Video Object Segmentation & Video Instance Segmentation

1. Video Object Segmentation using Space-Time Memory Networks
2. Fast End-to-End Embedding Learning for Video Object Segmentation
3. Collaborative Video Object Segmentation by Multi-Scale Foreground-Background Integration
4. Video Instance Segmentation

0 Video Object Segmentation (视频对象分割)

- **Target**
  - 将前景对象与背景区域进行分离的二值标记
- **Unsupervised**
  - 不需要任何手动注释
  - 通常基于物体运动与周围环境不同进行分割
- **Semi-supervised**
  - 利用第一帧的Mask 跟踪和分割给定的对象

二者都将目标对象视为一般对象，而不关心语义类别

0 Dataset

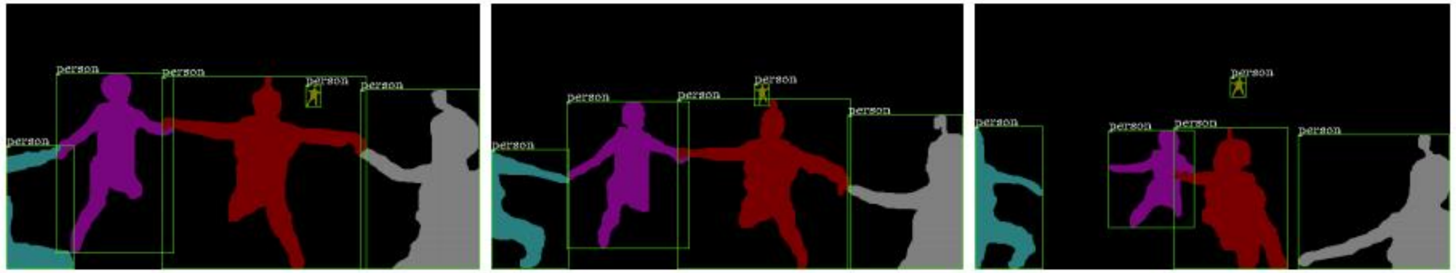
- **DAVIS(CVPR2016)**
  - DAVIS-2016数据集为**单对象**分割数据集，包含30个训练集，20个验证集
  - DAVIS-2017数据集为**多对象**分割数据集，一共有90个视频序列，包含60个训练视频，30个验证视频，验证集含59个对象组成

0 Dataset

- **Youtube-VOS** （for Video Object Segmentation ）
  - 由4453个高分辨率的YouTube视频和94个常用对象类别组成。
  - 验证集由474个视频组成，包括91个对象类别，其中65个为训练集中类别，其余26个为不可见类别对象。
  - 每个视频的长度约为3到6秒。
  - 30fps的帧速率每5帧手动跟踪对象边界

- **Youtube-VIS(ICCV2019)** （for Video Instance Segmentation ）
  - 由2883个高分辨率的YouTube视频组成，含40个类别的标签集

Video frames



Video instance annotations

0

Metrics –(A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation)

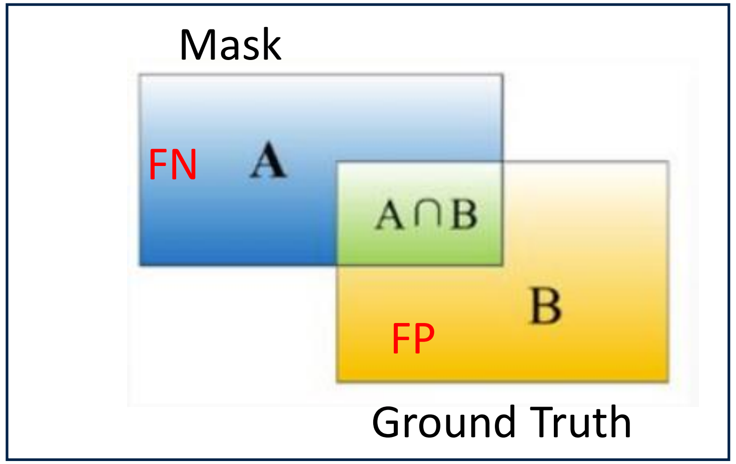
- **Region Similarity J （区域相似度）**

$$\mathcal{J} = \frac{|M \cap G|}{|M \cup G|}$$

- **Contour Accuracy F （轮廓精确度）**

将Mask 看成一系列闭合轮廓的集合，并计算基于轮廓的 F 度量

$$\mathcal{F} = \frac{2P_cR_c}{P_c+R_c} \quad \begin{cases} P_c = \dfrac{TP}{TP+FN} \\ R_c = \dfrac{TP}{TP+FP} \end{cases}$$

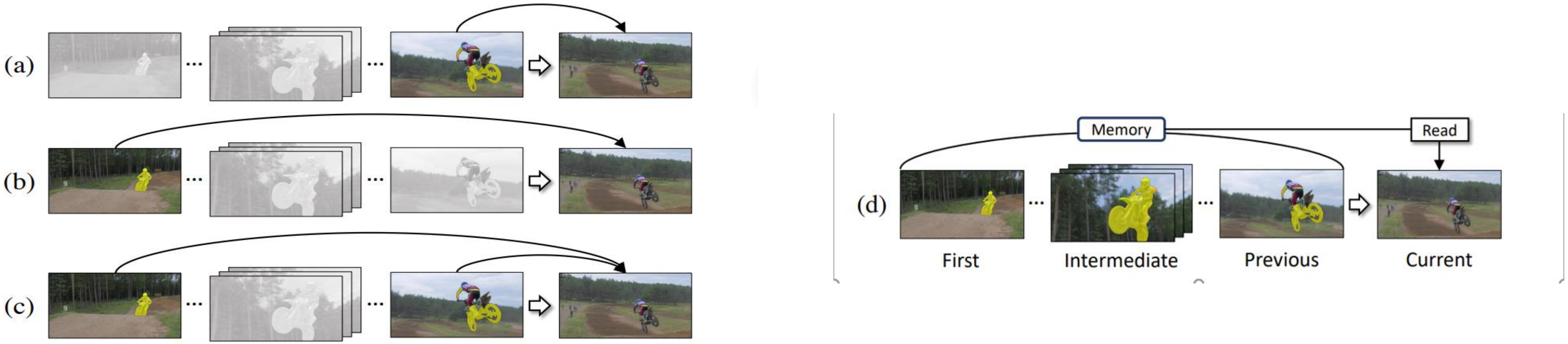


01 Video Object Segmentation using Space-Time Memory Networks

- **ICCV2019**

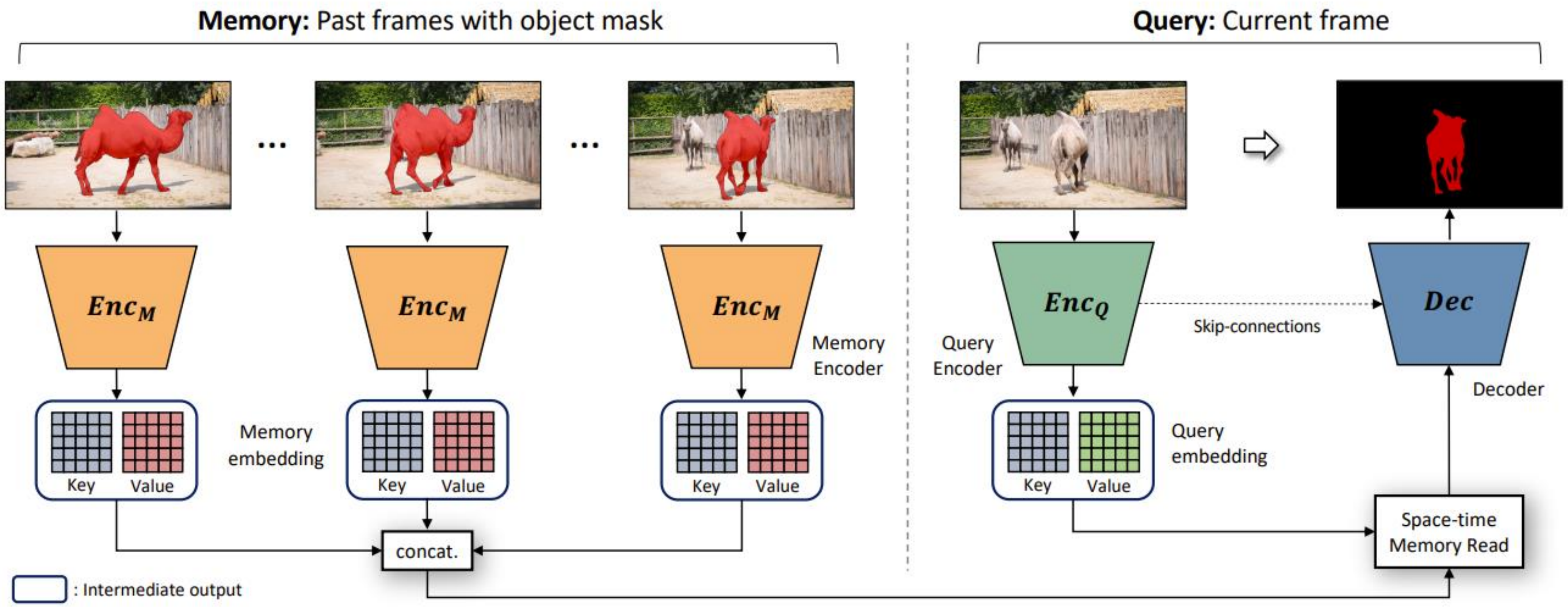
- **Motivation**
  - Available cues (e.g. video frame(s) with object masks) become richer.
  - However, the existing methods are unable to fully exploit this rich source of information.

