'22	multi-scale training	ResNet-101	32	**39.8**	**61.8**	**44.7**	**42.1**	**49.8**

Efficient VIS

- YTVIS 2021:

Method	Publication	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
MaskTrack R-CNN [28]	ICCV'19	28.6	48.9	29.6	26.5	33.8
SipMask⋆ [6]	ECCV'20	31.7	52.5	34.0	30.8	37.8
CrossVIS [29]	ICCV'21	33.3	53.8	37.0	30.1	37.6
EfficientVIS (Ours)	CVPR'22	**34.0**	**57.5**	**37.3**	**33.8**	**42.5**

Self-attention	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
Spatial (S)	32.8	56.3	34.9	36.1	42.2
Temporal (T)	30.4	49.0	32.1	34.2	39.1
Joint T-S	33.7	56.8	36.1	36.6	44.6
Factorised T-S (FTSA)	**37.0**	59.6	40.0	39.3	46.3

(a) **Self-attention schemes.** We perform different multi-head self-attention schemes on tracklet queries.

Length	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
$T=9$	35.3	57.6	38.5	37.7	43.5
$T=18$	36.4	58.1	39.5	39.1	46.8
$T=36$	**37.0**	59.6	40.0	39.3	46.3

(c) **Video clip length T.** We experiment with different number of frames for each video clip. Larger temporal receptive filed provides richer temporal context and therefore yields better performance.

Scheme	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
w/o CL	7.4	19.0	5.9	9.8	13.5
w/ CL	**35.3**	57.6	38.5	37.7	43.5

(e) **Correspondence learning (CL).** We train EfficientVIS with and without CL. After training, we test both using our fully end-to-end inference paradigm. $T=9$ in this study.

Method	Train Aug.	Epopchs	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
VisTR [25]	random crop	∼500	35.6	56.8	37.0	35.2	40.2
EfficientVIS	✗	33	**37.0**	59.6	40.0	39.3	46.3
EfficientVIS	multi-scale	33	**37.9**	59.7	43.0	40.3	46.6

(g) **Convergence speed. (EfficientVIS *vs.* VIS Transformer).** $T=36$ for both VisTR and EfficientVIS. VisTR is equipped with random cropping training augmentation by default.

Dynamic Conv	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
Still-image	36.0	59.5	39.5	38.4	45.3
Temporal	**37.0**	59.6	40.0	39.3	46.3

(b) **Still-image *vs.* Temporal - dynamic convolution**. Temporal dynamic convolution is more effective by taking into account temporal object context from nearby frames.

	Scheme	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
$T=9$	Hand-craft	33.7(−1.6)	55.5	36.4	33.9	40.3
	Fully e2e (ours)	**35.3**	57.6	38.5	37.7	43.5
$T=18$	VisTR [25]	29.7(−6.7)	50.4	31.1	29.5	34.4
	Hand-craft	34.6(−1.8)	55.3	37.4	36.6	44.6
	Fully e2e (ours)	**36.4**	58.1	39.5	39.1	46.8

(d) **Fully end-to-end (e2e) *vs.* Partially e2e**. Both "Hand-craft" and VisTR are partially e2e, where tracklet association within each clip is e2e but that between clips requires a hand-crafted linking. For "Hand-craft", we report the best results by varying matching score thresholds.

Query	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
Time shared	35.5	57.1	38.5	38.7	43.9
Time disentangled	**37.0**	59.6	40.0	39.3	46.3

(f) **Time disentangled *vs.* Time shared - query.** For each tracklet query, the time disentangled scheme uses T embeddings, while the time shared scheme only uses one embedding.

Video Frame Rate	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
Original FPS	**35.3**	57.6	38.5	37.7	43.5
1.5 FPS	**35.3**	57.4	39.1	37.5	42.8

(h) **Tracking in low frame rate videos ($T=9$).** We downsample the frame rate of the original YouTube-VIS videos to 1.5 FPS. EfficientVIS is not affected by low video frame rate or dramatic object motions.

VISOLO: Grid-Based Space-Time Aggregation for Efficient Online Video Instance Segmentation

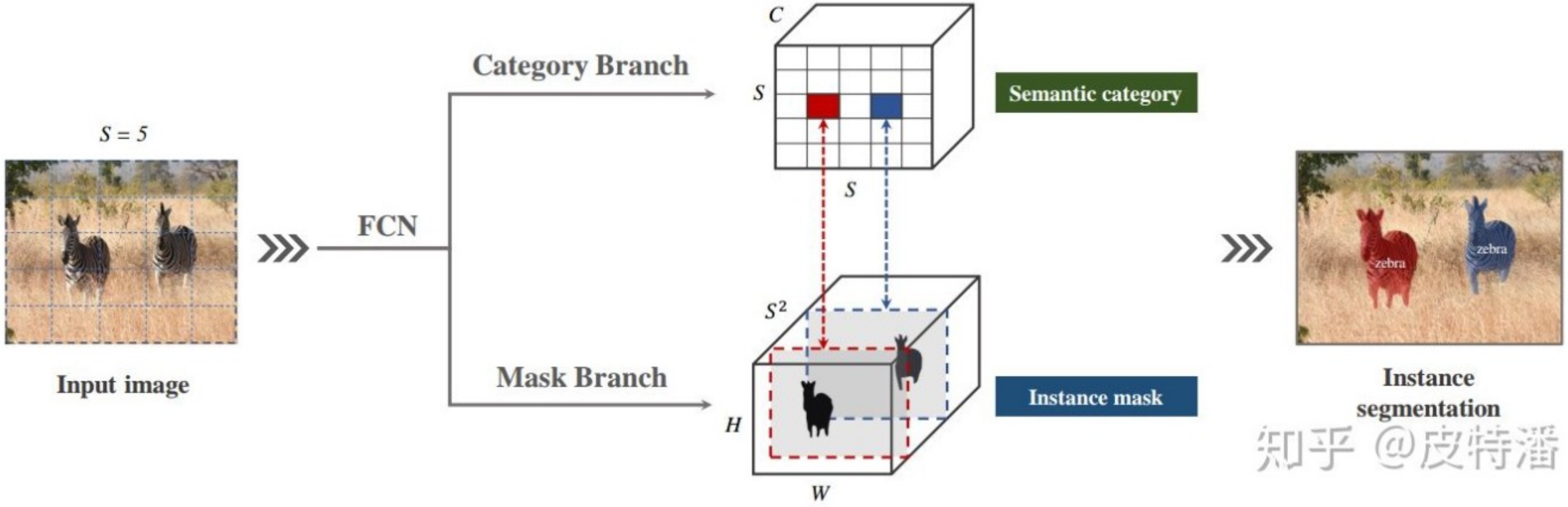
Su Ho Han[1], Sukjun Hwang[1], Seoung Wug Oh[2], Yeonchool Park[3],
Hyunwoo Kim[4], Min-Jung Kim[5] and Seon Joo Kim[1]

[1]Yonsei University [2]Adobe Research [3]LG Electronics [4]LG AI Research [5]KAIST

{hansuho123, sj.hwang, seonjookim}@yonsei.ac.kr seoh@adobe.com
yeonchool.park@lge.com hwkim@lgresearch.ai emjay73@kaist.ac.kr

