Occlusion-Aware Instance Segmentation via BiLayer Network Architectures

Lei Ke, Yu-Wing Tai, *Senior Member, IEEE,* and Chi-Keung Tang, *Fellow, IEEE*

Abstract—Segmenting highly-overlapping image objects is challenging, because there is typically no distinction between real object contours and occlusion boundaries on images. Unlike previous instance segmentation methods, we model image formation as a composition of two overlapping layers, and propose **B**ilayer **C**onvolutional **Net**work (**BCNet**), where the top layer detects occluding objects (occluders) and the bottom layer infers partially occluded instances (occludees). The explicit modeling of occlusion relationship with bilayer structure naturally decouples the boundaries of both the occluding and occluded instances, and considers the interaction between them during mask regression. We investigate the efficacy of bilayer structure using two popular convolutional network designs, namely, Fully Convolutional Network (FCN) and Graph Convolutional Network (GCN). Further, we formulate bilayer decoupling using the vision transformer (ViT), by representing instances in the image as separate learnable occluder and occludee queries. Large and consistent improvements using one/two-stage and query-based object detectors with various backbones and network layer choices validate the generalization ability of bilayer decoupling, as shown by extensive experiments on image instance segmentation benchmarks (COCO, KINS, COCOA) and video instance segmentation benchmarks (YTVIS, OVIS, BDD100K MOTS), especially for heavy occlusion cases. Code and data are available at https://github.com/lkeab/BCNet.

Index Terms—BCNet, Bilayer Decoupling, Occlusion-aware Instance Segmentation, Occlusion-aware Video Instance Segmentation.

1 Introduction

State-of-the-art approaches in instance segmentation often follow the Mask R-CNN [1] paradigm with the first stage detecting bounding boxes, followed by the second stage of segmenting instance masks. Mask R-CNN and its variants [2], [3], [4], [5], [6] have demonstrated notable performance, and most of the leading approaches in the COCO instance segmentation challenge [7] have adopted this pipeline. However, we note that most incremental improvement comes from better backbone architecture designs, with little attention paid in the instance mask regression after obtaining the ROI (Region-of-Interest) features from object detection. We observe that a lot of segmentation errors are caused by overlapping objects, especially for object instances belonging to the same class. This is because each instance mask is individually regressed, and the regression process implicitly assumes the object in an ROI has almost complete contour, since most objects in the training data in COCO do not exhibit significant occlusions.

We propose the Bilayer Convolutional Network (BCNet) with its core contribution illustrated in Figure 1. BCNet simultaneously regresses both occluding region (occluder) and partially occluded object (occludee) after ROI extraction, which groups the pixels belonging to the occluding region and treat them equally as the pixels of the occluded object but in *two separate image layers*, and thus naturally decouples the boundaries for both objects and considers the interaction between them during the mask regression stage.

Previous approaches resolve the mask conflict between neighboring objects through non-maximum suppression or additional post-processing [12], [13], [14], [15], [16]. Consequently, their results are over-smooth along boundaries or exhibit small gaps between neighboring objects. Furthermore, since the receptive field in the ROI observes multiple objects that belong to the same class, when the occluding regions were included as part of the occluded object, traditional mask head design falls short of resolving such conflict, leaving a large portion of error as shown in Figure 2. We compare BCNet with recent amodal segmentation methods [8], [9], which predict complete object masks, including the occluded region. However, these

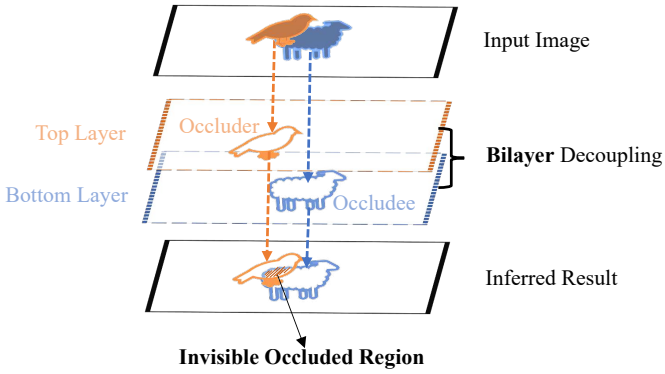


Fig. 1. **Simplified illustration on BCNet's key contribution**. Unlike previous segmentation approaches operating on a single image layer (i.e., directly on the input image), we decouple overlapping objects into *two image layers*, where the top layer deals with the occluding objects (**occluder**) and the bottom layer for **occludee** (which is also referred to as target object in other methods as they do not explicitly consider the occluder). The overlapping parts of the two image layers indicate the invisible region of the occludee, which is explicitly modeled by our occlusion-aware BCNet framework.

- *L. Ke and C.-K. Tang are with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology. E-mail: {lkeab, cktang}@cse.ust.hk.*
- *Y.-W. Tai is with Kuaishou Technology. E-mail: yuwing@gmail.com.*

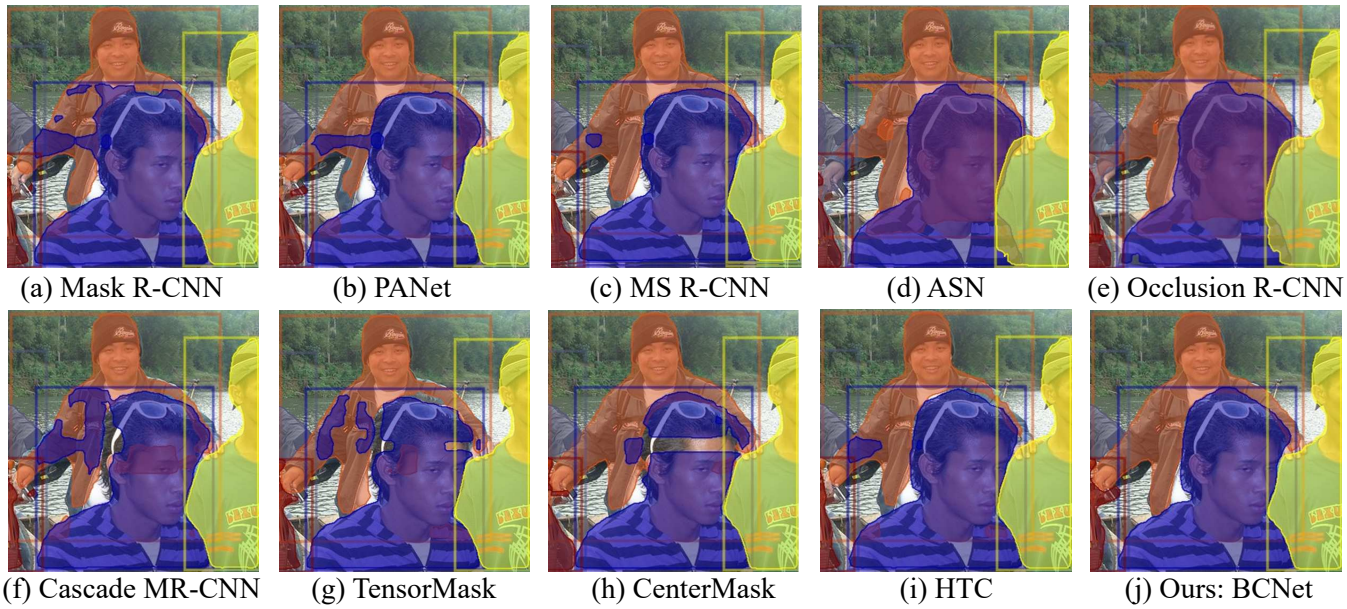


Fig. 2. Instance Segmentation on **COCO** [7] validation set by a) Mask R-CNN [1], b) PANet [2], c) Mask Scoring R-CNN [5], d) ASN [8], e) Occlusion R-CNN (ORCNN) [9], f) Cascade Mask R-CNN [3], g) TensorMask [10], h) CenterMask [11], i) HTC [6] and j) Our BCNet. Note that d) and e) are specially designed for amodal/occlusion mask prediction. In this example, the bounding box is given to compare the quality of different regressed instance masks.

amodal methods only regress single occluded target in the ROI, thus lacking occluder-occludee interaction reasoning, making their specially designed decoupling structure suffer when handling mask conflict between highly-overlapping objects. Correspondingly, Figure 3 compares the architecture of our BCNet with previous mask head designs [1], [2], [3], [5], [6], [8], [9], [11].

A preliminary version of BCNet appears in [17]. Our BCNet consists of two GCN layers with a cascaded structure, each respectively regresses the mask and boundaries of the occluding and partially occluded objects. We utilize GCN in our implementation because GCN can consider non-local relationship between pixels, allowing for propagating information across pixels despite the presence of occluding regions. The explicit bilayer occluder-occludee relational modeling within the same ROI also makes our final segmentation results more explainable than previous methods. We also experiment BCNet with pure FCN layers, and find that the bilayer structure still generalizes well, despite achieving inferior performance comparing to bilayer GCN. For object detector, we use the FCOS [18] owing to its efficient memory and running time, while noting that other state-of-the-art object detectors can also be used as demonstrated in our experiments.

Besides the aforementioned standard CNN and GCN architecture in the preliminary work [17], we further summarize the extensions as: 1) We implement BCNet using the emerging vision transformer (ViT) [19] for instance segmentation. 2) We perform extensive quantitative and qualitative analysis for the transformer-based BCNet, which achieves 44.6 mask AP on COCO by using R50-FPN. 3) We further apply BCNet to three complicated video instance segmentation benchmarks and obtain consistent improvement.

Our transformer-based BCNet explicitly decouples the instance queries by representing image objects into two individual groups, one representing the occluded objects (occludees), while the other for the corresponding occluding objects (occluders). Instead of using a single transformer decoder [20], we design a bilayer transformer decoder with a cascaded structure, where the first transformer decoder distills occluder information, which is then injected into the second transformer decoder for occludee mask prediction. In doing so, both instance queries and transformer decoders can perceive the occluder-occludee relations, contributing to the first occlusion-aware transformer structure.

Since our paper focuses on occlusion handling in instance segmentation, in addition to the original COCO evaluation, we extract a subset of COCO dataset containing both occluding objects and partially occluded objects to evaluate the robustness of our approach in comparison with other instance segmentation methods in occlusion handling. In this paper, we also contribute a large-scale occlusion-aware instance segmentation dataset SOD with ground-truth, complete object contours for *both* occluding and partially occluded objects. Extensive experiments show that our approach outperforms state-of-the-art methods in both the modal and amodal instance segmentation tasks.