- **Contribution**
  - Propose a novel solution for semi-supervised video object segmentation leveraging memory networks



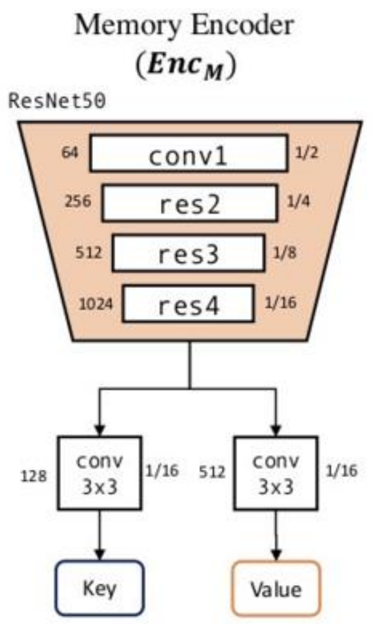
01 Video Object Segmentation using Space-Time Memory Networks

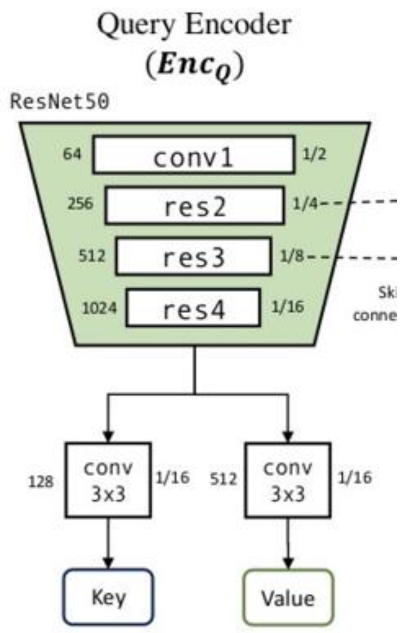
- **Overview of the framework**
  - 2 parts :**previous frame with mask** and **current frame**
  - 4 feature maps :**Key feature maps**(2) + **value feature maps**(2)



01 Video Object Segmentation using Space-Time Memory Networks

- **Encoder Networks**





$$\mathbf{k}^M \in \mathbb{R}^{T\times H\times W\times C/8}$$

$$\mathbf{v}^M \in \mathbb{R}^{T\times H\times W\times C/2}$$

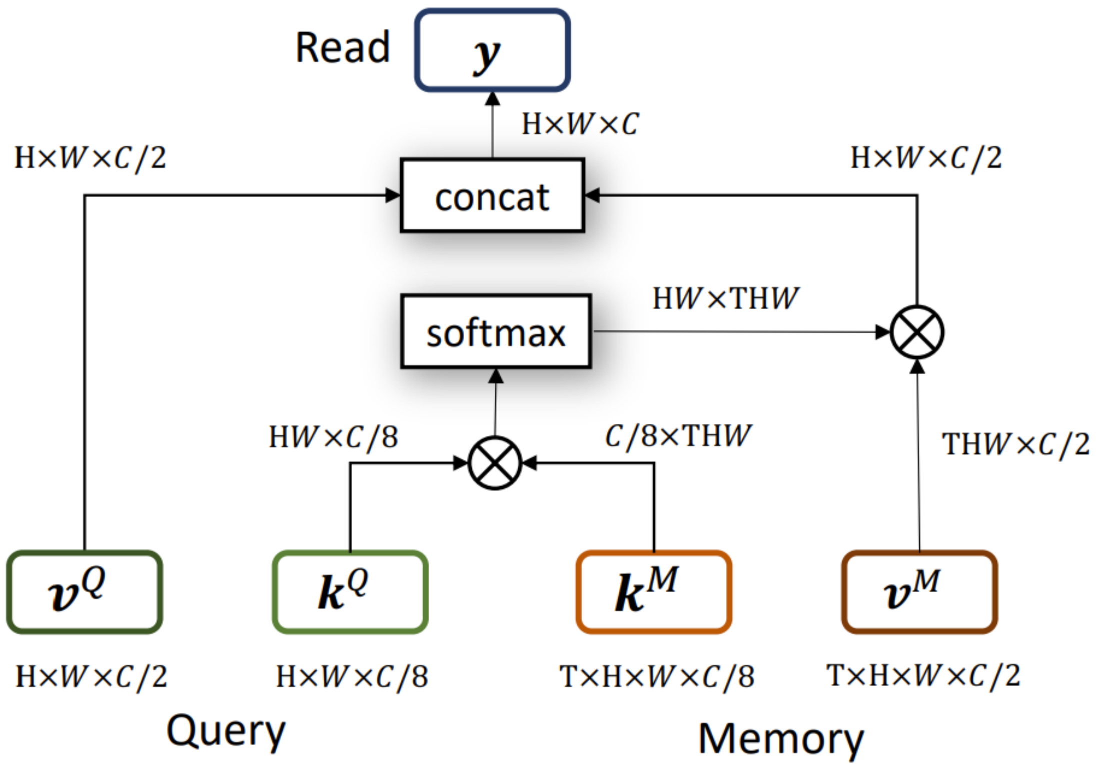
$$\mathbf{k}^Q \in \mathbb{R}^{H\times W\times C/8}$$

$$\mathbf{v}^Q \in \mathbb{R}^{H\times W\times C/2}$$

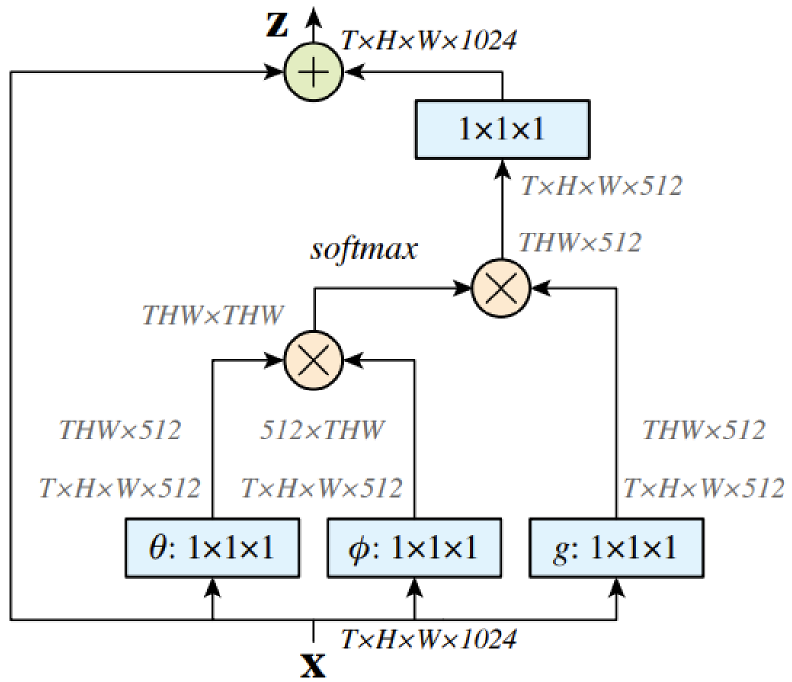
01 Video Object Segmentation using Space-Time Memory Networks

• STM Read

- **Soft weights** are first computed by measuring the similarities between all pixels of the query key map.
- Extract information from $\boldsymbol{v}^M$.
- Concat $\boldsymbol{v}^M$ and $\boldsymbol{v}^Q$.

