Learning Equivariant Segmentation with Instance-Unique Querying

Anonymous Author(s)
Affiliation
Address
email

Abstract

Prevalent state-of-the-art instance segmentation methods fall into a *query*-based scheme, in which instance masks are derived by querying the image feature using a set of instance-aware embeddings. In this work, we devise a new training framework that boosts query-based models through discriminative query embedding learning. It explores two essential properties, namely *dataset-level uniqueness* and *transformation equivariance*, of the relation between queries and instances. First, our algorithm uses the queries to retrieve the corresponding instances from the whole training dataset, instead of only searching within individual scenes. As querying instances across scenes is more challenging, the segmenters are forced to learn more discriminative queries for effective instance separation. Second, our algorithm encourages both image (instance) representations and queries to be equivariant against geometric transformations, leading to more robust, instance-query matching. We experimentally show, on top of four famous, query-based models (*i.e.*, CondInst, SOLOv2, SOTR, and Mask2Former), our algorithm provides solid performance gains (*e.g.*, +2.0 – 3.2 *AP*) on COCO dataset. Our code will be released.

1 Introduction

Instance segmentation, *i.e.*, labeling image pixels with classes and instances, plays a critical role in a wide range of applications, *e.g.*, autonomous driving, medical health, and augmented reality. Modern instance segmentation solutions are largely built upon three paradigms: *top-down* ('detect-then-segment') [1–18], *bottom-up* ('label-then-cluster') [19–26], and *single-shot* ('directly-predict') [27–42]. Among them, the top-leading algorithms [18, 32, 37–42] typically operate in a *query*-based mode, in which a set of instance-aware embeddings is learned and used to query the dense image feature for instance mask prediction. The key to their triumph is the instance-aware query vectors that are learned to encode the characteristics (*e.g.*, location, appearance) of instances [32, 41]. By straightforwardly minimizing the differences between the retrieved and groundtruth instance masks, the query-based methods, in essence, learn the query vectors for instance discrimination only *within individual scenes*.

As a result, existing query-based instance segmentation algorithms place a premium on intra-scene analysis during network training. Since the scenario in one single training scene is simple, *i.e.*, the diversity and volume of object instances as well as the complexity of the background are typically limited, learning to distinguish between object instances only within the same training scenes is less challenging, and inevitably hinders the discrimination potential of the learned instance queries.

This work brings a paradigm shift in training query-based instance segmenters: it goes beyond the *de facto*, within-scene training strategy by further considering the cross-scene level query embedding separation of different instances – *querying instances from the whole training dataset*. The underlying rationale is intuitive yet powerful: an advanced instance segmenter should be able to differentiate all the instances of the entire dataset, rather than only the ones within single scenes. Concretely, in our

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

training framework, the queries are not only learned to fire on the pixels of their counterpart instances in the current training image, but also forced to mismatch the pixels in other training images. By virtue of intra-and inter-scene instance disambiguation, our framework forces the query-based segmenters to learn more discriminative query vectors capable of uniquely identifying the corresponding instances even at the dataset level. To further facilitate the establishment of robust, one-to-one relation between queries and instances, we complement our training framework with a *transformation equivariance* constraint, accommodating the equivariance property of the instance segmentation task to geometric transformations. For example, if we crop or flip the input image, we expect the image (instance) features and query embeddings to change accordingly, so as to appropriately reflect the variation of instance patterns (*e.g.*, scale, position, shape, *etc*) caused by the input transformation.

Exploring intra-and inter-scene instance uniqueness as well as transformation equivariance leads to a general yet powerful training framework. Our algorithm, in principle, can be seamlessly incorporated into the training process of existing query-based instance segmenters. For comprehensive evaluation, we apply our algorithm to four representative, query-based models (*i.e.*, CondInst [32], SOLOv2 [37], SOTR [39], and Mask2Former [38]) with various backbones (*i.e.*, ResNet [43], Swin [44]). Experiments on COCO [45] verify our impressive performance, *i.e.*, **+2.8 – 3.1**, **+2.9 – 3.2**, **+2.4 – 2.6**, and **+2.0 – 2.4** AP gains over CondInst, SOLOv2, SOTR, and Mask2Former, respectively (see Fig. 1). These results are particularly impressive considering our training algorithm causes neither architectural change nor extra computational load during model deployment. Our code will be released.

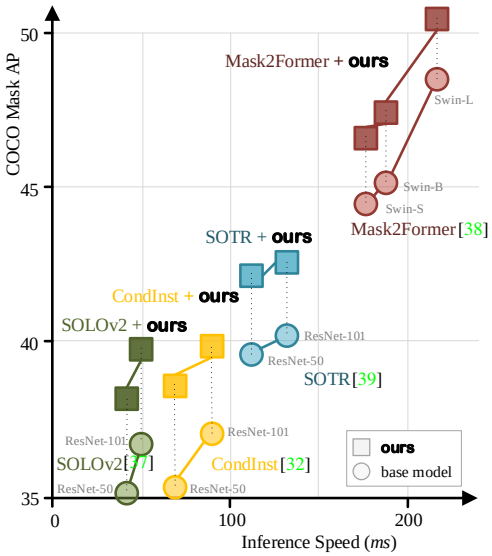

Figure 1: Our training algorithm yields solid performance gains over state-of-the-art query-based models [32, 37–39] without architectural modification and inference speed delay.

2 Related Work

This section summarizes the most relevant research on instance segmentation and equivariant learning.

Instance Segmentation. Existing deep learning based solutions can be broadly classified into three paradigms: top-down, bottom-up, and single-shot. Following the idea of '*detect-then-segment*', **top-down** methods [1–18] predict a bounding box for each object and then ouput an instance mask for each box. Though effective, this type of methods is complicated and dependent on the priori detection results. In contrast, **bottom-up** methods [19–26] adopt a '*label-then-cluster*' strategy: learning per-pixel embeddings and then grouping them into different instances. Albeit simple, this type of methods relies on the performance of post-processing and easily suffers from under-segment or over-segment problems. Inspired by the advance of single-stage object detection [46,47], a few recent efforts approach instance segmentation in a **single-shot** fashion, by coalescing detection and segmentation over pre-defined anchor boxes [27–31], or directly predicting instance masks from feature maps [32–42]. This type of methods is well recognized and generally demonstrates better speed-accuracy trade-off [48].

Despite the blossoming of diverse approaches, the vast majority of recent top-performing algorithms [18, 32, 37–42] fall into one grand category – *query*-based models. The query-based methods utilize compact, learnable embedding vectors to represent instances of interest and leverage them as queries to decode masks from image features. Their triumph is founded on comprehensively encoding instance-specific properties (*e.g.*, location, appearance) into the query vectors, which significantly increases prediction robustness. For instance, [32, 37] exploit the technique of dynamic filter [49] to generate instance-specific descriptors, which are convolved with image feature maps for instance mask decoding. Inspired by DETR [50], [38, 40, 41] alternatively leverage a Transformer decoder to obtain instance-aware query embeddings and cast instance segmentation as a set prediction problem.

