Approximate Size Targets Are Sufficient for Accurate Semantic Segmentation

Anonymous Author(s)
Affiliation
Address
email

Abstract

We propose a new general form of image-level supervision for semantic segmentation based on approximate targets for the relative size of segments. At each training image, such targets are represented by a categorical distribution for the "expected" average prediction over the image pixels. We motivate the zero-avoiding variant of KL divergence as a general training loss for any segmentation architecture leading to quality on par with the full pixel-level supervision. However, our image-level supervision is significantly less expensive, it needs to know only an approximate fraction of an image occupied by each class. Such estimates are easy for a human annotator compared to pixel-accurate labeling. Our loss shows significant robustness to size target errors, which may even improve the generalization quality. The proposed size targets can be seen as an extension of the standard class tags, which correspond to non-zero size targets in each image. Using only a minimal amount of extra information, our supervision improves and simplifies the training. It works on standard segmentation architectures as is, unlike tag-based methods requiring complex specialized modifications and multi-stage training.

1 Introduction

Our image-level supervision approach applies to any semantic segmentation model and does not require any modification. It can be technically described in one paragraph, as follows. Soft-max prediction $S_p = (S_p^1, \ldots, S_p^K)$ at any pixel p is a categorical distribution over K classes, including background. At any image, the average prediction over all image pixels, denoted by set Ω, is

$$\bar{S} := \frac{1}{|\Omega|} \sum_{p \in \Omega} S_p \tag{1}$$

where $\bar{S} = (\bar{S}^1, \ldots, \bar{S}^K)$ is also a categorical distribution over K classes. It is an image-level prediction of the relative or normalized sizes (volume, area, or cardinality) of the objects in the image. We assume that training images have approximate size targets represented by categorical distributions $v = (v_k)_{k=1}^K$, e.g. $v = (0, .15, 0, \ldots, 0, .75)$ for the middle image in Fig. 1 if *"bird"* is the second class and *"background"* is the last. This representation also applies to multi-label images. For each training image, our *size-target loss*

$$L_{size} \;\; = \;\; KL(v \| \bar{S}) \;\; = \;\; \sum_k v_k \ln \frac{v_k}{\bar{S}^k} \tag{2}$$

is based on Kullback–Leibler (KL) divergence. Figure 2(b) shows some results for a generic segmentation network (ResNet101 [4] backbone) trained on PASCAL [5] using only image-level supervision with approximate size targets (8% mean relative errors). Our total loss is very simple: it combines size-target loss (2) and a common CRF loss (3) [6].

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

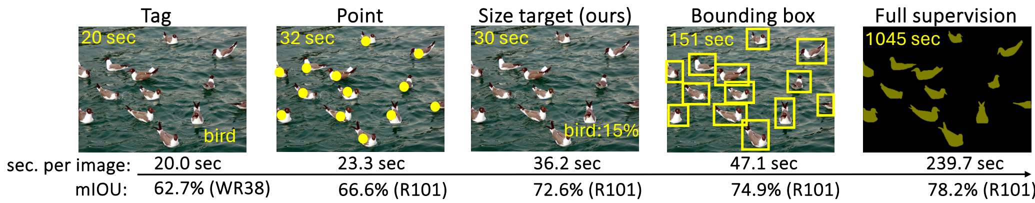

Figure 1: Supervision types for segmentation: labeling speed and accuracy on PASCAL. The top-left corner of each image shows its estimated labeling time based on observed instances. The table shows per-image labeling times averaged over the data and mean Intersection-over-Union (mIoU) for comparable end-to-end methods with similar ResNet backbones (ResNet101 or WideResNet38 [1]), for fairness. We obtained mIoU scores, except for the "tag" and "box" scores from [2] and [3]. Our supplemental materials detail evaluation of the labeling times and mIoU. For completeness, Tab.2 includes more complex architectures and multi-stage systems, e.g. for tags. This paper focuses on standard segmentation architectures for size supervision.