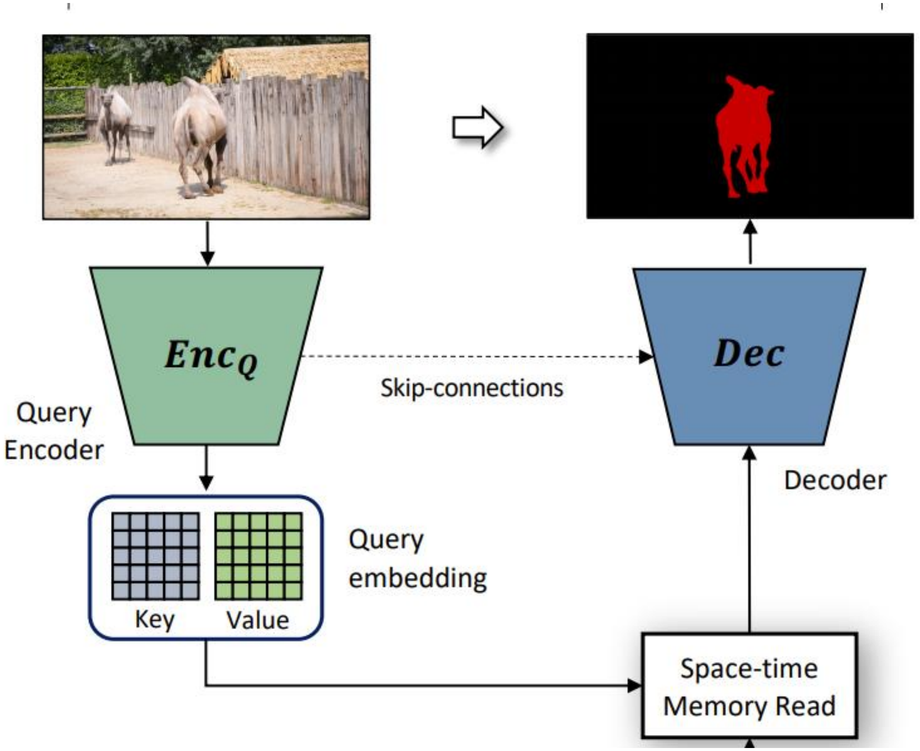


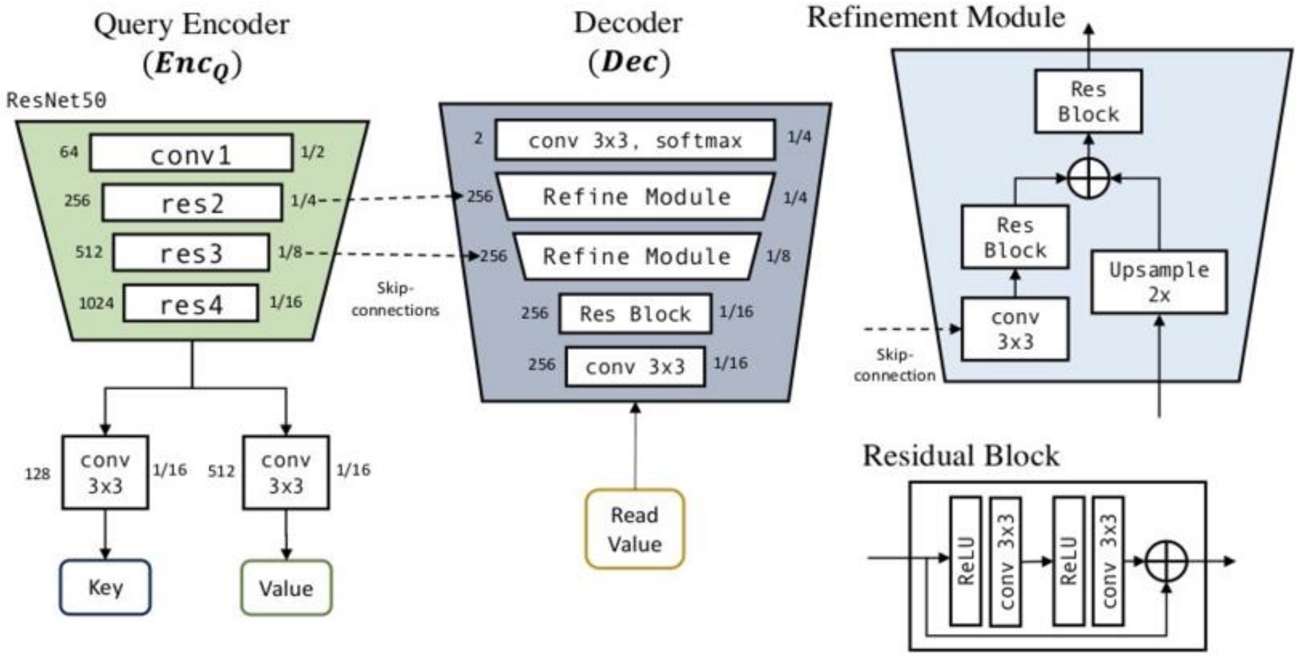
Compare with non-local :



01 Video Object Segmentation using Space-Time Memory Networks

- **Decoder**
  - The read output is first compressed to have 256 channels by a **convolutional layer** and **a residual block.**
  - A number **of refinement modules** upscale the feature map by a factor of two.
  - The decoder estimates the mask in 1/4 scale of the input image.





01 Video Object Segmentation using Space-Time Memory Networks

- **2 Stage Training**

  - **1)Pre-training on images:**

  A video clip that consists of 3 frames is generated by applying random affine transforms

  - **2)Main training on videos.**

  Sample 3 temporally ordered frames from Youtube-VOS or DAVIS-2017. The maximum number of frames to be skipped is gradually increased from 0 to 25.

- **Inference**

  - The **first and the previous frame** with object masks are the most important.
  - For the intermediate frames, we simply save a new memory frame **every 5 frames.**

01 Video Object Segmentation using Space-Time Memory Networks

- **The training process is more complicated, need large image data sets**
- **Multi-object Segmentation need a post-processing step**

Variants	Youtube-VOS	DAVIS-2017	
	Overall	$\mathcal{J}$	$\mathcal{F}$
Pre-training only	69.1	57.9	62.1
Main-training only	68.2	38.1	47.9
Full training	**79.4**	69.2	74.0

02 FEELVOS: Fast End-to-End Embedding Learning for Video Object Segmentation

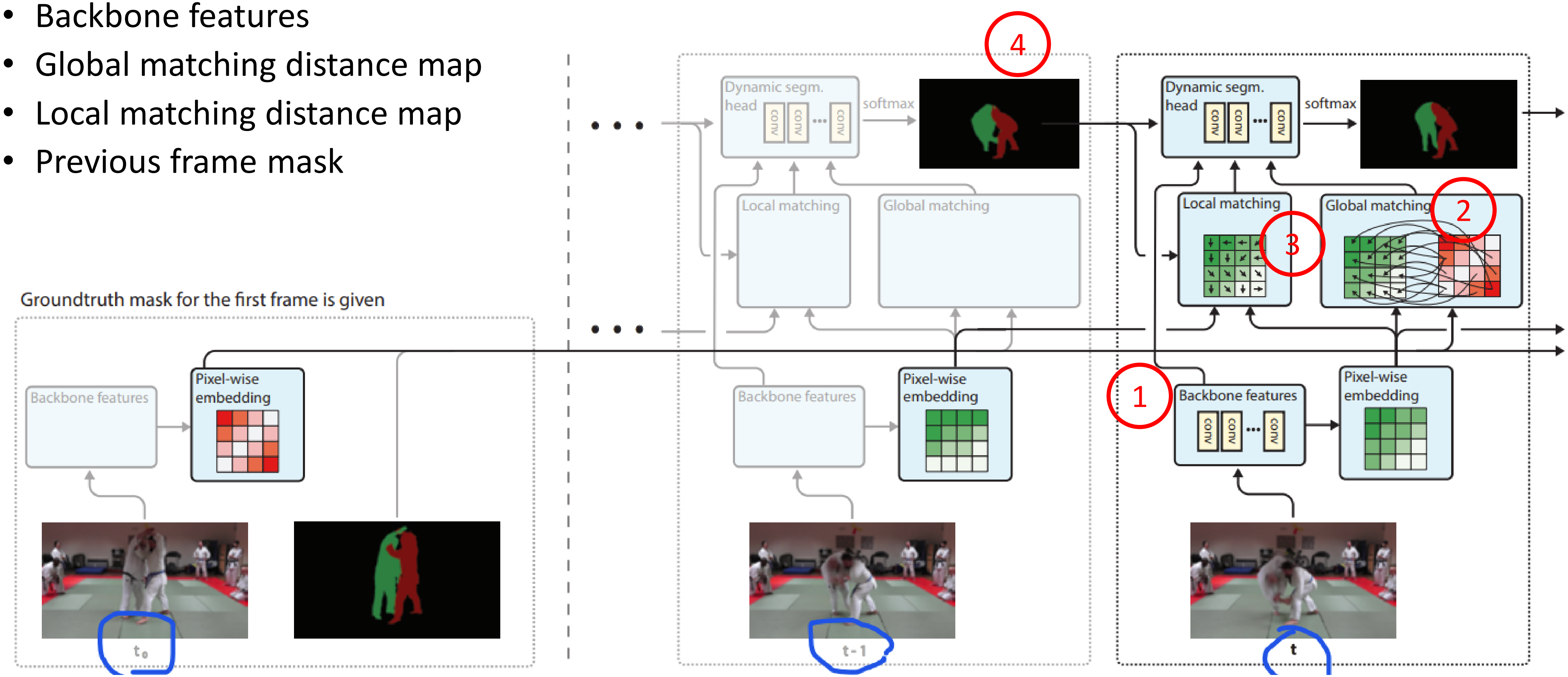
- **CVPR 2019**

- **Motivation**
  - Many of the recent successful methods for video object segmentation (VOS) are overly complicated

- **Contribution**
  - Design a **simple, fast, end-to-end, strong** Network without fine-turning
  - The network only need one-stage training and can handle Multi-object Segmentation without post-processing step

02 FEELVOS: Fast End-to-End Embedding Learning for Video Object Segmentatio

- **Overview of the framework**
  - Backbone features
  - Global matching distance map
  - Local matching distance map
  - Previous frame mask



02 FEELVOS: Fast End-to-End Embedding Learning for Video Object Segmentatio

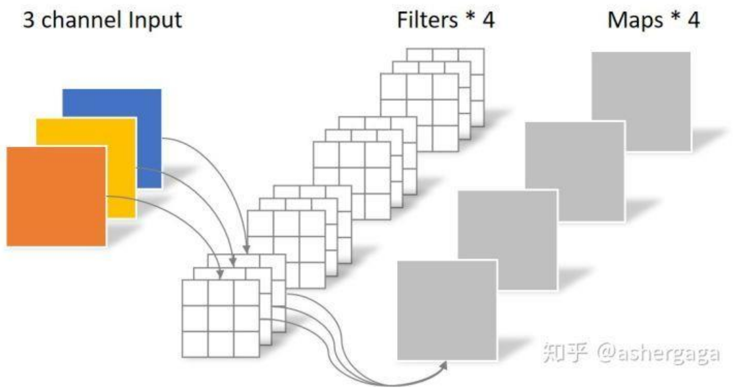
- **1. Backbone feature map**
  - DeepLab v3+

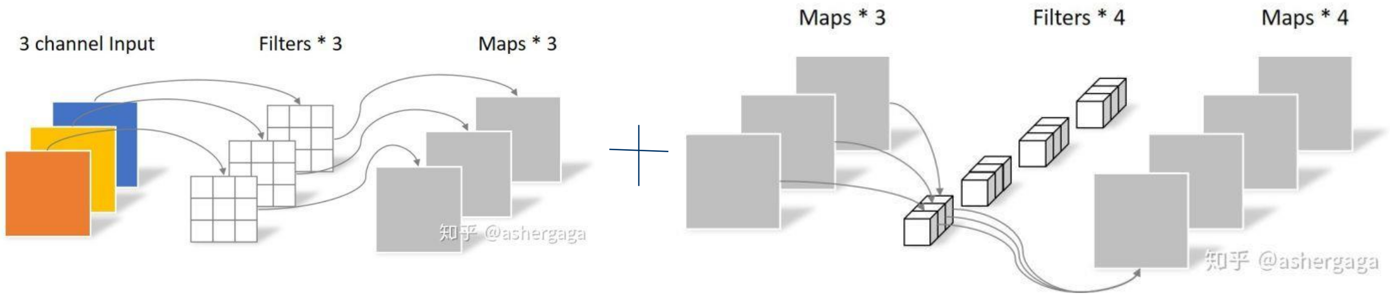
    depthwise separable convolutions (深度可分离卷积)

    batch normalization

    Atrous Spatial Pyramid Pooling

The **depthwise separable convolutions** divide the original convolution layer into two parts, which can achieve the same purpose and reduce the number of parameters





