Appendix

In the supplementary materials, we provide additional information, as listed below.

- Sec. A provides the details of our label map definition and annotation rules.
- Sec. B presents additional annotation visualizations.
- Sec. C provides more benchmarking results on COCONut.

A. Label Map Details and Annotation Instruction

Our label map definition strictly follows COCO [35]. However, the COCO definitions of specific categories might be ambiguous. Specifically, we have identified several conflicts between the 'thing' and 'stuff' categories, often resulting in potential mask overlaps. To mitigate the issue, we have meticulously redefined specific categories, detailed in Tab. 8, Tab. 9, and Tab. 10. The definitions of categories not included in the tables remain consistent with the original COCO label map.

B. Additional Annotation Visualizations

We present additional annotation visualizations for COCONut dataset. Specifically, Fig. 10 and Fig. 11 demonstrate the COCONut annotations for images sourced from COCO unlabeled set images and Objects365 [53]. As shown in the figures, COCONut provides annotations comprising a large number of classes and masks. Notably, the inclusion of Objects365 images enriches COCONut annotations by introducing a wider variety of classes and masks compared to the COCO images. We also provide additional annotation comparison between COCO and COCONut. In particular, Fig. 12 compares the COCO and COCONut annotations, where the common errors of COCO (*e.g.*, inaccurate boundary, loose polygon, missing masks, and wrong semantics) are all corrected in COCONut annotations. Fig. 13 and Fig. 14 provide more annotation comparisons between COCO, COCONut, and our expert raters. Fig. 15 and Fig. 16 provide more visualizations of prediction bias introduced by training data.

C. Additional Experimental Results

In this section, we outline the training and evaluation protocols utilized to benchmark COCONut across multiple tasks and the corresponding results in Sec. C.1 and Sec. C.2, respectively.

C.1. Training and Evaluation Protocols

Training Protocol: COCONut undertakes benchmarking across various tasks, encompassing panoptic segmentation [29], instance segmentation [22], semantic segmentation [25], object detection [15], and open-vocabulary segmentation [13, 17]. The kMaX-DeepLab [60, 65], tailored for universal segmentation, serves as the primary framework for panoptic, instance, and semantic segmentation in our experiments. Object detection relies on the DETA framework [43], while open-vocabulary segmentation utilizes the FC-CLIP framework [66].

Throughout each task, we strictly adhere to the training hyper-parameters defined by the respective frameworks, utilizing ResNet50 [23], Swin-L [38], and ConvNeXt-L [39] as the backbones.

Evaluation Protocol: When evaluating each task, we follow official settings meticulously. For panoptic, instance, and semantic segmentation tasks, metrics such as panoptic quality (PQ) [29], AP^{mask} [35], and mean Intersection-over-Union (mIoU) [15] are reported. Bounding box detection performance is measured using the AP^{box} metric. In line with prior methodologies [13, 61], open-vocabulary segmentation results undertake zero-shot evaluation on other segmentation datasets [15, 42, 69].

C.2. COCONut Empowers Various Tasks

In this section, we show the results for task-specific models trained on COCONut datasets including panoptic segmentation, instance segmentation, semantic segmentation, object detection, semantic mask conditional image synthesis.

Panoptic Segmentation: In Tab. 11, we benchmark kMaX-DeepLab on the task of panoptic segmentation. The results are the same as Tab. 6 in the main paper, where a panoptic model is evaluated on all three segmentation metrics.

Instance Segmentation: We benchmark kMaX-DeepLab on the task of instance segmentation. Different from Tab. 6 in the main paper where the mask AP is evaluated using a panoptic segmentation model, we train a task-specific model on instance segmentation with instance masks only. Tab. 12 summarizes the results. Similar to the findings in panoptic segmentation, we observe consistent improvements across various backbones (ResNet50 [23] and ConvNeXt-L [39]. Additionally, as we increase the size of training dataset, we observe that the improvement gain is decreasing while evaluated on the small COCO-val and relabeled COCO-val set, indicating the performance saturation on the small validation set. By contrast, the proposed COCONut-val set presents a more challenging validation set, where the improvement of stronger backbone and more training images are more noticeable.

Semantic Segmentation: We also conduct experiments on training a single semantic segmentation model with semantic masks. Results are shown in Tab. 13. Similar observations are made. We can see subsequent mIoU gains of increasing the training dataset size for semantic specific model. Additionally, we also verify the dataset on se-

	category	**COCO definition**	**COCONut definition**
'thing'	bed	None	A piece of furniture for sleep or rest, typically a framework with a mattress and coverings (from Google dictionary). Thus we will include the pillows, comforter, blanket, and bedding sheets along with the bed frame for bed.
'stuff'	blanket	A loosely woven fabric, used for warmth while sleeping.	As blanket in the bed is included in the category of bed, then we will label blanket on the other surface excluding bed, for example, blanket on the couch or blanket on the bench.
'stuff'	pillow	A rectangular cloth bag stuffed with soft materials to support the head.	To avoid the conflicts from bed in 'thing', we exclude the pillow in the bed while labeling pillow.
'thing'	dining table	None	A table on which meals are served in a dining room (from Google dictionary). In order to have consistent COCO's definition by viewing hundreds of examples, partial table placing with food is also considered as dining table.
'stuff'	table-merged	A piece of furniture with a flat top and one or more legs.	Exclude the cases from dining table aforementioned. Include console table, coffee table, desk and *etc*.
'stuff'	roof	The structure forming the upper covering of a building.	The structure forming the upper covering of a building or vehicle (from Google dictionary). Only the outside coverings will be labeled. COCO also labels the inner side of the coverings while they should be referred as ceiling instead.
'stuff'	house	A smaller size building for human habitation.	A building for human habitation, especially one that is lived in by a family or small group of people (from Google dictionary). Typically it refers to residential house meanwhile residential apartment building is not included. To avoid overlap from roof, a house will not to separated into the parts of roof and the remaining while this happens in COCO.
'stuff'	building-other-merged	Any other type of building or structures.	For the other types of buildings, it consists of diverse types of constructions, for example, churches, stadiums, and *etc*.
'stuff'	wall-tile	A building wall made of tiles, such as used in bathrooms and kitchens.	Follow the same definition from COCO.
'stuff'	wall-stone	A building wall made of stone.	Indoor wall with specific texture of stone and partial outside wall of the building instead of the whole building. In other word, the building built with stone will be labeled as building instead of wall-stone.
'stuff'	wall-wood	A building wall made of wooden material.	Indoor and outside wall made of wood instead of the whole building.
'stuff'	wall-brick	A building wall made of bricks of clay.	Indoor and outside wall made of bricks instead of the whole building.
'stuff'	wall-other-merged	Any other type of wall.	To avoid the conflicts wall categories, we will first label the categories with specific texture and at last we label wall-other-merged. In details, we only label indoor scenes for wall-other-merged, for outdoor scenes, we will use other categories. We also need to exclude the other objects hang on the wall, for example, the frames *etc*.

