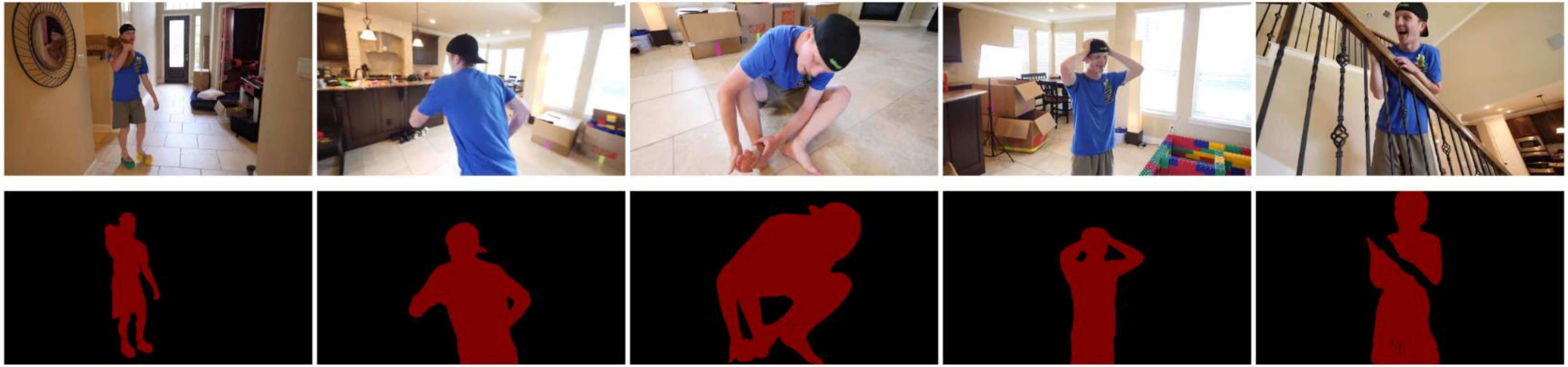
XMem: Long-Term Video Object Segmentation with an Atkinson-Shiffrin Memory Model

Ho Kei Cheng and Alexander G. Schwing

University of Illinois Urbana-Champaign
{hokeikc2, aschwing}@illinois.edu



Frame 0 (input) Frame 295 Frame 460 Frame 1285 Frame 2327

Ho Kei Cheng

其他姓名

引用次数

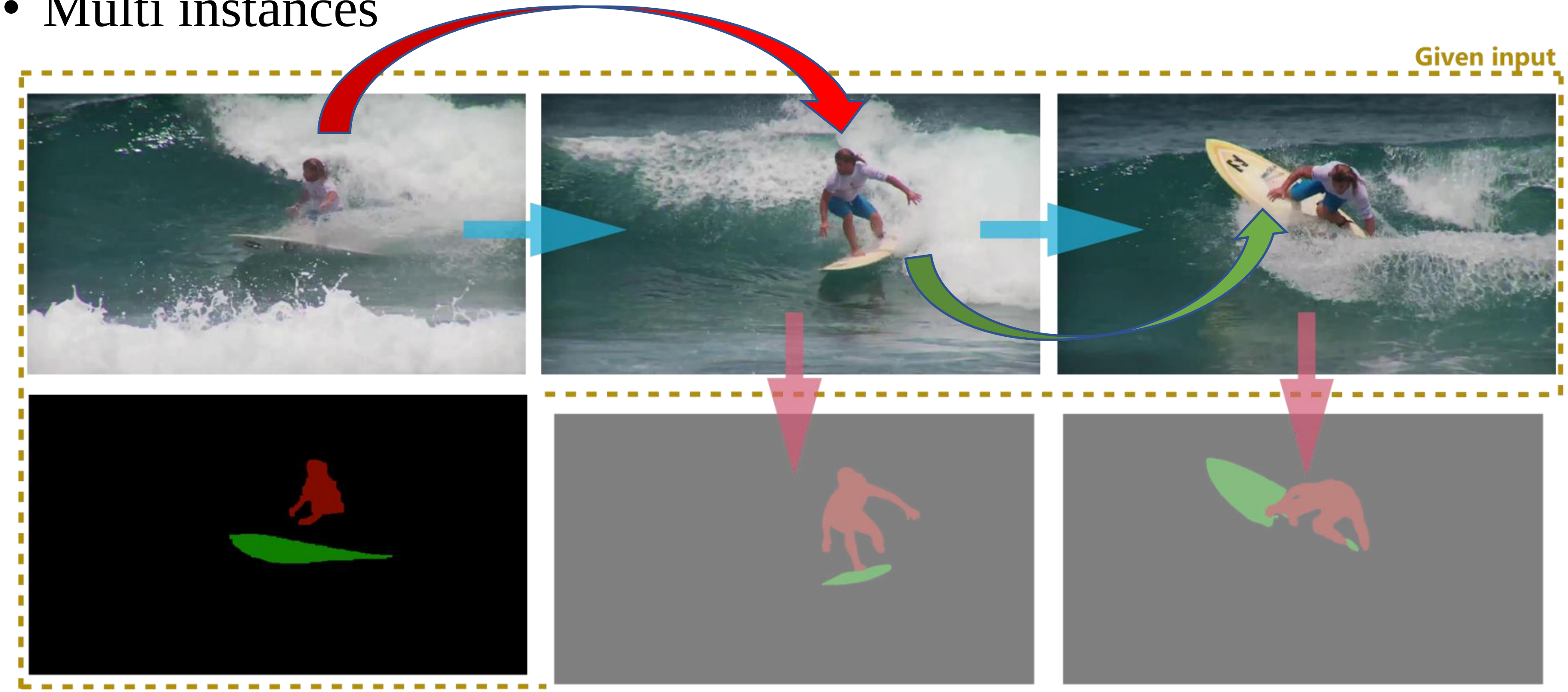
	总计	2017 年至今
引用	221	221
h 指数	3	3
i10 指数	3	3

CascadePSP: Toward Class-Agnostic and Very High-Resolution Segmentation via Global and Local Refinement HK Cheng, J Chung, YW Tai, CK Tang IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020 …	88	2020
Rethinking Space-Time Networks with Improved Memory Coverage for Efficient Video Object Segmentation HK Cheng, YW Tai, CK Tang Advances in Neural Information Processing Systems (NeurIPS), 2021	75	2021
Modular Interactive Video Object Segmentation: Interaction-to-Mask, Propagation and Difference-Aware Fusion HK Cheng, YW Tai, CK Tang IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021	56	2021
XMem: Long-Term Video Object Segmentation with an Atkinson-Shiffrin Memory Model HK Cheng, AG Schwing European Conference on Computer Vision (ECCV), 2022	2	2022

文章 1–4 展开

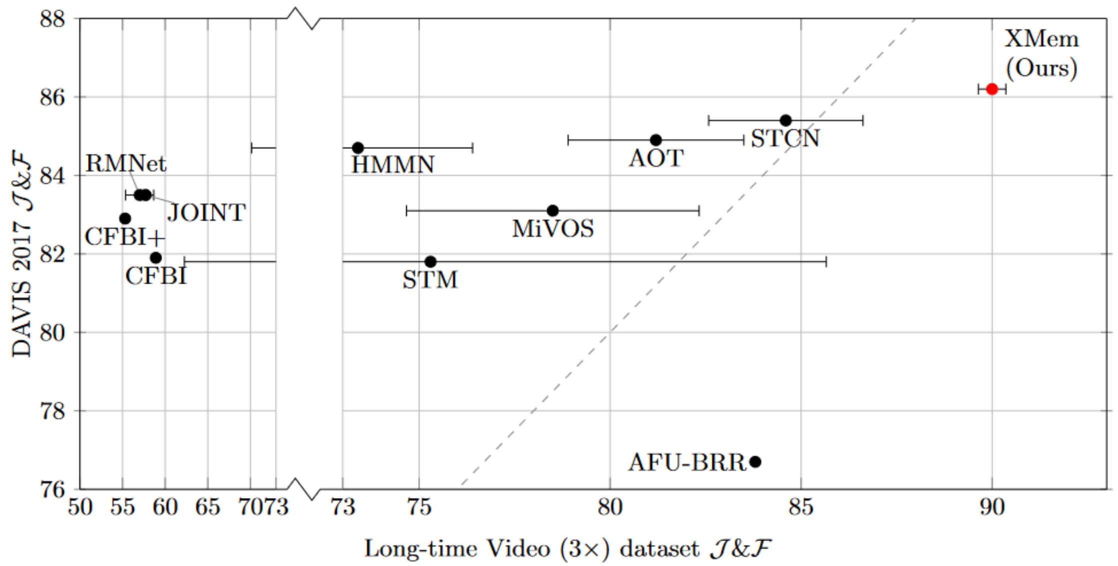
Video Object Segmentation (VOS)