2 Related Work

Image Instance Segmentation Two stage instance segmentation methods [1], [2], [3], [4], [6], [10], [22] achieve state-of-the-art performance by first detecting bounding boxes and then performing segmentation in each ROI region. FCIS [22] introduces the position-sensitive score maps within instance proposals for mask segmentation. Mask R-CNN [1] extends Faster R-CNN [23] with a FCN branch to segment objects in the detected box. PANet [2] further integrates multi-level feature of FPN to enhance feature representation. MS R-CNN [5] mitigates the misalignment between mask quality and score. CenterMask [11] is built upon the anchor free detector FCOS [18] with a SAG-Mask branch. In contrast, our BCNet is a *bilayer* mask prediction network for addressing the issues of heavy occlusion and overlapping objects

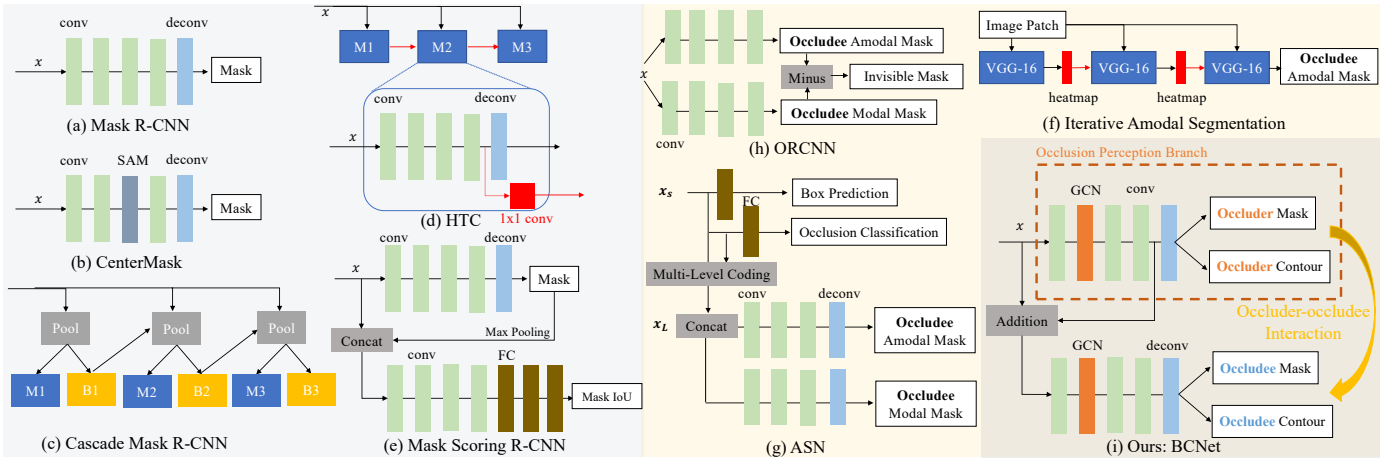


Fig. 3. A brief comparison of **mask head architectures**: a) Mask R-CNN [1], b) CenterMask [11], c) Cascade Mask R-CNN [3], d) HTC [6], e) Mask Scoring R-CNN [5], f) Iterative Amodal Segmentation [21], g) ASN [8], h) ORCNN [9], where f), g) and h) are specially designed for amodal/occlusion mask prediction, i) Ours: BCNet. The input $\mathbf{x}$ denotes CNN feature after ROI extraction. `Conv` is convolution layer with 3×3 kernel, `FC` is the fully connected layer, `SAM` is the spatial attention module. B_t and M_t respectively denote box and mask head at t-th stage. Unlike previous occlusion-aware mask heads, which only regress both modal and amodal masks from the occludee, our BCNet has a *bilayer GCN structure* and considers the **interactions between the top "occluder" and bottom "occludee"** in the same ROI. The **occlusion perception branch** explicitly models the occluding object by performing joint mask and contour predictions, and distills essential occlusion information for the second graph layer to segment target object ("occludee").

in two-stage instance segmentation. Experiments validate that our approach leads to significant performance gain on *overall* instance segmentation performance not limited to heavily occluded cases.

One-stage instance segmentation methods remove the bounding box detection and feature re-pooling steps. AdaptIS [24] produces masks for objects located on point proposals. PolarMask [25] models instance masks in polar coordinates by instance center classification and dense distance regression. YOLOACT [26] introduces prototype masks with per-instance coefficients. SOLO [27] applies the "instance categories" concept to directly output instance masks based on location and size. Grouping-based approaches [28], [29], [30], [31], [32], [33] regard segmentation as a bottom-up grouping task by first producing pixel-wise predictions followed by grouping object instances in the post-processing stage. There are also some GCN-based segmentation works [34], [35], [36], however, they mainly focus on the general human parsing and semantic segmentation tasks [37] without occlusion-aware modeling.

Transfomer-based Instance Segmentation Inspired by DETR [38], transformer-based instance segmentation methods [39], [40], [41], [42], [43] regard segmentation as set prediction. These methods represent the interested objects using instance queries, and jointly perform class, bounding box and mask predictions. QueryInst [39] adopts dynamic mask heads with mask information flow. Mask Transfiner [44], [45] produces high-quality instance segmentation by taking detected incoherent points as input queries and employing efficient quadtree transformer. Mask2Former [20] designs a masked cross-attention decoder to constrain the attention regions in [46], while [47] further boosts the query-based models by discriminative learning. Unlike these methods using a shared decoder, our transformer-based BCNet has a bilayer transformer structure with both occluder and occludee decoders. Each transformer decoder deals with the corresponding set of queries, and then communicates through a residue connection.

Occlusion Handling Methods for occlusion handling have been proposed [48], [49], [50], [50], [51], [52], [53], [54], [55]. Ghiasi *et al*. [56] model occlusion by learning deformable models for human pose estimation while [57] reconstructs dense 3D shape for vehicle pose. Tighe *et al*. [58] build a histogram to predict occlusion overlap scores between two classes for inferring occlusion order in the scene parsing task. Chen *et al*. [59] handle occlusion by incorporating category specific reasoning and exemplar-based shape prediction for instance segmentation. For pedestrian occlusion, bi-box regression is proposed in [60] for both full body and visible part estimation, while repulsion loss [61] and aggregation loss [62] are to improve the detection accuracy. SeGAN [63] learns occlusion patterns by segmenting and generating the invisible part of an object. OCFusion [64] uses an additional branch to model instances fusion process for replacing detection confidence in panoptic segmentation. A self-supervised scene de-occlusion method is proposed in [65] to complete the mask and content for the invisible object parts. VOIN [66] learns to inpaint the occluded video object using occlusion-aware shape and flow completion.

Compared to these methods, our BCNet tackles occlusion by explicitly modeling occlusion patterns in shape and appearance. This equips the segmentation model with strong occlusion perception and reasoning capability. Our bi-layer approach can be smoothly integrated into state-of-the-art segmentation framework for end-to-end training.

Amodal Instance Segmentation Different from traditional segmentation which only focuses on visible regions, amodal instance segmentation can predict the occluded parts of object instances. Li and Malik [21] first propose a method by extending [14], which iteratively enlarges the modal bounding box following the direction of high heatmap values and synthetically adds occlusion. Zhu *et al*. [54] propose a COCO amodal dataset with 5000 images from the original COCO and use AmodalMask as a baseline, which is SharpMask [67] trained on amodal ground truth. COCOA *cls* [9] augments this dataset by assigning class-labels to the objects

while SAIL-VOS dataset in [68] is targeted for video object segmentation. In autonomous driving, Qi *et al.* [8] establish the large-scale KITTI [69] InStance segmentation dataset (KINS) and present ASN to improve amodal segmentation performance.

Comparing to most of the amodal and occlusion reasoning methods which regress single occluded object boundary directly on the input (single-layered) image, our BCNet decouples overlapping objects in the same ROI into two disjoint graph layers by predicting the complete object segments (Figure 1), where the occludee is segmented under the guidance from the shape and location of the occluder.

3 Occlusion-Aware Instance Segmentation

We first describe the explicit occluder-occludee modeling of our proposed Bilayer Convolutional Network (BCNet) in Section 3.1, and then give an overview to the overall bilayer GCN-based instance segmentation framework in Section 3.2. Based on the principle of bilayer decoupling, we further design a bilayer transformer-based on Mask2Former [20] for occlusion-aware instance segmentation in Section 3.3. Finally, we specify the objective functions for the whole network optimization, and provide details of training and inference process.

BCNet is motivated by images with heavy occlusion, where multiple overlapping objects in the same bounding box may result in confusing instance contours from both real objects and occlusion boundaries. The mask head design of Mask R-CNN and its variants [3], [5], [6], [8], [9] in Figure 3 directly regresses the occludee with a fully convolutional network, which neglects both the occluding instances and the overlapping relations between objects. To mitigate this limitation, BCNet extends existing two stage instance segmentation methods, by adding an occlusion perception branch parallel to the traditional target prediction pipeline. Thus, the interactions between objects within the ROI region can be well considered during the mask regression stage.

To obtain occlusion relations among image objects, for amodal instance segmentation, such as KINS [8] and COCOA [54], ground truth for occluder and occludee is extracted from their annotated object depth/occlusion order. For conventional instance segmentation with no occlusion labeling, such as COCO [7], we simply regard the occludee as the target object inside the bounding box, while the occluder as the union of remaining objects inside the same bounding box with overlapping relation to the target object.

3.1 Bilayer Occluder-Occludee Modeling

Bilayer GCN Structure for Instance Segmentation Recently, Graph Convolutional Network (GCN) [73] has been adopted to model long-range relationships in images [74], [75], [76] and videos [72]. Given highly-overlapping objects, pixels belonging to the same partially occluded object may be separated into disjoint subregions by the occluder. Thus, we adopt GCN as our basic block due to its non-local property [71], where each graph node represents a single pixel on the feature map. To explicitly model the occluding region, we further extend the single GCN block to the bilayer GCN structure as shown in Figure 4, which constructs two orthogonal graphs in a single general framework.

Following [72], given an adjacency graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ with edges $\mathcal{E}$ among nodes $\mathcal{V}$, we represent the graph convolution operation as,

$$\mathbf{Z} = \sigma(\mathbf{A}\mathbf{X}\mathbf{W}_g) + \mathbf{X}, \tag{1}$$

where $\mathbf{X} \in \mathbb{R}^{N \times K}$ is the input feature, $N = H \times W$ is the number of pixel grids within the ROI region and K is the feature dimension for each node, $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix for defining neighboring relations of graph nodes by feature similarities, and $\mathbf{W}_g \in \mathbb{R}^{K \times K'}$ is the learnable weight matrix for the output transform, where $K' = K$ in our case. The output feature $\mathbf{Z} \in \mathbb{R}^{N \times K'}$ consists of the updated node feature by global information propagation within the whole graph layer, which is obtained after nonlinear functions $\sigma(\cdot)$ including layer normalization [77] and ReLU functions. We add a residual connection after the GCN layer.

To construct the adjacency matrix $\mathbf{A}$, we define the pairwise similarity between every two graph nodes $\mathbf{x}_i, \mathbf{x}_j$ by dot product similarity as,

$$\mathbf{A}_{ij} = softmax(F(\mathbf{x}_i, \mathbf{x}_j)), \tag{2}$$

$$F(\mathbf{x}_i, \mathbf{x}_j) = \theta(\mathbf{x}_i)^T \phi(\mathbf{x}_j), \tag{3}$$

where θ and ϕ are two trainable transformation function implemented by 1×1 convolution as shown in the non-local operator part of Figure 4, so that high confidence edge between two nodes corresponds to larger feature similarity.

In our bilayer GCN structure, we further define $\mathcal{G}^i$ to indicate the ith graph, X_{roi} for the input ROI feature and $\mathbf{W}_f$ for weights in FCN layers. The pertinent equations are:

$$\mathbf{Z}^1 = \sigma(\mathbf{A}^1 \mathbf{X}_f \mathbf{W}_g^1) + \mathbf{X}_f, \tag{4}$$

$$\mathbf{X}_f = \mathbf{Z}^0 \mathbf{W}_f^0 + \mathbf{X}_{roi}, \tag{5}$$

$$\mathbf{Z}^0 = \sigma(\mathbf{A}^0 \mathbf{X}_{roi} \mathbf{W}_g^0) + \mathbf{X}_{roi}. \tag{6}$$

For connecting the two GCN blocks, the output feature $\mathbf{Z}^0$ of the occluder from the first GCN is directly added to $\mathbf{X}_{roi}$ to obtain the fused *occlusion-aware* feature $\mathbf{X}_f$, which is the input for the second GCN layer to output $\mathbf{Z}^1$ for occludee mask prediction.

Compared to previous class-agnostic mask head with single layer structure, where there is only binary label (foreground/background) per pixel, the bilayer GCN additionally constructs a new semantic graph space for *occluding region*. Thus a pixel node in overlapping areas in ROI can concurrently correspond to two different states in bilayer graph. While other choices may exist, we believe modeling GCN as a dual-layered structure as shown in Figure 4 is a natural choice for handling occlusion.