Our contribution is orthogonal and these query-based segmenters can benefit. We scaffold a new training framework that sharpens the instance discriminative capability of the query-based segmenters. This is achieved by matching query embeddings with instances within and cross training scenes. Such intra-and inter-scene instance querying strategy is further enhanced by an equivariance regularisation term, addressing not only the uniqueness but also the robustness of instance-query relations.

Equivariant Representation Learning. Transformations play a critical role in learning expressive representations by transforming images as a means to reveal the intrinsic patterns from transformed visual structures [51]. Motivated by the concept of *translation equivariance* underlying the success of CNNs, numerous efforts (*e.g.*, capsule nets [52,53], group equivariant convolutions [54], and harmonic networks [55]) investigate learning more powerful representations *equivariant* to generic types of transformations [56, 57]. A representation f is said to be equivariant with a transformation g for input (say image) I if $f(g(I)) \approx g(f(I))$. In other words, the output representation $f(I)$ transforms in the same manner (or, in a broad sense, a predictable manner) given the input transformation g. Many recent self-supervised learning methods [58–60] encourage the representations to be *invariant* under transformations, *i.e.*, $f(g(I)) \approx f(I)$. As such, invariance can be viewed as a special case of equivariance [61] where the output representation $f(I)$ does not vary with the input transformation g.

In our training framework, we fully explore the inherent, transformation-equivariance nature of the instance segmentation task to pursue reliable, one-to-one correspondence between learnable queries and object instances. This is accomplished by promoting the equivariance of both query embeddings and feature representations with respect to spatial transformations, *i.e.*, cropping or flipping of an input image should result in correspondingly changed feature representation, query embeddings, as well as instance mask predictions. Our algorithm is also in contrast to the common data augmentation strategy, in which the transformed images and annotations are used directly as additional training examples, without any constraint about the relation between the representations (and queries) produced from the original and transformed views. Our experimental results (see §4.3) also evidence the superiority of our transformation equivariance learning over transformation-based data augmentation.

3 Methodology

Next, we first formulate instance segmentation from a classical view of mask prediction and classification (§3.1). Then we describe our new training framework (§3.2) and implementation details (§3.3).

3.1 Problem Formulation

Instance segmentation seeks a partition of an observed image $I \in \mathbb{R}^{H \times W \times 3}$ into K instance regions:

$$\{Y_k\}_{k=1}^{K} = \{(M_k, c_k)\}_{k=1}^{K}, \quad \text{where } M_k \in \{0,1\}^{H \times W}, \quad c_k \in \{1, \cdots, C\}. \tag{1}$$

Here the instances of interest are represented by a total of K *non-overlap*, binary masks $\{M_k\}_{k=1}^{K}$ as well as corresponding class labels $\{c_k\}_{k=1}^{K}$ (*e.g.*, table, chair, *etc*). For a pixel $i \in I$, its counterpart value in the k-*th* groundtruth mask, *i.e.*, $M_k(i)$, denotes whether i belongs to instance k (1) or not (0). Note that the number of instances, K, varies across different images. Existing mainstream (or, more precisely, most top-down and one-shot) solutions approach the task by decomposing the image I into a fixed-size set of soft masks. In this setting, each mask is associated with a probability distribution over all the C categories. The output can thus be represented as a set of N mask-probability pairs:

$$\{\hat{Y}_n\}_{n=1}^{N} = \{(\hat{M}_n, \hat{p}_n)\}_{n=1}^{N}, \quad \text{where } \hat{M}_n \in [0,1]^{H \times W}, \quad \hat{p}_n \in \triangle^C. \tag{2}$$

Here $\triangle^C$ stands for the C-dimensional probability simplex. The size of the prediction set, N, is usually set as a constant and much larger than the typical number of object instances in an image. Hence the training objective penalizes the errors of both label prediction and mask estimation:

$$\mathcal{L}(\{\hat{Y}_n\}_{n=1}^{N}, \{Y_k\}_{k=1}^{K}) = \sum\nolimits_{n=1}^{N} \mathcal{L}_{cls}(\hat{p}_n, c_{\sigma(n)}) + \mathcal{L}_{mask}(\hat{M}_n, M_{\sigma(n)}), \tag{3}$$

where σ refers to the matching between the prediction and groundtruth sets (established by certain rules [6,50]). The classification loss $\mathcal{L}_{cls}$ is typically the cross-entropy loss or focal loss [62]; the mask prediction loss $\mathcal{L}_{mask}$ can be the cross-entropy loss in [38], dice loss [63] in [32, 37], or focal loss in [50]. While many approaches supplement $\mathcal{L}$ with various extra losses (*e.g.*, bounding box loss [6,30,32,50], ranking loss [64], semantic segmentation loss [13, 30]), later we will show our *cross-scene* training scheme is fundamentally different from (yet complementary to) current *scene-wise* training paradigm.

3.2 Equivariant Learning with Intra-and Inter-Scene Instance Uniqueness

Query-based Instance Segmentation. As clearly indicated by Eq. 2, the prediction masks $\{\hat{M}_n\}_{n=1}^{N}$ are the means of separating instances at the pixel level. Current top-performing instance segmenters [18, 32, 37–41] typically generate mask predictions in a *query*-based fashion (see the middle part of Fig. 2). Let f be a *dense feature extractor* (*e.g.*, an encoder-decoder fully convolutional

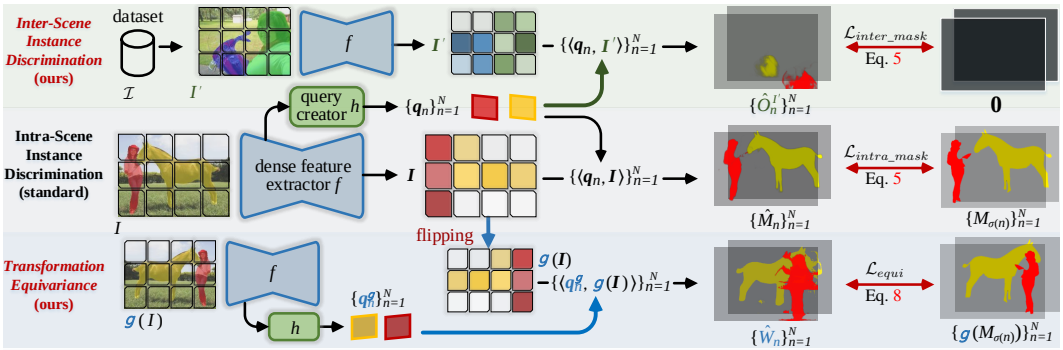

Figure 2: Overview of our new training framework for query-based instance segmentation. Rather than current *intra-scene* training paradigm, our framework addresses *inter-scene instance discrimination* and *transformation equivariance* for discriminative instance query embedding learning (see §3.2). To improve readability, for image I, we only plot one extra image I' for cross-scene training.

network [65]) that produces D-dimensional dense embedding $\boldsymbol{I}$ for image I, *i.e.*, $\boldsymbol{I}=f(I)\in\mathbb{R}^{H\times W\times D}$. Then a *query creator* h is adopted to produce a set of N instance-aware embedding vectors $\{\boldsymbol{q}_n\in\mathbb{R}^d\}_{n=1}^N$, which are used to query the image representation $\boldsymbol{I}$ for instance mask decoding:

$$\{\hat{M}_n\}_{n=1}^N=\{\langle\boldsymbol{q}_n,\boldsymbol{I}\rangle\}_{n=1}^N,\quad \text{where } \{\boldsymbol{q}_n\}_{n=1}^N=h(\boldsymbol{I}^{\Downarrow}). \tag{4}$$

