Causal Intervention for Weakly-Supervised Semantic Segmentation

Dong Zhang[1] **Hanwang Zhang**[2] **Jinhui Tang**[1*] **Xiansheng Hua**[3] **Qianru Sun**[4]
[1]School of Computer Science and Engineering, Nanjing University of Science and Technology;
[2]Nanyang Technological University; [3]Damo Academy, Alibaba Group; [4]Singapore Management University.

Abstract

We present a causal inference framework to improve Weakly-Supervised Semantic Segmentation (WSSS). Specifically, we aim to generate better pixel-level pseudo-masks by using only image-level labels — the most crucial step in WSSS. We attribute the cause of the ambiguous boundaries of pseudo-masks to the confounding context, *e.g.*, the correct image-level classification of "horse" and "person" may be not only due to the recognition of each instance, but also their co-occurrence context, making the model inspection (*e.g.*, CAM) hard to distinguish between the boundaries. Inspired by this, we propose a structural causal model to analyze the causalities among images, contexts, and class labels. Based on it, we develop a new method: Context Adjustment (CONTA), to remove the confounding bias in image-level classification and thus provide better pseudo-masks as ground-truth for the subsequent segmentation model. On PASCAL VOC 2012 and MS-COCO, we show that CONTA boosts various popular WSSS methods to new state-of-the-arts.[1]

1 Introduction

Semantic segmentation aims to classify each image pixel into its corresponding semantic class [37]. It is an indispensable computer vision building block for scene understanding applications such as autonomous driving [60] and medical imaging [20]. However, the pixel-level labeling is expensive, *e.g.*, it costs about 1.5 man-hours for one 500×500 daily life image [14]. Therefore, to scale up, we are interested in *Weakly-Supervised Semantic Segmentation* (WSSS), where the "weak" denotes a much cheaper labeling cost at the instance-level [10, 33] or even at the image-level [26, 63]. In particular, we focus on the latter as it is the most economic way — only a few man-seconds for tagging an image [31].

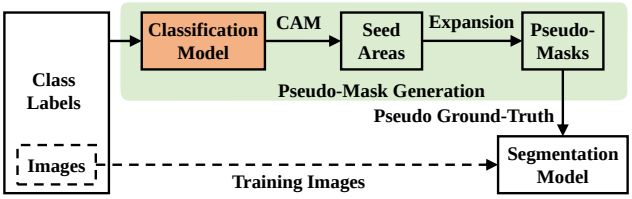

Figure 1: The prevailing pipeline for training WSSS. Our contribution is to improve the Classification Model, which is the foundation for better pseudo-masks.

The prevailing pipeline for training WSSS is depicted in Figure 1. Given training images with only image-level class labels, we first train a multi-label classification model. Second, for each image, we infer the class-specific seed areas, *e.g.*, by applying Classification Activation Map (CAM) [74] to the above trained model. Finally, we expand them to obtain the *Pseudo-Masks* [22, 63, 65], which are

*Corresponding author.

[1]Code is open-sourced at: https://github.com/ZHANGDONG-NJUST/CONTA

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

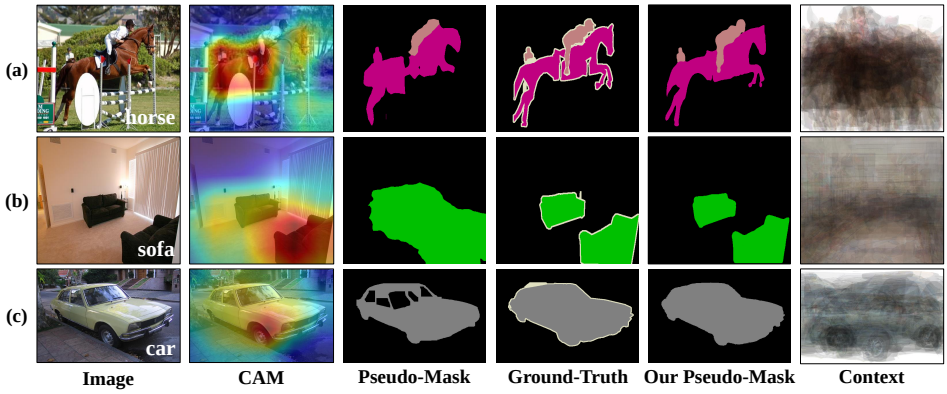

Figure 2: Three basic problems in existing pseudo-masks [63] (dataset: PASCAL VOC 2012 [14]): (a) Object Ambiguity, (b) Incomplete Background, (c) Incomplete Foreground. They usually combine to cause other complications. The context (mean image per class) may provide clues for the reasons.

used as the pseudo ground-truth for training a standard supervised semantic segmentation model [9]. You might be concerned, there is no free lunch — it is essentially ill-posed to infer pixel-level masks from only image-level labels, especially when the visual scene is complex. Although most previous works have noted this challenge [1, 22, 63], as far as we know, no one answers the whys and wherefores. In this paper, we contribute a formal answer based on causal inference [42] and propose a principled and fundamental solution.

As shown in Figure 2, we begin with illustrating the three basic problems that cause the complications in pseudo-mask generation:

Object Ambiguity: Objects are not alone. They usually co-occur with each other under certain contexts. For example, if most "horse" images are about "person riding horse", a classification model will wrongly generalize to "most horses are with people" and hence the generated pseudo-masks are ambiguous about the boundary between "person" and "horse".

Incomplete Background: Background is composed of (unlabeled) semantic objects. Therefore, the above ambiguity also holds due to the co-occurrence of foreground and background objects, *e.g.*, some parts of the background "floor" are misclassified as the foreground "sofa".

Incomplete Foreground: Some semantic parts of the foreground object, *e.g.*, the "window" of "car", co-vary with different contexts, *e.g.*, the window reflections of the surroundings. Therefore, the classification model resorts to using the less context-dependent (*i.e.*, discriminative) parts to represent the foreground, *e.g.*, the "wheel" part is the most representative of "car".

So far, we can see that all the above problems are due to the context prior in dataset. Essentially, the context is a *confounder* that misleads the image-level classification model to learn spurious correlations between pixels and labels, *e.g.*, the inconsistency between the CAM-expanded pseudo-masks and the ground-truth masks in Figure 2. More specifically, although the confounder is helpful for a better association between the image pixels X and labels Y via a model $P(Y|X)$, *e.g.*, it is likely a "sofa" when seeing a "floor" region, $P(Y|X)$ mistakenly 1) associates non-causal but positively correlated pixels to labels, *e.g.*, the "floor" region wrongly belongs to "sofa", 2) disassociates causal but negatively correlated ones, *e.g.*, the "window" region is wrongly classified as "non-car". To this end, we propose to use $P(Y|do(X))$ instead of $P(Y|X)$ to find what pixels truly cause the labels, where the do-operation denotes the pursuit of the causality between the cause X and the effect Y without the confounding effect [44]. The ideal way to calculate $P(Y|do(X))$ is to "physically" intervene X (a.k.a., randomised controlled trial [8]) — if we could have photographed any "sofa" under any context [13], then $P(sofa|do(X)) = P(sofa|X)$. Intrigued, you are encouraged to think about the causal reason why $P(car|X)$ can robustly localize the "wheel" region in Figure 2?[2]

In Section 3.1, we formulate the causalities among pixels, contexts, and labels in a unified Structural Causal Model [41] (see Figure 3 (a)). Thanks to the model, we propose a novel WSSS pipeline called:

[2] **Answer:** the "wheel" was photographed in every "car" under any context by the dataset creator

Context Adjustment (CONTA). CONTA is based on the backdoor adjustment [42] for $P(Y|do(X))$. Instead of the prohibitively expensive "physical" intervention, CONTA performs a practical "virtual" one from only the observational dataset (the training data *per se*). Specifically, CONTA is an iterative procedure that generates high-quality pseudo-masks. We achieve this by proposing an effective approximation for the backdoor adjustment, which fairly incorporates every possible context into the multi-label classification, generating better CAM seed areas. In Section 4.3, we demonstrate that CONTA can improve pseudo-marks by 2.0% mIoU on average and overall achieves a new state-of-the-art by 66.1% mIoU on the *val* set and 66.7% mIoU on the *test* set of PASCAL VOC 2012 [14], and 33.4% mIoU on the *val* set of MS-COCO [35].

2 Related Work

Weakly-Supervised Semantic Segmentation (WSSS). To address the problem of expensive labeling cost in fully-supervised semantic segmentation, WSSS has been extensively studied in recent years [1, 65]. As shown in Figure 1, the prevailing WSSS pipeline [26] with only the image-level class labels [2, 63] mainly consists of the following two steps: pseudo-mask generation and segmentation model training. The key is to generate the pseudo-masks as perfect as possible, where the "perfect" means that the pseudo-mask can reveal the entire object areas with accurate boundaries [1]. To this end, existing methods mainly focus on generating better seed areas [30, 63, 65, 64] and expanding these seed areas [1, 2, 22, 26, 61]. In this paper, we also follow this pipeline and our contribution is to propose an iterative procedure to generate high-quality seed areas.

