Cornerformer: Purifying Instances for Corner-based Detectors

Haoran Wei[1,⋆], Xin Chen[2], Lingxi Xie[2], and Qi Tian[2]

[1] University of Chinese Academy of Sciences
[2] Huawei Inc.
weihaoran18@mails.ucas.ac.cn
{chenxin061,198808xc}@gmail.com
tian.qi1@huawei.com

Abstract. Corner-based object detectors enjoy the potential of detecting arbitrarily-sized instances, yet the performance is mainly harmed by the accuracy of instance construction. Specifically, there are three factors, namely, 1) the corner keypoints are prone to false-positives; 2) incorrect matches emerge upon corner keypoint pull-push embeddings; and 3) the heuristic NMS cannot adjust the corners pull-push mechanism. Accordingly, this paper presents an elegant framework named Cornerformer that is composed of two factors. First, we build a Corner Transformer Encoder (CTE, a self-attention module) in a 2D-form to enhance the information aggregated by corner keypoints, offering stronger features for the pull-push loss to distinguish instances from each other. Second, we design an Attenuation-Auto-Adjusted NMS (A^3-NMS) to maximally leverage the semantic outputs and avoid true objects from being removed. Experiments on object detection and human pose estimation show the superior performance of Cornerformer in terms of accuracy and inference speed.

Keywords: Object detection, Corner-based, Corner Transformer encoder, Attenuation-Auto-Adjusted NMS

1 Introduction

Object detection, which aims to localize and classify objects of interest in an image, is an active and fundamental research direction in computer vision. Current state-of-the-art detectors can be roughly classified into two categories, *i.e.*, anchor-based [13,14,3,26,31,23] and anchor-free. Recently, anchor-free detectors [20,43,11,9,37,42] have become a research hotspot due to its flexibility and efficiency, among which, corner-based detector (*e.g.*, CornerNet [20] and its variants [11,9]) is one of the most popular flowchats.

One key factor of corner-based detectors is how to construct instances from corner points. CornerNet [20] proposed a corner pooling module using a serial operation of maximize-and-merge operations to enhance corner features, and applied a grouping method upon pull-push loss [28,20] to formulate corner points

⋆ This work was done when the first author was interning at Huawei Inc.

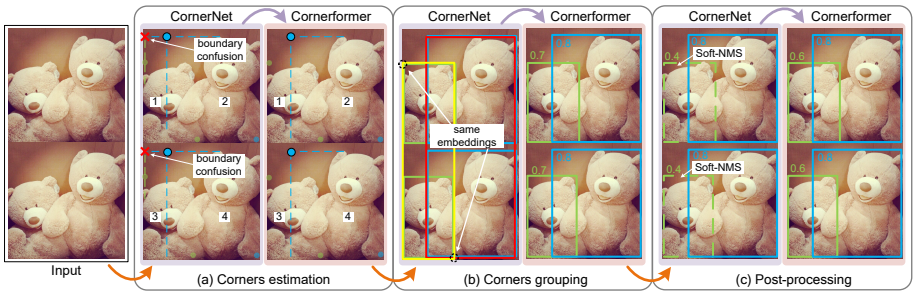

Fig. 1. Three problems of instance construction in corner-based detectors. (a) Boundries of nearby objects may coincide with each other, resulting in high-score false positives that belong to no object (red "×" upon the top-left corner). (b) The pull-push mechanism is prone to confusing highly similar keypoints that belongs to different objects, so as to produce predictions across objects (red and yellow bounding boxes). Here we show an extreme case that objects are identical. (c) The commonly used Soft-NMS improperly decays lower-scored bounding box of two overlapped objects to be removed in visualization (lower than 0.5-score).

into instances (bounding boxes), which minimize embedding distances of corner keypoints that belong to the same object and maximize those of different ones. To further enhance the correctness of corner grouping, CenterNet [11] introduced center points to filter false matched bounding boxes, while CentripetalNet [9] abandoned the 1D pull-push embeddings and presented a centripetal grouping method with a 2D-embedding form to better group paired corners.

As mentioned above, the correctness of corner grouping is one of the key factors in corner-based detectors. Although great progress has been made by previous methods [20,11,9] in corner matching, it is still an urgent demand to further improve the correctness and accuracy of instance construction of corner-based detectors. We believe the key factors that obstruct better instance construction lie in the following three aspects: 1) Corner keypoints are prone to false-positives due to boundary confusion that may largely disturb subsequent steps in the pipeline and produce inferior results (Figure 1(a)). 2) Incorrect matches emerge upon corner keypoint pull-push embeddings, resulting in irregular detection boxes, *e.g.*, boxes containing multiple objects (Figure 1(b)). 3) The heuristic NMS cannot adjust the corners pull-push mechanism due to its fixed decay factor setting and thus falsely discarding overlapped instances (Figure 1(c)).

Accordingly, we propose a Cornerformer framework that composed of two main modules, *i.e.,* a Corner Transformer Encoder (CTE, a self-attention module) and an Attenuation-Auto-Adjusted NMS (A^3-NMS), to efficiently address above stated drawbacks of corner-based detectors. Specifically, the CTE module is designed to be a 2D-Transformer, which can better capture nearby boundary information of a corner location with the help of self-attention mechanism, so as to effectively alleviate the boundary confusion problem and decrease false-

positive keypoints. Additionally, we implant multiple positional encodings into CTE, which makes CTE position-sensitive and dramatically enhance the distinguishing ability on similar keypoints from different objects. An A^3-NMS is a hyper-parameter-free Soft-NMS that can dynamically adjust attenuation weights of boxes to maximally leverage the semantic outputs and avoid true objects from being removed. Thus, the A^3-NMS is more suitable for the scenario of the corner pull-push mechanism than a vanilla Soft-NMS [2] that has a fixed decay weight.