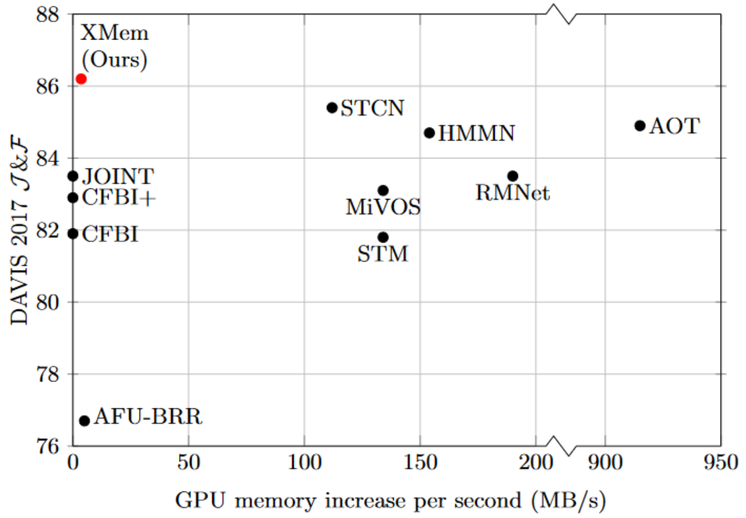
- Semi-supervised setting
  - Provide first frame annotation
  - Multi instances



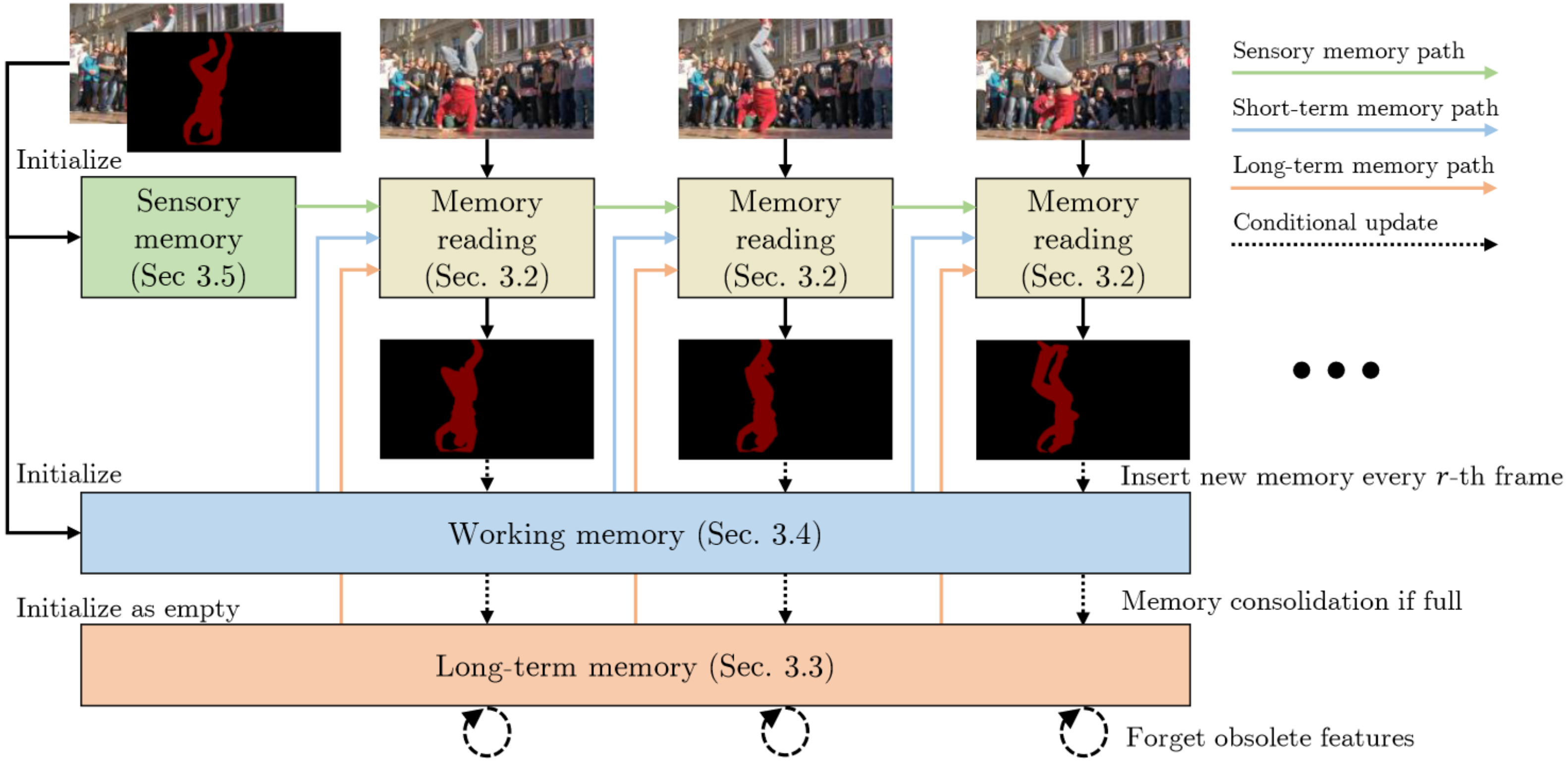
Video Object Segmentation (VOS)

- *Problem with long videos*
  - Sacrificed segmentation quality
  - **Reason**: The features of the memory frame are compressed and the information is lost
- *Problem of memory matching*
  - **Recurrent approaches**: Prone to drifting and struggle with occlusions (Low Performance);
  - **Attention based**: Required large amount of GPU
- *Contribution of **XMem***
  - **Architecture (**based on Atkinson–Shiffrin memory model**)** that can handle long video through a **Long-Term Memory**
  - New **Memory Reading** technic that can obtain good segmentation results while consuming only a small amount of GPU resources



