Benchmarking Object Detectors with COCO: A New Path Forward

Shweta Singh[1⋆] Aayan Yadav[1⋆] Jitesh Jain[2]
Humphrey Shi[2] Justin Johnson[3] Karan Desai[3]
[1]IIT Roorkee [2]Georgia Tech [3]University of Michigan

Abstract. The Common Objects in Context (COCO) dataset has been instrumental in benchmarking object detectors over the past decade. Like every dataset, COCO contains subtle errors and imperfections stemming from its annotation procedure. With the advent of high-performing models, we ask whether these errors of COCO are hindering its utility in reliably benchmarking further progress. In search for an answer, we inspect thousands of masks from COCO (2017 version) and uncover different types of errors such as imprecise mask boundaries, non-exhaustively annotated instances, and mislabeled masks. Due to the prevalence of COCO, we choose to correct these errors to maintain continuity with prior research. We develop **COCO-ReM** (**Re**fined **M**asks), a cleaner set of annotations with visibly better mask quality than COCO-2017. We evaluate fifty object detectors and find that models that predict visually sharper masks score higher on COCO-ReM, affirming that they were being incorrectly penalized due to errors in COCO-2017. Moreover, our models trained using COCO-ReM converge faster and score higher than their larger variants trained using COCO-2017, highlighting the importance of data quality in improving object detectors. With these findings, we advocate using COCO-ReM for future object detection research. Our dataset is available at https://cocorem.xyz

Keywords: object detection, segmentation, datasets and benchmarks

1 Introduction

Rigorous benchmarking of computer vision models relies on task-specific datasets annotated by humans [9, 10, 22, 25]. Understandably, human-annotated data is prone to errors and imperfections due to factors such as ambiguous instructions to annotators, differences in perspective of multiple annotators, *etc.* Errors in ground-truth training data provide a noisy learning signal for models, resulting in sub-par capabilities. Even more problematic is the presence of errors in ground-truth evaluation data, which can cause a misalignment between evaluation metrics and human judgment when comparing model performances. In this work, we comprehensively study the extent of this issue for the canonical computer vision task of object detection and segmentation.

⋆ Denotes equal contribution. Correspondence to: Karan Desai (kdexd@umich.edu)

Fig. 1: COCO with Refined Masks (COCO-ReM). Top: COCO-ReM improves upon the quality of COCO-2017 masks by handling occlusions consistently (*dining table*) and annotating instances exhaustively (*banana*). **Bottom:** Random examples from COCO-ReM showing the mask quality (single category per image, for clarity).

The Common Objects in Context (COCO) dataset [25] has been the standard benchmark for object detection and segmentation since its inception in 2014. COCO fostered the empirical study of modern object detectors by providing a large set of 123K images annotated with 896K instance masks. The Mask AP (average precision) of COCO detectors has improved dramatically, with ≈70% *relative gain* between the pioneering Mask R-CNN [16, 17] from 2017, and recent models published in 2022–23 such as ViTDet [3, 23] and other Transformer-based models [4, 11, 33] like Mask2Former [8] and OneFormer [19].

Despite its popularity, COCO is not without limitations. COCO masks are known to have *coarse* boundaries [1, 14]. We manually inspect thousands of masks in COCO (2017 version) and observe other imperfections. As shown in Figure 1 (top), COCO-2017 masks do not handle holes and occlusions properly (*dining table*), and sometimes have non-exhaustively annotated instances (*banana*). We elaborate on these flaws in Section 2.

From an evaluation perspective, imperfections in *ground-truth* data make the metrics less reliable. COCO-2017 AP would wrongfully penalize models that predict more precise masks than the imperfect *ground-truth* masks. Moreover, models trained with such imperfect masks may exploit unwanted biases from the training data, *e.g.,* learning to *never* predict masks with holes. Recently, Kirillov et al. observed that human evaluators consistently rated the masks predicted by their Segment Anything Model (SAM) to be of higher quality than a ViTDet model [23] trained on COCO, whereas the latter scored higher on COCO AP.

These observations raise an important question – *can COCO reliably benchmark the future progress in object detection research?* COCO is the *de-facto* benchmark for object detection – countless research papers report COCO results every year; libraries like Torchvision [29] and Detectron2 [36] readily support object detector development with COCO. Hence, to maintain continuity with prior research and tap into this rich open-source ecosystem, we choose to rectify the errors of COCO and reinforce its utility for future research.

Contributions. We develop **COCO-ReM** (**Re**fined **M**asks), a cleaner set of high-quality instance annotations for COCO images. The improved quality over COCO-2017 masks is *visually palpable*, as seen in Figure 1.

Our annotation pipeline (Section 3) utilizes SAM [21] to refine masks, and uses the LVIS dataset [14] and an ensemble of LVIS-trained models to exhaustively annotate *grouped* or *missing* instances. *We (the authors) manually verify the entire validation set to provide a strong quality guarantee for evaluation.*

In Section 4, we use COCO-ReM for benchmarking object detectors. We evaluate fifty object detectors and observe that their COCO-ReM AP produces a different model ranking than COCO-2017 AP. Surprisingly, we observe that *query-based* models (Mask2Former and OneFormer) score much higher on COCO-ReM than *region-based* models (ViTDet). Query-based models predict visually sharper masks – AP trends on COCO-ReM accurately reflect this, unlike COCO-2017. We observe that models trained using COCO-ReM converge faster and perform better than those trained using COCO-2017, highlighting the importance of mask quality in improving object detectors.