Here $\langle\cdot,\cdot\rangle$ is a certain similarity measure performed pixel-wise, and $\boldsymbol{I}^{\Downarrow}$ (typically) refers to a low-resolution feature representation of image I. Note that position information is integrated to either or both of $\boldsymbol{I}$ and $\boldsymbol{I}^{\Downarrow}$, to make the model *location-sensitive*. The query creator h is implemented as a dynamic network [32,37], or a Transformer decoder [38,40]. For dynamic network based h, it predicts N convolution filters $\{\boldsymbol{q}_n\}_{n=1}^N$ dynamically conditioned on the input $\boldsymbol{I}^{\Downarrow}$, and hence $\langle\cdot,\cdot\rangle$ refers to convolution ($d\neq D$). For Transformer decoder based h, it additionally leverages a set of N learnable positional embeddings (omitted for brevity) to gather instance-related context from $\boldsymbol{I}^{\Downarrow}$; the collected context is stored in $\{\boldsymbol{q}_n\}_{n=1}^N$ and $\langle\cdot,\cdot\rangle$ is computed as dot product for instance mask decoding ($d=D$). Eq. 4 informs that, query-based segmenters in essence learn N compact descriptors $\{\boldsymbol{q}_n\}_{n=1}^N$ to grasp critical characteristics (*e.g.*, appearance and location) of potentially interested instances, and use these instance descriptors as queries to retrieve corresponding pixels from $\boldsymbol{I}$. It is thus reasonable to assume the discriminative ability of the learned query embeddings is crucial for the performance of query-based methods. Viewed in this light, a question naturally arises: ❷ *how to learn discriminative instance query embeddings*? Yet, this fundamental question is largely ignored in the literature so far.

To respond ❷, we exploit two crucial properties of instance-query matching, namely *uniqueness* and *robustness*, during the learning of query-based segmenters, leading to a powerful training scheme.

Learning with Intra-and Inter-Scene Instance Uniqueness. If we closely scrutinize at the current *de facto* training regime (*cf.* Eq. 3) and the work mode of query-based segmenters (*cf.* Eq. 4), we can find: the mask prediction loss $\mathcal{L}_{mask}$ forces each query $\boldsymbol{q}_n$ to match the pixels $i\in I$ of its counterpart instance $k=\sigma(n)$, *i.e.*, $M_{\sigma(n)}(i)=1$, and mismatch the pixels $i'\in I$ of other instances $k\neq\sigma(n)$, *i.e.*, $M_{\sigma(n)}(i')=0$, *within* image I. That means, the segmenters are only taught to differentiate instances within *individual* scenes. Evidently, individual scenes are unable to cover diverse instance patterns; the scarcity of negative instances — an intrinsic limitation of the current *scene-wise* training regime — inevitably impairs query embedding learning. This analysis motivates us to design a *cross-scene* training strategy, where the segmenters are forced to learn to query the target instances from the whole training dataset, instead of only matching queries and instance pixels within the same scenes. Thus we take a step further by simultaneously leveraging the intra-and inter-scene context to promote query embedding learning. Query-based segmenters are encouraged to develop dataset-level, cross-instance discriminativeness and perform reliable mask prediction using instance uniqueness.

Let $\mathcal{I}$ be the training set of $|\mathcal{I}|$ images and $\{\boldsymbol{q}_n\}_{n=1}^N$ the instance queries generated from image $I\in\mathcal{I}$. During training, rather than only applying $\{\boldsymbol{q}_n\}_{n=1}^N$ for the counterpart image I to get N *intra-image* instance prediction masks $\{\hat{M}_n\}_{n=1}^N$ (*cf.* Eq. 4), we further use $\{\boldsymbol{q}_n\}_{n=1}^N$ to query other training images $I'\in\mathcal{I}$ for cross-scene instance disambiguation. This will result in N *inter-image* instance prediction masks $\{\hat{O}_n^{I'}\in[0,1]^{H\times W}\}_{n=1}^N$ for each I', *i.e.*, $\{\hat{O}_n^{I'}\}_{n=1}^N=\{\langle\boldsymbol{q}_n,\boldsymbol{I}'\rangle\}_{n=1}^N$, where $\boldsymbol{I}'\in\mathbb{R}^{H\times W\times D}$ is the feature map of I', *i.e.*, $\boldsymbol{I}'=f(I')$. For each query $\boldsymbol{q}_n$ of image I, given its corresponding intra-image instance prediction map $\hat{M}_n$, a total of $|\mathcal{I}|-1$ inter-image instance prediction maps $\{\hat{O}_n^{I'}\}_{I'\neq I}$, and

the original groundtruth mask $M_{\sigma(n)}$, we improve the mask prediction loss $\mathcal{L}_{mask}$ in Eq. 3 as a form of:

$$\sum\nolimits_{n=1}^{N} \left(\mathcal{L}_{intra_mask}(\hat{M}_n, M_{\sigma(n)}) + \frac{1}{|\mathcal{I}|-1} \sum\nolimits_{I' \neq I} \mathcal{L}_{inter_mask}(\hat{O}_n^{I'}, \mathbf{0})\right), \tag{5}$$

where the first term $\mathcal{L}_{intra_mask}$ is exactly the same as $\mathcal{L}_{mask}$ in Eq. 3, which is renamed for addressing its nature of inspiring intra-scene instance uniqueness. The newly added term $\mathcal{L}_{inter_mask}$ is to eliminate the instance ambiguity through inter-scene querying. Thus its training target is an all-zero query-pixel matching matrix $\mathbf{0}$ of size $H \times W$, *i.e.*, each query $\boldsymbol{q}_n$ of image I is forced to mismatch the instances in other images I'. Among the several choices for $\mathcal{L}_{inter_mask}$, *e.g.*, ℓ_1 loss, ℓ_2 loss, and cross-entropy loss, we find focal loss [62] is more favored (*cf.* §4.3). We speculate this is due to the advantage of focal loss in preventing the influence of massive trivial examples, as in our case. To further boost the effectiveness and efficiency of our algorithm, the following strategies are adopted:

- *External memory*: As the capacity of training batch is limited, we follow the recent practice in unsupervised representation learning [58, 66–68] to build an external memory to enable large-scale query-instance matching. The memory gathers instance pixel embeddings from several batches. Note that we drop background pixels, as we find their contribution is negligible (sometimes even negative). This also implies that cross-instance discrimination is the core challenge of this task.
- *Sparse sampling*: As local image contents are highly correlated, storing all the pixel samples will bring a lot of redundant information, which is useless for discriminative representation learning. With a similar spirit of some dense prediction methods [15, 68], only a small set of instance pixels are randomly sampled from each image and fed into the memory. In practice, we find this strategy greatly improves the diversity of the instance samples and eventually benefits the final performance.
- *Instance-balanced sampling*: Though effective, the sparse sampling strategy has a side-effect: it hurts the performance on small instances. This is because larger instances have more chance of being sampled for loss computation, enabling our training algorithm to give less priority towards smaller instance. We therefore opt to randomly sample a fixed number of pixels from each instance region. As a result, the performance is further improved. All the related experiments can be found in §4.3.