$$N_{depthwise} = 3 * 3 * 3 = 27 \qquad N_{depthwise} = 1 * 3 * 4 = 12$$

$$N_{conv} = 4 * 3 * 3 * 3 = 108$$

$$N_{\mathrm{sum}} = 27 + 12 = 39$$

02 FEELVOS: Fast End-to-End Embedding Learning for Video Object Segmentation

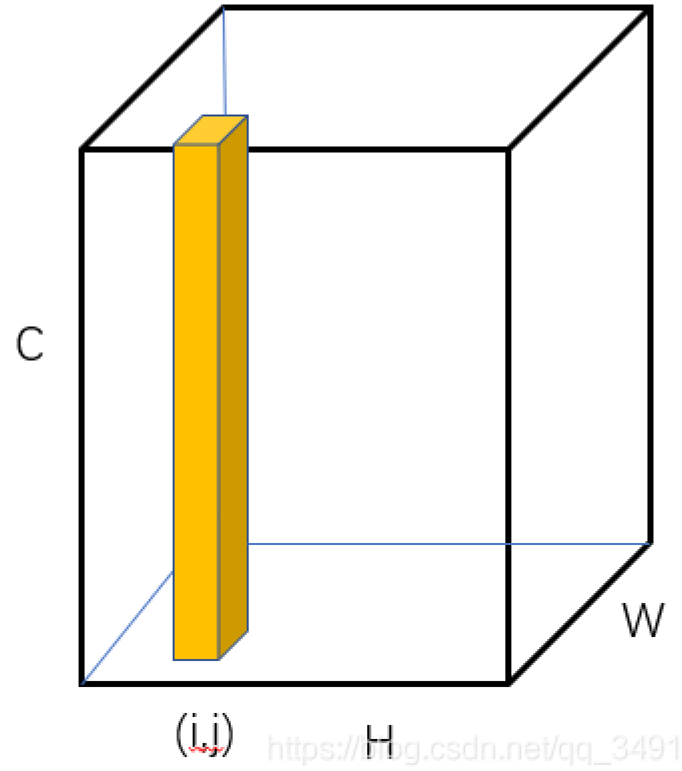
- **2. Global Matching and Local Matching**

Embedding feature vectors

backbone

- DeepLab v3+
  - depthwise separable convolutions (深度可分离卷积)
  - batch normalization
  - Atrous Spatial Pyramid Pooling
- 3x3 conv + 1x1 conv
- Channel = 100

$$d(p,q) = 1 - \frac{2}{1+\exp(\|e_p - e_q\|^2)}.$$



p和q来自不同的输入。根据q的不同，计算两种distance map：
Global distance map和 Local distance map

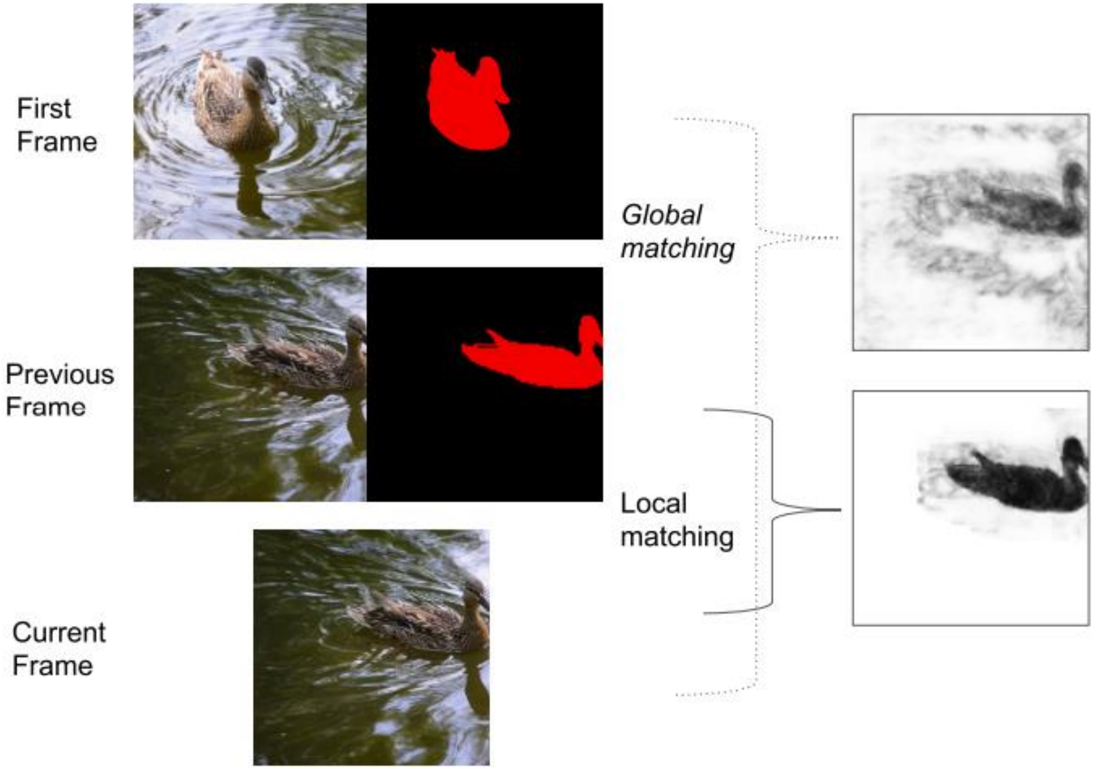
02 FEELVOS: Fast End-to-End Embedding Learning for Video Object Segmentation

- **Semantic Embedding.**
  - **Global Matching**

$$G_{t,o}(p) = \min_{q \in \mathcal{P}_{1,o}} d(p,q).$$

  - **Local Previous Frame Matching**

$$\hat{G}_{t,o}(p) = \begin{cases} \min_{q \in \mathcal{P}_{t-1,o}} d(p,q) & \text{if } \mathcal{P}_{t-1,o} \neq \emptyset \\ 1 & \text{otherwise} \end{cases}$$

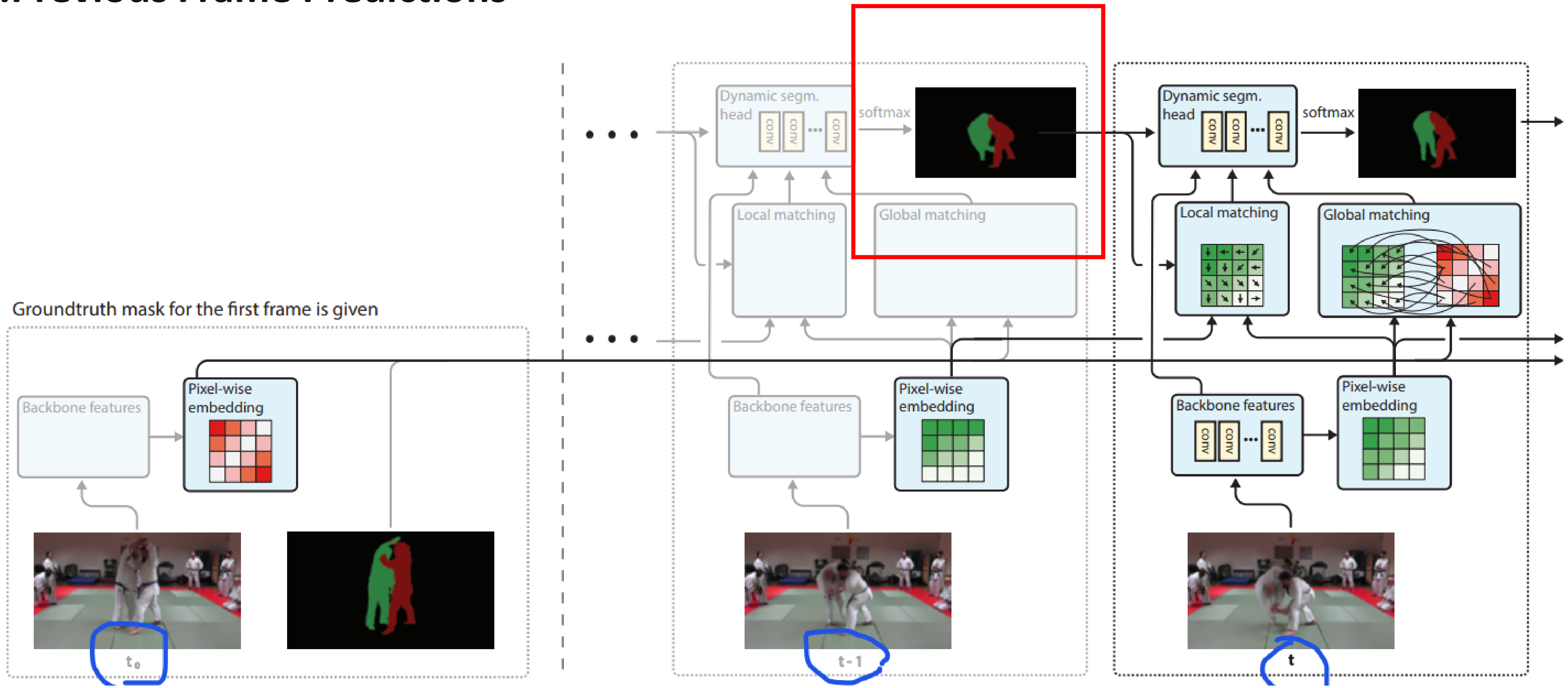


A given window size K , we only comprises $(2 * K + 1)^2$ elements

$$L_{t,o}(p) = \begin{cases} \min_{q \in \mathcal{P}^{p}_{t-1,o}} d(p,q) & \text{if } \mathcal{P}^{p}_{t-1,o} \neq \emptyset \\ 1 & \text{otherwise,} \end{cases}$$

