PolarNeXt: Rethink Instance Segmentation with Polar Representation

Jiacheng Sun[1], Xinghong Zhou[1], Yiqiang Wu[1], Bin Zhu[1], Jiaxuan Lu[2], Yu Qin[1], Xiaomao Li[1,*]
[1]Shanghai University, [2]Shanghai Artificial Intelligence Laboratory
{sunjc,jkchou,yiqiangwu,zhubin01,qy21722148,lixiaomao}@shu.edu.cn
lujiaxuan@pjlab.org.cn

Abstract

One of the roadblocks for instance segmentation today is heavy computational overhead and model parameters. Previous methods based on Polar Representation made the initial mark to address this challenge by formulating instance segmentation as polygon detection, but failed to align with mainstream methods in performance. In this paper, we highlight that Representation Errors, arising from the limited capacity of polygons to capture boundary details, have long been overlooked, which results in severe performance degradation. Observing that optimal starting point selection effectively alleviates this issue, we propose an Adaptive Polygonal Sample Decision strategy to dynamically capture the positional variation of representation errors across samples. Additionally, we design a Union-aligned Rasterization Module to incorporate these errors into polygonal assessment, further advancing the proposed strategy. With these components, our framework PolarNeXt achieves a remarkable performance boost of over 4.8% AP compared to other polar-based methods. PolarNeXt is markedly more lightweight and efficient than state-of-the-art instance segmentation methods, while achieving comparable segmentation accuracy. We expect this work will open up a new direction for instance segmentation in high-resolution images and resource-limited scenarios. Codes can be found at https://github.com/Sun15194/PolarNeXt.

1. Introduction

Object detection and instance segmentation are fundamental tasks in computer vision, aiming at identifying specific objects within images. Object detection [26, 27, 33], also known as box detection, localizes objects using rectangular boxes, which are efficient but coarse-grained. In comparison, instance segmentation offers additional geometric information, such as shape and boundary. Mainstream methods define instance segmentation as pixel-wise classification [7, 14, 34] or iterative deformation [9, 25, 39]. Despite the finer granularity, these methods introduce heavy computational overhead and model parameters, which restricts their applicability in high-resolution images or resource-limited scenarios [11].

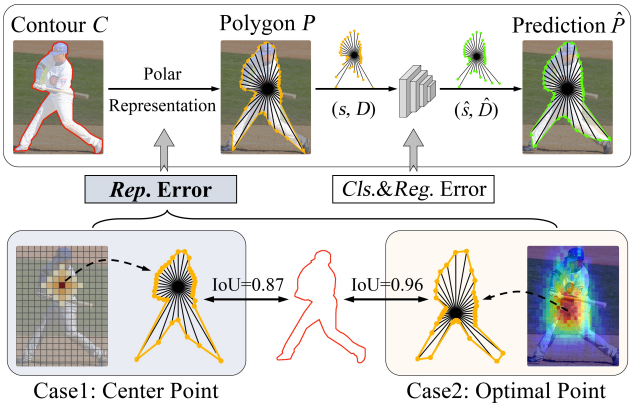

Figure 1. Illustration of the composition of prediction errors in polar-based methods. Classification and Regression (*Cls.*&*Reg.*) Errors are produced during network optimization, which reflects the deviation from starting point classification ($s \rightarrow \hat{s}$) and dense distance regression ($D \rightarrow \hat{D}$). The ignored Representation (*Rep.*) Error arises from the missing boundary details of instance contour C in Polar Representation, as exemplified by the two cases below. Obviously, the center point does not necessarily coincide with the optimal point, from which the assembled polygon with minimal *Rep.* Error can be predicted.

To combine the efficiency of detection with the granularity of segmentation, a pioneering attempt is **Polar Representation**. As shown at the top of Fig. 1, Polar Representation reconstructs the instance contour into a bounding polygon, assembled by dense rays emitted uniformly from a starting point. Building on this concept, typical polar-based methods [36–38] formulate instance segmentation as starting point classification and dense distance regression. These methods can be instantiated on one-stage object detectors [26, 33], directly inheriting their advantages in inference speed and model complexity. However, Polar Representation has not become a fundamental paradigm for in-

*Corresponding author.