These designs together form a cross-scene training framework, whose power lies in the query embedding uniqueness among plentiful instances. Our framework is elegant and principled: as suggested by Eq. 5, it can complement and integrate with current training paradigm of query-based methods.

Learning with Transformation Equivariance. So far, we have addressed ❷ from a fresh perspective of dataset-level instance-unique querying. Next, we complete our response from another essential viewpoint: *robustness*. That is, we expect the segmenters to build robust instance-query correspondences that are equivariant against input imagery transformations (*e.g.*, cropping, flipping, *etc*).

Our key insight for robustness is derived from the transformation equivariance nature of the task: flipping or cropping the image should result in an exact change in the instance segmentation mask. At the first glimpse, this character seems to have been already captured by the widely-used, transformation-based data augmentation technique. Under scrutiny, this view however is erroneous because: the representations (and instance queries) ought to be transformation equivariant first, then it is natural to achieve correctly transformed predictions. Yet, the current training strategy puts the cart before the horses: with input transformations, the segmenter is only trained to produce correspondingly transformed predictions, but without any reasonable constraints on the representations (and queries).

Given the analysis above, we supplement our framework with an equivariance based training objective, encouraging the segmenters to precisely and reliably anchor queries on instances. Let G be a group of transformations (*e.g.*, cropping, flipping, *etc*). For a query-based instance segmenter, we encourage its feature representation to be "equivariant against input imagery transformations", which states:

$$\forall g \in G : f(g(I)) \approx g(f(I)) = g(\boldsymbol{I}). \tag{6}$$

Intuitively, the output representation of f is encouraged to change in the same way to the transformation g applied to the input I. Similarly, the instance queries are also desired to be transformation equivariant, *i.e.*, they should be able to properly describe the transformation applied to their counterpart instances. For example, after cropping or flipping an input image, the queries are expected to still robustly capture the changed instance patterns (*e.g.*, shape, location, scale). We formulate this property as:

$$\forall g \in G : \{\langle \boldsymbol{q}_n^g, \boldsymbol{I}^g \rangle\}_{n=1}^{N} \approx \{g(M_{\sigma(n)})\}_{n=1}^{N}, \tag{7}$$

where $\boldsymbol{I}^g$ denotes the feature embedding of transformed image (instances) $g(I)$, *i.e.*, $\boldsymbol{I}^g = f(g(I))$; analogously, $\{\boldsymbol{q}_n^g\}_{n=1}^{N}$ indicates the query embeddings derived from $g(I)$. Eq. 7 states that, with

a transformed input image $g(I)$, the queries should be properly changed so as to correctly match the embeddings of transformed instances $\boldsymbol{I}^g$. This also explains the rationale behind the popular transformation-based augmentation technique from a view of equivariant query embedding learning.

Considering both Eq. 6 and 7, we can easily have $\forall g \in G\colon \{\langle \boldsymbol{q}_n^g, g(\boldsymbol{I})\rangle\}_{n=1}^N \approx \{g(M_{\sigma(n)})\}_{n=1}^N$, and hence our equivariance based training objective is formulated as (see the bottom part of Fig. 2):

$$\sum\nolimits_{n=1}^{N} \mathcal{L}_{equi}(\hat{W}_n^g, g(M_{\sigma(n)})), \quad \text{where } \hat{W}_n^g = \langle \boldsymbol{q}_n^g, g(\boldsymbol{I})\rangle. \tag{8}$$

As such, the desired equivariance property of both feature representations and instance query embeddings to input transformations are formulated in a unified training objective. For Eq. 8, the queries $\{\boldsymbol{q}_n^g\}_{n=1}^N$, created for transformed image $g(I)$, are first applied for the transformed representation $g(\boldsymbol{I})$ of I; then the training target is to minimize the difference between the retrieved instance masks $\{\hat{W}_n^g \in [0,1]^{H\times W}\}_{n=1}^N$ and the transformed groundtruth masks $\{g(M_{\sigma(n)})\}_{n=1}^N$. Therefore, $\mathcal{L}_{equi}$ can be easily implemented as any existing mask prediction loss, and effortless incorporated into our training framework. For the sake of simplicity, we set $\mathcal{L}_{equi}$ as the same form of $\mathcal{L}_{mask}$ in Eq. 3. In our experiments (*cf.* §4.3), we will demonstrate that our equivariance constraint can bring significantly larger performance gains than the conventional, transformation-based data augmentation technique.

3.3 Implementation Detail

External Memory and Sampling. For large-scale query-instance matching (*cf.* Eq. 5), we build an external memory. Due to the limited capacity of our GPU, we afford to maintain a queue of 100K pixel samples in the memory. We adopt sparse, instance-balanced sampling, *i.e.*, randomly selecting 50 (if possible) pixels from each instance. Note that the samples stored in the memory are only used for the computation of the inter-scene instance discrimination loss $\mathcal{L}_{inter_mask}$ in Eq. 5. The external memory is directly discarded after training, causing no extra computation budget during inference.

Training Objective. Ordinarily, our training framework is applicable for existing query-based models, and complementary to their training objectives. In our experiments, we approach our algorithm on four representative query-based methods [32, 37–39] by adding our inter-scene instance discrimination loss $\mathcal{L}_{inter_mask}$ (*cf.* Eq. 5) and equivariance loss $\mathcal{L}_{equi}$ (*cf.* Eq. 8) into their training targets. As mentioned before, we implement $\mathcal{L}_{inter_mask}$ (*cf.* Eq. 5) as the focal loss [62], whose hyper-parameters are set to $\alpha = 0.1$ and $\gamma = 2.5$. As for $\mathcal{L}_{equi}$, it is directly set as the same form of the loss used for mask prediction in the base segmenter, *i.e.*, dice loss [63] for [32, 37, 39], and a linear combination of the focal loss and dice loss for [38]. For far comparison, the set of transformations G, which is used for the computation of $\mathcal{L}_{equi}$, is {horizontal flipping, random cropping between 0.6 and 1.0}. This is consistent with the standard data augmentation setup adopted by existing instance segmentation networks. To balance the impacts of our two new training targets, *i.e.*, $\mathcal{L}_{inter_mask}$ and $\mathcal{L}_{equi}$, we multiply $\mathcal{L}_{equi}$ by a coefficient λ, which is empirically set as 3 (see related experiments in Table 3e&§4.3).

Reproducibility. Our algorithm is implemented in PyTorch. For all our experiments, the training and testing are conducted on eight NVIDIA Tesla A100 GPUs with a 80GB memory per-card. To make our work reproducible, our full implementations will be publicly released.

4 Experiments

4.1 Experimental Setup

Dataset. We conduct experiments on COCO [45]. There are 80 target classes for instance segmentation. As normal, we use `train2017` split (115k images) for training and `val2017` (5k images) for validation used in our ablation study. The main results are reported on `test-dev` (20k images).