Occluder-occludee Modeling We explicitly model occlusion patterns by detecting both contours and masks for the occluders using the first GCN layer. Since the second GCN layer jointly predicts contours for the occludee, the overlap between the two layers can be directly identified as occlusion boundary which can thus be distinguished from real object contour (e.g., the occluder and occludee prediction on the rightmost of Figure 4). The rationale behind this design is that such irregular occlusion boundary unrelated to the occludee is confusing, which in turn provides essential

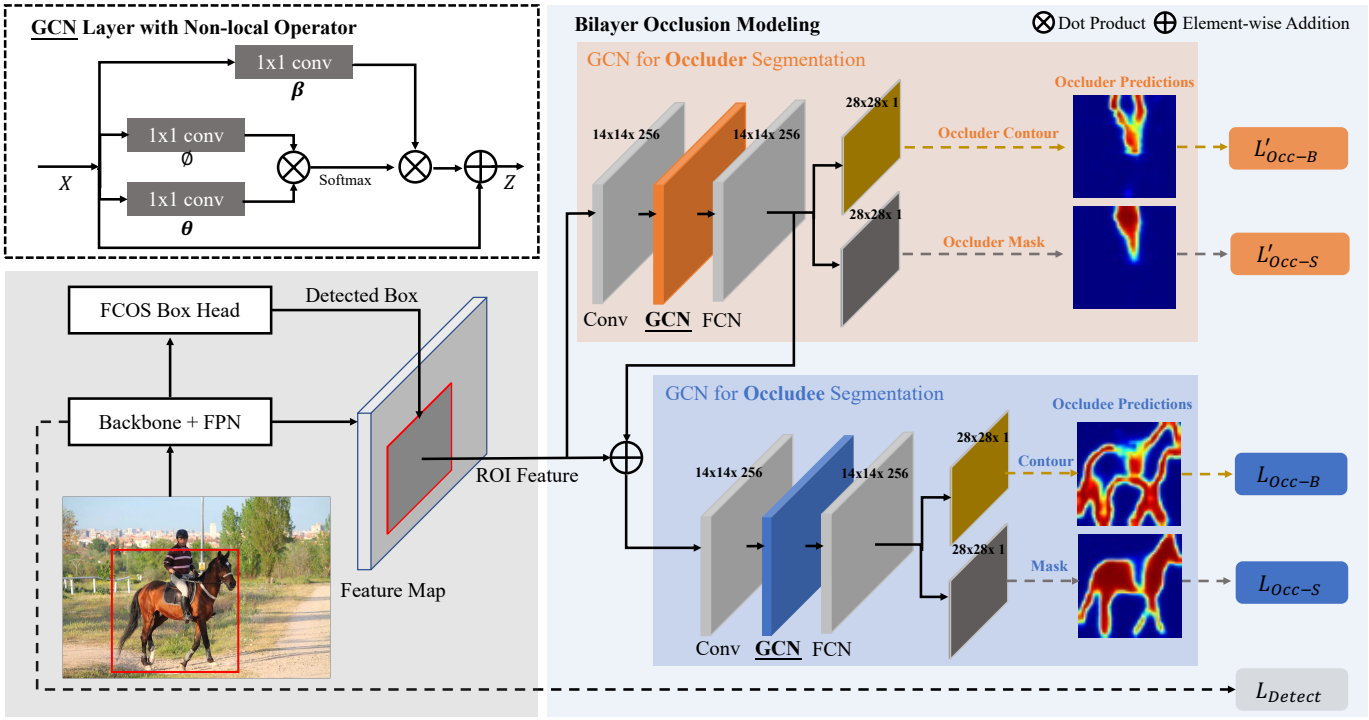


Fig. 4. Architecture of our BCNet for **GCN-based** Instance Segmentation with bilayer occluder-occludee relational modeling, which consists of three modules; (1) Backbone [70] with FPN for feature extraction from input image; (2) Detection branch [18] for predicting instance proposals; (3) BCNet with bilayer GCN structure for mask prediction. For cropped ROI feature, the first GCN explicitly models occluding regions (occluder) by simultaneously detecting occlusion contours and masks, which distills essential shape and position information to guide the second GCN in mask prediction for the occludee. We utilize the non-local operator [71], [72] detailed in Section 3.2 to implement the GCN layer. Visualization results are resized to squares.

cues for decoupling occlusion relations. Besides, accurate boundary localization explicitly contributes to segmentation mask prediction.

The module for occluder modeling is designed in a simple yet effective way: one 3×3 convolutional layer followed by one GCN layer and one FCN layer. Then we feed the output to the up-sampling layer and one 1×1 convolutional layer to obtain one channel feature map for joint boundary and mask predictions. The boundary detection for occluder is trained with loss $\mathcal{L}'_{\text{Occ-B}}$:

$$\mathcal{L}'_{\text{Occ-B}} = \mathcal{L}_{\text{BCE}}(W_B \mathcal{F}_{occ}(\mathbf{X}_{roi}), \mathcal{GT}_B), \tag{7}$$

where $\mathcal{L}_{\text{BCE}}$ denotes the binary cross-entropy loss, $\mathcal{F}_{occ}$ denotes the nonlinear transformation function of the occlusion modeling module, W_B is the boundary predictor weight, $\mathbf{X}_{roi}$ is the cropped FPN feature map given by RoIAlign operation for the target region, and $\mathcal{GT}_B$ is the off-the-shelf occluder boundary that can be readily computed from mask annotations.

For occluder mask prediction, it utilizes the shared feature $\mathcal{F}_{occ}(\mathbf{X}_{roi})$, which is jointly optimized by boundary prediction. The segmentation loss $\mathcal{L}'_{\text{Occ-S}}$ for occluder modeling is designed as

$$\mathcal{L}'_{\text{Occ-S}} = \mathcal{L}_{\text{BCE}}(W_S \mathcal{F}_{occ}(\mathbf{X}_{roi}), \mathcal{GT}_S), \tag{8}$$

where W_S denotes the trainable weight of segmentation mask predictor by 1×1 convolutional layer, and $\mathcal{GT}_S$ is the mask annotations for the occluder.

3.2 Bilayer GCN-based Instance Segmentation

Figure 4 gives the overall **architecture** of BCNet for addressing occlusion in instance segmentation. Following typical models [1], [11] for instance segmentation, our model has three parts: (1) Backbone [70] with FPN [78] for ROI feature extraction; (2) Object detection head in charge of predicting bounding boxes as instance proposals. We employ FCOS [18] as the object detector owing to its anchor-free efficiency though our method is flexible and can deploy any existing fully supervised object detectors [23], [79], [80]; (3) The *occlusion-aware mask head*, BCNet, uses bilayer GCN structure for decoupling overlapping relations and segments the instance proposals obtained from the object detection branch. BCNet reformulates the traditional class-agnostic segmentation as two complementary tasks: occluder modeling using the first GCN and occludee prediction with the second GCN, where the auxiliary predictions from the first GCN provide rich occlusion cues, such as shape and positions of occluding regions, to guide target (occludee) object segmentation.

Work Flow Given an input image, the backbone network equipped with FPN first extracts intermediate convolutional features for downstream processing. Then, the object detection head predicts bounding boxes with positions as well as categories for potential instances, and prepares the cropped ROI feature for BCNet to produce segmentation masks. The occlusion perception branch consists of the first GCN layer followed by FCN (two convolution layers), which is targeted for modeling occluding regions by jointly detecting contours and masks. Forming a residual connection, the distilled occlusion feature is element-wise added to the input ROI feature and passed to second GCN. Finally, the second GCN, which has a similar structure to the first GCN, segments the occludee guided by this occlusion-aware feature and out-

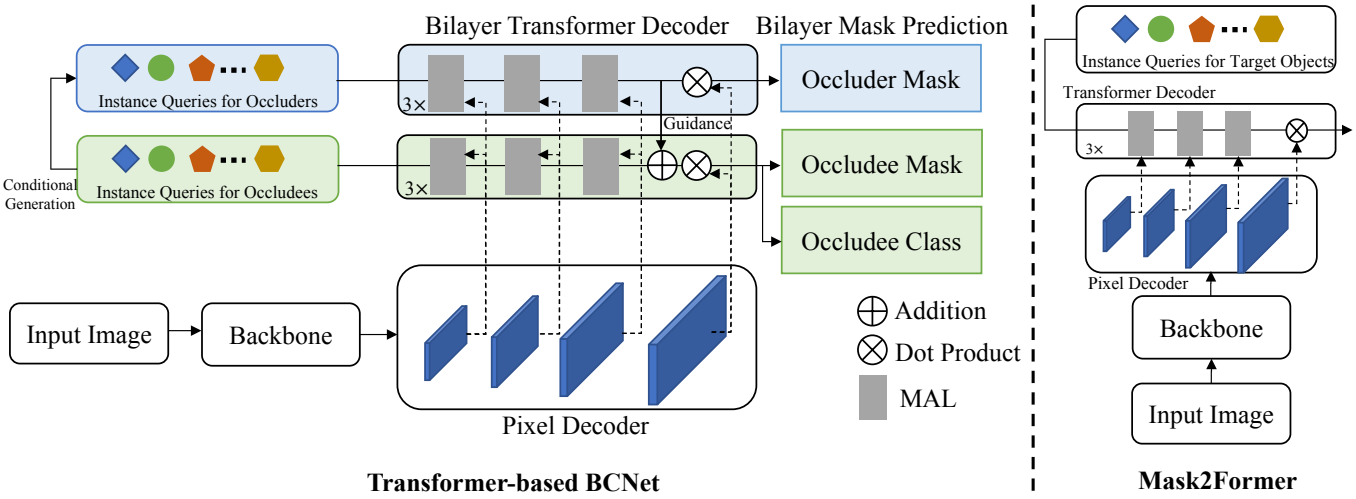


Fig. 5. **Left**: Architecture of our **transformer-based** BCNet built on [20] with bilayer transformer decoder. **Right**: Architecture of Mask2Former [20] for instance segmentation. Instead of adopting a single transformer decoder and only one set of instance queries, our bilayer transformer decoder models occluder-occludee relations by processing occluder and occludee queries in a cascaded manner. In the latter stage of the first transformer decoder, the learned shape and texture information of the occluder is injected to the second decoder to guide the target instance (occludee) segmentation by residue connection. MAL denotes the Masked cross-Attention Layer in [20]. Pixel decoder constructs a multi-scale feature pyramid from the original image for feeding into the transformer decoder.

puts contours and masks for the partially occluded instance.

3.3 Bilayer Transformer-based Instance Segmentation

Driven by the powerful object detection paradigms of DETR [38], transformer-based instance segmentation methods [20], [39], [40], [44] show ever increasing performance on COCO. While these methods excels in object bounding box detection, the problem of accurately delineating each distinct object from heavy occlusions remains elusive.

We build our transformer-based BCNet based on Mask2Former [20] owing to its simple and effective architecture. In Figure 5, comparing to [20] (right part), we explicitly divide the learnable instance queries into occluder and occludee sets respectively. To separately model occluder and occludee information in the image, our Bilayer Transformer decoder consists of two cascaded transformer decoders, instead of using a shared one with single query group to only focus the target object (occludee).

Instance Queries for Occluders and Occludees Transformer-based BCNet first initializes the instance queries of occludees as learnable positional embeddings. Then, to construct the occluder-occludee query pair for each image object, BCNet produces the same number of instance queries for occluders conditioned on their corresponding occludee queries. The conditional generation is based on a two-layer MLP, taking as input the query embeddings of occludees. In case of multiple occluders for an object (occludee), the occluder query group represents their grouped occlusion regions. To avoid matching conflicts, we copy bipartite matching between the occludee queries and ground truth, and then directly assign the matching correspondence to the occluder queries.