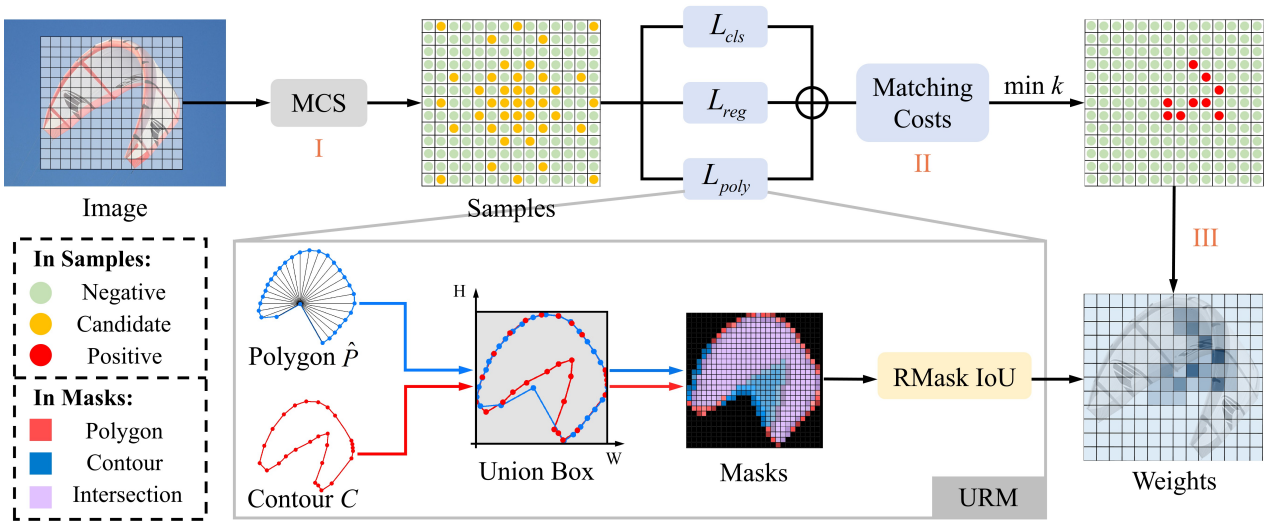

Figure 2. An overview of the proposed Adaptive Polygonal Sample Decision (APSD). The three stages of APSD are labeled as: I.Sampling Stage, II.Matching Stage, and III.Weighting Stage. The Union-aligned Rasterization Module (URM) takes the predicted polygons $\hat{P}$ of the candidates selected by Multi-layer Center Sampling (MCS) as input, and outputs their RMask IoU with the instance contours C. RMask IoU functions in both matching costs and sample weighting.

stance segmentation, due to the significant performance gap between polar-based methods and other advanced methods.

In this work, we highlight that an additional prediction error has been ignored in Polar Representation, which results in the performance degradation of polar-based methods. As illustrated at the bottom of Fig. 1, from different starting points, the assembled polygons exhibit varying degrees of deviation with their corresponding contours. This deviation, termed Representation (*Rep.*) Error, inevitably occurs in Polar Representation, where the boundary details of contours are complex for polygons to reflect, especially in disconnected or concave regions. However, existing training pipelines in polar-based methods only account for the Classification and Regression (*Cls.&Reg.*) Errors generated during network optimization. Their label assignment and sample weighting, collectively referred to as sample decisions, rely entirely on the handcrafted center prior, which fails to capture the positional variation of representation errors (as proven in Sec. 3.2(1)). Furthermore, even though a natural idea is to dynamically select the optimal starting point with minimal representation error, a reliable metric to incorporate this error into polygonal assessment is still lacking. As the only available metric, Polar IoU [36] exhibits assessment blindness to representation errors (as verified in Sec. 3.2(2)), which is too coarse-grained to support this idea.

To overcome these challenges, we propose PolarNeXt, a highly lightweight yet effective polar-based framework for real-time instance segmentation. The training pipeline of this framework includes an Adaptive Polygonal Sample Decision (APSD) strategy, enabling optimal starting point selection during inference. APSD assigns positive labels and higher weights to the candidate samples based on their matching costs, which account for the comprehensive errors between predictions and ground truths. To eliminate the assessment blindness to representation errors, a Union-aligned Rasterization Module (URM) is designed for more accurate polygonal assessment. The paired polygon and contour are aligned within their Union Box and then converted into rasterized masks for RMask IoU measurement. Here, RMask IoU serves as a new metric to evaluate the actual polygonal overlap, forming the core element of sample decisions and loss functions. Benefitting from these components, PolarNeXt achieves a remarkable improvement of over 4.8% AP compared to other polar-based methods. To the best of our knowledge, this is the first polar-based method competitive with state-of-the-art instance segmentation methods in performance. Notably, PolarNeXt requires only half or even less of their computational cost and inference time, which redemonstrates the superiority of Polar Representation in instance segmentation. The main contributions of this work can be summarized as follows:

- We conduct a thorough analysis of the implementation challenges of Polar Representation in instance segmentation, and reveal that the performance limitations of existing polar-based methods stem from their neglect of the representation error.
- We propose a new polar-based instance segmentation framework called PolarNeXt, including an APSD strategy to dynamically account for representation errors across samples and a URM module that eliminates the assessment blindness to these errors using RMask IoU.

- We fully leverage the advantages of Polar Representation in instance segmentation tasks. Without bells and whistles, PolarNeXt is markedly more lightweight and efficient than other mainstream methods, while aligning with them in terms of segmentation performance.

2. Related Work

Instance Segmentation. Mainstream instance segmentation methods can be classified into two categories based on their representation, *i.e.*, mask-based methods and contour-based methods. **Mask-based methods** treat instance segmentation as pixel-wise classification, in which binary masks are predicted within their Regions of Interest (RoIs). As a pioneering work, Mask R-CNN [14] introduces a detect-then-segment paradigm, built on the two-stage detector [27], with box proposals serving as the RoIs for pixel-by-pixel segmentation. Nowadays, state-of-the-art benchmark results are still held by the mask-based two-stage methods [2, 3, 15] constructed on strong detection baselines. Additionally, to reduce reliance on bounding boxes, one-stage methods [32, 34, 35] treat the entire image as the RoI, predicting masks by the assembly of encoded mask vectors and mask kernels from dynamic convolutions. Other mask-based methods focus on the high-quality boundary [17, 20], or the real-time inference [1, 7]. **Contour-based methods** take a different paradigm, formulating instance segmentation as iterative deformation based on the active contour mechanism of Snake [16]. Deep Snake [25] establishes the first multi-stage deformation network, which progressively refines the pre-predicted initial contours using their vertex features. Based on this baseline, further optimization strategies have been explored, such as Segment-wise Matching Scheme [24], Dynamic matching Loss [39], and Multi-scale Contour Refinement [9]. However, the high computational cost introduced by pixel-wise classification and iterative deformation limits the applicability and prevalence of these methods in high-resolution images or resource-limited scenarios [11]. To solve this problem, we bring back Polar Representation and redemonstrate its superiority in instance segmentation. Building on Polar Representation, our proposed PolarNeXt is markedly more lightweight and efficient than the aforementioned methods, while remaining competitive in segmentation performance.

