Segmenting objects from relational visual data

Segmenting Objects from Relational Visual Data

Xiankai Lu, *Member IEEE*, Wenguan Wang, *Member IEEE*
Jianbing Shen, *Senior Member IEEE*, David J. Crandall, *Member IEEE*, Luc Van Gool, *Member IEEE*

Abstract—In this article, we model a set of pixelwise object segmentation tasks — automatic video segmentation (AVS), image co-segmentation (ICS) and few-shot semantic segmentation (FSS) — in a unified view of segmenting objects from relational visual data. To this end, we propose an attentive graph neural network (AGNN) that addresses these tasks in a holistic fashion, by formulating them as a process of iterative information fusion over data graphs. It builds a fully-connected graph to efficiently represent visual data as nodes and relations between data instances as edges. The underlying relations are described by a differentiable attention mechanism, which thoroughly examines fine-grained semantic similarities between all the possible location pairs in two data instances. Through parametric message passing, AGNN is able to capture knowledge from the relational visual data, enabling more accurate object discovery and segmentation. Experiments show that AGNN can automatically highlight primary foreground objects from video sequences (*i.e.*, automatic video segmentation), and extract common objects from noisy collections of semantically related images (*i.e.*, image co-segmentation). AGNN can even generalize segment new categories with little annotated data (*i.e.*, few-shot semantic segmentation). Taken together, our results demonstrate that AGNN provides a powerful tool that is applicable to a wide range of pixel-wise object pattern understanding tasks with relational visual data. Our algorithm implementations have been made publicly available at https://github.com/carrierlxk/AGNN.

Index Terms—Graph Neural Network, Automatic Video Segmentation, Image Co-Segmentation, Few-shot Semantic Segmentation.

1 Introduction

The visual world is highly structured. Entities that are semantically related have similar visual appearance: both trucks and buses have wheels and cabins, for example. Entities also undergo continuous variations over time, and there is inherent visual correspondence between observations adjacent in a video clip. This structure explains in part how humans can learn new concepts rapidly from only a few examples. The rich structure between entities thus not only governs how our recorded visual data are arranged, but also helps us efficiently understand visual scenes.

In this article, we study the problem of how to model and leverage relationships between visual data (semantically related images and correlated video frames) to better identify and extract visual objects. This benefits many computer vision tasks, such as Automatic Video Segmentation (AVS, automatically segmenting primary foreground objects from video sequences; Fig. 1a), Image Co-Segmentation (ICS, extracting common objects from a collection of semantically related images; Fig. 1b), and Few-shot Semantic Segmentation [2] (FSS, learning to perform segmentation from only a few annotated examples; Fig. 1c). These tasks are essential building blocks for many real-world applications: ICS is useful in handling noisy Web photo collections, large-scale data annotation, and multi-camera visual signals, AVS is a core technique in video processing and understanding, and FSS is valuable when supervised examples are hard to acquire due to privacy, safety, economic, or ethical issues [3].

Fig. 1. Our AGNN provides a powerful framework that formulates (a) Automatic Video Segmentation (AVS), (b) Image Co-Segmentation (ICS), and (c) Few-shot Semantic Segmentation (FSS) from a unified view of segmenting objects from relational visual data.

Deep learning with neural networks has become the dominant solution for the above problems. Although modern neural networks have greatly advanced the development of their specific fields, modern network architectures often suffer from certain limitations, such as a limited ability to explicitly model the rich relations among visual data. For example, existing AVS methods are generally [4]–[7] built upon two-stream or recurrent networks, and thus focus primarily on local cues between successive frames, ignoring (or only weakly modeling) important correlations among distant frames. But successfully handling occlusions, scale variations, and appearance changes (Fig. 2(a)) resort to a more complete understanding of the video content from a global, instead of local view. For ICS, current approaches [8],

- *X. Lu is with School of Software, Shangdong University, China. (Email: carrierlxk@gmail.com)*
- *W. Wang and L. Van Gool are with ETH Zurich, Switzerland. (Email: wenguan.wang@gmail.com, vangool@vision.ee.ethz.ch)*
- *J. Shen is with Inception Institute of Artificial Intelligence, UAE. (Email: shenjianbingcg@gmail.com)*
- *D. Crandall is with the Luddy School of Informatics, Computing, and Engineering, Indiana University. (email: djcran@indiana.edu)*
- *A preliminary version of this work has appeared in ICCV 2019 [1].*
- *Corresponding author: Wenguan Wang*

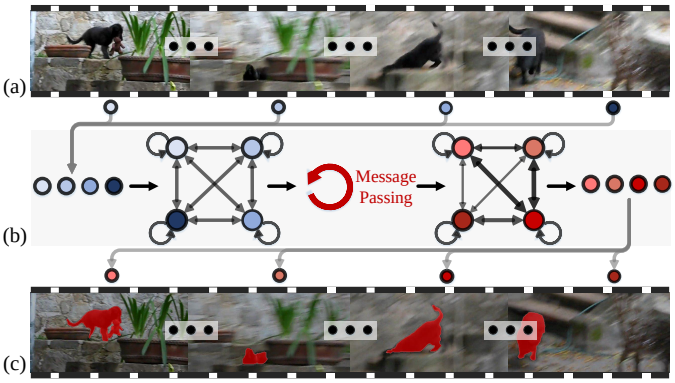

Fig. 2. Illustration of AGNN-based AVS solution. (a) Input video sequence, typically with object occlusion and scale variation. (b) Our AGNN represents the input video as a graph, where nodes (blue circles) are video frames, and edges (black arrows) are relations between corresponding frame pairs, captured by a neural attention mechanism. After several differentiable message passing iterations over the video graph, higher-order relations can be incorporated and more optimal foreground estimates are obtained. (c) Final segmentation results.

[9] employ Siamese networks to capture the correspondence between pairs of related images, which means processing a whole image collection can be quite complicated (as all the possible image pairs in the image collection should be considered, at least theoretically). As for FSS, popular solutions [10], [11] largely formulate the task from a metric learning perspective. In essence, they learn a contextual similarity measure, according to which they propagate the label information from support examples to target images. However, they regard each support image independently and thus fail to exploit knowledge from the correlations between support images.

In stark contrast to these methods which focus on specific fields, we seek to present a unified view for these three tasks — ICS, AVS, and FSS — from the perspective of segmenting objects from relational visual data. This approach has several advantages, including providing insight into the underlying mechanisms common between these tasks, allowing some of their inherent challenges to be better addressed. Specifically, we propose an Attentive Graph Neural Network (AGNN) that efficiently generalizes ICS, AVS, and FSS as an end-to-end, message passing-based graph information fusion procedure. AGNN provides a clean yet powerful framework that handles the limitations of current methods head-on, by comprehensively capturing context in relational visual data. In our approach, a fully-connected graph is constructed in which visual data are represented as nodes and relations between data instances are captured as edges (Fig. 2(b)). A differentiable neural attention mechanism is introduced to model the relations by considering the dependencies in all the possible pairs of positions (regions) in two nodes. By using recursive message passing to iteratively propagate information over the graph, with each node progressively updating its states by assimilating the information from other nodes, AGNN can leverage relational cues to mine object patterns in a step-by-step and global manner. In addition, by implementing the key operations in the iterative algorithm as Fully Convolutional Networks (FCNs), AGNN preserves spatial information, which makes it applicable to spatial prediction problems and significantly distinguishes it from MultiLayer Perceptron (MLP)-based Graph Neural Networks (GNNs).

Due to its recursive nature, AGNN is flexible enough to process variable numbers of nodes during inference, which is essential for AVS and ICS. In addition, AGNN provides serval unique advantages for AVS. First, since AGNN operates on multiple frames, it naturally leads to training data augmentation, as the combination of candidates is numerous. In addition, AGNN offers a powerful tool for modeling flexible long-term relations between video frames, thus yielding a more complete representation of video content and omitting time-consuming optical flow computations used in many prior AVS methods.

Experimental results on AVS, ICS, and FSS tasks consistently demonstrate the promising performance of our AGNN. The experiments also indicate that AGNN is able to not only capture correlations among similar video frames or semantically related static images, but also efficiently learn unseen semantics from only a few examples.

This paper builds upon our conference paper [1] and significantly extends it in several ways. First, we extend our model, AGNN, as a general framework that formulates diverse segmentation tasks, including AVS, ICS, and FSS. Second, we propose to address these tasks in a unified perspective of extracting objects from relational data, and more precisely state our motivations and contributions. Third, we provide more details regarding formulation and implementation of our AGNN model. Finally, more experiments are conducted on several representative datasets to demonstrate the effectiveness of our model.

2 Related Work

In §2.1, we first provide a brief overview of GNNs. Then, we review representative literature the fields of in AVS (§2.2), ICS (§2.3) and FSS (§2.4).

2.1 Graph Neural Networks (GNNs)

GNNs[12] are powerful tools for learning graph representations in an end-to-end manner, and can be divided into two broad classes: *graph convolutional networks* and *message passing graph networks*. The former [13]–[15]generalizes the convolution operation over non-Euclidean data. Their simple architecture makes them popular, but also limits their modeling capability for complex structures [16]. The latter [17], [18] parameterize all the nodes, edges, and information fusion steps in graph learning, leading to more complicated yet flexible architectures. GNNs have obtained wide success in many computer vision tasks, including human behavior understanding [19], [20], scene graph generation [21], human semantic parsing [22], active perception [23], [24], demonstrating their advantages in structured modeling.

Our AGNN falls into the latter class and enjoys several appealing characteristics. First, AGNN is unique in its ability to preserve spatial information, in contrast to conventional MLP-based GNNs, which is crucial for per-pixel object pattern understanding. Second, to efficiently capture relational information, AGNN exploits a neural differentiable attention mechanism, which accounts for detailed

correspondences between all the region pairs in two data instances, and thus produces discriminative edge features. Third, as far as we know, there is no prior attempt to formulate AVS, IOS, and FSS in a unified GNN framework.

