AGGMASK: EXPLORING LOCALLY AGGREGATED LEARNING OF MASK REPRESENTATIONS FOR INSTANCE SEGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recently proposed one-stage instance segmentation models (*e.g.*, SOLO) learn to directly predict location-specific object mask with fully-convolutional networks. They perform comparably well as the traditional two-stage Mask R-CNN model, yet enjoying much simpler architecture and higher efficiency. However, an intrinsic limitation of these models is that they tend to generate similar mask predictions for a single object at nearby locations, while most of them are directly discarded by non-maximum suppression, leading to a waste of some useful predictions that can supplement the final result. In this work, we aim to explore how the model can benefit from better leveraging the neighboring predictions while maintaining the architectural simplicity and efficiency. To this end, we develop a novel learning-based aggregation framework that learns to aggregate the neighboring predictions. Meanwhile, unlike original location-based masks, the segmentation model is implicitly supervised to learn location-aware *mask representations* that encode the geometric structure of nearby objects and complements adjacent representations with context. Based on the aggregation framework, we further introduce a mask interpolation mechanism that enables sharing mask representations for nearby spatial locations, thus allowing the model to generate much fewer representations for computation and memory saving. We experimentally show that by simply augmenting the baseline model with our proposed aggregation framework, the instance segmentation performance is significantly improved. For instance, it improves a SOLO model with ResNet-101 backbone by 2.0 AP on the COCO benchmark, with only about 2% increase of computation. Code and models are available at anonymous repository: https://github.com/advdfacd/AggMask.

1 INTRODUCTION

Semantic instance segmentation (Hariharan et al., 2014) aims to recognize and segment all the individual object instances of interested categories in an input image. Recently, Wang et al. (2019) proposed SOLO, a straightforward instance segmentation model that evenly divides the input image into spatial grid and predicts object mask segmentation and classification score for each grid location with fully convolutional networks (Long et al., 2015). Compared with the previous prevalent detect-then-segment methods (He et al., 2017; Li et al., 2017), SOLO is much simpler and more efficient by avoiding bounding box detection and feature re-pooling, while achieving comparable performance.

Albeit simple, as shown in Fig. 1 (a), the model tends to predict the same object at nearby locations, those predictions can supplement the final result as some may better segment certain object parts, but most of them are directly discarded by non-maximum suppression. It raises a natural question: *can we further improve the performance by leveraging the rich neighboring predictions*?

A straightforward idea is to perform mask voting that averages those neighboring predictions as they have sufficient overlap with the final results. Similar ideas have been explored in the object detection field, known as box voting (Gidaris & Komodakis, 2015). In this work, we first carefully examine such voting algorithm and find that it indeed improves the final performance, albeit the improvement is quite minor. We hypothesize that unlike the bounding boxes, object masks are much more complicated due to deformations and layering, thus cannot benefit much from simple post-

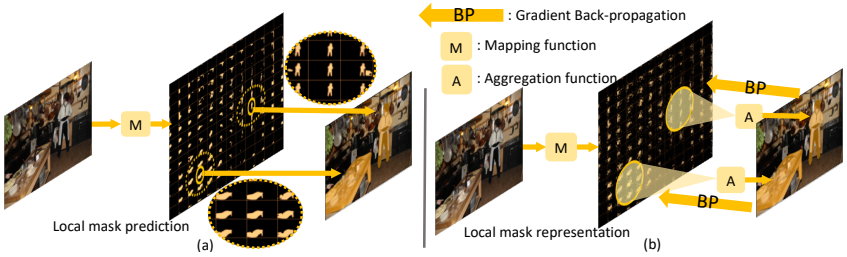

Figure 1: Illustration of the main idea. (a) SOLO (Wang et al., 2019) learns a mapping function with fully convolution network for predicting location-specific object masks; the predictions at neighboring locations can be similar but do not contribute to the final result. (b) The proposed AggMask aims to exploit the neighboring predictions by aggregating them in a learning-based fashion. Meanwhile, by back-propagating through the aggregation function, the mapping function instead learns to generate local shape descriptors (mask representations) that encode shape and layout information of nearby objects and complement each other in combination, so as to generate mask of higher quality.

training voting. We therefore propose a framework named *AggMask* that helps the model better harness the power of neighboring predictions by aggregating them in a learning-based fashion.

Specifically, as shown in Fig. 1 (b), we develop a learnable aggregation function that learns to combine the neighboring mask prediction of each spatial location. With gradients back-propagated through the aggregation function, the model learns to generate intermediate shape descriptors that capture the information of objects at nearby locations, we name those shape descriptors as *mask representations*, which stand in contrast with original location-specific object masks. The subsequent aggregation function can then gather complementary information from neighboring representations to improve segmentation quality. Motivated by the dynamic local filtering (Jia et al., 2016) that generates filter weights for adaptive local spatial transformation, we implement the aggregation function by a small multi-layer convolution network with dynamically generated weights. This form of dynamic aggregation can perform adaptive combination and better handle the variations of object shape due to diverse scale and spatial layout. Different from recent works (Wang et al., 2020; Tian et al., 2020) that use dynamically instantiated network for generating the segmentation mask with a shared base feature, we aim to adaptively combine the locally gathered predictions for final segmentation. Our method is orthogonal and can be combined with those methods.