Base Instance Segmenters, Backbones, and Competitors. To demonstrate the broad applicability and wide benefit of our algorithm, we approach our method on four widely recognized query-based instance segmentation models, *i.e.*, CondInst [32], SOLOv2 [37], SOTR [39], and Mask2Former [38], with diverse backbone networks, *i.e.*, ResNet-50/-101 [43] and Swin-Small/-Base/-Large [44]. For fair comparison, all these models and results are based on our reproduction, following the default hyper-parameter and augmentation recipes in AdelaiDet [69] and MMDetection [70]. In addition to focusing on the comparison with these four base segmentation models, we include a group of famous segmenters [6, 11, 12, 15, 18, 41, 42] for comprehensive evaluation. ***In our supplementary material, we report more experiments on two extra base models,*** *i.e.*, SparseInst [42] and SOLQ [41].

Training. Our implementation is based on AdelaiDet [69] and MMDetection [70], following default

Table 1: Quantitative results on COCO [45] `test-dev`. See §4.2 for details.

Method	Backbone	#Epoch	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Mask R-CNN[ICCV17] [6]	ResNet-101	12	36.1	57.5	38.6	18.8	39.7	49.5
Cascade Mask R-CNN[PAMI19] [11]	ResNet-101	12	37.3	58.2	40.1	19.7	40.6	51.5
HTC[CVPR19] [12]	ResNet-101	20	39.6	61.0	42.8	21.3	42.9	55.0
Point Rend[CVPR20] [15]	ResNet-50	12	36.3	56.9	38.7	19.8	39.4	48.5
QueryInst[ICCV21] [18]	ResNet-101	36	41.0	63.3	44.5	21.7	44.4	60.7
SOLQ[NeurIPS21] [41]	Swin-L	50	46.7	72.7	50.6	29.2	50.1	60.9
SparseInst[CVPR22] [42]	ResNet-50	36	37.9	59.2	40.2	15.7	39.4	56.9
CondInst[ECCV20] [32]	ResNet-50	12	35.5	55.8	37.7	16.8	39.2	50.6
+Ours			**38.6**↑3.1	**61.1**↑5.3	**41.2**↑3.5	**19.7**↑2.9	**41.1**↑1.9	**54.7**↑4.1
CondInst[ECCV20] [32]	ResNet-101		37.1	58.6	39.3	18.2	40.3	52.9
+Ours			**39.9**↑2.8	**62.7**↑4.1	**42.4**↑3.1	**20.8**↑2.6	**42.3**↑2.0	**55.7**↑2.8
SOLOv2[NeurIPS20] [37]	ResNet-50	12	35.1	55.5	37.0	13.9	38.4	53.7
+Ours			**38.3**↑3.2	**59.6**↑4.1	**40.6**↑3.6	**17.8**↑3.9	**41.8**↑3.4	**56.2**↑2.5
SOLOv2[NeurIPS20] [37]	ResNet-101		36.7	57.5	39.3	16.2	40.2	54.2
+Ours			**39.6**↑2.9	**60.6**↑3.1	**43.1**↑3.8	**20.0**↑3.8	**43.3**↑3.1	**56.9**↑2.7
SOTR[ICCV21] [39]	ResNet-50	24	39.6	60.7	42.6	10.3	58.7	72.1
+Ours			**42.2**↑2.6	**61.9**↑3.1	**43.9**↑2.3	**11.0**↑0.7	**60.5**↑2.8	**73.5**↑2.4
SOTR[ICCV21] [39]	ResNet-101		40.2	61.2	43.4	10.2	59.0	73.1
+Ours			**42.6**↑2.4	**64.1**↑2.9	**45.8**↑2.4	**11.2**↑1.0	**61.2**↑2.2	**75.3**↑2.2
Mask2Former[CVPR22] [38]	Swin-S	50	44.5	68.2	47.7	24.3	49.0	66.6
+Ours			**46.7**↑2.2	**70.7**↑2.5	**50.1**↑2.4	**27.0**↑2.7	**50.9**↑1.9	**68.8**↑2.2
Mask2Former[CVPR22] [38]	Swin-B		44.9	68.5	48.5	24.6	48.7	67.4
+Ours			**47.3**↑2.4	**71.3**↑2.8	**51.0**↑2.5	**28.3**↑2.7	**50.9**↑2.2	**69.4**↑2.0
Mask2Former[CVPR22] [38]	Swin-L	100	48.5	72.9	52.5	28.4	53.0	69.4
+Ours			**50.5**↑2.0	**75.1**↑2.2	**54.8**↑2.3	**31.2**↑2.8	**54.5**↑1.5	**71.2**↑1.8

training configurations. In particular, all the backbones are initialized using corresponding weights pre-trained on ImageNet-1K [71], while remaining layers are randomly initialized. We train models using SGD with initial learning rate of 0.01 for ResNet [43] backboned models and Adam with initial learning rate of $1e^{-5}$ for Swin [44] backboned models. The learning rate is scheduled following the polynomial annealing policy with batch size 16. We use multi-scale training [12, 16, 32, 42]: for ResNet backboned models, the short side of input is randomly chosen within [400, 1200] and long side is set to 1333; for Swin backboned models, we opt random scaling with a factor in [0.1, 2.0] followed by a fixed size to 1024×1024. ResNet backboned models are trained for 12 epochs, while Swin backboned models are trained for 50 epochs. Note that, compared with the standard, scene-wise training strategy, our algorithm only slows the training speed slightly ($\sim$5%; see Table 2&§4.3).

Testing. All our experimental results are reported with 1333×800 input resolution. For the sake of fairness, *we do not apply any test-time data augmentation*. Note that, during model deployment, our training algorithm does not bring any change to network architecture or additional computation cost.

Evaluation Metric. We adopt the public *cocoapi* to report AP, AP_{50}, AP_{75}, AP_S, AP_M, and AP_L.

4.2 Comparison to State-of-the-Arts

Quantitative Result. Table 1 reports comparison results with our four base models, *i.e.*, CondInst [32], SOLOv2 [37], SOTR [39], and Mask2Former [38], as well as several representative instance segmentation methods [6, 11, 12, 15, 18, 41, 42] on COCO [45] `test-dev`. Without bells and whistles, our new training framework provides remarkable performance improvements over the four query-based segmenters [32, 37–39] with different backbones. In particular, for CondInst [32] and SOLOv2 [37] — the two famous query-based approaches that adopt the CNN architecture and dynamic neural network, with ResNet-50 [43] backbone, our algorithm brings **3.1** and **3.2** AP gains, respectively. Similar improvements, *i.e.*, **2.8** and **2.9** in terms of AP, are obtained with ResNet-101. As for SOTR [39] — a very recent method that adopts a hybrid structure of Transformer [72] structure and dynamic convolution for query creation, our algorithm also greatly promotes its performance, *e.g.*, 39.6→**42.2** with ResNet-50 and 40.2→**42.6** with ResNet-101, in terms of AP. Moreover, when directly applying our algorithm to the newest query-based model — Mask2Former [38], we observe powerful performance gains across different Swin [44] backbones. For instance, with Swin-B, our algorithm surpasses the vanilla Mask2Former at different IoU thresholds, *i.e.*, **71.3** *vs* 68.5 AP_{50} and **51.0** *vs* 48.5 AP_{75}. Notably, with Swin-L as the Mask2Former's backbone, our approach surpasses all the other competitors in Table 1 and sets a new state-of-the-art record of **50.5** in AP for instance segmentation

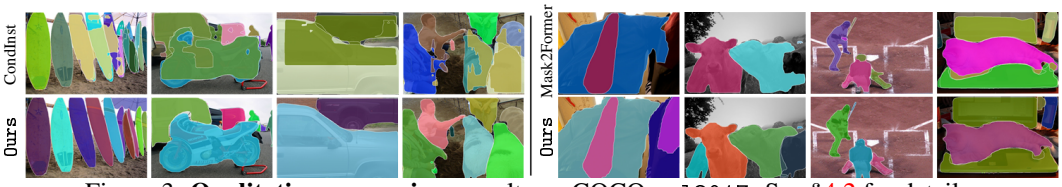

Figure 3: **Qualitative comparison** results on COCO `val2017`. See §4.2 for details.