1.1 Overview of weakly-supervised segmentation

By *weakly-supervised* semantic segmentation we refer to all methods that do not use full pixel-precise ground truth (GT) masks for training. Such full supervision is overwhelmingly expensive for segmentation and is unrealistic for many practical purposes, see the right image in Fig. 1. There are many forms of weak supervision for semantic segmentation, e.g. based on partial pixel-level ground truth defined by "seeds" [6, 7], boxes [3], or image-level class-tags [2, 8, 9], see Fig. 1. It is also common to incorporate self-supervision based on various augmentation ideas and contrastive losses [10–12].

Lack of supervision also motivates unsupervised loss functions such as standard old-school regularization objectives for *low-level* segmentation or clustering. For example, many methods [13, 14, 12] use variants of K-means objective (squared errors) enforcing the compactness of each class representation. It is also very common to use CRF-based pairwise loss functions [6, 7] that encourage segment shape regularity and alignment to intensity contrast edges in each image [15]. The last point addresses the well-known limitation of standard segmentation networks that often output low-resolution segments. Intensity contrast edges on the high-resolution input image is a good low-level cue of an object boundary and it can improve the details and localization of the semantic segments.

Conditional or Markov random fields (CRF or MRF) are common basic examples of pairwise graphical models. The corresponding unsupervised loss functions can be formulated for continuous soft-max predictions S_p produced by segmentation networks, e.g. [6, 7, 9]. Thus, it is natural to use relaxations of the standard discrete CRF/MRF models, such as *Potts* [16] or its *dense-CRF* version [17]. We use a bilinear relaxation of the general Potts model

$$L_{crf}(S) \quad = \quad \sum_k (\mathbf{1} - S^k)^\top W S^k \tag{3}$$

where $S := (S_p \,|\, p \in \Omega)$ is a field of all pixel-level soft-max predictions S_p in a given image, and $S^k := (S_p^k \,|\, p \in \Omega)$ is a vector of all pixel predictions specifically for class k. Matrix $W = [w_{pq}]$ typically represents some given non-negative affinities w_{pq} between pairs of pixels $p, q \in \Omega$. It is easy to interpret loss (3) assuming, for simplicity, that all pixels have confident *one-hot* predictions S_p so that each S^k is a binary indicator vector for segment k. The loss sums all weights w_{pq} between the pixels in different segments. Thus, the weights are interpreted as discontinuity penalties. The loss minimizes the discontinuity costs [16].

In practice, affinity weights w_{pq} are set close to 1 if two neighboring pixels p, q have similar intensities, and weight w_{pq} is set close to zero either when two pixels are far from each other on the pixel grid or if they have largely different intensities [6, 16, 17]. The affinity matrix W could be arbitrarily dense or sparse, e.g. many zeros when representing a 4-connected pixel grid. The non-zero discontinuity costs between neighboring pixels are often set by a Gaussian kernel $w_{pq} = \exp\frac{-\|I_p - I_q\|^2}{2\sigma^2}$ of given bandwidth σ, which works as a soft threshold for detecting high-contrast intensity edges in the image. Thus, loss (3) encourages both the alignment of the segmentation boundary to contrast edges in the (high-resolution) input image and the shortness/regularity of this boundary.


(a) (b) (c)

Figure 2: Semantic segmentation with standard DeepLabV3+(R101) segmentation models [18]: PASCAL validation results for training with (a) log-barrier (9) using class tags, (b) KL-divergence (2) using our approximate size targets, (c) cross-entropy with full (ground truth mask) supervision.

Weakly supervised segmentation methods may also use partial pixel-level ground truth where only some subset $Seeds \subset \Omega$ of image pixels has class labels [6, 7, 9]. In this case it is common to use *partial cross-entropy* loss

$$L_{pce}(S) \;=\; -\sum_{p \in Seeds} \ln S_p^{y_p} \tag{4}$$

where y_p is the ground truth label at a seed pixel p.

1.2 Related balancing losses

Segmentation and classification methods often use “balancing” losses. In the context of classification, image-level predictions can be balanced over the whole training data. For segmentation problems, pixel-level predictions can be balanced within each training image. Our loss is an example of size balancing. Below we review some examples of related balancing loss functions used in prior work.