2 Background: Revisiting COCO Masks

The annotation pipeline of a dataset determines the quality of annotations obtained. To design an effective annotation pipeline for COCO-ReM, we review the COCO annotation pipeline by Lin et al. [25] and understand its shortcomings.

COCO annotation pipeline. Lin et al. [25] employed a three-stage pipeline to collect instance mask annotations for COCO images:

1. **Category labeling:** Annotators assigned one or more labels to images depending on the visible content, choosing from a fixed set of categories.
2. **Instance spotting:** Annotators locate up to ten instances per label (from the previous stage) in the corresponding image, by providing clicks (points).
3. **Instance segmentation:** Finally, annotators clicked around the boundaries of each located instance (from the previous stage), rendering a *polygon* mask.

Lin et al. [25] also took numerous measures to improve annotation quality. The first two stages – *category labeling* and *instance spotting* – were performed by eight annotators per image to maximize recall. The *instance segmentation* stage required annotators to complete a training task before annotating actual instances in the dataset. Next, a different set of crowd workers verified the correctness and quality of annotated masks. Finally, the annotators used a *paint-brush* tool to mark remaining objects as *crowd regions*. These extensive measures avoided many scenarios that could have led to low-quality masks.

(a) Holes (b) Inconsistency in occlusion

(c) Missing annotations and labeling multiple instances as one

Fig. 2: Imperfections in COCO-2017 masks. All masks in general have coarse boundaries. Moreover, they lack holes when required (*scissors*) and do not handle occlusions consistently (*TV* with or without figurines). Annotations are sometimes non-exhaustive, either masks are missing (*apples* and *donuts*), or multiple instances are grouped into a single mask (*banana* and *dog*).

Imperfections in COCO-2017 masks. To further understand the flaws in COCO annotations, we visualized thousands of COCO-2017 masks and inspected their quality and label correctness. Through this exercise, we observed recurring types of errors in annotations, as shown in Figure 2.

1. **COCO-2017 masks have coarse boundaries.** This is a natural consequence of collecting masks as *polygon contours*, which only coarsely approximate the pixel-precise mask boundaries through piece-wise straight edges connecting points (clicks) supplied by annotators. This shortcoming is well-known and identified by several follow-up works [1, 14, 21].
2. **COCO-2017 masks lack holes when required.** The polygon-drawing interface of COCO did not allow annotators to *erase* the interior regions of polygons. Consequently, certain objects that should naturally contain holes, lack them. Figure 2a shows two examples – the handles of *scissors* and *cup*.
3. **COCO-2017 masks do not handle occlusions consistently.** Masks of occluded objects are either drawn *over* the other occluding objects (*amodal* masks) or *around* them (*modal* masks). For example, observe the *TV* masks in Figure 2b. While both, *amodal* and *modal* masks are valid, either one should be consistent in the dataset according to the task definition. We believe this inconsistency is a consequence of inadequate task instructions to annotators as well as the lack of holes in masks.
4. **COCO-2017 masks are often non-exhaustive.** This issue occurs in two modes, as shown in Figure 2c – either few instance masks of a category are missing[1] (*apple*, *donut*), or multiple instances are grouped in a single mask (*banana*, *dog*). This is a recurring issue throughout COCO – we believe that certain annotators may not have deemed masks with grouped instances to have a *wrong object contour*, due to oversight.

[1] These images have all their missing masks annotated as *non-crowd* regions.

5. **COCO-2017 contains few near-duplicate masks.** We found 410 pairs of masks that overlap with intersection-over-union (IoU) greater than 0.8 in the COCO-2017 validation set (≈2.3% of all instances). These pairs often segment the same underlying object with different labels. Examples include armchairs (*chair* and *couch*), calves (*cow* and *sheep*), and duffel bags (*suitcase* and *backpack*). We attribute this issue to the lack of a cross-category deduplication step in the COCO annotation pipeline.

All these imperfections hurt the reliability of the benchmarking process. They provide noisy supervision during training and wrongfully penalize correct model predictions during evaluation. Having characterized these imperfections, we make targeted design choices in our COCO-ReM annotation pipeline to rectify them.

3 COCO-ReM: COCO with Refined Masks

In this section, we outline our annotation process for **COCO-ReM**. Our goal is to rectify the imperfections in COCO-2017 annotations as discussed in Section 2 to maintain continuity with prior research. We hope that higher quality *ground-truth* masks would improve the reliability of the benchmarking process.

3.1 Annotation Pipeline

We develop a *semi-automatic* pipeline – first, we refine the masks using off-the-shelf models and LVIS dataset [14], followed by manual verification of the updated masks. We refine both, the training and validation set of COCO-2017, comprising ≈860K and ≈36K instances respectively. We thoroughly verify the validation set to ensure *consistency* and *pixel-precise mask quality* for model evaluation. As the training set is much larger, it is only processed through the automatic steps of our pipeline. Our pipeline comprises three stages (Figure 3).