VISOLO

- SOLO



S = 5
Input image
FCN
Category Branch
Mask Branch
C
S
S
S^2
H
W
Semantic category
Instance mask
zebra
zebra
Instance segmentation
知乎 @皮特潘

VISOLO

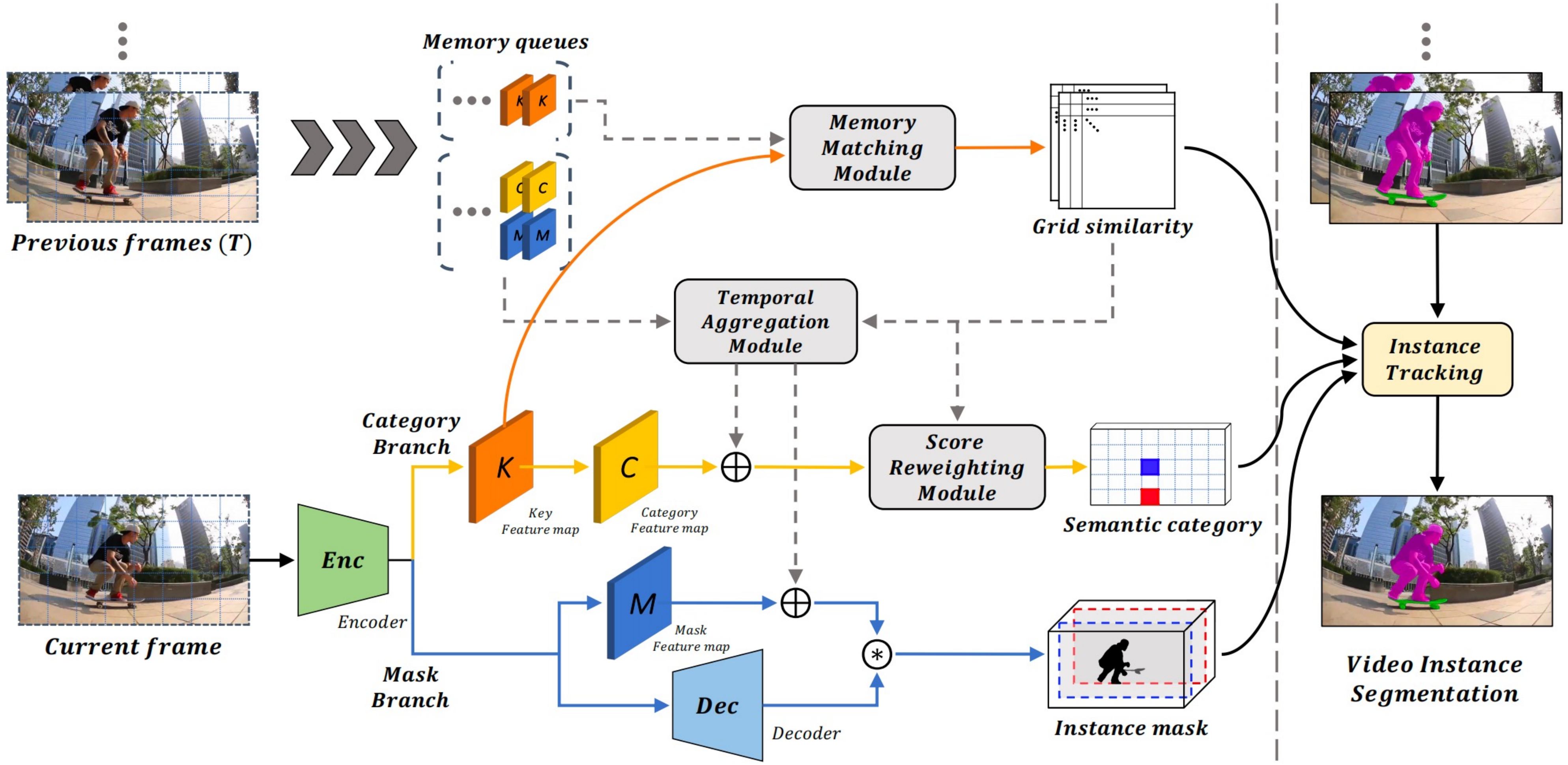


Figure 2. Overview of our framework VISOLO. We take ResNet50 [11] as the backbone network for the encoder. Our network consists of two branches: category branch, mask branch with three additional modules. The key (K), category (C) and mask(M) feature maps from the category and the mask branch are stored in the memory queues for future use. Dot arrows denote the use of the information from previous frames. '$\oplus$' denotes element-wise summation and '$\circledast$' denotes convolution.

VISOLO

- Memory Matching Module

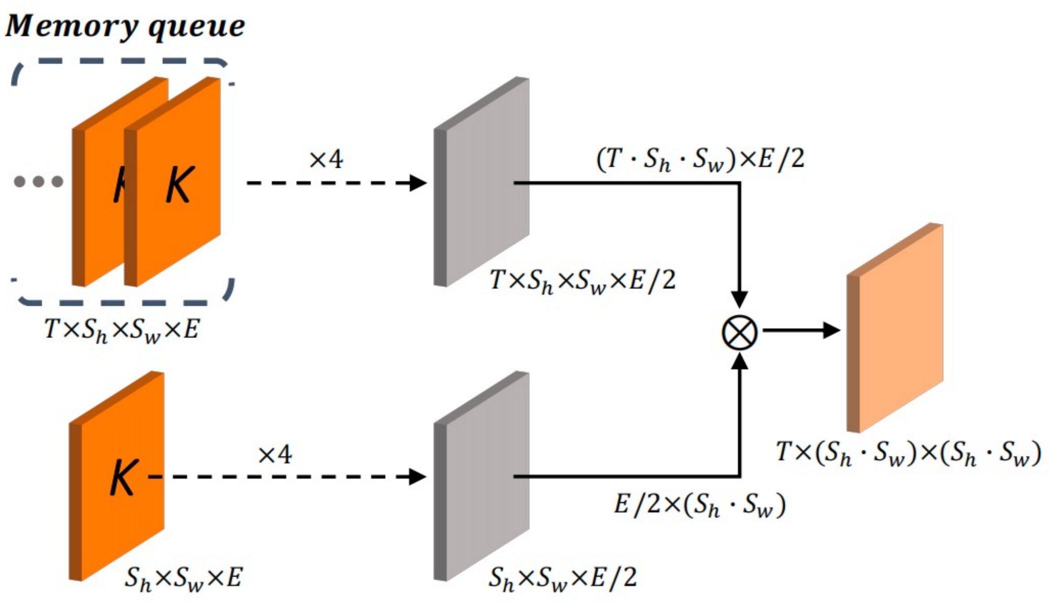
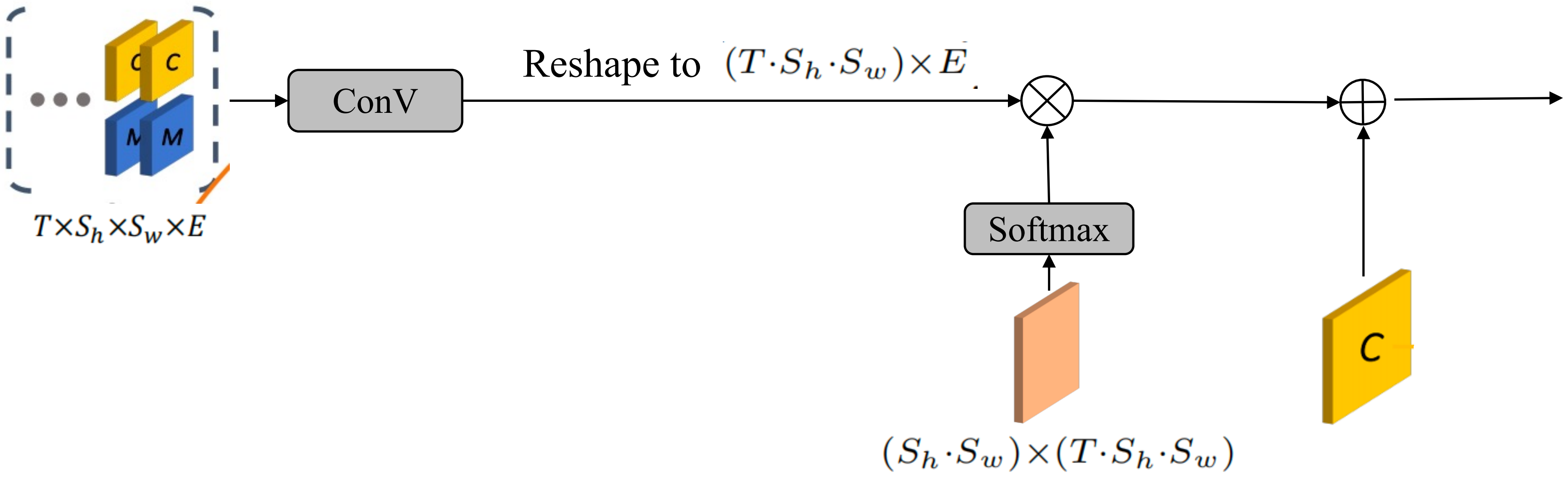