Fully supervised classification. It is common to modify the standard cross-entropy loss to account for unbalanced training data where some classes are represented more than others. One common example is *weighted cross-entropy*, e.g. defined in [19] for image-level predictions S_i as

$$L_{wce}(S) \;=\; -\sum_{i \in D} w_{y_i} \ln S_i^{y_i} \tag{5}$$

where class weights $w_k \propto \frac{1}{1-\beta^{v_k}}$ are motivated as a re-balancing factor based on the class distribution v in the training dataset D and β is a hyper-parameter. In the fully supervised setting, the purpose of re-weighting cross-entropy is not to make the predictions even closer to the known labels, but to decrease over-fitting to over-represented classes, which improves the model’s generality.

Unsupervised classification. In the context of clustering with soft-max models [20, 21] it is common to use *fairness* loss encouraging equal-size clusters. In this case, there is no ground truth and fairness is one of the discriminative properties enforced by the total loss in order to improve the model predictions on unlabeled training data. The fairness was motivated by information-theoretic arguments in [20] deriving it as a negative entropy of the data-set-level *average prediction* $\hat{S} := \frac{1}{|D|}\sum_{i \in D} S_i$ for dataset D

$$\begin{aligned} L_{fair}(\hat{S}) \;&=\; -H(\hat{S}) \;\equiv\; \sum_k \hat{S}^k \ln \hat{S}^k \\ &\overset{c}{=}\; \sum_k \hat{S}^k \ln \frac{\hat{S}^k}{1/K} \;\equiv\; KL(\hat{S}\|u) \end{aligned} \tag{6}$$

where $u = (\frac{1}{K}, \ldots, \frac{1}{K})$ is a uniform categorical distribution, and symbol $\overset{c}{=}$ indicates that the equality is up to some additive constant independent of $\hat{S}$. Perona et al. [21] pointed out the equivalent KL-divergence formulation of the fairness in (6) and generalized it to a balanced partitioning constraint

$$L_{bal}(\hat{S}) \;=\; KL(\hat{S}\|v) \tag{7}$$

with any given prior distribution v that could be different from uniform.

Semantic segmentation with image-level supervision. Most weakly-supervised semantic segmentation methods use losses based on segment sizes. This is particularly true for image-level supervision techniques [2, 9, 22, 23]. Clearly, segments for tag classes should have positive sizes, and segments for non-tag classes should have zero sizes.

Similarly to our paper, size-based constraints are often defined for the image-level *average prediction* $\bar{S}$, see (1), computed from pixel-level predictions S_p. Many generalized forms of pixel-prediction averaging can be found in the literature, where they are often referred to as *prediction pooling*. Some decay parameter often provides a wide spectrum of options from basic averaging to max-pooling. While the specific form of pooling matters, for simplicity, we discuss the corresponding balancing loss functions assuming basic average prediction $\bar{S}$ in (1).

One of the earliest works on tag-supervised segmentation [9] uses *log-barriers* to "expand" tag objects in each training image and to "suppress" the non-tag objects. Assuming image tags T, their *suppression loss* is defined as

$$L_{suppress}(\bar{S}) \quad \propto \quad -\sum_{k \notin T} \ln(1 - \bar{S}^k) \tag{8}$$

encouraging each non-tag class to have zero average prediction $\bar{S}^k$, which implies zero predictions S_p^k at each pixel. Their *expansion loss*

$$L_{expand}(\bar{S}) \quad \propto \quad -\sum_{k \in T} \ln \bar{S}^k. \tag{9}$$

encourages positive average predictions $\bar{S}^k$ and non-trivial tag class segments.

We observe that the expansion loss (9) may have a bias to equal-size segments, as particularly evident in the case of average predictions. Indeed, (9) implies

$$L_{expand}(\bar{S}) \quad \propto \quad KL(u_{\mathrm{T}} \| \bar{S}) \tag{10}$$

which is a special case of our size loss (2) when the size target $v = u_{\mathrm{T}}$ is a uniform distribution over tag classes. The intention of the log barrier loss (9) is to push image-level size prediction $\bar{S}$ from the boundaries of the probability simplex Δ_K corresponding to the zero-level for the tag classes T. Figure 2(a) shows the results for training based on the total loss combining CRF loss (3) with the log-barrier loss (9). Its unintended bias to equal-size segments (10) is obvious. Note that the mentioned decay parameter used for generalized average predictions should reduce such bias.