Polar Representation. Polar Representation initially appears in some interdisciplinary fields, such as automatic building segmentation [6] and medical cell detection [30]. Subsequently, ESE-Seg [38] migrates Polar Representation into instance segmentation tasks, introducing Chebyshev polynomial fitting to shorten the distance vectors and resist noise. PolarMask [36] and its variant [37] make the initial mark of polar-based methods on instance segmentation benchmarks. They maximize the advantage of the center prior by modified Center Sampling and Polar Centerness, along with a proposed Polar IoU metric for polygonal assessment. However, the representation error introduced by Polar Representation has been ignored in these polar-based methods, which results in severe performance degradation. To address this, we propose an APSD strategy to dynamically balance representation errors across samples, in conjunction with a URM module that incorporates these errors into polygonal assessment. Together, these components bring a notable performance boost to PolarNeXt.

3. Methods

In this work, PolarMask [36], a representative polar-based method, is used as a case study for Polar Representation. The design principle and training pipeline of PolarMask are reviewed in Sec. 3.1. Under this pipeline, the effectiveness of its sample decisions and polygonal assessment is investigated in Sec. 3.2. Finally, a novel polar-based framework called PolarNeXt is detailed in Sec. 3.3 to advance Polar Representation in instance segmentation tasks.

3.1. Preliminary

PolarMask defines instance segmentation as starting point classification and dense distance regression. As illustrated in Fig. 1, for a given instance contour C, bounding polygon $P = \psi(s, D)$ can be constructed through Polar Representation ψ, using a distance set $D = \{d_i | \, i = 1, ..., m\}$ from a starting point s. The starting point s and distance set D serve as supervision signals for the network, which then classifies a starting point $\hat{s}$ and regresses an equal-size set $\hat{D} = \{\hat{d}_i \big| \, i = 1, ..., m\}$ to construct predicted polygon $\hat{P} = \psi(\hat{s}, \hat{D})$. In the training pipeline of PolarMask, the center prior fully dictates label assignment and sample weighting, given that samples closer to the center tend to have richer receptive field and lower regression difficulty [19]. First, Center Sampling is applied on a single FPN [22] layer, directly assigning positive labels to the samples around instance centers. Second, Polar Centerness, which quantifies the proximity to the center, weights these positive samples to modulate their contribution to loss functions. Furthermore, due to the lack of polygon-specific IoUs, Polar IoU is proposed as a metric for polygonal assessment. It compares the consistency between target distance set D and predicted distance set $\hat{D}$ to approximate the overlap between P and $\hat{P}$. Additionally, Polar IoU is transformed into Polar IoU Loss to supervise distance regression. More details are provided in supplementary materials.

3.2. Investigation

In this section, we identify that two challenges lie in implementing Polar Representation for instance segmentation:

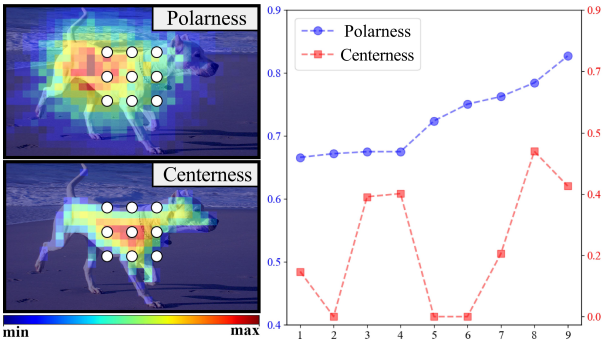

Figure 3. Illustration of the divergence between Polarness and Centerness. On the left, heatmaps of these two attributes are visualized to reveal the distributional divergence, with positive samples (9 white points) selected by Center Sampling marked. Then, attribute values of these samples are plotted to verify the quantitative divergence on the right, where samples are sorted by ascending Polarness.

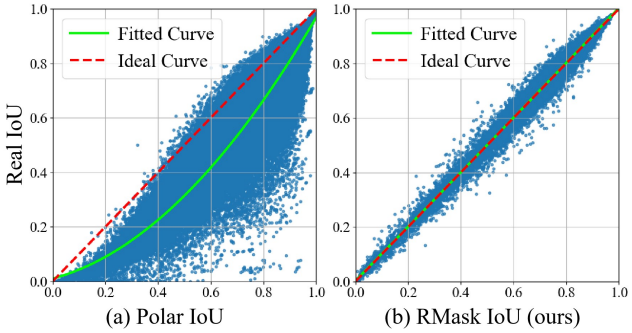

Figure 4. Comparison between Polar IoU and our proposed RMask IoU. The green solid lines denote the curves fitted by scatter points, while the red dashed lines indicate the desired linear correlation trend. Real IoU is the polygon-specific IoU between the predicted polygon and its corresponding instance contour, calculated using Weiler-Atherton algorithm [10].

(1) Diminished Effectiveness of Center Prior. The center prior plays a diminished role in sample decisions, which fails to capture the positional variation of representation errors. To validate this, a comparative experiment is conducted in Fig. 3, illustrating the divergence between the center prior and representation error. For a clear comparison, we define Polarness (Plr) as an inverse measure of Representation Error (RE), where $Plr = 1 - RE$. Likewise, normalized Polar Centerness is introduced to quantify the influence of the center prior. Experimental results show that Polarness presents no observable correlation with the distribution pattern of Centerness, while Centerness does not increase monotonically on these samples sorted by ascending Polarness. Accordingly, the samples selected by center sampling do not necessarily correspond to lower representation errors, and Centerness also fails to accurately modulate the weights of high-quality samples. Based on these observations, we conclude that the sample decisions entirely reliant on the center prior are less applicable in polar-based methods, as some suboptimal samples in center regions receive excessive attention.