Figure 3. Detailed implementation of the memory matching module operation. It takes the key feature maps from the memory queue and the category branch as inputs. S_h and S_w are the number of grids in height and width, respectively, and E is the input feature map dimension. Dot arrows denote convolutional layers and '$\otimes$' denotes matrix inner-product.

VISOLO

- Temporal Aggregation Module
  - gathers the appearance information from the past using the grid similarity



VISOLO

- Score Reweighting Module
  - using grid similarity to update current category score

$$\mathbf{Cat} \in \mathbb{R}^{S_h \times S_w \times C}$$

$$\mathbf{Sim} \in \mathbb{R}^{(S_h \cdot S_w) \times (S_h \cdot S_w)}$$

$$\mathbf{P} = \mathbf{Cat} \odot \mathrm{AVG}(\tilde{\mathbf{Sim}}_1, \tilde{\mathbf{Sim}}_2)$$

VISOLO

- Instance tracking
  - using grid similarity to track

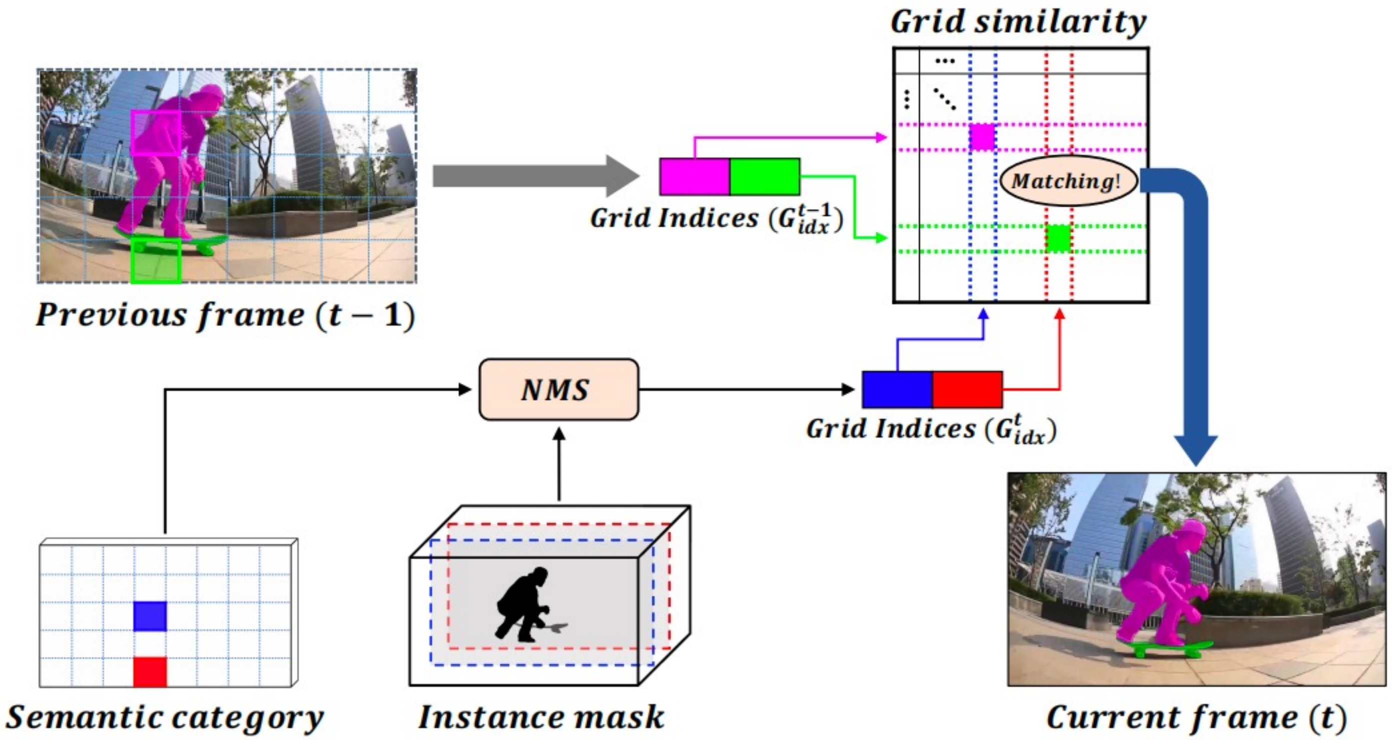


Figure 4. Overview of instance tracking operation between the current frame (t) and the previous frame ($t-1$). G_{idx} indicates the index of grids that contain the center of the instances.