Stage 1. Mask boundary refinement. In this stage, we aim to rectify the coarse boundary quality of COCO-2017 masks. A straightforward solution to achieve this is to re-annotate all masks through a web interface with *paintbrush* and *eraser* tools instead of drawing polygon contours. Moreover, this solution naturally allows including holes in masks where necessary. While accurate, this solution would sharply increase the mask annotation time and cost.

Along a different trend, prior works have explored *model-assisted interactive segmentation* as a feasible solution for collecting mask annotations with smooth boundaries [1, 14]. The key idea is to let an object detector predict a coarse initial mask and have the annotators refine it by prompting corrective mouse clicks. Inspired by the accuracy and practical feasibility of this approach, we use the Segment Anything Model (SAM [21]) to refine the COCO-2017 masks.

– **Refining masks using SAM.** SAM is a promptable segmentation model that predicts high-quality masks conditioned on user-provided point and box prompts. This promptable functionality makes SAM suitable for our use case. We prompted SAM using various combinations of prompts sampled using COCO-2017 masks and obtained its predictions as the refined masks. Refer to the appendix for details about this prompting methodology.

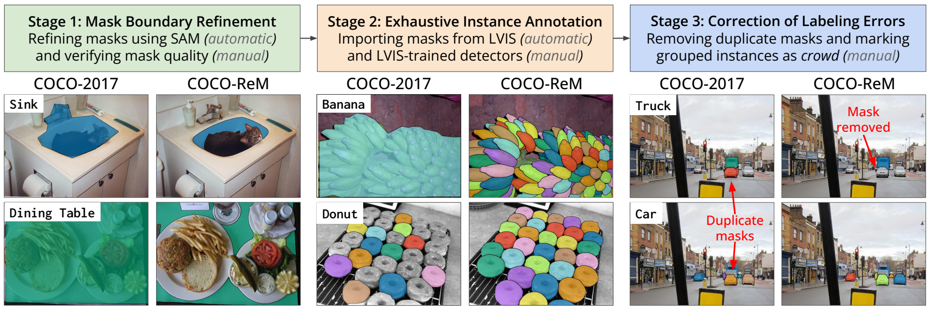

Fig. 3: COCO-ReM annotation pipeline. We develop a *semi-automatic* annotation pipeline comprising three stages to rectify the imperfections in COCO-2017 masks.

- **Manual verification of mask quality.** We visualized and manually inspected every mask created in the previous process. Through this inspection, we identified nearly 900 masks that have low quality or do not accurately cover the underlying object. These masks represent only 2.4% of 36K instances in the COCO-2017 validation set, greatly reducing our manual effort and showing the efficacy of our automated process. To correct these masks, we deployed a web browser demo of SAM and provided point prompts iteratively until obtaining a precise and accurate mask.

Through this stage, we obtain masks with more precise boundaries and better handling of occlusions, as illustrated in Figure 3 (left).

Stage 2. Exhaustive instance annotation. This stage addresses non-exhaustively annotated instances in COCO-2017 (Figure 2c). We note that manually re-annotating every image is the most reliable solution to ensure exhaustive instance coverage. However, the manual effort would not only require extensive instructions and verification but would also be highly cost-ineffective and may lead to redundant annotations. To mitigate these issues, we automatically import instances from the LVIS [14] dataset as a strong starting point.

- **Importing instances from LVIS.** The LVIS dataset comprises 1.2M high-quality instance annotations across 1203 categories for images in COCO, including all 80 COCO categories. Gupta et al. [14] took firm measures to ensure exhaustive annotations. Every image is labeled with a set of *positive* LVIS categories whose instances are exhaustively labeled, and a set of *negative* LVIS categories that are verifiably absent. The presence (or absence) of the remaining LVIS categories is *unverified* for the image.
  For the 80 COCO categories, LVIS contains ≈20K instances for images in the COCO-2017 validation set. To import these instances, we checked whether LVIS contained more instances of an *(image, category)* pair than COCO-2017. If so, we replaced all COCO-2017 instances of this pair with those from LVIS. Nearly 9% of *(image, category)* pairs in LVIS are flagged to have non-exhaustive instance annotations [14] – we ignored these while importing.
- **Discovering instances using LVIS-trained models.** Instances of a given *(image, category)* pair of COCO-2017 may be missing in LVIS if the category

		Instance Source			
Dataset	Split	COCO-2017	LVIS	LVIS models	Total
COCO-2017	`val`	36,781	-	-	36,781
COCO-ReM	`val`	33,498	6,135	1056	40,689
COCO-2017	`train`	860,001	-	-	860,001
COCO-ReM	`train`	738,353	354,674	-	1,093,027

Table 1: Instance statistics in COCO-ReM. We list the exact number of annotations obtained from different sources for both `train` and `val` splits of COCO-ReM. All masks undergo refinement through SAM before being included in COCO-ReM.

is *unverified* for the corresponding image. Instances of this pair may be non-exhaustive in COCO-2017, and hence remain unresolved after the previous automatic step. To resolve this, we use publicly available object detectors trained using LVIS. On the validation set, we ensemble mask predictions using the official checkpoints of three best-performing ViTDet models [23]. As model predictions are not as accurate as human-annotated data, we manually inspect ≈2500 masks predicted with high confidence and select 1056 masks. We refine these masks (Stage 1) and add them to COCO-ReM.