Experimental results on MS-COCO [24] object detection, and both MS-COCO and CrowdPose [22] human pose estimation show that after equipping with the proposed Cornerformer, state-of-the-art corner-based detectors enjoy a consistent performance improvement in terms of accuracy and inference speed. Taking the classic CornerNet baseline as an example, after replacing corner pooling and Soft-NMS with CTE and A^3-NMS in Cornerformer, we obtain a satisfactory accuracy boosting of 3.0% in terms of AP and an inference speed lifting of 2.5 FPS.

2 Related work

The success of deep neural networks (DNNs) [21,19] has largely promoted the development of object detection. Modern DNNs-based detectors can be simple divided into two categories: anchor-based [13,14,3,26,31,23] and anchor-free [20,43,11,9,37,42].

2.1 Anchor-based Detector

The anchor box is used to match a ground truth box and acts as a guidance for detectors to regress object bounding box. In Faster R-CNN [33], the design of anchor-based RPN made detectors end-to-end trainable. Later, anchor boxes were widely used in RPN-based two-stage detectors [13,33,41,14,3,27]. To further explore the efficiency of models, some anchor-based one-stage detectors [34,26,31,36,23,32] also appeared. They remove the RPN-stage and directly regress and classify anchor boxes. Despite the great success of the anchor mechanism, it also brings some drawbacks, *e.g.*, excessively many hyper-parameters, unstable IoU-based (>0.7) positives selection strategy, and complex network structure. These drawbacks drive the community to study anchor-free detection methods.

2.2 Anchor-free Detector

Anchor-free object detection is a very active research field in recent years. CornerNet [20] was the first keypoint-based approach, which predicted keypoints via generating and parsing heatmaps, and detected objects by predicting and grouping pairs of corner points. CenterNet [11] added a prediction branch of center points based on CornerNet settings, transforming corner matching into triplet matching. Upon CornerNet, CentripetalNet [9] proposed a new centripetal

grouping algorithm and achieved state-of-the-art performance. Besides, FCOS [37] proposed a dense regression anchor-free detector. It treats lots of pixels in bounding boxes as positive samples and directly regresses bounding boxes. Compared with anchor-based approaches, anchor-free methods enjoy flexibility and efficiency. However, the limitations of local modeling in CNN hinder its development.

Transformer is first proposed in natural language processing [38]. Compared with CNN, Transformer inplants self-attention mechanism into its basic operator, which is more suitable for capturing long-range contexts than convolution. Due to its superb ability, Transformers were introduced into computer vision [10,5,44,25] and soon leveraged in object detection. DETR [5] introduced Transformer [38] into object detection task for the first time, which eliminated many hand-craft modules (*e.g.,* anchor, NMS, and proposals) in previous detectors, and achieved on-par performance compared to classical CNN-based Faster R-CNN. Based on DETR, WB-DETR [25] replaced the ResNet [15] backbone with ViT [10] (a Transformer-based recognition system) to obtain a pure-Transformer detection system, making the detection pipeline neater.

Inspired by DETR, we explore how to integrate Transformer into keypoint-based detectors to improve the quality of instance construction.

3 Preliminary: The CornerNet Baseline

CornerNet utilizes heatmaps generated by the backbone to estimate corner keypoints and uses the pull-push loss to group embedding pairs. To refine corner coordinates extracted from heatmaps, CornerNet predict extra offsets. The pipeline of CornerNet is similar to the pipeline shown in Figure 2, with a corner pooling module replacing the CTE module, and a Soft-NMS module replacing the A^3-NMS module.

CornerNet applies a modified pixel-level focal loss [23] as the training objective for heatmaps of paired corners. The backbone with output stride brings about discretization error in the process of remapping corner locations. To address this problem, CornerNet additionally regress offsets to refine corner coordinates.

The Associative Embedding [28] method is applied for paired corner matching. More specifically, the "pull" loss is leveraged to group paired corners and the "push" loss to separate irrelevant corners:

$$\mathcal{L}_{pull} = \frac{1}{N}\sum_{k=1}^{N}\left[\left(e_{t_k}-e_k\right)^2+\left(e_{b_k}-e_k\right)^2\right], \tag{1}$$

$$\mathcal{L}_{push} = \frac{1}{N(N-1)}\sum_{k=1}^{N}\sum_{\substack{j=1\\ j\neq k}}^{N}\max\left(0,\Delta-\left|e_k-e_j\right|\right), \tag{2}$$

where e_k is the average of e_{t_k} and e_{b_k} and Δ is set to be 1. Pull-push loss is only applied at ground-truth corner locations [20].