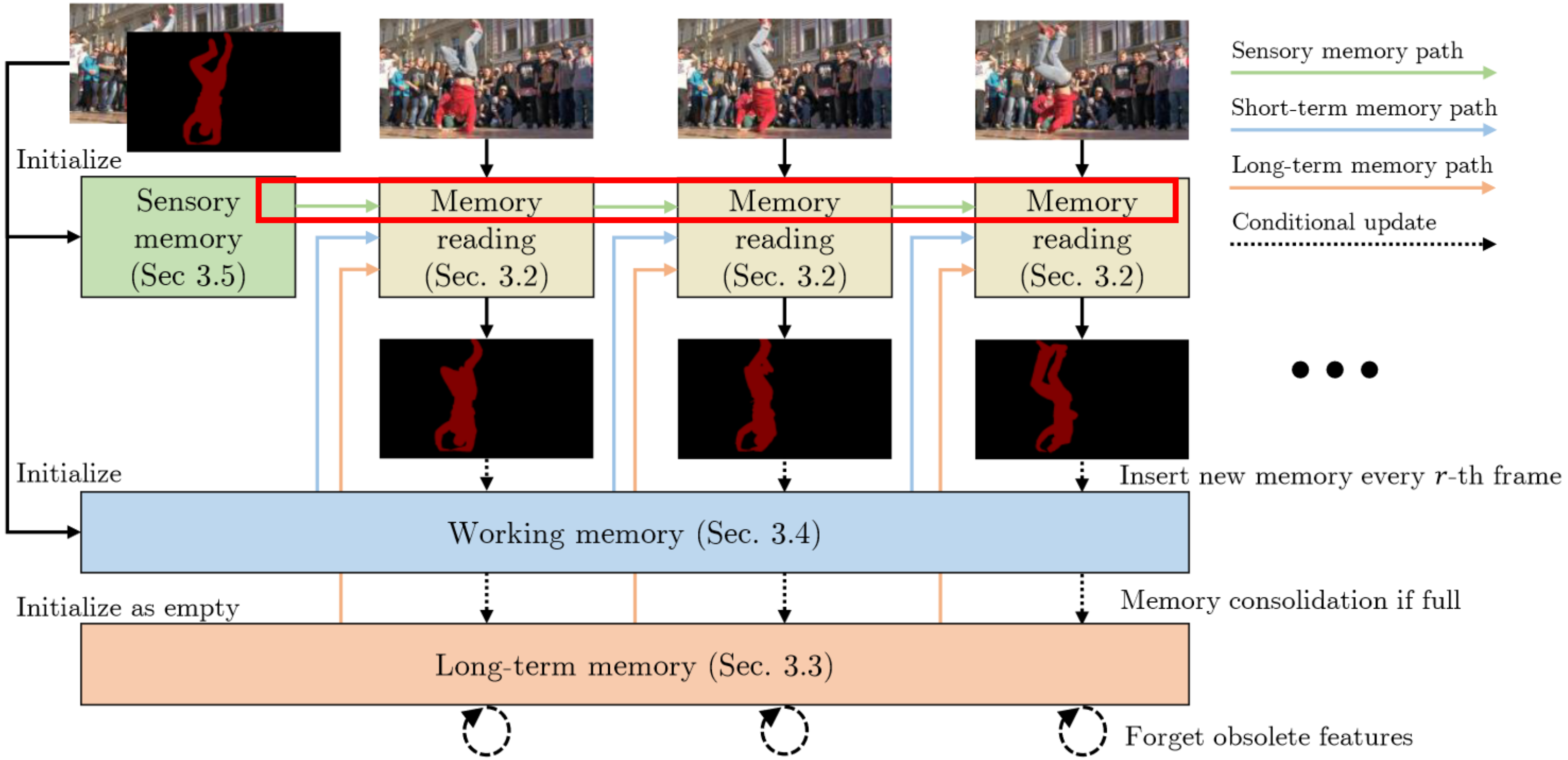


XMem —— Overview



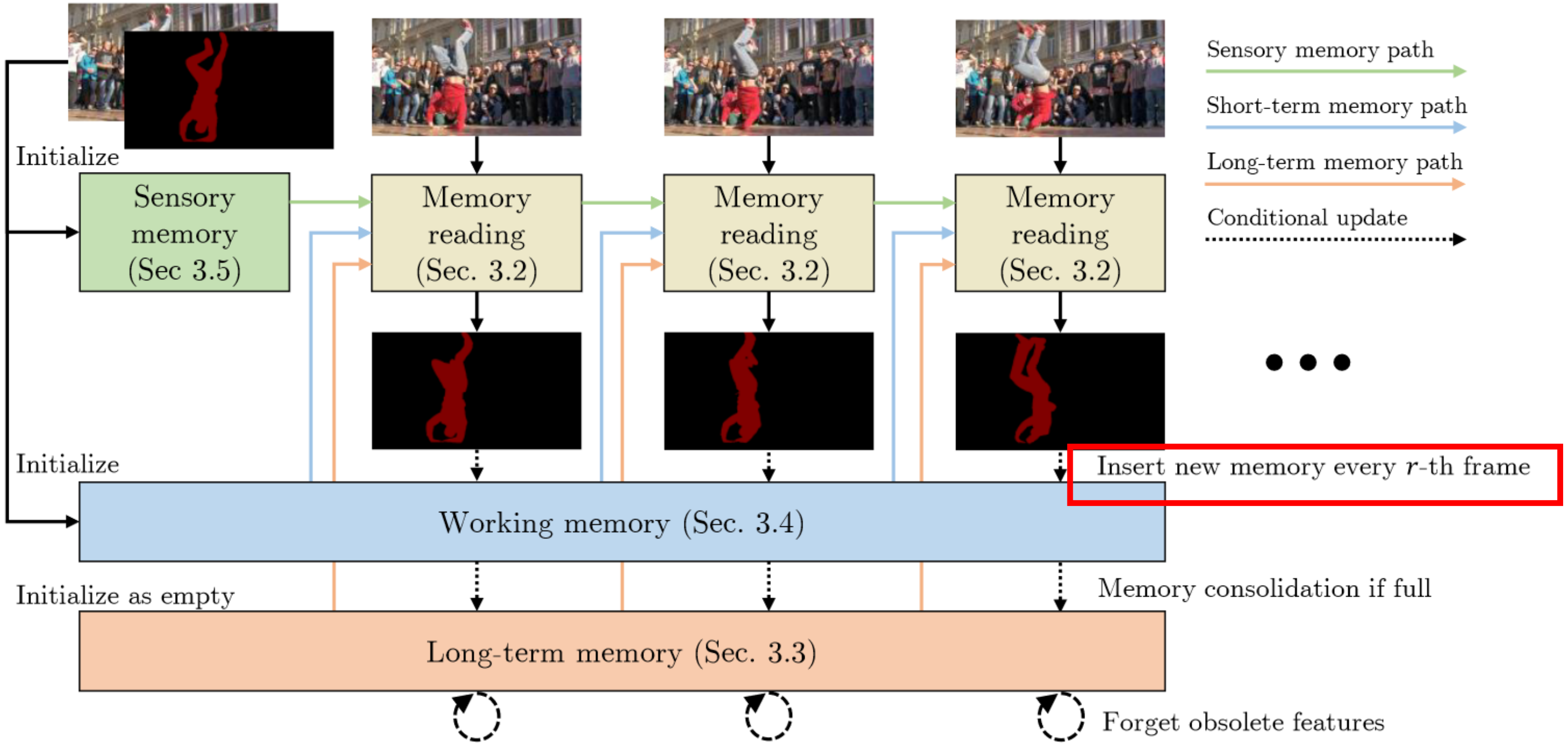
- Inspired by the Atkinson–Shiffrin memory model
  - **Sensory memory**: Cues used in decoding each frame
  - **Working memory**: Including only a few frames full memory
  - **Long-term memory**: Compressed memory of a large amount of frames