Alternatively, it may be safer to use barriers for $\bar{S}$ like

$$L_{flat} \quad = \quad -\sum_{k \in T} \ln \max\{\bar{S}^k, \epsilon\} \tag{11}$$

that have flat bottoms to avoid unintended bias to some specific size target inside the probability simplex Δ_K. Similar thresholded barriers are common [22].

1.3 Contributions

In general, it would be great to have effective image-level supervision for segmentation that only uses barriers like (9) or (11) since they do not require any specific size targets. This corresponds to tag-only supervision. However, our empirical results for semantic segmentation using such barriers were poor and comparable with those in [9]. A number of more recent semantic segmentation methods for tag-level supervision have considerably improved such results [12, 24–30], but they introduce significantly more complex multi-stage training procedures and various architectural modifications, which makes such methods hard to replicate, generalize, or to understand the results. We are focused on general easy-to-understand end-to-end training methods. Our main contributions are:

- We propose and evaluate a new general form of weak supervision, size targets. The size-target supervision can be approximate and is relatively easy to get from human annotators.
- We propose the zero-avoiding variant of KL divergence as a general training loss, allowing our end-to-end size-target approach to be integrated with any segmentation architecture.
- Comprehensive experiments with our size-target method demonstrate state-of-the-art performance across multiple datasets using standard segmentation models typically employed for full supervision, without any architectural modifications.

2 Size-target loss and its properties

Our proposed total loss is very simple

$$L_{total} \; := \; L_{size} + L_{crf} \tag{12}$$

where the two terms are our size-target loss (2) and standard CRF loss (3). The core new component is our size-target loss based on the *forward* KL-divergence. Our size-target loss (2) encourages specific target volumes for tag classes. Additionally, the size-target loss suppresses non-tag classes, encouraging zero volumes for classes not in the image. The CRF loss also contributes to the suppression of redundant classes. Therefore, unlike most prior work on image-level supervision for semantic segmentation, e.g. [9, 2, 12], we do not need separate suppression loss terms like (8). We validated this claim experimentally, they did not change the results.

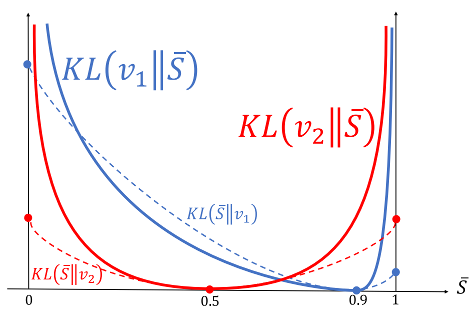

Figure 3: *Forward* vs *reverse* KL divergence. Assuming binary classification $K = 2$, we can represent all possible probability distributions as points on the interval [0,1]. The solid curves illustrate our "strong" size constraint, i.e. the *forward* KL-divergence $KL(v\|\bar{S})$ for the average prediction $\bar{S}$. We show two examples of volumetric prior $v_1 = (0.9, 0.1)$ (blue curve) and $v_2 = (0.5, 0.5)$ (red curve). For comparison, the dashed curves represent reverse KL divergence $KL(\bar{S}\|v)$.

The size-target loss can also be integrated into other weakly-supervised settings, e.g. partial cross-entropy loss (4) commonly used for seeds. We show that using approximate size targets can significantly improve the seed-supervised segmentation in [6] when the seed lengths are short, see the right plot of Fig. 4.

$$L^{'}_{total} \; := \; L_{size} + L_{crf} + L_{pce} \tag{13}$$

As is well known, KL divergence is asymmetric. In our work on image-level supervised segmentation, the order of the estimated and target distributions is crucial. The forward KL divergence possesses a zero-avoiding property, as illustrated in Fig. 3. Specifically, forward KL divergence imposes an infinite penalty when any class with a non-zero target is predicted as zero. In contrast, the penalty of the *reverse* KL divergence is finite and much weaker. When using reverse KL divergence, segmentation models tend to generate trivial solutions, predicting all pixels as the background class. This issue likely arises due to dataset imbalance, where the background class is prevalent. The zero-avoiding property of forward KL divergence ensures that segmentation models do not produce trivial solutions and predict all classes in the image tag sets.