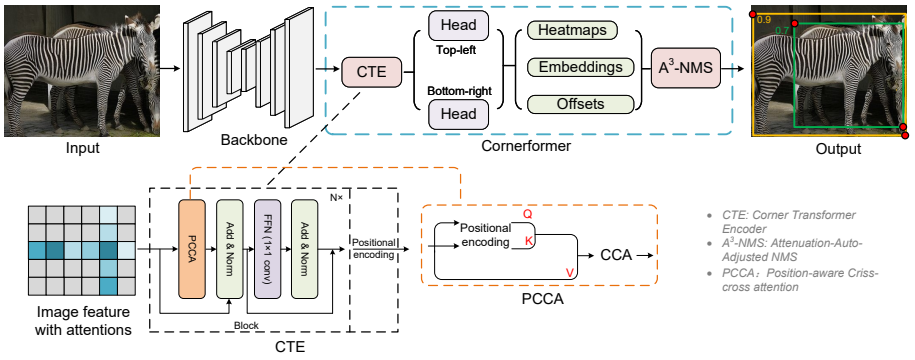

Fig. 2. The architecture of Cornerformer. A Cornerformer, composed of a CTE (Corner Transformer Encoder) module, keypoint grouping components, and an A^3-NMS module, takes image features extracted by the backbone as input, and generates predicted bounding boxes (paired corner keypoints) as output. The CTE, as a replacement of corner pooling module in CornerNet, captures contextual information and predicts corner keypoints (embeddings) more precisely from input feature maps. A^3-NMS is an improved NMS, which can maximally avoid true objects from being removed and is no longer restricted by manually set hyper-parameters.

Our Cornerformer is built upon the CornerNet baseline (or its variants, *i.e.*, CenterNet and CentripetalNet). In the rest of this paper, we apply the same settings used in corresponding baselines unless otherwise specified. For more details, please refer to the original papers [20,11,9].

4 Cornerformer

4.1 Towards Better Instance Construction

As stated in Section 1, there are three critical factors that hinder a better instance construction in corner-based object detection. Here we carry out a case analysis incorporating with Figure 1 to further dissect them. 1) **Corner keypoints are prone to false-positives due to boundary confusion**. As shown in Figure 1 (a). The top-most boundary of "teddy bear 2" and the left-most boundary of "teddy bear 1" coincides with each other, which will result in a high-score false estimation that neither belongs to "teddy bear 1" nor "teddy bear 2". Such needless keypoints will largely affect subsequent procedures and produce deteriorative predictions. 2) **Incorrect matches emerge upon corner keypoint pull-push embeddings.** The pull-push mechanism naturally lacks ability of distinguishing highly similar keypoints that belong to different objects, so that corner keypoints of different objects with similar appearances may be mistakenly grouped together. We show an extreme example that the input is generated by copy-and-paste the same image in Figure 1 (b). The pre-

dicted bounding boxes (red and yellow) are unnaturally composed of two different instances. 3) **The heuristic NMS cannot adjust the corners pull-push mechanism.** The commonly used Soft-NMS decays scores of overlapped bounding boxes with a fixed value, which may cause true detection boxes of overlapped objects being removed in visualization, as shown in Figure 1 (c).

Accordingly, we propose Cornerformer to address above stated problems of corner-based detectors in instance construction. As shown in Figure 2, Cornerformer is built upon a corner-based detector as baseline, *e.g.*, CornerNet [20] or CenterNet [11]. A Cornerformer consists of two main components, *i.e.*, a Corner Transformer Encoder (CTE) used to provide better corner features and lift the quality of pull-push embeddings, and an Attenuation-Auto-Adjusted NMS (A^3-NMS) used to maximally leverage the semantic outputs and avoid true objects from being removed. In the following, we delve into each part of Cornerformer and show how it helps a better instance construction.

4.2 Corner Transformer Encoder

Overview of the CTE. Corner Transformer Encoder (CTE) runs as a corner features enhancing module in Cornerformer. Corners often lay outside an object without explicit existent evidence, which needs feature enhancement via boundary contexts (the top-most, bottom-most, left-most, and right-most of an object). The CornerNet baseline applies a corner pooling module to enhance the visual reasoning of corners. However, corner pooling uses a serial operation of maximize-and-merge to extend corner information, suffering inefficient context around corner keypoints. As a special case, when a spatial location has similar boundary conditions, it is likely to be a high-score false corner estimation due to boundary confusion, as shown in Figure 1 (a). In addition, there is no optimization for pull-push embeddings in corner pooling, so that the pull-push mechanism may gather wrong corner pairs that have highly similar corner embeddings, as shown in Figure 1 (b). To this end, we propose CTE to better estimate and group target corners.

Position-aware Criss-Cross attention. To capture rich boundary contexts, the corner estimation module needs to look over information around potential corner keypoints, *i.e.*, horizontal and vertical possible spatial locations related to boundaries. Thus, we naturally adopt the self-attention mechanism to capture rich information around potential corners. More specifically, for one query element q in image features, the corresponding keys (k) are sampled in the same row and column. It is notable that convolution can also capture information from a nearby area, but handling a larger area corresponding to a large object requires to stack more convolutional layers, which brings potential computational burdens.