(2) Assessment Blindness of Polar IoU. As the only available metric, Polar IoU exhibits assessment blindness to representation errors. This limitation arises because Polar IoU evaluates the consistency between target polygon P and predicted polygon $\hat{P}$ by comparing their distance sets, entirely independent of instance contour C. To explore the impact of this blindness on polygonal assessment, a scatter plot is drawn to examine the effectiveness of Polar IoU. As shown in Fig. 4(a), the scatter points show a significant departure from the expected linear trend, with substantial noise introduced. For example, when Real IoU reaches 0.4, Polar IoU fluctuates substantially, ranging irregularly from 0.4 to 0.9. Based on these observations, we conclude that Polar IoU fails to function as a reliable metric for polygonal assessment.

3.3. PolarNeXt

To overcome these two challenges, a new framework named **PolarNeXt** is proposed, which is expected to become the next level of polar-based methods for instance segmentation. The training pipeline of PolarNeXt includes two tailored components for Polar Representation: Adaptive Polygonal Sample Decision (APSD) and Union-aligned Rasterization Module (URM). As illustrated in Fig. 2, APSD samples a wide range of candidates around instances on the input image. Then, those candidates with minimal matching costs are assigned positive labels and weighted by the quality scores of their predicted polygons. For polygonal assessment, URM aligns the paired polygon and contour within their Union Box, and converts them into rasterized masks for overlap measurement. The RMask IoU output by URM incorporates representation errors, serving as precise guidance in both label assignment and sample weighting. The detailed design principles of our proposed methods are outlined below.

Union-aligned Rasterization Module. To eliminate the assessment blindness to representation errors, a straightforward idea is to bridge the predicted polygon $\hat{P}$ with its instance contour C and measure their polygonal overlap. However, traditional polygon-specific IoU algorithms, such as Weiler-Atherton [10], are impractical for parallel processing, due to uncertain vertex counts and inevitable recursive operations. Inspired by 3D renderers [12, 18, 23], our solution is to convert $\hat{P}$ and C from vertex-set format into rasterized-mask format for mask-level overlap measurement, but an additional challenge lies in their alignment at a fixed resolution, given that no pre-predicted RoIs are available in one-stage detectors.

As shown in Fig. 2, we propose a Union-aligned Rasterization Module (URM) which employs Union Box, the minimal bounding box enclosing all the vertices of $\hat{P}$ and C, as a reference for coordinate alignment. Concretely, assuming the vertex sets of $\hat{P}$ and C are symbolized by $V_{\hat{P}}$ and V_C, with a desired mask resolution for rasterization specified as $H \times W$. The coordinate (v_x, v_y) of a vertex v can be adjusted by the union set $U = V_{\hat{P}} \cup V_C$ as follows:

$$\begin{pmatrix} v_x \\ v_y \end{pmatrix} = \begin{pmatrix} \frac{v_x - \min\limits_{u \in U} u_x}{\max\limits_{u \in U} u_x - \min\limits_{u \in U} u_x} \times W \\ \frac{v_y - \min\limits_{u \in U} u_y}{\max\limits_{u \in U} u_y - \min\limits_{u \in U} u_y} \times H \end{pmatrix}. \quad (1)$$

Then, the differentiable rasterizer in [18] is introduced to transform the aligned vertex sets into rasterized masks. For pixel $p(x, y)$ of a mask, where $0 \leq x \leq W$ and $0 \leq y \leq H$, two attributes are used to model its contribution value $M[x, y]$ to the mask: (1) the binary inside-outside state $IO(V, p)$ of pixel p relative to vertex set V, and (2) the closest distance $CD(V, p)$ from pixel p to vertex set V. The contribution value $M[x, y]$ is formulated as shown below:

$$M[x, y] = sigmoid(\frac{IO(V, p) \times CD(V, p)}{\tau}), \quad (2)$$

where τ is a hyperparameter for sharpness. Finally, two rasterized masks $M_{\hat{P}}$ and M_C with the same resolution $H \times W$ are produced, and their mask-level IoU, called RMask IoU, can be calculated as follows:

$$\text{RMask IoU} = \frac{2 \sum\limits_{i=1}^{H} \sum\limits_{j=1}^{W} \mathrm{M}_C[\mathrm{i}, \mathrm{j}] \cdot \mathrm{M}_{\hat{P}}[\mathrm{i}, \mathrm{j}]}{\sum\limits_{i=1}^{H} \sum\limits_{j=1}^{W} \mathrm{M}_C[\mathrm{i}, \mathrm{j}] + \sum\limits_{i=1}^{H} \sum\limits_{j=1}^{W} \mathrm{M}_{\hat{P}}[\mathrm{i}, \mathrm{j}]}. \quad (3)$$

As shown in Fig. 4(b), RMask IoU accurately reflects the actual overlap between polygons and contours. Further, we perform a simple deformation on RMask IoU to obtain a new loss function, RMask IoU Loss, for polygonal external constraints:

$$\text{RMask IoU Loss} = 1 - \text{RMask IoU}. \quad (4)$$

Adaptive Polygonal Sample Decision. The essence of sample decisions is to focus more attention on high-quality samples in each iteration. However, as an additional error introduced in Polar Representation, the representation error is susceptible to multiple factors, such as contour connectivity and convexity, which makes it challenging to be captured using the fixed priors. As shown in Fig. 2, our solution is to construct an optimal matching problem, in which positive labels and higher weights are assigned to those samples with lower matching costs. Specifically, we propose an Adaptive Polygonal Sample Decision (APSD) strategy to dynamically balance and minimize various types of errors across samples during training. For clarity, we divide this strategy into three stages:

(1) Sampling Stage. In this stage, a group of candidates is sampled from the multi-scale grids of FPN layers. Here, Center Sampling is still adopted to maintain the strong receptive field for candidates, with a slight adjustment. To expand the range of candidate selection, we put forward Multi-layer Center Sampling (MCS), which applies Center Sampling across all FPN layers, rather than sampling with an increasing radius on a single layer (as proven ineffective in Sec. 4.3).