XMem —— Overview



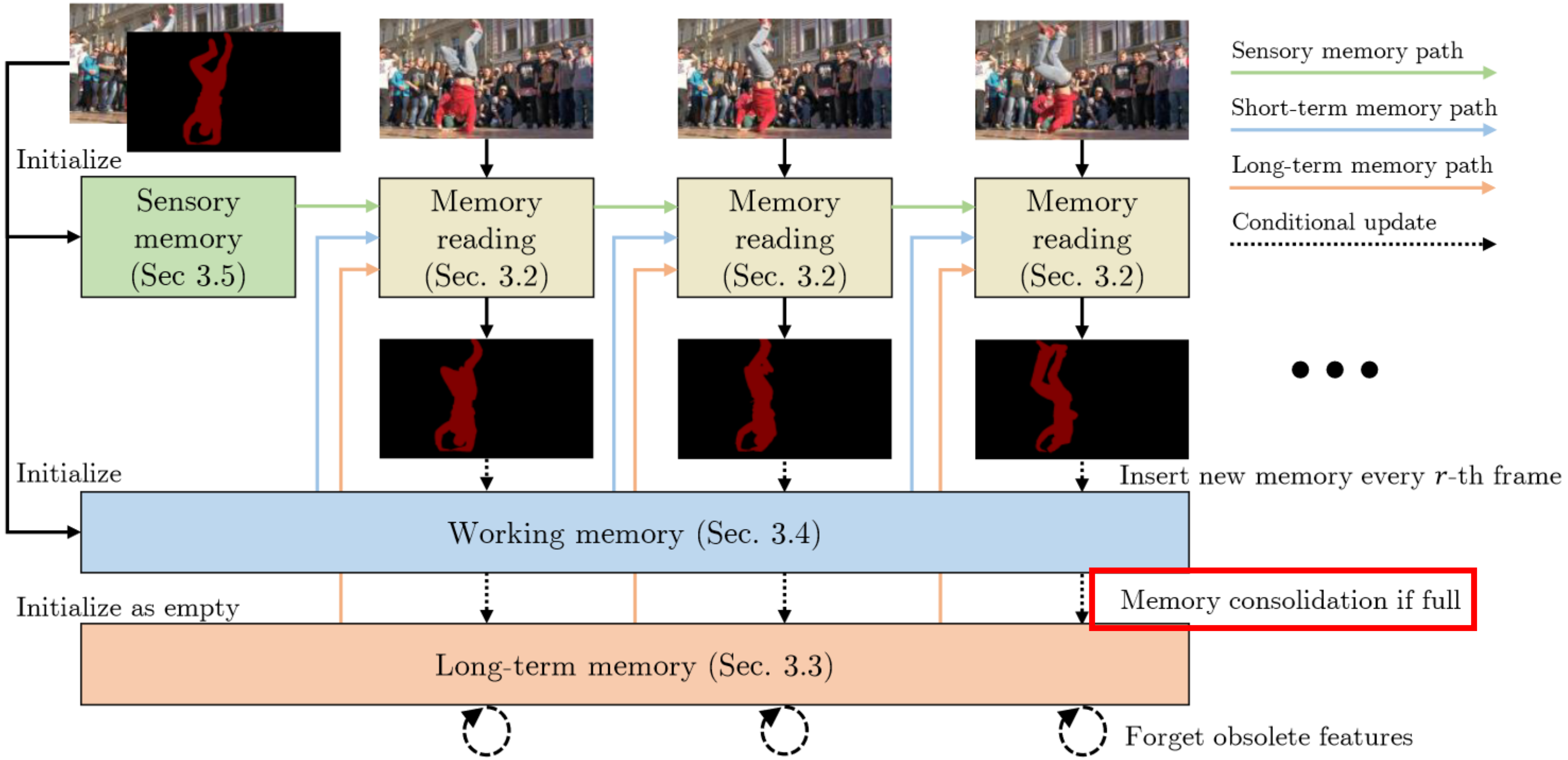
- Inspired by the Atkinson–Shiffrin memory model
  - **Sensory memory**: Cues used in decoding each frame
  - **Working memory**: Including only a few frames full memory
  - **Long-term memory**: Compressed memory of a large amount of frames

XMem —— Overview



- Inspired by the Atkinson–Shiffrin memory model
  - **Sensory memory**: Cues used in decoding each frame to improve temporal consistency
  - **Working memory**: Including only a few frames full memory
  - **Long-term memory**: Compressed memory of a large amount of frames

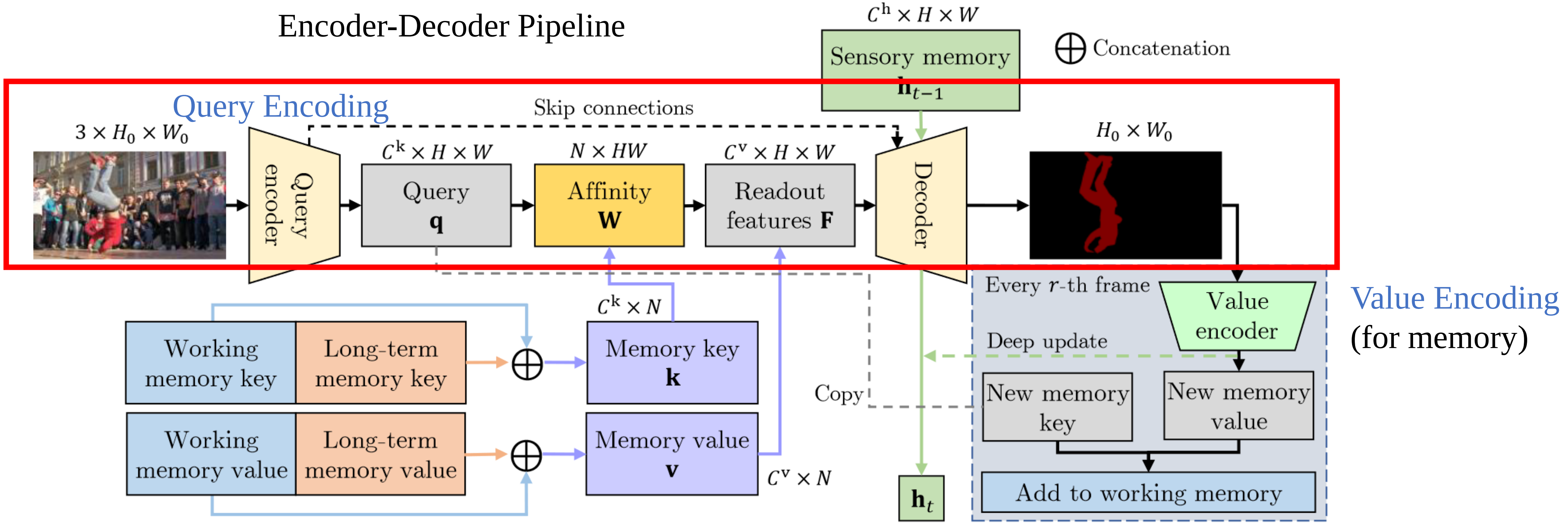
XMem —— Overview



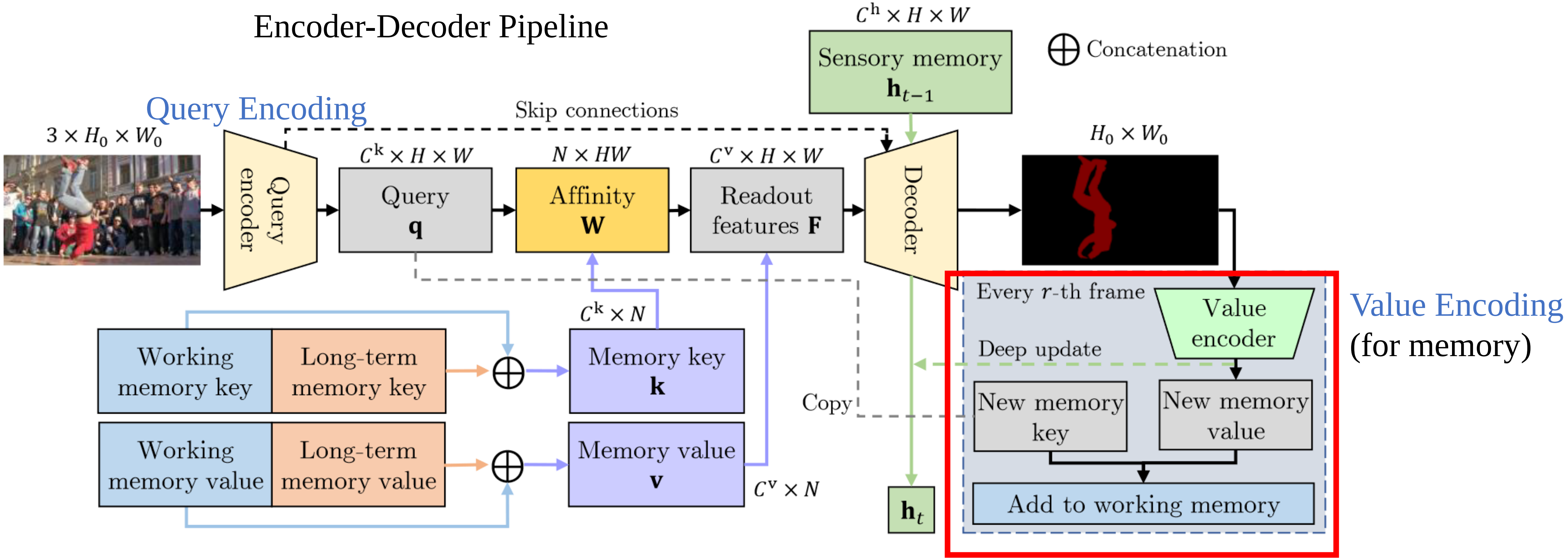
- Inspired by the Atkinson–Shiffrin memory model
  - **Sensory memory**: Cues used in decoding each frame
  - **Working memory**: Including only a few frames full memory
  - **Long-term memory**: Compressed memory of a large amount of frames