Table 8. **Clear Redefinition of Specific COCO Categories:** We present the class definitions by grouping confusing categories for easier comparison to facilitate their distinction (continued in Tab. 9).

	category	**COCO definition**	**COCONut definition**
‘stuff’	gravel	A loose aggregation of small water-worn or pounded stones.	Follow the same definition from COCO.
‘stuff’	railroad	A track made of steel rails along which trains run (incl. the wooden beams).	We found that railroad often consists of the gravel and the track. In this scenario, we separate the region of gravel to be labeled as gravel and the remaining parts of the track will be labeled as railroad.
‘stuff’	playingfield	A ground marked off for various games (incl. indoor and outdoor).	Follow the same definition. But we found COCO has a large amount of missing masks for playingfield which are mislabeled as dirt-merged instead. We label all the playingfields if they can be identified no matter they are grass based or dirt based grounds.
‘stuff’	dirt-merged	Soil or earth (incl. dirt path).	Follow the same definition but exclude dirt-based playingfields.
‘stuff’	pavement-merged	A typically raised paved path for pedestrians at the side of a road.	Follow definition from COCO, to be more concrete, it includes side walk.
‘stuff’	platform	A paved way leading from one place to another.	COCO does not have consistent labeling masks for platforms while some of them are labeled as pavement-merged. We have a unified definition to take care of these cases. In particular, we label all the paved way for transportation, for example, label the pavement area for the train, subway and *etc*. as platform.
‘stuff’	net	An open-meshed fabric twisted, knotted, or woven together at regular intervals.	Follow the same definition but exclude fence made by net.
‘stuff’	fence-merged	A thin, human-constructed barrier which separates two pieces of land.	COCO has inconsistent masks for fence-merged and net. We follow a consistent definition to distinguish net from fence when it is not used as a fence to separate two pieces of land.
‘thing’	potted plant	None	A plant that is grown in a container, and usually kept inside. There exist masks for flower placed in the vase which contradicting the definition of flower and vase. We exclude these scenarios from potted plant.
‘thing’	vase	None	A decorative container, typically made of glass or china and used as an ornament or for displaying cut flowers (google dictionary).
‘stuff’	flower	The seed-bearing part of a plant (incl. the entire flower).	COCO does not clarify that whether the flowers that are placed in the vase belong to potted plant or flower. This is confusing when our raters label the images. We give the definition to separate the flower, potted plant and vase. The potted plant will not include any plants which are flowers. Then the potted plant will be labeled together with the plants and pots. While for vase, if the vase has flower, then these parts need to be separate.

Table 9. **Clear Redefinition of Specific COCO Categories:** We present the class definitions by grouping confusing categories for easier comparison to facilitate their distinction (continued in Tab. 10).

mantic segmentation using ViT backbones [14], *e.g.*, ViT-Adapter[9]. We follow the same configuration and use the codebase from the paper[3] to conduct our experiments but replace the dataset from COCO-stuff to our COCONut se-

[3] https://github.com/czczup/ViT-Adapter.git

category	COCO definition	COCONut definition
food-other-merged	Any other type of food.	To avoid the conflicts from similar categories of 'thing', we explicitly highlight that we DO NOT label those categories. The categories include sandwich (burger), hot dog, pizza, donut, cake, broccoli and carrot. Excluding all the food aforementioned, the other types of food need to be labeled.
paper-merged	A material manufactured in thin sheets from the pulp of wood.	Include tissue, toilet paper, poster, kitchen paper towel, and *etc*. They are often shown with a single or multiple pieces of papers.
tree-merged	A woody plant, typically having a single trunk growing to a considerable height and bearing lateral branches at some distance from the ground.	Include bush.
fruit	The sweet and fleshy product of a tree or other plant.	Exclude fruits in 'thing', banana, orange and apple. Include tomato and all other kinds of fruit.