Visual Context. Visual context is crucial for recognition [13, 50, 59]. The majority of WSSS models [1, 22, 63, 65] implicitly use context in the backbone network by enlarging the receptive fields with the help of dilated/atrous convolutions [70]. There is a recent work that explicitly uses contexts to improve the multi-label classifier [55]: given a pair of images, it encourages the similarity of the foreground features of the same class and the contrast of the rest. In this paper, we also explicitly use the context, but in a novel framework of causal intervention: the proposed context adjustment.

Causal Inference. The purpose of causal inference [44, 48] is to empower models the ability to pursue the causal effect: we can remove the spurious bias [6], disentangle the desired model effects [7], and modularize reusable features that generalize well [40]. Recently, there is a growing number of computer vision tasks that benefit from causality [39, 45, 57, 58, 62, 69, 71]. In our work, we adopt the Pearl's structural causal model [41]. Although the Rubin's potential outcome framework [47] can also be used, as the two are fundamentally equivalent [18, 43], we prefer Pearl's because it can explicitly introduce the causality in WSSS — every node in the graph can be located and implemented in the WSSS pipeline. Nevertheless, we encourage readers to explore Rubin's when some causalities cannot be explicitly hypothesized and modeled, such as using the prospensity scores [3].

3 Context Adjustment

Recall in Figure 1 that the pseudo-mask generation is the bottleneck of WSSS, and as we discussed in Section 1, the inaccurate CAM-generated seed areas are due to the context confounder C that misleads the classification model between image X and label Y. In this section, we will use a causal graph to fundamentally reveal how the confounder C hurts the pseudo-mask quality (Section 3.1) and how to remove it by using causal intervention (Section 3.2).

3.1 Structural Causal Model

We formulate the causalities among pixel-level image X, context prior C, and image-level labels Y, with a Structural Causal Model (SCM) [41]. As illustrated in Figure 3 (a), the direct links denote the causalities between the two nodes: cause → effect. Note that the newly added nodes and links other than $X \rightarrow Y$[3] are not deliberately imposed on the original image-level classification; in contrast, they are the ever-overlooked causalities. Now we detail the high-level rationale behind the SCM and defer its implementation in Section 3.2.

[3]Some studies [51] show that label causes image ($X \leftarrow Y$). We believe that such anti-causal assumption only holds when the label is as simple as the disentangled causal mechanisms [40, 56] (*e.g.*, 10-digit in MNIST).

$\boldsymbol{C \rightarrow X}$. Context prior C determines what to picture in image X. By "context prior", we adopt the general meaning in vision: the relationships among objects in a visual scene [38]. Therefore, C tells us where to put "car", "road", and "building" in an image. Although building a generative model for $C \rightarrow X$ is extremely challenging for complex scenes [24], fortunately, as we will introduce later in Section 3.2, we can avoid it in causal intervention.

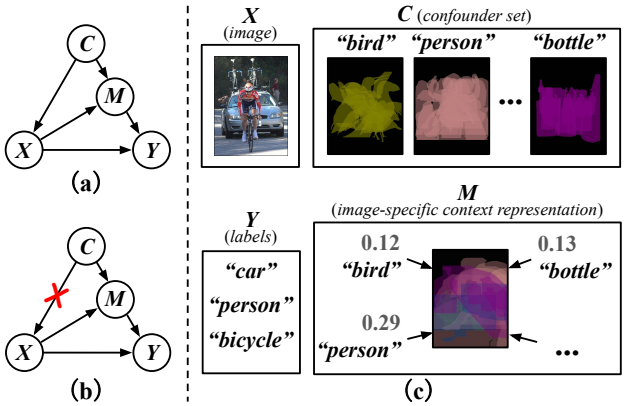

Figure 3: (a) The proposed Structural Causal Model (SCM) for causality of multi-label classifier in WSSS, (b) The intervened SCM for the causality of multi-label classifier in WSSS, (c) The realization of each component in CONTA.

$\boldsymbol{C \rightarrow M \leftarrow X}$. M is an image-specific representation using the contextual templates from C. For example, a car image can be delineated by using a "car" context template filled with detailed attributes, where the template is the prototypical shape and location of "car" (foreground) in a scene (background). Note that this assumption is not *ad hoc* in our model, in fact, it underpins almost every concept learning method from the classic Deformable Part Models [15] to modern CNNs [17], whose cognitive evidence can be found in [29]. A plausible realization of M and C used in Section 3.2 is illustrated in Figure 3 (c).

$\boldsymbol{X \rightarrow Y \leftarrow M}$. A general C cannot directly affect the labels Y of an image. Therefore, besides the conventional classification model $X \rightarrow Y$, Y is also the effect of the X-specific mediation M. $M \rightarrow Y$ denotes an obvious causality: the contextual constitution of an image affects the image labels. It is worth noting that even if we do not explicitly take M as an input for the classification model, $M \rightarrow Y$ still holds. The evidence lies in the fact that visual contexts will emerge in higher-level layers of CNN when training image classifiers [72, 74], which essentially serve as a feature map backbone for modern visual detection that highly relies on contexts, such as Fast R-CNN [16] and SSD [36]. To think conversely, if $M \not\rightarrow Y$ in Figure 3 (a), the only path left from C to Y: $C \rightarrow X \rightarrow Y$, is cut off conditional on X, then no contexts are allowed to contribute to the labels by training $P(Y|X)$, and thus we would never uncover the context, *e.g.*, the seed areas. So, WSSS would be impossible.

So far, we have pinpointed the role of context C played in the causal graph of image-level classification in Figure 3 (a). Thanks to the graph, we can clearly see how C confounds X and Y via the backdoor path $X \leftarrow C \rightarrow M \rightarrow Y$: even if some pixels in X have nothing to do with Y, the backdoor path can still help to correlate X and Y, resulting the problematic pseudo-masks in Figure 2. Next, we propose a causal intervention method to remove the confounding effect.

3.2 Causal Intervention via Backdoor Adjustment

We propose to use causal intervention: $P(Y|do(X))$, as the new image-level classifier, which removes the confounder C and pursues the true causality from X to Y so as to generate better CAM seed areas. As the "physical" intervention — collecting objects in any context — is impossible, we apply the backdoor adjustment [44] to "virtually" achieve $P(Y|do(X))$. The key idea is to 1) cut off the link $C \rightarrow X$ in Figure 3 (b), and 2) stratify C into pieces $C = \{c\}$. Formally, we have:

$$P(Y|do(X)) = \sum_{c} P\left(Y|X, M = f(X, c)\right) P(c), \tag{1}$$

where $f(\cdot)$ is a function defined later in Eq. (3). As C is no longer correlated with X, the causal intervention makes X have a fair opportunity to incorporate every context c into Y's prediction, subject to a prior $P(c)$.

However, C is not observable in WSSS, let alone stratifying it. To this end, as illustrated in Figure 3 (c), we use the class-specific average mask in our proposed Context Adjustment (CONTA) to approximate the confounder set $C = \{c_1, c_2, ..., c_n\}$, where n is the class size in dataset and $c \in \mathbb{R}^{h \times w}$ corresponds to the $h \times w$ average mask of the i-th class images. M is the X-specific mask which can be viewed

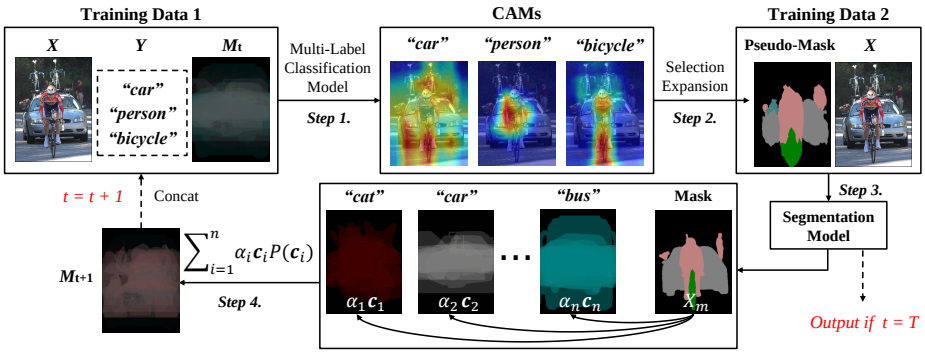

Figure 4: Overview of our proposed Context Adjustment (CONTA). M_t is an empty set when $t = 0$.

as a linear combination of $\{c\}$. Note that the rationale behind our C's implementation is based on the definition of context: the relationships among the objects [38], and thus each stratification is about one class of object interacting with others (*i.e.*, the background). So far, how do we obtain the unobserved masks? In CONTA, we propose an iterative procedure to establish the unobserved C.