2.2 Automatic Video Segmentation (AVS)

AVS is a long-studied computer vision problem which attempts to segment video foreground objects without test-time human interaction. Please refer to a recent survey [25] for more detailed literature review. Conventional methods typically use hand-crafted features (*e.g.*, color, optical flow, trajectory) [26]–[29] and certain heuristic assumptions related to the foreground (*e.g.*, local motion differences [27], background priors [30]). Some others explore more efficient foreground object representations, such as point trajectories [31]–[33] or object proposals [34]–[36]. They are typically non-learning methods working in a purely *unsupervised* fashion. Due to the limited representation ability of hand-crafted features, they often fail due to challenging factors such as fast motion, large appearance variation, and occlusion, and can fail in scenarios in which their heuristic assumptions do not work. Recent deep learning-based approaches learn more powerful object features from massive training data, yielding a *zero-shot* solution [37]–[39] (*i.e.*, no human interaction during testing), typically through recurrent neural networks [7], [40], or two stream architectures [4], [6], [41], [42], to address appearance or motion cues in a local and sequential manner. Some of the latest ones address learning object patterns from unlabeled videos [43] or object hotspot tracking [44]. Some recent efforts were made to tackle AVS in the instance level. Instance-level AVS needs to not only separate the primary video objects from the background, but also discriminate different object instances. Ventura *et al*. [37] propose a recurrent network based model which consists of a spatial LSTM for discovering each instance and a temporal LSTM for associating instances across different frames. In [45], object proposals are associated according to both local and global cues. More recently, Dave *et al*. [46] address object instance discrimination and segmentation through bottom-up, motion-aware objectness information.

In comparison, AGNN provides a unified, end-to-end trainable, graph model-based AVS solution. This modeling strategy distinctively differentiates it from current popular recurrent network-based methods: through iteratively propagating messages over the graph, AGNN can capture long-term cross-frame correlations. This provides an insightful glimpse into the problem that addresses the value of global context in videos, in contrast to current algorithms primarily considering sequential information within short-term temporal segments. Moreover, AGNN utilizes a differentiable attention mechanism to capture cross-frame correlations, avoiding time-consuming optical flow computations and learning more foreground-related cues.

2.3 Image Co-Segmentation (ICS)

ICS [47]–[50] aims to jointly segment common objects in a given noisy set of related images. Traditional methods usually formulate ICS as minimizing an energy function defined over the whole or a part of the image set and consider intra- and inter-image cues [51]–[54]. So far, only a few solutions have been proposed to specifically address ICS [8], [9], [55]–[57] through deep learning techniques, mainly due to the lack of a proper, end-to-end modeling strategy for the problem. They mainly address ICS through a pairwise comparison protocol and employ a Siamese network to capture the similarity between two related images [8], [9], cannot directly operate on multiple images, and require sophisticated inference.

Our AGNN-based ICS solution offers significant advantages. First, previous methods consider ICS as a pairwise image matching problem, while we formulate ICS as an information propagation and fusion process among multiple images. This means our model leverages more information in the image collection. Second, the Siamese network-based systems only handle pairwise relations, while our message passing-based iterative inference can learn higher-order relations among multiple images. Third, our method is based on the graph model, yielding a general and elegant framework for ICS modeling and allowing us to process variable numbers of nodes.

2.4 Few-Shot Segmentation (FSS)

FSS aims to learn to perform segmentation from only a few annotated images (*support* set) over new images (*query* set) from the same classes [2]. Earlier methods are parameter optimization-based [2], [58], [59], sharing the spirit of conditioning segmentation network parameter modulation on the knowledge from the support. Recent approaches [11], [60] are metric learning based, *i.e.*, performing segmentation through pixel-level matching between the support and query images within a learnable semantic embedding space. Thus they primarily focus on how to learn a good embedding space that can generalize well on unseen classes; though avoiding sensitive and expensive network parameter optimization, they ignore the relations among support samples. In this article, rather than previous FSS methods regarding each support image independently, AGNN arranges the support set as graph-structured, over which it performs information diffusion to better mine the context in the support set so as to facilitate the query prediction. Although Garcia *et al*. [61] also explored context in the support set, they focused on few-shot classification and their model is built upon conventional MLP-based GNNs.

3 Methods

In §3.1, we first give a brief introduction to generic formulations of message passing based GNN models. Then, we elaborate on our proposed AGNN framework (§3.2). Then, we detail how to apply our AGNN model to AVS, ICS, and FSS tasks in §3.3, §3.4, and §3.5, respectively. Finally, in §3.6, we provide more implementation details for above tasks.

3.1 General Formulations of GNNs

To make this article self-contained, we first briefly introduce the basic ideas and notions of GNNs. GNNs are powerful models for collectively aggregating information from data represented in graph domains [12], [17]. Specifically, a GNN

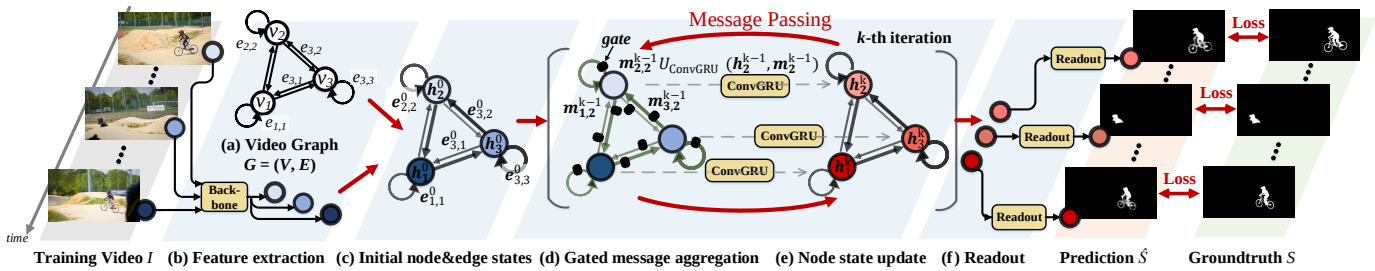

Fig. 3. Our AGNN-based AVS solution during the training phase (§3.2). (a) We represent the input video $\mathcal{I}$ as a fully connected graph $\mathcal{G}$. (b) Initial frame features are extracted from the backbone network. (c) According to (a) and (b), the node and edge states are initialized through Eqs. 3, 4, 5, respectively. (d,e) AGNN recursively performs gated message aggregation (Eq. 9) and node/edge state updating (Eq. 10) over $\mathcal{G}$. (f) After several message passing iterations, a readout function (Eq. 11) is used to obtain the node predictions. Zoom in for details.

model is defined according to a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Each node $v_i \in \mathcal{V}$ takes a unique value from $\{1, \dots, |\mathcal{V}|\}$, and is associated with an initial *node representation* (or *node state* or *node embedding*) $\boldsymbol{v}_i$. Each edge $e_{i,j} \in \mathcal{E}$ is a pair $e_{i,j} = (v_i, v_j) \in |\mathcal{V}| \times |\mathcal{V}|$, with an *edge representation* $\boldsymbol{e}_{i,j}$. For each node v_i, we learn an updated node representation $\boldsymbol{h}_i$ by aggregating representations of its neighbors. Here $\boldsymbol{h}_i$ is used to produce an output $\boldsymbol{o}_i$, *i.e.*, a node label. More specifically, GNNs map graph $\mathcal{G}$ to the node outputs $\{\boldsymbol{o}_i\}_{i=1}^{|\mathcal{V}|}$ through two phases. First, a parametric *message passing phase* runs for K steps, which recursively propagates messages and updates node representations. At iteration k, we update the state of each node v_i according to its received message $\boldsymbol{m}_i^k$ (*i.e.*, summarized information from its neighbors $\mathcal{N}_i$) and its previous state $\boldsymbol{h}_i^{k-1}$:

$$\begin{aligned} \text{message aggregation: } \boldsymbol{m}_i^k &= \sum\nolimits_{v_j \in \mathcal{N}_i} \boldsymbol{m}_{j,i}^k \\ &= \sum\nolimits_{v_j \in \mathcal{N}_i} M(\boldsymbol{h}_j^{k-1}, \boldsymbol{e}_{i,j}^{k-1}), \\ \text{node representation update: } \boldsymbol{h}_i^k &= U(\boldsymbol{h}_i^{k-1}, \boldsymbol{m}_i^k), \end{aligned} \quad (1)$$

where $\boldsymbol{h}_i^0 = \boldsymbol{v}_i$, $M(\cdot)$ and $U(\cdot)$ are the *message function* and *state update function*, respectively. After k iterations of aggregation, $\boldsymbol{h}_i^k$ captures the relations within the k-hop neighborhood of node v_i.

Second, a *readout phase* maps the node representation $\boldsymbol{h}_i^K$ of the final K-iteration to the node output $\boldsymbol{o}_i$, through a *readout function* $R(\cdot)$:

$$\text{readout:} \quad \boldsymbol{o}_i = R(\boldsymbol{h}_i^K). \quad (2)$$

The message function $M(\cdot)$, update function $U(\cdot)$, and readout function $R(\cdot)$ are all learned differentiable functions.

Our AGNN essentially extends traditional fully connected GNNs to (1) preserve spatial features; (2) capture pairwise relations (edges) via a differentiable attention mechanism; and (3) address AVS, ICS, and FSS in a unified framework. Next, we use AVS as an exemplar task to detail our AGNN framework. In §3.6, we specify how to extend AGNN to ICS and FSS.

3.2 Attentive Graph Neural Network

We take object-level AVS as an exemplar task to introduce the main ideas and core components of our AGNN model.

Problem Definition and Notations. For object-level AVS, given an input video sequence $\mathcal{I} = \{I_i \in \mathbb{R}^{w \times h \times 3}\}_{i=1}^N$ with N frames in total, the goal is to generate a corresponding sequence of binary segment masks: $\mathcal{S} = \{S_i \in \{0,1\}^{w \times h}\}_{i=1}^N$. To achieve this, AGNN represents $\mathcal{I}$ as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where node $v_i \in \mathcal{V}$ represents the i-th frame I_i, and edge $e_{i,j} = (v_i, v_j) \in \mathcal{E}$ indicates the relation from I_i to I_j. To comprehensively capture the underlying relationships between video frames, we assume $\mathcal{G}$ is fully connected and includes self-connections at each node (see Fig. 3(a)). For clarity, we refer to $e_{i,i}$, which connects a node v_i to itself, as a *loop-edge*; and $e_{i,j}$, which connects two different nodes v_i and v_j, as a *line-edge*.

The core idea of our AGNN is to perform K message propagation iterations over $\mathcal{G}$ to mine the rich structures between nodes, thus further comprehensively capturing the context in $\mathcal{I}$. This helps to better estimate the foreground from a global view. In addition, because of the nature of pixel prediction for such segmentation task, we maintain the spatial dependency on each node and mine the underlying relationship across different nodes (*i.e.*, frames), which is achieved by basing all the components of AGNN on convolution operations. The segmentation predictions $\hat{\mathcal{S}}$ are read from the final node states $\{\boldsymbol{h}_i^K\}_{i=1}^N$. Next, we describe each component of our model in detail.

FCN-Based Node Embedding. We use DeepLabV3 [62], a classical FCN based semantic segmentation architecture, to extract frame features as node representations (see Fig. 3(b) and Fig. 4(a)). Node v_i's initial embedding $\boldsymbol{h}_i^0$ can be computed as:

$$\boldsymbol{h}_i^0 = \boldsymbol{v}_i = F_{\text{DeepLab}}(I_i) \in \mathbb{R}^{W \times H \times C}, \quad (3)$$

where $\boldsymbol{h}_i^0$ is a 3D tensor feature with $W \times H$ spatial resolution and C channels, which preserves spatial information as well as high-level semantic information.