Table 2: Analysis of essential components on COCO [45] `val2017`. See §4.3 for details.

Inter-scene $\mathcal{L}_{inter_mask}$ (Eq. 5)	Equivariance $\mathcal{L}_{equi}$ (Eq. 8)	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	Training speed (hour/epoch)
		35.5	55.9	37.4	16.9	38.9	50.2	1.51
✓		37.6	58.4	40.3	18.2	40.8	54.0	1.57
	✓	36.5	58.0	39.0	17.1	39.5	51.6	1.52
✓	✓	$38.1_{\uparrow 2.6}$	$60.2_{\uparrow 4.3}$	$40.6_{\uparrow 3.2}$	$18.4_{\uparrow 1.5}$	$40.9_{\uparrow 2.0}$	$54.3_{\uparrow 4.1}$	1.59

on COCO. Furthermore, in spite of different base segmentation network architectures and backbones, our training algorithm consistently improves the performance for object instances of different sizes.

These results are particularly impressive, considering the fact that the performance improvement is purely brought by a new training strategy, without any network architectural modification. Overall, our extensive experimental results manifest the effectiveness and wide benefit of our algorithm.

Qualitative Result. Fig. 3 provides qualitative comparisons of our algorithm against CondInst [32] (left) and Mask2Former [38] (right) on several challenging examples of COCO `val2017`. As can be seen, with the help of our training algorithm, CondInst and Mask2Former can better distinguish between huddled and similar instances, hence generating higher-quality segmentation maps.

4.3 Diagnostic Experiment

For thorough examination, we conduct a series of ablative studies on COCO [45] `val2017`. We adopt CondInst [32] as our base segmentation network with ResNet-50 [43] backbone. To perform extensive experiments, we train models for 12 epochs while keeping other hyper-parameters unchanged.

Key Component Analysis. First, we study the efficacy of the core components of our algorithm, *i.e.*, inter-scene instance discrimination loss $\mathcal{L}_{inter_mask}$ (*cf.* Eq. 5) and equivariance loss $\mathcal{L}_{equi}$ (*cf.* Eq. 8). In Table 2, 1*st* row gives the performance of our base segmenter CondInst. For 2*nd* and 3*rd* rows, the scores are obtained by adopting $\mathcal{L}_{inter_mask}$ and $\mathcal{L}_{equi}$ individually. And 4*th* row reports the scores of our full algorithm. **i)** 1*st vs* 2*nd* row: $\mathcal{L}_{inter_mask}$ leads to notable performance improvements against the baseline (*e.g.*, 35.5→**37.6** *AP*). This verifies our first core hypothesis, *i.e.*, learning large-scale instance separation benefits discriminative query embedding learning. **ii)** 1*st vs* 3*rd* row: $\mathcal{L}_{equi}$ greatly boots the performance of the baseline (*e.g.*, 35.5→**36.5** *AP*), evidencing our second core hypothesis, *i.e.*, exploiting the transformation equivariance nature of the task facilitates robust query-instance matching. **iii)** 1*st vs* 2*nd vs* 3*rd vs* 4*th* row: combing all contributions together results in the largest gain over the baseline (*e.g.*, 35.5→**38.1** *AP*). This suggests that $\mathcal{L}_{inter_mask}$ and $\mathcal{L}_{equi}$ are able to work in a collaborative manner, and confirms the effectiveness of our overall algorithmic design.

Training Speed. We also report training speed in Table 2; related statistics are gathered on eight A100 GPUs with a batch size of 16. We can find that, compared with the current scene-wise training regime, our cross-scene training algorithm only brings slight delay (∼5%). More specifically, the computation of our inter-scene instance discrimination loss $\mathcal{L}_{inter_mask}$ takes relatively more time, due to the use of an external memory for large-scale query-instance matching.

Next we study some core designs for inter-scene instance discrimination loss $\mathcal{L}_{inter_mask}$ (*cf.* Eq. 5). The results in Tables 3a, 3b, and 3c are reported without considering equivariance loss $\mathcal{L}_{equi}$ (*cf.* Eq. 8).

Loss Design for Inter-Scene Instance Uniqueness Learning. In Table 3a, we investigate four different implementation forms of our inter-scene instance discrimination loss $\mathcal{L}_{inter_mask}$, namely ℓ_1 loss, ℓ_2 loss, cross-entropy loss, and focal loss [62]. We can find that focal loss is more favored. In our case, most instance (pixel) samples from other scenes can be easily recognized as negative, causing a huge imbalance between hard and easy negative samples. Therefore, training under such an environment results in ℓ_1 loss, ℓ_2 loss, and cross-entropy loss focusing more on easy samples. In

Table 3: A set of ablative studies on COCO [45] `val2017`. The adopted algorithm designs and hyper-parameter settings are marked in red. See §4.3 for details.

$\mathcal{L}_{inter_mask}$	AP	AP_S	AP_M	AP_L
-	35.5	16.9	38.9	50.2
ℓ_1 loss	36.0	17.1	39.2	51.7
ℓ_2 loss	35.9	17.6	39.1	51.6
cross-entropy	37.3	18.1	40.6	53.3
focal loss	37.6	18.6	40.6	54.0

(a) inter-scene instance discrimination loss $\mathcal{L}_{inter_mask}$ (*cf.* Eq. 5)

sampling	AP	AP_S	AP_M	AP_L
dense	37.1	18.0	40.4	53.0
sparse (0.5k pixel/img.)	37.2	17.8	40.5	53.2
sparse (1.0k pixel/img.)	37.3	17.9	40.5	53.4
instance-balanced (10 pixel/ins.)	37.4	18.7	40.5	53.6
instance-balanced (50 pixel/ins.)	37.6	18.6	40.6	54.0

(b) sampling strategy

capacity	AP	AP_S	AP_M	AP_L
-	35.5	16.9	38.9	50.2
mini-batch (*w/o* memory)	36.9	17.8	40.1	53.1
10k pixels	37.0	17.9	40.3	53.2
50k pixels	37.3	18.1	40.4	53.6
70k pixels	37.5	18.5	40.4	53.7
100k pixels	37.6	18.6	40.6	54.0