VISOLO

	Methods	Backbone	FPS	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
Offline	MaskProp [2]	ResNet-50	–	40.0	–	42.9	–	–
	SeqMask-RCNN [16]	ResNet-50	3.8	40.4	63.0	43.8	41.1	49.7
	VisTR [31]	ResNet-50	51.1	35.6	56.8	37.0	35.2	40.2
	IFC [13]	ResNet-50	107.1	41.2	65.1	44.6	42.3	49.6
Near Online	STEm-Seg [1]	ResNet-101	3.0	34.6	55.8	37.9	34.4	41.6
Online	MaskTrack-RCNN [33]	ResNet-50	26.1	30.3	51.1	32.6	31.0	35.5
	SipMask [4]	ResNet-50	35.5	33.7	54.1	35.8	35.4	40.1
	SG-Net [20]	ResNet-50	23.0*	34.8	56.1	36.8	35.8	40.8
	SG-Net [20]	ResNet-101	19.8*	36.3	57.1	39.6	35.9	43.0
	CompFeat [9]	ResNet-50	–	35.3	56.0	38.6	33.1	40.3
	CrossVIS [34]	ResNet-50	25.6	36.3	56.8	38.9	35.6	40.7
	CrossVIS [34]	ResNet-101	23.3	36.6	57.3	39.7	36.0	42.0
	STMask [14]	ResNet-50†	26.1	33.5	52.1	36.9	31.1	39.2
	STMask [14]	ResNet-101‡	22.4	36.8	56.8	38.0	34.8	41.8
	Our VISOLO	ResNet-50	40.0	38.6	56.3	43.7	35.7	42.5

Table 1. Quantitative evaluation on **YouTube-VIS 2019** [33] validation set. [20] does not provide official checkpoints, so we infer the speed reported in [20] (FPS with superscript ”∗”). ”†” and ”‡” indicate the ResNet-50-DCN and ResNet-101-DCN, respectively.

VISOLO

Methods	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
MaskTrack-RCNN	28.6	48.9	29.6	26.5	33.8
SipMask	31.7	52.5	34.0	30.8	37.8
CrossVIS	34.2	54.4	37.9	30.4	38.2
STMask	30.6	49.4	32.0	26.4	36.0
Our VISOLO	36.9	54.7	40.2	30.6	40.9

SR	TA (Category)	TA (Mask)	AP	AP_{50}	AP_{75}
			34.6	51.5	36.8
✓			35.6	53.8	37.9
	✓		36.4	54.4	39.3
✓	✓		37.7	56.6	40.3
✓	✓	✓	38.6	56.3	43.7

Table 3. Ablation study of the **S**core **R**eweighting module (SR) and the **T**emporal **A**ggregation module (TA), estimated on YouTube-VIS 2019 dataset.

Memory frames	FPS	AP	AP_{50}	AP_{75}
2 frames	40.4	36.7	54.2	40.4
10 frames	39.5	37.5	55.3	41.3
20 frames	38.7	37.7	55.4	41.4
Every 5 frames	40.0	38.6	56.3	43.7

Table 4. The number of reference frames for temporal aggregation module analysis on the validation sets of YouTube-VIS 2019 dataset [33]. We compare results by different memory storing rules.

VISOLO

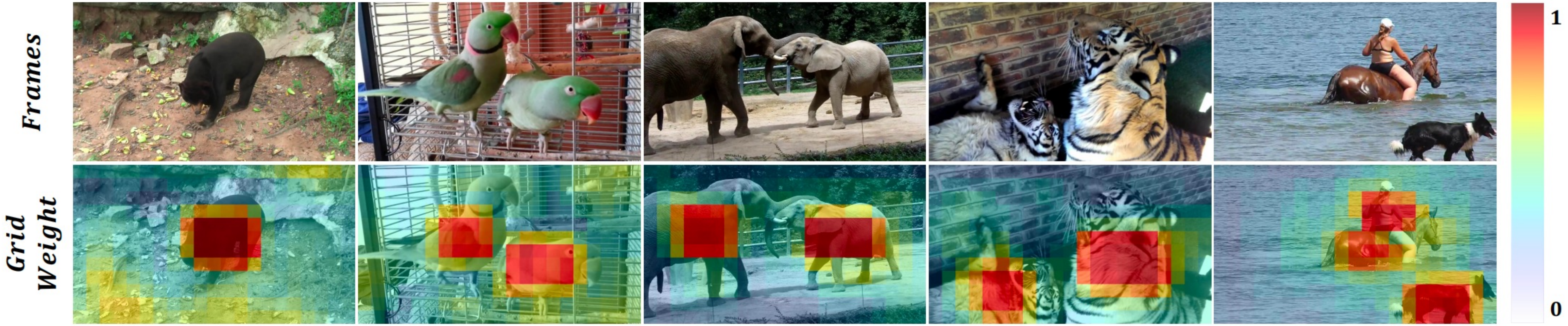


Figure 6. Visualization of weights for each grid in the score reweighting module at the second row. The first row shows the original frames.

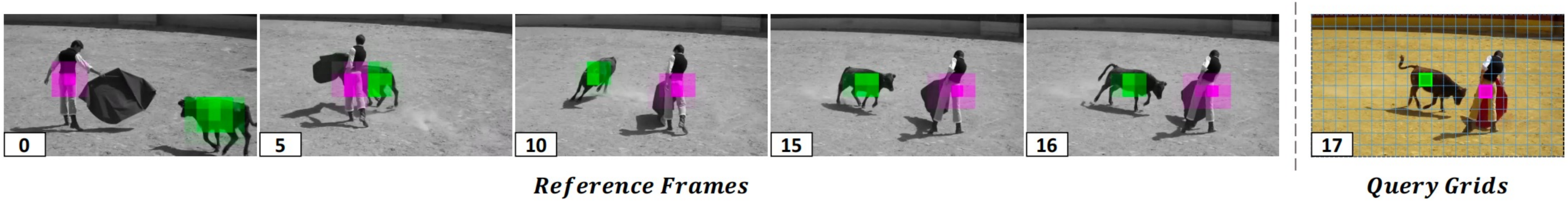


Figure 7. Visualization of our temporal aggregation module operation. We first compute the grid similarities between query grids and all grids of reference frames, and obtain the soft weight by a softmax operation. Then, we visualize the normalized soft weights of the reference frames. The query grids and weights of each grid of reference frames with respect to the query grids are assigned with different colors.

Temporally Efficient Vision Transformer for Video Instance Segmentation

Shusheng Yang[1,3*], Xinggang Wang[1†], Yu Li[4*], Yuxin Fang[1],
Jiemin Fang[2,1], Wenyu Liu[1], Xun Zhao[3], Ying Shan[3]
[1]School of EIC, Huazhong University of Science & Technology
[2]Institute of Artificial Intelligence, Huazhong University of Science & Technology
[3]Applied Research Center (ARC), Tencent PCG [4]International Digital Economy Academy (IDEA)