Figure 4 illustrates the overview of CONTA. The input is training images with only class labels (**Training Data 1**, $t = 0$), the output is a segmentation model ($t = T$), which is trained on CONTA generated pseudo-masks (**Training Data 2**). Before we delve into the steps below, we highlight that CONTA is essentially an EM algorithm [66], if you view Eq. (1) as an objective function (where we omit the model parameter Θ) of observed data X and missing data C. Thus, its convergence is theoretically guaranteed. As you may realize soon, the E-step is to calculate the expectation ($\sum_c$ in Eq. (1)) over the estimated masks in $C|(X, \Theta_t)$ (***Step 2, 3, 4***); and the M-step is to maximize Eq. (1) for Θ_{t+1} (***Step 1***).

Step 1. **Image Classification.** We aim to maximize $P(Y|do(X))$ for learning the multi-label classification model, whereby the subsequent CAM will yield better seed areas. Our implementation for Eq. (1) is:

$$P(Y|do(X); \Theta_t) = \prod_{i=1}^{n} \left[\mathbb{1}_{i \in Y} \frac{1}{1 + \exp(-s_i)} + \mathbb{1}_{i \notin Y} \frac{1}{1 + \exp(s_i)} \right], \tag{2}$$

where $\mathbb{1}$ is 1/0 indicator, $s_i = f(X, M_t; \theta_t^i)$ is the i-th class score function, consisting of a class-shared convolutional network on the channel-wise concatenated feature maps $[X, M_t]$, followed by a class-specific fully-connected network (the last layer is based on a global average pooling [34]). Overall, Eq. (2) is a joint probability over all the n classes that encourages the ground-truth labels $i \in Y$ and penalizes the opposite $i \notin Y$. In fact, the negative log-likelihood loss of Eq. (2) is also known as the multi-label soft-margin loss [49]. Note that the expectation $\sum_c$ is absorbed in M_t, which will be detailed in ***Step 4***.

Step 2. **Pseudo-Mask Generation.** For each image, we can calculate a set of class-specific CAMs [74] using the trained classifier above. Then, we follow the conventional two post-processing steps: 1) We select hot CAM areas (subject to a threshold) for seed areas [2, 63]; and 2) We expand them to be the final pseudo-masks [1, 26].

Step 3. **Segmentation Model Training.** Each pseudo-mask is used as the pseudo ground-truth for training any standard supervised semantic segmentation model. If $t = T$, this is the model for delivery; otherwise, its segmentation mask can be considered as an additional post-processing step for pseudo-mask smoothing. For fair comparisons with other WSSS methods, we adopt the classic DeepLab-v2 [9] as the supervised semantic segmentation model. Performance boost is expected if you adopt more advanced ones [32].

Step 4. **Computing M_{t+1}.** We first collect the predicted segmentation mask X_m of every training image from the above trained segmentation model. Then, each class-specific entry c in the confounder set C is the averaged mask of X_m within the corresponding class and is reshaped into a $hw \times 1$ vector. So far, we are ready to calculate Eq. (1). However, the cost of the network forward pass for all the n classes is expensive. Fortunately, under practical assumptions (see Appendix 2), we can adopt the Normalized Weighted Geometric Mean [68] to move the outer sum $\sum_c P(\cdot)$ into the feature level: $\sum_c P(Y|X, M)P(c) \approx P(Y|X, M = \sum_c f(X, c)P(c))$, thus, we only need to feed-forward the

network once. We have:

$$M_{t+1} = \sum_{i=1}^{n} \alpha_i c_i P(c_i),\ \alpha_i = softmax\left(\frac{(\mathbf{W}_1 X_m)^T(\mathbf{W}_2 c_i)}{\sqrt{n}}\right), \tag{3}$$

where α_i is the normalized similarity (softmax over n similarities) between X_m and the i-th entry c_i in the confounder set C. To make CONTA beyond the dataset statistics *per se*, $P(c_i)$ is set as the uniform $1/n$. $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{n \times hw}$ are two learnable projection matrices, which are used to project X_m and c_i into a joint space. $\sqrt{n}$ is a constant scaling factor that is used as for feature normalization as in [62].

4 Experiments

We evaluated the proposed CONTA in terms of the model performance quantitatively and qualitatively. Below we introduce the datasets, evaluation metric, and baseline models. We demonstrate the ablation study, show the effectiveness of CONTA on different baselines, and compare it to the state-of-the-arts. Further details and results are given in Appendix.

4.1 Settings

Datasets. PASCAL VOC 2012 [14] contains 21 classes (one background class) which includes 1,464, 1,449 and 1,456 images for *training*, *validation* (*val*) and *test*, respectively. As the common practice in [1, 63], in our experiments, we used an enlarged training set with 10,582 images, where the extra images and labels are from [19]. MS-COCO [35] contains 81 classes (one background class), 80k, and 40k images for *training* and *val*. Although pixel-level labels are provided in these benchmarks, we only used image-level class labels in the training process.

Evaluation Metric. We evaluated three types of masks: CAM seed area mask, pseudo-mask, and segmentation mask, compared with the ground-truth mask. The standard mean Intersection over Union (mIoU) was used on the *training* set for evaluating CAM seed area mask and pseudo-mask, and on the *val* and *test* sets for evaluating segmentation mask.

Baseline Models. To demonstrate the applicability of CONTA, we deployed it on four popular WSSS models including one seed area generation model: SEAM [63], and three seed area expansion models: IRNet [1], DSRG [22], and SEC [26]. Specially, DSRG requires the extra saliency mask [23] as the supervision. General architecture components include a multi-label image classification model, a pseudo-mask generation model, and a segmentation model: DeepLab-v2 [9]. Since the experimental settings of them are different, for fair comparison, we adopted the same settings as reported in the official codes. The detailed implementations of each baseline + CONTA are given in Appendix 3.

4.2 Ablation Study

Our ablation studies aim to answer the following questions. **Q1**: *Does CONTA merely take the advantage of the mask refinement? Is M_t indispensable?* We validated these by concatenating the segmentation mask (which is more refined compared to the pseudo-mask) with the backbone feature map, fed into classifiers. Then, we compared the newly generated results with the baseline ones. **Q2**: *How many rounds?* We recorded the performances of CONTA in each round. **Q3**: *Where to concatenate M_t?* We adopted the channel-wise feature map concatenation $[X, M_t]$ on different blocks of the backbone feature maps and tested which block has the most

Setting		CAM	Pseudo-Mask	Seg. Mask
	Upperbound [37]	–	–	80.8
	Baseline* [63]	55.1	63.1	64.3
(**Q1**)	$M_t \leftarrow$ Seg. Mask	55.0	62.7	64.0
(**Q2**)	Round = 1	55.6	64.2	65.0
	Round = 2	55.9	64.8	65.8
	Round = 3	**56.2**	**65.4**	**66.1**
	Round = 4	56.1	64.8	65.5
(**Q3**)	Block-2	55.5	64.3	65.2
	Block-3	55.6	64.5	65.3
	Block-4	56.0	65.1	65.9
	Block-5	**56.2**	**65.4**	**66.1**
	Dense	56.1	**65.4**	66.0
(**Q4**)	$C_{\text{Pseudo-Mask}}$	56.0	65.2	65.8
	$C_{\text{Seg. Mask}}$	**56.2**	**65.4**	**66.1**

Table 1: Ablation results on PASCAL VOC 2012 [14] in mIoU (%). "*" denotes our re-implemented results. "Seg. Mask" refers to the segmentation mask on the *val* set. "–" denotes that it is N.A. for the fully-supervised models.

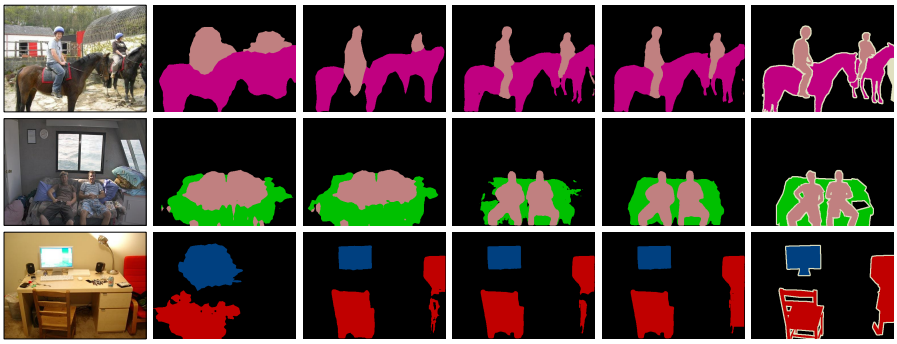

Figure 5: Visualization of pseudo-masks (baseline: SEAM [63], dataset: PASCAL VOC 2012 [14]).

improvement. **Q4**: *What is in the confounder set?* We compared the effectiveness of using the pseudo-mask and the segmentation mask to construct the confounder set C.

Due to page limit, we only showed ablation studies on the state-of-the-art WSSS model: SEAM [63], and the commonly used dataset – PASCAL VOC 2012; other methods on MS-COCO are given in Appendix 4. We treated the performance of the fully-supervised DeepLab-v2 [9] as the upperbound.

A1: Results in Table 1 (**Q1**) show that using the segmentation mask instead of the proposed M_t (concatenated to block-5) is even worse than the baseline. Therefore, the superiority of CONTA is not merely from better (smoothed) segmentation masks and M_t is empirically indispensable.

