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 Bilayer Convolutional Network (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, BDD100 K MOTS), especially for heavy occlusion cases.

Index Terms—BCNet, bilayer decoupling, occlusion-aware instance segmentation, occlusion-aware video instance segmentation.

I. 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 Fig. 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 Fig. 2. We compare BCNet with recent amodal segmentation methods [8], [9], which predict complete
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.

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 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. Bt and Mt respectively denote box and mask head at tth 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”).

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.

II. 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 mismatch 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 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. AdapTS [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.

Transformer-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 Transfimer [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. OC Fusion [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 (Fig. 1), where the occludee is segmented under the guidance from the shape and location of the occluder.

### III. Occlusion-Aware Instance Segmentation

We first describe the explicit occluder-occludee modeling of our proposed Bilayer Convolutional Network (BCNet) in Section III-A, and then give an overview to the overall bilayer GCN-based instance segmentation framework in Section III-B. Based on the principle of bilayer decoupling, we further design a bilayer transformer-based on Mask2Former [20] for occlusion-aware instance segmentation in Section III-C. 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 Fig. 3 directly regress 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.

#### A. 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 Fig. 4, which constructs two orthogonal graphs in a single general framework.

Following [72], given an adjacency graph $G = (V, E)$ with edges $E$ among nodes $V$, we represent the graph convolution operation as,

$$ Z = \sigma(A X W_g) + X, \quad (1) $$

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

To construct the adjacency matrix $A$, we define the pairwise similarity between every two graph nodes $x_i, x_j$ by dot product similarity as,

$$ A_{ij} = softmax(F(x_i, x_j)), \quad (2) $$

$$ F(x_i, x_j) = \theta(x_i)^T \phi(x_j), \quad (3) $$

where $\theta$ and $\phi$ are two trainable transformation function implemented by $1 \times 1$ convolution as shown in the non-local operator part of Fig. 4, so that high confidence edge between two nodes corresponds to larger feature similarity.

In our bilayer GCN structure, we further define $G^1$ to indicate the $i$th graph, $X_{roi}$ for the input ROI feature and $W_f$ for weights in FCN layers. The pertinent equations are:

$$ Z^1 = \sigma(A^1 X_f W^1_g) + X_f, \quad (4) $$

$$ X_f = Z^0 W^0_0 + X_{roi}, \quad (5) $$

$$ Z^0 = \sigma(A^0 X_{roi} W^0_0) + X_{roi}. \quad (6) $$

For connecting the two GCN blocks, the output feature $Z^0$ of the occluder from the first GCN is directly added to $X_{roi}$ to obtain the fused occlusion-aware feature $X_f$, which is the input for the second GCN layer to output $Z^1$ 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 Fig. 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 Fig. 4). The rationale behind this design is that such irregular occlusion boundary unrelated to the occludee is confusing, which in turn provides essential 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 $3 \times 3$ convolutional layer followed by one GCN layer and one FCN layer. Then we feed the output to the up-sampling layer and one $1 \times 1$ convolutional layer to obtain one channel feature map for joint boundary and mask predictions. The boundary detection for occluder is trained with loss $\mathcal{L}'_{\text{Occ-B}}$:

$$
\mathcal{L}'_{\text{Occ-B}} = \mathcal{L}_{\text{BCE}}(W_{B}\mathcal{F}_{\text{occ}}(X_{\text{roi}}), \mathcal{G}T_B),
$$

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

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

$$
\mathcal{L}'_{\text{Occ-S}} = \mathcal{L}_{\text{BCE}}(W_S\mathcal{F}_{\text{occ}}(X_{\text{roi}}), \mathcal{G}T_S),
$$

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

**B. Bilayer GCN-Based Instance Segmentation**

Fig. 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 the 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 outputs contours and masks for the partially occluded instance.

C. 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 excel 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 Fig. 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). 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 Fig. 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.

D. 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 $L$ as,

$$
L = \lambda_1 L_{Detect} + \lambda_2 L_{Occluder} + \lambda_3 L_{Occludee} 
$$

(9)

$$
L_{Occluder} = \lambda_2 L_{Occ-B} + \lambda_3 L_{Occ-S} 
$$

(10)
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.