The location-specific masks can be redundant and costly in computation and memory. This is also another intrinsic limitation of vanilla SOLO so the spatial grid resolution is limited. Since the proposed aggregated learning can generate mask representations that encode neighboring objects information, we propose to alleviate the spatial redundancy by allowing nearby locations to share the same set of representations and deploy the dynamic aggregation to generate masks that differentiate nearby objects. It behaves similarly to bilinear interpolation and thus named mask interpolation.

To summarize, the contributions are: 1) We identify the discarded neighboring prediction issue of SOLO model and carefully examine the widely used voting algorithm to leverage the neighboring predictions. 2) By analyzing deficiency of the simple voting scheme, we develop a learning-based aggregation framework. It enables the learning of intermediate mask representations that capture context object information and improves the final result by adaptively combining the neighboring representations. 3) We further propose mask interpolation that effectively reduces the number of generated representations to save computation and memory. 4) We conduct extensive experiments and model analysis with the COCO and high-quality LVIS benchmarks to validate and understand the proposed method. On COCO, a ResNet-101 based SOLO model gets 2.0 AP boost with only 2% additional Flops. We also evaluate its generalization on SOLOv2 (Wang et al., 2020). We believe the simple and effective framework and interpretable mask representations shed light for future research.

2 Related Work

Instance Segmentation. This area is previously dominated by detect-and-segment methods, or region-based methods (Li et al., 2017; He et al., 2017; Dai et al., 2016; Chen et al., 2019a; Lee &

Park, 2019; Chen et al., 2018) which predicts instance masks under box region for each detected objects. *E.g.*, Mask R-CNN employs an additional convolutional mask branch on Faster R-CNN (Ren et al., 2015) to predict instance masks; Some recent works (Fu et al., 2019; Lee & Park, 2019) replace the detection network with a fast one-stage model and employ a lightweight head for mask prediction. The feature re-pooling and subsequent per ROI processing in those approaches can be costly, recently there has been a surge of interest in one-stage methods (Bolya et al., 2019; Sofiiuk et al., 2019; Chen et al., 2019b; Tian et al., 2020; Xie et al., 2019; Chen et al., 2020; Qi et al., 2020; Zhang et al., 2020; Wang et al., 2019; 2020), which directly predict instance masks along with classification scores. *E.g.,* YOLACT (Bolya et al., 2019) employs a mask assembly mechanism that linearly combines prototypes with learned weights. TensorMask (Xie et al., 2019) generalizes dense object detection to dense instance segmentation with an aligned feature representation. SOLO (Wang et al., 2019) directly predicts location-specific masks with specifically designed FCN architecture and objectives, which can be viewed as SOTA one-stage instance segmentation model.

Locally Aggregated Learning. Aggregating information from the neighborhood has been prevailing in various research fields. *E.g.*, convolutional networks (Fukushima, 1980; LeCun et al., 1995) can be viewed as stacking convolution operators that fuse information from spatially neighboring pixels to get new representation at each spatial location; Graph nerual networks (GNN) (Scarselli et al., 2008; Battaglia et al., 2016) including variants Graph convolutional networks (GCN) (Kipf & Welling, 2016) and GraphSAGE (Hamilton et al., 2017) follow a recursive neighborhood aggregation and transformation scheme to learn high quality representation of graphs data. We are inspired by this formulation and try to explore the aggregated learning in one-stage instance segmentation.

3 PRELIMINARIES

3.1 SOLO FORMULATION

For an input image I, instance segmentation aims to predict set of $\{c_k, M_k\}_{k=1}^n$, where c_k is the object class label and M_k is a binary 2-d map that segments the object, n varies with the number of objects in the image. Without loss of generality, SOLO (Wang et al., 2019) spatially divides the input image into G by G grid and learns two simultaneous mapping functions, which predict the semantic category c_{ij} and the corresponding mask segmentation M_{ij} for each spatial grid location:

$$\mathcal{F}_c(I, \theta_c) : I \mapsto \{c_{ij} \in \mathbb{R}^C | i, j = 0, 1, \ldots, G\}, \tag{1}$$

$$\mathcal{F}_m(I, \theta_m) : I \mapsto \{M_{ij} \in \mathbb{R}^{H \times W} | i, j = 0, 1, \ldots, G\}. \tag{2}$$

Here θ_c and $\theta_{\hat{m}}$ denote parameters for the above two mapping functions respectively. $\mathcal{F}_c$ and $\mathcal{F}_m$ are implemented with fully convolutional network and share the backbone parameters. C is the number of object categories. Each element of c_{ij} is in the value range $(0, 1)$ and indicates whether an object of the corresponding category exists at the location (i, j). H, W are the height and width of the output mask that aligns with full image content. With the semantic category and mask segmentation predicted for each spatial location, the instance segmentation result is acquired by simple post-processing of non-maximal suppression (NMS). For a detected object at location (i, j), the model tends to generate similar mask prediction at nearby locations, *e.g.,* $M_{i,j-1}$, however, $M_{i,j-1}$ is discarded by NMS and does not contribute to the final prediction.