A2: Here, $[X, M_t]$ was applied to block-5, and the segmentation masks were used to establish the confounder set C. From Table 1 (**Q2**), we can observe that the performance starts to saturated at round 3. In particular, when round = 3, CONTA can achieve the unanimously best mIoU on CAM, pseudo-mask, and segmentation mask. Therefore, we set #round = 3 in the following CONTA experiments. We also visualized some qualitative results of the pseudo-masks in Figure 5. We can observe that CONTA can gradually segment clearer boundaries when compared to the baseline results, *e.g.*, person's leg vs. horse, person's body vs. sofa, chair's leg vs. background, and horse's leg vs. background.

A3: In addition to $[X, M_t]$ on various backbone blocks, we also reported a dense result, *i.e.*, $[X, M_t]$ on block-2 to block-5. In particular, $[X, M_t]$ was concatenated to the last layer of each block. Before the feature map concatenation, the map size of M_t should be down-sampled to match the corresponding block. Results in Table 1 (**Q3**) show that the performance at block-2/-3 are similar, and block-4/-5 are slightly higher. In particular, when compared to the baseline, block-5 has the most mIoU gain by 1.1% on CAM, 2.3% on pseudo-mask, and 1.8% on segmentation mask. One possible reason is that feature maps at block-5 contain higher-level contexts (*e.g.*, bigger parts, and more complete boundaries), which are more consistent with M_t, which are essential contexts. Therefore, we applied $[X, M_t]$ on block-5.

A4: From Table 1 (**Q4**), we can observe that using both of the pseudo-mask and the segmentation mask established C ($C_{\text{Pseudo-Mask}}$ and $C_{\text{Seg. Mask}}$) can boost the performance when compared to the baseline. In particular, the segmentation mask has a larger gain. The reason may be that the trained segmentation model can smooth the pseudo-mask and thus using higher-quality masks to approximate the unobserved confounder set is better.

Method	Backbone	CAM	Pseudo-Mask	Seg. Mask
SEC [26]	VGG-16	46.5	53.4	50.7
+ CONTA	VGG-16	47.9$_{+1.4}$	55.7$_{+2.3}$	53.2$_{+2.5}$
SEAM* [63]	ResNet-38	55.1	63.1	64.3
+ CONTA	ResNet-38	**56.2**$_{+1.1}$	65.4$_{+2.3}$	**66.1**$_{+1.8}$
IRNet* [1]	ResNet-50	48.3	65.9	63.0
+ CONTA	ResNet-50	48.8$_{+0.5}$	**67.9**$_{+2.0}$	65.3$_{+2.3}$
DSRG [22]	ResNet-101	47.3	62.7	61.4
+ CONTA	ResNet-101	48.0$_{+0.7}$	64.0$_{+1.3}$	62.8$_{+1.4}$

Table 2: Different baselines+CONTA on PASCAL VOC 2012 [14] dataset in mIoU (%). "*" denotes our re-implemented results. "Seg. Mask" refers to the segmentation mask on the *val* set.

4.3 Effectiveness on Different Baselines

To demonstrate the applicability of CONTA, in addition to SEAM [63], we also deployed CONTA on IRNet [1], DSRG [22], and SEC [26]. In particular, the round was set to 3 for SEAM, IRNet and

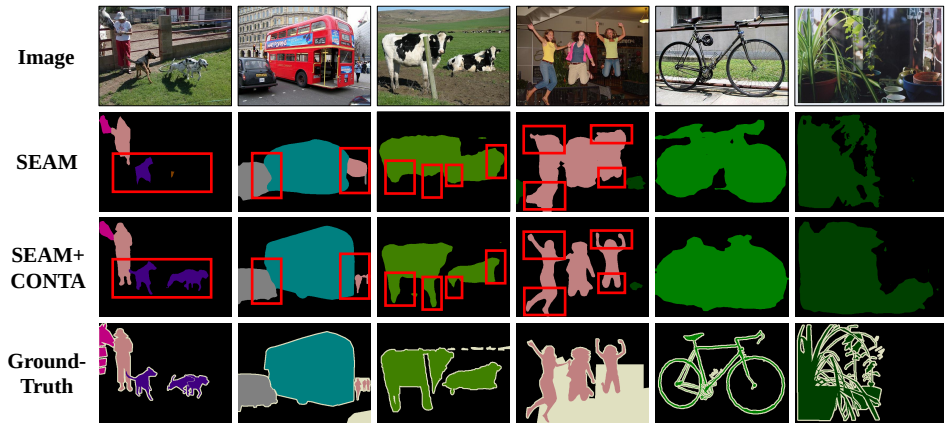

Figure 6: Visualization of segmentation masks, the last two columns show two failure cases (dataset: PASCAL VOC 2012 [14]). The red rectangle highlights the better areas for SEAM+CONTA.

SEC, and was set to 2 for DSRG. Experimental results on PASCAL VOC 2012 are shown in Table 2. We can observe that deploying CONTA on different WSSS models improve all their performances. There are the averaged mIoU improvements of 0.9% on CAM, 2.0% on pseudo-mask, and 2.0% on segmentation mask. In particular, CONTA deployed on SEAM can achieve the best performance of 56.2% on CAM and 66.1% on segmentation mask. Besides, CONTA deployed on IRNet can achieve the best performance of 67.9% on the pseudo-mask. The above results demonstrate the applicability and effectiveness of CONTA.

4.4 Comparison with State-of-the-arts

Table 3 lists the overall WSSS performances. On PASCAL VOC 2012, we can observe that CONTA deployed on IRNet with ResNet-50 [21] achieves the very competitive 65.3% and 66.1% mIoU on the *val* set and the *test* set. Based on a stronger backbone ResNet-38 [67] (with fewer layers but wider channels), CONTA on SEAM achieves state-of-the-art 66.1% and 66.7% mIoU on the *val* set and the *test* set, which surpasses the previous best model 1.2% and 1.0%, respectively. On MS-COCO, CONTA deployed on SEC with VGG-16 [54] achieves 23.7% mIoU on the *val* set, which surpasses the previous best model by 1.3% mIoU. Besides, on stronger backbones and WSSS models, CONTA can also boost the performance by 0.9% mIoU on average.

Method	Backbone	*val*	*test*
AffinityNet [2]	ResNet-38	61.7	63.7
RRM [73]	ResNet-38	62.6	62.9
SSDD [52]	ResNet-38	**64.9**	65.5
SEAM [63]	ResNet-38	64.5	**65.7**
IRNet [1]	ResNet-50	63.5	64.8
IRNet+CONTA	ResNet-50	65.3	66.1
SEAM+CONTA	ResNet-38	**66.1**	**66.7**

(a) PASCAL VOC 2012 [14].

Method	Backbone	*val*
BFBP [50]	VGG-16	20.4
SEC [26]	VGG-16	22.4
SEAM* [63]	ResNet-38	31.9
IRNet* [1]	ResNet-50	**32.6**
SEC+CONTA	VGG-16	23.7
SEAM+CONTA	ResNet-38	32.8
IRNet+CONTA	ResNet-50	**33.4**

(b) MS-COCO [35].

Table 3: Comparison with state-of-the-arts in mIoU (%). "*" denotes our re-implemented results. The **best** and **second best** performance under each set are marked with corresponding formats.

Figure 6 shows the qualitative segmentation mask comparisons between SEAM+CONTA and SEAM [63]. From the first four columns, we can observe that CONTA can make more accurate predictions on object location and boundary, *e.g.*, person's leg, dog, car, and cow's leg. Besides, we also show two failure cases of SEAM+CONTA in the last two columns, where bicycle and plant can not be well predicted. One possible explanation is that the segmentation mask is directly obtained from the 8× down-sampled feature maps, so some complex-contour objects can not be accurately delineated. This problem may be alleviated by using the encoder-decoder segmentation model, *e.g.*, SegNet [4], and U-Net [46]. More visualization results are given in Appendix 5.

5 Conclusion

We started from summarizing the three basic problems in existing pseudo-masks of WSSS. Then, we argued that the reasons are due to the context prior, which is a confounder in our proposed causal graph. Based on the graph, we used causal intervention to remove the confounder. As it is unobserved, we devised a novel WSSS framework: Context Adjustment (CONTA), based on the backdoor adjustment. CONTA can promote all the prevailing WSSS methods to the new state-of-the-arts. Thanks to the causal inference framework, we clearly know the limitations of CONTA: the approximation of the context confounder, which is proven to be ill-posed [11]. Therefore, as moving forward, we are going to 1) develop more advanced confounder set discovery methods and 2) incorporate observable expert knowledge into the confounder.

Acknowledgements

The authors would like to thank all the anonymous reviewers for their constructive comments and suggestions. This work was partially supported by the National Key Research and Development Program of China under Grant 2018AAA0102002, the National Natural Science Foundation of China under Grant 61732007, the China Scholarships Council under Grant 201806840058, the Alibaba Innovative Research (AIR) programme, and the NTU-Alibaba JRI.