3 Experiments

3.1 Experimental settings

Datasets. We evaluate our approach on three segmentation datasets: PASCAL VOC 2012 [5], MS COCO 2014 [31], and 2017 ACDC Challenge[1] [32]. The PASCAL dataset contains 21 classes. We adopt the augmented training set with 10,582 images [33], following the common practice [34, 9]. Validation and testing contain 1449 and 1456 images. Seed supervision of the PASCAL dataset is from [7]. COCO has 81 classes with 80K training and 40K validation images. ACDC Challenge is to segment the left ventricular endocardium. The training and validation sets contain 1674 and 228 images. The exact size targets are extracted from the ground truth masks.

Approximate size targets. We train segmentation models using approximate size targets $v = (v_k)_{k=1}^K$ generated for each image either by human annotators or by corrupting the exact size targets $\hat{v} = (\hat{v}_k)_{k=1}^K$ with different levels of noise. In all cases, we report the segmentation accuracy on validation data together with *mean relative error* (mRE) of the corresponding corrupted size targets. For each training image containing class k, the *relative error* for the size target v_k is defined as

$$RE(v_k) = \frac{|v_k - \hat{v}_k|}{\hat{v}_k} \tag{14}$$

[1]https://www.creatis.insa-lyon.fr/Challenge/acdc/

where $\hat{v}_k$ is the exact size. mRE averages RE over all images and all classes. For human annotated size targets $v = (v_k)_{k=1}^K$, the relative size errors are computed directly from the definition (14).

When used, synthetic targets $v = (v_k)_{k=1}^K$ are generated by corrupting the exact targets $\hat{v} = (\hat{v}_k)_{k=1}^K$

$$v_k \longleftarrow (1+\epsilon)\hat{v}_k \quad \text{for} \quad \epsilon \sim \mathcal{N}(0,\sigma) \tag{15}$$

where ϵ is white noise with standard deviation σ controlling the level of corruption and operator $\longleftarrow$ represents re-normalization ensuring corrupted targets $(v_k)_{k=1}^K$ add up to one. Equation (15) defines random variable v_k as a function of ϵ. Thus, in this case, mRE can be analytically estimated from σ

$$mRE \;=\; E\left(\frac{|v_k - \hat{v}_k|}{\hat{v}_k}\right) \;\approx\; E(|\epsilon|) \;=\; \sqrt{\frac{2}{\pi}}\,\sigma \tag{16}$$

where E is the expectation operator. The approximation in the middle uses (15) as an equality ignoring re-normalization of the corrupted sizes, and the last equality is a closed-form expression for the *mean absolute deviation* (MAD) of the Normal distribution $\mathcal{N}(0,\sigma)$.

Evaluation metrics for segmentation. We employ *mean Intersection-over-Union* (mIoU) as the evaluation criteria for PASCAL and COCO, and *mean Dice similarity coefficient* (DSC) for the ACDC dataset. The quality on the PASCAL test set is assessed on the online evaluation server.

Implementation details. We evaluate our approach with two types of ResNet-based [4] and one vision transformer (ViT) based [35] segmentation models on the PASCAL and COCO datasets. ResNet-based models follow the implementation of DeepLabV3+ [18] using the backbone of ResNet101 (R101) or the backbone of WideResNet-38 (WR38) [1]. For brevity, we name them R101-based or WR38-based DeepLabV3+ models. For the ViT-based network, We follow the implementation of Segmenter [36], adopting its ViT-B/16 backbone and linear decoder. For experiments on the ACDC datasets, we use MobileNetV2-based [37] DeepLabv3+ model. The R101, WR38, and MobileNetV2 backbones are ImageNet [38] pre-trained. ViT-B/16 backbone is pre-trained on ImageNet-21K [39] and fine-tuned on ImageNet-1k [38]. We directly evaluate our size-target approach on top of the standard architectures without any modification.