3.2 PILOT EXPERIMENTS: MASK VOTING

In this subsection, we present pilot experiments and analysis to validate whether the SOLO model can benefit from the rich neighboring predictions by adopting the voting algorithm (Gidaris & Komodakis, 2015). We compute the mask IOU as the similarity measure to gather the neighboring mask predictions used for voting with three schemes: 1) simple averaging (**avg. v.**); 2) weighted averaging with corresponding classification score (**score v.**); 3) weighted averaging with corresponding IOU (**IOU. v.**). For each scheme, we perform grid search to find the optimal IOU threshold used for voting. As shown in Tab 1, refining the mask prediction during inference with the voting procedure can actually improve the final instance segmentation performance (*e.g.,* 0.2 AP with simple averaged voting). However, even with weighted voting, the improvement is minor and at most 0.3 in AP. The voting procedure is simple post-processing and requires no modification to a trained model, but object shape may have complex spatial layout, thus the simple voting algorithm may be suboptimal.

-	baseline	+avg. v.	+score v.	+IOU v.
AP	35.8	36.0	36.1	36.1

Table 1: Results of different voting strategies with SOLO-ResNet50. v. stands for voting.

Figure 2: Mask voting example. Voting denotes voting result. GT is the desired ground truth.

Fig. 2 gives a toy example to illustrate that voting may provide suboptimal results as it averages the candidates and is incapable to selectively combine segmentation of different parts.

4 Proposed Method

To address the above discussed limitation of simple post-training voting and more effectively leverage the neighboring predictions, we propose an aggregated learning framework. Specifically, we aggregate the neighboring mask predictions with a learnable aggregation function:

$$M_{ij} = \text{Agg}(M_{ij} \oplus \mathcal{N}(M_{ij}), \theta_a). \tag{3}$$

Here $\text{Agg}(\cdot)$ and θ_a are aggregation function and its parameter. $\oplus$ means the concatenation operation, and $\mathcal{N}(\hat{M}_{ij})$ denotes the set of neighboring predictions that will be used for predicting the final instance mask at location (i, j). We consider either 4 or 8 nearest neighbors, *i.e.*, $\{\hat{M}_{i-1,j}, \hat{M}_{i,j-1}, \hat{M}_{i+1,j}, \hat{M}_{i,j+1}\}$ or $\{\hat{M}_{i+p,j+q}\}\backslash\hat{M}_{ij}$ with $p, q \in \{-1, 0, 1\}$. More complex designs may further improve performance but are beyond the scope of this work.

The mask prediction M_{ij} is forced to be in the value range $(0, 1)$ with sigmoid activation function. To allow more freedom for learning the intermediate mask representation, we remove the sigmoid activation before aggregation. To complement the top-down instance specific information in mask representation, we also append bottom-up multi-level context feature F_c of only a few channels that summarizes both high-level semantics and low-level fine details to guide the aggregation process:

$$M_{ij} = \text{Agg}(\hat{M}_{ij} \oplus \mathcal{N}(\hat{M}_{ij}) \oplus F_c, \theta_a). \tag{4}$$

where $\hat{M}_{ij}$ is raw mask prediction without sigmoid activation, *i.e.,* the mask representations to be learned. The aggregation is differentiable and no explicit supervision is imposed for learning $\hat{M}_{ij}$.

Dynamic Aggregation In Eqn. (4), the same set of aggregation parameters θ_a is shared across all grid locations. However, such a location-invariant scheme may be suboptimal for desired adaptive combination as object shape varies significantly due to different spatial layouts, scales and context. Inspired by dynamic local filter proposed in (Jia et al., 2016), we extend our aggregation function by additionally learning a mapping function that generates position-specific aggregation parameters:

$$\mathcal{F}_a(I, \theta_a) : I \mapsto \{\theta_{ij} \in \mathbb{R}^{H \times W} | i, j = 0, 1, \ldots, G\}, \tag{5}$$

We then apply the predicted parameters for aggregating neighboring local mask representations:

$$M_{ij} = \text{Agg}(\hat{M}_{ij} \oplus \mathcal{N}(\hat{M}_{ij}) \oplus F_c, \theta_{ij}). \tag{6}$$

Such a dynamic aggregation scheme not only achieves desired self-adaptive combination, but also enable an interesting extension of AggMask discussed next.