### Table I

| Oclusion (Occluder) Modeling | COCO-OCC | COCO |
|-----------------------------|----------|------|
| Contour                     | AP       | AP50 |
| Mask                        | AP       | AP50 |
| O                  | 29.04    | 49.22 | 32.65 | 52.39 |
| ✓                           | 29.65    | 49.42 | 33.25 | 52.83 |
| ✓                           | 30.18    | 49.94 | 33.41 | 53.02 |
| ✓                           | 30.37    | 50.40 | 33.43 | 53.12 |

### Table II

| Target (Occudee) Modeling | COCO-OCC | COCO |
|---------------------------|----------|------|
| Guidance                  | AP       | AP50 |
| Contour                   | AP       | AP50 |
| Mask                      | AP       | AP50 |
| ✓                         | 29.45    | 49.73 | 32.56 | 52.21 |
| ✓                         | 30.37    | 50.40 | 33.43 | 53.12 |
| ✓                         | 30.68    | 50.62 | 33.62 | 53.26 |

\[
\mathcal{L}_{\text{Occudee}} = \lambda_4 \mathcal{L}_{\text{Occ-B}} + \lambda_5 \mathcal{L}_{\text{Occ-S}},
\]

(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 occudee, which are similar to (7) and (8). \( \mathcal{L}_{\text{Detect}} \) supervises both the position prediction and the category classification borrowed from the FCOS [18] detector,

\[
\mathcal{L}_{\text{Detect}} = \mathcal{L}_{\text{Regression}} + \mathcal{L}_{\text{Centerness}} + \mathcal{L}_{\text{Class}},
\]

(12)

and \( \lambda_1, \lambda_2, \lambda_3, \lambda_4 \) and \( \lambda_5 \) 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,

\[
\mathcal{L}_{\text{Detect}} = \mathcal{L}_{\text{Box}} + \mathcal{L}_{\text{Matching}} + \mathcal{L}_{\text{Class}},
\]

(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].

### Table III

| Structure      | FCN  | GCN  | COCO-OCC | COCO |
|----------------|------|------|----------|------|
|                | AP   | AP50 | AP       | AP50 |
| Single Layer   | ✓    | ✓    | 28.43    | 48.24 | 33.01 | 52.62 |
|                |      |      | 29.63    | 49.59 | 33.14 | 52.81 |
|                |      |      | 51.4M    | 51.4M |
| Bilayer        | ✓    | ✓    | 30.12    | 49.92 | 33.16 | 52.80 |
|                |      |      | 53.4M    | 53.4M |

### Table IV

| Model                      | COCO-OCC | COCO |
|----------------------------|----------|------|
|                            | AP       | AP50 |
| FCOS [11] + Baseline       | 28.43    | 48.24 | 33.01 | 52.62 |
| FCOS [18] + Ours           | 30.69    | 50.62 | 33.62 | 53.26 |
| Faster R-CNN [11] + Baseline | 29.67    | 49.95 | 33.45 | 53.70 |
| Faster R-CNN [23] + Ours   | 31.71    | 51.15 | 34.61 | 54.41 |
| Query-based Detector [20] + Baseline | 39.23    | 50.62 | 41.13 | 62.30 |
| Query-based Detector [20] + Ours | 41.47    | 52.03 | 42.51 | 64.23 |

Training. During training, following Mask R-CNN [11], 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 occudee 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 90 K iterations which starts with 1 K 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 (10) and (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.

IV. 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 Fig. 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 100 K 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 VIII.

V. EXPERIMENTS

A. Experimental Setup

**COCO and COCO-OCC.** We conduct experiments on COCO dataset [7], where we train on 2017train (115 k images) and evaluate results on both 2017val and 2017test-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 (5 k 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 2017val, 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 131 k 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. BDD100 K covers the self-driving scenario while YTVIS and OVIS have more diverse object categories.

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

**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 II, 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 III 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.

**Using FCN or GCN?** Table III also reveals the advantage of GCN over FCN, where GCN achieves consistent superior performance both in the single 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 down-sampled to only 14×14 (spatial size) with 256 channels.

**Effect of Bilayer Transformer Decoder.** Table V tabulates the effect of our transformer-based BCNet with Bilayer Transformer Decoder. Compared to the shared standard 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, 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 IV, 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.