Broader Impact

The positive impacts of this work are two-fold: 1) it improves the fairness of the weakly-supervised semantic segmentation model, which can prevent the potential discrimination of deep models, *e.g.*, an unfair AI could blindly cater to the majority, causing gender, racial or religious discrimination; 2) it allows some objects to be accurately segmented without extensive multi-context training images, *e.g.*, to segment a car on the road, by using our proposed method, we don't need to photograph any car under any context. The negative impacts could also happen when the proposed weakly-supervised semantic segmentation technique falls into the wrong hands, *e.g.*, it can be used to segment the minority groups for malicious purposes. Therefore, we have to make sure that the weakly-supervised semantic segmentation technique is used for the right purpose.

References

[1] Jiwoon Ahn, Sunghyun Cho, and Suha Kwak. Weakly supervised learning of instance segmentation with inter-pixel relations. In *CVPR*, 2019. 2, 3, 5, 6, 7, 8, 15, 17, 19

[2] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *CVPR*, 2018. 3, 5, 8, 15, 16

[3] Peter C Austin. An introduction to propensity score methods for reducing the effects of confounding in observational studies. *Multivariate Behavioral Research*, 46(3):399–424, 2011. 3

[4] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *TPAMI*, 39(12):2481–2495, 2017. 8

[5] Pierre Baldi and Peter Sadowski. The dropout learning algorithm. *Artificial Intelligence*, 210:78–122, 2014. 15

[6] Elias Bareinboim and Judea Pearl. Controlling selection bias in causal inference. In *Artificial Intelligence and Statistics*, 2012. 3

[7] Michel Besserve, Rémy Sun, and Bernhard Schölkopf. Counterfactuals uncover the modular structure of deep generative models. In *ICLR*, 2020. 3

[8] Thomas C Chalmers, Harry Smith Jr, Bradley Blackburn, Bernard Silverman, Biruta Schroeder, Dinah Reitman, and Alexander Ambroz. A method for assessing the quality of a randomized control trial. *Controlled Clinical Trials*, 2(1):31–49, 1981. 2

[9] 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. *TPAMI*, 40(4):834–848, 2017. 2, 5, 6, 7

[10] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*, 2015. 1

[11] Alexander D'Amour. On multi-cause causal inference with unobserved confounding: Counterexamples, impossibility, and alternatives. In *AISTATS*, 2019. 9

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 15, 16

[13] Nikita Dvornik, Julien Mairal, and Cordelia Schmid. Modeling visual context is key to augmenting object detection datasets. In *ECCV*, 2018. 2, 3

[14] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111(1):98–136, 2015. 1, 2, 3, 6, 7, 8, 17, 18, 21

[15] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 32(9):1627–1645, 2009. 4

[16] Ross Girshick. Fast r-cnn. In *ICCV*, 2015. 4

[17] Ross Girshick, Forrest Iandola, Trevor Darrell, and Jitendra Malik. Deformable part models are convolutional neural networks. In *CVPR*, 2015. 4

[18] Ruocheng Guo, Lu Cheng, Jundong Li, P Richard Hahn, and Huan Liu. A survey of learning causality with data: Problems and methods. *CSUR*, 53(4):1–37, 2020. 3

[19] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *ICCV*, 2011. 6

[20] Mohammad Havaei, Axel Davy, David Warde-Farley, Antoine Biard, Aaron Courville, Yoshua Bengio, Chris Pal, Pierre-Marc Jodoin, and Hugo Larochelle. Brain tumor segmentation with deep neural networks. *Medical Image Analysis*, 35(3):18–31, 2017. 1

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 8, 16

[22] Zilong Huang, Xinggang Wang, Jiasi Wang, Wenyu Liu, and Jingdong Wang. Weakly-supervised semantic segmentation network with deep seeded region growing. In *CVPR*, 2018. 1, 2, 3, 6, 7, 15, 17, 18, 20

[23] Huaizu Jiang, Jingdong Wang, Zejian Yuan, Yang Wu, Nanning Zheng, and Shipeng Li. Salient object detection: A discriminative regional feature integration approach. In *CVPR*, 2013. 6, 16

[24] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *CVPR*, 2018. 4

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 15

[26] Alexander Kolesnikov and Christoph H Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *ECCV*, 2016. 1, 3, 5, 6, 7, 8, 15, 17, 18, 20

[27] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NeurIPS*, 2011. 15, 16, 17

[28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 15, 16

[29] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. 4

[30] Jungbeom Lee, Eunji Kim, Sungmin Lee, Jangho Lee, and Sungroh Yoon. Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference. In *CVPR*, 2019. 3

[31] Qizhu Li, Anurag Arnab, and Philip HS Torr. Weakly-and semi-supervised panoptic segmentation. In *ECCV*, 2018. 1

[32] Yanwei Li, Lin Song, Yukang Chen, Zeming Li, Xiangyu Zhang, Xingang Wang, and Jian Sun. Learning dynamic routing for semantic segmentation. In *CVPR*, 2020. 5

[33] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *CVPR*, 2016. 1

[34] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. In *ICLR*, 2014. 5

[35] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 3, 6, 8, 17, 19, 20

[36] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. 4

[37] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1, 6, 17, 18, 19, 20

[38] David Marr. Vision: A computational investigation into the human representation and processing of visual information. *MIT Press*, 1982. 4, 5

[39] Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. Counterfactual vqa: A cause-effect look at language bias. In *arXiv*, 2020. 3

[40] Giambattista Parascandolo, Niki Kilbertus, Mateo Rojas-Carulla, and Bernhard Schölkopf. Learning independent causal mechanisms. In *ICML*, 2018. 3

[41] Judea Pearl. *Causality: Models, Reasoning and Inference*. Springer, 2000. 2, 3, 14

[42] Judea Pearl. Interpretation and identification of causal mediation. *Psychological Methods*, 19(4):459–481, 2014. 2, 3

[43] Judea Pearl et al. Causal inference in statistics: An overview. *Statistics surveys*, 3:96–146, 2009. 3

[44] Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016. 2, 3, 4, 14

[45] Jiaxin Qi, Yulei Niu, Jianqiang Huang, and Hanwang Zhang. Two causal principles for improving visual dialog. In *CVPR*, 2020. 3

[46] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 8

[47] Donald B Rubin. Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American Statistical Association*, 100(469):322–331, 2005. 3

[48] Donald B Rubin. Essential concepts of causal inference: a remarkable history and an intriguing future. *Biostatistics & Epidemiology*, 3(1):140–155, 2019. 3

[49] Ethan M Rudd, Manuel Günther, and Terrance E Boult. Moon: A mixed objective optimization network for the recognition of facial attributes. In *ECCV*, 2016. 5

[50] Fatemehsadat Saleh, Mohammad Sadegh Aliakbarian, Mathieu Salzmann, Lars Petersson, Stephen Gould, and Jose M Alvarez. Built-in foreground/background prior for weakly-supervised semantic segmentation. In *ECCV*, 2016. 3, 8

[51] Bernhard Schölkopf, Dominik Janzing, Jonas Peters, Eleni Sgouritsa, Kun Zhang, and Joris Mooij. On causal and anticausal learning. In *ICML*, 2012. 3

[52] Wataru Shimoda and Keiji Yanai. Self-supervised difference detection for weakly-supervised semantic segmentation. In *ICCV*, 2019. 8

[53] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *CVPR*, 2016. 15

[54] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 8, 16

[55] Guolei Sun, Wenguan Wang, Jifeng Dai, and Luc Van Gool. Mining cross-image semantics for weakly supervised semantic segmentation. In *ECCV*, 2020. 3

[56] Raphael Suter, Djordje Miladinovic, Bernhard Schölkopf, and Stefan Bauer. Robustly disentangled causal mechanisms: Validating deep representations for interventional robustness. In *ICML*, 2019. 3

[57] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. In *NeurIPS*, 2020. 3

[58] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In *CVPR*, 2020. 3

[59] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. In *CVPR*, 2019. 3

[60] Michael Treml, José Arjona-Medina, Thomas Unterthiner, Rupesh Durgesh, Felix Friedmann, Peter Schuberth, Andreas Mayr, Martin Heusel, Markus Hofmarcher, Michael Widrich, et al. Speeding up semantic segmentation for autonomous driving. In *NeurIPS*, 2016. 1

[61] Paul Vernaza and Manmohan Chandraker. Learning random-walk label propagation for weakly-supervised semantic segmentation. In *CVPR*, 2017. 3

[62] Tan Wang, Jianqiang Huang, Hanwang Zhang, and Qianru Sun. Visual commonsense r-cnn. In *CVPR*, 2020. 3, 6

[63] Yude Wang, Jie Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In *CVPR*, 2020. 1, 2, 3, 5, 6, 7, 8, 15, 17, 19, 21

[64] Yunchao Wei, Jiashi Feng, Xiaodan Liang, Ming-Ming Cheng, Yao Zhao, and Shuicheng Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *CVPR*, 2017. 3

[65] Yunchao Wei, Huaxin Xiao, Honghui Shi, Zequn Jie, Jiashi Feng, and Thomas S Huang. Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation. In *CVPR*, 2018. 1, 3