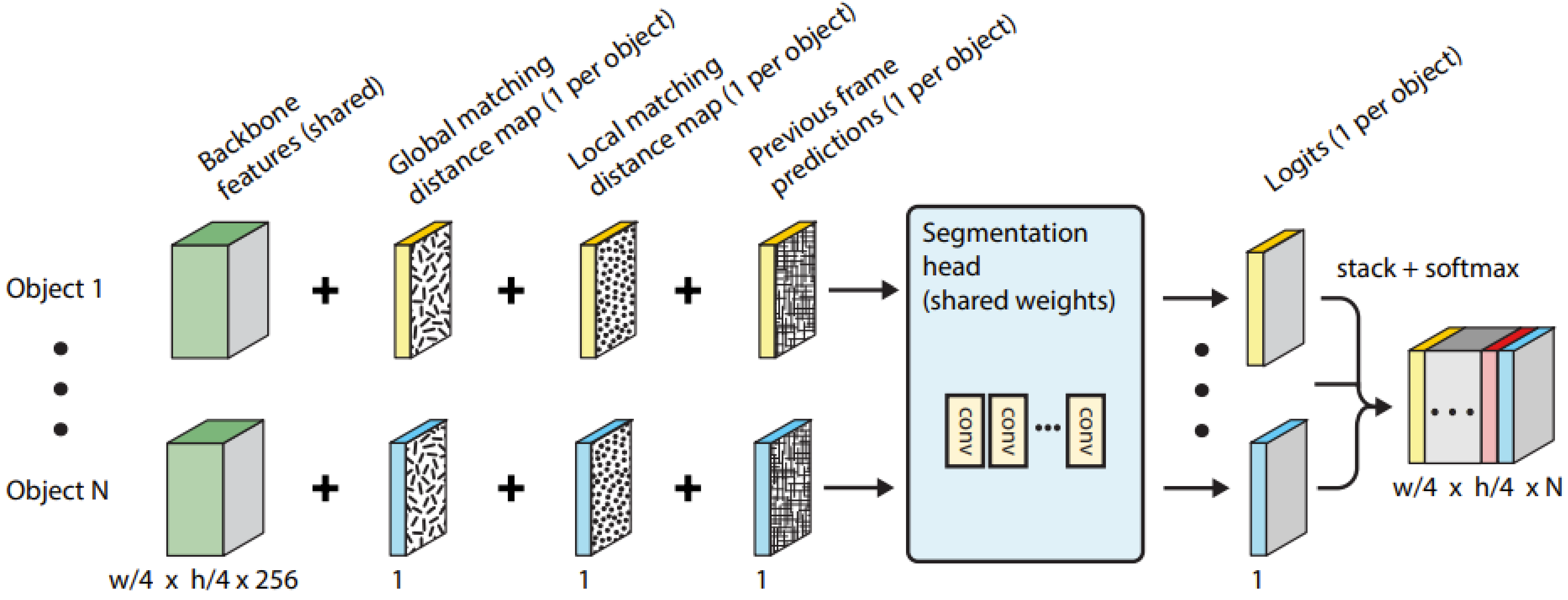
02 FEELVOS: Fast End-to-End Embedding Learning for Video Object Segmentation

- **3.Previous Frame Predictions**



02 FEELVOS: Fast End-to-End Embedding Learning for Video Object Segmentation

- **Dynamic Segmentation Head**



- **4 depthwise separable convolutional layers:**
  a dimensionality of 256, a kernel size of 7 $\times$ 7
- **A ReLU activation function.**

02 FEELVOS: Fast End-to-End Embedding Learning for Video Object Segmentation

- **Training Details**

  - **Backbone**

    Using weights for DeepLabv3+ which were pre-trained on COCO
  - **Global Matching**

    Randomly subsample the pixels from the first frame to contain at most 1024 pixels per object
  - **Local Previous Frame Matching**

    K = 15
  - **Dataset**
    - 3 frames
    - DAVIS 2017 training set (60 videos) and YouTube-VOS training set (3471 videos).
  - **Loss**

    Bootstrapped cross entropy loss, which only takes into account the 15% hardest pixels for calculating the loss

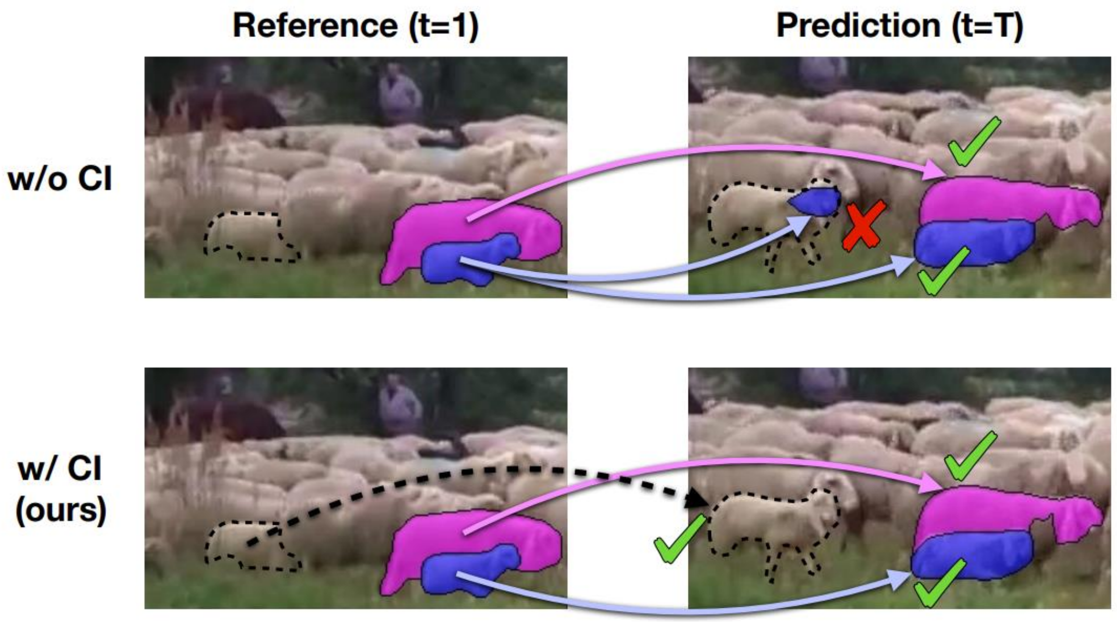
	FF-GM	PF-LM	PF-GM	PFP	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
1	✓	✓		✓	65.9	72.3	69.1
2	✓		✓	✓	61.2	67.3	64.2
3	✓			✓	49.9	59.8	54.9
4	✓				47.3	57.9	52.6
5	✓	✓			60.4	66.2	63.3
6		✓		✓	53.8	58.3	56.1

03 CFBI：Collaborative Video Object Segmentation by Multi-Scale Foreground-Background Integration（ECCV2020）

- **Motivation**
  - Background should be equally treated

- **Contribution**
  - . Global Matching and Local Matching
    Forground and background
    Pixel-level matching and instance-level embedding

  - FPN multiscale matching.
  - A Atrous Matching (AM) algorithm, which can significantly save computation and memory usage of matching processes.



03 CFBI: Collaborative Video Object Segmentation by Multi-Scale Foreground-Background Integration

- **Collaborative Pixel-level Matching**

$$D(p,q) = \begin{cases} 1 - \frac{2}{1+exp(||e_p - e_q||^2 + \boxed{b_B})} & \text{if } q \in B_t \\ 1 - \frac{2}{1+exp(||e_p - e_q||^2 + \boxed{b_F})} & \text{if } q \in F_t \end{cases}$$

b_B and b_F are trainable background bias and foreground bias

- **Compared with FEELVOS**

$$d(p,q) = 1 - \frac{2}{1 + \exp(||e_p - e_q||^2)}.$$

$$L_{t,o}(p) = \begin{cases} \min_{q \in \mathcal{P}^p_{t-1,o}} d(p,q) & \text{if } \mathcal{P}^p_{t-1,o} \neq \emptyset \\ 1 & \text{otherwise,} \end{cases}$$

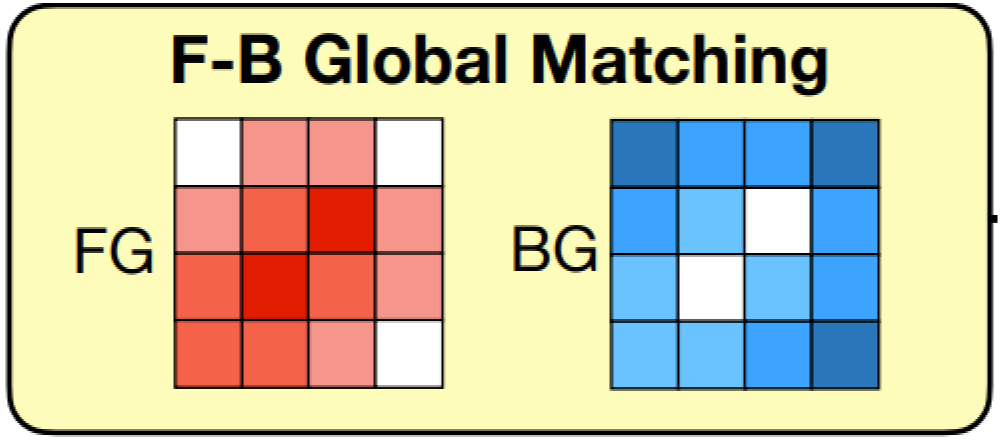
03 CFBI: Collaborative Video Object Segmentation by Multi-Scale Foreground-Background Integration

• Collaborative Pixel-level Matching

- **Foreground-Background Global Matching**
  (from t = 0 frame)

$$G_o(p) = \min_{q \in \mathcal{P}_{1,o}} D(p, q).$$

$$\overline{G}_o(p) = \min_{q \in \overline{\mathcal{P}}_{1,o}} D(p, q).$$



03 CFBI: Collaborative Video Object Segmentation by Multi-Scale Foreground-Background Integration

- **Collaborative Pixel-level Matching**
  - **Foreground-Background Multi-Local Matching**

    (from t = T-1 frame)