Table 10. **Clear Redefinition of Specific COCO Categories:** We clearly redefine certain COCO categories to avoid annotation confusion.

method	backbone	training set	COCO-val PQ	relabeled COCO-val PQ	COCONut-val PQ
kMaX-DeepLab	ResNet50	COCO	53.3	55.1	53.1
		COCONut-S	51.7	58.9	56.7
		COCONut-B	53.4	60.2	58.1
		COCONut-L	54.1	60.7	60.7
	ConvNeXt-L	COCO	57.9	60.4	58.3
		COCONut-S	55.9	64.4	59.4
		COCONut-B	57.8	64.9	61.3
		COCONut-L	58.1	65.1	62.7

Table 11. **Benchmarking Task-Specific Panoptic Segmentation Models:** kMaX-DeepLab is trained with *panoptic* segmentation annotations across various training and validation sets.

method	backbone	training set	COCO-val mIoU	relabeled COCO-val mIoU	COCONut-val mIoU
kMaX-DeepLab	ResNet50	COCO	59.5	64.6	62.9
		COCONut-S	59.3	66.4	65.1
		COCONut-B	63.5	67.3	66.5
		COCONut-L	64.2	68.0	67.8
	ConvNeXt-L	COCO	67.1	70.9	68.1
		COCONut-S	66.1	71.9	69.9
		COCONut-B	67.4	72.4	71.3
		COCONut-L	67.5	72.7	72.6

Table 13. **Benchmarking Task-Specific Semantic Segmentation Models:** kMaX-DeepLab is trained with *semantic* segmentation annotations across various training and validation sets.

method	backbone	training set	COCO-val AP^{mask}	relabeled COCO-val AP^{mask}	COCONut-val AP^{mask}
kMaX-DeepLab	ResNet50	COCO	44.1	44.6	41.9
		COCONut-S	40.9	49.2	44.9
		COCONut-B	41.2	50.3	46.2
		COCONut-L	41.4	50.9	47.1
	ConvNeXt-L	COCO	49.2	50.2	47.1
		COCONut-S	45.5	55.8	51.2
		COCONut-B	46.4	56.7	52.9
		COCONut-L	47.0	57.0	53.8

Table 12. **Benchmarking Task-Specific Instance Segmentation Models:** kMaX-DeepLab is trained with *instance* segmentation annotations across various training and validation sets.

		evaluation set (mIoU)		
backbone	training dataset	COCO-val	relabeled COCO-val	COCONut-val
ViT-Adapter-B	COCO	61.2	64.5	61.8
	COCONut-S	60.6	66.0	64.9
	COCONut-B	61.3	66.9	66.3
	COCONut-L	62.4	67.7	67.1
ViT-Adapter-L	COCO	66.6	69.9	67.5
	COCONut-S	65.2	71.0	69.5
	COCONut-B	66.4	72.1	70.7
	COCONut-L	67.2	72.3	71.0

Table 14. **Benchmarking plain ViT backbone for Semantic Segmentation:** Mask2Former w/ ViT-Adapter is trained with *semantic* segmentation annotations across various training and validation sets.

mantic segmentation dataset. Similar observation is made: the model saturates when testing on our relabeled COCO-val set but the performance is improved on COCONut-val.

Open-Vocabulary Segmentation: Tab. 15 summarizes the results for open-vocabulary segmentation using FC-CLIP. As shown in the table, FC-CLIP benefits from COCONut's high-quality and large-scale annotations, achieving the performance of 27.5 PQ on ADE20K, setting a new state-of-the-art.

Bounding Box Object Detection: The results for object detection are shown in Tab. 16. As shown in the table, the detection model with ResNet50 benefits significantly from the high quality data COCONut-S with a large margin of 4.3 on relabeled COCO-val set. Similarly, subsequent gains from training size are observed for both backbones.

Mask Conditional Image Synthesis: We conduct mask conditional image synthesis to verify the annotation quality for generation. We employ a mask-conditional model GLIGEN [34] and train the model on paired image-mask data from COCO and COCONut-S separately. Once we have the trained model checkpoint, we perform inference on

			ADE20K-150			A-847	PC-459	PC-59	PAS-21
method	backbone	training data	PQ	AP^{mask}	mIoU	mIoU	mIoU	mIoU	mIoU
FC-CLIP	ConvNeXt-L	COCO	26.8	16.8	34.1	14.8	18.2	58.4	81.8
		COCONut-S	27.3	17.3	33.8	15.3	20.4	57.5	82.1
		COCONut-B	27.4	17.4	33.7	15.5	20.1	58.5	82.0
		COCONut-L	27.5	17.4	33.9	15.6	20.6	58.0	81.9

Table 15. **Benchmarking Open-Vocabulary Segmentation:** We ablate the effect of using different training data to train the mask proposal network of FC-CLIP [66]. The performance is evaluated on multiple segmentation datasets in a zero-shot manner.

			COCO-val	relabeled COCO-val	COCONut-val
method	backbone	training set	AP^{box}	AP^{box}	AP^{box}
DETA	ResNet50	COCO	50.4	49.5	46.1
		COCONut-S	47.8	53.8	49.5
		COCONut-B	50.4	54.4	51.4
		COCONut-L	50.6	54.9	53.7
	Swin-L	COCO	59.1	58.6	56.1
		COCONut-S	54.5	61.3	58.9
		COCONut-B	59.3	62.2	60.1
		COCONut-L	60.1	62.3	61.7

Table 16. **Benchmarking Bounding Box Object Detection:** We conduct the experiments using the DETA framework [43], employing various backbones and diverse training and validation sets. The backbones are exclusively pretrained on ImageNet [50].

		COCO-val		relabeled COCO-val		COCONut-val	
method	training set	FID ↓	mIoU ↑	FID ↓	mIoU ↑	FID ↓	mIoU ↑
GLIGEN	COCO	18.51	32.1	-	33.7	17.4	30.9
	COCONut-S	18.39	30.4	-	34.8	16.8	32.6

Table 17. **Benchmarking Mask-Conditional Image Synthesis:** We conduct the experiments using the GLIGEN framework [34] mIoU is measured with another off-the-shelf Mask2Former [12], as a referee.

mask-conditioned generation by giving masks from COCO val set, relabeled COCO-val set, and COCONut-val set individually to compute FID. The lower FID shows better image synthesis performance. Besides, we adopt the off-the-shelf Mask2Former [12] model to perform semantic segmentation by giving the generated images as input and report mIoU for evaluation. As shown in Tab. 17, our high-quality mask annotation can result in better image synthesis with 18.39 FID on COCO-val set and 16.8 FID on COCONut-val set. Besides, the higher-quality generated images can be better inferred via the higher segmentation mIoU scores. Even for a more challenging val set, training on COCONut-S outperforms the COCO dataset.