TeViT

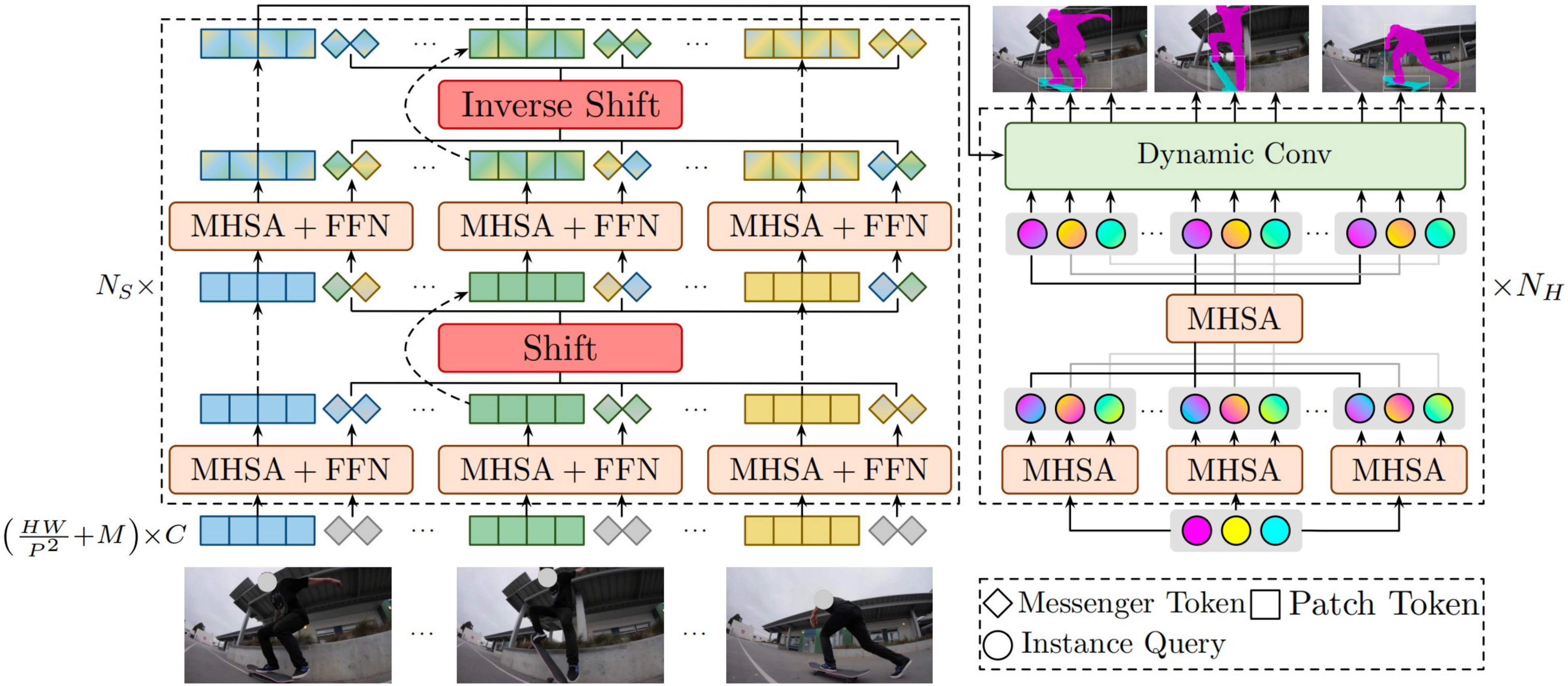
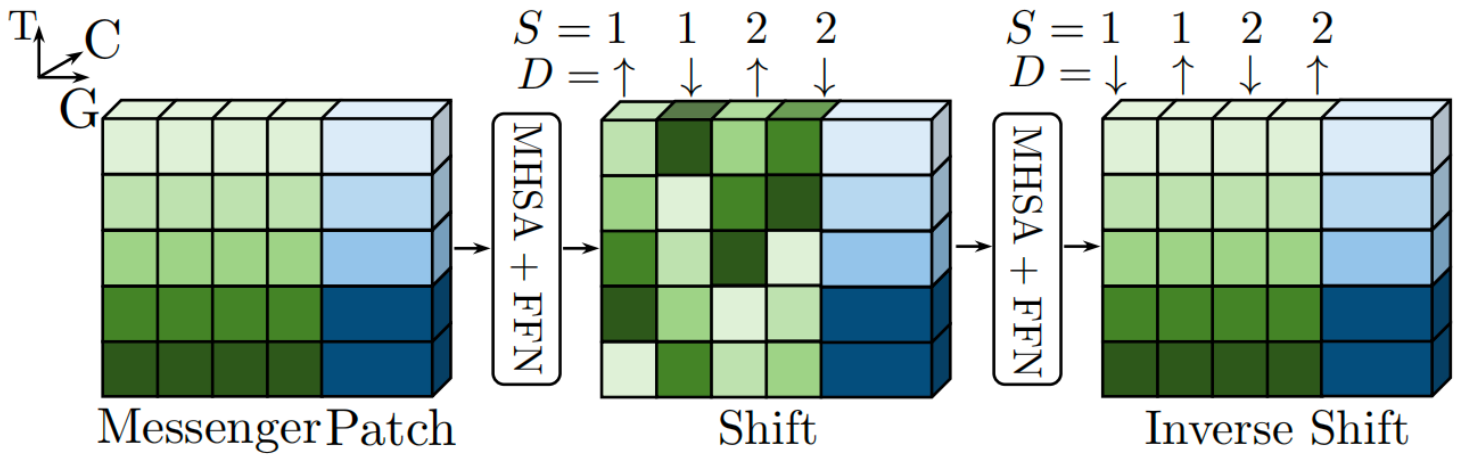


Figure 1. The overall illustration of our TeViT framework. TeViT contains a messenger shift transformer backbone and a series of spatiotemporal query-driven instance heads. The messenger shift mechanism performs efficient frame-level temporal modeling by simply shifting messenger tokens along the temporal axis. Spatiotemporal query interaction conducts two successive and parameter-shared multi-head self attention (MHSA) with feed forward network (FFN) upon video instance queries. The "Dynamic Conv" design follows QueryInst [18]. Best viewed in color.

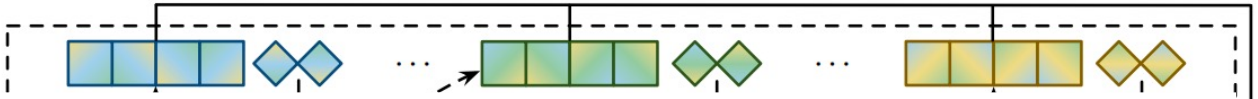
TeViT

- Messenger Shift Transformer Backbone:



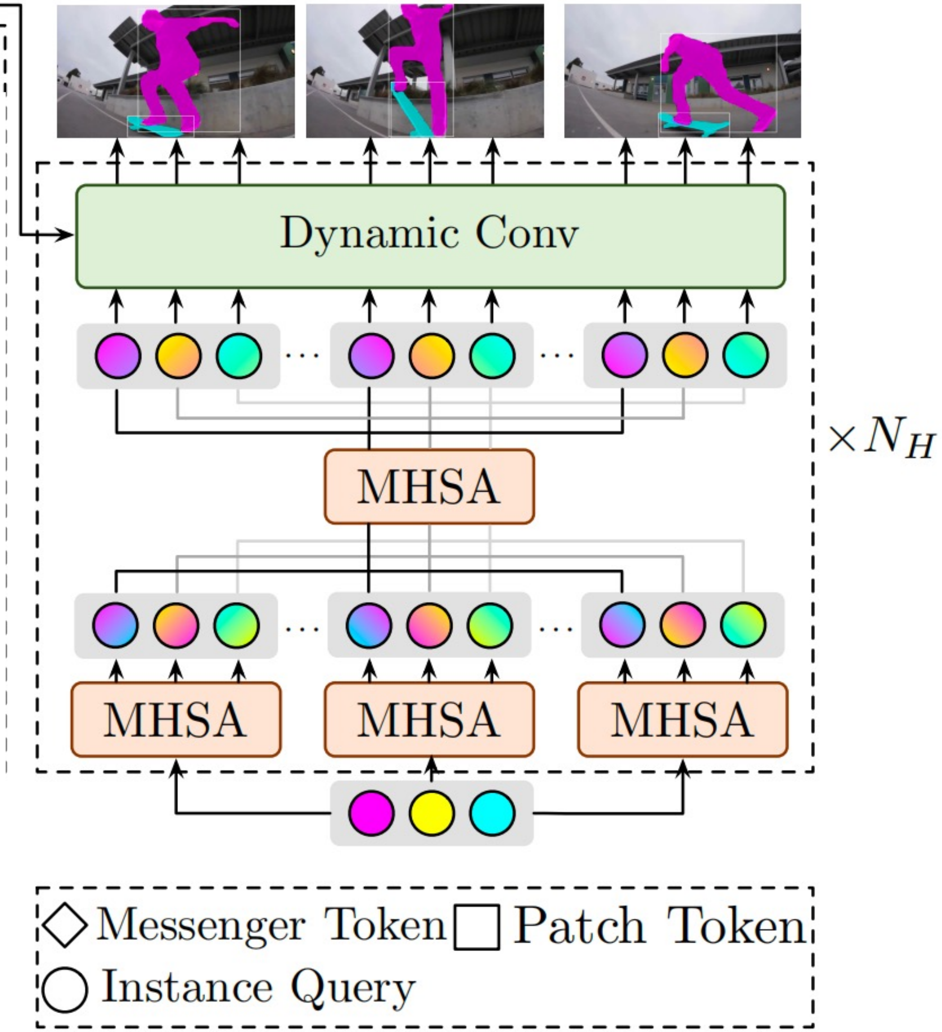
TeViT

- Spatiotemporal Query Interaction Head



$$\hat{Q}^{1:T}_{1:N_q} = \left\{ \mathrm{MHSA}\left(Q^{i}_{1:N_q} \right) \right\}_{i=1}^{T}$$

$$\widetilde{Q}^{1:T}_{1:N_q} = \left\{ \mathrm{MHSA}\left(\hat{Q}^{1:T}_{j} \right) \right\}_{j=1}^{N_q}$$



TeViT

- YTVIS 2019:

VisTR [57]	ResNet-50		51.1	36.2	59.8	36.9	37.2	42.4
VisTR [57]	ResNet-101		43.5	40.1	64.0	45.0	38.3	44.9
EfficientVIS [58]	ResNet-50	✓	36.0	37.9	59.7	43.0	40.3	46.6
EfficientVIS [58]	ResNet-101	✓	32.0	39.8	61.8	44.7	42.1	49.8
IFC [23]	ResNet-50	✓	107.1	41.2	65.1	44.6	42.3	49.6
IFC [23]	ResNet-101	✓	89.4	42.6	66.6	46.3	43.5	51.4
TeViT (ours)	MsgShifT		68.9	45.9	69.1	50.4	44.0	53.4
TeViT (ours)	MsgShifT	✓	68.9	**46.6**	**71.3**	**51.6**	**44.9**	**54.3**

- YTVIS 2021

Methods	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
MaskTrack R-CNN† [62, 63]	28.6	48.9	29.6	26.5	33.8
SipMask† [6, 63]	31.7	52.5	34.0	30.8	37.8
CrossVIS [63]	34.2	54.4	37.9	30.4	38.2
IFC [23]	35.2	57.2	37.5	–	–
TeViT	37.9	61.2	42.1	35.1	44.6

TeViT

MSM	STQI	GFLOPs	AP $\pm\sigma_{AP}$	AP_{50}	AP_{75}	AR_1	AR_{10}
		81.97	42.5 ± 0.47	67.6	44.0	43.0	52.7
✓		82.19	$43.1_{\uparrow(+0.6)} \pm 0.71$	67.2	47.8	43.5	52.4
	✓	81.97	$45.2_{\uparrow(+2.7)} \pm 0.85$	68.9	50.2	**44.0**	53.0
✓	✓	82.19	$\mathbf{45.9}_{\uparrow(+3.4)} \pm 0.58$	**69.1**	**50.4**	**44.0**	**53.4**

Table 4. Component-wise analysis on TeViT. MSM denotes the messenger shift mechanism and STQI denotes spatiotemporal query interaction. Without applying STQI implies only one MHSA is performed for query interaction within each frame (excluding Eq. 4).

Interaction	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
Spatial Only [18]	43.1	67.2	47.8	43.5	52.4
Fused Space-Time [57]	$43.9_{\uparrow(+0.8)}$	**69.5**	48.4	42.9	52.0
Ours	$\mathbf{45.9}_{\uparrow(+2.7)}$	69.1	**50.4**	**44.0**	**53.4**

Table 5. Variants of spatiotemporal query interaction. "Spatial Only" denotes the image-level instance segmentation heads in [18], "Fused Space-Time" denotes applying MHSA to all video instance queries at a single run, which is the same as in [57].

Manip.	AP $\pm$ σ_{AP}	AP_{50}	AP_{75}
None	45.2 ± 0.85	68.9	50.2
MHSA + FFN	44.5 ± 1.07	69.2	49.3
Shift	$\mathbf{45.9 \pm 0.58}$	**69.1**	**50.4**

Table 6. Study of the manipulations upon messenger tokens. Our method obtains the highest AP and a relatively stable performance (σ_{AP}) among all settings.

Manip.	AP	AP_{50}	AP_{75}
None	45.2	68.9	50.2
Conv	41.8	63.7	45.1
MHSA + FFN	43.1	67.2	49.1
Msg Shift	**45.9**	**69.1**	**50.4**

Table 7. Study of frame-level feature aggregation. Compared to other frame-level feature manipulations, our messenger shift (Row 4) obtains the best results.

M	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
8	45.3	69.0	48.9	**44.5**	52.4
16	45.4	**70.3**	49.9	44.0	51.7
32	**45.9**	69.1	**50.4**	44.0	**53.1**

Table 8. Impact of messenger token numbers. M indicates the number of messenger tokens. We increase M from 8 to 32 and observe the effects on final performance.

GMP VIS

A Graph Matching Perspective with Transformers on Video Instance Segmentation

Zheyun Qin[1*], Xiankai Lu[1*], Xiushan Nie[2], Yilong Yin[1†], Jianbing Shen[3]

[1] School of Software, Shandong University [2] School of Cpmputer Science and Technology, Shandong Jianzhu University

