

# "Zero-Shot Video Object Segmentation via Attentive Graph Neural Networks"

Proceedings of ICCV 2019  
Seoul, South Korea

Frederick Spreafichi

University of Trieste

June 13, 2025

# Problem Description

**Zero-shot video object segmentation (ZVOS)** refers to the problem of segmenting one or more objects in a video without having access to any pixel-level annotations (masks) of those objects at test time, and without requiring that the model has seen the exact object category during training.



# Problem Description

Formally, let

$$I = \{ I_i \in \mathbb{R}^{H \times W \times 3} \mid i = 1, \dots, N \}$$

be a sequence of  $N$  color frames (height  $H$ , width  $W$ ). The goal is to predict a **Segmentation Mask**

$$S = \{ S_i \in \{0, 1\}^{H \times W} \mid i = 1, \dots, N \}$$

where

$$S_i(x, y) = \begin{cases} 1 & \text{if pixel } (x, y) \text{ belongs to a primary object (foreground)} \\ 0 & \text{if pixel } (x, y) \text{ is background} \end{cases}$$

# Why ZVOS is a challenging task

- **No mask supervision at test time:** the model must infer foreground without any ground-truth annotations
- **Unseen categories:** objects in the video may belong to classes never encountered during training
- **Large appearance variation:** changes in scale, pose, lighting, and background over time
- **Occlusions and fast motion:** partial/full occlusion and motion blur make tracking and segmentation harder
- **Background clutter:** dynamic or textured backgrounds can confuse foreground detection
- **Temporal consistency:** maintaining coherent masks across frames without drift or fragmentation

# Graph Attention Networks (GATs)

## • GATs

- Weights neighbor-node contributions instead of uniform aggregation
- **Assign higher importance to more relevant neighbors** during message passing
- **Dynamically adjust neighbor's influence** for more expressive node representations

## • AGNN Extension

- **Adapts attention to pixel-wise relations** within and between video frames
- **Captures higher-order dependencies** ideal for zero-shot video object segmentation

# AGNN for Zero-Shot Video Object Segmentation

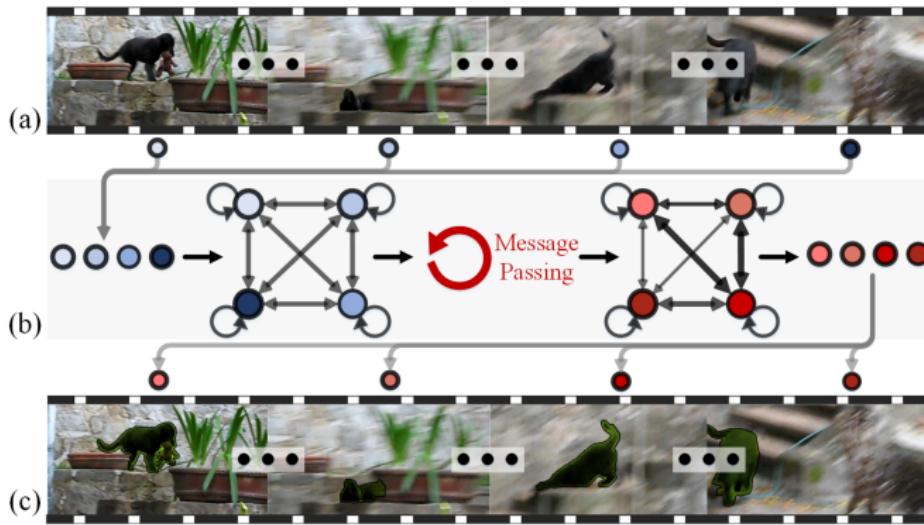


Figure: AGNN for ZVOS

# Network Architecture

- ① FCN-Based Node Embedding
- ② Intra-Attention Based Loop-Edge Embedding
- ③ Inter-Attention Based Line-Edge Embedding
- ④ Gated Message Aggregation
- ⑤ ConvGRU Based Node-State Update
- ⑥ Read-Out Function