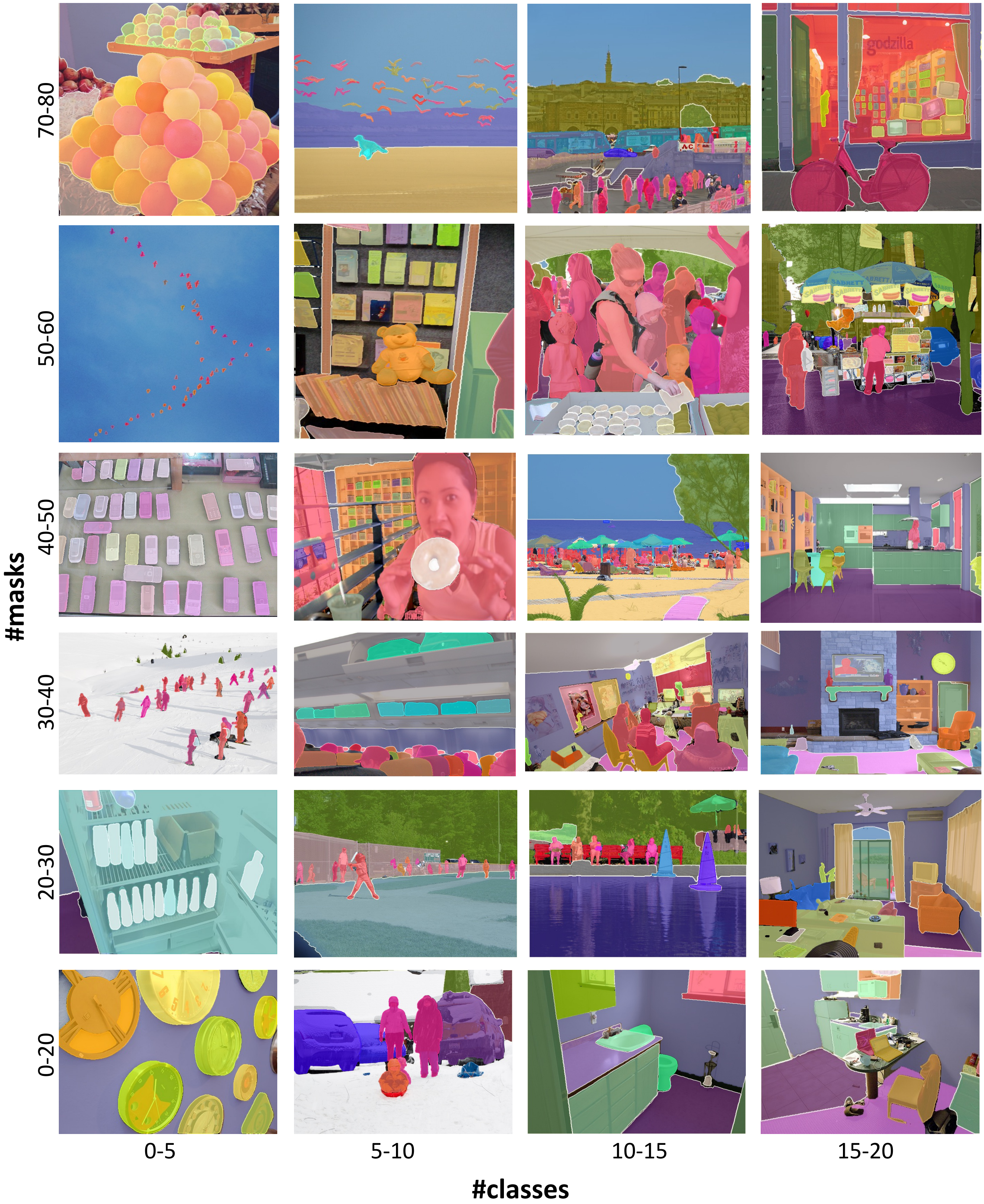


Figure 10. **Visualization of COCONut Annotations:** This figure demonstrates COCONut annotations with images sourced from COCO unlabeled set images. COCONut provides annotations comprising a large number of classes and masks.