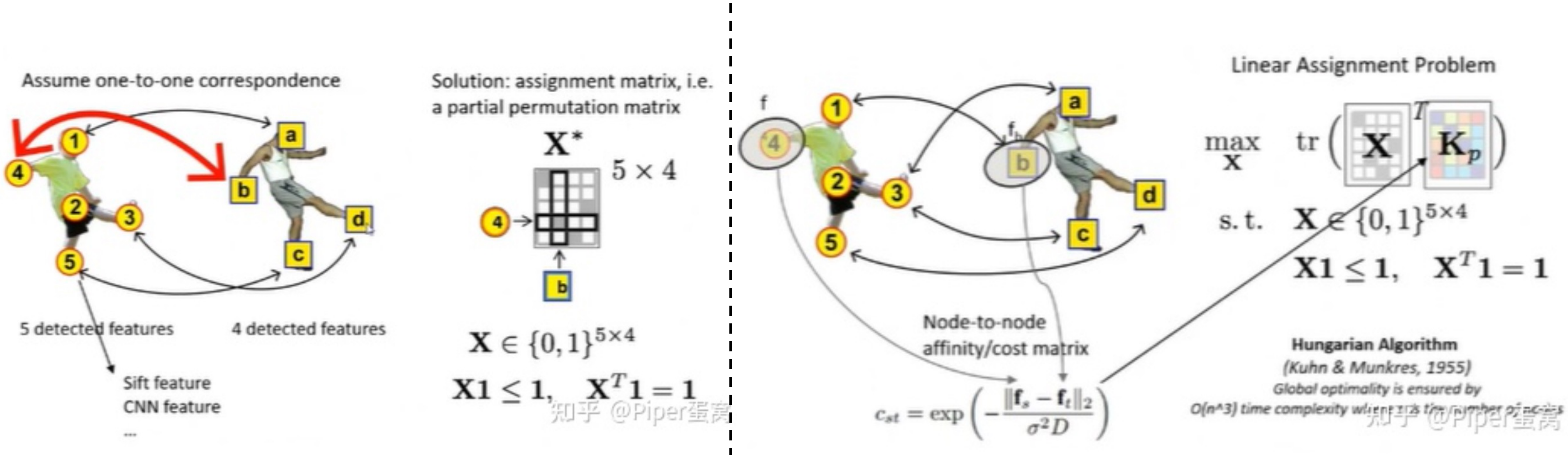
[3] SKL-IOTSC, University of Macau

zyqin@mail.sdu.edu.cn, carrierlxk@gmail.com, niexsh@hotmail.com

ylyin@sdu.edu.cn, shenjianbingcg@gmail.com

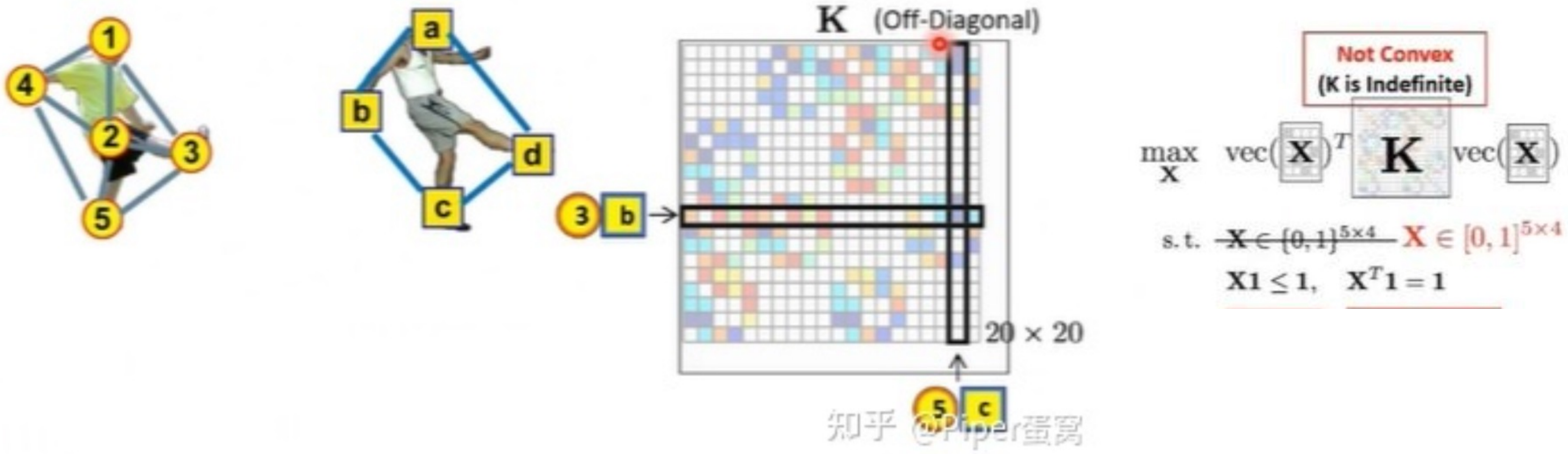
GMP VIS

- Bipartite graph matching （Node-wise linear assignment problem）：



GMP VIS

- Quadratic Assignment Problem（Edge-wise graph matching）：



GMP VIS

- Koopmans-Beckmann's type:

$$\underset{\mathbf{\Pi}}{\text{maximize}} \quad \mathcal{J}(\mathbf{\Pi}) = \mathrm{tr}(\mathbf{A}_1\mathbf{\Pi}\mathbf{A}_2\mathbf{\Pi}^\top) + \mathrm{tr}(\mathbf{B}^\top\mathbf{\Pi}),$$

$$\text{s.t.} \quad \mathbf{\Pi}\mathbf{1}_n = \mathbf{1}_n, \mathbf{\Pi}^\top\mathbf{1}_n = \mathbf{1}_n,$$

$$\mathbf{\Pi} \in \{0,1\}^{n\times n}$$

$\mathbf{A}_1 \in \mathbb{R}^{n\times n}$, $\mathbf{A}_2 \in \mathbb{R}^{n\times n}$:weighted adjacency matrices of graph G1 and G2

$\mathbf{B} \in \mathbb{R}^{n\times n}$:node-to-node affinity between G1 and G2

- After Reformulation and Convex Relaxation

$$\mathbf{\Pi}^* = \arg\min_{\mathbf{\Pi}} \frac{1}{2}||\mathbf{A_1}\mathbf{\Pi} - \mathbf{\Pi}\mathbf{A_2}||_F^2 - \mathrm{tr}(\mathbf{B}^\top\mathbf{\Pi}).$$

  - For two nodes i, i′ ∈ G1 and their corresponding nodes j, j′ ∈ G2
  - the difference of the weight of edge (i, i′) and (j, j′)
  - the node affinities between i and j.