[66] CF Jeff Wu. On the convergence properties of the em algorithm. *The Annals of Statistics*, 1(1):95–103, 1983. 5

[67] Zifeng Wu, Chunhua Shen, and Anton Van Den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *Pattern Recognition*, 90(1):119–133, 2019. 8, 15

[68] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 5, 15

[69] Xu Yang, Hanwang Zhang, and Jianfei Cai. Deconfounded image captioning: A causal retrospect. In *arXiv*, 2020. 3

[70] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 3, 15, 16

[71] Zhongqi Yue, Hanwang Zhang, Qianru Sun, and Xiansheng Hua. Interventional few-shot learning. In *NeurIPS*, 2020. 3

[72] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *CVPR*, 2018. 4

[73] Bingfeng Zhang, Jimin Xiao, Yunchao Wei, Mingjie Sun, and Kaizhu Huang. Reliability does matter: An end-to-end weakly supervised semantic segmentation approach. In *AAAI*, 2020. 8

[74] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016. 1, 4, 5, 16, 17

Appendix for "Causal Intervention for Weakly-Supervised Semantic Segmentation"

This appendix includes the derivation of backdoor adjustment for the proposed structural causal model (Section 1), the normalized weighted geometric mean (Section 2), the detailed implementations for different baseline models (Section 3), the supplementary ablation studies (Section 4), and more visualization results of segmentation masks (Section 5).

1 Derivation of Backdoor Adjustment for the Proposed Causal Graph

In the main paper, we used backdoor adjustment [44] to perform the causal intervention. In this section, we show the derivation of backdoor adjustment for the proposed causal graph (in Figure 3(b) of the main paper), by leveraging the following three do-calculus rules [41].

Given an arbitrary causal directed acyclic graph $\mathcal{G}$, there are four nodes respectively represented by X, Y, Z, and W. Particularly, $\mathcal{G}_{\overline{X}}$ denotes the intervened causal graph where all *incoming* arrows to X are deleted, and $\mathcal{G}_{\underline{X}}$ denotes another intervened causal graph where all *outgoing* arrows from X are deleted. We use the lower cases x, y, z, and w to represent the respective values of nodes: $X = x$, $Y = y$, $Z = z$, and $W = w$. For any interventional distribution compatible with $\mathcal{G}$, we have the following three rules:

Rule 1. Insertion/deletion of observations:

$$P(y|do(x), z, w) = P(y|do(x), w), \text{if } (Y \perp\!\!\!\perp Z|X, W)_{\mathcal{G}_{\overline{X}}}. \tag{A1}$$

Rule 2. Action/observation exchange:

$$P(y|do(x), do(z), w) = P(y|do(x), z, w), \text{if } (Y \perp\!\!\!\perp Z|X, W)_{\mathcal{G}_{\overline{X}\underline{Z}}}. \tag{A2}$$

Rule 3. Insertion/deletion of actions:

$$P(y|do(x), do(z), w) = P(y|do(x), w), \text{if } (Y \perp\!\!\!\perp Z|X, W)_{\mathcal{G}_{\overline{X}\overline{Z(W)}}}, \tag{A3}$$

where $Z(W)$ is a subset of Z that are not ancestors of any specific nodes related to W in $\mathcal{G}_{\underline{X}}$. Based on these three rules, we can derive the interventional distribution $P(Y|do(X))$ for our proposed causal graph (in Figure 3(b) of the main paper) by:

$$\begin{aligned}
P(Y|do(X)) &= \sum_c P(Y|do(X), c)P(c|do(X)) && \text{(A4)}\\
&= \sum_c P(Y|do(X), c)P(c) && \text{(A5)}\\
&= \sum_c P(Y|X, c)P(c) && \text{(A6)}\\
&= \sum_c P(Y|X, c, M)P(M|X, c)P(c) && \text{(A7)}\\
&= \sum_c P(Y|X, c, M = f(X, c))P(c) && \text{(A8)}\\
&= \sum_c P(Y|X, M = f(X, c))P(c), && \text{(A9)}
\end{aligned}$$

where Eq. A4 and Eq. A7 follow the law of total probability. We can obtain Eq. A5 via **Rule 3** that given $c \perp\!\!\!\perp X$ in $\mathcal{G}_{\overline{X}}$, and Eq. A6 can be obtained via **Rule 2** which changes the intervention term into observation as $Y \perp\!\!\!\perp X|c$ in $\mathcal{G}_{\underline{X}}$. Eq. A8 is because in our causal graph, M is an image-specific context representation given by the function $f(X, c)$, and Eq. A9 is essentially equal to Eq. A8.

2 Normalized Weighted Geometric Mean

This is Appendix to Section 3.2 "Step 4. Computing M_{t+1}". In Section 3.2 of the main paper, we used the Normalized Weighted Geometric Mean (NWGM) [68] to move the outer sum $\sum_c P(\cdot)$ into the feature level: $\sum_c P(Y|X,M)P(c) \approx P(Y|X,M = \sum_c f(X,c)P(c))$. Here, we show the detailed derivation. Formally, our implementation for the positive term (*i.e.*, $\mathbb{1}_{i\in Y}$ in Eq.(2) of the main paper) can be derived by:

$$
\begin{aligned}
P(Y|do(X)) &= \sum_c \frac{exp(s_1(c))}{exp(s_1(c)) + exp(s_2(c))} P(c) && \text{(A10)}\\
&= \sum_c Softmax(s_1(c))P(c) && \text{(A11)}\\
&\approx \text{NWGM}(Softmax(s_1(c))) && \text{(A12)}\\
&= \frac{\prod_c [exp(s_1(c)]^{P(c)}}{\prod_c [exp(s_1(c)]^{P(c)} + \prod_c [exp(s_2(c)]^{P(c)}} && \text{(A13)}\\
&= \frac{exp(\sum_c (s_1(c)P(c)))}{exp(\sum_c (s_1(c)P(c))) + exp(\sum_c (s_2(c)P(c)))} && \text{(A14)}\\
&= \frac{exp(\mathbb{E}_c(s_1(c)))}{exp(\mathbb{E}_c(s_1(c))) + exp(\mathbb{E}_c(s_2(c)))} && \text{(A15)}\\
&= Softmax(\mathbb{E}_c(s_1(c)), && \text{(A16)}
\end{aligned}
$$

where $s_1(\cdot)$ denotes the positive predicted score for the class label which is indeed associated with the input image, and $s_2(c) = 0$ under this condition. We can obtain Eq. A10 via our implementation of the multi-label image classification model, and obtain Eq. A11 and Eq. A16 via the definition of the softmax function. Eq. A12 can be obtained via the results in [5]. Eq. A13 to Eq. A15 follow the derivation in [68]. Since $s_1(\cdot)$ in our implementation is a linear model, we can use Eq.(3) in the main paper to compute M_{t+1}. In addition to the positive term, we can also obtain derivation for the negative term (*i.e.*, $\mathbb{1}_{i\notin Y}$ in Eq.(2) of the main paper) through the similar process as above.

3 More Implementation Details

This is Appendix to Section 4.1 "Settings". In Section 4.1 of the main paper, we deployed CONTA on four popular WSSS models including SEAM [63], IRNet [1], DSRG [22], and SEC [26]. In this section, we show the detailed implementations of these four models.

3.1 Implementation of SEAM+CONTA

Backbone. ResNet-38 [67] was adopted as the backbone network. It was pre-trained on ImageNet [12] and its convolution layers of the last three blocks were replaced by dilated convolutions [70] with a common input stride of 1 and their dilation rates were adjusted, such that the backbone network can return a feature map of stride 8, *i.e.*, the output size of the backbone network was $1/8$ of the input.

Setting. The input images were randomly re-scaled in the range of $[448, 768]$ by the longest edge and then cropped into a fix size of 448×448 using zero padding if needed.

Training Details. The initial learning rate was set to 0.01, following the poly policy $lr_{init} = lr_{init}(1 - itr/max\_itr)^{\rho}$ with $\rho = 0.9$ for decay. Online hard example mining [53] was employed on the training loss to preserve only the top 20% pixel losses. The model was trained with batch size as 8 for 8 epochs using Adam optimizer [25]. We deployed the same data augmentation strategy (*i.e.*, horizontal flip, random cropping, and color jittering [28]), as in AffinityNet [2], in our training process.

Hyper-parameters. The hard threshold parameter for CAM was set to 16 by default and changed to 4 and 24 to amplify and weaken background activation, respectively. The fully-connected CRF [27]

was used to refine CAM, pseudo-mask, and segmentation mask with the default parameters in the public code. For seed areas expansion, the AffinityNet [2] was used with the search radius as $\gamma = 5$, the hyper-parameter in the Hadamard power of the affinity matrix as $\beta = 8$, and the number of iterations in random walk as $t = 256$.