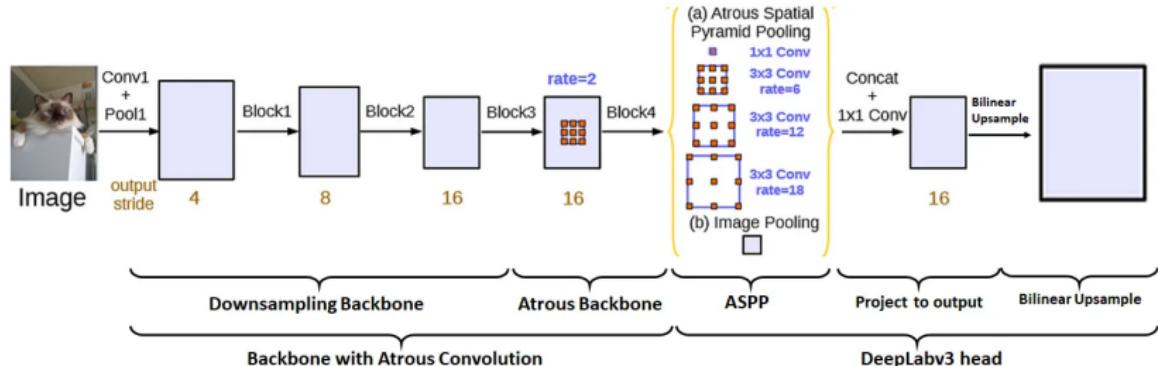
# FCN-Based Node Embedding - DeepLabV3

- Use **DeepLabV3** (a Fully Convolutional Network for semantic segmentation) to extract initial node embeddings:

$$h_i^0 = v_i = F_{\text{DeepLab}}(I_i) \in \mathbb{R}^{H \times W \times C}$$

- **Embeddings preserve spatial structure and high-level semantic features**, essential for pixel-wise segmentation
- **Atrous (dilated) convolutions enlarge the receptive field** without losing resolution
- **Atrous Spatial Pyramid Pooling** (ASPP) applies parallel dilated convolutions at multiple rates to **capture multi-scale context**
- Output: spatially aware and semantically rich feature maps

# DeepLabV3



(a)

# Intra-Attention Based Loop-Edge Embedding

- A **loop-edge**  $e_{i,i}$  **connects node  $v_i$  to itself** in the AGNN graph
- Its **embedding**  $e_{i,i}^k$  is computed by:

$$e_{i,i}^k = \alpha \cdot \text{SoftMax}\left(\left(W_f * h_i^k\right)\left(W_h * h_i^k\right)^T\right)\left(W_\ell * h_i^k\right) + h_i^k$$

- **Intra-attention models long-range dependencies within the same frame**
- **Preserves spatial structure**
- $\alpha$  is a learnable scalar;  $W_f, W_h, W_\ell$  are convolution kernels

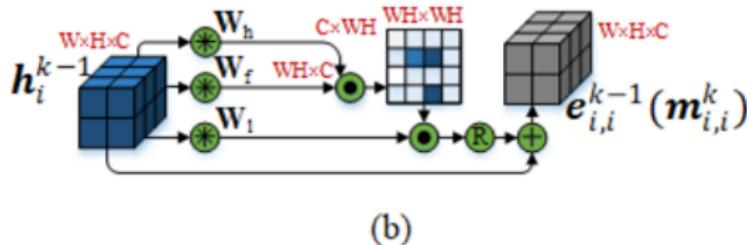
# Inter-Attention Line-Edge Embedding

- A **line-edge**  $e_{i,j}$  **connects node**  $v_i$  **to node**  $v_j$
- Inter-attention computes:

$$\begin{cases} e_{i,j}^k = h_i^k W_c (h_j^k)^T \\ e_{j,i}^k = (e_{i,j}^k)^T \end{cases}$$