This stage augments COCO-ReM with additional instances from external sources and includes more exhaustive annotations, see example in Figure 3 (middle).

Stage 3. Correction of labeling errors. This is the final stage of our pipeline, performed manually, only for the validation set. Recall from Section 2 that we found 410 near-duplicate mask pairs in the COCO-2017 validation set. We manually inspected these and retained one mask per pair with the most accurate label, such as *chair* for armchairs and *car* for SUVs (Figure 3 (right)). We also observed that ≈100 grouped instances were not resolved through Stage 2. We annotated these masks as *crowd* to accurately reflect their characteristics.

3.2 Mask Characteristics

Instance statistics. Our annotation pipeline imports instances from external sources – LVIS dataset and LVIS-trained models – to annotate instances exhaustively. As a result, every instance in COCO-ReM traces to one of these sources or COCO-2017. As shown in Table 1, COCO-ReM validation set extends COCO-2017 by including ≈6K masks from LVIS [14] and ≈1K masks from LVIS-trained models [3, 23]. Since these masks replace a subset of COCO-2017 masks (see Stage 2), we retain ≈33.5K masks from COCO-2017 (out of 36K masks). Overall, COCO-ReM validation set has ≈4K more instances than COCO-2017.

On the other hand, the training set of COCO-ReM comprises a total of 1.09M high-quality masks, as compared to 860K masks in the COCO-2017 training set. The difference in instance counts between COCO-ReM and COCO-2017 sharply highlights the extent of non-exhaustively annotated instances in COCO-2017.

Mask IoU. Figure 4 (left) shows the distribution of IoU between the (refined) masks of COCO-ReM (validation set) and their corresponding source masks in

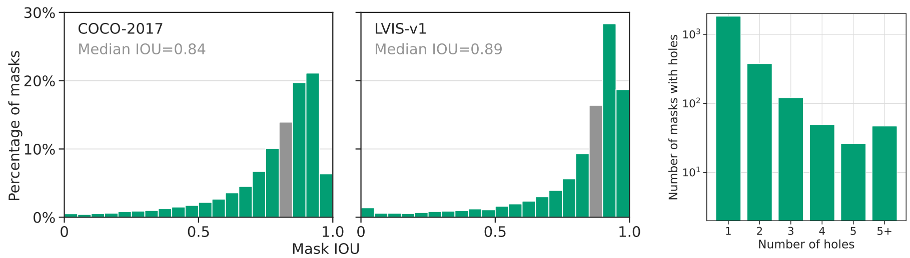

Fig. 4: Left: Mask IoU distribution. Distribution of IoU between masks in COCO-ReM validation set and their source dataset. **Right: Masks with holes.** Nearly 2000 masks in COCO-ReM validation set have holes, that were missing in COCO-2017.

COCO-2017 and LVIS datasets. The median IoU is slightly less than one, indicating that most masks have a proper coarse shape, but lack precise boundaries. This median estimate matches the findings of Gupta et al. [14], albeit observed from a smaller sample of 100 masks collected from human annotators.

Holes in masks. Figure 4 (right) shows that ≈2000 masks in COCO-ReM validation set have at least one hole. By design, COCO-2017 lacks these holes. See the appendix for qualitative examples and additional analysis per category.

4 Experiments

In our experiments, we aim to demonstrate the utility of COCO-ReM for reliable benchmarking of object detection research. Recall Section 1, we hypothesize that the imperfections in COCO-2017 may cause two issues: (1) During evaluation, models may be undesirably penalized despite predicting correct masks, yielding counter-intuitive modeling observations, and (2) During training, models receive noisy supervision and/or may learn unwanted biases (*e.g.,* never predict masks with holes), resulting in sub-par capabilities. We claim that high-quality masks of COCO-ReM can alleviate these issues. To this end, we use COCO-ReM to evaluate and train object detectors, and compare trends with COCO-2017.

4.1 Evaluation using COCO-ReM

In this section, we evaluate *fifty* publicly available COCO object detectors using the COCO-ReM validation set. We focus on the validation set as the test set annotations of COCO-2017 are not publicly available. We cover a wide range of models from existing literature. All models can be categorized as either *region-based* or *query-based*:

- **Region-based models**, following Mask R-CNN [17]. These models first predict labeled boxes, followed by binary masks inside those boxes. We also include *cascade* models, following Cascade R-CNN [3]. These models extend the region-based design by iteratively refining their predicted bounding boxes before making the final mask prediction.