3.2 Implementation of IRNet+CONTA

Backbone. ResNet-50 [21] was used as the backbone network (pre-trained on ImageNet [12]). The adjusted dilated convolutions [70] were used in the last two blocks with a common input stride of 1, such that the backbone network can return a feature map of stride 16, *i.e.*, the output size of the backbone network was $1/16$ of the input.

Setting. The input image was cropped into a fix size of 512×512 using zero padding if needed.

Training Details. The stochastic gradient descent was used for optimization with $8,000$ iterations. Learning rate was initially set to 0.1, and decreased using polynomial decay $lr_{init} = lr_{init}(1 - itr/max\_itr)^{\rho}$ with $\rho = 0.9$ at every iteration. The batch size was set to 16 for the image classification model and 32 for the inter-pixel relation model. The same data augmentation strategy (*i.e.*, horizontal flip, random cropping, and color jittering [28]) as in AffinityNet [2] was used in the training process.

Hyper-parameters. The fully-connected CRF [27] was used to refine CAM, pseudo-mask, and segmentation mask with the default parameters given in the original code. The hard threshold parameter for CAM was set to 16 by default and changed to 4 and 24 to amplify and weaken the background activation, respectively. The radius γ that limits the search space of pairs was set to 10 when training, and reduced to 5 at inference (conservative propagation in inference). The number of random walk iterations t was fixed to 256. The hyper-parameter β in the Hadamard power of the affinity matrix was set to 10.

3.3 Implementation of DSRG+CONTA

Backbone. ResNet-101 [21] was used as the backbone network (pre-trained on ImageNet [12]) where dilated convolutions [70] were used in the last two blocks, such that the backbone network can return a feature map of stride 16, *i.e.*, the output size of the backbone network was $1/16$ of the input.

Setting. The input image was cropped into a fix size of 321×321 using zero padding if needed.

Training Details. The stochastic gradient descent with mini-batch was used for network optimization with $10,000$ iterations. The momentum and the weight decay were set to 0.9 and 0.0005, respectively. The batch size was set to 20, and the dropout rate was set to 0.5. The initial learning rate was set to 0.0005 and it was decreased by a factor of 10 every $2,000$ iterations.

Hyper-parameters. For seed generation, pixels with the top 20% activation values in the CAM were considered as foreground (objects) as in [74]. For saliency masks, the model in [23] was used to produce the background localization cues with the normalized saliency value 0.06. For the similarity criteria, the foreground threshold and the background threshold were set to 0.99 and 0.85, respectively. The fully-connected CRF [27] was used to refine pseudo-mask and segmentation mask with the default parameters in the public code.

3.4 Implementation of SEC+CONTA

Backbone. VGG-16 [54] was used as the backbone network (pre-trained on ImageNet [12]), where the last two fully-connected layers were substituted with randomly initialized convolutional layers, which have 1024 output channels and kernels of size 3, such that the output size of the backbone network was $1/8$ of the input.

Setting. The input image was cropped into a fix size of 321×321 using zero padding if needed.

Training Details. The weights for the last (prediction) layer were randomly initialized from a normal distribution with mean 0 and variance 0.01. The stochastic gradient descent was used for the network optimization with $8,000$ iterations, the batch size was set to 15, the dropout rate was set to 0.5 and the

Setting		CAM	Pseudo-Mask	Seg. Mask
	Upperbound [37]	–	–	72.3
	Baseline* [1]	48.3	65.9	63.0
(**A1**)	$M_t \leftarrow$ Seg. Mask	48.1	65.5	62.1
(**A2**)	Round = 1	48.5	66.9	64.2
	Round = 2	48.7	67.6	65.0
	Round = 3	**48.8**	**67.9**	**65.3**
	Round = 4	48.6	67.2	64.9
(**A3**)	Block-2	48.3	66.2	63.4
	Block-3	48.4	66.6	63.8
	Block-4	48.7	67.3	64.6
	Block-5	**48.8**	**67.9**	**65.3**
	Dense	48.7	67.6	65.1
(**A4**)	$C_{\text{Pseudo-Mask}}$	48.6	67.4	65.0
	$C_{\text{Seg. Mask}}$	**48.8**	**67.9**	**65.3**

Table A1: Ablations of IRNet [1]+CONTA on PASCAL VOC 2012 [14] in mIoU (%). "*" denotes our re-implemented results. "Seg. Mask" refers to the segmentation mask of the *val* set. "–" denotes that the result is N.A. for the fully-supervised model.

weight decay parameter was set to 0.0005. The initial learning rate was 0.001 and it was decreased by a factor of 10 every $2,000$ iterations.

Hyper-parameters. For seed generation, pixels with the top 20% activation values in the CAM were considered as foreground (objects) as in [74]. The fully-connected CRF [27] was used to refine pseudo-mask and segmentation mask with the spatial distance was multiplied by 12 to reflect the fact that the original image was down-scaled to match the size of the predicted segmentation mask, and the other parameters are consistent with the public code.

4 More Ablation Study Results

This is Appendix to Section 4.2 "Ablation Study". In Section 4.2 of the main paper, we showed the ablation study results of SEAM [63]+CONTA on PASCAL VOC 2012 [14]. In this section, we show the results of IRNet [1]+CONTA, DSRG [22]+CONTA, and SEC [26]+CONTA on PASCAL VOC 2012. Besides, we also show the results of SEAM+CONTA, IRNet+CONTA, DSRG+CONTA, and SEC+CONTA on MS-COCO [35].

4.1 PASCAL VOC 2012

Table A1, Table A2, and Table A3 show ablation results of IRNet+CONTA, DSRG+CONTA, and SEC+CONTA on PASCAL VOC 2012, respectively. We can observe that IRNet+CONTA and SEC+CONTA can achieve the best performance at round= 3, and DSRG+CONTA can achieve the best mIoU score at round= 2. In addition to results of SEAM+CONTA in our main paper, we can see that IRNet+CONTA can achieve the second best mIoU results: 48.8% on CAM, 67.9% on pseudo-mask, and 65.3% on segmentation mask.

4.2 MS-COCO

Table A4, Table A5, Table A6, and Table A7 show the respective ablation results of SEAM+CONTA, IRNet+CONTA, DSRG+CONTA, and SEC+CONTA on MS-COCO. We can see that SEAM+CONTA, IRNet+CONTA and, SEC+CONTA can achieve the top mIoU at round= 3, and DSRG+CONTA can achieve the best performance at round= 2. In particular, we see that the

Setting		CAM	Pseudo-Mask	Seg. Mask
	Upperbound [37]	–	–	77.7
	Baseline* [22]	47.3	62.7	61.4
(**A1**)	$M_t \leftarrow$ Seg. Mask	47.0	61.9	61.1
(**A2**)	Round = 1	47.7	63.5	62.2
	Round = 2	**48.0**	**64.0**	**62.8**
	Round = 3	47.8	63.8	62.5
	Round = 4	47.4	63.5	62.1
(**A3**)	Block-2	47.4	62.9	61.7
	Block-3	47.6	63.2	62.1
	Block-4	47.9	63.7	62.6
	Block-5	**48.0**	**64.0**	**62.8**
	Dense	47.8	63.8	62.7
(**A4**)	$C_{\text{Pseudo-Mask}}$	47.8	63.6	62.5
	$C_{\text{Seg. Mask}}$	**48.0**	**64.0**	**62.8**

Table A2: Ablations of DSRG [22]+CONTA on PASCAL VOC 2012 [14] in mIoU (%). "*" denotes our re-implemented results. "Seg. Mask" refers to the segmentation mask of the *val* set. "–" denotes that the result is N.A. for the fully-supervised model.

Setting		CAM	Pseudo-Mask	Seg. Mask
	Upperbound [37]	–	–	71.6
	Baseline* [26]	46.5	53.4	50.7
(**A1**)	$M_t \leftarrow$ Seg. Mask	46.4	53.1	50.3
(**A2**)	Round = 1	47.1	54.3	51.7
	Round = 2	47.6	55.1	52.6
	Round = 3	**47.9**	**55.7**	**53.2**
	Round = 4	47.7	55.6	53.0
(**A3**)	Block-2	46.8	53.9	51.2
	Block-3	47.1	54.5	51.5
	Block-4	47.6	55.1	52.4
	Block-5	**47.9**	**55.7**	**53.2**
	Dense	47.8	55.6	53.0
(**A4**)	$C_{\text{Pseudo-Mask}}$	47.7	55.3	52.9
	$C_{\text{Seg. Mask}}$	**47.9**	**55.7**	**53.2**

Table A3: Ablations of SEC [26]+CONTA on PASCAL VOC 2012 [14] in mIoU (%). "*" denotes our re-implemented results. "Seg. Mask" refers to the segmentation mask of the *val* set. "–" denotes that the result is N.A. for the fully-supervised model.

mIoU scores of IRNet+CONTA are the best on MS-COCO as respectively 28.7% on CAM, 35.2% on pseudo-mask, and 33.4% on segmentation mask.