---

**C. Performance Comparison and Analysis**

**Comparison With Amodal Segmentation Methods.** Tables VI and VII 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. Figs. 7 and 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 VIII, our BCNet with Faster R-CNN detector has 31.71 AP versus 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 key-point/pose usage, achieving large 4.3 mask AP improvement.

Comparison With SOTA Methods. Table X 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. Fig. 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 shown in Fig. 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 Fig. 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.

Anomalous Results Comparison on KINS. In Fig. 9, we additionally provide qualitative anomalous segmentation results comparison between Mask R-CNN + ASN module [8] and our BCNet on KINS [8] test set. Take the first case as an example, our BCNet infers more reasonable anomalous car shape even when

---

**TABLE X**

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  | AP50 | AP75 | AP25 | AP45 | AP65 |
|-------------------------|-----------|----------|-----|------|------|------|------|------|
| 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    | 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    | 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 |
| ISTK [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.
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.

**Table XI**

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  | AP50 | AP75 | AR1 | AR10 |
|-------------------|-----|------|------|-----|------|
| 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 XII**

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  | AP50 | AP75 | AR1 | AR10 |
|-------------------|-----|------|------|-----|------|
| MaskTrack [82]    | 10.8| 25.3 | 8.5  | 7.9 | 14.9 |
| SupMask [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 |

**Table XIII**

STATE-OF-THE-ART COMPARISON OF BCNet BUILT ON PCAN [85] ON THE BDD100 K SEGMENTATION TRACKING VALIDATION SET. I: ImageNet, C: COCO, S: Cityscapes. B: BDD100 K. “-Fix” MEANS ADOPTING THE PRETRAINED MODEL FROM THE BDD100 K TRACKING SET, FIXING THE EXISTING PARTS, AND ONLY TRAINING THE ADDED MASK HEAD

| Method            | Online | mMOTS† | mMOTSI† | mEP† | ID sw ‡ | mAP‡ |
|-------------------|--------|--------|---------|------|---------|------|
| SsortLA           | ✓      | 10.3   | 59.9    | 21.8 | 1595    | 22.2 |
| MaskTrack [92]    | ✓      | 12.3   | 59.9    | 36.2 | 9116    | 22.0 |
| STEm-Seg [97]     | ×      | 12.2   | 58.2    | 25.4 | 8732    | 21.8 |
| QTTrack-mot-e [98]| ✓      | 23.5   | 46.3    | 44.5 | 973     | 25.1 |
| QTTrack-mot-e-fx [98] | ✓ | 23.5 | 46.3 | 44.5 | 973 | 25.1 |
| PCAN [85]          | ✓      | 27.4   | 66.7    | 65.1 | 876     | 26.6 |
| PCAN [85] + BCNet  | ✓      | 28.5   | 67.6    | 46.1 | 825     | 29.0 |

**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 XI show an improvement of 2.1 AP. We also show one challenging qualitative results comparison in Fig. 12, where the overlapping regions between the two tandem skydivers are much better segmented by BCNet. For experiments on OVIS in Table XII, 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 BDD100 K MOTS, we augment the mask head of PCAN [85] with our GCN-based BCNet in Table XIII.
where MOTSQA 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 mMOTSQA 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 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 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.