Bilayer Transformer Decoder Instead of solely separating input queries as occluders and occludees, comparing to conventional transformer, our Bilayer Transformer Decoder is composed of two decoders in a cascaded structure. In Figure 5, the first transformer decoder takes the instance queries of the occluders as input and predicts their object masks. Guided by occluder information from the first decoder, the second transformer decoder takes the occludee instance queries, and regresses the object masks for the target objects (occludee). The bilayer decoder design prevents intervention between two sets of instance queries during the self-attention between input queries. Thus, the occluder query of one instance does not need to attend to the queries from the occludee set. However, similar to GCN-based BCNet, the overlapping information flows from the occluder decoder to occludee decoder by a residual connection. We validate the benefit of our bilayer transformer decoder design and occlusion-aware guidance in experimental section.

3.4 End-to-end Parameter Learning

The whole instance segmentation framework can be trained in an end-to-end manner defined by a multi-task loss function $\mathcal{L}$ as,

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{Detect}} + \mathcal{L}_{\text{Occluder}} + \mathcal{L}_{\text{Occludee}}, \tag{9}$$

$$\mathcal{L}_{\text{Occluder}} = \lambda_2 \mathcal{L}'_{\text{Occ-B}} + \lambda_3 \mathcal{L}'_{\text{Occ-S}} \tag{10}$$

$$\mathcal{L}_{\text{Occludee}} = \lambda_4 \mathcal{L}_{\text{Occ-B}} + \lambda_5 \mathcal{L}_{\text{Occ-S}}, \tag{11}$$

where $\mathcal{L}_{\text{Occ-B}}$ and $\mathcal{L}_{\text{Occ-S}}$ denote respectively the boundary detection and mask segmentation losses in the second GCN layer for the occludee, which are similar to Eq. 7 and Eq. 8. $\mathcal{L}_{\text{Detect}}$ supervises both the position prediction and the category classification borrowed from the FCOS [18] detector,

$$\mathcal{L}_{\text{Detect}} = \mathcal{L}_{\text{Regression}} + \mathcal{L}_{\text{Centerness}} + \mathcal{L}_{\text{Class}}, \tag{12}$$

and λ_1, λ_2, λ_3, λ_4 and λ_5 are hyper-parameter weights to balance the loss functions, which are tuned to be $\{1, 0.5, 0.25, 0.5, 1.0\}$ respectively on the validation set. For transformer-based BCNet, the $\mathcal{L}_{\text{Detect}}$ is adapted to,

$$\mathcal{L}_{\text{Detect}} = \mathcal{L}_{\text{Box}} + \mathcal{L}_{\text{Matching}} + \mathcal{L}_{\text{Class}}, \tag{13}$$

where $\mathcal{L}_{\text{Matching}}$ denotes bipartite matching loss between predicted and ground truth objects, and $\mathcal{L}_{\text{Box}}$ is bounding

box regression loss using weighted combination of L1 loss and IoU loss following [38].

Training: During training, following Mask R-CNN [1], GCN-based BCNet only samples RPN proposals with both highest IoU (at least larger than 0.5) to the GT boxes and high classification confidence for mask head training. On COCO, for each sampled proposal box, its occludee is simplified as the target object belonging to its best matched GT box. For training the first GCN layer of BCNet, since partial occlusion cases only occupy a small fraction compared to the complete objects in COCO, we filter out part of the non-occluded ROI proposals to keep occlusion cases taking up 50% for balance sampling. SGD with momentum is employed for training 90K iterations which starts with 1K constant warm-up iterations. The batch size is set to 16 and initial learning rate is 0.01. In ablation study, ResNet-50-FPN [70] is used as backbone and the input images are resized without changing the aspect ratio by keeping the shorter side and longer side of no more than 600 and 900 pixels respectively. For leaderboard comparison, we adopt the scale-jitter where the shorter image side is randomly sampled from [640, 800] following 3× schedule in [10], [11], [26]. For the transformer-based BCNet, we follow the same training schedules and setting in [20], where we train the model for 50 epochs with a batch size of 16 and large-scale jittering [81]. For fair comparison, transformer-based BCNet follows the same segmentation loss in Mask2Former *without* boundary detection mentioned in Eq. 10 and Eq. 11, increasing the training time of Mask2Former by 20%. Since there are no RoI proposals in Mask2Former, we adopt the complete GT mask annotation to determine the occluder pixels, i.e., the union/grouping of objects spatially neighboring to the target occludee. We take 100 instance queries per image for occluders and occludees respectively.

Inference: During inference, the mask head in GCN-based BCNet predicts masks for the occluded target object in the high-score box proposals (no more than 50) generated by the FCOS detector, where the first GCN layer only produces occlusion-aware feature as input for the second GCN.

4 Synthetic Occlusion Dataset

In this section, we provide details about the proposed Synthetic Occlusion Dataset (SOD) for instance segmentation. SOD facilitates occluded objects understanding.

Occlusion Synthesis Process As shown in Figure 6, to diversify the occlusion patterns, we construct the large-scale Synthetic Occlusion Dataset (SOD) by sampling both occluding and occluded instances from the Complete Object Bank (COB) following uniform class distribution. COB consists of images for *non-occluded single* object with corresponding complete mask and contour annotation, which has 80 categories with total instances number over 60,000. Then, a synthetic image based on the original image corresponding to the occluded target is produced by placing the occluding instance at a random image position (generated by grid search) which satisfies the object overlapping rate between 0.2 to 0.5. The synthetic occlusion dataset contains 100K such occluded images with **amodal** contours/masks for *both* occluding and partially occluded objects. We show the benefit of additionally training BCNet on SOD in Table 8.

5 Experiments

5.1 Experimental Setup

COCO and COCO-OCC We conduct experiments on COCO dataset [7], where we train on 2017*train* (115k images) and evaluate results on both 2017*val* and 2017*test-dev* using the standard metrics. For further investigating segmentation performance with occlusion handling, we propose a subset split, called COCO-OCC, which contains 1,005 images extracted from the validation set (5k images) where the overlapping ratio between the bounding boxes of objects is at least 0.2. Segmenting COCO-OCC with highly overlapping objects is much more difficult than 2017*val*, where we observe a performance gap around 3.0AP for the same model in the experiment section. Besides, we also validate the synthetic SOD dataset on COCO-OCC.

KINS and COCOA We also evaluate BCNet on two amodal instance segmentation benchmarks: (1) **KINS** [8], built on the original KITTI [69], is the largest amodal segmentation benchmark for traffic scenes with both annotated amodal and modal masks for instances. BCNet is trained on the training split (7,474 images and 95,311 instances) and tested on the testing split (7,517 images and 92,492 instances) following the setting in [8]. (2) **COCOA** [54] is a subpart of COCO [7], where we train BCNet on the official training split (2,500 images) and test on the validation split (1,323 images). Note that each instance has no class label and we only use the modal and amodal mask labels for COCOA.

Youtube-VIS, OVIS and BDD100K MOTS We further evaluate the GCN-based BCNet on three large VIS/MOTS benchmarks: 1) **YTVIS** [82] is a Video Instance Segmentation (VIS) benchmark, which contains 2,883 videos with 131k annotated object instances of 40 categories. We also report the results of BCNet on OVIS [83], a new VIS dataset on occlusion learning; 2) **OVIS** has 607, 140 and 154 videos for training, validation and test respectively. To evaluate BCNet in video instance segmentation, we only replace the frame-level mask head of Mask Track R-CNN [82] and CMTrack RCNN [83] while leaving the other model components unchanged; 3) **BDD100K MOTS** [84] is a large-scale Multiple Object Tracking and Segmentation (MOTS) dataset of BDD100K [84], which includes 154 videos (30,817 images) for training, 32 videos (6,475 images) for validation, and 37 videos (7,484 images) for testing. We integrate the mask head of BCNet into PCAN [85] and adopt the well-established MOTS metrics [86] for results comparison. BDD100K covers the self-driving scenario while YTVIS and OVIS have more diverse object categories.

5.2 Ablation Study

Effect of Explicit Occlusion Modeling We validate the efficacy of different components proposed for explicit occlusion modeling on the first GCN layer. Table 1 tabulates the quantitative comparison: 1) Baseline: BCNet with no explicit occlusion modeling targets; 2) modeling segmentation masks for occluding regions (**occluder**); 3) modeling contours of the occluding regions; 4) **joint** occlusion modeling on both masks and contours. Compared to the baseline, joint occlusion modeling produces the most obvious improvement especially for the heavy occlusion cases, which promotes mask AP on the standard validation set from 32.65 to 33.43,

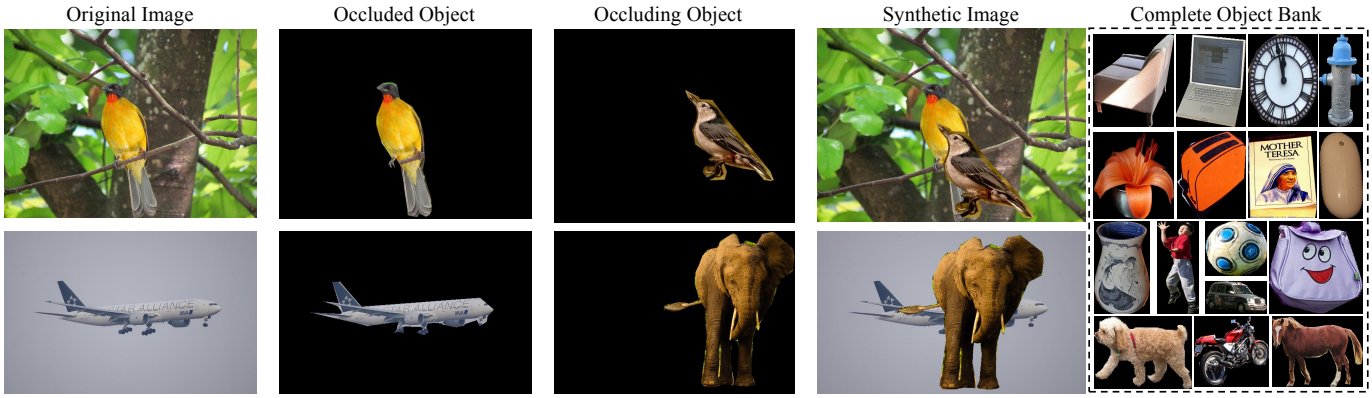


Fig. 6. Occlusion synthesis for producing *Synthetic Occlusion Dataset (SOD)* by sampling both occluding and occluded instances from the collected Complete Object Bank (COB), followed by grid searching the occluded positions in the image. COB from COCO is produced by conditionally filtering out the objects with bounding boxes overlapping rate over 5% and mask area smaller than 32×32, followed by manual selection.

and the AP on the proposed COCO-OCC split is increased from 29.04 to 30.37.

TABLE 1
Effect of the first GCN for occlusion modeling by predicting contours and masks on COCO with ResNet-50-FPN model.

Occlusion (Occluder) Modeling		COCO-OCC		COCO	
Contour	Mask	AP	AP_{50}	AP	AP_{50}
		29.04	49.22	32.65	52.39
	✓	29.65	49.42	33.25	52.82
✓		30.18	49.94	33.41	53.02
✓	✓	**30.37**	**50.40**	**33.43**	**53.12**

Effect of Bilayer Occluder-occludee Modeling Built on the first GCN layer with explicit occlusion modeling, we further validate the second GCN layer in Table 2, which demonstrates the importance of *occlusion-aware* feature *guidance* for the second GCN layer to segment target object (**occludee**) by boosting 1.23 AP on COCO-OCC, and 1.06 AP on COCO respectively. Table 3 shows the results comparison on adopting the proposed *bilayer structure* and existing direct regression model with single layer. On the COCO-OCC split, bilayer GCN improves AP from 29.63 to 30.68 compared to single GCN, and bilayer FCN boosts the performance of single FCN from 28.43 to 30.12.