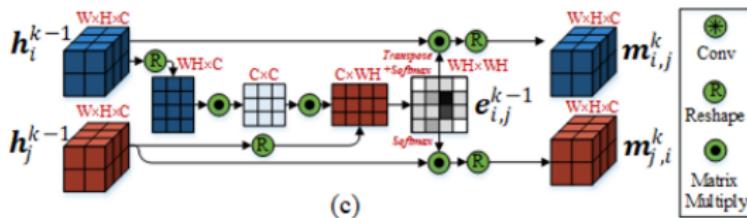
- $W_c \in \mathbb{R}^{C \times C}$  learnable weight matrix;  $h_i^k \in \mathbb{R}^{(W H) \times C}$  flattened feature tensors
- **The embedding  $e_{i,j}^k$  measures how strongly  $v_i$  influences  $v_j$  during message passing**
- Allows discovering semantic matches between frames without recurring to explicit optical flow

# Attention Mechanisms



(b)

Figure: intra-attention



(c)

Figure: inter-attention

# Gated Message Aggregation

- **Self-Loop Message**

$$m_{i,i}^k = e_{i,i}^{k-1}$$

- **Neighbor Message**

$$m_{j,i}^k = M(h_j^{k-1}, e_{j,i}^{k-1}) = \text{SoftMax}(e_{j,i}^{k-1}) h_j^{k-1}$$

- $\text{SoftMax}(e_{j,i}^{k-1})$  normalizes each row of the edge embedding, turning it into weights over spatial positions
- After computing, reshape  $m_{j,i}^k$  back into a  $W \times H \times C$  tensor
- Aggregating messages lets **each node gather information from itself and neighbors**
- Attention ensures **nodes focus only on the most useful parts of a neighbor's features**

# Gated Message Aggregation

- **Learnable Gating** After forming each message  $m_{j,i}^k \in \mathbb{R}^{W \times H \times C}$ , we compute a **confidence score per channel**:

$$g_{j,i}^k = G(m_{j,i}^k) = \sigma(\text{GAP}(W_g * m_{j,i}^k + b_g)) \in [0, 1]^C$$

- GAP( $\cdot$ ) applies global-average pooling over spatial dimensions, **summarizing each channel into one scalar**
- A sigmoid  $\sigma(\cdot)$  then maps these summaries into  $[0, 1]$ , producing **one gating coefficient per channel**
- Channels with higher  $g_{j,i}^k(c)$  are considered more reliable and will pass through

# Gated Message Aggregation

- **Gated Aggregation** Combine all self-loop and neighbor messages, but modulate each by its gate:

$$m_i^k = \sum_{v_j \in \mathcal{V}} g_{j,i}^k \odot m_{j,i}^k$$

where  $\odot$  denotes channel-wise (Hadamard) multiplication:  
 $(g \odot m)(x, y, c) = g(c) m(x, y, c)$

- **Why include gating?**
  - Not all neighbors are equally informative - some may be noisy (occluded, out-of-focus,...)
  - The gate learns to down-weight unreliable channels, **improving robustness**
  - Aggregating only the “trusted” features leads to **cleaner node updates** and better performance on noisy video frames

# ConvGRU-Based Node-State Update

- At iteration  $k$ , given previous state  $h_i^{k-1}$  and aggregated message  $m_i^k$ :

$$h_i^k = U_{\text{ConvGRU}}(h_i^{k-1}, m_i^k) \in \mathbb{R}^{W \times H \times C}$$

- GRU gates decide how much past vs. new information to use**
- Convolutional layers replace dense multiplications, **preserving spatial structure**
- Gates:**
  - Update gate:** Controls incorporation of new message
  - Reset gate:** Controls forgetting of previous state when computing candidate

# ConvGRU Benefits

- **Spatial consistency:** since all operations are convolutional, the new state retains spatial structure ( $H, W$ )
- **Parameter efficiency:** fewer learnable parameters than fully connected GRU
- **Temporal memory:** over multiple message-passing iterations the hidden state carries forward spatial features refined by earlier rounds of attention and gating
- **Channel-wise adaptivity:** the update and reset gates operate per-channel and per-pixel, allowing more flexibility