**VI. 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 Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 2980–2988.
[2] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, “Path aggregation network for instance segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2018, pp. 8759–8768.
[3] Z. Cai and N. Vasconcelos, “Cascade R-CNN: Delving into high quality object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2018, pp. 6154–6162.
[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 Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2018, pp. 4013–4022.
[5] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, “Mask scoring R-CNN,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 6402–6411.
[6] K. Chen et al., “Hybrid task cascade for instance segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 4969–4978.
[7] T.-Y. Lin et al., “Microsoft COCO: Common objects in context,” in Proc. Eur. Conf. Comput. Vis., 2014, pp. 740–755.
[8] L. Qi, L. Jiang, S. Liu, X. Shen, and J. Jia, “Amodal instance segmentation with RINS dataset,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 3009–3018.
[9] P. Pollmann, R. K. Nig, P. H. Rininger, M. Klostermann, and T. B. Tiger, “Learning to see the invisible: End-to-end trainable amodal instance segmentation,” in Proc. IEEE Winter Appl. Comput. Vis., 2019, pp. 1328–1336.
[10] X. Chen, R. Girshick, K. He, and P. Dollár, “TensorMask: A foundation for dense object segmentation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 2061–2069.
[11] Y. Lee and J. Park, “CenterMask: Real-time anchor-free instance segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 13903–13912.
[12] S. Liu, X. Qi, J. Shi, H. Zhang, and J. Jia, “Multi-scale patch aggregation (MPA) for simultaneous detection and segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 3141–3149.
[13] J. Dai, K. He, and J. Sun, “Instance-aware semantic segmentation via multi-task network cascades,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 3150–3158.
[14] K. Li, B. Hariharan, and J. Malik, “Iterative instance segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 3659–3667.
[15] P. Krähenbühl and V. Koltun, “Efficient inference in fully connected CRFs with Gaussian edge potentials,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2011, pp. 109–117.
[16] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, “Hypercolumns for object segmentation and fine-grained localization,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2015, pp. 447–456.
[17] L. Ke, Y.-W. Tai, and C.-K. Tang, “Deep occlusion-aware instance segmentation with overlapping bilayers,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 4018–4027.

[18] Z. Tian, C. Shen, H. Chen, and T. He, “FCOS: Fully convolutional one-stage object detection,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 9626–9635.

[19] A. Kolesnikov et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” in Proc. Int. Conf. Learn. Representations, 2021.

[20] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, “Mask-att-attention mask transformer for universal image segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 1280–1289.

[21] K. Li and J. Malik, “Amodal instance segmentation,” in Proc. Eur. Conf. Comput. Vis., 2022, pp. 677–693.

[22] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, “Fully convolutional instance-aware semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 4438–4446.

[23] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2015, pp. 91–99.

[24] K. Sofluik, O. Barinova, and A. Konushin, “AdaptIS: Adaptive instance selection network,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 7584–7592.

[25] E. Xie et al., “PolarMask: Single shot instance segmentation with polar representation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 12190–12199.

[26] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, “YOLACT: Real-time instance segmentation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 9156–9165.

[27] X. Wang, T. Kong, C. Shen, Y. Jiang, and L. Li, “SOLO: Segmenting objects by locations,” 2019, arXiv:1912.04488.

[28] A. Kirillov, E. Levinkov, B. Andres, B. Savchynskyy, and C. Rother, “InstanceCut: From edges to instances with multicut,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2017, pp. 7322–7331.

[29] A. Arnah and P. H. Torr, “Pixelwise instance segmentation with a dynamically instantiated network,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 879–888.

[30] S. Liu, J. Jia, S. Idler, and R. Urtasun, “SGN: Sequential grouping networks for instance segmentation,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 3516–3524.

[31] Y. Liu et al., “Affinity derivation and graph merge for instance segmentation,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 708–724.

[32] M. Bai and R. Urtasun, “Deep watershed transform for instance segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 2858–2866.

[33] S. Kong and C. C. Fowlkes, “Recurrent pixel embedding for instance grouping,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2018, pp. 9018–9027.

[34] T. Li, Z. Liang, S. Zhao, J. Gong, and J. Shen, “Self-learning with rectification strategy for human parsing,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 9260–9269.

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

[36] Y. Pang, Y. Li, J. Shen, and L. Shao, “Towards bridging semantic gap to improve semantic segmentation,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 4229–4238.

[37] L. Li, T. Zhou, W. Wang, J. Li, and Y. Yang, “Deep hierarchical semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2022, pp. 1256–1267.

[38] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 213–229.

[39] Y. Fang et al., “Instances as queries,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 6899–6909.

[40] B. Dong, F. Zheng, T. Wang, X. Zhang, and Y. Wei, “SOLQ: Segmenting objects by learning queries,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2021, pp. 21898–21909.

[41] Y. Wang et al., “End-to-end video instance segmentation with transformers,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 8737–8746.

[42] B. Guo, D. Niu, L. Qu, and Z. Li, “SOTR: Segmenting objects with transformers,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 7137–7146.