As shown in Figure 2, we adopt the Position-aware Criss-Cross Attention (PCCA) as the self-attention module, which is designed upon CCA [16] and enhanced with positional encoding. Vanilla CCA is used only to capture contextual information in horizontal and vertical directions for using light-weight computation and memory, which we find suitable for capturing boundary contextual

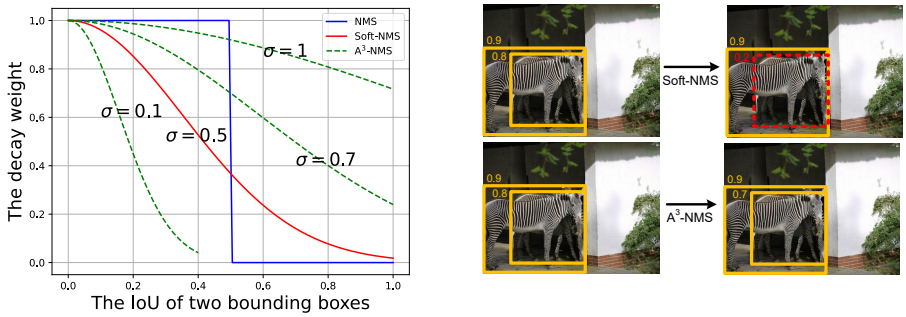

Fig. 3. Comparison of NMS, Soft-NMS, and A^3-NMS. Soft-NMS uses a fixed decay weight value (σ) in both the scenario of decaying redundant boxes, where σ should be as large as possible to remove redundant boxes, and the scenario of overlapped objects, where σ should be as small as possible to retain bounding boxes of different objects. In contrast, our A^3-NMS accomplishes an adaptive decay ratio upon embeddings to effectively address such scenarios.

information for corner keypoints. To further alleviate the false-positive problem of similar features, we equip a positional encoding [5] in "Q" and "K" of the original CCA to create a new Position-aware CCA. The positional encoding is as follows:

$$\begin{cases} \mathrm{PE}(pos, 2i) = \sin(pos/10000^{2i/d}) \\ \mathrm{PE}(pos, 2i+1) = \cos(pos/10000^{2i/d}), \end{cases} \tag{3}$$

where pos means the position, i indicates the channel ID, and d is 128 representing the total number of dimensions.

Settings of Transformer blocks. One Transformer block in CTE is composed of a PCCA (as the self-attention module) and a 1×1 convolutional layer (as the feed forward network, FFN). Different from the vanilla 1D-Transformer block [5] that expects a sequence as input, the CTE is a 2D-form Transformer which can take 2D image features as input directly. We also find that multiple Transformer blocks in series can generate more robust features for corner estimation.

Optimizing pull-push embeddings. We use the pull-push loss to group corner pairs, as mentioned in Section 3. The pull-push loss utilizes a self-supervised way to "pull" embeddings of corresponding corners and "push" irrelevant ones upon object features. When there are more than one objects with similar appearance in an image, as shown in Figure 1 (b), the pull-push loss fails to distinguish different instances effectively. Thus, except for richer contexts embedded in corner keypoints, we further insert a global positional encoding after Transformer blocks to make CTE position-sensitive, so that the pull-push loss can better distinguish if a pair of corners that have similar feature responses belong to the same instance.

4.3 Attenuation-Auto-Adjusted NMS

Overview of the A^3-NMS. The heuristic Non-Maximum Suppression (NMS) has become the *de-facto* standard applied to suppress and filter out false-positives for detectors. The original NMS cannot solve the problem of largely overlapped objects. Soft-NMS addresses such problem by softening the suppression process with a score decay mechanism. The most commonly used decay function is as follows:

$$W_{iou(M,b_i)} = \exp(-\frac{iou(M,b_i)^2}{\sigma}) \tag{4}$$

where M is the highest score of box, b_i indicates the current box. σ is a hyper-parameter often set to be 0.5.

However, Soft-NMS still lacks flexibility because of its fixed σ value. As long as the IoU of two bounding boxes is the same, Soft-NMS decays the lower-scored one to the same score, whatever the scenario is overlapped instances or duplicated bounding boxes. To address such a limitation, we design the A^3-NMS that introduces adjustable attenuation and relies on no mannually set hyper-parameters to flexibly handle both cases of duplicated boxes and overlapped instances.

Adjust attenuation upon embeddings. To address the above problem, A^3-NMS applies a dynamic adjusted decay function as following :

$$W_{iou(M,b_i)} = \exp(-\frac{iou(M,b_i)^2}{f(|e_M - e_i|)}) \tag{5}$$

where e_M and e_i are embeddings of corresponding boxes. M represents the max-score box. f is a function to smooth embedding distance between e_M and e_i, and in this case we use the *tanh* function. We take mean value of paired corner embeddings as box embedding, *i.e.*, e_M or e_i. Thus, we can easily obtain distance of bounding boxes by calculating vector distance of their box embeddings. Since paired corners or boxes are grouped via the pull-push mechanism in corner-based detectors, the distance of box embeddings should be small if two boxes belong to the same instance, while it should be large if two boxes belong to different objects. The fixed σ in Eqn. (4) becomes a dynamic value learned by the network in Eqn. (5) so that the adjusted attenuation factor can well fit each situation. We plot curves of the decay function of different versions of NMS in Figure 3, from which we can easily find that A^3-NMS can efficiently handle such scenarios.

More specifically, A^3-NMS runs in post-processing, and (e_M, e_i) are a pair of self-adjusted embeddings for the corresponding boxes, which belong to the side outputs of corner-based detectors. They contain useful information that measures the closeness of two detection boxes. That said, the smaller $f(|e_M - e_i|)$ is, the more similar b_i and b_M are, and thus the heavier b_i is suppressed. Compared to Soft-NMS that relies on a fixed factor σ, such a mechanism is more flexible and accurate.