XMem —— Overview

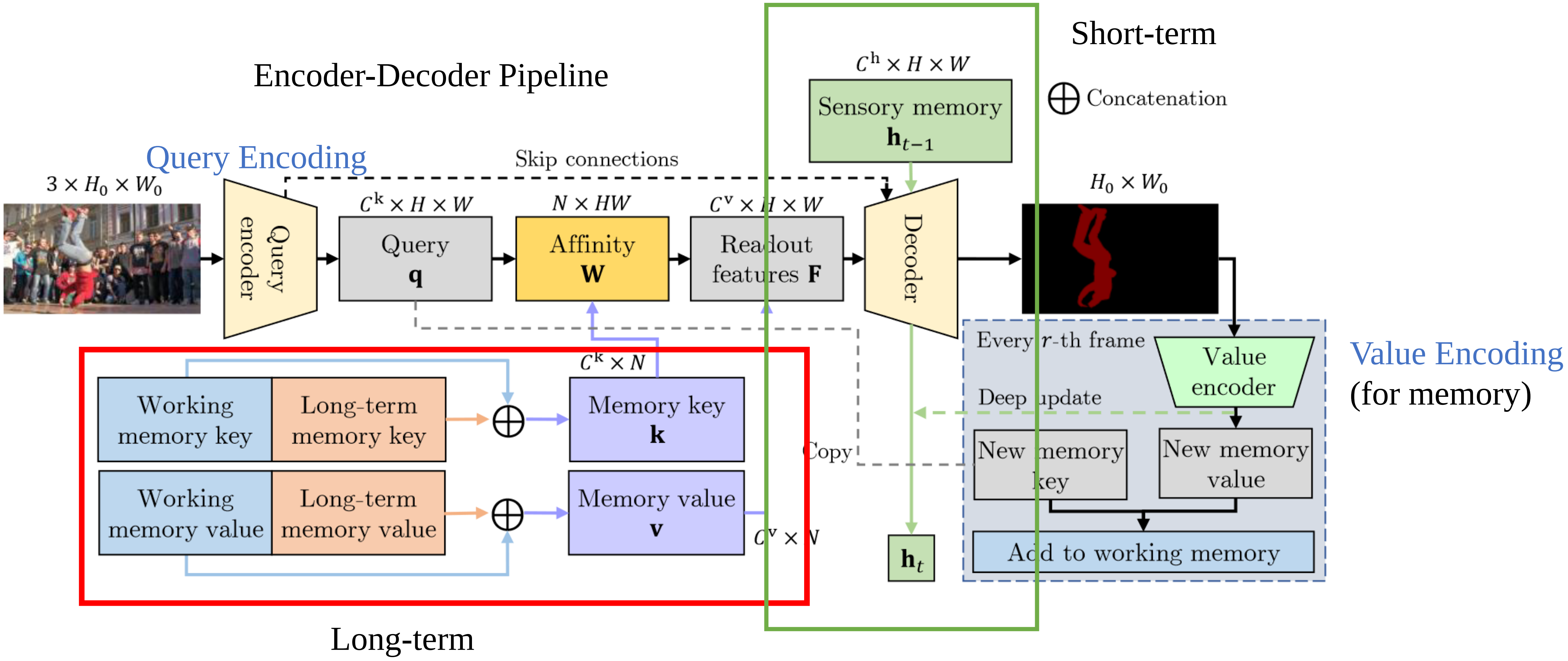
Encoder-Decoder Pipeline



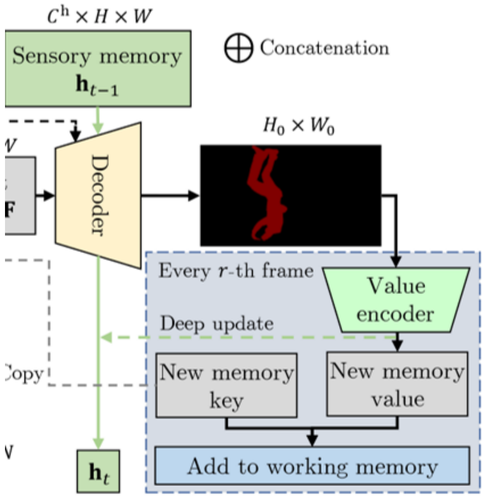
XMem —— Overview

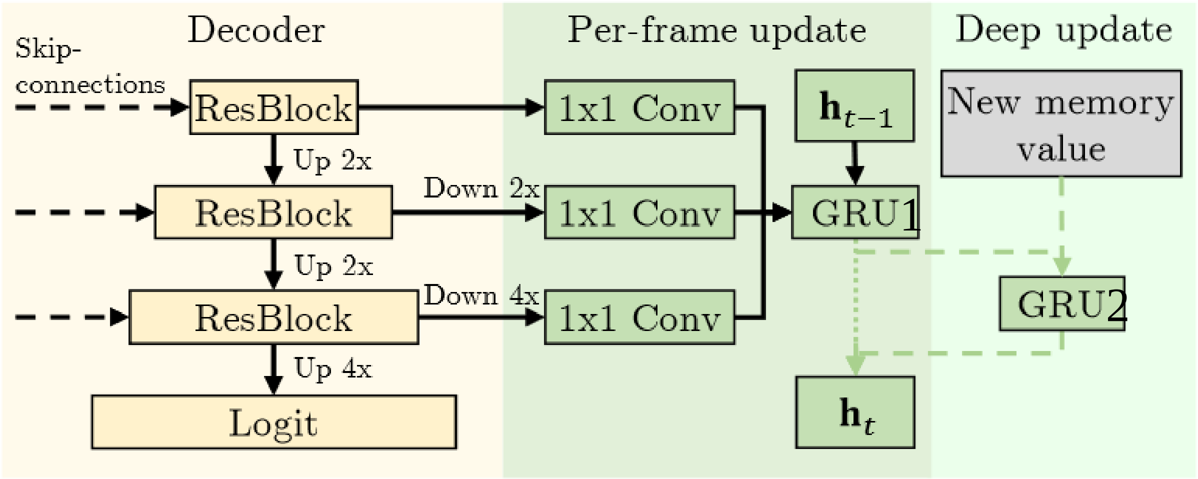


XMem —— Overview



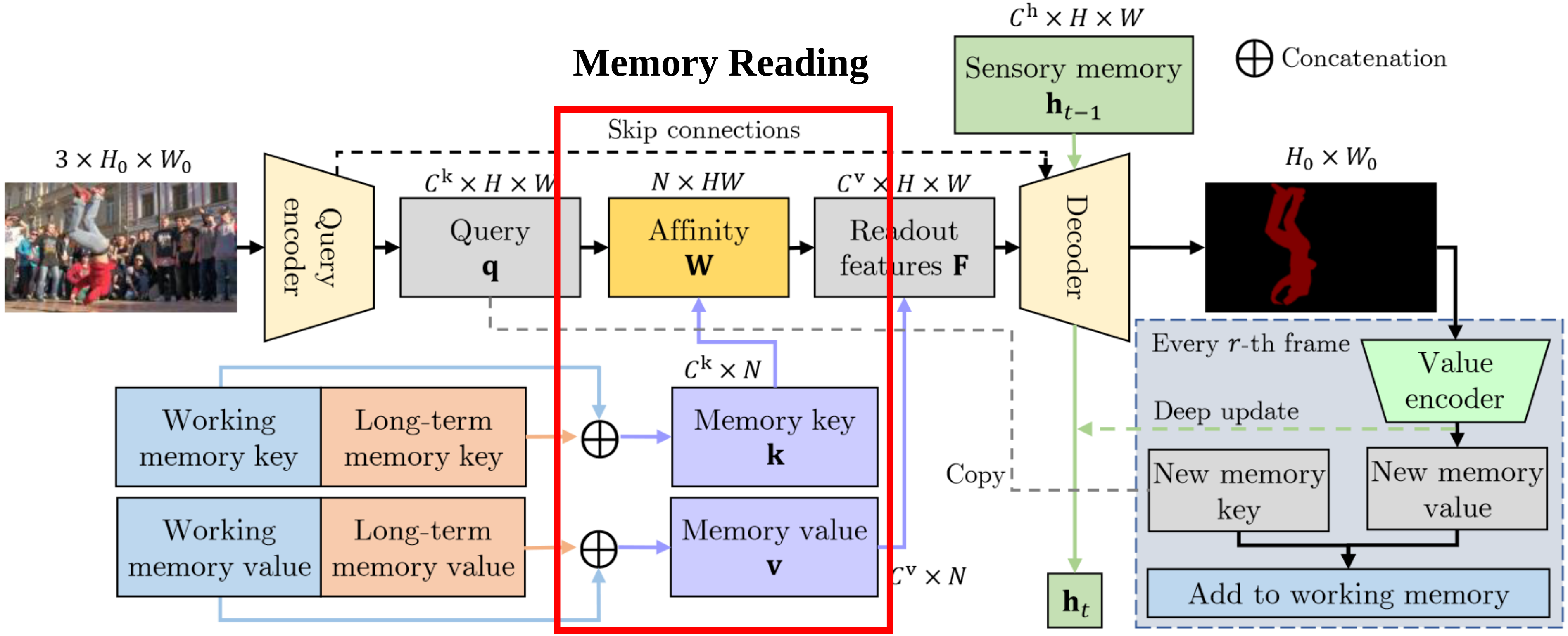
Sensory Memory





- Retains low-level information which nicely complements the lack of temporal locality in the working/long-term memory
- Hidden feature **h** of **GRU1** is updated in each frame (sensory)
- Perform *deep update* every r-th frame using new memory value with **GRU2.** Advantages are:
  - discard redundant information that has already been saved to the working memory;
  - receive updates from a deep network (i.e., the value encoder) with minimal overhead as we are reusing existing features.

XMem —— Overview



Memory Reading

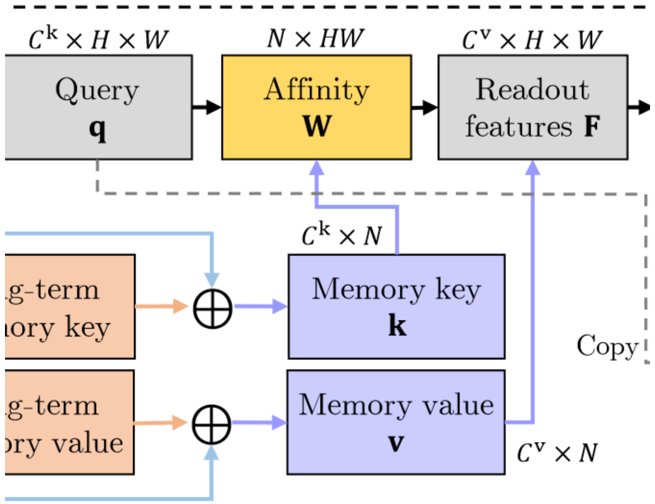