Mask Interpolation In SOLO model, the grid resolution is limited as finer grids would result in quadratically increasing mask representations and cause remarkably increased computation and memory cost, *e.g.*, a 20×20 grid requires 400 mask predictions, while 40×40 would require 1600 predictions. However, a finer grid, especially on low-level features, is beneficial to recognizing the small objects or distinguishing the nearby objects. To address the limitation, we extend AggMask to a variant with a larger grid resolution G for classification but a smaller grid resolution G' ($G' < G$) for mask representation. The mapping functions for mask representation then becomes:

$$\mathcal{F}_{\hat{m}}(I, \theta_{\hat{m}}) : I \mapsto \{\hat{M}_{ij} \in \mathbb{R}^{H \times W} | i, j = 0, 1, \ldots, G'\}, \tag{7}$$

To generate mask predictions with finer classification grids, we simply allow nearby classification locations to share the same set of neighboring mask representations for predicting the object mask:

$$M_{ij} = \text{Agg}(\hat{M}_{\lfloor i * \frac{G'}{G} \rfloor \lfloor j * \frac{G'}{G} \rfloor} \oplus \mathcal{N}(\hat{M}_{\lfloor i * \frac{G'}{G} \rfloor \lfloor j * \frac{G'}{G} \rfloor}) \oplus F_c, \theta_{ij}). \tag{8}$$

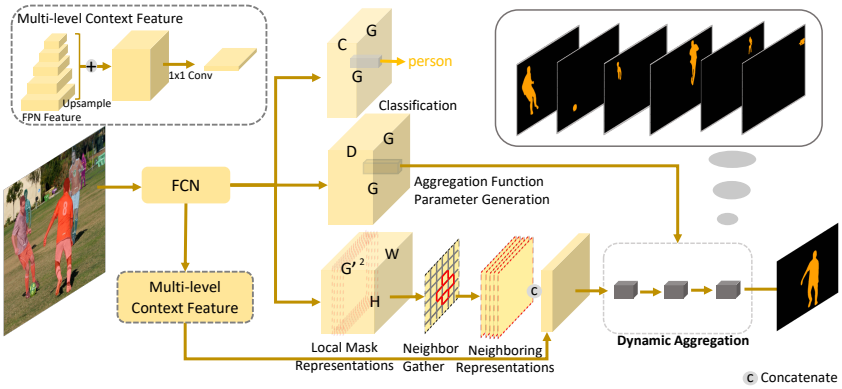

Figure 3: Network Architecture. For objects at different locations, their corresponding spatially neighboring mask representations are gathered and aggregated to form the segmentation. The framework is fully convolutional and end-to-end trained. Note with mask interpolation, the grid resolution of mask representation (G') can be smaller than classification (G) to save computation and memory.

For instance when the grid resolution is G'=10 for mask representation and G=20 for classification, both the masks $M_{15,15}$ and $M_{14,14}$ are predicted using the same set of neighboring representations of $\hat{M}_{7,7}$, but with different dynamically generated weights $\theta_{15,15}$ and $\theta_{14,14}$. This can be viewed as a spatial interpolation operation to obtain finer grid mask predictions given coarser grid mask representations, which we name *mask interpolation*. It effectively reduces the number of representations needed for a certain grid resolution, which offers two advantages. 1) We can increase a model's classification grid resolution to improve its discrimination ability for scenes of more dense or small objects, with small computation overhead; 2) we can decrease the mask representation grid resolution to save computation and memory cost with a negligble drop in performance.

Network Architecture Fig. 3 depicts the architecture of AggMask. The mapping function $\mathcal{F}_a$ shares the feature extraction network with $\mathcal{F}_c$ and $\mathcal{F}_m$. We instantiate the dynamic aggregation function with a small multi-layer convolution network, which learns arbitrary nonlinear transformation. The multi-level context information is obtained by upsampling and combining FPN features, followed by 1x1 convolution to obtain compact feature (*e.g.*, 16 channels). The whole model is fully convolutional and trained with the same loss formulation as in SOLO (Wang et al., 2020).

Relation with Existing Works Some recent works are closely related to our method. Bolya et al. (2019) developed YOLACT that linearly combines a base set of prototype features to form the mask segmentation, however, the prototypes are globally shared and the model requires a box detector to wipe the noise outside object region. While proposed method learns locally shared mask representations that only encode information of spatially neighboring objects and no box detection is required. Recently proposed CondInst (Tian et al., 2020) and SOLOv2 (Wang et al., 2020) can be viewed as extensions of YOLACT, as YOLACT genearates linear combination weights which is equivalent to 1×1 convolution without bias, while CondInst and SOLOv2 extend that to general dynamic convolution. Our AggMask is built on the perspective of leveraging neighboring prediction, which is orthogonal and can be further applied to methods like CondInst and SOLOv2. We demonstrate in experiment that SOLOv2 can be further improved when augmented with our AggMask.