Images are resized to 512×512 for PASCAL and COCO, and 256×256 for ACDC. We employ color jittering and horizontal flipping for data augmentation. Segmentation models are trained with stochastic gradient descent on one RTX A6000 GPU with 48 GB GDDR6: 60 epochs for PASCAL and COCO, and 200 epochs for ACDC, with a polynomial learning rate scheduler (power of 0.9). Batch sizes are set to 16 for ResNet and 20 for ViT models on PASCAL, 12 on ACDC, and 12 (ResNet) and 16 (ViT) for MS COCO. The initial learning rate is 0.005 for ACDC and PASCAL's ResNet models, and 0.0005 for PASCAL's ViT models. The initial learning rate on COCO is 0.0005 for ResNet and 0.0001 for ViT models. Loss function (12) is employed for size-target supervision. Loss (13) is only used for seed supervision in Sec. 3.3. The implementation of CRF loss (3) is the same as [6]. We use $2\mathrm{e}^{-9}$ as the weight of the CRF term following the strategy in [6]. Size-target loss (2) and pCE (4) are used for medical images.

3.2 Robustness to Size Errors

We show the size targets can be approximate. The left plot in Fig. 4 illustrates the robustness of our approach to size errors. Segmentation models are trained with synthetic size targets subjected to varying levels of corruption, as defined in (15). The validation accuracy (solid red line) only drops slightly when mRE (16) remains below 16%. The CRF loss (3) further enhances the robustness (solid blue line). When the relative error (mRE) is 4%, there is a noticeable increase in validation accuracy. The downward trend of the training accuracy (dashed blue line) suggests that the observed increases in validation accuracy at $mRE = 4\%$ stem from improved neural network generalization.

3.3 Enhancing seed-based segmentation with size targets

Our size-target approach can be integrated with partial ground truth mask supervision (seeds). The right plot in Fig. 4 demonstrates the results of seed-supervised semantic segmentation with and without size-target supervision. Size targets significantly enhance performance, especially when the seed lengths are short. Without size targets, segmentation performance degrades dramatically as the seed length decreases. Notably, when only one pixel is labeled for each object (seed length ratio $= 0.0$), size-target supervision boosts accuracy from 66.6% to 74%, approaching the performance of full seed supervision (seed length ratio = 1.0).

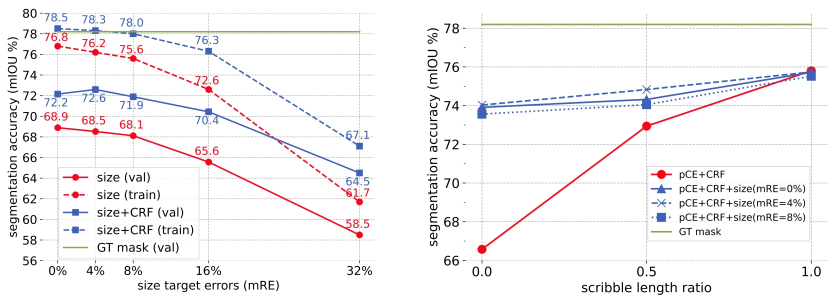

Figure 4: Segmentation results on the PASCAL dataset with R101-based DeeplabV3+ networks. The green bar in both plots indicates the segmentation accuracy for full ground truth masks (i.e. full supervision). The left plot shows the training and validation accuracy using approximate size targets. The segmentation is trained using losses (2) (red curve) or (12) (blue curve), where size targets are subject to various levels of corruption (15,16). The right plot shows validation accuracy for seed supervision of varying lengths with (blue curve) and without (red curve) using size targets. The line styles of the blue curves differentiate among various levels of corruption.

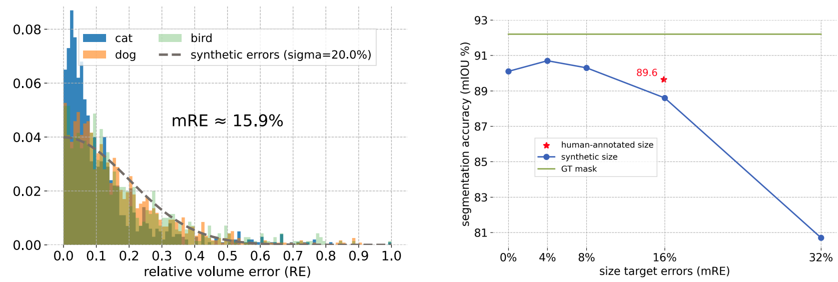

Figure 5: Left plot shows the quality of human annotations in terms of relative errors for the dog, cat, and bird classes within the PASCAl dataset. The histograms are normalized by the number of images in each class. The mean relative error for the three classes is 15.9%. For comparison, the dashed line shows the relative error distribution of synthetic size targets as defined in (15) for $\sigma = 20.0\%$ which aligns with the mRE of 15.9%, see (16). The right plot presents 4-way multi-class (cat, dog, bird, and background) segmentation accuracy using human-annotated (red star at $mRE = 15.9\%$) and synthetic (blue curve) size targets, employing ResNet101-based DeeplabV3+ networks. Consistent with experiments in Sec. 3.2, synthetic size targets are generated at various levels of corruption. The green line indicates the segmentation accuracy of full supervision using ground truth masks.