# Read-Out Function

- After  $K$  message-passing iterations, concatenates the final state  $h_i^K$  and initial state  $h_i^0$  along the channel axis to find the **predicted segmentation mask**:

$$\hat{S}_i = \text{R}_{\text{FCN}}([h_i^K, h_i^0]) \in [0, 1]^{W \times H}$$

- Architecture:**
  - Two  $3 \times 3$  convolutional layers to refine features
  - One  $1 \times 1$  convolutional layer followed by sigmoid to produce a normalized pixel-wise mask
- Key Point:** Implemented as a **lightweight FCN** to preserve **spatial resolution** and **global and local context awareness**

# AGNN Network

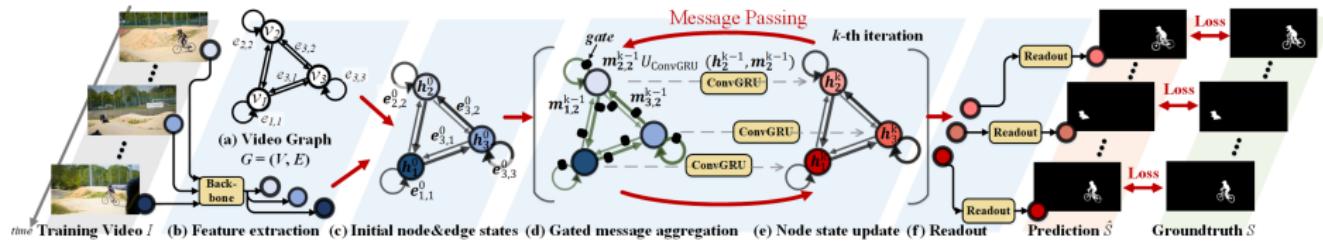


Figure: AGNN during training

# Loss Function

- Pixel-level weighted binary cross-entropy

$$L(S, \hat{S}) = - \sum_{x=1}^{W \times H} \left[ (1 - \eta) S_x \log(\hat{S}_x) + \eta (1 - S_x) \log(1 - \hat{S}_x) \right].$$

- Where:

- $S_x \in \{0, 1\}$ : ground-truth label at pixel  $x$
- $\hat{S}_x \in [0, 1]$ : predicted probability at pixel  $x$
- $\eta$ : ratio of foreground to background pixels in  $S$

- Mitigate class imbalance

- Encourages the model to **learn features that correctly segment objects** rather than defaulting to background
- Weighted loss helps **AGNN learn robust object-level features**

# Environment Details

- **OS:** Ubuntu 22.04 LTS on WSL2
- **CPU:** 12th Gen Intel Core i7-12700KF (10 cores / 20 threads)
- **GPU:** NVIDIA RTX 3060 (12 GB GDDR6, Driver 572.61, cuDNN 8.3)
- **Python / Libraries:** Python 3.10, PyTorch 1.13.1, CUDA 11.6

# Dataset and Metrics

- **Dataset:** DAVIS17 - A benchmark for video object segmentation, featuring high-resolution video sequences with pixel-level annotations for multiple objects across frames
- **Region Similarity  $J$  (IoU):** Average Intersection-over-Union between each predicted mask and ground truth, computed over all frames
- **Boundary Accuracy  $F$ :** Contour measure between predicted and ground truth boundaries, using a 2-pixel tolerance
- **Temporal Stability  $T$ :** Measures how smoothly the predicted mask evolves between consecutive frames. In other words, it is high when most of the object area stays consistent from frame  $t$  to frame  $t + 1$