$$\mathbf{F} = \mathbf{v}\mathbf{W}(\mathbf{k}, \mathbf{q}). \qquad W(k,q) = Softmax(S(k,q)) \qquad \boxed{\mathbf{S}(\mathbf{k}, \mathbf{q})_{ij} = -\mathbf{s}_i \sum_{c}^{C^{\mathrm{k}}} \mathbf{e}_{cj} \left(\mathbf{k}_{ci} - \mathbf{q}_{cj}\right)^2,}$$

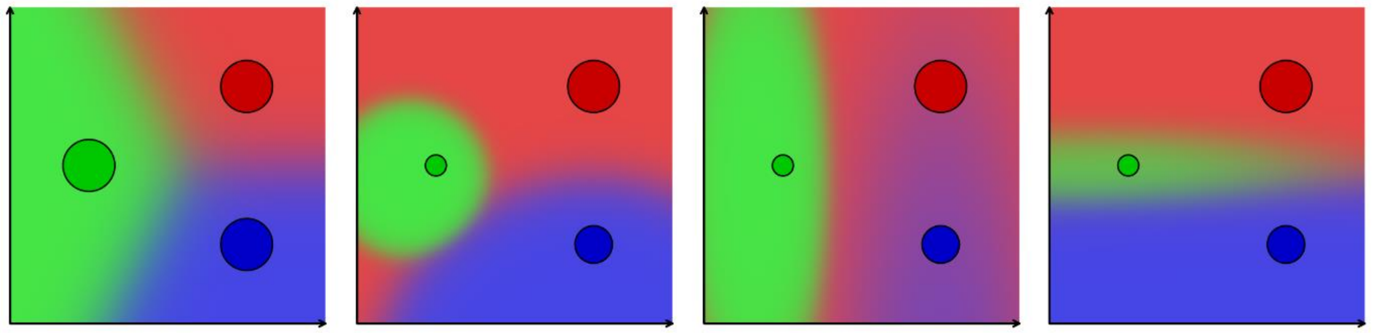
Here, $\mathbf{k} \in \mathbb{R}^{C^{\mathrm{k}} \times N}$ and $\mathbf{v} \in \mathbb{R}^{C^{\mathrm{v}} \times N}$ are C^{k}- and C^{v}-dimensional keys and values

shrinkage term $\mathbf{s} \in [1, \infty)^N$

selection term $\mathbf{e} \in [0, 1]^{C^{\mathrm{k}} \times HW}$

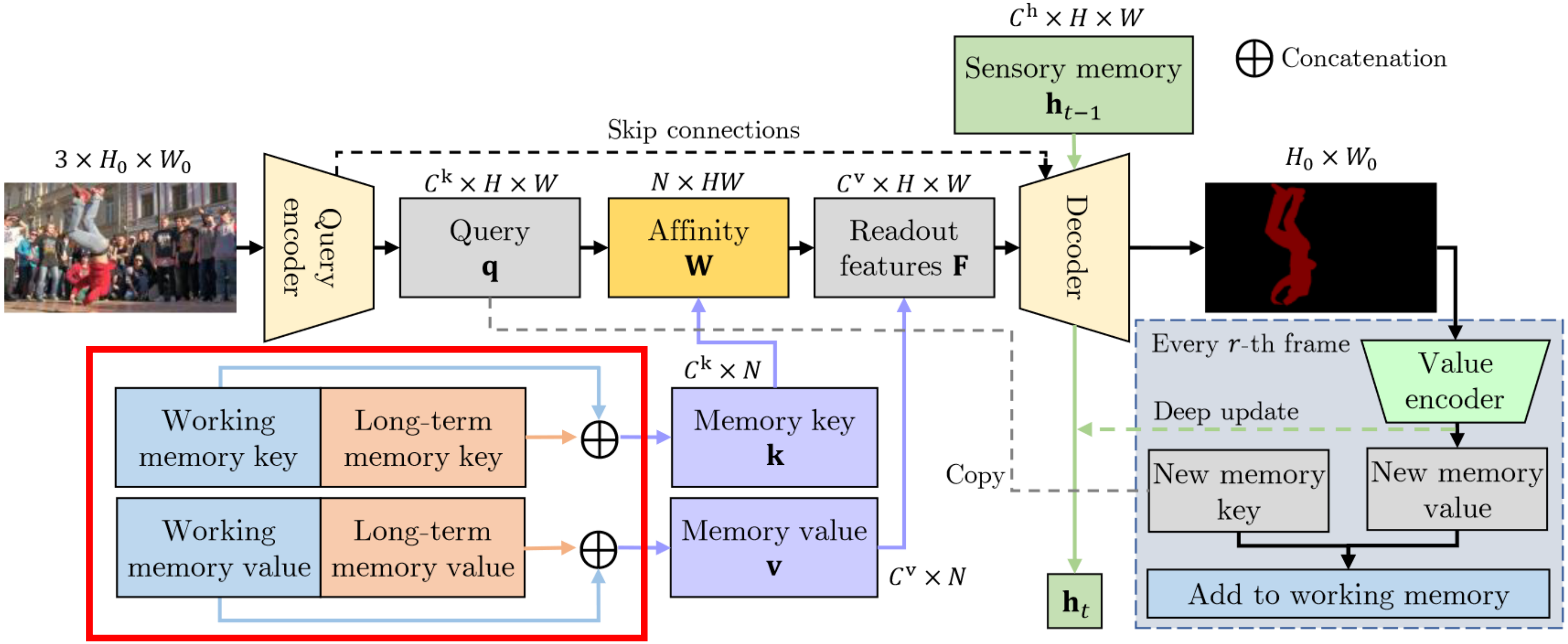
- ***s*** **directly scales the similarity and explicitly encodes confidence**
- ***e*** **controls the relative importance of each channel in the key space such that attention is given to the more discriminative channels**