Intra-Attention Based Loop-Edge Embedding. A loop-edge $e_{i,i} \in \mathcal{E}$ is a special edge that connects a node to itself. The loop-edge embedding $\boldsymbol{e}_{i,i}^k$ is used to capture the intra relations within node representation $\boldsymbol{h}_i^k$ (*i.e.*, internal frame representation). We formulate $\boldsymbol{e}_{i,i}^k$ as an *intra-attention* mechanism [63], [64], which has been proven complementary to convolutions and helpful for modeling long-range, multi-level dependencies across image regions [65]. In particular, the intra-attention calculates the response at a position by attending to all the positions within the same node embedding (see Fig. 3(c) and Fig. 4(b)):

$$\begin{aligned} \boldsymbol{e}_{i,i}^k &= F_{\text{intra-att}}(\boldsymbol{h}_i^k) \in \mathbb{R}^{W \times H \times C} \\ &= \alpha \, \text{softmax}\big((\boldsymbol{W}_f * \boldsymbol{h}_i^k)(\boldsymbol{W}_h * \boldsymbol{h}_i^k)^\top\big)(\boldsymbol{W}_l * \boldsymbol{h}_i^k) + \boldsymbol{h}_i^k, \end{aligned} \quad (4)$$

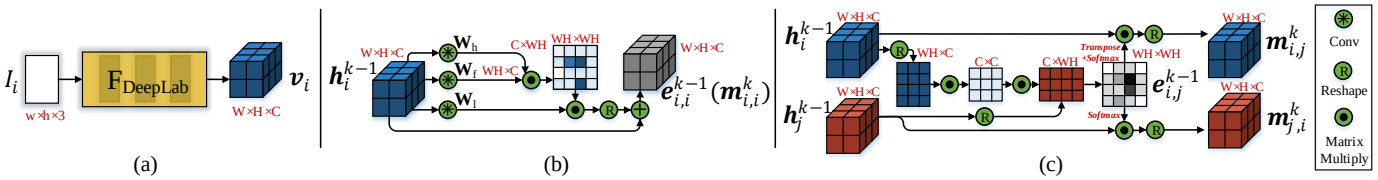

Fig. 4. Detailed illustration of our (a) node embedding (Eq. 3), (b) intra-attention based loop-edge embedding and corresponding loop-message generation (Eq. 4), (c) inter-attention based straight-edge embedding and corresponding neighbor message generation (Eq. 5).

where '$*$' represents the convolution operation, $\boldsymbol{W}$s indicate learnable convolution kernels, and α is a learnable scale parameter. Eq. 4 makes the output element of each position in $\boldsymbol{e}_{i,i}^k$ encode the contextual information as well as original information, thus enhancing the representation ability.

Inter-Attention Based Line-Edge Embedding. A line-edge $e_{i,j} \in \mathcal{E}$ connects two different nodes v_i and v_j. The line-edge embedding $\boldsymbol{e}_{i,j}^k$ is used to mine the relation from node v_i to v_j, in the node embedding space (see Fig. 3(b)). Here we compute an *inter-attention* mechanism [66] to capture the bi-directional relations between two nodes v_i and v_j (see Fig. 3(c) and Fig. 4(c)):

$$\begin{aligned}\boldsymbol{e}_{i,j}^k &= F_{\text{inter-att}}(\boldsymbol{h}_i^k, \boldsymbol{h}_j^k) = \boldsymbol{h}_i^k \boldsymbol{W}_c \, \boldsymbol{h}_j^{k\top} \in \mathbb{R}^{(WH)\times(WH)}, \\ \boldsymbol{e}_{j,i}^k &= F_{\text{inter-att}}(\boldsymbol{h}_j^k, \boldsymbol{h}_i^k) = \boldsymbol{h}_j^k \boldsymbol{W}_c^\top \boldsymbol{h}_i^{k\top} \in \mathbb{R}^{(WH)\times(WH)},\end{aligned} \tag{5}$$

where $\boldsymbol{e}_{i,j}^k = \boldsymbol{e}_{j,i}^{k\top}$. Here $\boldsymbol{e}_{i,j}^k$ is the outgoing edge feature and $\boldsymbol{e}_{j,i}^k$ the incoming one, for node v_i. $\boldsymbol{W}_c \in \mathbb{R}^{C\times C}$ indicates a learnable weight matrix. $\boldsymbol{h}_j^k \in \mathbb{R}^{(WH)\times C}$ and $\boldsymbol{h}_i^k \in \mathbb{R}^{(WH)\times C}$ are flattened into matrix representations. Each element in $\boldsymbol{e}_{i,j}^k$ reflects the similarity between each row of $\boldsymbol{h}_i^k$ and each column of $\boldsymbol{h}_j^{k\top}$. As a result, $\boldsymbol{e}_{i,j}^k$ can be viewed as an *importance* map of node v_i's embedding to v_j at all the positions, and vice versa. By attending to each node pair, $\boldsymbol{e}_{i,j}^k$ explores their joint representations in the node embedding space.

Gated Message Aggregation. In our AGNN, for the message passed in the self-loop, we view the loop-edge embedding $\boldsymbol{e}_{i,j}^{k-1}$ itself as a message (Fig. 4(b)), since it already contains the contextual and original node information (Eq. 4):

$$\boldsymbol{m}_{i,i}^k = \boldsymbol{e}_{i,i}^{k-1} \in \mathbb{R}^{W\times H\times C}. \tag{6}$$

For the message $\boldsymbol{m}_{j,i}$ passed from v_j to v_i (Fig. 4(c)), we have:

$$\boldsymbol{m}_{j,i}^k = M(\boldsymbol{h}_j^{k-1}, \boldsymbol{e}_{i,j}^{k-1}) = \text{softmax}(\boldsymbol{e}_{i,j}^{k-1}) \boldsymbol{h}_j^{k-1} \in \mathbb{R}^{(WH)\times C}, \tag{7}$$

where softmax$(\cdot)$ normalizes each row of the input. Thus, each row (position) of $\boldsymbol{m}_{j,i}^k$ is a weighted combination of each row (position) of $\boldsymbol{h}_j^{k-1}$, where the weights come from the corresponding column of $\boldsymbol{e}_{i,j}^{k-1}$. In this way, the message function $M(\cdot)$ assigns its edge-weighted feature (*i.e.*, message) to the neighbor nodes [18]. Then, $\boldsymbol{m}_{j,i}^k$ is reshaped back to a 3D tensor with a size of $W \times H \times C$.

In addition, because some nodes are noisy due to camera shift or out-of-view, their messages may be useless or even harmful. We apply a learnable gate $G(\cdot)$ to measure the confidence of a message $\boldsymbol{m}_{j,i}$ ($\boldsymbol{m}_{i,i}$):

$$\begin{aligned}\boldsymbol{g}_{j,i}^k &= G(\boldsymbol{m}_{j,i}^k) = \sigma\big(F_{\text{GAP}}(\boldsymbol{W}_g * \boldsymbol{m}_{j,i}^k + b_g)\big) \in [0,1]^C, \\ \boldsymbol{g}_{i,i}^k &= G(\boldsymbol{m}_{i,i}^k) = \sigma\big(F_{\text{GAP}}(\boldsymbol{W}_g * \boldsymbol{m}_{i,i}^k + b_g)\big) \in [0,1]^C,\end{aligned} \tag{8}$$

where $F_{\text{GAP}}(\cdot)$ indicates the use of global average pooling to generate channel-wise responses, σ is the logistic sigmoid function, and $\boldsymbol{W}_g$ and b_g are the trainable convolution kernel and bias, respectively.

Following Eq. 1, we collect the messages from the neighbors and self-loop via gated summarization (see Fig. 3(d)):

$$\boldsymbol{m}_i^k = \sum\nolimits_{v_j \in \mathcal{V}} \boldsymbol{g}_{j,i}^k \star \boldsymbol{m}_{j,i}^k \in \mathbb{R}^{W\times H\times C}, \tag{9}$$

where '$\star$' denotes the channel-wise Hadamard product. Here, the gate mechanism is used to filter out irrelevant information from noisy frames.

ConvGRU based Node-State Update. In step k, after aggregating all the information from the neighbor nodes and itself (Eq. 9), v_i gets a new state $\boldsymbol{h}_i^k$ by taking into account its prior state $\boldsymbol{h}_i^{k-1}$ and its received message $\boldsymbol{m}_i^k$. To preserve the spatial information conveyed in $\boldsymbol{h}_i^{k-1}$ and $\boldsymbol{m}_i^k$, we leverage ConvGRU [67] to update the node state (Fig. 3(e)):

$$\boldsymbol{h}_i^k = U_{\text{ConvGRU}}(\boldsymbol{h}_i^{k-1}, \boldsymbol{m}_i^k) \in \mathbb{R}^{W\times H\times C}. \tag{10}$$

ConvGRU is proposed as a convolutional counterpart to previous MLP based GRU [68], and introduces convolution operation into input-to-state and state-to-state transitions.

Readout Function. After K message passing iterations, we obtain the final state $\boldsymbol{h}_i^K$ for each node v_i. Finally, in the readout phase, we get a segmentation prediction map $\hat{\boldsymbol{S}} \in [0,1]^{W\times H}$ from $\boldsymbol{h}_i^K$ through a readout function $R(\cdot)$ (see Fig. 3(f)). Slightly different from Eq. 2, we concatenate the final node state $\boldsymbol{h}_i^K$ and the original node feature $\boldsymbol{v}_i$ (*i.e.*, $\boldsymbol{h}_i^0$) together and feed the combined feature into $R(\cdot)$:

$$\hat{S}_i = R_{\text{FCN}}([\boldsymbol{h}_i^K, \boldsymbol{v}_i]) \in [0,1]^{W\times H}. \tag{11}$$

Again, to preserve spatial information, the readout function is implemented as a small FCN.

As a message passing-based GNN model, these functions share weights among all the nodes. Moreover, our whole model is end-to-end trainable, as all the functions in AGNN are parameterized by neural networks.

3.3 AGNN for AVS

Network Configuration. We use the first five convolution blocks of DeepLabV3 [62] as our backbone for feature extraction. For an input video $\mathcal{I}$, each frame I_i (with a resolution of 473×473) is represented as a node v_i in the video graph $\mathcal{G}$ and associated with an initial node state $\mathbf{v}_i = \mathbf{h}_i^0 \in \mathbb{R}^{60\times 60\times 256}$. Then, after a total of $K(=3)$ message passing iterations, for each node v_i, we use the readout function in Eq. 11 to obtain a corresponding segmentation prediction map $\hat{S} \in [0,1]^{60\times 60}$. The convolution operations in the intra-attention (Eq. 4) and update function (Eq. 10) are realized with 1×1 convolution layers. The readout function (Eq. 11) consists of two 3×3 convolution layers cascaded by a 1×1 convolution layer with the *sigmoid* activation function.