GMP VIS

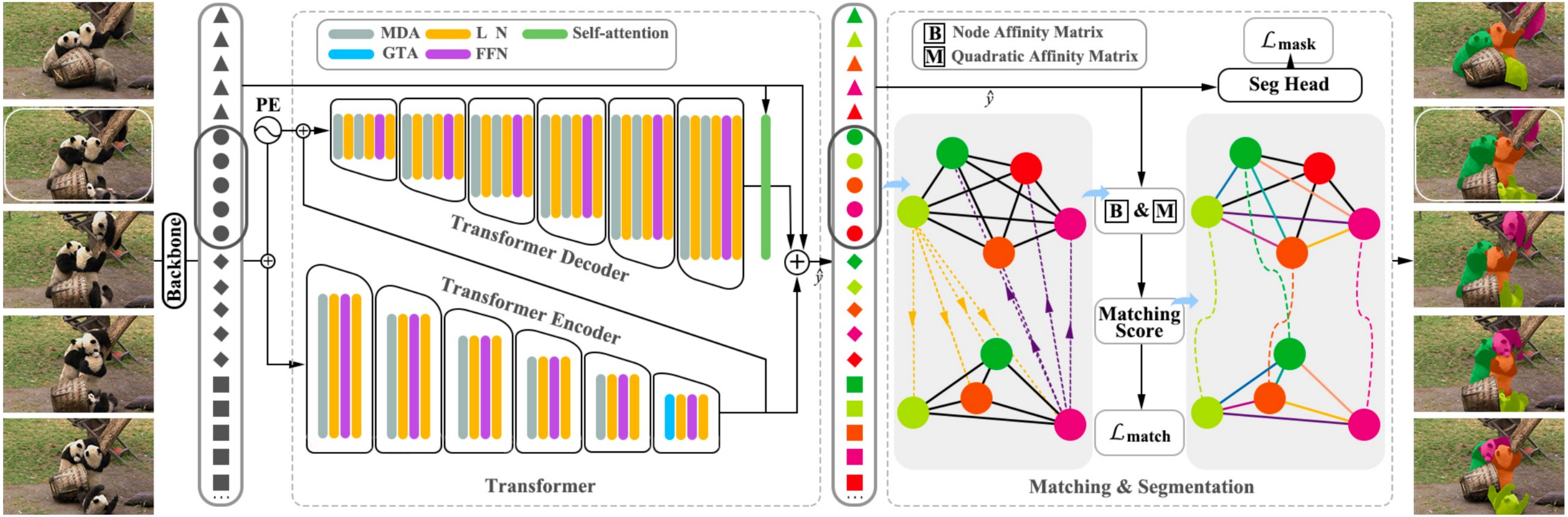


Figure 2. The overall framework of the proposed GMP-VIS. (1) a CNN backbone that extracts feature representation of multiple images. (2) an encoder-decoder Transformer with the multi-head deformable attention (MDA) that models the relations of pixel-level features and enhances the instance-level features with global temporal aggregation module (GTA), where PE is the position embedding. (3) an instance sequence matching and segmentation module supervises the model and outputs the final mask sequences.

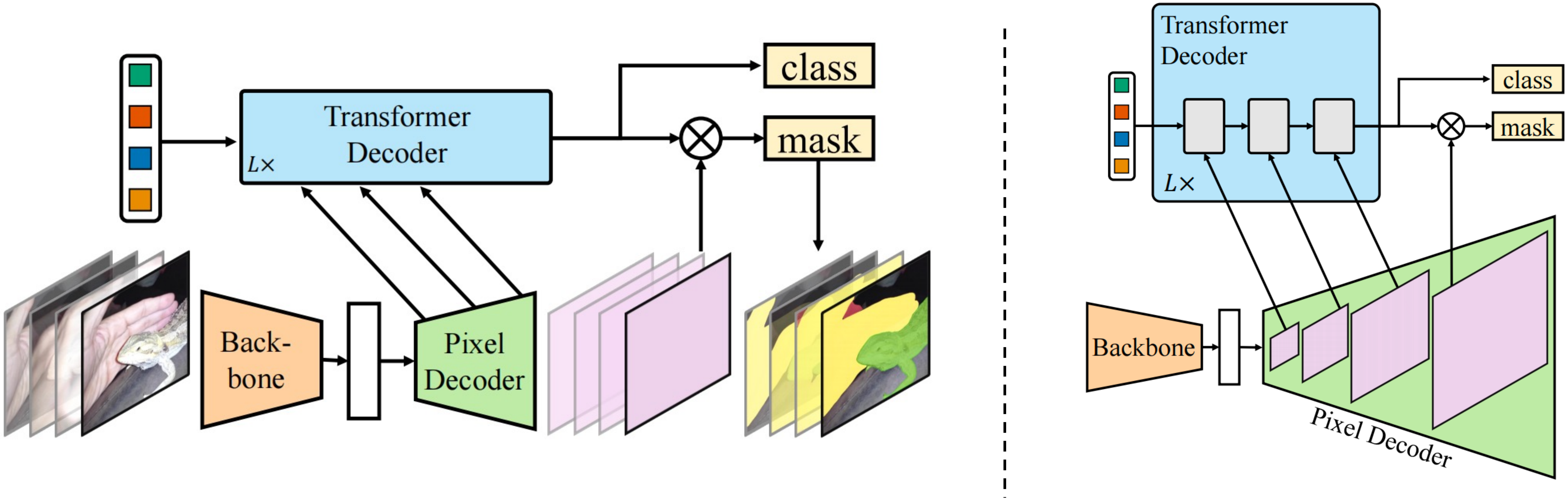
GMP VIS

- YTVIS 2019:

	Method	Backbone	FPS	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
Tracking-by-detection	OSMN MaskProp[CVPR18] [59]	ResNet-50	-	23.4	36.5	25.7	28.9	31.1
	IoUTracker+[ICCV19] [58]	ResNet-50	-	23.6	39.2	25.5	26.2	30.9
	DeepSORT[ICIP17] [55]	ResNet-50	-	26.1	42.9	26.1	27.8	31.3
	FEELVOS[CVPR19] [50]	ResNet-50	-	26.9	42.0	29.7	29.9	33.4
	OSMN[CVPR18] [59]	ResNet-50	-	27.5	45.1	29.1	28.6	33.1
	SeqTracker[ICCV19] [58]	ResNet-50	-	27.5	45.7	28.7	29.7	32.5
	MaskTrack R-CNN[ICCV19] [58]	ResNet-50	32.0	30.3	51.1	32.6	31.0	35.5
	MaskTrack R-CNN[ICCV19] [58]	ResNet-101	20.0	31.8	53.0	33.6	33.2	37.6
	VisSTG[ICCV21] [52]	ResNet-50	-	35.2	55.7	38.0	33.6	38.5
	CrossVIS[ICCV21] [60]	ResNet-50	39.8	36.3	56.8	38.9	35.6	40.7
	CrossVIS[ICCV21] [60]	ResNet-101	35.6	36.6	57.3	39.7	36.0	42.0
	MaskProp[CVPR20] [5]	ResNet-50	-	40.0	-	42.9	-	-
	MaskProp[CVPR20] [5]	ResNet-101	-	42.5	-	45.6	-	-
Bottom-up	STEm-Seg[ECCV20] [3]	ResNet-50	10.5	30.6	50.7	33.5	31.6	37.0
	STEm-Seg[ECCV20] [3]	ResNet-101	10.0	34.6	55.8	37.9	34.4	41.6
	HEVis[ACM MM21] [44]	ResNet-50	13.0	32.7	53.5	33.6	32.9	38.2
	HEVis[ACM MM21] [44]	ResNet-101	12.0	35.3	53.5	34.6	34.9	40.2
	VisTR[CVPR21] [54]	ResNet-50	69.9	35.6	56.8	37.0	35.2	40.2
	VisTR[CVPR21] [54]	ResNet-101	57.7	38.6	61.3	42.3	37.6	44.2
	GMP-Vis	ResNet-50	**73.7**	**37.4**	**57.4**	**37.2**	**39.5**	**44.2**
	GMP-Vis	ResNet-101	**60.1**	**40.4**	**55.6**	**40.5**	**42.8**	**46.6**

Mask2Former VIS

- One query is responsible for the prediction of an instance over the entire video sequence
- Matching during Training : Bipartite graph matching
- Optimization:
  - masked-attention
  - calculating mask loss on few randomly sampled points
  - Swap the order of self-attention and cross attention



Video K-Net VIS

- Generate query on each frame, and then perform data association
- Matching during Training : Optimizing the prediction of two frames that match each other