# Training Setup

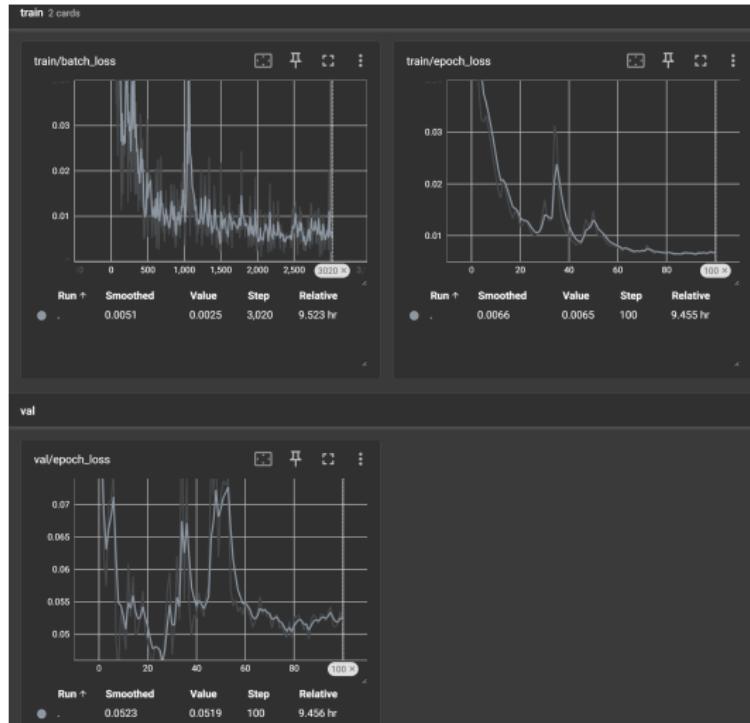
- **Sampling Strategy:**

- Split each training video (length  $N$  frames) into  $N'$  segments
- Randomly select one frame from each segment for training

- **Implementation Details:**

- Due to computational limits, select 2 training videos per batch
- Set  $N' = 3$  (one frame from each of 3 segments)
- Number of message-passing iterations  $K = 3$
- Trained for 100 epochs total

# Training



**Figure:** Training and validation loss

# Training

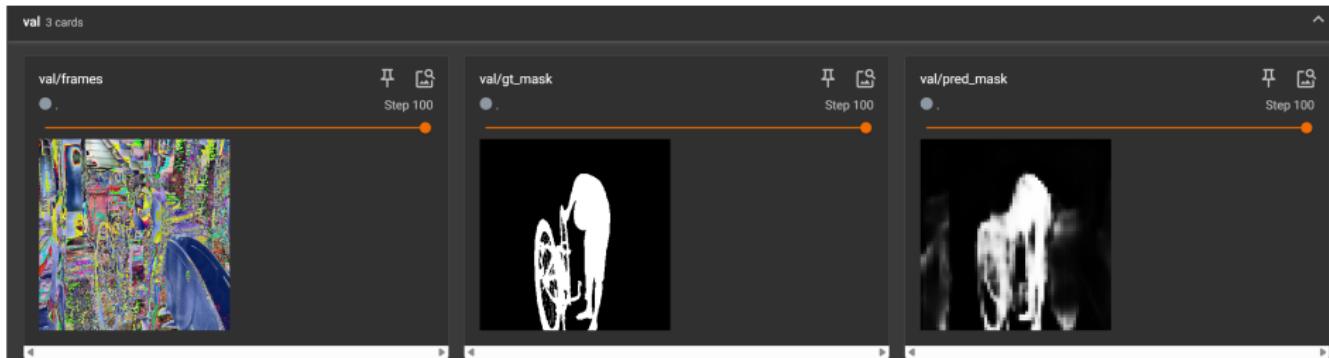


Figure: Validation

# Testing Setup

- Select an input video  $I$  of  $N$  frames at  $473 \times 473$  resolution
- Split  $I$  into  $T = \frac{N}{N'}$  subsets:  $\{I_1, \dots, I_T\}$
- Each subset  $I_t$  contains  $N'$  frames:
$$\{I_t, I_{t+T}, \dots, I_{N-T+t}\}$$
- Feed each subset into AGNN to obtain segmentation masks for its frames
- **Apply Conditional Random Field (CRF) as post-processing to refine boundaries**

# Examples



Figure: Frame from Balckswan video:  $\bar{J} = 0.886$ ;  $\bar{F} = 0.701$ ;  $\bar{T} = 0.906$



Figure: Frame from Breakdance  $\bar{J} = 0.317$ ;  $\bar{F} = 0.190$ ;  $\bar{T} = 0.120$