TABLE 2
Effect of the second GCN for detecting occludee contours for final mask prediction ***guided*** by the output of first GCN.

Target (Occludee) Modeling			COCO-OCC		COCO	
Guidance	Contour	Mask	AP	AP_{50}	AP	AP_{50}
		✓	29.45	49.73	32.56	52.21
✓		✓	30.37	50.40	33.43	53.12
✓	✓	✓	**30.68**	**50.62**	**33.62**	**53.26**

Using FCN or GCN? Table 3 also reveals the advantage of GCN over FCN, where GCN achieves consistent superior performance both in the singe layer and bilayer structure. We also compute parameters number of each model and find that although GCN has more trainable parameters, the increased model size is acceptable compared to performance gain, because the feature size of input ROI has been downsampled to only 14×14 (spatial size) with 256 channels.

Effect of Bilayer Transformer Decoder Table 5 tabulates the effect of our transformer-based BCNet with Bilayer Transformer Decoder. Compared to the standard shared transformer decoder [20] with single set of instance queries (200), our bilayer transformer decoder training for 36 epochs with both occluder and occludee queries respectively improves 1.50 AP on COCO-OCC, and 1.01 AP on COCO. By further injecting the occlusion-aware guidance from the first transformer decoder to the second decoder, the mask AP can respectively be boosted from 40.17 to 41.23 on COCO-OCC,

TABLE 3
Effect of **bilayer structure** using **GCN vs. FCN** implementation.

Structure	FCN	GCN	COCO-OCC		COCO		Params
			AP	AP_{50}	AP	AP_{50}	
Single Layer	✓		28.43	48.24	33.01	52.62	51.0M
		✓	29.63	49.59	33.14	52.81	51.4M
Bilayer	✓		30.12	49.04	33.16	52.80	53.4M
		✓	**30.68**	**50.62**	**33.62**	**53.26**	54.0M

TABLE 4
Influence of the object detector (FCOS vs. Faster R-CNN vs. Query-based detector [20]) on BCNet.

Model	COCO-OCC		COCO		Params
	AP	AP_{50}	AP	AP_{50}	
FCOS [11] + Baseline	28.43	48.24	33.01	52.62	51.0M
FCOS [18] + Ours	**30.68**	**50.62**	**33.62**	**53.26**	54.0M
Faster R-CNN [1] + Baseline	29.67	49.95	33.45	53.70	60.0M
Faster R-CNN [23] + Ours	**31.71**	**51.15**	**34.61**	**54.41**	63.2M
Query-based Detector [20] + Baseline	39.23	50.62	41.13	62.50	81.6M
Query-based Detector [20] + Ours	**41.67**	**52.03**	**42.51**	**64.23**	89.7M

Fig. 7. Qualitative results comparison of the **amodal** mask predictions on **COCOA** [54] by AmodalMRCNN [9], ORCNN [9] and our method using ResNet-50, where BCNet hallucinates a more reasonable shape for the baby carriage without producing a large portion of segmentation error. We remove the "stuff" background for more clarity.

TABLE 5
Effect of the **Bilayer Transformer Decoder** for the transformer-based BCNet.

Transformer-based BCNet				COCO-OCC		COCO		#params.	FLOPs	fps
Shared decoder (100Q)	Shared decoder (200Q)	Bi-decoder (200Q)	Occlusion-guidance	AP	AP_{50}	AP	AP_{50}			
✓				38.67	58.73	41.51	61.73	44.0M	226G	8.6
	✓			39.01	59.90	41.82	62.14	44.0M	356G	8.2
		✓		40.17	61.20	42.62	63.02	53.8M	361G	8.0
		✓	✓	**41.23**	**62.12**	**43.21**	**64.21**	53.8M	362G	8.0

TABLE 6
Results on the COCOA dataset.

Model	AP_{all}	AP_t	AP_s
AmodalMask [54]	5.7	5.9	0.8
AmodalMRCNN [9]	21.51	21.09	9.0
ORCNN [9]	20.32	20.63	7.8
BCNet	**23.09**	**22.72**	**9.53**

TABLE 7
Results on the KINS dataset.

Model	AP_{Det}	AP_{Seg}
Mask R-CNN [9]	26.97	24.93
Mask R-CNN + ASN [8]	27.86	25.62
PANet [2]	27.39	25.99
PANet + ASN [8]	28.41	26.81
BCNet	**28.87**	**27.30**

TABLE 8
Results on COCO-OCC split.

Model	AP	AP_{50}
Mask R-CNN [70]	29.67	49.95
CenterMask [11]	29.05	49.07
MS R-CNN [5]	30.32	50.01
Ours	31.71	51.15
Ours + SOD	**32.89**	**53.25**

TABLE 9
Results on the OCHuman [87] *val* using R50-FPN.

Method	AP	AP_M	AR_L
Mask R-CNN [1]	16.3	19.4	11.3
BCNet	**20.6**	**23.3**	**13.8**

and from 42.62 to 43.21 on COCO validation set.

Influence of Object Detector To investigate the influence of object detectors to BCNet, besides using one-stage detector FCOS [18], we also use representative two-stage and query-based detectors Faster R-CNN [23] to perform experiments. As shown in Table 4, the performance gain brought by BCNet is consistent, with an improvement of 2.23 (for FCOS), 2.04 (for Faster R-CNN) mask AP on COCO-OCC respectively. The query-based BCNet improves 1.38 mask AP on COCO, and 2.44 mask AP on COCO-OCC. Note the baseline in one/two-stage detector denotes mask head design in Mask R-CNN, while the baseline in query-based detector denotes the mask head design of Mask2Former.

5.3 Performance Comparison and Analysis

Comparison with Amodal Segmentation Methods Table 6 and Table 7 compare BCNet with other SOTA amodal segmentation methods on both the COCOA [54] and KINS [8] datasets, where: 1) AmodalMask [54] directly predicts amodal masks from image patches; 2) Occlusion RCNN (ORCNN) [9] is an extension of Mask R-CNN with both amodal and modal mask heads; 3) ASN module [8] contains additional occlusion classification branch and multi-level coding. Compared to these occlusion handling approaches, our bilayer GCN with cascaded structure still performs favorably against the state-of-the-art methods, which shows the effectiveness of BCNet in decoupling overlapping objects and mask completion under the amodal segmentation setting. Figure 7 and Figure 9 show the qualitative comparison on COCOA and KINS respectively.

Evaluation on Occluded Images We adopt COCO-OCC split to compare the occlusion handling ability of BCNet with other methods on images with highly overlapping objects. As shown in Table 8, our BCNet with Faster R-CNN detector has 31.71 AP *vs.* 30.32 for the Mask Scoring R-CNN [5]. By further training BCNet on the synthetic occlusion dataset (SOD), the performance of AP and AP_{50} is significantly promoted to 32.89 and 53.25 respectively, which shows the advantage brought by this new dataset.

We also evaluate GCN-based BCNet on OCHuman [87]. The mask AP for Mask R-CNN (baseline) is 16.3. Although not specifically designed for handling human occlusions, our BCNet reaches 20.6 mask AP without any keypoint/pose usage, achieving large 4.3 mask AP improvement.

Comparison with SOTA Methods Table 10 compares BCNet with state-of-the-art instance segmentation methods on COCO dataset. BCNet achieves consistent improvement on different backbones and object detectors, demonstrating its effectiveness by outperforming both PANet [2] and Mask Scoring R-CNN [5] by 1.5 mask AP using two-stage detector Faster R-CNN, exceeding CenterMask [11] by 1.3 AP using one-stage detector FCOS, improving Mask2Former [20] by 0.9 AP using **query-based** detector. Our single two-stage based model achieves comparable result with HTC [6], which uses a 3-stage cascade refinement with multiple object detectors and mask heads, and far more parameters. Without bells and whistles, our transformer-based BCNet achieves 44.6 mask AP only using R50-FPN as backbone.

Qualitative Evaluation on COCO. Figure 8 shows qualitative comparison of CenterMask [11] and BCNet on images with overlapping objects using **FCOS detector**. In each ROI region, GCN-1 detects occluding regions while GCN-2 models the partially occluded instance by directly regressing the contours and masks. For example, BCNet decouples the occluding and occluded baseball players in similar clothes into GCN-1 and GCN-2 respectively, and detects the left leg missed by CenterMask. We also provide more qualitative results of our GCN-based BCNet compared to the Mask Scoring R-CNN [5] on COCO *test-dev* set are shown in Figure 10, both using ResNet-101-FPN and Faster R-CNN detector [23]. Our proposed method is robust enough to deal with various occlusion cases, such as highly overlapping zebras and human hands. The contour and mask predictions by the two GCN layers for the occluder (GCN-1) and occludee (GCN-2) in the same ROI region also makes BCNet more explainable compared to previous methods. In Figure 11, we further show qualitative results comparison of transformer-based BCNet with single and bilayer transformer decoder, where our BCNet can even handle well the highly occluded giraffe and motorcycle.

Amodal results comparison on KINS In Figure 9, we additionally provide qualitative **amodal** segmentation results comparison between Mask R-CNN + ASN module [8] and

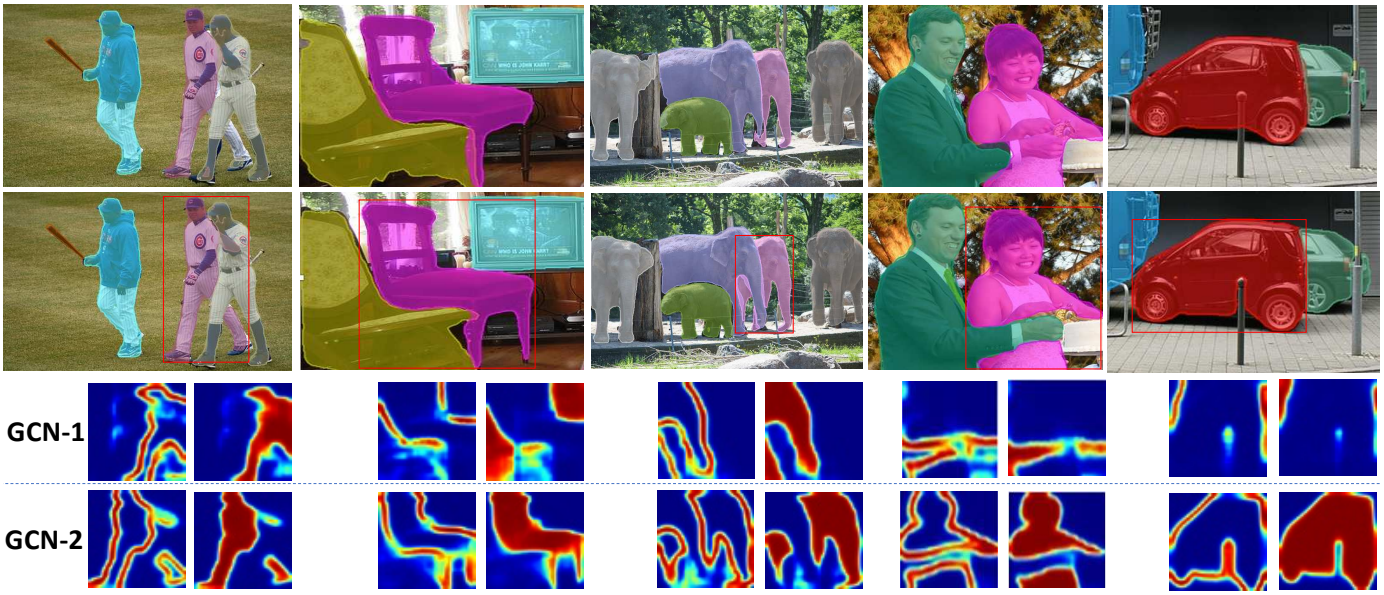


Fig. 8. Qualitative instance segmentation results of CenterMask [11] (top row) and our BCNet (middle row) on **COCO** [7], both using ResNet-101-FPN and **FCOS detector** [18]. The bottom row visualizes squared heatmap of contour and mask predictions by the two GCN layers for the occluder and occludee in the same **ROI region** specified by the red bounding box, which also makes the final segmentation result of BCNet more explainable than previous methods.