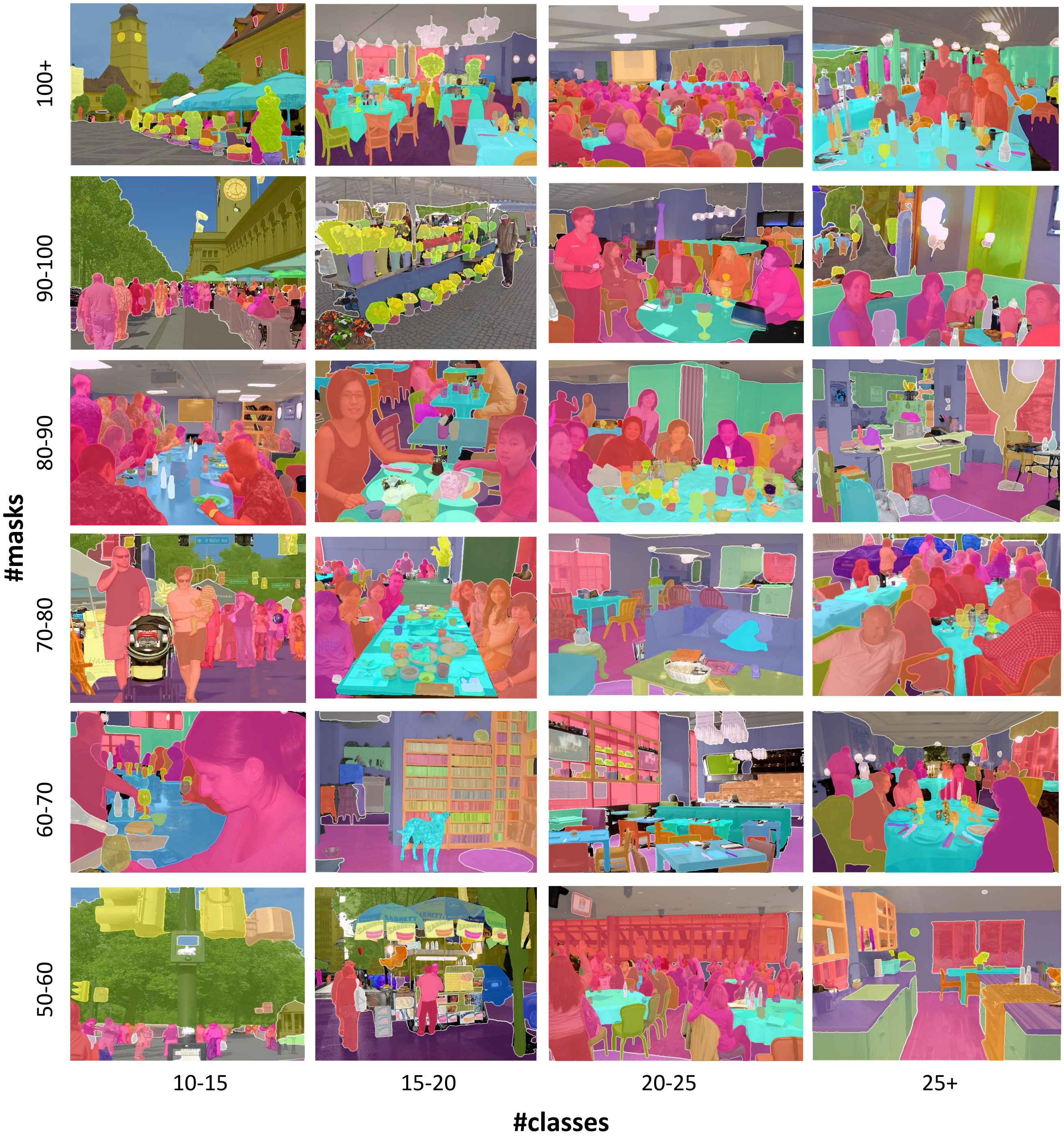


Figure 11. **Visualization of COCONut Annotations:** This figure showcases COCONut annotations using images sourced from both the COCO unlabeled set and Objects365 images. The inclusion of Objects365 images enriches COCONut annotations by introducing a wider variety of classes and masks compared to the COCO images.

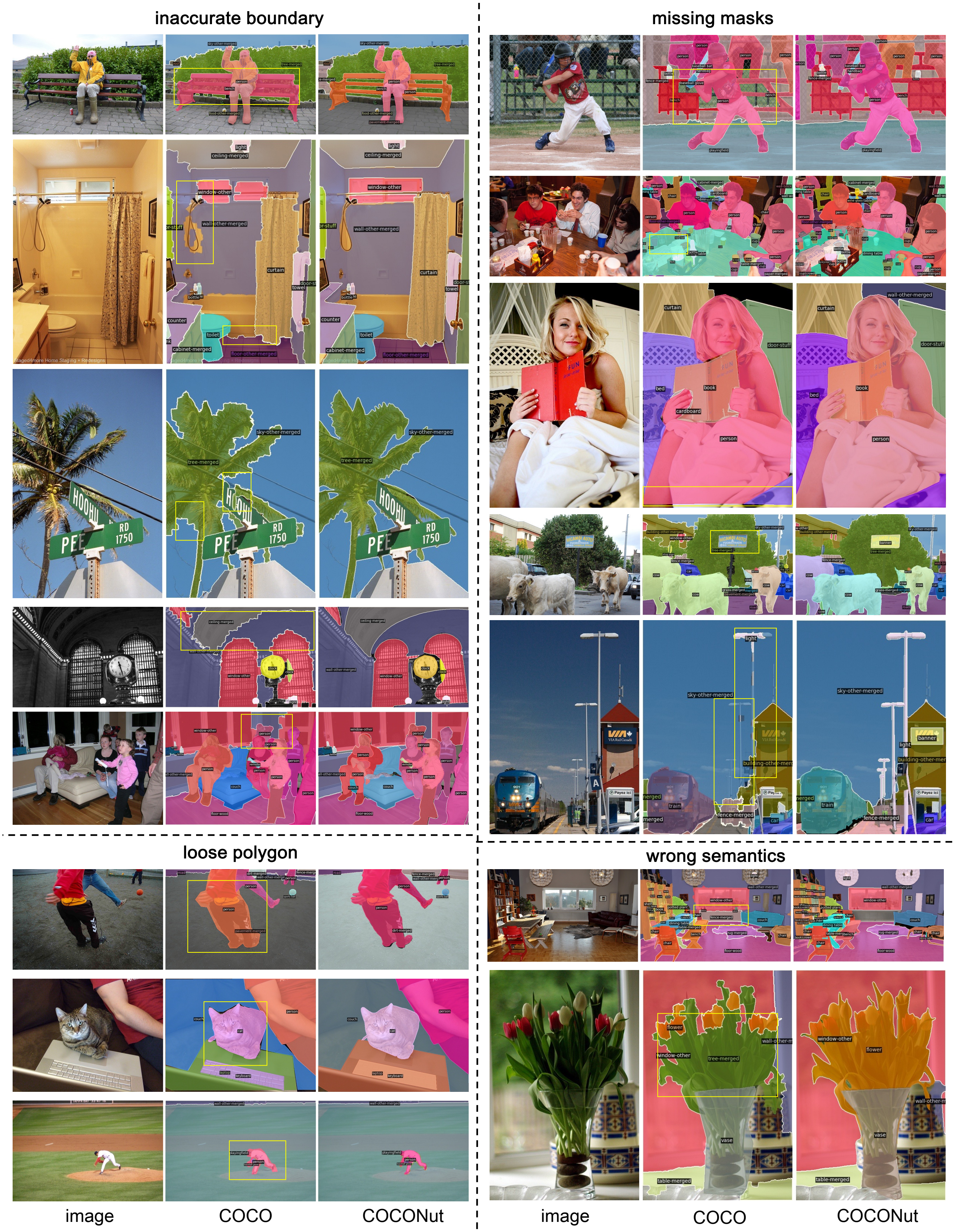


Figure 12. **Visualization Comparison Between COCO and COCONut:** COCONut effectively mitigates the annotations errors by COCO. The yellow boxes highlight the erroneous areas in COCO.