Table 1. Performance comparison (%) with state-of-the-art detectors on MS COCO test-dev. DR and KB are abbreviations of dense regression and keypoint-based, respectively. ×2 means that two Corner Transformer Encoders without parameters sharing are used for the top-left and bottom-right corners, respectively. Blue up-arrows indicate the improved values compared with baselines. * indicates multi-scale testing. Input resolution represents training input size.

Method	Backbone	Input Size	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AP	FPS
Two-stage:									
Cascade R-CNN [3]	ResNet-101	1333 × 800	62.1	46.3	23.7	45.5	55.2	42.8	-
Sparse R-CNN [35]	ResNeXt-101	1333 × 800	**66.3**	51.2	28.6	49.2	58.7	46.9	-
CPN [12]	Hourglass-104	1333 × 800	65.0	**51.0**	26.5	**50.2**	**60.7**	47.0	5.2
One-stage, anchor-based:									
RetinaNet [23]	ResNeXt-101	1333 × 800	61.1	44.1	24.1	44.2	51.2	40.8	5.4
YOLOv4 [1]	CSPDarkNet-53	608 × 608	65.7	47.3	26.7	46.7	53.3	43.5	-
ATSS [40] w/ DCN [8]	ResNet-101	1333 × 800	64.7	50.4	27.7	49.8	58.4	46.3	8.4
One-stage, anchor-free (DR)									
FoveaBox [18]	ResNeXt-101	1333 × 800	61.9	45.2	24.9	46.8	55.6	42.1	5.1
FCOS [37]	ResNeXt-101	1333 × 800	62.1	45.2	25.6	44.9	52.0	42.1	7.3
Reppoints [39] w/ DCN	ResNet-101	1333 × 800	66.1	49.0	26.6	48.6	57.5	45.0	8.7
One-stage, anchor-free (KB)									
ExtremeNet [43]	Hourglass-104	511 × 511	55.5	43.2	20.4	43.2	53.1	40.2	3.1
CenterNet [42]	Hourglass-104	512 × 512	61.1	45.9	24.1	45.5	52.8	42.1	7.8
CornerNet [20]	Hourglass-104	511 × 511	56.5	43.1	19.4	42.7	53.9	40.5	4.1
CornerNet w/ Cornerformer	Hourglass-104	511 × 511	61.2	46.2	21.5	44.8	56.5	42.6 ↑ 2.1	7.4
CornerNet w/ Cornerformer×2	Hourglass-104	511 × 511	61.7	46.8	22.3	45.6	57.1	43.5 ↑ 3.0	5.6
CenterNet [11]	Hourglass-104	511 × 511	62.4	48.1	25.6	47.4	57.4	44.9	3.3
CenterNet w/ Cornerformer	Hourglass-104	511 × 511	63.0	49.7	26.0	48.6	59.3	46.1 ↑ 1.2	5.9
CenterNet w/ Cornerformer×2	Hourglass-104	511 × 511	64.3	50.2	27.1	49.5	59.2	46.8 ↑ 1.9	4.8
CenterNet w/ Cornerformer×2*	Hourglass-104	511 × 511	64.9	51.4	28.8	50.1	59.5	47.7	-
CentripetalNet [9]	Hourglass-104	511 × 511	63.1	49.7	25.3	48.7	59.2	46.1	3.4
CentripetalNet w/ Cornerformer	Hourglass-104	511 × 511	64.5	50.3	26.2	49.4	59.6	47.1 ↑ 1.0	6.5
CentripetalNet w/ Cornerformer×2	Hourglass-104	511 × 511	64.6	50.6	26.7	49.5	59.9	47.4 ↑ 1.3	5.0
CentripetalNet w/ Cornerformer×2*	Hourglass-104	511 × 511	65.4	51.8	28.9	50.8	60.2	48.5	-

5 Experiments

5.1 Datasets, Metrics, and Implementation Details

Object detection. We evaluate the effectiveness of the proposed Cornerformer on COCO [24] dataset. COCO is a large-scale and challenging benchmark in object detection, which contains 80 categories and more than 1.5 million object instances. We train all models on the train2017 and carry out all ablations with val2017. We compare with other state-of-the-art methods using the test-dev. Besides, there are few heavily occluded objects in MS-COCO.

Besides, to better support our claim, we conduct additional ablation experiments on Citypersons [7] (a pedestrian detection dataset). Citypersons contains six different labels, *i.e.,* ignore regions, pedestrians, riders, sitting persons, other persons with unusual postures, and group of people. We keep and merge the labels of pedestrians and riders that accounts a large proportion in vanilla data. There are 18204 persons in 2471 images on our processed training set. We show

the performance of the proposed Cornerformer on the validation set that contains 439 images and 3666 persons.

Human pose estimation. To further test the generalization ability of the proposed CTE, we integrate it into the bottom-to-up human pose estimation model, HigherHRNet [6]. We evaluate the effectiveness of the CTE module on both COCO dataset, which contains $250k$ person instances labeled with 17 keypoints, and Crowdpose dataset [22], which contains more crowded scenes.