(a) L2 similarity (b) With shrinkage (c) With both (query 1) (d) With both (query 2)

XMem —— Overview



Memory Consolidation

Other Memories

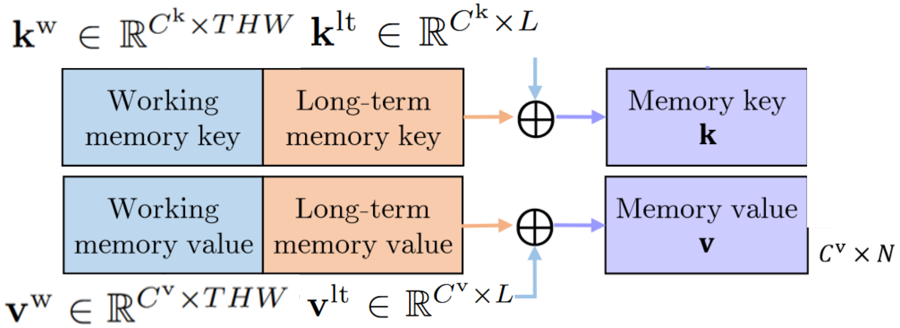
- Working Memory

$\mathbf{k}^{\mathrm{w}} \in \mathbb{R}^{C^{\mathrm{k}} \times THW}$ $\quad \mathbf{v}^{\mathrm{w}} \in \mathbb{R}^{C^{\mathrm{v}} \times THW}$

T is the first frame and the last r-1 frames (r = 5)

- Long-Term Memory
  - Compression 1: $\mathbf{k}^{\mathrm{c}} \subset \mathbf{k}^{\mathrm{w}}$ and $\mathbf{v}^{\mathrm{c}} \subset \mathbf{v}^{\mathrm{w}}$ (T1~Tt-r)
  - Compression 2: $\mathbf{k}^{\mathrm{p}} \subset \mathbf{k}^{\mathrm{c}}$ (**Prototype selection**), $\mathbf{v}^{\mathrm{p}} = \mathbf{v}^{\mathrm{c}}\mathbf{W}(\mathbf{k}^{\mathrm{c}}, \mathbf{k}^{\mathrm{p}})$. (**Memory Potentiation**)
  - **Removing Obsolete Features**
    - Introduce a least-frequently-used (LFU) eviction algorithm
    - Selection is also based on **cumulative affinity** (similar to Prototype selection) after top-k filtering[1]
- Total

Cheng, H.K., Tai, Y.W., Tang, C.K.: Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion. In: CVPR (2021)

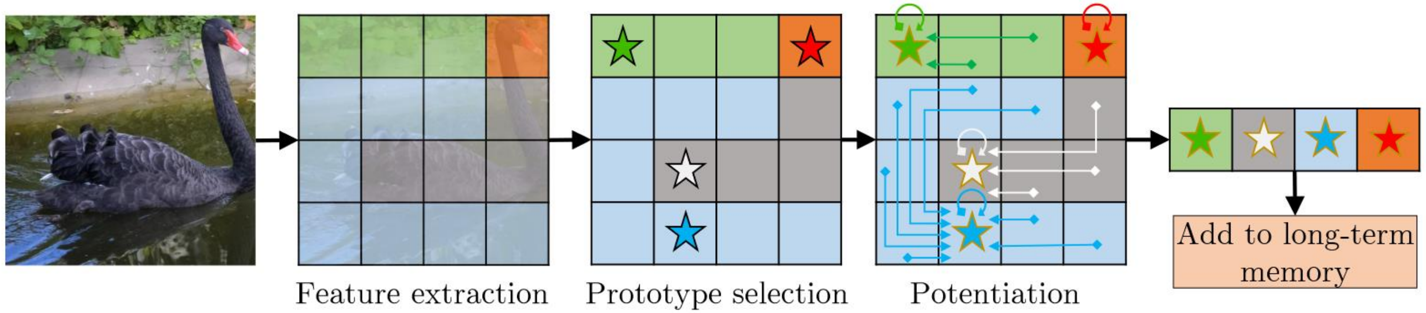


$N = THW + L.$

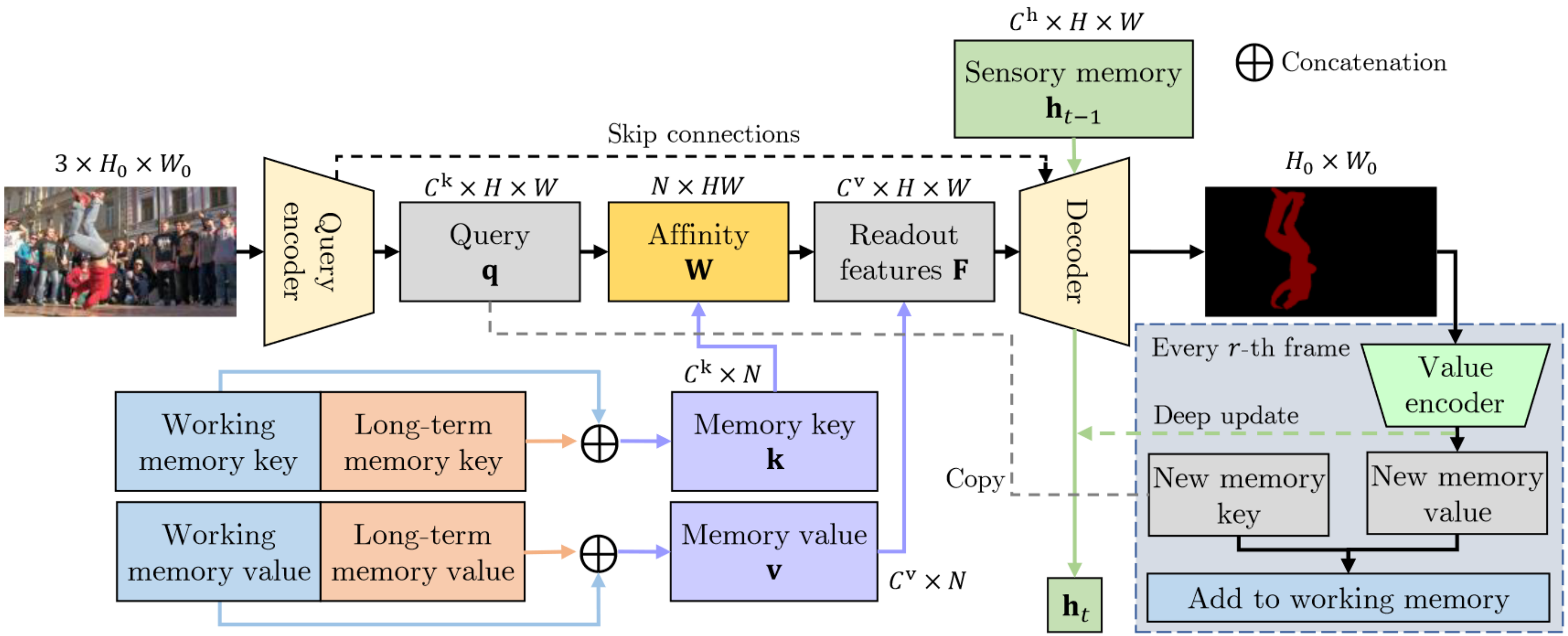
Long-Term Memory

- Prototype Selection
  - Pick Top-**P** frequently **used** candidates as prototypes $\mathbf{k}^{\mathrm{p}} \in \mathbb{R}^{C^{\mathrm{k}} \times P}$
  - **Usage** is defined by its cumulative total affinity in **W** and normalized by the duration that each candidate is in the working memory
- Memory Potentiation
  - Apply channel-wise potentiation to prevent aliasing
  - The enhancement is achieved by aggregating the affinity pixels of the feature map, and the calculation formula of the similarity matrix can be reused

$$\mathbf{v}^{\mathrm{p}} = \mathbf{v}^{\mathrm{c}}\mathbf{W}(\mathbf{k}^{\mathrm{c}}, \mathbf{k}^{\mathrm{p}}). \quad W(k,q) = Softmax(S(k,q)) \quad \mathbf{S}(\mathbf{k}, \mathbf{q})_{ij} = -\mathbf{s}_i \sum_{c}^{C^{\mathrm{k}}} \mathbf{e}_{cj} \left(\mathbf{k}_{ci} - \mathbf{q}_{cj}\right)^2,$$



XMem —— Overview



Experiments Result on Long Videos

Table 1. Quantitative comparisons on the Long-time Video dataset [29].