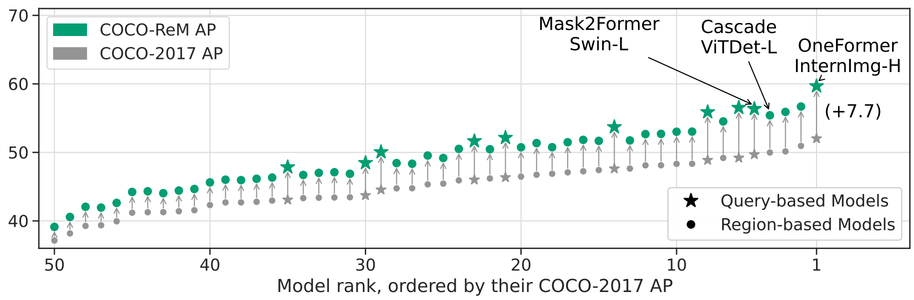

Fig. 5: Evaluating fifty object detectors using COCO-ReM. All models score higher on COCO-ReM than COCO-2017. Query-based models (★) score much higher than region-based models on COCO-ReM, yielding opposite trends than COCO-2017.

– **Query-based models**, inspired by the Detection Transformer design [4]. Instead of predicting masks within image regions, these models start with a set of image-independent query vectors, then decode labeled masks through a Transformer [33] by iteratively attending to image embeddings. Among query-based models, we consider Mask2Former [8] and OneFormer [19].

We ensure diverse image backbones in our model collection. The architecture and capacity of the backbone significantly impact object detector performance. Thus, we include convolutional backbones(ResNet [16], RegNet [31], ConvNeXt [27], InternImage [35]), hierarchical vision transformers (Swin [26], MViTv2 [24], DiNAT [15]), and plain vision transformers [11]. We include backbones of different sizes for a single model, *e.g.,* for Cascade ViTDet [23], we include ViT-B (141M parameters), ViT-L (361M parameters), ViT-H (692M parameters).

We believe that our model collection is sufficiently diverse to draw meaningful conclusions from comparing trends between COCO-ReM and COCO-2017

Testing protocol. We compute the mask Average Precision (AP) of all models using the official COCO evaluation API in Detectron2 [36]. For simplicity, we adopt a common testing protocol for all models – we resize the longest edge of input images to 1024 pixels (without changing the aspect ratio), and limit the number of detections per image to 100.

Result 1. All models score higher on COCO-ReM. In Figure 5, we plot the mask AP trends for all fifty object detectors sorted by their COCO-2017 AP. All fifty models achieve higher AP on COCO-ReM despite being trained using only COCO-2017 masks. This result validates our hypothesis that the imperfections of COCO-2017 may have wrongfully penalized all these models.

The difference between COCO-ReM AP and COCO-2017 AP is larger with recent high-performing models – the current state-of-the-art model (OneFormer InternImage-H) scores $+\mathbf{7.7}$ points higher AP on COCO-ReM than COCO-2017. This increasing gap in AP suggests that the imperfections in ground-truth have worsened the issue of over-penalization and made COCO-2017 AP less adept at fine-grained assessment and ranking of recent models.

Result 2. Query-based detectors score much higher on COCO-ReM.

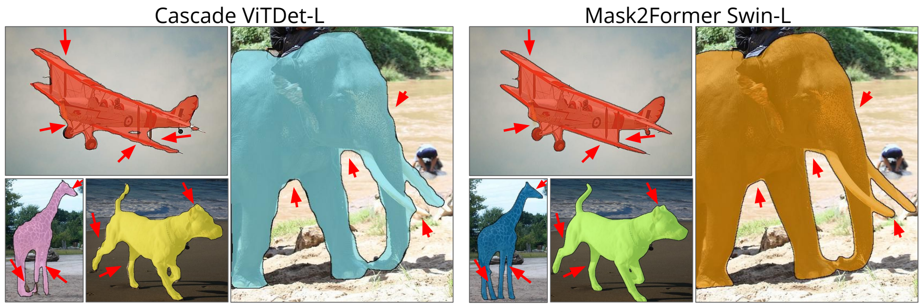

Fig. 6: Mask predictions from models with opposite rank order on COCO-ReM and COCO-2017. Qualitative examples of ViTDet (*region-based* model) and Mask2Former (*query-based* model) show that the latter predicts visually sharper masks. COCO-ReM AP correctly ranks Mask2Former higher, unlike COCO-2017 AP.

Interestingly, we observe that the trend in AP values calculated using COCO-ReM does not align perfectly with COCO-2017. We observe frequent *flips* in ranks of several model pairs on COCO-ReM. While the model rankings do not transfer between benchmarks perfectly, we observe a consistent pattern in model pairs with flipped rankings. Notice the star markers (★) in Figure 5 – the flips are attributed to significantly higher scores of query-based models (Mask2Former [8] and OneFormer [19]) than region-based models on COCO-ReM.

To further understand the cause of this performance trend, we observe the predicted masks of two models having opposite rank order according to COCO-ReM and COCO-2017 AP: Cascade ViTDet-L [23] and Mask2Former Swin-L [8]. The latter is a query-based model. Figure 6 shows some predicted masks by both of these models. Through a cursory glance, we can observe that Mask2Former predicts visually sharper masks whereas masks of ViTDet have coarse boundaries. This human judgment aligns better with COCO-ReM AP, which correctly rates Mask2Former higher. On the other hand, COCO-2017 AP rates ViTDet higher, yielding a counter-intuitive modeling observation.