Metrics. We use the AP (average precision) metric to measure performance of both object detection and human pose estimation. AP in object detection (COCO) is computed over ten different IoU thresholds (*i.e.,* 0.5:0.05:0.95) and all categories, which is considered as the most important metric on the object detection task. For Citypersons, since the annotation (bounding box) is not as precise as COCO, the AP under a high IoU is meaningless, so we only test the AP with 0.5 IoU. Instead of IoU, the AP used in human pose estimation task is computed upon Object Keypoint Similarity (OKS): $\text{OKS} = \frac{\sum_i \exp(-{d_i}^2/2s^2k_i^2)\delta(v_i > 0)}{\sum_i \delta(v_i > 0)}$ Here, d_i represents the Euclidean distance between a detected keypoint and its corresponding ground truth, v_i is the visibility flag of the ground truth, s is the object scale, and k_i is a per-keypoint constant that controls falloff [6].

Training details. We follow settings of corner-based models to train new detector equipped with the proposed Cornerformer on 16 NVIDIA RTX 3090 GPUs. Standard cropping, horizontal flipping, and color jittering are employed as data augmentation. All models are fine-tuned from the pre-trained Hourglass [20] backbone with randomly initialized Cornerformer layers for $250k$ iterations with a batch size of 64. The learning rate is set to $2.5e{-}4$ and dropped $10\times$ at $200k$ iteration. An Adam [17] optimizer is applied to optimize model parameters. For human pose estimation task, we choose the HigherHRNet [6] as baseline and follow its training settings.

Inference details. We test inference speed of baseline models and their Cornerformer variants on a workstation with an NVIDIA Titan XP GPU. To guarantee a fair comparison with baselines, We strictly follow test settings of corresponding baselines.

5.2 Object Detection Results

We implant the proposed Cornerformer into several classical corner-based detectors, *e.g.*, CornerNet [20], CenterNet [11], and CentripetalNet [9], and evaluate the effectiveness of our design via performance comparisons, in terms of accuracy and inference speed.

Comparison on accuracy. As shown in Table 1, CornerNet [20] with Cornerformer improved by 2.1% on AP (from 40.5% to 42.6%), which shows the advantage of the proposed Cornerformer on better construct instances. For large objects, AP_L increases from 53.9% to 56.5%, which is arguably owing to the feasibility of CTE to well capture long-range contexts. As shown in Figure 4, the corners estimated by Cornerformer are more precise. For CenterNet [11] baseline, Cornerformer also brings it a decent promotion of 1.2% on AP. For

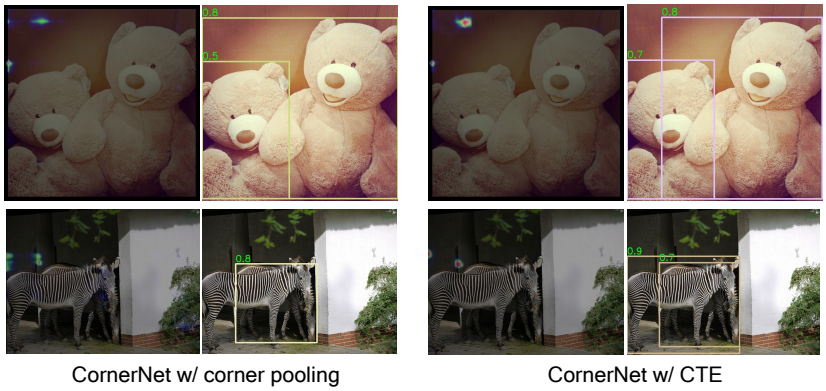

Fig. 4. Corner estimation visualized comparison of corner pooling and the proposed Corner Transformer Encoder. Compared with corner pooling, CTE can reduce false-positive points which have confused boundary visual appearances.

CentripetalNet [9] which does not use pull-push embeddings, we add a branch to obtain pull-push embeddings to apply the whole Cornerformer and harvest an improvement of 1.0%, proving our Cornerformer is very solid.

A single CTE in Cornerformer can capture both horizontal and vertical boundary information to enhance both top-left and bottom-right corners features. Under this setting, top-left and bottom-right corners share the same set of learned parameters, which may limit the representative power. To better distinguish different type of corners (top-left and bottom-right), we use two individual CTEs (represented as Cornerformer ($\times 2$)) to further test its ability. As we can see in Table 1, Cornerformer ($\times 2$), obtains improvements of 3.0%, 1.9%, and 1.3% upon CornerNet, CenterNet, and CentripetalNet baselines, respectively, further demonstrating the effectiveness of the proposed Cornerformer.

Comparison on inference speed. We also compare the inference speed of our method with baselines. As illustrated in Table 1, the inference speed of CornerNet with Cornerformer is 7.4 FPS, which is 3.3 FPS faster than the vanilla CornerNet using corner pooling. Besides, CenterNet with Cornerformer as well as CentripetalNet with Cornerformer is also more efficient than its baseline. It's worth noting that even if equipped with two CTEs, the inference speed is still faster than those corner pooling counterparts. The major speedup is brought by replacing corner pooling (a serial operation that requires a `for` loop) with CTE that is computed in parallel.

5.3 Pose Estimation Results

A Cornerformer is consisted of CTE and A^3-NMS, where CTE is used to enhance the corner features and improve the pull-push grouping as mentioned in Section 4.2. CTE is made up of multiple Transformer blocks and when stacking multiple blocks, a CTE is able to capture global contexts. Besides, CTE is

Fig. 5. Effectiveness of CTE in human pose estimation. Here we apply an extreme case – copy-and-paste the same object. HigherHRNet with CTE (right column) can distinguish local responses with similar visual appearances in different locations, making the pull-push grouping more reasonable, while the original HigherHRNet failed.