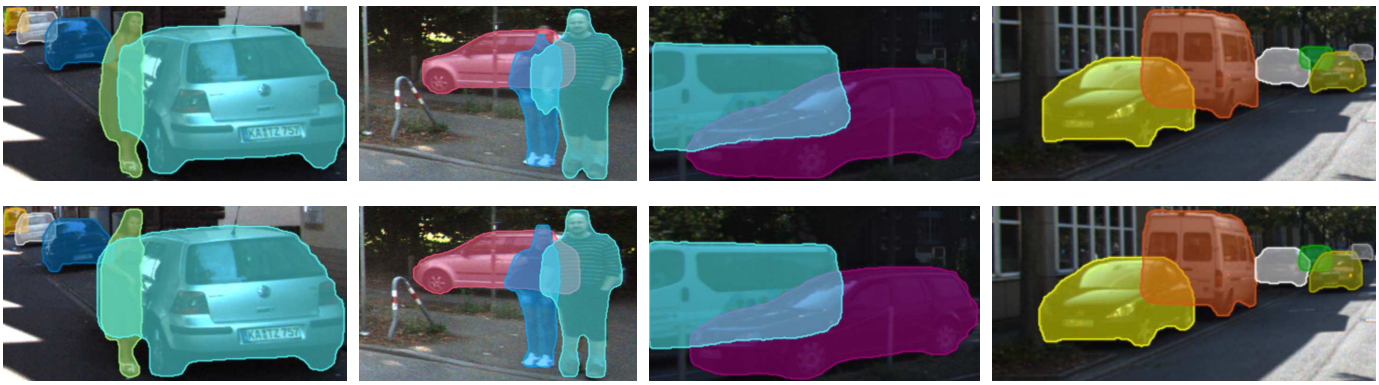


Fig. 9. Qualitative **amodal** results comparison between Mask R-CNN + ASN module [8] (top row) and our BCNet (bottom row) for the mask predictions on **KINS** test set [8], both using ResNet-101-FPN and Faster R-CNN detector [23], where the mask shape of the **invisible/occluded regions** are more reasonably estimated by BCNet.

our BCNet on KINS [8] test set. Take the first case as an example, our BCNet infers more reasonable amodal car shape even when the front part of the car is heavily occluded by the standing woman.

Evaluation on Video Instance Segmentation For experiments on YTVIS, we replace the mask head of Mask Track R-CNN with our GCN-based BCNet. The results in Table 11 show an improvement of 2.1 AP. We also show one challenging qualitative results comparison in Figure 12, where the overlapping regions between the two tandem skydivers are much better segmented by BCNet. For experiments on OVIS in Table 12, we adopt CMTrack RCNN [83] as the baseline, where BCNet achieves significant performance boost from 15.4 to 17.1, showing its efficacy of handling heavy occlusion in videos. Note that BCNet does not utilize temporal information while OVIS is a challenging video instance segmentation benchmark specifically designed to contain occluded video objects.

Evaluation on Multiple Object Tracking and Segmentation For experiments on BDD100K MOTS, we augment the mask head of PCAN [85] with our GCN-based BCNet in Table 13, where MOTSA measures segmentation as well as tracking quality, and ID Switches measure consistency in object identity. The quantitative results reveal an mAP advantage of 1.4 points, and mMOTSA gain over 1.0 points. The end-to-end training with new mask head also brings down ID Switches by 6% due to the improved instance features for association. The advancements demonstrate that our bilayer structure also generalizes to autonomous driving vehicles by providing more accurate segmentation masks.

Limitation and Future Work Although achieving large and consistent performance gain, we identify three design limitations for BCNet: 1) When dealing with unknown occluding objects of novel classes, the first GCN layer (transformer decoder) for detecting occluding objects may provide inaccurate occluder information for the second GCN layer (transformer decoder) to predict final occludee masks. This may cause BCNet to reduce to conventional instance segmentation models, outputting masks covering both the occluders and the occludee. For handling novel

TABLE 10
Comparison with SOTA methods on COCO *test-dev* set. Mask AP is reported and all entries are single-model results. Note that HTC [6] adopts 3-stage cascade refinement with multiple object detectors and mask heads. All of the methods are trained on COCO *train2017*.

Method	Backbone	Type	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Mask R-CNN [1]	R50-FPN	Two-stage	35.6	57.6	38.1	18.7	38.3	46.6
PANet [2]	R50-FPN	Two-stage	36.6	58.0	39.3	16.3	38.1	**52.4**
BCNet + Faster R-CNN [23]	R50-FPN	Two-stage	**38.4**	**59.6**	**41.5**	**21.9**	**40.9**	49.3
Mask R-CNN [1]	R101-FPN	Two-stage	37.0	59.2	39.5	17.1	39.3	52.9
MaskLab [4]	R101-FPN	Two-stage	37.3	59.8	39.6	19.1	40.5	50.6
Mask Scoring R-CNN [5]	R101-FPN	Two-stage	38.3	58.8	41.5	17.8	40.4	54.4
BMask R-CNN [88]	R101-FPN	Two-stage	37.7	59.3	40.6	16.8	39.9	**54.6**
HTC [6]	R101-FPN	Two-stage	39.7	**61.8**	43.1	21.0	42.2	53.5
BCNet + Faster R-CNN [23]	R101-FPN	Two-stage	**39.8**	61.5	**43.1**	**22.7**	**42.4**	51.1
YOLACT [26]	R101-FPN	One-stage	31.2	50.6	32.8	12.1	33.3	47.1
TensorMask [10]	R101-FPN	One-stage	37.1	59.3	39.4	17.4	39.1	51.6
ShapeMask [89]	R101-FPN	One-stage	37.4	58.1	40.0	16.1	40.1	53.8
CenterMask [11]	R101-FPN	One-stage	38.3	-	-	17.7	40.8	**54.5**
BlendMask [90]	R101-FPN	One-stage	38.4	60.7	41.3	18.2	41.5	53.3
BCNet + FCOS [18]	R101-FPN	One-stage	**39.6**	**61.2**	**42.7**	**22.3**	**42.3**	51.0
ISTR [43]	R50-FPN	Query-based	38.6	-	-	22.1	40.4	50.6
QueryInst [39]	R50-FPN	Query-based	39.9	62.2	43.0	22.9	41.7	51.9
SOLQ [40]	R50-FPN	Query-based	39.7	-	-	21.5	42.5	53.1
Mask Transfiner [44]	R50-FPN	Query-based	41.6	63.9	45.5	24.2	44.6	55.2
Mask2Former [20]	R50-FPN	Query-based	43.6	66.5	47.9	23.5	47.4	64.1
Transformer-based BCNet	R50-FPN	Query-based	**44.6**	**68.1**	**48.7**	**24.1**	**47.7**	**66.7**

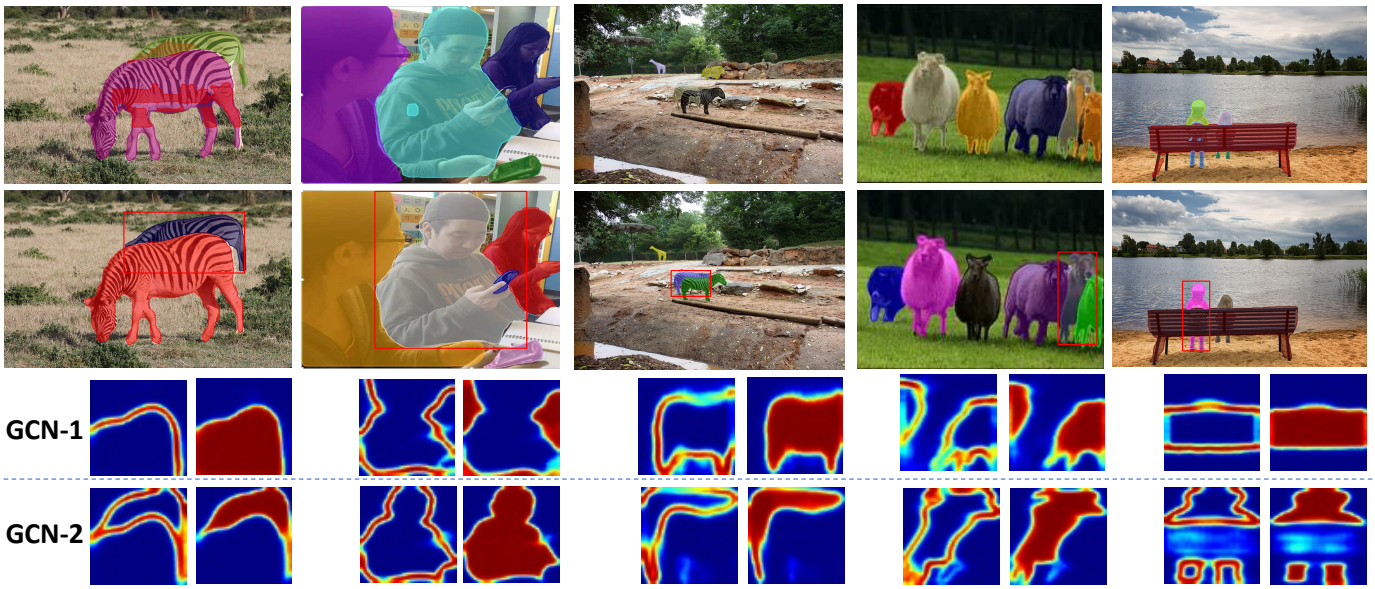


Fig. 10. Qualitative results of Mask Scoring R-CNN [5] (top row) and our BCNet (middle row) on **COCO** *test-dev* set, both using ResNet-101-FPN and **Faster R-CNN [23]**. The bottom row visualizes squared heatmap of contour and mask predictions by the two GCN layers for the occluder and occludee in the same **ROI region** specified by the red bounding box, which also makes the final segmentation result of BCNet more explainable than previous methods.

TABLE 11
State-of-the-art comparison of BCNet built on Mask Track R-CNN [82] on the YouTube-VIS validation set, using ResNet-50 as backbone. Results are reported in terms of mask accuracy (AP) and recall (AR).

Method	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
OSMN [91]	23.4	36.5	25.7	28.9	31.1
FEELVOS [92]	26.9	42.0	29.7	29.9	33.4
DeepSORT [93]	26.1	42.9	26.1	27.8	31.3
MaskTrack R-CNN [82]	30.3	51.1	32.6	31.0	35.5
MaskTrack R-CNN [82] + **BCNet**	**32.4**	**53.9**	**34.0**	**33.9**	**39.1**

TABLE 12
State-of-the-art comparison of BCNet built on CMTrack RCNN [83] on the OVIS validation set, using ResNet-50 as backbone. Results are reported in terms of mask accuracy (AP) and recall (AR).