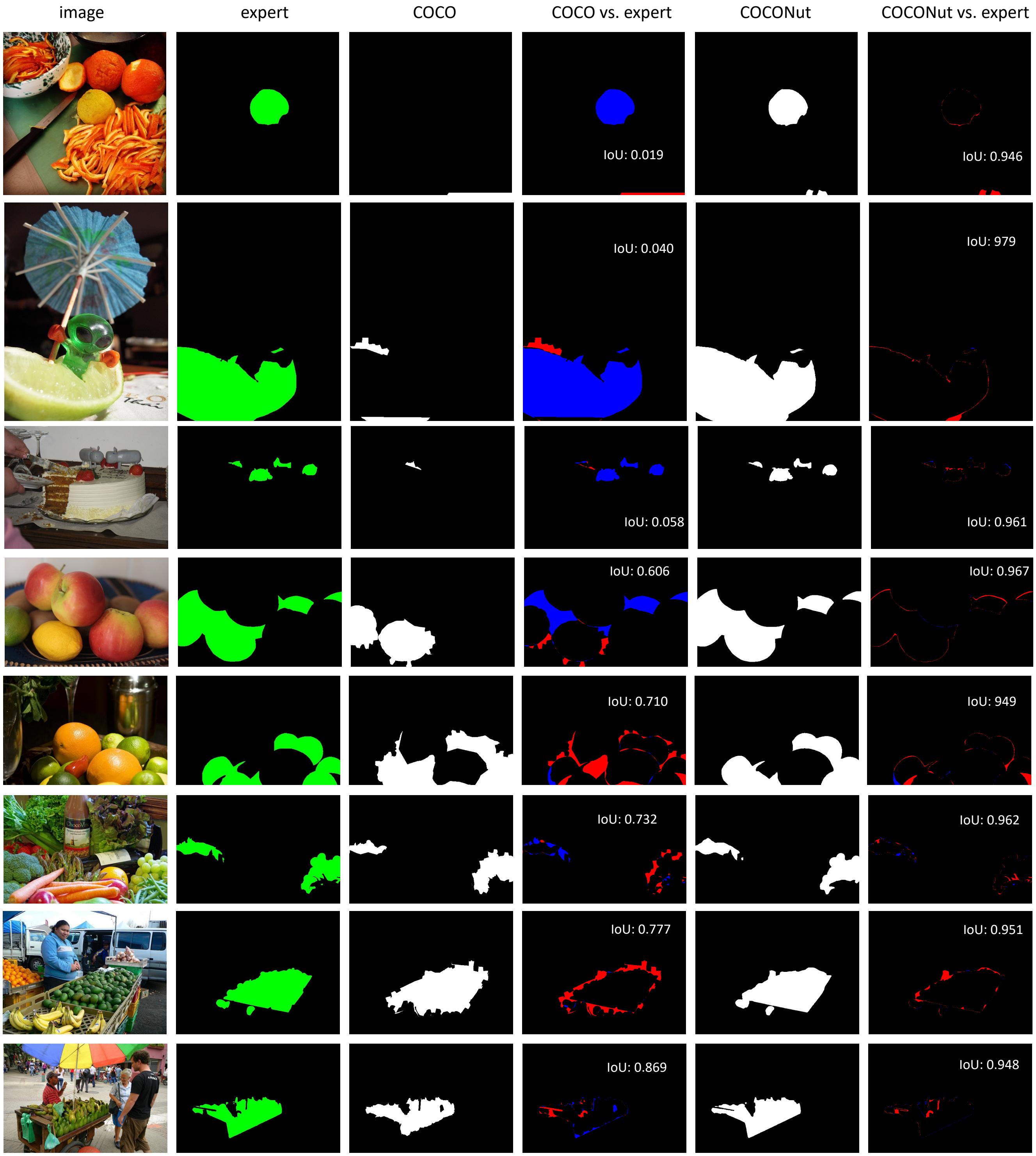


Figure 13. **Annotation Comparison:** We show annotations obtained by COCO, COCONut with Point2Mask for 'stuff', and our expert rater. COCONut's annotation exhibits sharper boundaries, closely resembling expert results, as evident from higher IoU values. The blue and red regions correspond to extra and missing regions, respectively, compared to the expert mask.