Table 2. Performance comparison of bottom-to-up human keypoint estimators on COCO test-dev. The CTE brings an improvement of 0.8% in terms of AP upon the HigherHRNet baseline.

Method	AP	AP_{50}	AP_{75}	AP_M	AP_L
OpenPose [4]	61.8	84.9	67.5	57.1	68.2
Hourglass [28]	65.5	86.8	72.3	60.6	72.6
SPM [29]	66.9	88.5	72.9	62.6	73.1
PersonLab [30]	68.7	89.0	75.4	64.1	75.5
HigherHRNet [6]	70.5	89.3	77.2	66.6	75.8
HigherHRNet w/ CTE	71.3	90.1	78.0	67.2	76.9

designed to be position-sensitive and pull-push-enhanced, so that it can better distinguish between similar appearances from different locations. Accordingly, we apply CTE to bottom-to-up human keypoints estimator. To test this conjecture, we take HigherHRNet [6] as the baseline. As shown in Table 2 and 3, HigherHRNet with Cornerformer gains an improvement of 0.8% and 1.3% on COCO and Crowdpose, respectively. Such improvements show that Cornerformer is effective not only for the corner-based detectors, but also for human pose estimation tasks. As shown in Figure 5, we copy-and-paste an image to visualize the effectiveness of Cornerformer on improving pull-push grouping. We can see that HigherHRNet cannot distinguish objects that have the same characteristics in different locations. CTE can overcome such problems, making the pull-push grouping more reasonable, further proving the position-sensitive CTE can make more accurate keypoint estimation.

5.4 Ablation Study

In this section, we conduct ablation analyses on COCO val2017 mainly for object detection and partially for human pose estimation. We mainly utilize CornerNet [20] as the baseline and use a single CTE for Cornerformer.

Table 3. Comparison on Crowdpose test dataset. Superscripts E, M, and H of AP stand for easy, medium and hard. The CTE brings an improvement of 1.3% in terms of AP upon the HigherHRNet.

Method	AP	AP_{50}	AP_{75}	AP_E	AP_M	AP_H
HigherHRNet [6]	67.6	87.4	72.6	75.8	68.1	58.9
HigherHRNet [6] w/ CTE	68.9	88.9	73.5	77.2	69.6	60.3

Table 4. Effectiveness of Cornerformer. We compare the performance of CTE and A^3-NMS with the original corner pooling and Soft-NMS on COCO val split and Citypersons to validate the effectiveness of the proposed Cornerformer. CP represents corner pooling. AP_c means the AP gained on Citypersons.

CTE	CP	A^3-NMS	Soft-NMS	AP	AP_{50}	AP_{75}	AP_c
×	×	×	✓	36.9	52.2	38.9	-
×	✓	×	✓	39.1	54.4	40.1	29.1
✓	×	×	✓	40.8	56.5	43.6	40.3
✓	×	✓	×	41.3	57.2	44.1	45.4

Effectiveness of the Cornerformer. In this part, we compare CTE and A^3-NMS with corner pooling (presented as CP in Table 4) and Soft-NMS, respectively, to validate the effectiveness of components in Cornerformer. As shown in Table 4, CTE achieves a 1.7% improvement compared to corner pooling. Compared with corner pooling, CTE can distinguish different corner positions with similar boundary visual appearances, while corner pooling fails. When we utilize A^3-NMS to replace Soft-NMS, a consistent improvement of 0.5% on AP is obtained on COCO and 5.1% AP gains on Citypersons, which validates the effectiveness of A^3-NMS on preventing overlapped object boxes from being falsely decayed. Compared with the baseline (36.9% on COCO and 29.1% on Citypersons), the Cornerformer (CTE + A^3-NMS) counterpart improves detection performance by 4.4% and 16.3% respectively. Besides, for Citypersons, compared with commonly used benchmarks, *e.g.*, Faster-RCNN (25.0% on AP) and RetinaNet (27.9% on AP), CornerNet equipped with Cornerformer can obtain a large improvement (45.4% on AP), further demonstrating the effectiveness of the proposed Cornerformer firmly.

Positional encoding in CTE. We embed positional encoding in self-attention module of CTE to help a model better distinguish corners with similar boundary features in the same row or column. Besides, to help the model "pull" or "push" corners better, we embed an additional positional encoding of full image in the output of CTE , as shown in Figure 2. We conduct experiments to test if these designs are resultful. As shown in Table 5, PCCA (with a positional encoding upon CCA) is 0.8% higher than CCA. Further embedding an additional positional encoding (represented as PE) in the output of CTE brings another

Table 5. Effectiveness of positional encoding. CCA is the original Criss-Cross attention [16]. PCCA is the proposed position-aware CCA. PE represents the positional encoding embedded in the output of Corner Transformer Encoder. All results are obtained with COCO val split.

CCA	PCCA	PE	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
✓	×	×	39.3	54.9	40.0	19.1	40.9	52.7
×	✓	×	40.1	55.6	42.9	19.4	41.5	53.0
×	✓	✓	40.8	56.5	43.6	19.6	42.1	53.4

0.7% performance gain. The above results prove the significance of positional encoding for corner predicting and grouping.

Table 6. Effects on number of CTE blocks. We verify that how many blocks of CTE are reasonable for different tasks on COCO val split.