Guided by these trends, we reaffirm our claim that benchmarking newer models using the COCO-2017 may be misleading from a research perspective, and the community should adopt our COCO-ReM annotations moving forward. We include numerical results of Figure 5 in the appendix.

4.2 Understanding the Difference in AP

Recall Section 4.1 (Result 1) – all models score higher average precision (AP) on COCO-ReM, as compared to COCO-2017. *What explains the difference in AP?* To understand the reasons underlying this difference, we observe the components of COCO AP for a few high-performing models from Section 4.1. For this analysis, we consider two Cascade ViTDet models [23] using ViT-L/H backbones [11], and two Mask2Former models [8] using Swin-B/L backbones [26].

AP per IoU threshold. The calculation of COCO AP involves an average of per-category AP scores of 80 COCO categories, each computed at ten different

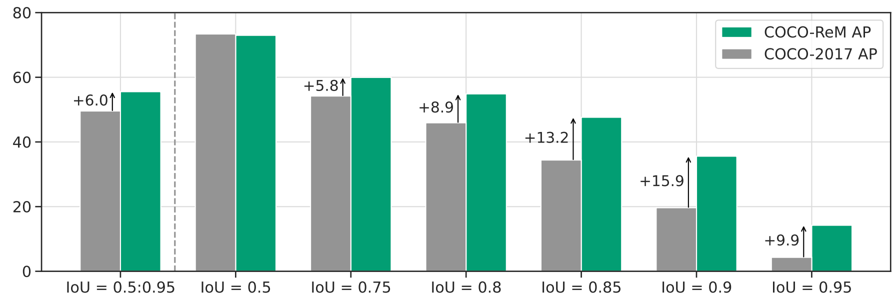

Fig. 7: AP per IoU threshold. While models score similarly at IoU $= 0.5$ on COCO-ReM and COCO-2017, they score much higher on COCO-ReM at higher IoU thresholds (0.75 and beyond), contributing to the overall difference in AP. Due to the improved mask quality of COCO-ReM, the AP at high IoU thresholds becomes more sensitive and enables better assessment of model capabilities.

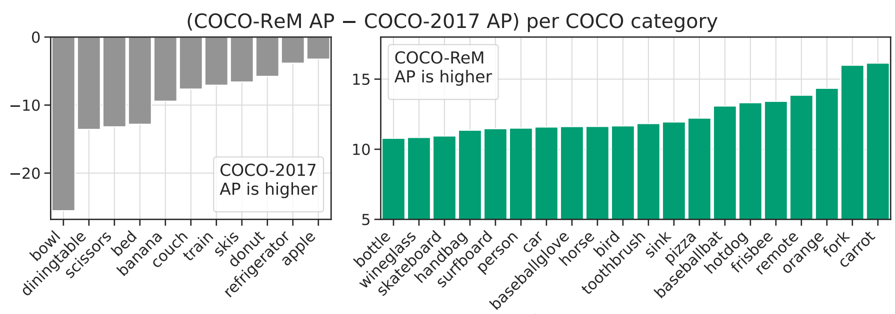

Fig. 8: AP per category. Models score higher AP on COCO-ReM for 69 out of 80 categories. The masks of the remaining 11 categories (left) in COCO-2017 do not handle occlusions consistently (*dining table*, *bed*), lack holes (*scissors*, *donut*), or have grouped instances (*banana*, *apple*). Models learn these biases from COCO-2017 training masks and score lower COCO-ReM AP.

IoU thresholds, $t \in \{0.5, 0.55, \dots, 0.9, 0.95\}$. In Figure 7, we show components of AP at different IoU thresholds, averaged for the four models we use. On average, models score similarly at IoU $= 0.5$ for COCO-ReM and COCO-2017. However, COCO-ReM AP is much larger for IoU $= 0.75$ and beyond. Hence, higher scores on COCO-ReM can be attributed to AP at higher IoU thresholds.

Intuitively, AP at IoU $= t$ rewards model predictions that overlap with a ground-truth mask of the same category with IoU $\geq t$. AP at IoU $= 0.5$ requires a very *coarse* overlap between the predicted and ground-truth masks, whereas AP at high IoU thresholds have stricter overlap requirements. Since IoU is symmetric, imprecise boundaries of ground-truth masks (COCO-2017) can wrongfully penalize valid predictions and lead to lower AP scores. COCO-ReM fixes this issue, as evident with higher AP at high IoU thresholds. This analysis hints that *precise* masks of COCO-ReM improve the sensitivity of AP and enable fine-grained assessment of high-performing object detectors.

AP per category. Next, we calculate AP for every COCO category by averaging across all IoU thresholds. Figure 8 shows the difference in COCO-ReM

COCO evaluation set ⟹	2017	ReM-S1	ReM-S2a	ReM (final)
AP	49.6	57.3	55.6	55.5
Number of instances	36,781	36,781	41,168	40,689

Table 2: COCO-ReM ablations. AP of four high-performing models (Cascade ViTDet-B/L and Mask2Former Swin-B/L) using intermediate versions from our COCO-ReM annotation pipeline. Trends suggest that the coarse boundary quality of COCO-2017 masks was a major source of noise in the benchmarking process.