Training Phase. As we operate on batches of a certain size (which is allowed to vary, depending on the GPU memory size), we leverage a random sampling strategy to train AGNN. Specifically, we split each training video $\mathcal{I}$ with a total of N frames into N' segments ($N' \leq N$) and randomly select one frame from each segment. Then we feed the N' sampled frames into a batch and train AGNN. Thus the relationships among all the N' sampling frames in each batch are represented using an N'-node graph. Such a sampling strategy provides robustness to variations and enables the network to fully exploit all frames. The diversity among the samples enables our model to better capture the underlying relationships and improve its generalizability. Let us denote the ground-truth segmentation mask and predicted foreground map for a training frame I_i as $S \in \{0,1\}^{60\times 60}$ and $\hat{S} \in [0,1]^{60\times 60}$. Our model is trained through the weighted binary cross entropy loss (see Fig. 3):

$$\mathcal{L}(S,\hat{S}) = -\sum\nolimits_{x}^{W\times H} (1-\eta) S_x \log(\hat{S}_x) + \eta(1-S_x)\log(1-\hat{S}_x), \quad (12)$$

where η indicates the ratio of foreground-background pixel number in S. Note that since AGNN handles multiple frames at the same time, it leads to a remarkably efficient training data augmentation strategy, as the combination of candidates are numerous.

Testing Phase. After training, we can apply the learned AGNN to perform per-pixel object prediction over unseen videos. For an input test video $\mathcal{I}$ with N frames (with 473×473 resolution), we split $\mathcal{I}$ into T subsets: $\{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_T\}$, where $T = N/N'$. Each subset contains N' frames with an interval of T frames: $\mathcal{I}_t = \{I_t, I_{t+T}, \ldots, I_{N-T+t}\}$. Then we feed each subset into AGNN to obtain the segmentation maps of all the frames in the subset. In practice, we set $N'=5$ during testing. We quantitatively study this setting in §4.4. As our AGNN does not require time-consuming optical flow computation and processes N' frames in one feed-forward propagation, it achieves a fast speed of $0.28s$ per frame. Following the widely used protocol [5], [7], [40], [69], we apply CRF [70] as a post-processing step with defaults parameter setting in [40], [69]. We show how to apply our trained AGNN model for handling instance-level AVS in the supplementary material.

3.4 AGNN for ICS

For ICS, AGNN is applied over a noisy collection of semantically related images $\mathcal{I}=\{I_i\}_{i=1}^N$, and generates corresponding binary segment masks: $\mathcal{S}=\{S_i\}_{i=1}^N$. The network architecture and training protocol are similar to the ones in the object-level AVS setting §3.3.

When processing a test image, ICS should make use of the whole related image group (instead of only sampling a few images during training). To this end, for each image I_i to be segmented, we uniformly split the other $N-1$ images into T groups and each group contains $N'-1$ images (*i.e.*, $T=(N-1)/(N'-1)$). Then we feed the first image group and I_i to a batch of size N', and store the node state for I_i. After that, we feed the next group and the store node state of I_i to get a new state of I_i. After T steps, the final state of I_i contains its relations to all other images and is used to produce its final co-segmentation result.

3.5 AGNN for FSS

For FSS, we denote the support images as $\mathcal{I} = \{I_i\}_{i=1}^N$, referring to a N-shot segmentation setting. The goal is to leverage the annotated support set to extract corresponding semantics in query images.

Following conventions [2], [11], we employ the first three blocks of VGG16 [78] as the backbone network to extract support and query features prototypes. We construct a N-node support graph $\mathcal{G}$ from $\mathcal{I}$. The core idea is to let AGNN mine the context in the support set and get updated node embeddings, from which more discriminative semantic representations can be derived. Then we calculate the distance between the query feature maps at each spatial location with the semantic representations, viewing the segmentation as classification at each spatial location. Specifically, with the annotations of the support set, we apply masked average pooling [10] over the support visual features to better extract class-specific information (including the background). Resulting vectors are used to initialize the node embeddings $\{\boldsymbol{h}_i^K\}_{i=1}^N$. Through several massage propagation iterations, the final node embeddings $\{\boldsymbol{h}_i^0\}_{i=1}^N$ are more powerful than before as they fully exploit knowledge from the support. As the node embeddings are vectorized, we use a vanilla MLP based GRU [68] to achieve iterative node state updating (Eq. 10). Then, for each query image, the segmentation is achieved via computing the cosine distance between the outputs of AGNN and query features at each spatial location [11]. Finally, we apply a softmax over the distances to produce a probability map over semantic classes.

3.6 Implementation Details

We implement our full algorithm by Pytorch. All experiments are conducted on a Nvidia TITAN Xp GPU.

AGNN for AVS. Following [40], [79], both static data from image salient object segmentation datasets, MSRA10K [80], DUT [81], and video data from the training set of DAVIS$_{16}$ are iteratively used to train our model. In a 'static-image' iteration, we randomly sample 6 images from the static training data to train our backbone network (DeepLabV3) to extract more discriminative foreground features. To train the backbone network, a 1×1 convolution layer with *sigmoid* activation functions is appended as an intermediate output layer, which can access the static image supervision signal. This is followed by a 'dynamic-video' iteration, in which we use the sampling strategy described in §3.3 to sample 6 video frames to train our whole AGNN model. The 'static-image' and 'dynamic-video' iterations are executed alternately. We randomly select 2 videos from the training set and sample 3 frames ($N'=3$) per video, due to computation limitations. In addition, we set the total number of iterations as $K=3$. The entire network is trained using the SGD optimizer with a 'ploy' learning rate schedule [62]: $lr = init\_lr \times (1-\frac{iters}{total\_iters})^{power}$, in which $power=0.9$ and the initial learning rate: $init\_lr = 2.5\times 10^{-4}$. The *total_iters* is $epochs\times batch\_size$. During training, the batch size and total epochs are set to 8 and 50. Data augmentation (*e.g.*, flipping, scaling and cropping) is also adopted for both static images and video data. The overall training time is about 20 hours. The input image size is 378×378 during training to save GPU memory and is enlarged to 473×473 to maintain a

TABLE 1
Quantitative object-level AVS results on the val set of DAVIS$_{16}$ [71] (§4.1) with IoU $\mathcal{J}$, boundary accuracy $\mathcal{F}$, and time stability $\mathcal{T}$. We also report the recall and the decay performance over time for both $\mathcal{J}$ and $\mathcal{F}$. The speed is also reported. The best two entries in each row are marked in gray. (* indicates deep learning based methods. The best scores are marked in **bold**. These notes also apply to the other tables.)

	Method	MSG [31]	NLC [28]	CUT [72]	FST [27]	*SFL [4]	*LMP [5]	*FSEG [6]	*LVO [7]	*ARP [35]	*PDB [40]	*MOT [73]	*LSMO [74]	*AGS [41]	*COSNet [69]	*Epo+ [75]	*AGNN
$\mathcal{J}$	Mean ↑	53.3	55.1	55.2	55.8	67.4	70.0	70.7	75.9	76.2	77.2	77.2	78.2	79.7	80.5	80.6	**81.3**
	Recall ↑	61.6	55.8	57.5	64.9	81.4	85.0	83.0	89.1	91.1	93.1	87.8	91.1	89.1	93.1	**95.2**	93.1
	Decay ↓	2.4	12.6	2.2	**0.0**	6.2	1.3	1.5	**0.0**	7.0	0.9	5.0	4.1	1.9	4.4	2.2	4.4
$\mathcal{F}$	Mean ↑	50.8	52.3	55.2	51.1	66.7	65.9	65.3	72.1	70.6	74.5	77.4	75.9	77.4	79.4	75.5	**79.7**
	Recall ↑	60.0	61.0	51.9	51.6	77.1	79.2	73.8	83.4	83.5	84.4	84.4	84.7	85.8	**89.5**	87.9	88.5
	Decay ↓	5.1	11.4	3.4	2.9	5.1	2.5	1.8	1.3	7.9	**-0.2**	3.3	3.5	0.0	5.0	2.4	5.1
$\mathcal{T}$	Mean ↓	30.2	42.5	27.7	36.6	28.2	57.2	32.8	26.5	39.3	29.1	27.9	**21.2**	26.7	18.4	19.3	33.7
Time (s) ↓		169.0	12.0	35.0	51.4	7.9	18.3	7.2	13.5	124.7	0.7	1.0	>2.5	0.6	0.48	3.0	0.6

TABLE 2
Quantitative object-level AVS performance of each category on Youtube-Objects [76] (§4.1) with IoU $\mathcal{J}$. We show the performance for each of the 10 categories from the dataset. The final row shows an average over all the videos.

Method	LTV [32]	FST [27]	COSEG [77]	*ARP [35]	*LVO [7]	*PDB [40]	*FSEG [6]	*SFL [4]	*MOT [73]	*LSMO [74]	*AGS [41]	*COSNet [69]	*JHT [44]	*AGNN
Airplane (6)	13.7	70.9	69.3	73.6	86.2	78.0	81.7	65.6	77.2	60.5	**87.7**	81.1	81.8	86.0
Bird (6)	12.2	70.6	76.0	56.1	81.0	80.0	63.8	65.4	42.2	59.3	76.7	75.7	**81.2**	75.7
Boat (15)	10.8	42.5	53.5	57.8	68.5	58.9	**72.3**	59.9	49.3	62.1	72.2	71.3	67.6	68.7
Car (7)	23.7	65.2	70.4	33.9	69.3	76.5	74.9	64.0	68.6	72.3	78.6	77.6	79.5	**82.4**
Cat (16)	18.6	52.1	66.8	30.5	58.8	63.0	68.4	58.9	46.3	66.3	**69.2**	66.5	65.8	65.9
Cow (20)	16.3	44.5	49.0	41.8	68.5	64.1	68.0	51.1	64.2	67.9	64.6	69.8	66.2	**70.5**
Dog (27)	18.2	65.3	47.5	36.8	61.7	70.1	69.4	54.1	66.1	70.0	73.3	76.8	73.4	**77.1**
Horse (14)	11.5	53.5	55.7	44.3	53.9	67.6	60.4	64.8	64.8	65.4	64.4	67.4	69.5	**72.2**
Motorbike (10)	10.6	44.2	39.5	48.9	60.8	58.3	62.7	52.6	44.6	55.5	62.1	67.7	**69.3**	63.8
Train (5)	19.6	29.6	53.4	39.2	**66.3**	35.2	62.2	34.0	42.3	38.0	48.2	46.8	49.7	47.8
Mean $\mathcal{J}$↑	15.5	53.8	58.1	46.2	67.5	65.4	68.4	57.0	58.1	64.3	69.7	70.5	70.9	**71.4**
Time (s) ↓	1.0	51.4	>10.0	124.7	13.5	0.7	7.2	7.9	1.0	>2.5	0.6	0.48	0.3	0.6

higher spatial resolution during testing. For the inference speed, a forward pass with one image (batch) takes around 0.3 s, while CRF-based post-processing takes about 0.3 s.