Method	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
MaskTrack [82]	10.8	25.3	8.5	7.9	14.9
SipMask [94]	10.2	24.7	7.8	7.9	15.8
QueryInst [39]	14.7	34.7	11.6	9.0	21.2
CrossVIS [95]	14.9	32.7	12.1	10.3	19.8
STMask [96]	15.4	33.8	12.5	8.9	21.3
CMTrack RCNN [83]	15.4	33.9	13.1	9.3	20.0
CMTrack RCNN [83] + **BCNet**	**17.1**	**35.8**	**14.2**	**10.9**	**21.3**

objects, one straightforward solution is to train BCNet in a class-agnostic manner as [99]; 2) BCNet only focuses on the mask head design, thus the segmentation performance

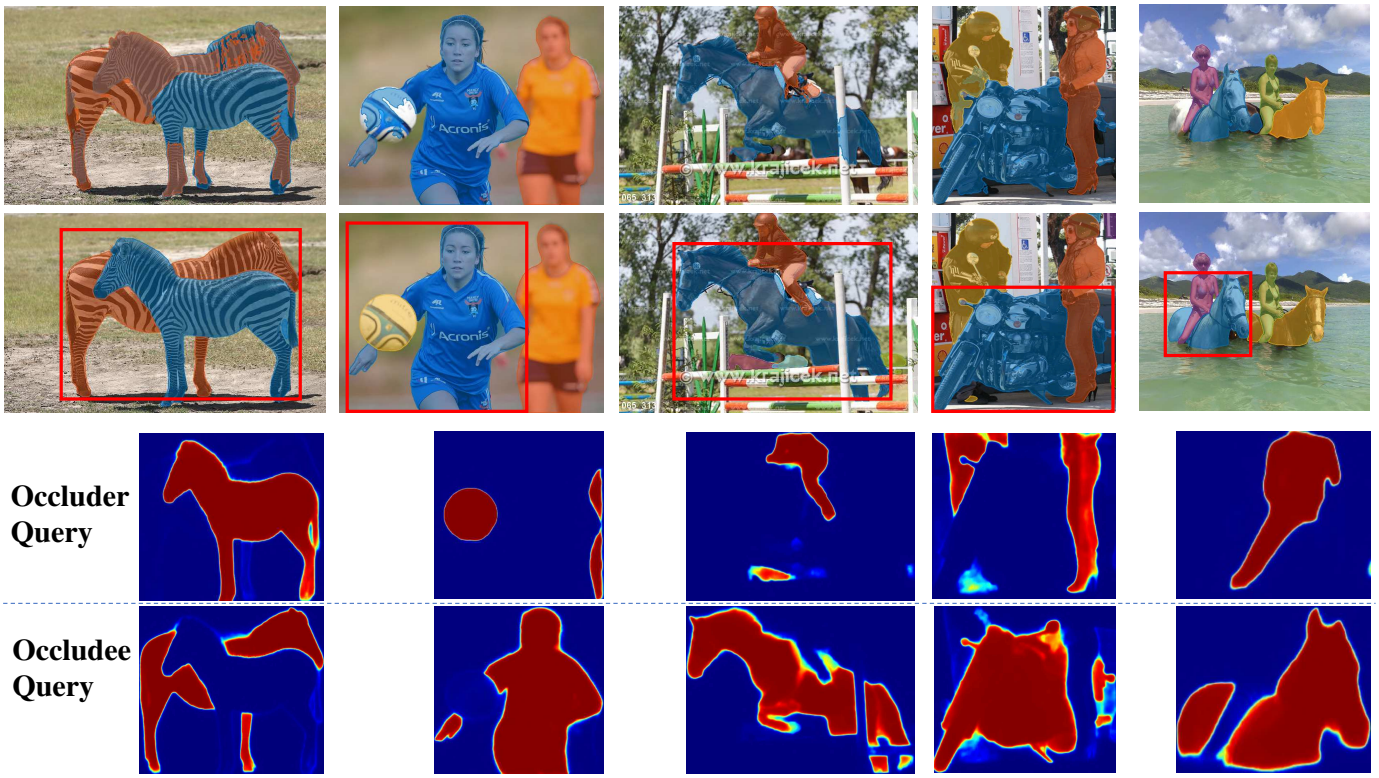


Fig. 11. Qualitative instance segmentation results of **transformer-based** BCNet with single transformer decoder (top row) and bilayer transformer structure (middle row) on **COCO** [7], both using ResNet-50-FPN. The bottom row visualizes squared heatmap of mask predictions by the occluder and occludee queries in the same region specified by the red bounding box.

Fig. 12. Qualitative results comparison between Mask Track R-CNN [82] (top row) and Mask Track R-CNN [82] + BCNet (bottom row) using R50-FPN as backbone on YTVIS validation set. BCNet produces more accurate segmentation results inside the overlapping regions between the two tandem skydivers, by replacing the frame-level mask head of Mask Track R-CNN.

TABLE 13
State-of-the-art comparison of BCNet built on PCAN [85] on the BDD100K segmentation tracking validation set. I: ImageNet. C: COCO. S: Cityscapes. B: BDD100K. "-fix" means adopting the pretrained model from the BDD100K tracking set, fixing the existing parts, and only training the added mask head.

Method	Online	mMOTSA↑	mMOTSP↑	mIDF↑	ID sw.↓	mAP↑
SortIoU	✓	10.3	59.9	21.8	15951	22.2
MaskTrackRCNN [92]	✓	12.3	59.9	26.2	9116	22.0
STEm-Seg [97]	×	12.2	58.2	25.4	8732	21.8
QDTrack-mots [98]	✓	22.5	59.6	40.8	1340	22.4
QDTrack-mots-fix [98]	✓	23.5	66.3	44.5	973	25.5
PCAN [85]	✓	27.4	66.7	45.1	876	26.6
PCAN [85] + **BCNet**	✓	**28.5**	**67.6**	**46.1**	**825**	**28.0**

will be heavily influenced by the accuracy of the one/two-stage bounding box detectors; 3) BCNet is designed on single images which cannot utilize temporal cues in videos. Temporal information entails multiple views of the same dynamic moving objects for establishing correspondence. Further upgrading BCNet with temporal reasoning has the potential to further boost the performance of detecting and segmenting occluded video objects, a future research direction for pursuit.

6 Conclusion

We propose BCNet, an effective mask prediction network for addressing instance segmentation in the presence of highly-overlapping objects in both image and video instance segmentation. BCNet achieves consistent gains on overall performance using different backbones and one/two-stage object detectors in both the modal and amodal settings. We further explore the bilayer decoupling strategy on vision transformers (ViT) by representing instances in the image as separate occluder and occludee queries groups, and design the bilayer transformer decoder. With explicit occluder-occludee modeling, occluding and occluded instances are decoupled into two disjoint graph spaces, where the interaction between objects are explicitly considered. This effective approach will benefit future research in both occlusion handling and instance segmentation.

REFERENCES

[1] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017. 1, 2, 3, 5, 7, 8, 9, 11

[2] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *CVPR*, 2018. 1, 2, 9, 11

[3] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *CVPR*, 2018. 1, 2, 3, 4

[4] L.-C. Chen, A. Hermans, G. Papandreou, F. Schroff, P. Wang, and H. Adam, "Masklab: Instance segmentation by refining object detection with semantic and direction features," in *CVPR*, 2018. 1, 2, 11

[5] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, "Mask scoring r-cnn," in *CVPR*, 2019. 1, 2, 3, 4, 9, 11

[6] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang *et al.*, "Hybrid task cascade for instance segmentation," in *CVPR*, 2019. 1, 2, 3, 4, 9, 11

[7] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014. 1, 2, 4, 7, 10, 12

[8] L. Qi, L. Jiang, S. Liu, X. Shen, and J. Jia, "Amodal instance segmentation with kins dataset," in *CVPR*, 2019. 1, 2, 3, 4, 7, 9, 10

[9] P. Follmann, R. K. Nig, P. H. Rtinger, M. Klostermann, and T. B. Ttger, "Learning to see the invisible: End-to-end trainable amodal instance segmentation," in *WACV*, 2019. 1, 2, 3, 4, 8, 9

[10] X. Chen, R. Girshick, K. He, and P. Dollár, "Tensormask: A foundation for dense object segmentation," in *ICCV*, 2019. 2, 7, 11

[11] Y. Lee and J. Park, "Centermask: Real-time anchor-free instance segmentation," in *CVPR*, 2020. 2, 3, 5, 7, 8, 9, 10, 11

[12] S. Liu, X. Qi, J. Shi, H. Zhang, and J. Jia, "Multi-scale patch aggregation (mpa) for simultaneous detection and segmentation," in *CVPR*, 2016. 1

[13] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *CVPR*, 2016. 1

[14] K. Li, B. Hariharan, and J. Malik, "Iterative instance segmentation," in *CVPR*, 2016. 1, 3

[15] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," in *NeurIPS*, 2011. 1

[16] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *CVPR*, 2015. 1

[17] L. Ke, Y.-W. Tai, and C.-K. Tang, "Deep occlusion-aware instance segmentation with overlapping bilayers," in *CVPR*, 2021. 2

[18] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *ICCV*, 2019. 2, 5, 6, 8, 9, 10, 11

[19] A. Kolesnikov, A. Dosovitskiy, D. Weissenborn, G. Heigold, J. Uszkoreit, L. Beyer, M. Minderer, M. Dehghani, N. Houlsby, S. Gelly, T. Unterthiner, and X. Zhai, "An image is worth 16x16 words: Transformers for image recognition at scale," in *ICLR*, 2021. 2

[20] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *CVPR*, 2022. 2, 3, 4, 6, 7, 8, 9, 11

[21] K. Li and J. Malik, "Amodal instance segmentation," in *ECCV*, 2016. 3

[22] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation," in *CVPR*, 2017. 2

[23] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NeurIPS*, 2015. 2, 5, 8, 9, 10, 11

[24] K. Sofiiuk, O. Barinova, and A. Konushin, "Adaptis: Adaptive instance selection network," in *ICCV*, 2019. 3

[25] E. Xie, P. Sun, X. Song, W. Wang, X. Liu, D. Liang, C. Shen, and P. Luo, "Polarmask: Single shot instance segmentation with polar representation," in *CVPR*, 2020. 3

[26] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "Yolact: real-time instance segmentation," in *ICCV*, 2019. 3, 7, 11

[27] X. Wang, T. Kong, C. Shen, Y. Jiang, and L. Li, "Solo: Segmenting objects by locations," *arXiv preprint arXiv:1912.04488*, 2019. 3

[28] A. Kirillov, E. Levinkov, B. Andres, B. Savchynskyy, and C. Rother, "Instancecut: from edges to instances with multicut," in *CVPR*, 2017. 3

[29] A. Arnab and P. H. Torr, "Pixelwise instance segmentation with a dynamically instantiated network," in *CVPR*, 2017. 3

[30] S. Liu, J. Jia, S. Fidler, and R. Urtasun, "Sgn: Sequential grouping networks for instance segmentation," in *ICCV*, 2017. 3

[31] Y. Liu, S. Yang, B. Li, W. Zhou, J. Xu, H. Li, and Y. Lu, "Affinity derivation and graph merge for instance segmentation," in *ECCV*, 2018. 3

[32] M. Bai and R. Urtasun, "Deep watershed transform for instance segmentation," in *CVPR*, 2017. 3

[33] S. Kong and C. C. Fowlkes, "Recurrent pixel embedding for instance grouping," in *CVPR*, 2018. 3

[34] T. Li, Z. Liang, S. Zhao, J. Gong, and J. Shen, "Self-learning with rectification strategy for human parsing," in *CVPR*, 2020. 3

[35] X. Lu, W. Wang, M. Danelljan, T. Zhou, J. Shen, and L. Van Gool, "Video object segmentation with episodic graph memory networks," in *ECCV*, 2020. 3

[36] Y. Pang, Y. Li, J. Shen, and L. Shao, "Towards bridging semantic gap to improve semantic segmentation," in *ICCV*, 2019. 3

[37] L. Li, T. Zhou, W. Wang, J. Li, and Y. Yang, "Deep hierarchical semantic segmentation," in *CVPR*, 2022. 3

[38] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *ECCV*, 2020. 3, 6, 7

[39] Y. Fang, S. Yang, X. Wang, Y. Li, C. Fang, Y. Shan, B. Feng, and W. Liu, "Instances as queries," in *ICCV*, 2021. 3, 6, 11