AP and COCO-2017 AP per category, averaged across the same four models. Models score higher on COCO-ReM for 69 out of 80 categories, which is reflected in the higher overall COCO-ReM AP. Significant improvements are observed in categories with generally consistent appearance (*carrot* and *orange*), and fine details (*fork*). Intuitively speaking, the larger difference in per-category AP can be attributed to the better quality of ground-truth masks in COCO-ReM.

For 11 categories, models score lower on COCO-ReM than COCO-2017. These categories either have holes in their natural appearance or are often heavily occluded (*bowl*, *dining table*, *scissors*). Masks in COCO-ReM include holes and handle occlusions more consistently than in COCO-2017. Hence, lower per-category AP with COCO-ReM implies that the imperfections of COCO-2017 training masks impart these undesirable biases to trained models – upon checking the model predictions, we observed that they often include occluding objects for *dining table* and lack holes for *scissors*. Such predictions are not rewarded by COCO-ReM, hence incur a drop in the per-category AP.

4.3 Ablations

In this section, we conduct a simple ablation study to observe the contribution of different stages in our annotation pipeline. For the sake of simplicity, we select the same four models as chosen in Section 4.2. We evaluate these models using two versions of COCO-ReM validation set:

1. **ReM-S1:** This version is obtained after the Stage 1 (mask boundary refinement) in our annotation pipeline. It comprises the same number of instances as COCO-2017, each with an improved boundary quality.
2. **ReM-S2a:** This version is obtained after importing instances from the LVIS dataset – the *automatic* part of Stage 2 (exhaustive instance annotation). We exclude the *manual* part of Stage 2 from this ablation for simplicity.

Results are shown in Table 2. The large difference between COCO-ReM AP and COCO-2017 AP appears right after Stage 1, indicating that the coarse mask boundaries are a major source of noise that hurts the reliability of COCO-2017 AP. Subsequent stages incur a negligible drop in AP, likely due to larger of the validation set, as COCO AP is known to favor smaller evaluation sets [14].

4.4 Training with COCO-ReM

Our evaluation shows COCO-ReM as a useful benchmark for accurate modeling conclusions. In this section, we train strong baseline models using COCO-ReM.

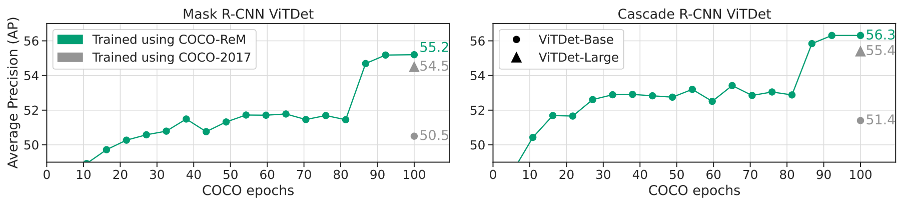

Fig. 9: Training with COCO-ReM. A ViTDet model trained using COCO-ReM training set converges faster and performs better than its counterpart trained using COCO-2017, and closely matches a 3× larger ViTDet-L trained using COCO-2017.

Implementation details. We train ViTDet models [23] using their official open-source implementation [36], leaving all hyperparameters unchanged. We train two models – Mask R-CNN ViTDet-B (111M parameters) and Cascade ViTDet-B (141M parameters). Li et al. [23] trained this model for 100 COCO epochs across 64× A100 GPUs (batch size of 64). To manage these costs, we implement gradient accumulation for every 8 iterations to emulate a batch size of 64 using our 8× A40 GPUs, resulting in an 11 day-long training schedule. We could only afford to train the *base* (ViT-B) models due to limited resources. The gorund truth annotations for test set is inaccessible so, we use the validation set solely for evaluation purposes for fair assessment of models.

Results. Figure 9 shows that our ViTDet-B models trained using COCO-ReM significantly outperform the same ViTDet-B models trained using COCO-2017. Higher quality of training data enables faster convergence. Moreover, our ViTDet-B models are competitive with 3× larger ViTDet-L models trained using COCO-2017, indicating better parameter efficiency. While numerous prior works improve object detectors through modeling innovations, our results highlight the critical role of data quality in improving object detectors.

5 Related Work

Object detection and image segmentation. Visual perception is fundamental for any vision system to understand the world around it. At the core of a visual perception system is the task of image segmentation that aims at recognizing the entities (objects) in a given scene and densely labeling every pixel constituting the corresponding object. Image segmentation can broadly be divided into three sub-tasks: semantic, instance, and panoptic segmentation. On the one hand, semantic segmentation [5, 6, 28] targets obtaining a single amorphous binary mask covering all pixels for the corresponding category. On the other hand, instance segmentation [3, 18] involves detecting pixels for only foreground “thing" regions with clear boundaries and differentiating among separate instances for a category. Panoptic segmentation [7, 20, 34] is a combination of the former two tasks targeted at predicting distinct binary masks for “thing" regions and a single amorphous mask for the “stuff" regions.