AGNN for ICS. Following [8], [9], we use the training data of PASCAL VOC to train AGNN. In each iteration, we randomly select two semantic classes and sample a group of three images per class. All other parameter settings are the same as object-level AVS.

AGNN for FSS. As is standard, AGNN is trained on three splits of PASCAL-5^i [2] and evaluated on the remaining one in a cross-validation fashion. During training, we build several *episodes* from the training splits. To instantiate the 1-way 5-shot segmentation setting, each episode consists of five support images (from a same class) and one query image. The whole model is trained by the cross entropy loss and optimized by SGD with the momentum of 0.9. The initialized learning rate is set to $1e{-}3$ and decreased by 0.1 every 10000 iterations. During testing, as in [11], we sample 1000 episodes for evaluation and average the results from 5 runs with different random seeds to obtain stable results.

4 Experiments

In this section, we comprehensively examine the performance of AGNN on three tasks, *i.e.*, AVS (§4.1), ICS (§4.2), and FSS (§4.3). Finally, in §4.4, we conduct an ablation study to evaluate essential components of AGNN.

4.1 Performance on AVS

4.1.1 Experimental Setup

Datasets and Metrics: We use two well-known datasets:

- **DAVIS$_{16}$ [71]** consists of 50 videos total (30 for training and 20 for testing) with pixel-wise annotations for every frame. Following the standard protocol, three evaluation criteria are used: Intersection-over-Union (IoU) $\mathcal{J}$, boundary accuracy $\mathcal{F}$, and time stability $\mathcal{T}$.
- **Youtube-Objects [76]** consists of 126 video sequences belonging to 10 object categories and contains more than 20000 frames in total. We test our method on the whole dataset. We follow convention and use $\mathcal{J}$ to measure the segmentation performance.

4.1.2 Quantitative Performance

Val-set of DAVIS$_{16}$. We compare the proposed AGNN with 15 famous AVS methods on DAVIS$_{16}$ benchmark. The results are summarized in Table 1. We can see that our AGNN outperforms the second best results (*i.e.*, Epo+ [75]) on DAVIS$_{16}$ benchmark in terms of the two most important indicators: mean $\mathcal{J}$ (81.3% *vs* 80.6%) and $\mathcal{F}$ (79.7% *vs* 75.5%). The performance gain proves the advantage of our AGNN, which considers the rich relations among a set of video frames, instead of pairwise relation in COSNet or local motion cues in Epo+. Despite that MOT and LSMO take both RGB image and optical flow as input, the performances are inferior to AGNN (-4.1%/-3.1% in Mean $\mathcal{J}$ and -2.3%/-3.8% in Mean $\mathcal{F}$). Compared to PDB, which uses the same training protocol and training datasets, our AGNN yields significant performance improvements of 4.1% and 5.2% in terms of mean $\mathcal{J}$ and mean $\mathcal{F}$, respectively.

Youtube-Objects. Table 2 illustrates the results of all compared methods for different categories. Our approach again outperforms all the compared methods by a large margin. By means of considering the local sequential relationship among different frames, AGS [41] gains much better performance than other competitors (69.7% in mean $\mathcal{J}$). However, our AGNN achieves superior performance (71.4% *vs* 69.7%) to AGS over 10 categories. We attribute the performance gain to that our AGNN can make full use of long-term video

TABLE 3
Attribute-based object-level AVS performance on the DAVIS$_{16}$ dataset[71]. For each method, the corresponding column indicates the mean ($\mathcal{J}\uparrow$) over all videos with that specific attribute (*e.g.*, LR).

Attr	MSG [31]	NLC [28]	CUT [72]	FST [27]	*SFL [4]	*LMP [5]	*FSEG [6]	*LVO [7]	*ARP [35]	*PDB [40]	*MOT [73]	*LSMO [74]	*AGS [41]	*COSNet [69]	*Epo+ [75]	***AGNN**
AC	56.9	60.8	58.0	56.4	59.9	71.5	70.0	74.0	78.1	77.0	77.3	76.5	79.9	82.9	83.0	**83.1**
BC	60.8	34.4	52.1	56.0	76.3	72.8	76.7	78.2	70.7	76.9	77.1	76.1	80.7	79.6	78.1	**80.8**
CS	53.6	50.0	67.1	56.2	73.3	71.8	76.7	80.7	76.0	78.4	81.8	80.5	82.4	82.3	82.0	**83.6**
DB	47.1	48.3	35.4	46.9	27.0	58.3	50.0	55.2	71.7	62.4	55.3	60.6	66.5	68.7	**72.0**	67.3
DEF	48.4	58.3	55.9	52.1	66.6	70.7	69.5	75.2	76.6	76.3	78.7	76.2	77.7	78.5	**81.0**	79.8
EA	51.4	45.6	49.3	55.3	67.5	67.4	69.0	73.7	71.1	75.9	75.4	78.0	77.6	76.7	76.0	**78.2**
FM	44.1	56.5	52.3	56.2	61.6	65.5	69.9	70.5	75.3	76.4	75.7	76.4	**79.1**	77.2	78.0	78.4
HO	48.8	55.2	51.2	52.2	61.2	66.7	65.2	71.9	75.2	73.9	74.9	73.9	76.2	76.2	**78.0**	77.4
IO	56.1	55.0	59.6	51.0	66.5	67.7	66.9	75.1	76.7	74.6	77.0	74.0	77.4	76.9	**78.0**	**78.0**
LR	53.7	57.2	54.1	57.0	66.8	67.2	71.8	75.0	74.3	77.7	78.7	**81.4**	**81.4**	77.9	75.0	80.4
MB	39.8	53.6	51.0	50.1	65.6	63.4	65.4	71.1	72.9	74.0	74.6	73.5	76.2	76.0	75.9	**76.6**
OCC	43.0	68.5	40.8	50.3	67.9	66.6	64.3	73.6	74.3	77.9	76.8	**80.4**	78.3	73.9	75.0	77.4
OV	46.4	52.4	58.0	58.7	65.4	60.9	72.3	71.5	79.6	77.6	77.3	74.3	81.2	80.5	**83.0**	78.8
SC	42.7	52.9	46.8	47.9	63.8	62.3	61.5	70.5	71.1	72.2	73.3	**74.3**	73.6	69.5	72.0	71.6
SV	51.4	47.6	48.1	50.3	63.8	65.9	65.5	72.9	74.7	74.4	75.1	75.1	77.6	77.2	76.0	**78.6**
Avg.	53.3	55.1	55.2	55.8	67.4	70.0	70.7	75.9	76.2	77.2	78.2	77.2	79.7	80.5	80.6	**81.3**

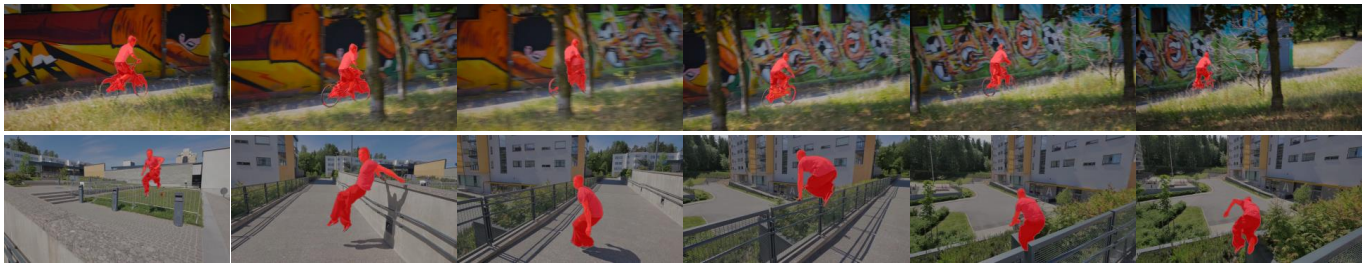

Fig. 5. Qualitative object-level AVS results on DAVIS$_{16}$ [71] (from top to bottom: *parkour* and *bmx-tree*). It can be observed that the proposed algorithm is applicable to the primary target with shape deformation, similar target distraction, and fast motion scenarios.

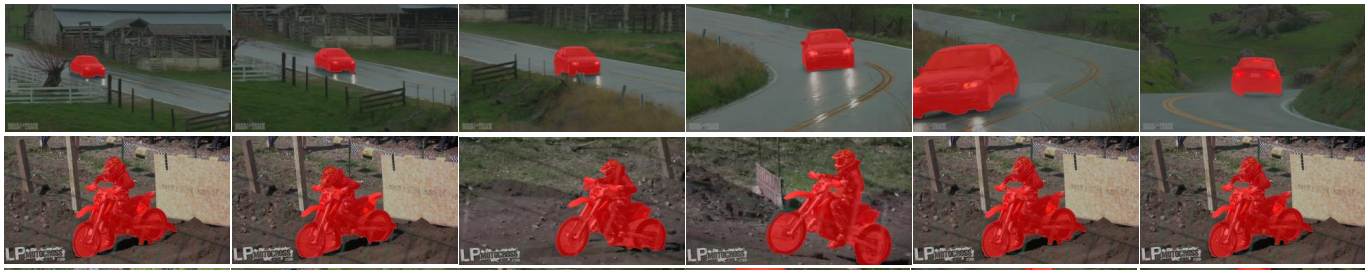

Fig. 6. Qualitative object-level AVS results on Youtube-Objects [76] (from top to bottom: *car0001* and *motorbike0002*). It can be observed that the proposed algorithm is applicable to handle various challenging factors, such as view changes, background clutter, and large shape deformation.

context by recursive message passing. It is worth noting that LSMO [74] and MOT [73] show large performance drop on Youtube-objects dataset (rank sixth and seventh), compared with their behaviors on DAVIS$_{16}$. In contrast, our method consistently achieves state-of-the-art over these two datasets without online learning [73] or complex object proposals [35], which demonstrates the strong generalization capability of AGNN. To further verify the computation efficiency of the proposed AGNN, we conduct running time comparisons on both DAVIS$_{16}$ and Youtube-objects. We can see that our AGNN runs faster than most of its counterparts. This is because AGNN does not need to compute optical flow [4]–[6], [27], [28], [30], [73], [74], perform online learning [73], or incorporate object proposal generation [28], [35]. In particular, AGNN achieves comparable processing speed to AGS [41] (0.6 s per frame) and PDB [40] (0.7 s per frame) with better performance.