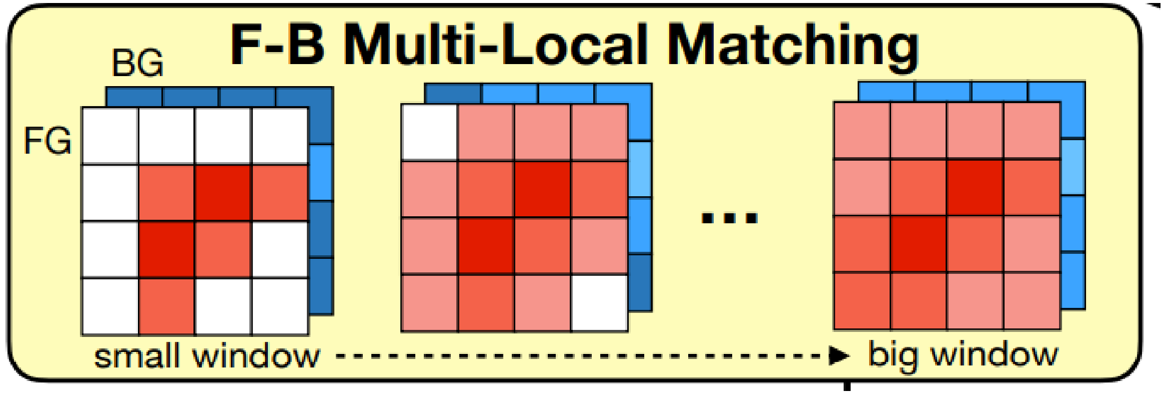
(a) Slow moving rate (b) Fast moving rate

$$ML_{T,o}(p,K) = \{L_{T,o}(p,k_1), L_{T,o}(p,k_2), ..., L_{T,o}(p,k_n)\},$$

$$L_{T,o}(p,k) = \begin{cases} \min_{q\in\mathcal{P}^{p,k}_{T-1,o}} D_{T-1}(p,q) & \text{if } \mathcal{P}^{p,k}_{T-1,o}\neq\emptyset \\ 1 & \text{otherwise} \end{cases}.$$

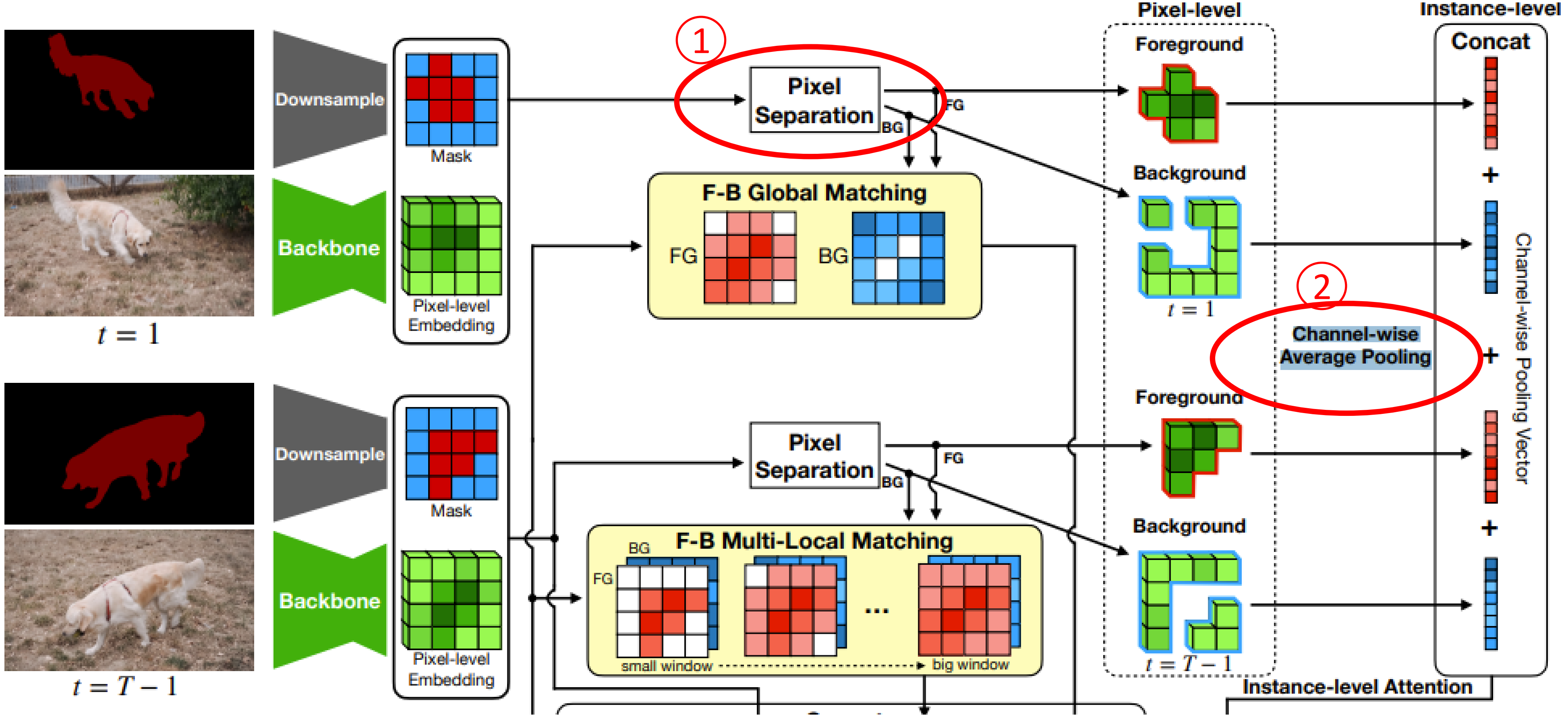
$$\overline{ML}_{T,o}(p,K) = \{\overline{L}_{T,o}(p,k_1), \overline{L}_{T,o}(p,k_2), ..., \overline{L}_{T,o}(p,k_n)\},$$

$$\overline{L}_{T,o}(p,k) = \begin{cases} \min_{q\in\overline{\mathcal{P}}^{p,k}_{T-1,o}} D_{T-1}(p,q) & \text{if } \overline{\mathcal{P}}^{p,k}_{T-1,o}\neq\emptyset \\ 1 & \text{otherwise} \end{cases}.$$



03 CFBI: Collaborative Video Object Segmentation by Multi-Scale Foreground-Background Integration

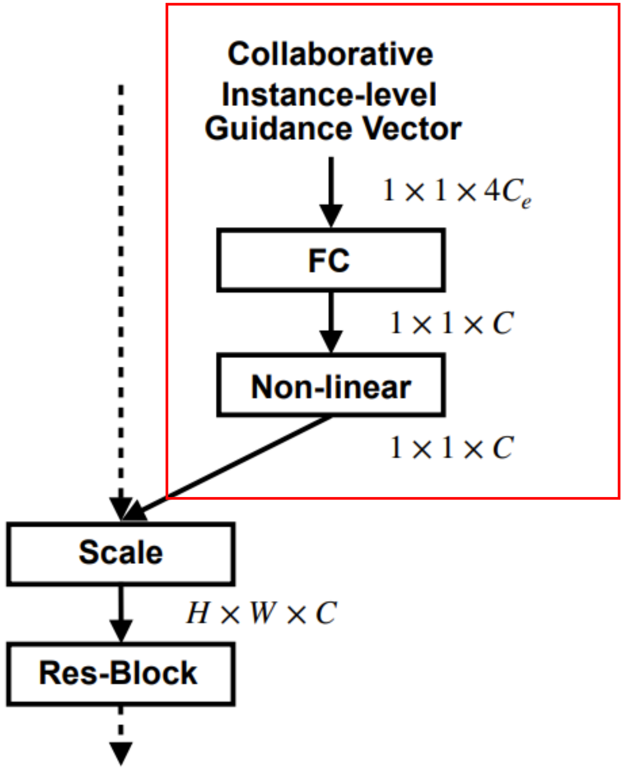
- **Collaborative Instance-level Attention**
  - **Get guidance vector**



03 CFBI: Collaborative Video Object Segmentation by Multi-Scale Foreground-Background Integration

- **Collaborative Instance-level Attention**
  - **Attention mechanism**

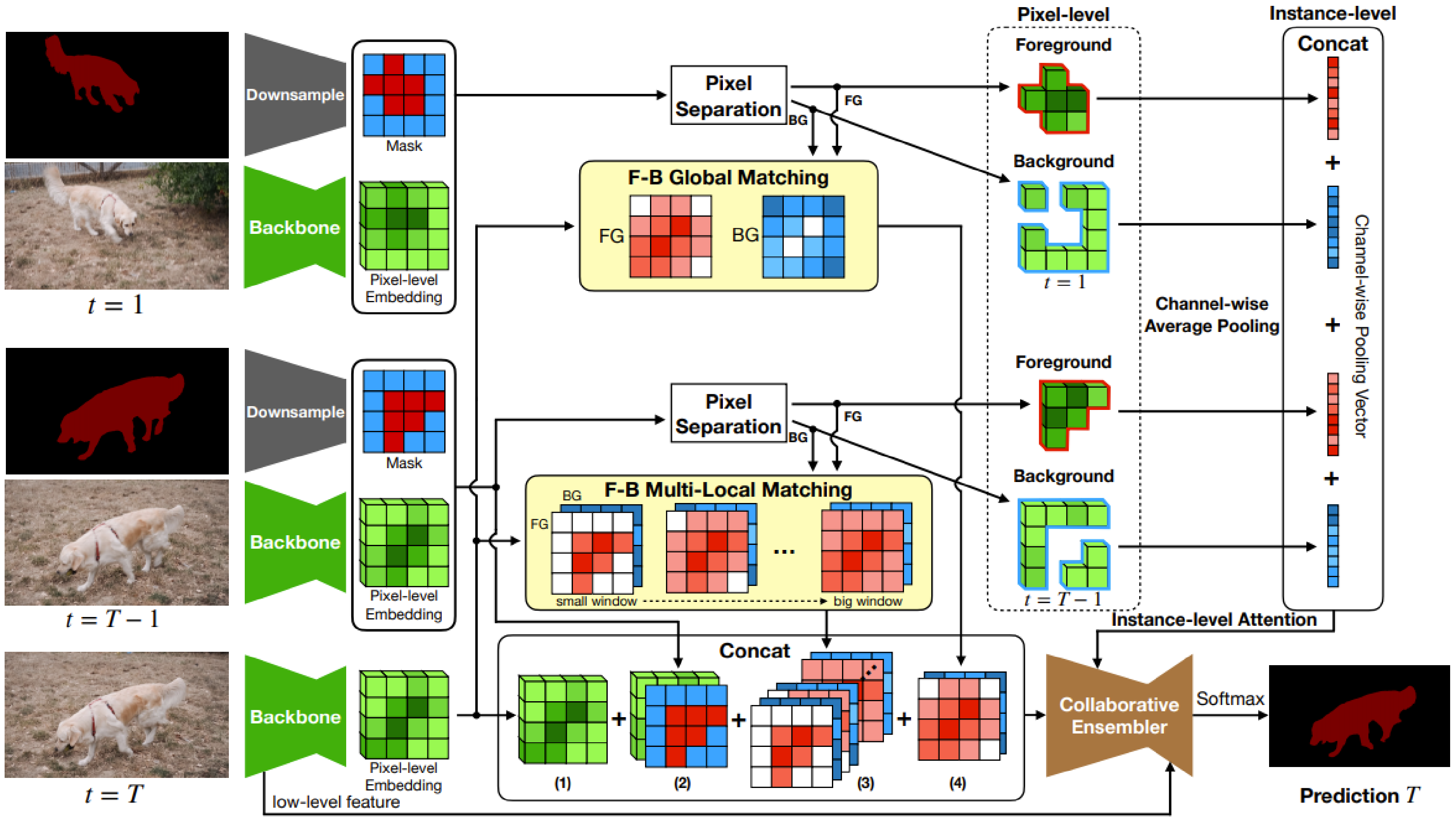
1. Concat the guidance vector
2. a fully-connected (FC) layer
3. a non-linear activation function
4. Give each channel a weight