5 Experiments

Dataset. We adopt COCO (Lin et al., 2014) and LVIS (Gupta et al., 2019) datasets for experiments. We report the standard COCO-style mask AP using the median of 3 runs. As AP for COCO may not fully reflect the improvement in mask quality due to its coarse ground-truth annotations (Gupta et al., 2019; Kirillov et al., 2019), we additionally report AP evaluated on the 80 COCO category subset of LVIS, which has high-quality instance mask annotations, denoted as AP*. Note we directly evaluate COCO-trained models against higher-quality LVIS annotations without training on it.

Models and Implementation Details. In addition to SOLO (Wang et al., 2019), we also examine the generalizability of proposed method on recent SOLOv2 (Wang et al., 2020). Unless otherwise specified, we use 4 neighbors for the aggregation and 16 channels for the multi-level context feature. We implement the dynamic aggregation with a 3-layer group convolution network for batched computation of different locations. The grid resolution, except for the mask interpolation experiment, is set as 40, 36, 24, 16 and 12 for different FPN levels. Other settings, including the training schedule, loss weight, and label assignment rules are the same as SOLO and SOLOv2 for fair comparison.

5.1 MAIN RESULTS

Comparison with Direct Mask Prediction Baselines. As shown in Tab. 2, AggMask significantly boosts both baseline SOLO and SOLOv2 across different backbones (*e.g.*, 2.1 AP and 1.0 AP improvement over SOLO and SOLOv2 on ResNet-50 backbone). The performance gap becomes larger when evaluating on LVIS dataset (AP^*), and especially significant for AP with high IoU (*i.e.*, AP^*_{70}, AP^*_{80} and AP^*_{90}). As high IOU AP is challenging metric that requires much higher segmentation quality, this shows the effectiveness of proposed method in improving mask segmentation quality. In terms of inference speed, AggMask introduces small additional cost compared to baselines.

Table 2: Comparison with baseline SOLO (Wang et al., 2019) and SOLOv2 (Wang et al., 2020). R50 and R101 denote ResNet-50 and ResNet-101 backbones. AP^* is mask AP evaluated on the 80 COCO categories subset of LVIS benchmark. Inference speed is measured with V100 GPU.

Model	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l	AP^*	AP^*_{50}	AP^*_{60}	AP^*_{70}	AP^*_{80}	AP^*_{90}	FPS
SOLO-R50	35.8	57.1	37.8	15.0	37.8	53.6	37.0	58.2	51.8	43.6	32.2	14.2	12.9
+AggMask	37.9	58.2	40.9	16.2	41.5	56.3	40.0	60.4	54.9	47.1	36.4	17.4	11.4
Δ-	+2.1	+1.1	+3.1	+1.2	+3.7	+2.7	+3.0	+2.2	+3.1	**+3.5**	**+4.2**	**+3.2**	-
SOLO-R101	37.1	58.1	39.5	15.6	41.0	55.5	38.6	59.7	53.5	45.8	34.3	15.6	11.5
+AggMask	38.6	58.9	41.7	16.5	42.6	57.7	41.2	61.9	55.8	48.4	37.7	18.5	10.2
Δ-	+1.5	+0.8	+2.2	+0.9	+1.6	+2.2	+2.6	+2.2	+2.3	**+2.6**	**+3.4**	**+2.9**	-
SOLOv2-R50	37.7	58.5	40.2	15.6	41.3	56.6	40.0	60.3	54.4	47.0	36.5	18.5	16.4
+AggMask	38.7	59.2	41.6	17.1	42.5	57.2	41.0	60.9	54.9	47.9	37.7	19.9	14.0
Δ -	+1.0	+0.7	+1.4	+1.5	+1.2	+0.6	+1.0	+0.6	+0.5	**+0.9**	**+1.2**	**+1.4**	-
SOLOv2-R101	38.5	59.1	41.3	17.1	42.5	56.8	41.1	61.9	55.9	48.0	37.3	18.8	13.4
+AggMask	39.4	60.0	42.3	16.8	43.6	58.5	42.3	62.3	56.9	49.7	38.9	20.5	11.5
Δ -	+0.9	+0.9	+1.0	-0.3	+1.1	+1.7	+1.2	+0.4	+1.0	**+1.7**	**+1.6**	**+1.7**	-

Incorporating Mask Interpolation. We study two instantiations to demonstrate the merit of mask interpolation. 1) **"+cls"**: increasing the classification grid resolution from [40, 36, 24, 16, 12] to [50, 40, 24, 16, 12] for each FPN level while keeping the mask grid resolution unchanged; 2) **"-mask"**: reducing the mask grid resolution from [40, 36, 24, 16, 12] to [20, 18, 12, 8, 6] while maintaining that

Table 3: Results of AggMask with mask interpolation. **+AggMask+cls** means increasing classification grid resolution; **+AggMask-mask** means reducing mask grid resolution.