(2) Matching Stage. The purpose of this stage is to select positive or negative labels for candidates based on their matching costs. The Matching cost is defined as a weighted summation of classification loss L_{cls}, regression loss L_{reg}, and polygon loss L_{poly}, which together evaluate the optimized cost from the prediction to ground truth. In this work, we employ Focal Loss [28], Polar IoU Loss [36], and the proposed RMask IoU Loss as classification loss, regression loss, and polygon loss, respectively. For a given instance, the matrix of matching costs $Cost = \{c_i | i = 1, ..., n\}$ for n candidates can be formulated as follows:

$$Cost = \lambda_{cls} L_{cls} + \lambda_{reg} L_{reg} + \lambda_{poly} L_{poly}, \quad (5)$$

where λ_{cls}, λ_{reg} and λ_{poly} stand for the loss coefficients. Afterward, a fixed number k of positive labels are assigned to the candidates with minimal costs, and the others are negative:

$$X_i = \begin{cases} Positive, & c_i \in \text{top-}k_{\min}(c_1, c_2, ..., c_n) \\ Negative, & c_i \notin \text{top-}k_{\min}(c_1, c_2, ..., c_n), \end{cases} \quad (6)$$

where X_i refers to the label assigned to the *i*-th candidates. Notably, if a sample is matched by two instances simultaneously, the one with the largest area will be chosen.

(3) Weighting Stage. In the final stage, each positive sample is provided with a quality score, which modulates its contribution to loss functions. Our approach here is straightforward but effective, replacing the heuristic Centerness with the RMask IoU output by URM. Compared to Centerness, RMask IoU treats each positive sample equally, especially those high-quality samples located outside the contour, whose weights are zeroed out by Centerness (e.g., samples 5 and 6 in Fig. 3).

Loss Function. The overall loss function L consists of four parts:

$$L = \lambda_{cls} L_{cls} + \lambda_{reg} L_{reg} + \lambda_{poly} L_{poly} + \lambda_{sre} L_{sre}, \quad (7)$$

where L_{sre} denotes the Cross-Entropy Loss in PolarMask used to supervise the Centerness head. In this case, we replace the original supervision signals with RMask IoU in the Centerness head.

Type	Method	Backbone	Epoch	Size	AP	AP_{50}	AP_{75}	FPS	Parms↓	FLOPs↓
Contour	DeepSnake [25]	DLA-34	160	512	30.3	-	-	24	-	-
	E2EC [39]	DLA-34	150	512	33.8	52.9	32.8	35	29.80	121
	PolySnake [9]	DLA-34	250	512	34.4	-	-	18	-	-
	DANCE [24]	R50-FPN	36	800	36.8	58.5	39.0	16	44.60	274
Mask	Mask R-CNN [14]	R50-FPN	36	800	37.5	59.1	40.2	33	44.40	240
	Cascade R-CNN [2]	R50-FPN	36	800	38.4	59.7	41.5	23	77.33	1804
	HTC [3]	R50-FPN	20	800	38.7	60.1	41.9	17	80.26	1735
	MS R-CNN [15]	R50-FPN	24	800	36.8	56.7	39.7	30	60.74	279
	PointRend [17]	R50-FPN	36	800	38.4	59.5	41.5	22	60.22	266
	CondInst [32]	R50-FPN	36	800	36.2	57.2	38.7	29	34.17	308
	SOLO [34]	R50-FPN	36	800	36.4	57.9	38.9	25	36.31	332
	SOLOv2 [35]	R50-FPN	36	800	38.4	59.5	41.2	35	46.60	227
	YOLACT [1]	R50-FPN	55	800	29.0	48.7	29.8	47	35.30	237
	SparseInst [7]	R50-FPN	146	800	33.2	53.4	34.7	59	32.80	201
Polar	ESE-Seg [38]	R50-FPN	300	800	21.6	48.7	22.4	38	-	-
	PolarMask [36]	R50-FPN	36	800	31.3	52.5	32.3	42	34.74	264
	PolarNeXt (ours)	**R50-FPN**	**36**	**800**	**36.1**	**59.7**	**37.3**	**49**	**32.36**	**186**

Table 1. Experimental results on COCO test-dev. "Size" refers to the length of the shortest edge of input images. "↓" means smaller values are preferred.

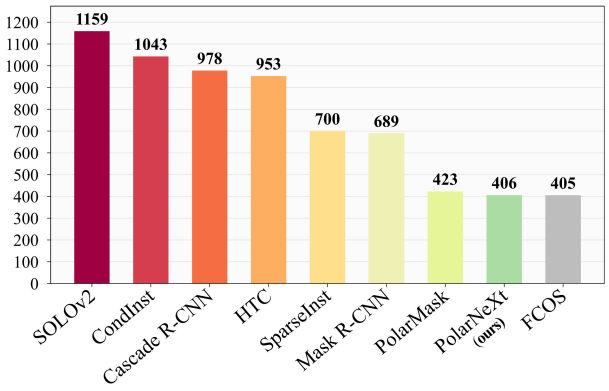

Figure 5. Comparison of Memory Usage (MB) for inference. During inference on COCO test-dev images, the peak memory usage is recorded as the final result of each method.

Method	AP	AP_{50}	FPS	Mems↓
Mask R-CNN [14]	31.5	58.6	16	1968
SOLOv2 [35]	27.2	48.5	17	2076
CondInst [32]	33.3	59.1	15	2319
PolarMask [36]	27.4	52.3	28	575
PolarNeXt (ours)	**30.7**	**58.8**	**31**	**566**

Table 2. Experimental results on Cityscapes val set. "Mems" stands for the memory usage (MB).