3.4 Human-annotated size targets

Annotation tool. In this section, our approach is evaluated with size targets annotated by humans. We annotated training images for a subset of PASCAL classes, including cat, dog, and bird. A user interface with an assistance tool was developed to facilitate the annotation. The assistance tool overlays grid lines partitioning the image into 5×4 small rectangles or 3×3 large rectangles. Users can determine the size of a class in an image by counting rectangles (fractions allowed) or entering the percentage relative to the image size. Annotators can choose finer or coarser partitioning for each image depending on the object size. We evaluate relative errors with (14) for human annotations. Empirical evidence shows that annotators are approximately two times more accurate with the assistance tool, especially for small objects in the image. The last two columns of Table 1 report the annotation speed per image and mean relative error (14) for each class. The left plot in Fig. 5 shows the histograms of relative errors for human annotations. The histograms illustrate that annotated size errors are mostly below 10%, but occasional large mistakes (heavy tails) raise the mean error.

Segmentation with human-annotated size. Segmentation models trained with human-annotated size targets show robustness to human "heavy tail" errors. We compare the accuracy for human-annotated and synthetic size targets in the right plot of Fig. 5. The accuracy for human-annotated size (indicated by the red star in the plot) approaches 97.2% (89.6%/92.2%) of the full supervision performance, demonstrating that size-target approach is significantly robust to human errors. Binary segmentation accuracy for each class is reported in the shaded cells in Table 1. The performance of

supervision	gt mask	gt size	human-annotated size		
	mIoU	mIoU	mIoU	speed	mRE
cat	90.6%	88.8%	88.0%	12.6s	12.3%
dog	88.1%	84.3%	84.5%	9.1s	16.6%
bird	88.8%	86.2%	86.4%	15.2s	20.1%

Table 1: Human-annotated size targets. Two columns on the right show the average speed and relative error for each class we annotated. The shaded cells compare the accuracy of binary segmentation models trained with ground truth masks, ground truth size, and human-annotated size.

binary segmentation models trained with human-annotated size targets is comparable to those trained with precise size targets.

3.5 Comparison with the state-of-the-art methods

Our general training losses are applied to three standard architectures (R101-DeepLabV3+, WR38-DeepLabV3+, and ViT-Linear) for semantic segmentation as is, without any modification. Our results are highlighted in Table 2. The models are trained using synthetic size targets with an approximate mean relative error (mRE) of 8%. We chose this corruption level because its performance is close to human annotations, as shown in the right plot of Figure 5. Since our single-stage (end-to-end) approach is completely general, it is possible to use it in specialized architectures or complex training procedures. Likely, this would further improve the results, but this is not the focus of our work. The rest of Table 2 shows the results for semantic segmentation methods (of different complexities) for weak and full supervision. Methods are divided into multi-stage and single-stage methods, grouped by their backbones. Typical single-stage methods improve their results using complex architectural or training modifications such as additional training branches, extra refinement modules, or specialized training strategies. However, we achieve state-of-the-art using only standard segmentation architectures, commonly used in full supervision. The R101-based DeepLabV3+ model trained with approximate size targets approaches 92% (71.9/78.2) of its full supervision performance on PASCAL. The WR38-based DeepLabV3+ model trained with approximate size-target supervision surpasses other methods employing the same backbone by approximately 10%. Using the standard vision transformer architecture [36], the size-target approach achieves approximately 96% of the