5 More Visualizations

This is Appendix to Section 4.4 "Comparison with State-of-the-arts". More segmentation results are visualized in Figure A1. We can observe that most of our resulting masks are of high quality. The

Setting		CAM	Pseudo-Mask	Seg. Mask
	Upperbound* [37]	–	–	44.8
	Baseline* [63]	25.1	31.5	31.9
(**A1**)	$M_t \leftarrow$ Seg. Mask	24.8	31.1	31.4
(**A2**)	Round = 1	25.7	31.9	32.4
	Round = 2	26.2	32.2	32.7
	Round = 3	**26.5**	**32.5**	**32.8**
	Round = 4	26.3	32.1	32.6
(**A3**)	Block-2	25.7	32.0	32.3
	Block-3	25.9	32.1	32.4
	Block-4	26.3	32.4	32.6
	Block-5	**26.5**	**32.5**	**32.8**
	Dense	**26.5**	32.4	32.5
(**A4**)	$C_{\text{Pseudo-Mask}}$	26.4	32.0	32.6
	$C_{\text{Seg. Mask}}$	**26.5**	**32.5**	**32.8**

Table A4: Ablation results of SEAM [63]+CONTA on MS-COCO [35] in mIoU (%). "*" denotes our re-implemented results. "Seg. Mask" refers to the segmentation mask of the *val* set. "–" denotes that the result is N.A. for the fully-supervised model.

Setting		CAM	Pseudo-Mask	Seg. Mask
	Upperbound* [37]	–	–	42.5
	Baseline* [1]	27.4	34.0	32.6
(**A1**)	$M_t \leftarrow$ Seg. Mask	27.1	33.5	32.3
(**A2**)	Round = 1	28.0	34.3	32.9
	Round = 2	28.4	34.8	33.2
	Round = 3	**28.7**	**35.2**	**33.4**
	Round = 4	28.5	35.0	33.2
(**A3**)	Block-2	27.7	34.3	32.8
	Block-3	27.9	34.5	32.9
	Block-4	28.4	34.9	33.2
	Block-5	**28.7**	**35.2**	**33.4**
	Dense	28.6	**35.2**	33.1
(**A4**)	$C_{\text{Pseudo-Mask}}$	28.5	35.0	33.2
	$C_{\text{Seg. Mask}}$	**28.7**	**35.2**	**33.4**

Table A5: Ablation results of IRNet [1]+CONTA on MS-COCO [35] in mIoU (%). "*" denotes our re-implemented results. "Seg. Mask" refers to the segmentation mask of the *val* set. "–" denotes that the result is N.A. for the fully-supervised model.

segmentation masks predicted by SEAM+CONTA are more accurate and have better integrity, *e.g.*, for cow, horse, bird, person lying next to the dog, and person standing next to the cows. In particular, SEAM+CONTA works better to prediction the edges of some thin objects or object parts, *e.g.*, the tail (or the head) of bird, car, and person in the car.

Setting		CAM	Pseudo-Mask	Seg. Mask
	Upperbound [37]	–	–	45.0
	Baseline* [22]	19.8	26.1	25.6
(**A1**)	$M_t \leftarrow$ Seg. Mask	19.5	25.9	25.5
(**A2**)	Round = 1	20.5	26.9	26.1
	Round = 2	**20.9**	**27.5**	**26.4**
	Round = 3	20.7	27.2	26.2
	Round = 4	20.4	26.9	26.0
(**A3**)	Block-2	20.1	26.8	25.9
	Block-3	20.2	27.0	26.0
	Block-4	20.5	27.2	26.2
	Block-5	**20.9**	**27.5**	**26.4**
	Dense	20.8	27.3	26.1
(**A4**)	$C_{\text{Pseudo-Mask}}$	20.7	27.2	26.1
	$C_{\text{Seg. Mask}}$	**20.9**	**27.5**	**26.4**

Table A6: Ablation results of DSRG [22]+CONTA on MS-COCO [35] in mIoU (%). "*" denotes our re-implemented results. "Seg. Mask" refers to the segmentation mask of the *val* set. "–" denotes that the result is N.A. for the fully-supervised model.

Setting		CAM	Pseudo-Mask	Seg. Mask
	Upperbound [37]	–	–	41.0
	Baseline* [26]	18.7	24.0	22.4
(**A1**)	$M_t \leftarrow$ Seg. Mask	18.1	23.5	21.2
(**A2**)	Round = 1	20.1	24.4	23.0
	Round = 2	21.2	24.7	23.4
	Round = 3	**21.8**	**24.9**	**23.7**
	Round = 4	21.4	24.5	23.5
(**A3**)	Block-2	19.5	24.2	22.7
	Block-3	19.9	24.4	22.9
	Block-4	20.6	24.7	23.5
	Block-5	**21.8**	**24.9**	**23.7**
	Dense	**21.8**	24.6	23.5
(**A4**)	$C_{\text{Pseudo-Mask}}$	21.5	24.7	23.4
	$C_{\text{Seg. Mask}}$	**21.8**	**24.9**	**23.7**

Table A7: Ablation results of SEC [26]+CONTA on MS-COCO [35] in mIoU (%). "*" denotes our re-implemented results. "Seg. Mask" refers to the segmentation mask of the *val* set. "–" denotes that the result is N.A. for the fully-supervised model.

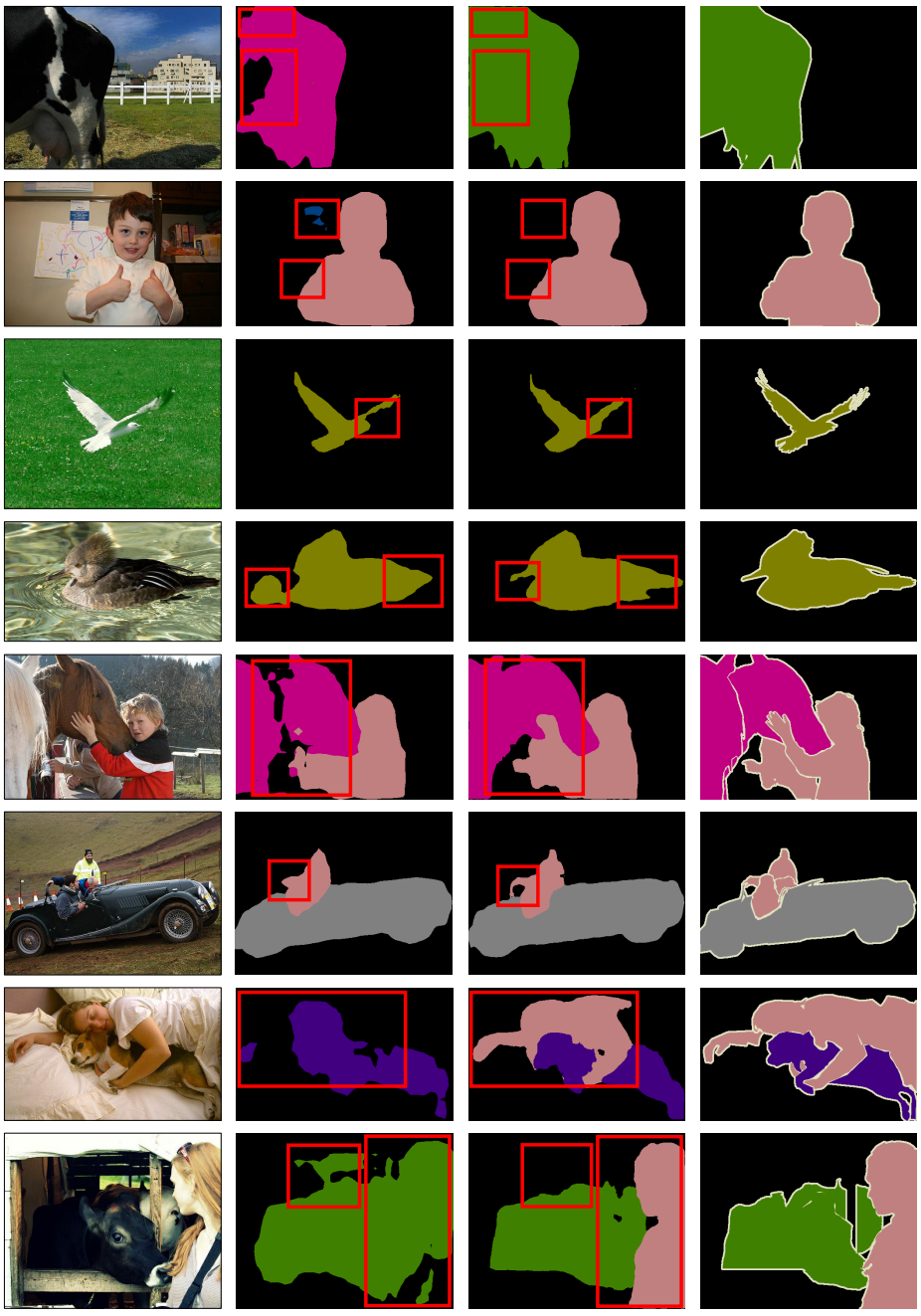

Figure A1: More visualization results. Samples are from PASCAL VOC 2012 [14]. Red rectangles highlight the improved regions predicted by SEAM [63]+CONTA.