(c) memory capacity

transformation equivariance	AP	AP_S	AP_M	AP_L
w/o Aug.	35.3	16.2	39.2	49.9
w/ Aug. ($\{\langle \boldsymbol{q}_n^g, \boldsymbol{I}^g \rangle\}_n \approx \{g(M_{\sigma(n)})\}_n$)	35.5	16.9	38.9	50.2
feature *only* ($\boldsymbol{I}^g \approx g(\boldsymbol{I})$)	36.3	16.5	39.5	51.4
$\mathcal{L}_{equi}$ ($\{\langle \boldsymbol{q}_n^g, g(\boldsymbol{I}) \rangle\}_n \approx \{g(M_{\sigma(n)})\}_n$)	36.5	17.1	39.5	51.6

(d) transformation equivariance modeling

coefficient	AP	AP_S	AP_M	AP_L
1	37.0	17.9	40.4	52.1
3	37.6	18.6	40.6	54.0
5	37.5	18.3	40.7	53.6
10	37.3	18.1	40.8	53.3

(e) coefficient between $\mathcal{L}_{inter_mask}$ and $\mathcal{L}_{equi}$

contrast, focal loss drives the training more towards the sparse set of hard negative samples, thus preventing the gradient of $\mathcal{L}_{inter_mask}$ from being dominated by the massive, easy negative samples.

Sampling Strategy. We further study the impact of the three sampling strategies (*cf.* §3.2) for the computation of $\mathcal{L}_{inter_mask}$, namely dense sampling (storing all the pixels of each image into the memory), sparse sampling (randomly sampling a small set, *e.g.*, 0.5k or 1.0k, of pixels from each image), and instance-balanced sampling (randomly sampling a small fixed-size set, *e.g.*, 10 or 50, of pixels from each instance). Table 3b proves that, sparse sampling works overall better than dense sampling, as it improves the diversity of the stored samples. However, sparse sampling leads to inferior performance on small instances, *e.g.*, 18.0 *vs* 17.8 AP_S for 'sparse (0.5k pixel/img.)', as smaller instances are less likely sampled for loss computation. Fortunately, instance-balanced sampling inherits sparse sampling's desired merits but without its defects, hence yielding more impressive results.

Memory Capacity. Table 3c shows the influence of the capacity of the external memory. 1*st* row gives the results of the base segmenter. 2*nd* row lists the scores obtained by computing $\mathcal{L}_{inter_mask}$ within each mini-batch, which are already much better, *e.g.*, 35.5 *vs* **36.9** *AP*. 3*rd* - 5*th* rows demonstrate that, **i)** with the aid of an external memory, the performance can be further boosted; and **ii)** enlarging memory is always helpful. These results verify our key idea of learning large-scale query-instance matching. Note that the optimal configuration, *i.e.*, 100k pixels, does not yet reach the point of performance saturation, but rather the upper limit of our hardware's computational budget.

Transformation Equivariant Learning. We next study our equivariance loss design $\mathcal{L}_{equi}$ (*cf.* Eq. 8). The results in Table 3d are reported without considering the inter-scene instance discrimination loss $\mathcal{L}_{inter_mask}$ (*cf.* Eq. 5). The first two rows respectively present the performance of the base segmenter *w/o* and *w/* transformation-based data augmentation. As we mentioned before, training with current data augmentation technique can be viewed as learning equivariant query embeddings (*cf.* Eq. 7). In 3*rd* row we show the performance by only encouraging equivariant image representation learning (*cf.* Eq. 6). Comparing these three baselines with our final equivariance loss $\mathcal{L}_{equi}$ that simultaneously addresses equivariant image representation and query embedding learning, we can find: **i)** exploring the equivariance nature of the task is indeed essential; and **ii)** our equivariance loss $\mathcal{L}_{equi}$ is much more effective than current transformation-based data augmentation technique (*e.g.*, **36.5** *vs* 35.5 *AP*).

Loss Term Coefficient. For completeness, the results with different coefficients, which control the balance between our two training objectives, *i.e.*, $\mathcal{L}_{inter_mask}$ and $\mathcal{L}_{equi}$, are reported in Table 3e.

5 Conclusion and Discussion

Aiming at sharpening the instance discrimination ability of query-based segmenters, we devise a novel framework that formulates two crucial properties of the query-instance relationship, *i.e.*, *dataset-level uniqueness* and *transformation equivariance*, as network training targets. Extensive empirical analysis demonstrates that our method is flexible yet powerful, allowing it to benefit the existing and growing body of query-based instance segmentation methods. We argue deductively that the proposed training framework has the potential to be applied to broader range of dense prediction tasks, *i.e.*, query-based detection and panoptic segmentation. These questions remain open for our future endeavor.

References

[1] Pedro O Pinheiro, Ronan Collobert, and Piotr Dollar. Learning to segment object candidates. In *NeurIPS*, 2015. 1, 2

[2] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, 2015. 1, 2

[3] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *CVPR*, 2016. 1, 2

[4] Pedro O Pinheiro, Tsung-Yi Lin, Ronan Collobert, and Piotr Dollár. Learning to refine object segments. In *ECCV*, 2016. 1, 2

[5] Bernardino Romera-Paredes and Philip Hilaire Sean Torr. Recurrent instance segmentation. In *ECCV*, 2016. 1, 2

[6] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 1, 2, 3, 6, 7

[7] Mengye Ren and Richard S Zemel. End-to-end instance segmentation with recurrent attention. In *CVPR*, 2017. 1, 2

[8] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *CVPR*, 2018. 1, 2

[9] 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 *CVPR*, 2018. 1, 2

[10] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring r-cnn. In *CVPR*, 2019. 1, 2

[11] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: high quality object detection and instance segmentation. *IEEE TPAMI*, 43(5):1483–1498, 2019. 1, 2, 6, 7

[12] 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*, 2019. 1, 2, 6, 7

[13] Hao Chen, Kunyang Sun, Zhi Tian, Chunhua Shen, Yongming Huang, and Youliang Yan. Blendmask: Top-down meets bottom-up for instance segmentation. In *CVPR*, 2020. 1, 2, 3

[14] Tianheng Cheng, Xinggang Wang, Lichao Huang, and Wenyu Liu. Boundary-preserving mask r-cnn. In *ECCV*, 2020. 1, 2

[15] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *CVPR*, 2020. 1, 2, 5, 6, 7

[16] Lei Ke, Yu-Wing Tai, and Chi-Keung Tang. Deep occlusion-aware instance segmentation with overlapping bilayers. In *CVPR*, 2021. 1, 2, 7

[17] Thang Vu, Haeyong Kang, and Chang D Yoo. Scnet: Training inference sample consistency for instance segmentation. In *AAAI*, 2021. 1, 2

[18] Yuxin Fang, Shusheng Yang, Xinggang Wang, Yu Li, Chen Fang, Ying Shan, Bin Feng, and Wenyu Liu. Instances as queries. In *ICCV*, 2021. 1, 2, 3, 6, 7

[19] Anurag Arnab and Philip HS Torr. Bottom-up instance segmentation using deep higher-order crfs. *arXiv preprint arXiv:1609.02583*, 2016. 1, 2

[20] Anurag Arnab and Philip HS Torr. Pixelwise instance segmentation with a dynamically instantiated network. In *CVPR*, 2017. 1, 2

[21] Min Bai and Raquel Urtasun. Deep watershed transform for instance segmentation. In *CVPR*, 2017. 1, 2