Methods	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l	AP^*	AP^*_{50}	AP^*_{60}	AP^*_{70}	AP^*_{80}	AP^*_{90}	FPS
SOLO-R101	37.1	58.1	39.5	15.6	41.0	55.5	38.6	59.7	53.5	45.8	34.3	15.6	11.5
+AggMask	38.6	58.9	41.7	16.5	42.6	57.7	41.2	61.9	55.8	48.4	37.7	18.5	10.2
+AggMask+cls	**39.1**	59.8	42.1	17.1	43.2	58.0	**41.6**	62.6	57.0	48.7	38.0	18.6	10.0
+AggMask-mask	38.8	59.4	41.8	16.5	42.8	58.3	41.1	61.5	56.0	48.4	37.6	18.7	**10.9**
decoupled-SOLO	37.9	58.9	40.6	16.4	42.1	56.3	39.6	60.5	54.5	46.5	35.6	16.7	10.6
SOLOv2-R101	38.5	59.1	41.3	17.1	42.5	56.8	41.1	61.9	55.9	48.0	37.3	18.8	13.4
+AggMask	39.4	60.0	42.3	16.8	43.6	58.5	42.3	62.3	56.9	49.7	38.9	20.5	11.5
+AggMask+cls	**39.7**	60.6	43.1	17.9	43.8	58.8	**42.8**	63.3	57.6	50.3	39.4	20.7	11.2
+AggMask-mask	39.2	60.0	42.2	16.9	43.3	58.5	42.2	62.4	56.5	49.4	38.8	20.1	**12.1**

for classification. The former aims to improve the discrimination ability for small objects, while the latter aims to reduce the memory and computation cost of mask representations. Note for "-mask", the number of representations is reduced quadratically. As shown in Tab. 3, **"+cls"** improves the overall performance (38.8 to 39.1 AP), especially for small and medium-sized objects as reflected in AP_s and AP_m, while the time overhead is almost negligible (10.0 *vs.* 10.2 FPS). On the other hand, reducing the mask grid resolution improves the inference speed from 10.2 to 10.9 FPS while giving a comparable performance. A similar trend is observed for both SOLO and SOLOv2. Our "-mask" scheme also outperforms decoupled SOLO (Wang et al., 2019) in terms of both accuracy and speed. As shown in Tab. 4 and Fig. 4, for the SOLO baseline model, AggMask and its mask interpolation variant of **+cls** scheme introduce small additional computation ($\sim$1.4% and $\sim$2.3%). While **-mask** scheme can significantly reduce Flops of mask head cross FPN levels.

-	AggMask	Mask itp.	Flops	Params
SOLO -R101			422.8G	55.1M
	✓		428.3G	55.4M
	✓	**+cls**	433.8G	55.4M
	✓	**-mask**	390.5G	54.6M

Table 4: Flops & parameter count analysis. Flops is averaged over COCO `val` set.

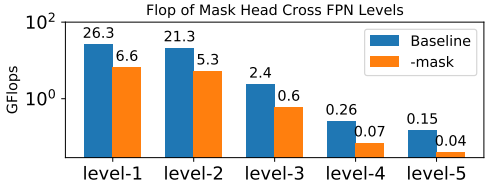

Figure 4: Flop reduction by mask interpolation. Averaged over COCO `val` set.

Comparison with state-of-the-arts. We then compare our method with state-of-the-art instance segmentation methods: PolarMask (Xie et al., 2019), YOLACT (Bolya et al., 2019), TensorMask (Chen et al., 2019b) and Mask R-CNN (He et al., 2017), on COCO `test-dev` split. As shown in Tab. 5, SOLO and SOLOv2 augmented with AggMask outperform all baselines while maintaining comparable inference speed, which well proves its effectiveness.

Table 5: Comparison with other methods on COCO `test-dev` set, all with ResNet-101 backbone.

Methods	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l	FPS
PolarMask (Xie et al., 2019)	30.4	51.9	31.0	13.4	32.4	42.8	12.3
YOLACT (Bolya et al., 2019)	31.2	50.6	32.8	12.1	33.3	47.1	23.4
TensorMask (Chen et al., 2019b)	37.1	59.3	39.4	17.4	39.1	51.6	2.6
Mask R-CNN (He et al., 2017)	37.8	59.8	40.7	**20.5**	40.4	49.3	10.0
SOLO (Wang et al., 2019)	37.8	59.5	40.4	16.4	40.6	54.2	11.5
SOLOv2 (Wang et al., 2020)	39.7	60.7	42.9	17.3	42.9	57.4	13.4
SOLO+AggMask+cls	39.5	60.6	42.9	17.4	42.5	56.1	10.0
SOLOv2+AggMask+cls	**40.6**	**61.9**	**43.8**	18.5	**43.7**	**57.6**	11.2

5.2 Model Analysis

How Does Aggregation Help? By visualizing the learned mask representations (Fig. 5), we find the activated area is larger than that of SOLO, showing our representations provide context information. Meanwhile, we observe they capture complementary information of neighboring objects, which helps to segment the object when combined. Some qualitative results are shown in Fig. 6.