4.1.3 Qualitative Performance

Fig. 5 and Fig. 6 depict our segmentation results on several challenging video sequences *parkour*, *bmx-tree*, *car0001*, and *motorbike0002* of DAVIS$_{16}$ and Youtube-Objects, respectively. The primary objects undergo significant scale variation (*e.g.*, *car0001*), deformation (*e.g.*, *parkour*) and view changes (*e.g.*, *car0001* and *motorbike0002*), but our AGNN still generates high-quality results. Furthermore, for *bmx-tree*, our AGNN can segment the objects under occlusion by taking advantage of global information.

4.1.4 Attribute-Based Study

Next we provide an attribute-based study on DAVIS$_{16}$, enabling a more in-depth analysis. From Table 3 we can find AGNN outperforms other competitors across most attribute categories, such as appearance changes (AC), background clutter (BC), camera-shake (CS), dynamic background (DB),

TABLE 4
Quantitative ICS performance on PASCAL VOC [82] (§4.2) with mean IoU $\mathcal{J}$.

Method	CSC [83]	MRW [84]	GO-FMR [85]	ICSC [86]	OCC [87]	*FCNs [88]	*CA [8]	*FCA [8]	*CSA [8]	*DOCS [9]	*COA [55]	***AGNN**
Mean $\mathcal{J}$ ↑	46.0	33.0	52.0	45.0	40.0	55.2	59.2	59.4	59.8	57.8	60.0	**60.8**
Time (s) ↓	251.5	-	-	73.0	-	3.7	0.2	0.3	1.7	14.5	15.6	0.8

TABLE 5
Quantitative ICS performance on Internet [89] (§4.2) with mean IoU $\mathcal{J}$. We show the per-class performance and overall average.

Method	DC [90]	Internet [89]	TDK [91]	GO-FMR [85]	SGC^3 [92]	ICSC [86]	*DDCRF [93]	*CA [8]	*FCA [8]	*CSA [8]	*DOCS [9]	*COA [55]	***AGNN**
Car	37.1	64.4	64.9	66.8	66.4	71.0	72.0	80.0	76.9	79.9	82.7	82.0	**84.0**
Horse	30.1	51.6	33.4	58.1	55.3	60.0	65.0	67.3	69.1	71.4	64.6	61.0	**72.6**
Airplane	15.3	57.3	46.2	60.4	42.8	61.0	67.7	72.8	70.6	73.1	70.3	67.0	**76.1**
Avg.	27.5	57.3	46.2	54.8	60.4	64.0	67.7	70.3	72.8	70.6	73.1	67.7	**77.6**
Time(s)	-	7.0	-	-	10.9	73.0	-	0.2	0.3	1.7	14.5	15.6	0.8

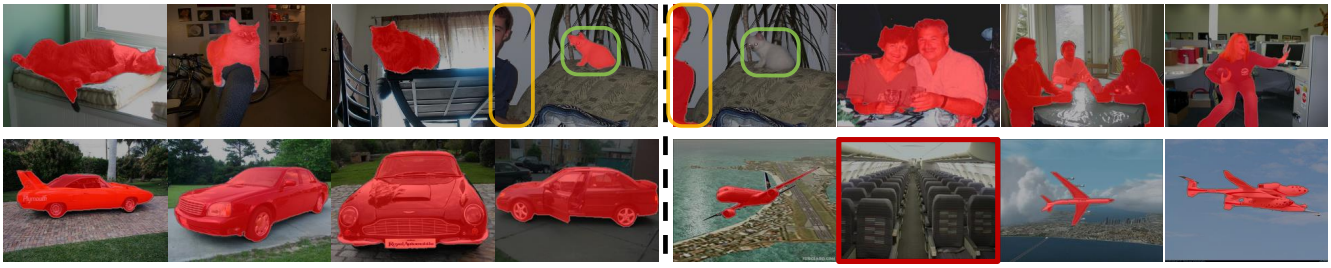

Fig. 7. Qualitative ICS results (§4.2) on PASCAL VOC [82] (top: *cat* and *person* image collections) and Internet [89] (bottom: *car* and *airplane* image collections). Noisy samples are labeled in red rectangles.

deformation (DEF), edge ambiguity (EA), fast motion (FM), heterogeneous object (HO), interacting objects (IO), low resolution (LR), motion blur (MB), shape complexity (SC) and scale variation (SV). Especially for the videos in which the primary objects undergo noticeable appearance variation, camera shake, and complex boundaries, AGNN achieves average $\mathcal{J}$ of 83.1%, 83.6% and 78.6%, which are much better than all compared methods.

4.2 Performance on ICS

4.2.1 Experimental Setup

Datasets and Metrics: We perform experiments on two well-known ICS datasets:

- **PASCAL VOC[82]** contains 1464 training images and 1449 validation images. Following [9], we split the validation set into 724 validation and 725 test images, and use mean IoU $\mathcal{J}$ as the performance measure.
- **Internet [89]** has 1306 car, 879 horse, and 561 airplane images. As in [8], [85], the performance is reported on a subset of Internet (100 images per class) with mean $\mathcal{J}$.

4.2.2 Quantitative Performance

It is challenging to segment the objects in PASCAL VOC as they vary greatly in scale and appearance. Moreover, some images have multiple objects belonging to different categories. Table 4 shows quantitative results on PASCAL VOC. FCNs [88] segment each image individually (without considering other related images) and thus give poor performance. Both [8] and [9] consider relations within image pairs and gain better results. AGNN achieves the best performance (60.8%) as it better utilizes relational information from multiple images, enabling it to better identify common object patterns. On the Internet dataset, the quantitative results in Table 5 show that AGNN sets a new state-of-the-art on each class and on average. Moreover, we can see most end-to-end deep learning-based ICS methods achieve faster inference speed than traditional clustering-based ICS methods and our method is among the fastest ones.

4.2.3 Qualitative Performance

Fig. 7 visualizes some representative co-segmentation results. Specifically, the first four images in the top row belong to the *Cat* category, while the last four images all contain the *Person* semantics. For the two groups, the corresponding common semantics are successfully extracted; even for a single image, the predictions are different when considering different related images (highlighted in green and yellow circles of the first row). For the second row, AGNN also performs well in cases with significant intra-class appearance change (*Car*). For some samples (red boxes) that do not contain the common objects among the category, our AGNN can filter out these noisy samples successfully (due to the gate mechanism).

4.3 Performance on FSS

4.3.1 Experimental Setup

We perform experiments on PASCAL-5^i [2], a gold-standard dataset for FSS. In PASCAL-5^i, 20 categories are evenly divided into 4 folds (splits): {aeroplane, bicycle, bird, boat, bottle}, {bus, car, cat, chair, cow}, {diningtable, dog, horse, motorbike, person}, {potted plant, sheep, sofa, train, tv/monitor}, within three folds as the training classes and

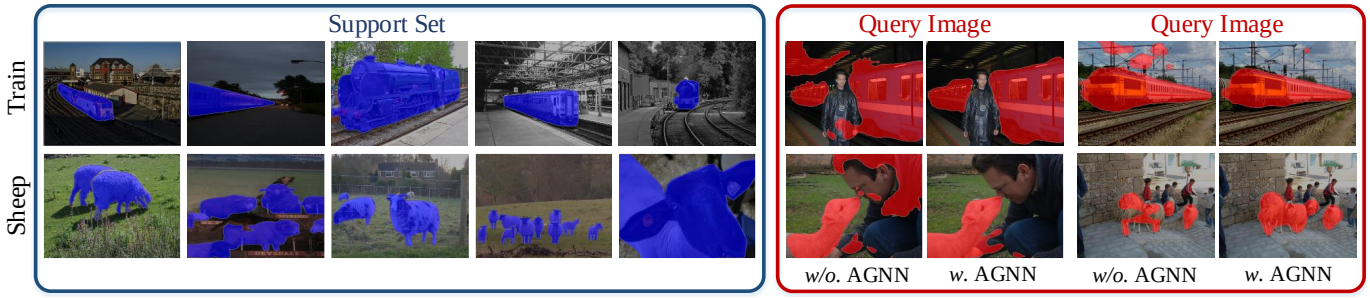

Fig. 8. Qualitative FSS results on PASCAL-5^i [2] (§4.3). The first five columns are annotated support images and the last two columns are query predictions *w/o.* and *w.* AGNN.

TABLE 6
Quantitative FSS performance on PASCAL-5^i [2] (§4.3) the with 1-way 5-shot segmentation setting in terms of mean IoU $\mathcal{J}$.

Method	split-1	split-2	split-3	split-4	Mean $\mathcal{J}\uparrow$	Time (s) $\downarrow$
*OSLSM [2]	35.9	58.1	42.7	39.1	43.9	0.21
*co-FCN [94]	37.5	50.0	44.1	33.9	41.4	-
*SG-One [10]	41.9	58.6	48.6	39.4	47.1	0.35
*AMP [58]	41.8	55.5	50.3	39.9	46.9	0.19
*FWB [59]	50.9	62.8	56.5	**50.1**	55.1	0.39
*PANet [11]	51.8	**64.6**	**59.8**	46.5	55.7	0.27
***AGNN**	**53.2**	64.3	58.4	49.2	**56.3**	0.28

TABLE 7
Ablation study on the test set of DAVIS_{16} [71] (§4.4) with different graph structures, message passing steps, and input images numbers.

Components	Module	DAVIS_{16} mean $\mathcal{J}$	$\Delta\mathcal{J}$
Reference	**Full model** (3 Iterations, N′= 5)	81.3	-
Graph Structure	*w/o.* AGNN	72.2	-9.1
	w/o. Intra-Attention Loop-Edge	79.1	-2.2
	w/o. Inter-Attention Line-Edge	73.8	-7.5
	w/o. Gated Message (Eq. 9)	80.1	-1.2
	Intra-Attention → Dilation Conv	79.3	-1.0
	Inter-Attention → convLSTM	77.2	-4.1
Message Passing	1 iteration	79.3	-2.0
	2 iterations	80.6	-0.7
	4 iterations	81.3	0.0
Input Frames	N′= 3	80.0	-1.3
	N′= 6	81.3	0.0
	N′= 7	81.3	0.0
Post-process	*w/o.* CRF	80.2	-1.1

one fold as the testing class. We report the average performance, in terms of IoU $\mathcal{J}$, over 4 testing folds and overall mean values.