[43] J. Hu et al., “ISTR: End-to-end instance segmentation via transformers,” 2021, arXiv:2105.00637.
A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the KITTI vision benchmark suite,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2012, pp. 3354–3361.

K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778.

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.

X. Wang and A. Gupta, “Videos as space-time region graphs,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 413–431.

T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in Proc. Int. Conf. Learn. Representations, 2017.

Y. Chen, M. Rohrbach, Z. Yan, Y. Shuicheng, J. Feng, and Y. Kalantidis, “Graph-based global reasoning networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 433–442.

L. Zhang, X. Li, A. Arnab, K. Yang, Y. Tong, and P. H. Torr, “Dual graph convolutional network for semantic segmentation,” in Proc. Brit. Mach. Conf. Vis., 2019, pp. 254–267.

Y. Li and A. Gupta, “Beyond grids: Learning graph representations for visual recognition,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2018, pp. 9245–9255.

J. L. Bu, J. R. Kiros, and G. E. Hinton, “Layer normalization,” 2016, arXiv:1607.06450.

T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 936–944.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 779–788.

T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal loss for dense object detection,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 2999–3007.

G. Ghiasi et al., “Simple copy-paste is a strong data augmentation method for instance segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 2917–2927.

L. Yang, Y. Fan, and N. Xu, “Video instance segmentation,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 5187–5196.

J. Qi et al., “Occluded video instance segmentation,” 2021, arXiv:2102.01558.

F. Yu et al., “BDD100K: A diverse driving dataset for heterogeneous multitask learning,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 2633–2642.

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 Proc. Adv. Neural Inf. Process. Syst., 2021, pp. 1192–1203.

P. Voigtlaender et al., “MOTS: Multi-object tracking and segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 7934–7943.

S.-H. Zhang et al., “Pose2Seg: Detection free human instance segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 889–898.

T. Cheng, X. Wang, L. Huang, and W. Liu, “Boundary-preserving mask R-CNN,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 660–676.

W. Kuo, A. Angelova, J. Malik, and T.-Y. Lin, “ShapeMask: Learning to segment novel objects by refining shape priors,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 9206–9215.

H. Chen, K. Sun, Z. Tian, C. Shen, Y. Huang, and Y. Yan, “BlendMask: Top-down meets bottom-up for instance segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 8570–8578.

L. Yang, Y. Wang, X. Xiong, J. Yang, and A. K. Katsaggelos, “Efficient video object segmentation via network modulation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2018, pp. 6499–6507.

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 Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 9473–9482.

N. Wojke, A. Bewley, and D. Paulus, “Simple online and realtime tracking with a deep association metric,” in Proc. IEEE Int. Conf. Image Process., 2017, pp. 3645–3649.

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 Proc. Eur. Conf. Comput. Vis., 2020, pp. 1–18.

S. Yang et al., “Crossover learning for fast online video instance segmentation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 8023–8032.

M. Li, S. Li, L. Li, and L. Zhang, “Spatial feature calibration and temporal fusion for effective one-stage video instance segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 11210–11219.

A. Athar, S. Mahadevan, A. Ošep, L. Leal-Taixé, and B. Leibe, “Stem-seg: Spatio-temporal embedding vision, context segmentation in videos,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 158–177.

J. Pang et al., “Quasi-dense similarity learning for multiple object tracking,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2021, pp. 164–173.

X. Gu, T.-Y. Lin, W. Kao, and Y. Cui, “Open-vocabulary object detection via vision and language knowledge distillation,” in Proc. Int. Conf. Learn. Representations, 2021.

KE et al.: OCCLUSION-AWARE INSTANCE SEGMENTATION VIA BILAYER NETWORK ARCHITECTURES

Lei Ke received the BEng degree in software engineering from Wuhan University. He is currently working toward the PhD degree in the Department of Computer Science and Engineering, 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.

Yu-Wing Tai (Senior Member, IEEE) received the BEng (First Class Honors) and MPhil degrees from the Department of Computer Science and Engineering, HKUST, in 2003 and 2005, respectively, and PhD degree from the National University of Singapore, in 2009. He is a senior research Director with Kuaishou Technology and an adjunct professor with the CSE Department of HKUST. 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 with 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 graphics, computer vision, 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 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.