Method	Long-time Video (1×)			Long-time Video (3×)			$\Delta_{1\times\to3\times}$
	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
CFBI+ [61]	50.9	47.9	53.8	55.3	54.0	56.5	4.4
RMNet [54]	59.8 ±3.9	59.7 ±8.3	60.0 ±7.5	57.0 ±1.6	56.6 ±1.5	57.3 ±1.8	-2.8
JOINT [33]	67.1 ±3.5	64.5 ±4.2	69.6 ±3.9	57.7 ±0.2	55.7 ±0.3	59.7 ±0.2	-9.4
CFBI [59]	53.5	50.9	56.1	58.9	57.7	60.1	5.4
HMMN [44]	81.5 ±1.8	79.9 ±1.2	83.0 ±1.5	73.4 ±3.3	72.6 ±3.1	74.3 ±3.5	-8.1
STM [36]	80.6 ±1.3	79.9 ±0.9	81.3 ±1.0	75.3 ±13.0	74.3 ±13.0	76.3 ±13.1	-5.3
MiVOS* [8]	81.1 ±3.2	80.2 ±2.0	82.0 ±3.1	78.5 ±4.5	78.0 ±3.7	79.0 ±5.4	-2.6
AOT [60]	84.3 ±0.7	83.2 ±3.2	85.4 ±3.3	81.2 ±2.5	79.6 ±3.0	82.8 ±2.1	-3.1
AFB-URR [29]	83.7	82.9	84.5	83.8	82.9	84.6	0.1
STCN [9]	87.3 ±0.7	85.4 ±1.1	89.2 ±1.1	84.6 ±1.9	83.3 ±1.7	85.9 ±2.2	-2.7
XMem (Ours)	**89.8**±0.2	**88.0**±0.2	**91.6**±0.2	**90.0**±0.4	**88.2**±0.3	**91.8**±0.4	0.2

Experiments Result on Short Videos

Table 2. Quantitative comparisons on three commonly used short-term datasets. * denotes BL30K [8] pretraining. Bold and underline denote the best and the second-best respectively in each column. † denotes FPS re-timed on our hardware. On YouTubeVOS, we re-run AOT with all input frames (improving its performance) for a fair comparison.

	YT-VOS 2018 val [57]						DAVIS 2017 val [41]				DAVIS 2016 val [40]			
Method	$\mathcal{G}$	$\mathcal{J}_s$	$\mathcal{F}_s$	$\mathcal{J}_u$	$\mathcal{F}_u$	FPS	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	FPS	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	FPS
STM [36]	79.4	79.7	84.2	72.8	80.9	-	81.8	79.2	84.3	11.1†	89.3	88.7	89.9	14.0†
AFB-URR [29]	79.6	78.8	83.1	74.1	82.6	-	76.9	74.4	79.3	6.8†	-	-	-	-
CFBI [59]	81.4	81.1	85.8	75.3	83.4	3.4	81.9	79.1	84.6	5.9	89.4	88.3	90.5	6.2
RMNet [54]	81.5	82.1	85.7	75.7	82.4	-	83.5	81.0	86.0	4.4†	88.8	88.9	88.7	11.9
HMMN [44]	82.6	82.1	87.0	76.8	84.6	-	84.7	81.9	87.5	9.3†	90.8	89.6	92.0	13.0†
MiVOS* [8]	82.6	81.1	85.6	77.7	86.2	-	84.5	81.7	87.4	11.2	91.0	89.6	92.4	16.9
STCN [9]	83.0	81.9	86.5	77.9	85.7	<u>13.2</u>†	85.4	82.2	88.6	<u>20.2</u>†	91.6	**90.8**	92.5	<u>26.9</u>†
JOINT [33]	83.1	81.5	85.9	78.7	86.5	-	83.5	80.8	86.2	6.8†	-	-	-	-
STCN* [9]	84.3	83.2	87.9	79.0	87.3	<u>13.2</u>†	85.3	82.0	88.6	<u>20.2</u>†	<u>91.7</u>	90.4	<u>93.0</u>	<u>26.9</u>†
AOT [60]	85.5	84.5	<u>89.5</u>	79.6	88.2	6.4	84.9	82.3	87.5	18.0	91.1	90.1	92.1	18.0
XMem (Ours)	<u>85.7</u>	<u>84.6</u>	89.3	<u>80.2</u>	<u>88.7</u>	**22.6**	<u>86.2</u>	<u>82.9</u>	<u>89.5</u>	**22.6**	91.5	90.4	92.7	**29.6**
XMem* (Ours)	**86.1**	**85.1**	**89.8**	**80.3**	**89.2**	**22.6**	**87.7**	**84.0**	**91.4**	**22.6**	**92.0**	<u>90.7</u>	**93.2**	**29.6**

Table 3. Results on DAVIS 2017 test-dev. ‡: uses 600p videos.

	DAVIS 2017 td		
Method	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
STM‡ [36]	72.2	69.3	75.2
RMNet [54]	75.0	71.9	78.1
STCN [9]	76.1	73.1	80.0
CFBI+‡ [61]	78.0	74.4	81.6
HMMN [44]	78.6	74.7	82.5
MiVOS* [8]	78.6	74.9	82.2
AOT [60]	79.6	75.9	83.3
STCN* [9]	79.9	76.3	83.5
XMem (Ours)	81.0	77.4	84.5
XMem* (Ours)	<u>81.2</u>	<u>77.6</u>	<u>84.7</u>
XMem*‡ (Ours)	**82.5**	**79.1**	**85.8**

Ablation Studies

Table 4. Ablation on our memory stores. Standard deviations for $L_{1\times}$ are omitted.

Setting	Y_{18}	D_{17}	$L_{1\times}$	FPS_{D17}	FPS_{Y18}
All memory stores	85.7	86.2	**89.8**	22.6	22.6
No sensory memory	84.4	85.1	87.9	23.1	23.1
No working memory	72.7	77.6	38.7	**31.8**	**28.1**
No long-term memory	**85.9**	**86.3**	n/a	17.6	10.0

Table 5. Ablation on the two scaling terms in memory reading.

Setting	Y_{18}	D_{17}
With both terms	**85.7**	**86.2**
With shrinkage **s** only	85.1	85.6
With selection **e** only	84.8	84.8
With neither	85.0	85.1

Table 6. Comparisons between different memory consolidation methods.

Setting		$L_{3\times}$	Compress ratio
Random	$P = 64$	89.5 ±0.8	**12625%**
K-means centroid	$P = 64$	89.5 ±0.5	**12625%**
Usage-based	$P = 64$	**89.6** ±0.4	**12625%**
Random	$P = 128$	89.7 ±0.7	6328%
K-means centroid	$P = 128$	82.4 ±10.3	6328%
Usage-based	$P = 128$	**90.0** ±0.4	6328%
Random	$P = 256$	89.8 ±0.7	3164%
K-means centroid	$P = 256$	74.5 ±17.0	3164%
Usage-based	$P = 256$	**90.1** ±0.4	3164%
No potentiation		87.9 ±0.2	
With potentiation		**90.0** ±0.4	

Table 7. Comparisons between different strategies for handling long videos.

Setting	$L_{1\times}$	$L_{3\times}$	$\Delta_{1\times\to3\times}$
Consolidation	**89.8** ±0.2	**90.0** ±0.4	0.2
Eager compression	87.8 ±0.3	87.3 ±1.3	-0.5
Sparse insertion	**89.8** ±0.4	87.3 ±1.0	-2.5
Local window	86.2 ±1.5	85.5 ±0.9	-0.7

Table 8. Ablation on the deep update frequency of sensory memory.

Setting	Y_{18}	D_{17}	FPS
Every r-th frame	**85.7**	**86.2**	**22.6**
Every single frame	85.5	86.1	18.5
No deep update	85.3	85.4	**22.6**