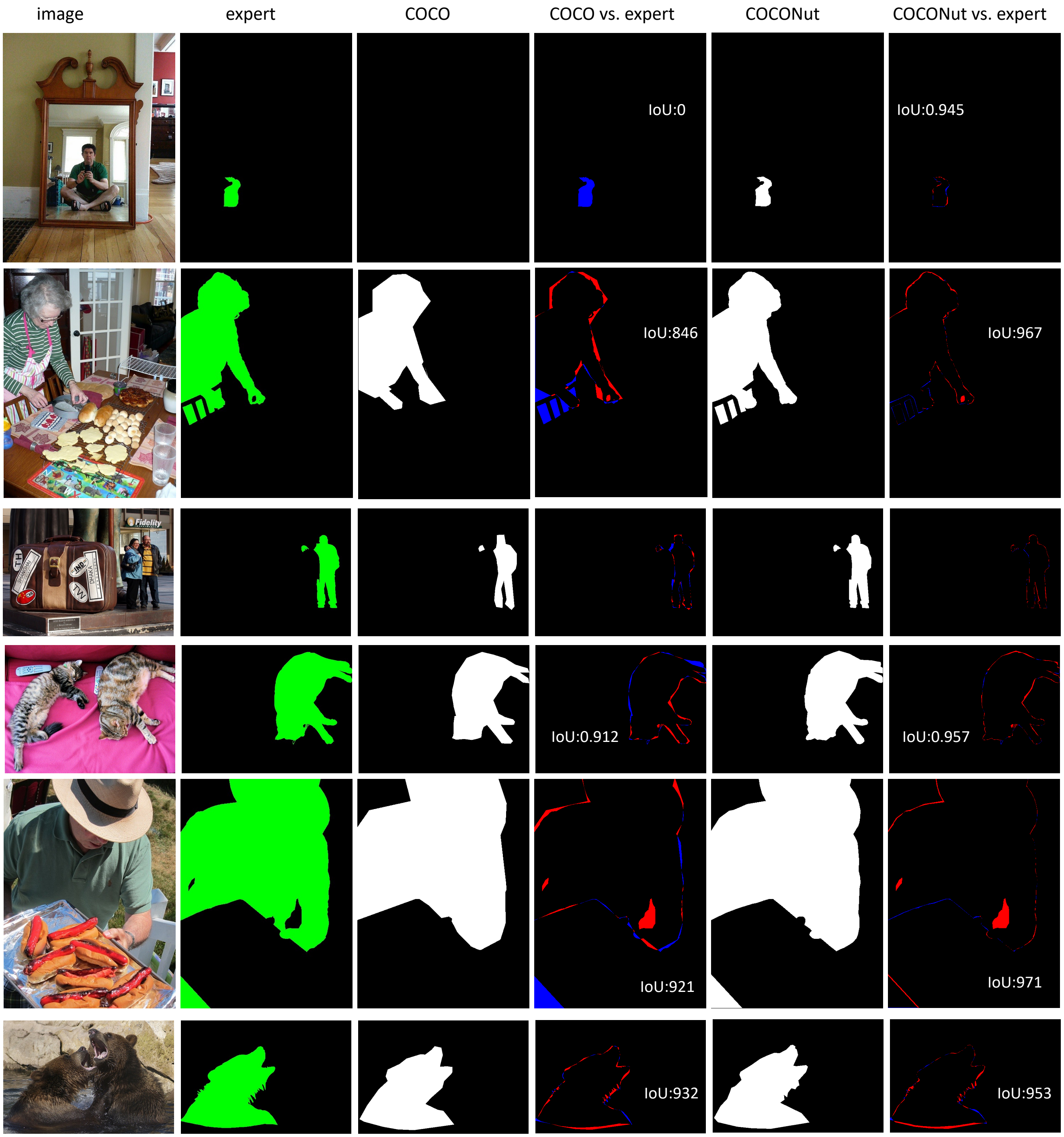
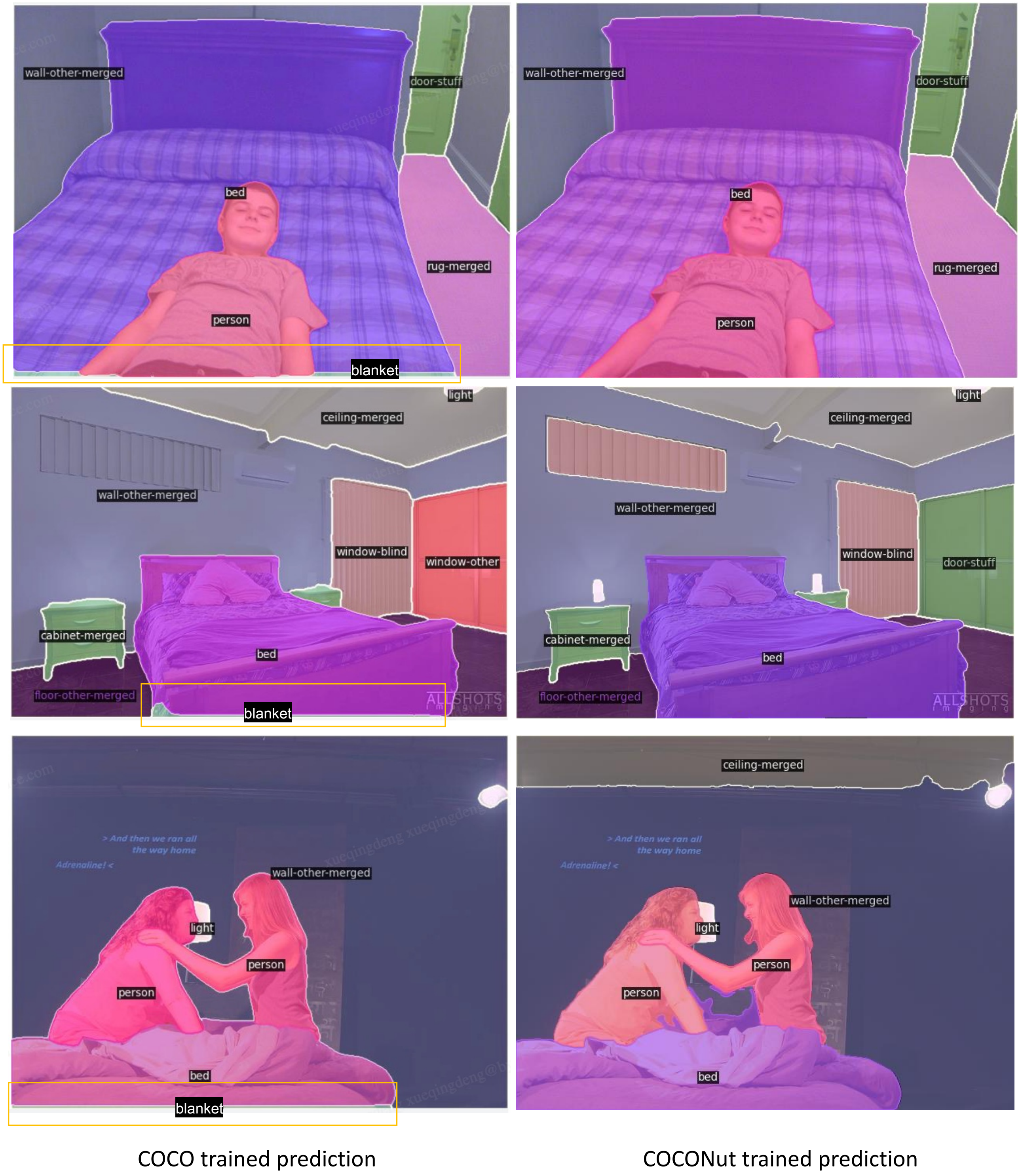