[40] B. Dong, F. Zeng, T. Wang, X. Zhang, and Y. Wei, "Solq: Segmenting objects by learning queries," in *NeurIPS*, 2021. 3, 6, 11

[41] Y. Wang, Z. Xu, X. Wang, C. Shen, B. Cheng, H. Shen, and H. Xia, "End-to-end video instance segmentation with transformers," in *CVPR*, 2021. 3

[42] R. Guo, D. Niu, L. Qu, and Z. Li, "Sotr: Segmenting objects with transformers," in *ICCV*, 2021. 3

[43] J. Hu, L. Cao, Y. Lu, S. Zhang, K. Li, F. Huang, L. Shao, and R. Ji, "Istr: End-to-end instance segmentation via transformers," *arXiv preprint arXiv:2105.00637*, 2021. 3, 11

[44] L. Ke, M. Danelljan, X. Li, Y.-W. Tai, C.-K. Tang, and F. Yu, "Mask transfiner for high-quality instance segmentation," in *CVPR*, 2022. 3, 6, 11

[45] L. Ke, H. Ding, M. Danelljan, Y.-W. Tai, C.-K. Tang, and F. Yu, "Video mask transfiner for high-quality video instance segmentation," in *ECCV*, 2022. 3

[46] B. Cheng, A. G. Schwing, and A. Kirillov, "Per-pixel classification is not all you need for semantic segmentation," 2021. 3

[47] W. Wang, J. Liang, and D. Liu, "Learning equivariant segmentation with instance-unique querying," in *NeurIPS*, 2022. 3

[48] J. Sun, Y. Li, S. B. Kang, and H.-Y. Shum, "Symmetric stereo matching for occlusion handling," in *CVPR*, 2005. 3

[49] J. Winn and J. Shotton, "The layout consistent random field for recognizing and segmenting partially occluded objects," in *CVPR*, 2006. 3

[50] T. Gao, B. Packer, and D. Koller, "A segmentation-aware object detection model with occlusion handling," in *CVPR*, 2011. 3

[51] X. Chen and A. L. Yuille, "Parsing occluded people by flexible compositions," in *CVPR*, 2015. 3

[52] Y. Yang, S. Hallman, D. Ramanan, and C. C. Fowlkes, "Layered object models for image segmentation," *TPAMI*, vol. 34, no. 9, pp. 1731–1743, 2011. 3

[53] E. Hsiao and M. Hebert, "Occlusion reasoning for object detectionunder arbitrary viewpoint," *TPAMI*, vol. 36, no. 9, pp. 1803–1815, 2014. 3

[54] Y. Zhu, Y. Tian, D. Metaxas, and P. Dollár, "Semantic amodal segmentation," in *CVPR*, 2017. 3, 4, 7, 8, 9

[55] X. Yan, F. Wang, W. Liu, Y. Yu, S. He, and J. Pan, "Visualizing the invisible: Occluded vehicle segmentation and recovery," in *ICCV*, 2019. 3

[56] G. Ghiasi, Y. Yang, D. Ramanan, and C. C. Fowlkes, "Parsing occluded people," in *CVPR*, 2014. 3

[57] L. Ke, S. Li, Y. Sun, Y.-W. Tai, and C.-K. Tang, "Gsnet: Joint vehicle pose and shape reconstruction with geometrical and scene-aware supervision," in *ECCV*, 2020. 3

[58] J. Tighe, M. Niethammer, and S. Lazebnik, "Scene parsing with object instances and occlusion ordering," in *CVPR*, 2014. 3

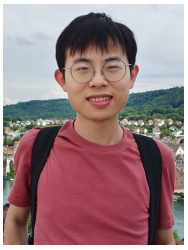
[59] Y.-T. Chen, X. Liu, and M.-H. Yang, "Multi-instance object segmentation with occlusion handling," in *CVPR*, 2015. 3

[60] C. Zhou and J. Yuan, "Bi-box regression for pedestrian detection and occlusion estimation," in *ECCV*, 2018. 3

[61] X. Wang, T. Xiao, Y. Jiang, S. Shao, J. Sun, and C. Shen, "Repulsion loss: Detecting pedestrians in a crowd," in *CVPR*, 2018. 3

[62] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Occlusion-aware r-cnn: detecting pedestrians in a crowd," in *ECCV*, 2018. 3

[63] K. Ehsani, R. Mottaghi, and A. Farhadi, "Segan: Segmenting and generating the invisible," in *CVPR*, 2018. 3
[64] J. Lazarow, K. Lee, and Z. Tu, "Learning instance occlusion for panoptic segmentation," in *CVPR*, 2020. 3
[65] X. Zhan, X. Pan, B. Dai, Z. Liu, D. Lin, and C. C. Loy, "Self-supervised scene de-occlusion," in *CVPR*, 2020. 3
[66] L. Ke, Y.-W. Tai, and C.-K. Tang, "Occlusion-aware video object inpainting," in *ICCV*, 2021. 3
[67] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár, "Learning to refine object segments," in *ECCV*, 2016. 3
[68] Y.-T. Hu, H.-S. Chen, K. Hui, J.-B. Huang, and A. G. Schwing, "Sail-vos: Semantic amodal instance level video object segmentation-a synthetic dataset and baselines," in *CVPR*, 2019. 4
[69] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*, 2012. 4, 7
[70] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016. 5, 7, 9
[71] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *CVPR*, 2018. 4, 5
[72] X. Wang and A. Gupta, "Videos as space-time region graphs," in *ECCV*, 2018. 4, 5
[73] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017. 4
[74] Y. Chen, M. Rohrbach, Z. Yan, Y. Shuicheng, J. Feng, and Y. Kalantidis, "Graph-based global reasoning networks," in *CVPR*, 2019. 4
[75] L. Zhang, X. Li, A. Arnab, K. Yang, Y. Tong, and P. H. Torr, "Dual graph convolutional network for semantic segmentation," in *BMVC*, 2019. 4
[76] Y. Li and A. Gupta, "Beyond grids: Learning graph representations for visual recognition," in *NeurIPS*, 2018. 4
[77] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016. 4
[78] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *CVPR*, 2017. 5
[79] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *CVPR*, 2016. 5
[80] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *ICCV*, 2017. 5
[81] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph, "Simple copy-paste is a strong data augmentation method for instance segmentation," in *CVPR*, 2021. 7
[82] L. Yang, Y. Fan, and N. Xu, "Video instance segmentation," in *ICCV*, 2019. 7, 11, 12
[83] J. Qi, Y. Gao, Y. Hu, X. Wang, X. Liu, X. Bai, S. Belongie, A. Yuille, P. Torr, and S. Bai, "Occluded video instance segmentation," *arXiv preprint arXiv:2102.01558*, 2021. 7, 10, 11
[84] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in *CVPR*, 2020. 7
[85] L. Ke, X. Li, M. Danelljan, Y.-W. Tai, C.-K. Tang, and F. Yu, "Prototypical cross-attention networks for multiple object tracking and segmentation," in *Advances in Neural Information Processing Systems*, 2021. 7, 10, 12
[86] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe, "Mots: Multi-object tracking and segmentation," in *CVPR*, 2019. 7
[87] S.-H. Zhang, R. Li, X. Dong, P. Rosin, Z. Cai, X. Han, D. Yang, H. Huang, and S.-M. Hu, "Pose2seg: Detection free human instance segmentation," in *CVPR*, 2019. 9
[88] T. Cheng, X. Wang, L. Huang, and W. Liu, "Boundary-preserving mask r-cnn," in *ECCV*, 2020. 11
[89] W. Kuo, A. Angelova, J. Malik, and T.-Y. Lin, "Shapemask: Learning to segment novel objects by refining shape priors," in *ICCV*, 2019. 11
[90] H. Chen, K. Sun, Z. Tian, C. Shen, Y. Huang, and Y. Yan, "BlendMask: Top-down meets bottom-up for instance segmentation," in *CVPR*, 2020. 11
[91] L. Yang, Y. Wang, X. Xiong, J. Yang, and A. K. Katsaggelos, "Efficient video object segmentation via network modulation," in *CVPR*, 2018. 11
[92] P. Voigtlaender, Y. Chai, F. Schroff, H. Adam, B. Leibe, and L.-C. Chen, "Feelvos: Fast end-to-end embedding learning for video object segmentation," in *CVPR*, 2019. 11, 12
[93] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *IEEE international conference on image processing (ICIP)*, 2017. 11
[94] J. Cao, R. M. Anwer, H. Cholakkal, F. S. Khan, Y. Pang, and L. Shao, "Sipmask: Spatial information preservation for fast image and video instance segmentation," in *ECCV*, 2020. 11
[95] S. Yang, Y. Fang, X. Wang, Y. Li, C. Fang, Y. Shan, B. Feng, and W. Liu, "Crossover learning for fast online video instance segmentation," in *ICCV*, 2021. 11
[96] M. Li, S. Li, L. Li, and L. Zhang, "Spatial feature calibration and temporal fusion for effective one-stage video instance segmentation," in *CVPR*, 2021. 11
[97] A. Athar, S. Mahadevan, A. Ošep, L. Leal-Taixé, and B. Leibe, "Stem-seg: Spatio-temporal embeddings for instance segmentation in videos," in *ECCV*, 2020. 12
[98] J. Pang, L. Qiu, X. Li, H. Chen, Q. Li, T. Darrell, and F. Yu, "Quasi-dense similarity learning for multiple object tracking," in *CVPR*, 2021. 12
[99] X. Gu, T.-Y. Lin, W. Kuo, and Y. Cui, "Open-vocabulary object detection via vision and language knowledge distillation," 2022. 11



Lei Ke is a Ph.D. candidate in the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology, advised by Chi-Keung Tang and Yu-Wing Tai. He is also a visiting scholar in the Computer Vision Laboratory of ETH Zürich since 2021. His research interests include image/video instance segmentation and object tracking. He received the BEng degree in Software Engineering from Wuhan University.

Yu-Wing TAI is a senior research director at Kuaishou Technology and an adjunct professor at CSE Department of HKUST. He received his BEng (First Class Honors) and MPhil degrees from the Department of Computer Science and Engineering, HKUST in 2003 and 2005 and PhD degree from the National University of Singapore in 2009. He was a research director of YouTu research lab of Tencent from January 2017 to April 2020. He was a principle research scientist of SenseTime Group Limited from September 2015 to December 2016. He was an associate professor at the KAIST from July 2009 to August 2015. He is an associate editor of IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI). He regularly served as an area chair/technical program committee member of CVPR/ICCV/ECCV. His research interests include deep learning, computer vision and image/video processing.

Chi-Keung TANG received the MSc and PhD degrees in Computer Science from the University of Southern California, Los Angeles, in 1999 and 2000, respectively. Since 2000, he has been with the CSE Department at HKUST where he is currently a full professor. He was on sabbatical at the University of California, Los Angeles, in 2008. He was an adjunct researcher at the Visual Computing Group of Microsoft Research Asia. His research areas are computer vision, computer graphics and machine learning. He was an associate editor of IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) and was on the editorial board of International Journal of Computer Vision (IJCV). He served as an area chair for ACCV 2006, ICCV 2007, ICCV 2009, ICCV 2011, ICCV 2015, ICCV 2017, ICCV 2019, ECCV 2020, CVPR2021, ICCV 2021, ECCV 2022, and as a technical papers committee member for the inaugural SIGGRAPH Asia 2008, SIGGRAPH 2011, SIGGRAPH Asia 2011, SIGGRAPH 2012, SIGGRAPH Asia 2012, SIGGRAPH Asia 2014 and SIGGRAPH Asia 2015. He is a Fellow of the IEEE Computer Society, and has served on the IEEE Fellow Review Committee.