4. Experiments

4.1. Experiments and Implementation Details

Experiment Settings. The experiments are conducted on MS COCO [21] benchmark and Cityscapes [8] dataset. MS COCO contains 80 classes with 115k training, 5k validation, and 20k testing images. Cityscapes has 2795 training images from 8 classes and another 500 images with high-quality annotations for validation. In terms of evaluation metrics, the accuracy of segmentation is evaluated by the standard AP metric, denoted as mask AP. Furthermore, mask AP can be divided into AP_{50}/AP_{75} based on different IoU thresholds, as well as AP_S/AP_M/AP_L for small/medium/large objects. All inference experiments are conducted on a single NVIDIA 4090D GPU, where FPS, Parms, and FLOPs are calculated separately for inference speed, model complexity, and computational overhead. Notably, in all our experiments, TensorRT or FP16 is not used for acceleration.

Implementation Details. PolarNeXt is instantiated on the PolarMask network, using a backbone ResNet-50 [13] pre-trained on ImageNet [29], and removed the auxiliary box branch. All models are implemented on the MMDetection toolbox [4], trained on 2 GPUs with 2 images per GPU, and optimized with SGD. Weight decay and momentum are configured to 0.0001 and 0.9, respectively. The base learning rate is set to 0.01 without any decay scheme. For a fair comparison, following common practice, data augmentation only contains random flip and scale jitter unless specified. In APSD, the radius of Center Sampling for candidate selection at each FPN layer is set to 1.5 and the number k of assigned positive labels is set to 9 for each instance. Loss coefficients λ_{cls}, λ_{reg}, λ_{poly}, and λ_{sre} are all empirically set to 1.0. Following the rasterizer configuration in [18], we set the desired mask resolution to 64×64 and the rasterization sharpness to $\tau = 0.1$.

URM			APSD	AP
weight	loss	cost		
				29.1
✓				30.3 (+1.2)
✓	✓			31.0 (+1.9)
			✓	28.6 (−0.5)
✓	✓		✓	30.7 (+1.6)
✓	✓	✓	✓	**33.9 (+4.8)**

Table 3. Ablation studies on URM and APSD. "weight" means whether Centerness is replaced by RMask IoU for sample weighting. "loss" refers to the use of RMask IoU Loss in loss functions. "cost" indicates the incorporation of RMask IoU Loss into matching costs.

Polar IoU Loss	AP	AP_S	AP_M	AP_L
w/o.	32.8	14.3	35.1	48.5
w/.	**33.9**	**16.4**	**35.9**	**49.4**

Table 4. Feasibility analysis of removing Polar IoU Loss. The symbols "w/o." and "w/." indicate whether the Polar IoU Loss is employed.

4.2. Main Results

MS COCO. Following common practice, we evaluate PolarNeXt on the MS COCO benchmark and compare test-dev results to some state-of-the-art models in Tab. 1. In terms of computational cost, PolarNeXt is more lightweight than the other models with R50-FPN backbone, only requiring 32.8 MB Parameters and 186 GB FLOPs. Even when compared to the contour-based methods using DLA-34 backbone, it remains on par. In terms of inference speed, PolarNeXt is second only to the prevalent SpareInst, but it achieves higher accuracy (36.1% *vs.* 33.2% AP) with a shorter training schedule (36 *vs.* 146 epochs). In terms of segmentation accuracy, PolarNeXt delivers competitive results compared to other more complex methods. Particularly, without bells and whistles, PolarNeXt achieves a 4.8% mask AP improvement over PolarMask, redemonstrating the potential of Polar Representation in instance segmentation. Moreover, we visualize the memory usage of some representative methods during inference in Fig. 5. Obviously, PolarNeXt shows its consistency with the classical object detector FCOS, with memory usage significantly lower than other instance segmentation methods, only half or even less.

Cityscapes. As an extended comparative experiment in Tab. 2, Cityscapes dataset, characterized by high-resolution images, is utilized to examine the robustness and generalization ability of our model. Notably, all models are directly trained on the Cityscapes training set under 64 epochs, without COCO initialization. Since fragmented instances frequently occur in Cityscapes, the Multi-component Detection Strategy suggested in [25] is applied to PolarNeXt and PolarMask, leading to approximately a 1.5% AP improvement. Under the increased resolution of input images (from 800×1333 to 1024×2048), mask-based methods suffer from the heavy memory usage introduced by dense pixel classification, which is nearly four times greater than that of polar-based methods. Meanwhile, the speed advantage of polar-based methods becomes more apparent, requiring only half the inference time compared to mask-based methods. As expected, PolarNeXt maintains a 3.3% AP improvement over PolarMask, further demonstrating the effectiveness of our proposed method.

Candidate Selection	Layer	AP	*T.T.*
Center Sampling (*s.r.*=1.5)	single	31.0	13h
Center Sampling (*s.r.*=2.0)		30.6 (−0.4)	14h
All Grid Points	multi	-	-
Points within Boxes		33.1 (+2.1)	28h
Points within Contours		33.0 (+2.0)	25h
Center Sampling (*s.r.*=1.5)		**33.9 (+2.9)**	**16h**

Table 5. The effectiveness of some approaches for candidate selection. "single" denotes selecting candidates on a specific FPN layer based on the instance size, and "multi" refers to selecting candidates across all FPN layers. "*T.T.*" and "*s.r.*" are abbreviations for training time and sampling radius, respectively.

Method	AP^b	AP^b_{50}	AP^b_S	AP^b_M	AP^b_L
PolarMask	16.3	38.3	16.0	28.4	13.2
PolarNeXt	**19.8**	**46.7**	**20.2**	**32.2**	**15.9**

Table 6. Comparison of boundary quality between PolarMask and PolarNeXt.

4.3. Ablation Studies

In this section, we conduct ablations to assess the influence of the proposed components on instance segmentation performance, and then analyze their details in the following. All experiments are conducted on MS COCO val under a 1x training schedule.