Individual Design Choices. We carefully examine each proposed component to quantitatively justify our design choices (Tab. 6): 1) *Dynamic aggregation vs. static aggregation*. We compare dynamic aggregation with a static alternative that uses fixed learned weights for aggregation. It outperforms the static counterpart by large margins (37.9 *vs.* 36.5 in AP, and 40.0 *vs.* 38.5 in AP^*). This verifies our assumption that dynamically generated weights is essential to achieve desired adaptive aggregation. 2) *Number of Neighbors*. When neighbor number $|\mathcal{N}| = 0$, the model is reduced to learning a dynamic transformation from each local mask representation into the final mask, this setting already brings 1.4 AP improvement upon the baseline. Introducing neighboring mask representations further improves AP by 0.7. This suggests that neighboring representations provide complementary information, which helps refine the final instance masks when aggregated. The AP is diminishing when $|\mathcal{N}| = 8.$, possibly because the representation from far-away grid location may

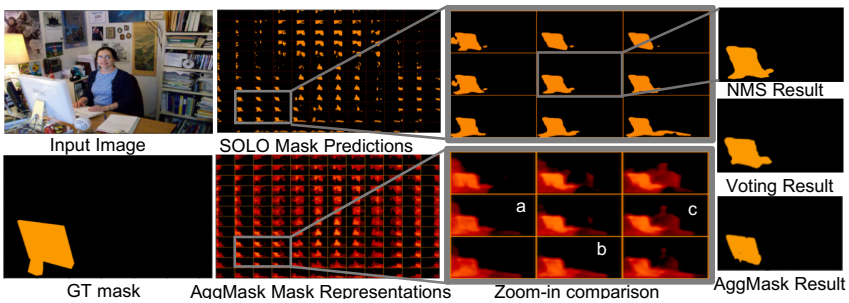

Figure 5: Visualization of mask representations. Compared to baseline SOLO mask prediction, we find our mask representations 1) have larger high-response area, indicating it attends to surrounding context; and 2) capture complementary information, *e.g.*, in addition to the monitor, 'a' has attention on the lamp, while 'b' and 'c' attends to the desk and people respectively. The quality of segmentation is higher than the baseline model and simply refining with the voting algorithm.

Figure 6: Example results of Mask R-CNN, SOLO and AggMask. Right three columns are zoom-in views of yellow rectangle areas, showing that AggMask can handle severe occlusions in crowded scenes (a), it also more accurately segment large objects benefited from the aggregation (b).

contain context that is irrelevant to the current location, especially for small-sized objects. 3) *Multi-level context information*. When the multi-level context feature is removed from the aggregation process, the performance drops from 37.9 to 37.1 in AP and 40.0 to 39.2 in AP*, demonstrating multi-level feature is beneficial for supplementing dynamic aggregation with bottom-up semantic information. If we remove mask representations and only use context information, AP drops steeply to 33.8. This means the local mask representation is crucial for segmenting the objects while the context information is complementary. 4) *Number and size of dynamic convolution kernels*. For the dynamic aggregation network, we find the best performance is achieved with 3 layers; the improvement is diminishing with more layers. For the convolution kernel, 3×3 is better than 1×1 (37.9 vs 37.7 in AP), meaning a larger receptive field for the aggregation layers generates better results.

Table 6: Model analysis results. "base" denotes baseline SOLO-R50 model. Due to space limit, we only report AP and AP* results, please refer to Tab. 7 in Appendix for detailed results.

	base.	Aggregation		$\lvert\mathcal{N}\rvert$			Multi-level context			#Conv layers				Conv kernel	
		dynamic	static	0	4	8	w/	w/o	only	1	2	3	4	1x1	3x3
AP	35.8	37.9	36.5	37.2	37.9	37.7	37.9	37.1	33.8	37.1	37.6	37.9	37.8	37.7	37.9
AP*	37.0	40.0	38.5	39.3	40.0	39.8	40.0	39.2	34.6	39.3	39.8	40.0	39.7	39.5	40.0

6 Conclusion

In this work, we explore a novel aggregated learning framework that leverages the rich neighboring predictions based on the recent SOLO model, the framework in turn enables the model to generate interpretable location-aware mask representations that encodes context objects information. It is validated that the simple method significantly improves the segmentation quality of baseline models. We believe our findings provide useful insight for future research on instance segmentation.

REFERENCES

Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, et al. Interaction networks for learning about objects, relations and physics. In *Advances in neural information processing systems*, pp. 4502–4510, 2016.

Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: real-time instance segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9157–9166, 2019.

Hao Chen, Kunyang Sun, Zhi Tian, Chunhua Shen, Yongming Huang, and Youliang Yan. Blendmask: Top-down meets bottom-up for instance segmentation. *arXiv preprint arXiv:2001.00309*, 2020.

Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4974–4983, 2019a.

Liang-Chieh Chen, Alexander Hermans, George Papandreou, Florian Schroff, Peng Wang, and Hartwig Adam. Masklab: Instance segmentation by refining object detection with semantic and direction features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4013–4022, 2018.