Leverage a full scale of foreground-background information to guide the prediction further

03 CFBI: Collaborative Video Object Segmentation by Multi-Scale Foreground-Background Integration

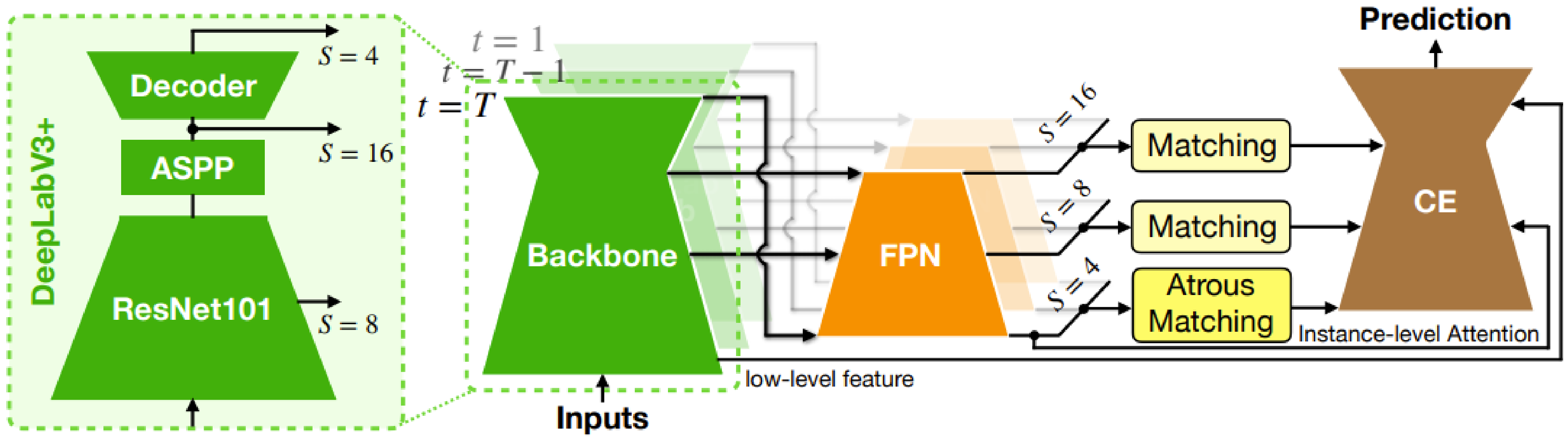
- **An overview of CFBI**



① Backbone 的feature
② 前一帧的Mask
③ Local Matching
④ Global Matching

03 CFBI: Collaborative Video Object Segmentation by Multi-Scale Foreground-Background Integration

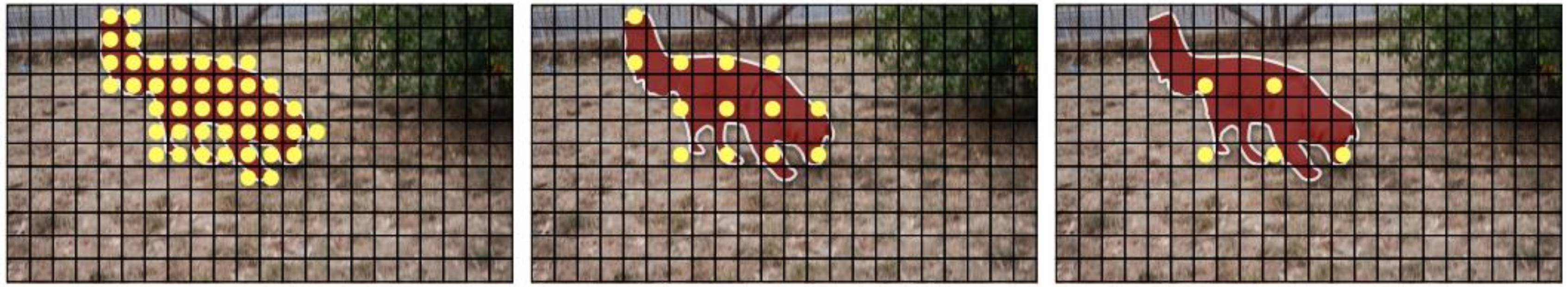
- **Multi-scale Matching(CFBI+)**



strides	Channel	Window sizes
4	32	{4, 8, 12, 16, 20, 24}
8	64	{2, 4, 6, 8, 10, 12}
16	128	{4, 6, 8, 10}

03 CFBI: Collaborative Video Object Segmentation by Multi-Scale Foreground-Background Integration

- **Atrous Matching (AM)**



(a) $l = 1$ (original matching) (b) $l = 2$ (c) $l = 3$

Matching

$$G_o(p) = \min_{q \in \mathcal{P}_{1,o}} D(p,q).$$

Atrous Matching (AM)

$$G_o^l(p) = \min_{q \in \mathcal{P}_{1,o}^l} D(p,q), \qquad \mathcal{P}_{1,o}^l = \{q_{x,y} \in \mathcal{P}_{1,o}, \forall x, y \in \{l, 2l, 3l, ...\}\}$$

03 CFBI: Collaborative Video Object Segmentation by Multi-Scale Foreground-Background Integration

- **Training details**

TABLE 4
Ablation of background embedding on the DAVIS-2017 validation split. P and I denote the pixel-level matching and instance-level attention, respectively. *: removing the foreground and background bias.

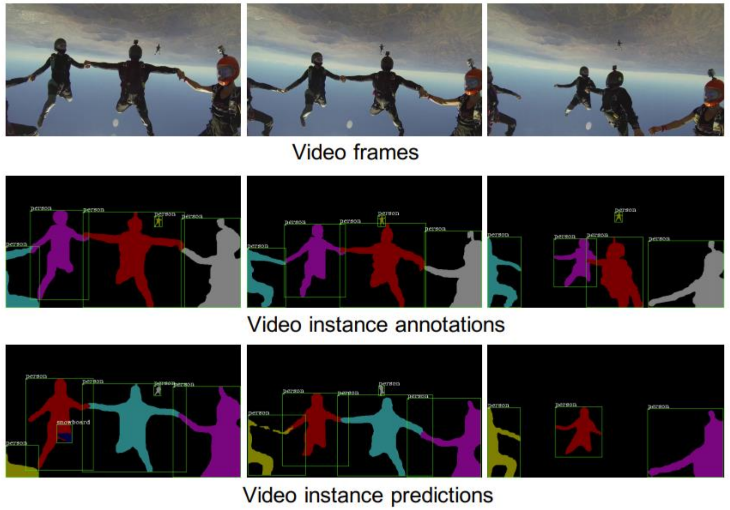
P	I	Avg	$\mathcal{J}$	$\mathcal{F}$
✓	✓	74.9	72.1	77.7
✓*	✓	72.8	69.5	76.1
✓		73.0	69.9	76.0
	✓	72.3	69.1	75.4
		70.9	68.2	73.6

TABLE 5
Ablation of atrous matching. We evaluate the speed and performance of CFBI on the YouTube-VOS validation split using different atrous matching factors (l). $l = 1$ is equivalent to original matching.

l	1	2	3	4
Global Matching				
Avg	81.4	81.3	80.7	79.9
t/s	0.29	0.15	0.13	0.12
Multi-local Matching				
Avg	81.4	80.8	80.1	79.5
t/s	0.29	0.26	0.25	0.25