Figure 14. **Annotation Comparison:** We show annotations obtained by COCO, COCONut with Box2Mask for 'thing', and our expert rater. COCONut's annotation exhibits sharper boundaries, closely resembling expert results, as evident from higher IoU values. The blue and red regions correspond to extra and missing regions, respectively, compared to the expert mask.



COCO trained prediction

COCONut trained prediction

Figure 15. **Influence of Training Data on Predictions:** We present predictions from two models: one trained on original COCO (left) and the other on COCONut (right). The COCO-trained model predicts a small isolated mask, influenced by the biases inherent in the COCO coarse annotations.

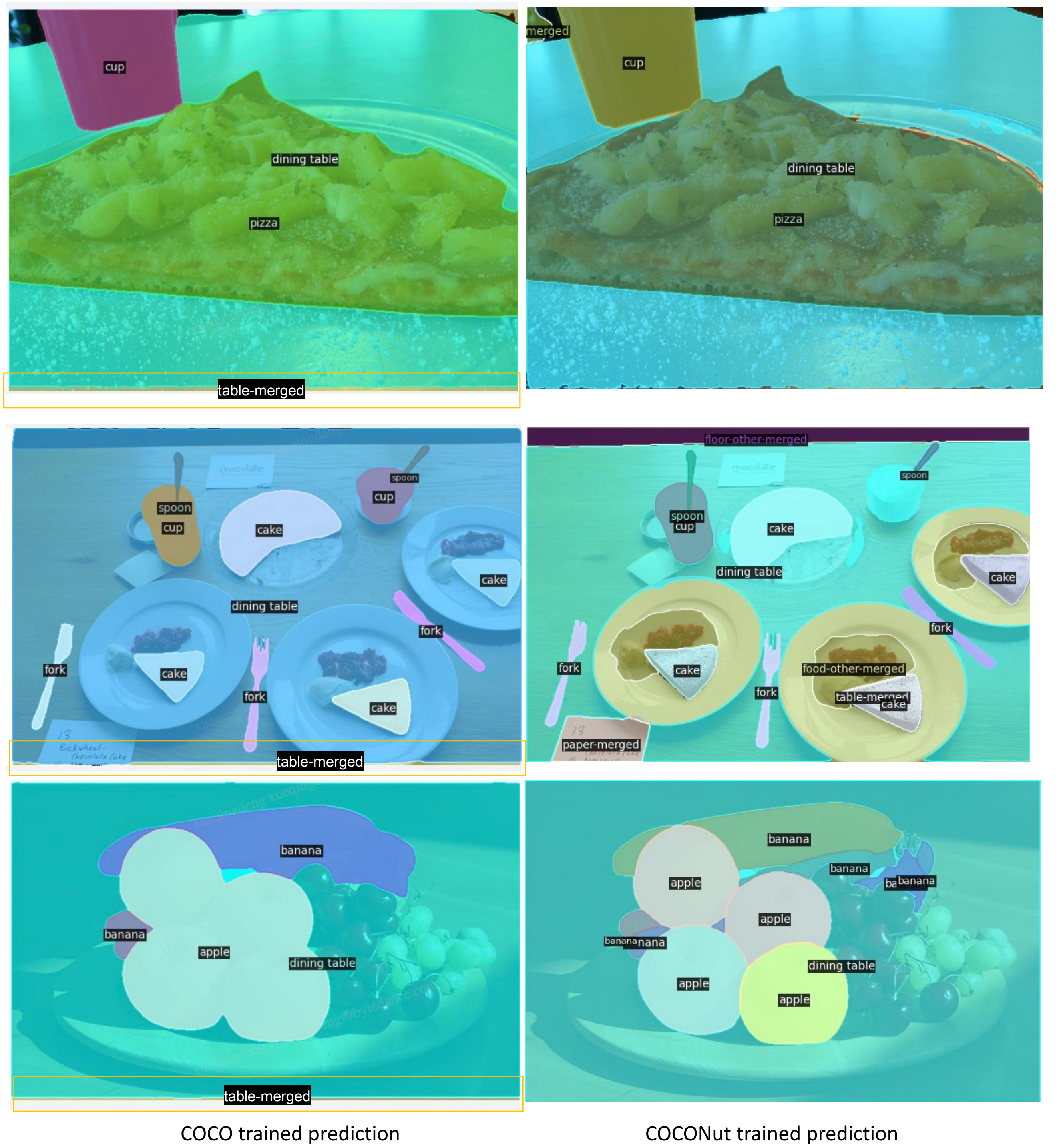


Figure 16. **Influence of Training Data on Predictions:** We present predictions from two models: one trained on original COCO (left) and the other on COCONut (right). The COCO-trained model predicts a small isolated mask, influenced by the biases inherent in the COCO coarse annotations.