4.3.2 Quantitative Performance

Table 6 reports quantitative comparison results on PASCAL-5^i with the 1-way 5-shot setting. Our model achieves an IoU score of 56.3%, surpassing the state-of-the-art method, PANet [11], by 0.6%. This implies the importance of comprehensively exploring the context in the support set for FSS. We also provide the FSS inference time for all compared methods. We can see that the inference speed of our AGNN is on par with current arts.

4.3.3 Qualitative Performance

As shown in Fig. 8, our model gives reasonable results on new classes with only five annotated support images. In addition, comparing the last two columns, we see that the segmentation results become more precise after applying AGNN. Taken together, these results demonstrate AGNN can successfully learn new concepts and generalize to unseen classes, making full use of the context from only a few labeled examples.

4.4 Ablation Study

We perform an ablation study on the object-level AVS task to investigate the effect of each component of AGNN.

Effectiveness of Our AGNN: To quantify the efficacy of our AGNN, we derive a baseline *w/o AGNN*, which indicates the results from our backbone model, DeepLabV3. As shown in Table 7, AGNN indeed brings significant performance improvements (72.2%→81.3% in term of mean $\mathcal{J}$). Moreover, we investigate the importance of intra-attention and inter-attention modules. We can see that removing either the intra-attention or inter-attention module hurts performance (-2.2%/-7.5% in Mean $\mathcal{J}$). To further show the advantages of intra-attention and inter-attention modules in long-term temporal correlation modeling and spatial context capturing, we substitute these two modules with convLSTM and dilation convolution, respectively. From Table 7 we see that performance of convLSTM and dilation convolution-based variants are inferior to the full AGNN model (77.2% *vs* 81.3%, 79.3% *vs* 81.3%).

Gated Message Aggregation Strategy: In Eq. 9, we equip the message passing with a channel-wise gated mechanism to decrease the negative influence of irrelevant frames. To evaluate this design, we offer a baseline *w/o Gated Message*, which aggregates messages directly. A performance degradation is observed after excluding the gates. To further intuitively show the strong learning ability our framework, following the experimental protocol in slow feature analysis [95], we visualize how foreground feature embeddings change over time in Fig. 9. We see that the embeddings learned by our AGNN are significantly more stable than the baseline methods. In particular, disabling gated message passing (green line) makes the network behave similar to original DeepLabv3 (blue line). This suggests that our AGNN is capable of capturing stable global invariance information [96] with recursive message passing.

Number of Message Passing Iterations K: To investigate the effect of the number of message passing iterations K,

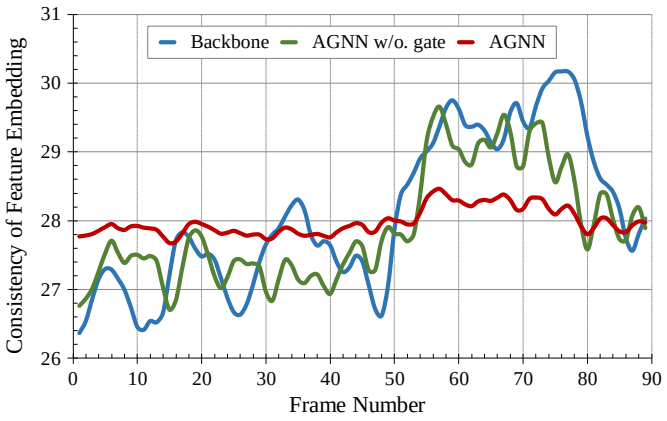

Fig. 9. Consistency of feature embedding over time, reported on the test set of DAVIS$_{16}$ [71] (§4.4).

we report the performance as a function of K. We find that more iterations ($1 \rightarrow 3$) achieves better results, while the performance seems to converges at $K = 3$.

Number of Nodes N' During Inference: We also report performance with different values of the number of nodes N'. We observe that the performance increases with more input frames up to about 5, while from 5 to 7, the final performance does not change significantly (probably due to temporal redundancy in video clips).

5 Conclusion

Large-scale deep learning techniques have achieved great advances in different object segmentation related tasks, which were unimaginable just several years ago. This leads to a question: What is the next? Rather than current popular solutions striving for designing specific network architectures to best fit their specific tasks, we make a further step towards a generic deep learning framework, AGNN, that formulates diverse segmentation tasks, including AVS, ICS, and FSS, from a unified view of segmenting objects from relational visual data. This not only provides insight into the underlying natures of these tasks, but also pushes their research boundaries. Specifically, AGNN achieved AVS, ICS, and FSS through an iterative neural graph algorithm, efficiently addressing the limitations of current solutions by comprehensive relation modeling. In essence, AGNN leverages a neural attention mechanism to fully capture the relations between data instances and performs recursive message passing to progressively mine context information. Through extensive experiments on several representative AVS, ICS, and FSS datasets, we have demonstrated that AGNN is a generic framework that is able to generate promising results over dynamic video data or a group of semantically related images, and even generalizes to unseen classes with only a few annotated examples.

References

[1] W. Wang, X. Lu, J. Shen, D. J. Crandall, and L. Shao, "Zero-shot video object segmentation via attentive graph neural networks," in *IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9236–9245.

[2] A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots, "One-shot learning for semantic segmentation," in *British Machine Vision Conference*, 2017.

[3] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Computing Surveys*, vol. 53, no. 3, pp. 1–34, 2020.

[4] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang, "Segflow: Joint learning for video object segmentation and optical flow," in *IEEE Int. Conf. Comput. Vis.*, 2017, pp. 686–695.

[5] P. Tokmakov, K. Alahari, and C. Schmid, "Learning motion patterns in videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 531–539.

[6] S. D. Jain, B. Xiong, and K. Grauman, "Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 686–695.

[7] P. Tokmakov, K. Alahari, and C. Schmid, "Learning video object segmentation with visual memory," in *IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4491–4500.

[8] H. Chen, Y. Huang, and H. Nakayama, "Semantic aware attention based deep object co-segmentation," in *Asi. Conf. Comput. Vis.*, 2018, pp. 135–150.

[9] W. Li, O. H. Jafari, and C. Rother, "Deep object co-segmentation," in *Asi. Conf. Comput. Vis.*, 2018, pp. 435–450.

[10] X. Zhang, Y. Wei, G. Kang, Y. Yang, and T. Huang, "Self-produced guidance for weakly-supervised object localization," pp. 597–613, 2018.

[11] K. Wang, J. H. Liew, Y. Zou, D. Zhou, and J. Feng, "Panet: Few-shot image semantic segmentation with prototype alignment," in *IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9196–9205.

[12] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Network and Learning Systems*, vol. 20, no. 1, pp. 61–80, 2009.

[13] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," in *Proc. Advances Neural Inf. Process. Syst.*, 2015, pp. 2224–2232.

[14] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *Proc. ACM Int. Conf. Mach. Learn.*, 2016, pp. 2014–2023.

[15] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2017.

[16] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," *arXiv preprint arXiv:1709.05584*, 2017.

[17] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. ACM Int. Conf. Mach. Learn.*, 2017, pp. 1263–1272.

[18] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Representations*, 2018.

[19] T. Zhou, S. Qi, W. Wang, J. Shen, and S.-C. Zhu, "Cascaded parsing of human-object interaction recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021.

[20] L. Fan, W. Wang, S. Huang, X. Tang, and S.-C. Zhu, "Understanding human gaze communication by spatio-temporal graph reasoning," in *IEEE Int. Conf. Comput. Vis.*, 2019, pp. 5724–5733.

[21] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei, "Scene graph generation by iterative message passing," in *IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5410–5419.

[22] W. Wang, T. Zhou, S. Qi, J. Shen, and S.-C. Zhu, "Hierarchical human semantic parsing with comprehensive part-relation modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021.

[23] Z. Zheng, W. Wang, S. Qi, and S.-C. Zhu, "Reasoning visual dialogs with structural and partial observations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6669–6678.

[24] H. Wang, W. Wang, W. Liang, C. Xiong, and J. Shen, "Structured scene memory for vision-language navigation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8455–8464.

[25] W. Wang, T. Zhou, F. Porikli, D. Crandall, and L. V. Gool, "A survey on deep learning technique for video segmentation," *arXiv preprint arXiv:2107.01153*, 2021.

[26] T. Brox and J. Malik, "Object segmentation by long term analysis of point trajectories," in *The Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 282–295.

[27] A. Papazoglou and V. Ferrari, "Fast object segmentation in unconstrained video," in *IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1777–1784.

[28] A. Faktor and M. Irani, "Video segmentation by non-local consensus voting," in *British Machine Vision Conference*, 2014.

[29] Y.-H. Tsai, M.-H. Yang, and M. J. Black, "Video segmentation via object flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3899–3908.

[30] W. Wang, J. Shen, and F. Porikli, "Saliency-aware geodesic video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3395–3402.

[31] P. Ochs and T. Brox, "Object segmentation in video: A hierarchical variational approach for turning point trajectories into dense regions," in *IEEE Int. Conf. Comput. Vis.*, 2011, pp. 1583–1590.

[32] P. Ochs, J. Malik, and T. Brox, "Segmentation of moving objects by long term video analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 6, pp. 1187–1200, 2014.

[33] W. Wang, J. Shen, F. Porikli, and R. Yang, "Semi-supervised video object segmentation with super-trajectories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 985–998, 2018.

[34] D. Zhang, O. Javed, and M. Shah, "Video object segmentation through spatially accurate and temporally dense extraction of primary object regions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 628–635.

[35] Y. J. Koh and C. Kim, "Primary object segmentation in videos based on region augmentation and reduction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7417–7425.

[36] Y. Jun Koh, Y.-Y. Lee, and C.-S. Kim, "Sequential clique optimization for video object segmentation," in *The Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 517–533.

[37] C. Ventura, M. Bellver, A. Girbau, A. Salvador, F. Marques, and X. Giro-i Nieto, "Rvos: End-to-end recurrent network for video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5277–5286.

[38] X. Lu, W. Wang, J. Shen, D. Crandall, and J. Luo, "Zero-shot video object segmentation with co-attention siamese networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.

[39] X. Lu, W. Wang, M. Danelljan, T. Zhou, J. Shen, and L. Van Gool, "Video object segmentation with episodic graph memory networks," in *The Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 661–679.

[40] H. Song, W. Wang, S. Zhao, J. Shen, and K.-M. Lam, "Pyramid dilated deeper convlstm for video salient object detection," in *The Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 744–760.

[41] W. Wang, H. Song, S. Zhao, J. Shen, S. Zhao, S. C. Hoi, and H. Ling, "Learning unsupervised video object segmentation through visual attention," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3064–3074.

[42] G. Li, Y. Xie, T. Wei, K. Wang, and L. Lin, "Flow guided recurrent neural encoder for video salient object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3243–3252.