Xinlei Chen, Ross Girshick, Kaiming He, and Piotr Dollár. Tensormask: A foundation for dense object segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2061–2069, 2019b.

Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3150–3158, 2016.

Cheng-Yang Fu, Mykhailo Shvets, and Alexander C Berg. Retinamask: Learning to predict masks improves state-of-the-art single-shot detection for free. *arXiv preprint arXiv:1901.03353*, 2019.

Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980.

Spyros Gidaris and Nikos Komodakis. Object detection via a multi-region and semantic segmentation-aware cnn model. In *Proceedings of the IEEE international conference on computer vision*, pp. 1134–1142, 2015.

Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, pp. 1024–1034, 2017.

Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *European Conference on Computer Vision*, pp. 297–312. Springer, 2014.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. In *Advances in Neural Information Processing Systems*, pp. 667–675, 2016.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. *arXiv preprint arXiv:1912.08193*, 2019.

Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

Youngwan Lee and Jongyoul Park. Centermask: Real-time anchor-free instance segmentation. *arXiv preprint arXiv:1911.06667*, 2019.

Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2359–2367, 2017.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

Lu Qi, Xiangyu Zhang, Yingcong Chen, Yukang Chen, Jian Sun, and Jiaya Jia. Pointins: Point-based instance segmentation. *arXiv preprint arXiv:2003.06148*, 2020.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pp. 91–99, 2015.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.

Konstantin Sofiiuk, Olga Barinova, and Anton Konushin. Adaptis: Adaptive instance selection network. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 7355–7363, 2019.

Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. *arXiv preprint arXiv:2003.05664*, 2020.

Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. Solo: Segmenting objects by locations. *arXiv preprint arXiv:1912.04488*, 2019.

Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. Solov2: Dynamic, faster and stronger. *arXiv preprint arXiv:2003.10152*, 2020.

Enze Xie, Peize Sun, Xiaoge Song, Wenhai Wang, Xuebo Liu, Ding Liang, Chunhua Shen, and Ping Luo. Polarmask: Single shot instance segmentation with polar representation. *arXiv preprint arXiv:1909.13226*, 2019.

Rufeng Zhang, Zhi Tian, Chunhua Shen, Mingyu You, and Youliang Yan. Mask encoding for single shot instance segmentation. *arXiv preprint arXiv:2003.11712*, 2020.

A APPENDIX

A.1 DETAILED RESULTS OF MODEL ANALYSIS

The detailed results of model analysis (*i.e.*, Tab. 6 of the main submission) are given in Tab. 7.

Table 7: Detailed results of model analysis (*i.e.*, Tab. 6 of main submission).

	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l	AP^*	AP^*_{50}	AP^*_{60}	AP^*_{70}	AP^*_{80}	AP^*_{90}
Baseline	35.8	57.1	37.8	15.0	37.8	53.6	37.0	58.2	51.8	43.6	32.2	14.2
agg-D	37.9	58.2	40.9	16.2	41.5	56.3	40.0	60.4	54.9	47.1	36.4	17.4
agg-S	36.5	56.4	39.0	15.2	39.8	54.8	38.5	59.6	53.7	45.8	34.3	15.5
$\|\mathcal{N}\|$-0	37.2	57.2	40.1	15.3	40.6	56.0	39.3	60.0	53.8	46.8	36.0	17.0
$\|\mathcal{N}\|$-4	37.9	58.2	40.9	16.2	41.5	56.3	40.0	60.4	54.9	47.1	36.4	17.4
$\|\mathcal{N}\|$-8	37.7	57.9	40.6	15.6	41.4	55.9	39.8	60.3	54.5	46.7	36.2	17.6
multi-level-context-w	37.9	58.2	40.9	16.2	41.5	56.3	40.0	60.4	54.9	47.1	36.4	17.4
multi-level-context-w/o	37.1	57.0	40.1	15.0	40.7	56.1	39.2	59.7	54.0	46.2	35.6	16.7
multi-level-context-only	33.8	52.9	36.1	14.2	37.5	49.1	34.6	53.9	48.2	40.6	30.1	14.3
conv-layer-1	37.1	57.2	40.0	15.5	40.1	56.2	39.3	59.8	54.4	46.1	35.5	16.9
conv-layer-2	37.6	58.1	40.4	15.2	41.0	56.3	39.8	59.8	54.2	46.6	35.8	18.0
conv-layer-3	37.9	58.2	40.9	16.2	41.5	56.3	40.0	60.4	54.9	47.1	36.4	17.4
conv-layer-4	37.8	58.4	40.5	16.1	41.4	56.7	39.7	59.7	54.0	46.8	35.8	17.8
conv-kernel-1x1	37.7	58.1	40.6	16.0	41.2	56.3	39.5	59.8	54.2	46.4	35.4	17.7
conv-kernel-3x3	37.9	58.2	40.9	16.2	41.5	56.3	40.0	60.4	54.9	47.1	36.4	17.4