Number of CTE Blocks	0	1	2	3	4
CornerNet [20] w/ CTE (AP)	36.9	38.7	39.9	40.8	40.4
HigherHRNet [6] w/ CTE (AP)	67.1	67.5	68.2	67.7	-

Number of CTE blocks. To validate the influence of the number of CTE blocks, we conduct experiments to test performance with different number of CTE blocks. As shown in Table 6, three CTE blocks is the best for object detection [20]. The reason that 4 CTE blocks brings slight performance drop compared to 3 may lie in insufficient training duration or data samples.

Besides, we further observe the performance on human pose estimation with respect to the number of CTE blocks. As shown in Table 6, two is the best choice for this task [6]. Three blocks CTE causes an AP drop, mainliy because an odd number of blocks leads to a focus on boundaries, which is ineffective for tasks with keypoints inside objects.

6 Conclusion

In this paper, we propose Cornerformer to address potential problems of corner-based detectors in instance construction. In Cornerformer, we design a 2D Corner Transformer Encoder to optimize corner estimation and pull-push grouping. Besides, upon the pull-push embeddings, we present an Attenuation-Auto-Adjusted NMS to further break limits of the heuristic NMS. Our research reveals that much room is left in constructing objects from mid-level visual cues. We hope that the simple and efficient design of Cornerformer will attract more attention to instance construction in corner-based detectors.

References

1. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)
2. Bodla, N., Singh, B., Chellappa, R., Davis, L.S.: Soft-nms–improving object detection with one line of code. In: Proceedings of the IEEE international conference on computer vision. pp. 5561–5569 (2017)
3. Cai, Z., Vasconcelos, N.: Cascade r-cnn: Delving into high quality object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6154–6162 (2018)
4. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7291–7299 (2017)
5. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision. pp. 213–229. Springer (2020)
6. Cheng, B., Xiao, B., Wang, J., Shi, H., Huang, T.S., Zhang, L.: Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5386–5395 (2020)
7. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3213–3223 (2016)
8. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 764–773 (2017)
9. Dong, Z., Li, G., Liao, Y., Wang, F., Ren, P., Qian, C.: Centripetalnet: Pursuing high-quality keypoint pairs for object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10519–10528 (2020)
10. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
11. Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., Tian, Q.: Centernet: Keypoint triplets for object detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 6569–6578 (2019)
12. Duan, K., Xie, L., Qi, H., Bai, S., Huang, Q., Tian, Q.: Corner proposal network for anchor-free, two-stage object detection. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16. pp. 399–416. Springer (2020)
13. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448 (2015)
14. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
16. Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., Liu, W.: Ccnet: Criss-cross attention for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 603–612 (2019)

17. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
18. Kong, T., Sun, F., Liu, H., Jiang, Y., Li, L., Shi, J.: Foveabox: Beyound anchor-based object detection. IEEE Transactions on Image Processing **29**, 7389–7398 (2020)
19. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
20. Law, H., Deng, J.: Cornernet: Detecting objects as paired keypoints. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 734–750 (2018)
21. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
22. Li, J., Wang, C., Zhu, H., Mao, Y., Fang, H.S., Lu, C.: Crowdpose: Efficient crowded scenes pose estimation and a new benchmark. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10863–10872 (2019)
23. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
24. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
25. Liu, F., Wei, H., Zhao, W., Li, G., Peng, J., Li, Z.: Wb-detr: Transformer-based detector without backbone. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2979–2987 (2021)
26. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European conference on computer vision. pp. 21–37. Springer (2016)
27. Lu, X., Li, B., Yue, Y., Li, Q., Yan, J.: Grid r-cnn. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7363–7372 (2019)
28. Newell, A., Huang, Z., Deng, J.: Associative embedding: End-to-end learning for joint detection and grouping. In: Advances in neural information processing systems. pp. 2277–2287 (2017)
29. Nie, X., Feng, J., Zhang, J., Yan, S.: Single-stage multi-person pose machines. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6951–6960 (2019)
30. Papandreou, G., Zhu, T., Chen, L.C., Gidaris, S., Tompson, J., Murphy, K.: Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 269–286 (2018)
31. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7263–7271 (2017)
32. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
33. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)
34. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229 (2013)

35. Sun, P., Zhang, R., Jiang, Y., Kong, T., Xu, C., Zhan, W., Tomizuka, M., Li, L., Yuan, Z., Wang, C., et al.: Sparse r-cnn: End-to-end object detection with learnable proposals. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14454–14463 (2021)
36. Tan, M., Pang, R., Le, Q.V.: Efficientdet: Scalable and efficient object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10781–10790 (2020)
37. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 9627–9636 (2019)
38. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. arXiv preprint arXiv:1706.03762 (2017)
39. Yang, Z., Liu, S., Hu, H., Wang, L., Lin, S.: Reppoints: Point set representation for object detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 9657–9666 (2019)
40. Zhang, S., Chi, C., Yao, Y., Lei, Z., Li, S.Z.: Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9759–9768 (2020)
41. Zhang, X., Wan, F., Liu, C., Ji, R., Ye, Q.: Freeanchor: Learning to match anchors for visual object detection. arXiv preprint arXiv:1909.02466 (2019)
42. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. arXiv preprint arXiv:1904.07850 (2019)
43. Zhou, X., Zhuo, J., Krahenbuhl, P.: Bottom-up object detection by grouping extreme and center points. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 850–859 (2019)
44. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)