In this work, we particularly look at the task of instance segmentation and contribute high-quality mask annotations to benchmark further progress.

Image segmentation benchmarks. In the last decade, considerable advancements have been observed in the classical vision task of image segmentation. A part of this progress is owed to the introduction of numerous datasets for benchmarking image segmentation models. In one of the earliest efforts, Everingham et al. [12] introduced the PASCAL-VOC-2012 challenge to recognize 20 classes in about 10K images. More challenging datasets were introduced in the following years, with ADE20K [10] covering 150 classes. Lin et al. [25] introduced the concept of “stuff" and “thing" classes in their COCO dataset with a total of 171 classes: 80 “thing" and 91 “stuff". The Cityscapes [9], Mapillary-Vistas [30], and KITTI [13] datasets introduced image segmentation datasets focused on autonomous driving applications.

Gupta et al. [14] introduced the LVIS dataset with exhaustive annotations covering 1203 categories for COCO images. OpenImages [1] introduced 2.8M mask annotations covering 350 classes. Recently, the SA-1B dataset [21] introduced over 1.1B fine-grained, *class-agnostic* mask annotations for over 11M images. Despite various datasets, COCO has remained the de-facto benchmark for object detectors in instance segmentation. In this work, we examined the COCO instance annotations and proposed a new benchmark with superior mask quality.

Re-assesing datasets for benchmarking. Human-annotated datasets are always prone to errors and imperfections, as no human is perfect. Some prior works have shared our motivation and conducted in-depth re-assessment studies, the closest to ours are works that re-assess the ImageNet dataset [2, 32]. Concurrent to our work, Zimmermann et al. [37] also re-assess COCO annotations, however they opt for recollecting *polygon masks* from human annotators.

6 Conclusion

In this work, we revisit the COCO-2017 mask annotations and find major issues involving, but not limited to their boundary quality and exhaustiveness. These issues raise significant concerns about the dependability of COCO-2017 to benchmark recent and future object detectors accurately. Consequently, we introduce our **COCO-ReM** benchmark consisting of high-quality mask annotations obtained using an effective *semi-automatic* annotation pipeline.

On comparing the mask AP trends for existing object detectors on COCO-ReM and COCO-2017, we notice significant changes in model rankings, notably contradicting the conclusions drawn through COCO-2017. Moreover, training using COCO-ReM shows that paying attention to mask quality is crucial to advancing the capabilities of object detectors. We hope that COCO-ReM will aid the future research in object detection.

Acknowledgments

We thank Mohamed El Banani, Agrim Gupta, and Nilesh Kulkarni for helpful discussions and feedback. We thank Palak Jagtap for her help during manual verification of masks. We thank all the authors of the prior works for publicly releasing the pre-trained model checkpoints, which helped us maintain empirical comprehensiveness in our evaluation study. We thank Vision and Language Group, IIT Roorkee for being a constant support system throughout the process.

Bibliography

[1] Rodrigo Benenson, Stefan Popov, and Vittorio Ferrari. Large-scale interactive object segmentation with human annotators. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 4, 5, 14

[2] Lucas Beyer, Olivier J Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with imagenet? *arXiv preprint arXiv:2006.07159*, 2020. 14

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into high quality object detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 7, 8, 13

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. 2, 9

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015. 13

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. In *TPAMI*, 2017. 13

[7] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 13

[8] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 9, 10

[9] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 14

[10] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. Semantic understanding of scenes through the ade20k dataset. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 14

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer,

Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. 2, 9, 10

[12] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012. 14

[13] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 14

[14] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 3, 4, 5, 6, 7, 8, 12, 14

[15] Ali Hassani and Humphrey Shi. Dilated neighborhood attention transformer. *arXiv:2209.15001*, 2022. 9

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 9

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017. 2, 8

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017. 13

[19] Jitesh Jain, Jiachen Li, MangTik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. OneFormer: One Transformer to Rule Universal Image Segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 9, 10

[20] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 13

[21] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment Anything. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2023. 2, 3, 4, 5, 14

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2012. 1

[23] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2022. 2, 7, 9, 10, 13

[24] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale

vision transformers for classification and detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 9

[25] 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 *Proceedings of European Conference on Computer Vision (ECCV)*, 2014. 1, 2, 3, 14

[26] 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 *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2021. 9, 10

[27] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 9

[28] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 13

[29] TorchVision maintainers and contributors. TorchVision: PyTorch's Computer Vision library. https://github.com/pytorch/vision, 2016. 3

[30] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017. 14

[31] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 9

[32] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *Proceedings of the International Conference on Machine Learning (ICML)*, 2019. 14

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 2, 9

[34] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. MaX-DeepLab: End-to-end panoptic segmentation with mask transformers. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 13

[35] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 9

[36] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 3, 9, 13

[37] Eric Zimmermann, Justin Szeto, Jerome Pasquero, and Frederic Ratle. Benchmarking a benchmark: How reliable is ms-coco? *arXiv preprint arXiv:2311.02709*, 2023. 14