[43] X. Lu, W. Wang, J. Shen, Y.-W. Tai, D. J. Crandall, and S. C. Hoi, "Learning video object segmentation from unlabeled videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8960–8970.

[44] L. Zhang, J. Zhang, Z. Lin, R. Mech, H. Lu, and Y. He, "Unsupervised video object segmentation with joint hotspot tracking," in *The Proc. Eur. Conf. Comput. Vis.*, 2020.

[45] J. Luiten, I. E. Zulfikar, and B. Leibe, "Unovost: Unsupervised offline video object segmentation and tracking," in *Proc. IEEE/CVF Winter Conf. on Applications of Comput. Vis.*, 2020, pp. 2000–2009.

[46] A. Dave, P. Tokmakov, and D. Ramanan, "Towards segmenting anything that moves," in *IEEE Int. Conf. Comput. Vis. Workshop*, 2019, pp. 1493–1502.

[47] C. Rother, T. Minka, A. Blake, and V. Kolmogorov, "Cosegmentation of image pairs by histogram matching-incorporating a global constraint into MRFs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 993–1000.

[48] L. Mukherjee, V. Singh, and C. R. Dyer, "Half-integrality based algorithms for cosegmentation of images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 2028–2035.

[49] D. S. Hochbaum and V. Singh, "An efficient algorithm for co-segmentation," in *IEEE Int. Conf. Comput. Vis.*, 2009, pp. 269–276.

[50] W. Wang and J. Shen, "Higher-order image co-segmentation," *IEEE Trans. Multimedia*, vol. 18, no. 6, pp. 1011–1021, 2016.

[51] S. Vicente, V. Kolmogorov, and C. Rother, "Cosegmentation revisited: Models and optimization," in *The Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 465–479.

[52] G. Kim, P. E. Xing, L. Fei-Fei, and T. Kanade, "Distributed cosegmentation via submodular optimization on anisotropic diffusion," in *IEEE Int. Conf. Comput. Vis.*, 2011, pp. 169–176.

[53] J. C. Rubio, J. Serrat, A. Lopez, and N. Paragios, "Unsupervised co-segmentation through region matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 749–756.

[54] S. Vicente, C. Rother, and V. Kolmogorov, "Object cosegmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 2217–2224.

[55] K.-J. Hsu, Y.-Y. Lin, and Y.-Y. Chuang, "Co-attention cnns for unsupervised object co-segmentation," in *Proc. Int. Join. Conf. Arti. Intell.*, 2018, pp. 748–756.

[56] S. Banerjee, A. Hati, S. Chaudhuri, and R. Velmurugan, "Cosegnet: Image co-segmentation using a conditional siamese convolutional network," in *Proc. Int. Join. Conf. Arti. Intell.*, 2019, pp. 673–679.

[57] M. Zhen, S. Li, L. Zhou, J. Shang, H. Feng, T. Fang, and L. Quan, "Learning discriminative feature with crf for unsupervised video object segmentation," in *The Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 445–462.

[58] M. Siam, B. N. Oreshkin, and M. Jagersand, "Amp: Adaptive masked proxies for few-shot segmentation," in *IEEE Int. Conf. Comput. Vis.*, 2019, pp. 5248–5257.

[59] K. Nguyen and S. Todorovic, "Feature weighting and boosting for few-shot segmentation," in *IEEE Int. Conf. Comput. Vis.*, 2019, pp. 622–631.

[60] X. Zhang, Y. Wei, Y. Yang, and T. S. Huang, "Sg-one: Similarity guidance network for one-shot semantic segmentation," *IEEE Transactions on Cybernetics*, vol. 50, no. 9, pp. 3855–3865, 2020.

[61] V. Garcia and J. Bruna, "Few-shot learning with graph neural networks," *Proc. Int. Conf. Learn. Representations*, 2018.

[62] L. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *CoRR*, vol. abs/1706.05587, 2017.

[63] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Advances Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[64] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7794–7803.

[65] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *Proc. ACM Int. Conf. Mach. Learn.*, 2019, pp. 7354–7363.

[66] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical question-image co-attention for visual question answering," in *Proc. Advances Neural Inf. Process. Syst.*, 2016, pp. 289–297.

[67] N. Ballas, L. Yao, C. Pal, and A. Courville, "Delving deeper into convolutional networks for learning video representations," in *Proc. Int. Conf. Learn. Representations*, 2016.

[68] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *Empirical Methods in Natural Language Processing*, 2014, pp. 1724–1734.

[69] X. Lu, W. Wang, C. Ma, J. Shen, L. Shao, and F. Porikli, "See more, know more: Unsupervised video object segmentation with co-attention siamese networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3623–3632.

[70] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," *Advances in neural information processing systems*, vol. 24, pp. 109–117, 2011.

[71] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 724–732.

[72] M. Keuper, B. Andres, and T. Brox, "Motion trajectory segmentation via minimum cost multicuts," in *IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3271–3279.

[73] M. Siam, C. Jiang, S. Lu, L. Petrich, M. Gamal, M. Elhoseiny, and M. Jagersand, "Video segmentation using teacher-student adaptation in a human robot interaction (hri) setting," in *Proc. IEEE Conf. Robot. Autom*, 2019, pp. 50–56.

[74] P. Tokmakov, C. Schmid, and K. Alahari, "Learning to segment moving objects," *Int. J. Comput. Vis.*, vol. 127, no. 3, pp. 282–301, 2019.

[75] I. Akhter, M. Ali, M. Faisal, and R. Hartley, "Epo-net: Exploiting geometric constraints on dense trajectories for motion saliency," in *The IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 1873–1882.

[76] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari, "Learning object class detectors from weakly annotated video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3282–3289.
[77] Y.-H. Tsai, G. Zhong, and M.-H. Yang, "Semantic co-segmentation in videos," in *The Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 760–775.
[78] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Proc. Int. Conf. Learn. Representations*, 2014.
[79] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung, "Learning video object segmentation from static images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3491–3500.
[80] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 569–582, 2015.
[81] C. Yang, L. Zhang, H. Lu, X. Ruan, and M. Yang, "Saliency detection via graph-based manifold ranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 3166–3173.
[82] M. Everingham, S. M. A. Eslami, L. J. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, 2015.
[83] A. Faktor and M. Irani, "Co-segmentation by composition," in *IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1297–1304.
[84] C. Lee, W. Jang, J. Sim, and C. Kim, "Multiple random walkers and their application to image cosegmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3837–3845.
[85] R. Quan, J. Han, D. Zhang, and F. Nie, "Object co-segmentation via graph optimized-flexible manifold ranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 687–695.
[86] K. R. Jerripothula, J. Cai, and J. Yuan, "Image co-segmentation via saliency co-fusion," *IEEE Trans. Multimedia*, vol. 18, no. 9, pp. 1896–1909, 2016.
[87] K. R. Jerripothula, J. Cai, J. Lu, and J. Yuan, "Object co-skeletonization with co-segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3881–3889.
[88] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.
[89] M. Rubinstein, A. Joulin, J. Kopf, and C. Liu, "Unsupervised joint object discovery and segmentation in internet images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 1939–1946.
[90] A. Joulin, F. Bach, and J. Ponce, "Discriminative clustering for image co-segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 1943–1950.
[91] X. Chen, A. Shrivastava, and A. Gupta, "Enriching visual knowledge bases via object discovery and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 2027–2034.
[92] Z. Tao, H. Liu, H. Fu, and Y. Fu, "Image cosegmentation via saliency-guided constrained clustering with cosine similarity," in *AAAI Conference on Artificial Intelligence*, 2017, pp. 4285–4291.
[93] Z.-H. Yuan, T. Lu, and Y. Wu, "Deep-dense conditional random fields for object co-segmentation," in *Proc. Int. Join. Conf. Arti. Intell.*, 2017, pp. 3371–3377.
[94] K. Rakelly, E. Shelhamer, T. Darrell, A. Efros, and S. Levine, "Conditional networks for few-shot semantic segmentation," in *Proc. Int. Conf. Learn. Representations–Workshop*, 2018.
[95] L. Wiskott and T. J. Sejnowski, "Slow feature analysis: Unsupervised learning of invariances," *Neural Computation*, vol. 14, no. 4, pp. 715–770, 2002.
[96] D. Jayaraman and K. Grauman, "Slow and steady feature analysis: Higher order temporal coherence in video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3852–3861.

Xiankai Lu is a Research Professor in the School of Software, Shandong University. From 2018 to 2020, he was a research associate with Inception Institute of Artificial Intelligence, Abu Dhabi, UAE. His research interest includes object tracking and video object segmentation. He received the Ph.D. degree from Shanghai Jiao Tong University in 2018.

Wenguan Wang received his Ph.D. degree from Beijing Institute of Technology in 2018. He is currently a postdoc researcher at ETH Zurich, Switzerland. From 2016 to 2018, he was a joint Ph.D. candidate in University of California, Los Angeles. From 2018 to 2019, he was a senior scientist at Inception Institute of Artificial Intelligence, UAE. His current research interests include computer vision, image processing and deep learning.

Jianbing Shen (M'11-SM'12) is currently acting as the Lead Scientist with the Inception Institute of Artificial Intelligence, Abu Dhabi, United Arab Emirates.He has published about 100 journal and conference papers such as *IEEE TPAMI*, *CVPR*, and *ICCV*. He has obtained many honors including the Fok Ying Tung Education Foundation from Ministry of Education, the Program for Beijing Excellent Youth Talents from Beijing Municipal Education Commission, and the Program for New Century Excellent Talents from Ministry of Education. His research interests include computer vision and deep learning. He is an Associate Editor of *IEEE TIP*, *IEEE TNNLS* and other journals.

David Crandall is an Associate Professor in the School of Informatics and Computing, Indiana University. He received the Ph.D. degree from Cornell University in 2008, and the B.S. and M.S. degrees from Pennsylvania State University, State College in 2001. His research interests include computer vision, machine learning, and data mining. He is the recipient of a National Science Foundation CAREER Award and a Google Faculty Research Award. Currently, he is an Associate Editor of *IEEE TPAMI* and *IEEE TMM*.

Luc Van Gool received the degree in electromechanical engineering at the Katholieke Universiteit Leuven, in 1981. Currently, he is a professor at the Katholieke Universiteit Leuven in Belgium and the ETH in Zurich, Switzerland. He leads computer vision research at both places, and also teaches at both. He has been a program committee member of several major computer vision conferences. His main interests include 3D reconstruction and modeling, object recognition, tracking, and gesture analysis, and the combination of those. He received several Best Paper awards, won a David Marr Prize and a Koenderink Award, and was nominated Distinguished Researcher by the IEEE Computer Science committee. He is a co-founder of 10 spin-off companies. He is a member of the IEEE.