Component Impact and Correlation. The effectiveness of URM and APSD is reported in Tab. 3. Using URM alone brings a 1.9% AP improvement on the baseline, while combining it with APSD further boosts the performance by an additional 2.9% AP. It is worth noting that APSD fails to function independently of URM, the absence of which results in a decrease in performance. To be specific, without APSD, URM contributes to the sample weighting and loss function, leading to a 1.2% and 0.7% AP increase, respectively. Conversely, when APSD is applied, the matching costs have to incorporate RMask IoU Loss; otherwise, this strategy will fail or even backfire. In view of this, we conclude that the effectiveness of our adaptive strategy hinges on reliable polygonal assessment for support.

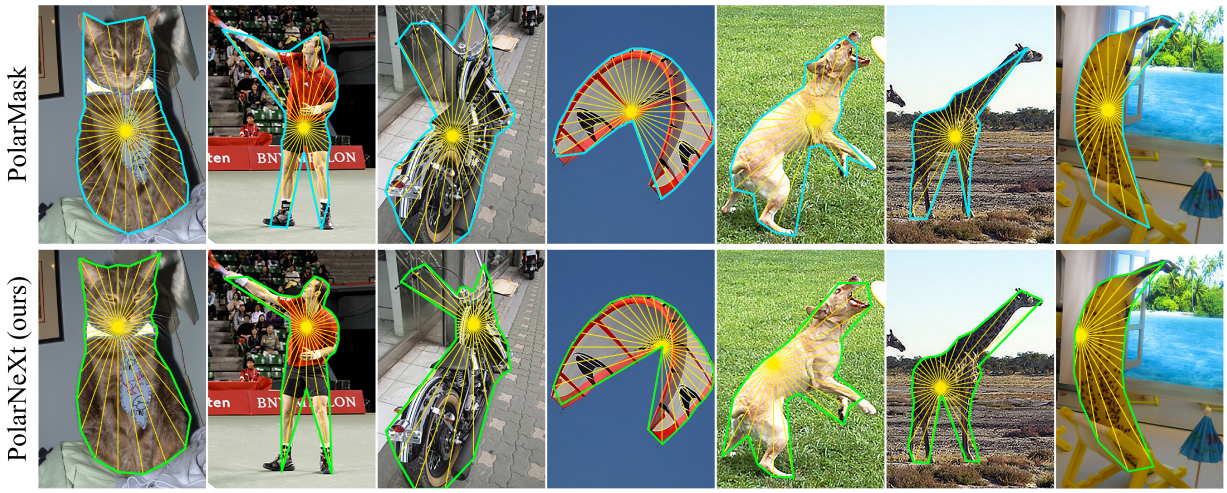

Figure 6. Visualization with Rays for PolarMask and PolarNeXt.

Candidate Selection. In Tab. 5, we compare some approaches to expand the range of candidate selection. On the one hand, Center Sampling on multiple FPN layers is more effective and efficient than the dense point selection within boxes or contours. In particular, the approach suggested in [31], which selects all grid points of FPN, results in unacceptable training time and GPU memory overflow. On the other hand, directly expanding the radius of Center Sampling on a single layer causes a 0.4% drop in AP, as the additional selected samples tend to fall outside the receptive field. Accordingly, the center prior still has a positive influence on sample decisions.

RMask IoU *vs.* Polar IoU. In Fig. 4, two scatter plots are drawn to compare Polar IoU and RMask IoU in polygonal assessment. These scatter points denote all the polygons predicted during inference on MS COCO val images. The fitted curve in Fig. 4(b) is almost aligned with the ideal curve, whereas the one in Fig. 4(a) shows significant deviation. This indicates that RMask IoU better reflects the actual polygonal quality compared to Polar IoU.

The necessity of Polar IoU Loss. In Tab. 4, we explore the feasibility of directly removing Polar IoU Loss in our proposed training pipeline. Experimental results indicate that this operation results in a 1.1% AP degradation in performance. The impact on small objects is more notable, with AP_S decreased by 2.1%. We analyze that there may be a slight granularity missing as well as feature misalignment during the conversion from vertex sets to rasterized masks. Moreover, as mentioned in [18], aliasing occurs in the deconvolutional process of FPN when pre-predicted RoIs are unavailable in one-stage detectors. Fortunately, the combination of Polar IoU Loss and RMask IoU Loss brings a notable boost, with the two complementing each other. We believe that Polar IoU Loss serves as a polygonal internal constraint for dense distance regression, while RMask IoU Loss provides a polygonal external constraint to incorporate the representation deviation.

4.4. Boundary Quality Analysis

In Tab. 6, we introduce Boundary AP [5] (AP^b for short) to evaluate the boundary quality of predicted polygons. Experimental results show that PolarNeXt achieves a significant improvement in boundary quality compared to PolarMask, boosting AP^b by 3.5%. Furthermore, we present some segmentation results with differences in starting points in Fig. 6. Obviously, compared to the center-fixed starting points in PolarMask, the dynamically selected starting points in PolarNeXt are more flexible, which enhances the representation capability of bounding polygons.

5. Conclusion

In this paper, we introduce a novel polar-based framework, PolarNeXt, which redemonstrates the superiority of Polar Representation in instance segmentation. To account for the previously ignored representation error, tailored training strategy APSD and assessment module URM are proposed. Together, these components significantly enhance PolarNeXt, yielding notable performance improvements over other polar-based methods. By fully leveraging the advantages of Polar Representation, PolarNeXt achieves comparable segmentation accuracy while requiring only half or even less of the computational cost and inference time compared to mainstream mask-based and contour-based methods. We anticipate that PolarNeXt will set a new standard for polar-based methods in instance segmentation.