04 Video Instance Segmentation

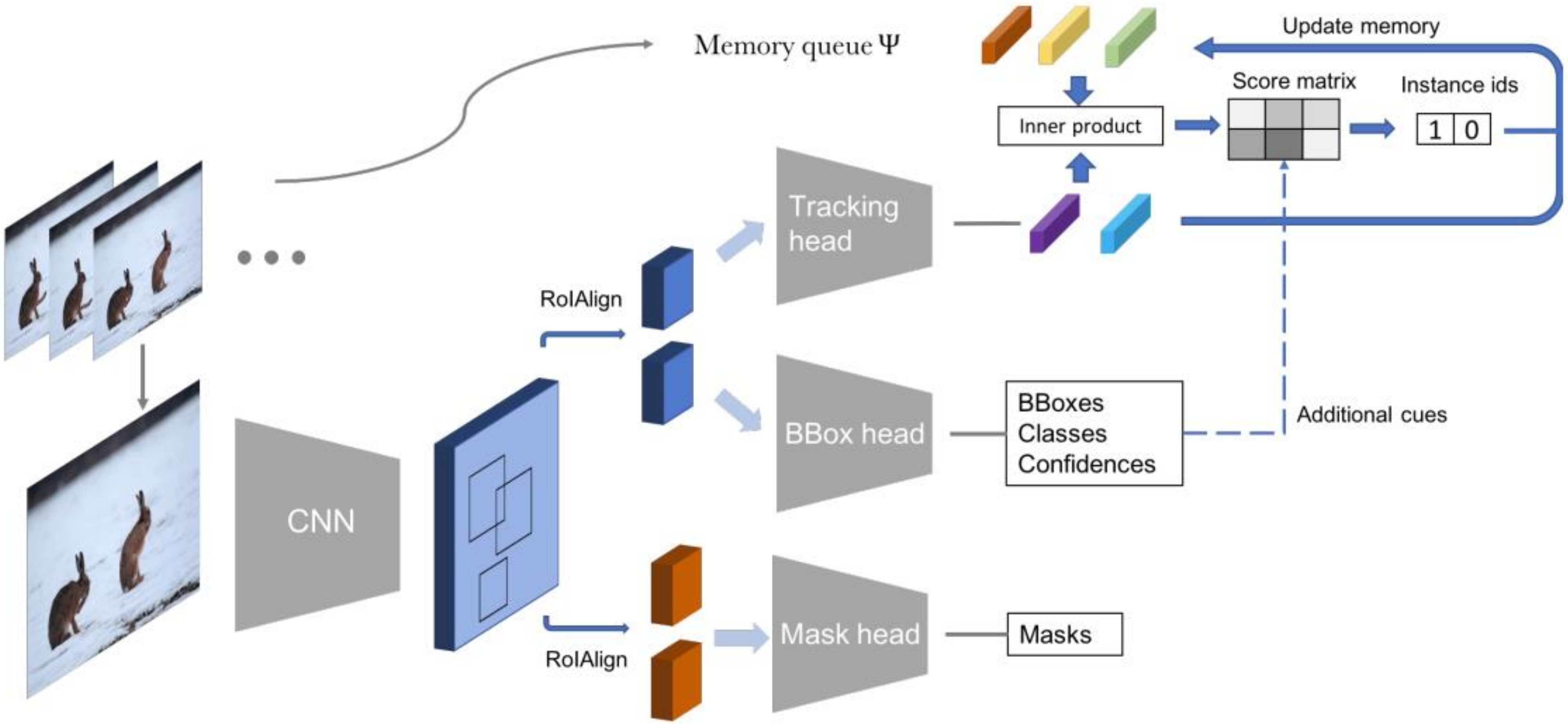
ICCV2019



- **Contribution**
  - **1) A new computer vision task**
    - The goal of this new task is simultaneous **detection, segmentation and tracking** of instances in videos
  - **2) Propose a large-scale benchmark called YouTube-VIS**
    - Based on YouTube-VOS
    - 2883 high-resolution videos and 40 common categories.
  - **3) Propose a novel algorithm called MaskTrack R-CNN for this task**
    - A tracking branch to Mask R-CNN to jointly perform the detection, segmentation and tracking tasks simultaneously.

04 Video Instance Segmentation

- **An overview of MaskTrack R-CNN**



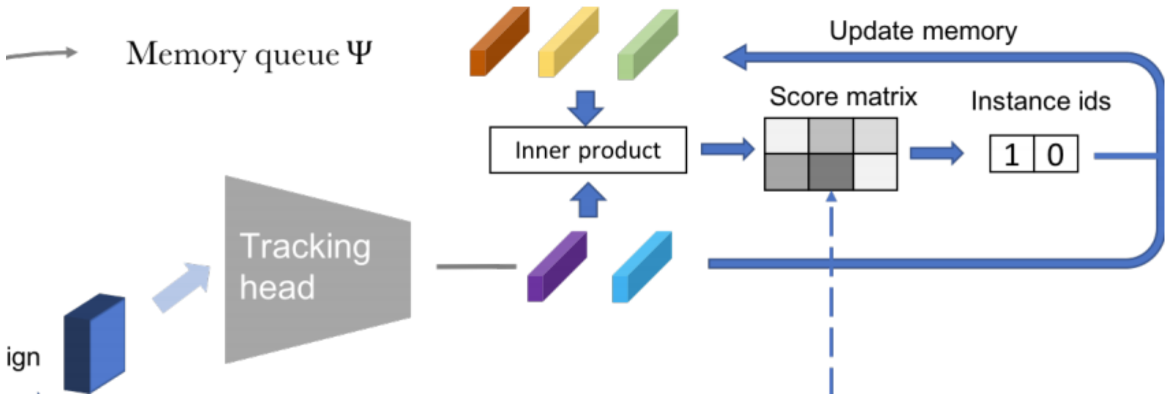
04 Video Instance Segmentation

• New Tracking Branch

- Two fully connected layers.

The first fully connected layer transforms the 7 × 7 × 256 input feature maps to 1-D 1024 dimensions.
The second fully connected layer also maps its input to 1-D 1024 dimensions.

$$p_i(n) = \begin{cases} \frac{e^{\mathbf{f}_i^\mathsf{T} \mathbf{f}_n}}{1+\Sigma_{j=1}^{N} e^{\mathbf{f}_i^\mathsf{T} \mathbf{f}_j}} & n \in [1, N] \\ \frac{1}{1+\Sigma_{j=1}^{N} e^{\mathbf{f}_i^\mathsf{T} \mathbf{f}_j}} & n = 0 \end{cases}$$



$$L_{track} = -\Sigma_i \log(p_i(y_i))$$

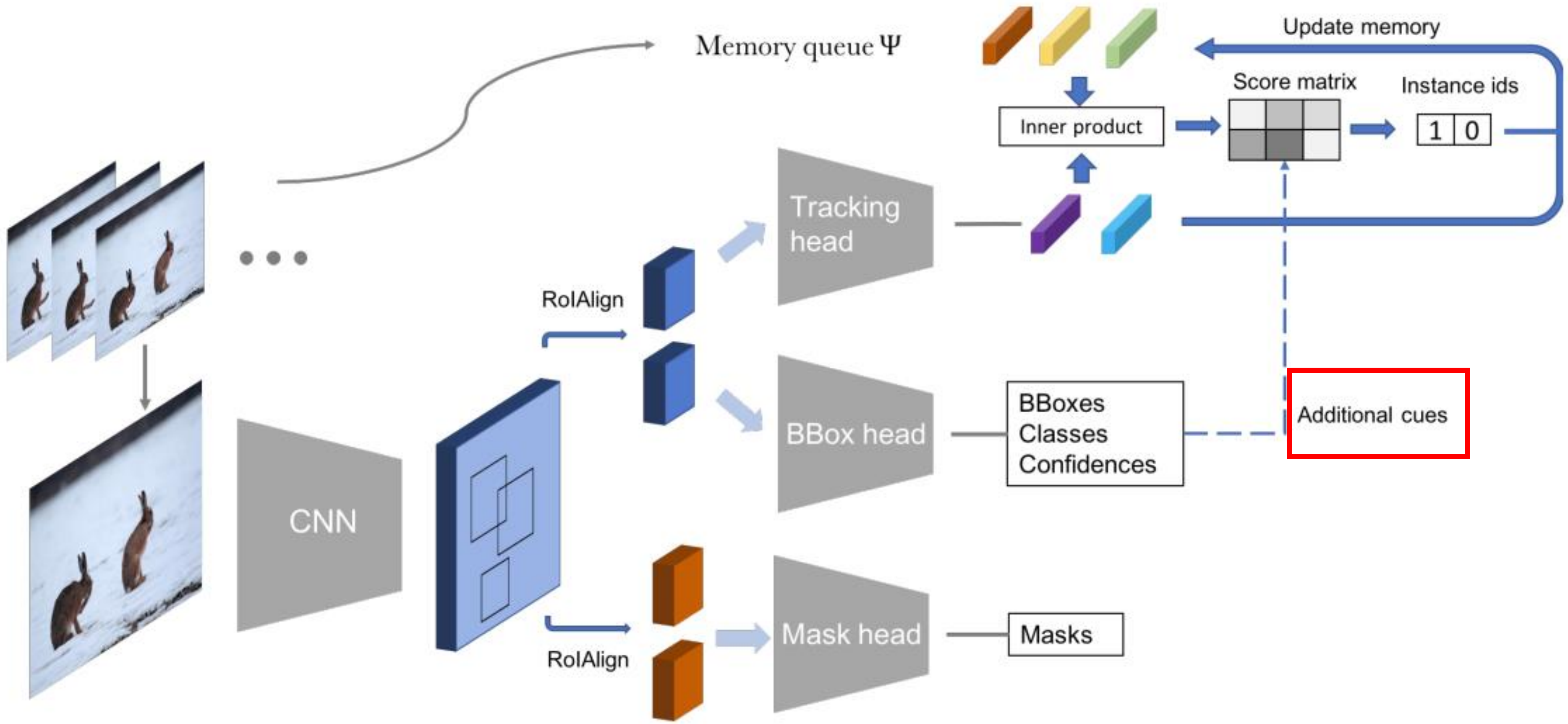
y_i is the ground truth label.

04 Video Instance Segmentation

- **Combining Other Cues**

$$v_i(n) = \log p_i(n) + \alpha \log s_i + \beta \mathrm{IoU}(b_i, b_n) + \gamma \delta(c_i, c_n)$$

b_i, c_i and s_i denote its bounding box prediction, category label and detection score



04 Video Instance Segmentation

• Evaluation Metrics

- **1) IoU**

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

- **2) AP(average precision)**
  - AP is averaged over multiple intersection-over-union (IoU) thresholds
  - IoU thresholds： 10 IoU thresholds from 50% to 95% at step 5%
- **3) AR(average recall)**
  - AR is defined as the maximum recall given some fixed number of segmented instances per video.