[22] Shu Liu, Jiaya Jia, Sanja Fidler, and Raquel Urtasun. Sgn: Sequential grouping networks for instance segmentation. In *ICCV*, 2017. 1, 2

[23] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *NeurIPS*, 2017. 1, 2

[24] Shu Kong and Charless C Fowlkes. Recurrent pixel embedding for instance grouping. In *CVPR*, 2018. 1, 2

[25] Davy Neven, Bert De Brabandere, Marc Proesmans, and Luc Van Gool. Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth. In *CVPR*, 2019. 1, 2

[26] Naiyu Gao, Yanhu Shan, Yupei Wang, Xin Zhao, Yinan Yu, Ming Yang, and Kaiqi Huang. Ssap: Single-shot instance segmentation with affinity pyramid. In *ICCV*, 2019. 1, 2

[27] Jifeng Dai, Kaiming He, Yi Li, Shaoqing Ren, and Jian Sun. Instance-sensitive fully convolutional networks. In *ECCV*, 2016. 1, 2

[28] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *CVPR*, 2017. 1, 2

[29] Xinlei Chen, Ross Girshick, Kaiming He, and Piotr Dollár. Tensormask: A foundation for dense object segmentation. In *ICCV*, 2019. 1, 2

[30] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *ICCV*, 2019. 1, 2, 3

[31] Jiale Cao, Rao Muhammad Anwer, Hisham Cholakkal, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. Sipmask: Spatial information preservation for fast image and video instance segmentation. In *ECCV*, 2020. 1, 2

[32] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. In *ECCV*, 2020. 1, 2, 3, 4, 6, 7, 8, 13

[33] 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*, 2020. 1, 2

[34] Youngwan Lee and Jongyoul Park. Centermask: Real-time anchor-free instance segmentation. In *CVPR*, 2020. 1, 2

[35] Yuqing Wang, Zhaoliang Xu, Hao Shen, Baoshan Cheng, and Lirong Yang. Centermask: single shot instance segmentation with point representation. In *CVPR*, 2020. 1, 2

[36] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. Solo: Segmenting objects by locations. In *ECCV*, 2020. 1, 2

[37] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. SOLOv2: Dynamic and fast instance segmentation. In *NeurIPS*, 2020. 1, 2, 3, 4, 6, 7, 13

[38] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022. 1, 2, 3, 4, 6, 7, 8, 13

[39] Ruohao Guo, Dantong Niu, Liao Qu, and Zhenbo Li. Sotr: Segmenting objects with transformers. In *ICCV*, 2021. 1, 2, 3, 6, 7, 13

[40] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *NeurIPS*, 2021. 1, 2, 3, 4

[41] Bin Dong, Fangao Zeng, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. Solq: Segmenting objects by learning queries. In *NeurIPS*, 2021. 1, 2, 3, 6, 7

[42] 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*, 2022. 1, 2, 6, 7

[43] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 6, 7, 8

[44] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 2, 6, 7

[45] 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. 2, 6, 7, 8, 9

[46] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *CVPR*, 2017. 2

[47] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, 2019. 2

[48] Wenchao Gu, Shuang Bai, and Lingxing Kong. A review on 2d instance segmentation based on deep neural networks. *Image and Vision Computing*, 2022. 2

[49] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. In *NeurIPS*, 2016. 2

[50] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 2, 3

[51] Guo-Jun Qi, Liheng Zhang, Feng Lin, and Xiao Wang. Learning generalized transformation equivariant representations via autoencoding transformations. *IEEE TPAMI*, 2020. 3

[52] Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-encoders. In *International Conference on Artificial Neural Networks*, 2011. 3

[53] Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with em routing. In *ICRL*, 2018. 3

[54] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *ICML*, 2016. 3

[55] Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Harmonic networks: Deep translation and rotation equivariance. In *CVPR*, 2017. 3

[56] Uwe Schmidt and Stefan Roth. Learning rotation-aware features: From invariant priors to equivariant descriptors. In *CVPR*, 2012. 3

[57] Kihyuk Sohn and Honglak Lee. Learning invariant representations with local transformations. In *IMCL*, 2012. 3

[58] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 3, 5

[59] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 3

[60] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021. 3

[61] Rumen Dangovski, Li Jing, Charlotte Loh, Seungwook Han, Akash Srivastava, Brian Cheung, Pulkit Agrawal, and Marin Soljačić. Equivariant contrastive learning. In *ICLR*, 2022. 3

[62] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 3, 5, 6, 8

[63] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3DV*, 2016. 3, 6

[64] Kemal Oksuz, Baris Can Cam, Emre Akbas, and Sinan Kalkan. Rank & sort loss for object detection and instance segmentation. In *ICCV*, 2021. 3

[65] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 4

[66] Xun Wang, Haozhi Zhang, Weilin Huang, and Matthew R Scott. Cross-batch memory for embedding learning. In *CVPR*, 2020. 5

[67] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 5

[68] Wenguan Wang, Tianfei Zhou, Fisher Yu, Jifeng Dai, Ender Konukoglu, and Luc Van Gool. Exploring cross-image pixel contrast for semantic segmentation. In *ICCV*, 2021. 5

[69] Zhi Tian, Hao Chen, Xinlong Wang, Yuliang Liu, and Chunhua Shen. AdelaiDet: A toolbox for instance-level recognition tasks. `https://git.io/adelaidet`, 2019. 6

[70] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 6

[71] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 7

[72] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 7

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] Our main claims accurately reflect the paper's contributions and scope.
   (b) Did you describe the limitations of your work? [Yes] The limitations are discussed in §D of the supplemental material.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] The potential negative societal impacts are discussed in §D of the supplemental material.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] We have already read the ethics review guidelines and ensured that our paper conforms to them.
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results?[N/A] Our work does not involve any theoretical result.
   (b) Did you include complete proofs of all theoretical results?[N/A] Our work does not involve any theoretical result.

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] The pseudo-code is included in the supplemental material. The datasets used in this work are public and well cited in our paper. We believe we have provided sufficient details of our algorithm. We promise our code shall be publicly released right after acceptance.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We report the training details and hyperparameters in §4.1.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We follow the standard evaluation protocol in this field [32,37–39] to report and compare the results. Running all our experiments multiple times is too expensive to afford.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We report our hardware specification in the Reproducibility part of §3.3.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We cite all used codes, data, and models in this paper.
   (b) Did you mention the license of the assets? [Yes] CondInst and SOLOv2 are implemented based on their released code (`https://github.com/aim-uofa/AdelaiDet`) with Apache2.0 license. SOTR is implemented based on its released code (`https://github.com/easton-cau/SOTR`) with MIT license. Mask2former is implemented based on the reproducing by *MMDetection* (`https://github.com/open-mmlab/mmdetection/tree/master/configs/mask2former`) with Apache2.0 license.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] SparseInst implemented based on its released code (`https://github.com/hustvl/SparseInst`) with MIT license. SOLQ implemented based on its released code (`https://github.com/megvii-research/SOLQ`) with Apache2.0 license.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] MS COCO dataset is public and licensed under a Creative Commons Attribution 4.0 License.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] The data used in our paper neither contain any personally identifiable information nor offensive content.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]