Acknowledgments. This work is supported by the National Natural Science Foundation of China [Grant No. 62225308], and the Shanghai Technical Service Computing Center of Science and Engineering, Shanghai University.

References

[1] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *ICCV*, pages 9157–9166, 2019. 3, 6

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: High quality object detection and instance segmentation. *IEEE TPAMI*, 43(5):1483–1498, 2019. 3, 6

[3] 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 *CVPR*, pages 4974–4983, 2019. 3, 6

[4] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 6

[5] Bowen Cheng, Ross Girshick, Piotr Dollár, Alexander C Berg, and Alexander Kirillov. Boundary iou: Improving object-centric image segmentation evaluation. In *CVPR*, pages 15334–15342, 2021. 8

[6] Dominic Cheng, Renjie Liao, Sanja Fidler, and Raquel Urtasun. Darnet: Deep active ray network for building segmentation. In *CVPR*, pages 7431–7439, 2019. 3

[7] Tianheng Cheng, Xinggang Wang, Shaoyu Chen, Wenqiang Zhang, Qian Zhang, Chang Huang, Zhaoxiang Zhang, and Wenyu Liu. Sparse instance activation for real-time instance segmentation. In *CVPR*, pages 4433–4442, 2022. 1, 3, 6

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016. 6

[9] Hao Feng, Keyi Zhou, Wengang Zhou, Yufei Yin, Jiajun Deng, Qi Sun, and Houqiang Li. Recurrent generic contour-based instance segmentation with progressive learning. *IEEE TCSVT*, 2024. 1, 3, 6

[10] Günther Greiner and Kai Hormann. Efficient clipping of arbitrary polygons. *ACM Transactions on Graphics (TOG)*, 17 (2):71–83, 1998. 4

[11] Wenchao Gu, Shuang Bai, and Lingxing Kong. A review on 2d instance segmentation based on deep neural networks. *Image and Vision Computing*, 120:104401, 2022. 1, 3

[12] Shir Gur, Tal Shaharabany, and Lior Wolf. End to end trainable active contours via differentiable rendering. *arXiv preprint arXiv:1912.00367*, 2019. 4

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 6

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 1, 3, 6

[15] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring r-cnn. In *CVPR*, pages 6409–6418, 2019. 3, 6

[16] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *IJCV*, 1(4):321–331, 1988. 3

[17] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *CVPR*, pages 9799–9808, 2020. 3, 6

[18] Justin Lazarow, Weijian Xu, and Zhuowen Tu. Instance segmentation with mask-supervised polygonal boundary transformers. In *CVPR*, pages 4382–4391, 2022. 4, 5, 6, 8

[19] Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *NeurIPS*, 33:21002–21012, 2020. 3

[20] Justin Liang, Namdar Homayounfar, Wei-Chiu Ma, Yuwen Xiong, Rui Hu, and Raquel Urtasun. Polytransform: Deep polygon transformer for instance segmentation. In *CVPR*, pages 9131–9140, 2020. 3

[21] 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*, pages 740–755, 2014. 6

[22] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017. 3

[23] Huan Ling, Jun Gao, Amlan Kar, Wenzheng Chen, and Sanja Fidler. Fast interactive object annotation with curve-gcn. In *CVPR*, pages 5257–5266, 2019. 4

[24] Zichen Liu, Jun Hao Liew, Xiangyu Chen, and Jiashi Feng. Dance: A deep attentive contour model for efficient instance segmentation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 345–354, 2021. 3, 6

[25] Sida Peng, Wen Jiang, Huaijin Pi, Xiuli Li, Hujun Bao, and Xiaowei Zhou. Deep snake for real-time instance segmentation. In *CVPR*, pages 8533–8542, 2020. 1, 3, 6, 7

[26] Joseph Redmon. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 1

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE TPAMI*, 39(6):1137–1149, 2016. 1, 3

[28] T-YLPG Ross and GKHP Dollár. Focal loss for dense object detection. In *CVPR*, pages 2980–2988, 2017. 5

[29] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115:211–252, 2015. 6

[30] Uwe Schmidt, Martin Weigert, Coleman Broaddus, and Gene Myers. Cell detection with star-convex polygons. In *Medical Image Computing and Computer Assisted Intervention*, pages 265–273, 2018. 3

[31] Peize Sun, Yi Jiang, Enze Xie, Wenqi Shao, Zehuan Yuan, Changhu Wang, and Ping Luo. What makes for end-to-end object detection? In *International Conference on Machine Learning*, pages 9934–9944, 2021. 8

[32] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. In *ECCV*, pages 282–298, 2020. 3, 6

[33] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: A simple and strong anchor-free object detector. *IEEE TPAMI*, 44(4):1922–1933, 2020. 1

[34] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. Solo: Segmenting objects by locations. In *ECCV*, pages 649–665, 2020. 1, 3, 6

[35] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. Solov2: Dynamic and fast instance segmentation. *NeurIPS*, 33:17721–17732, 2020. 3, 6

[36] 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. In *CVPR*, pages 12193–12202, 2020. 1, 2, 3, 5, 6

[37] Enze Xie, Wenhai Wang, Mingyu Ding, Ruimao Zhang, and Ping Luo. Polarmask++: Enhanced polar representation for single-shot instance segmentation and beyond. *IEEE TPAMI*, 44(9):5385–5400, 2021. 3

[38] Wenqiang Xu, Haiyang Wang, Fubo Qi, and Cewu Lu. Explicit shape encoding for real-time instance segmentation. In *ICCV*, pages 5168–5177, 2019. 1, 3, 6

[39] Tao Zhang, Shiqing Wei, and Shunping Ji. E2ec: An end-to-end contour-based method for high-quality high-speed instance segmentation. In *CVPR*, pages 4443–4452, 2022. 1, 3, 6