# Results Summary - DAVIS17

	$\bar{J}$	$\bar{F}$	$\bar{T}$
Overall Mean	0.635	0.406	0.723

Video	$\bar{J}$	$\bar{F}$	$\bar{T}$
blackswan	0.886	0.701	0.906
lab-coat	0.919	0.637	0.884
cows	0.827	0.507	0.945
car-shadow	0.770	0.455	0.875
breakdance	0.317	0.190	0.120
india	0.179	0.200	0.338

# Interpreting the Results - DAVIS17

- **High-Scoring Videos** (e.g., blackswan, lab-coat, cows):
  - Large, high-contrast foreground occupies most of the frame
  - Minimal occlusion and smooth, predictable motion
- **Low-Scoring Videos** (e.g., breakdance, india):
  - Rapid, erratic motion and cluttered backgrounds (breakdance) or very small, low-contrast objects (india)
  - Fixed receptive field struggles to extract stable foreground signals under these conditions
- **Boundary vs. Region Scores:**
  - High IoU (region) but mask contours often lie several pixels away from ground-truth edges, lowering the boundary F-measure

# Results Summary – YouTube-Objects

$\bar{J}$
Overall Mean <b>0.484</b>

Class	$\bar{J}$
aeroplane	0.315
car	0.601
cow	0.489
dog	0.495
motorbike	0.529
train	0.427



# Peper's Results: DAVIS17 & YouTube-Objects

Table: DAVIS17 Test-Dev

Method	$\bar{J}$	$\bar{F}$	$\overline{J \& F}$
RVOS [63]	39.0	42.8	43.7
<b>AGNN</b>	<b>58.9</b>	<b>65.7</b>	<b>61.1</b>

Table: YouTube-Objects Per-Category (mean  $F$  scores)

Method	Airplane (6)	Bird (6)	Boat (15)	Car (7)	Cat (16)	Cow (20)	Dog (27)	Horse (14)	Motorbike (10)	Train (5)	Avg.
FST [43]	70.9	70.6	42.5	65.2	52.1	44.5	65.3	53.5	44.2	29.6	53.8
COSEG [60]	69.3	76.0	53.5	70.4	66.8	49.0	47.5	55.7	39.5	53.4	58.1
ARP [27]	73.6	56.1	57.8	33.9	30.5	41.8	36.8	44.3	48.9	39.2	46.2
LVO [58]	86.2	81.0	68.5	69.3	58.8	68.5	61.7	53.9	60.8	66.3	67.5
PDB [55]	78.0	80.0	58.9	76.5	63.0	64.1	70.1	67.6	58.3	35.2	65.4
FSEG [21]	81.7	63.8	72.3	74.9	68.4	68.0	69.4	60.4	62.7	62.2	68.4
SFL [7]	65.6	65.4	59.9	64.0	58.9	51.1	54.1	64.8	52.6	34.0	57.0
AGS [69]	87.7	76.7	72.2	78.6	69.2	64.6	73.3	64.4	62.1	48.2	69.7
<b>AGNN</b>	81.1	75.9	70.7	78.1	67.9	69.7	77.4	67.3	68.3	47.8	<b>70.8</b>

# Conclusions

- **Fully-connected video graph:** frames as nodes with learnable attention edges - efficiently captures inter-frame correlations without optical flow
- **Recursive message passing (ConvGRU):** iteratively propagates information to mine higher-order frame relations, retaining only the most relevant information
- **Spatial-preserving readout:** maintains per-pixel spatial structure for accurate segmentation
- **DAVIS17 Test-Dev:**  $\overline{J\&F} = 61.1$  vs. 43.7 for RVOS (new state-of-the-art result)
- **YouTube-Objects:** average  $F=70.8$  (best published zero-shot result)

## References

- Wenguan Wang, Xiankai Lu, Jianbing Shen, David Crandall, Ling Shao. *Zero-Shot Video Object Segmentation via Attentive Graph Neural Networks*. arXiv:2001.06807 (2020).
- Inception Institute of Artificial Intelligence (UAE) and Indiana University (USA).