Backbone	Decoder	Architectural/training modification	Supervision	PASCAL Val	PASCAL Test	COCO Val
		Multi-stage methods				
R101	DeepLabV3+	MARS [40] arXiv'23	tags	77.7	77.2	49.4
R101	DeepLabV2	MatLabel [41] ICCV'23	tags	73.0	72.7	45.6
WR38	LargeFOV	MCT [42] CVPR'22	tags	71.9	71.6	42.0
WR38	LargeFOV	MCTOCR [43] CVPR'23	tags	72.7	72.0	42.5
SWIN	DeepLabV2	ReCAM [44] CVPR'22	tags	71.8	72.2	47.9
ViT-S	"Grad-clip"	WeakTr [26] arXiv'23	tags	78.4	79.0	50.3
		Single-stage (end-to-end) methods				
R101	DeeplabV3+	-	size (8%)	71.9	72.4	45.0
R101	DeeplabV3+	-	full	78.2	78.2	60.4
WR38	DeepLabV3+	SSSS [2] CVPR'20	tags	62.7	64.3	-
WR38	Conv	RRM [45] AAAI'20	tags	62.6	62.9	-
WR38	DeeplabV3+	-	size (8%)	72.7	72.6	-
ViT-B	LargeFOV	ToCo [28] CVPR'23	tags	71.1	72.2	42.3
ViT-B	Conv	SeCo [29] arXiv'24	tags	74.0	73.8	46.7
ViT-B	LargeFOV	CoSA [30] arXiv'24	tags	76.2	75.1	51.0
ViT-B	Linear	-	size (8%)	78.1	78.2	56.3
ViT-B	Linear	-	full	81.4	80.7	-

Table 2: Semantic segmentation results (mIoU%) on PASCAL and COCO. The supervision column indicates a form of supervision: image-level class *tags*, *size* targets (our highlighted results), or *full* supervision with pixel-accurate masks. The percentage after "size" is the accuracy (mRE) of our corrupted size targets (15,16). Our approach does not require any complex architectural modification or multi-stage training procedures needed for tag supervision, see "Modification" column.

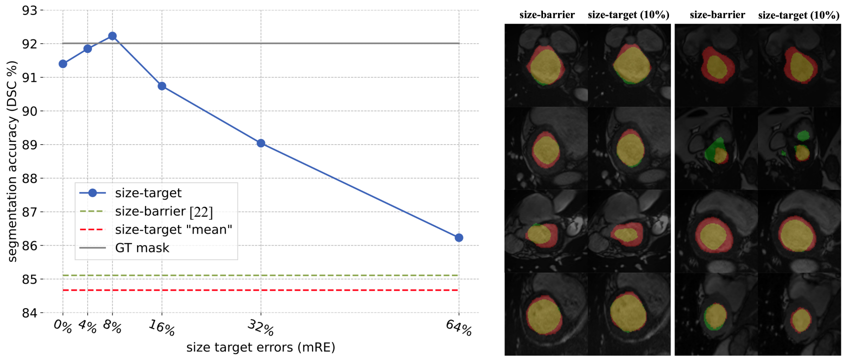

Figure 6: Size-targets (2) vs. size-barriers (17) on the ACDC dataset. The left plot shows the accuracy of the binary segmentation models (MobileNetV2-based DeeplabV3+) measured by DSC. The blue curve shows size-target accuracy with various levels of corruption. The dashed green line shows the accuracy of the size-barrier technique [22]. The dashed red line shows the accuracy using the mean size target for all training images. The gray line indicates the result of full supervision. The right image shows randomly selected qualitative results of size-barrier [22] and approximate size target ($mRE = 8\%$). Yellow shows true positive pixels, green is false positive, and red is false negatives.

full supervision performance on the Pascal dataset. Despite its simplicity, the size-target approach outperforms other complex single-stage methods on both datasets.

3.6 Medical data: size-target vs. size-barrier

Our method is also promising for medical image segmentation, benefiting from the consistency in object sizes across similar medical images, which healthcare professionals can easily estimate. We compare our size-target approach with the thresholded size-barrier technique [22], proposed for the weakly supervised medical image semantic segmentation. The size-barrier loss enforces inequality size constraints. Given the lower bound of each class, the thresholded size-barrier loss is

$$L_{flat\_sq}(S) \;\; = \;\; \sum_k \left(\max\{a_k - \bar{S}^k, 0\}\right)^2, \tag{17}$$

where a_k is a lower bound of class k. We train binary segmentation models with a combination of partial cross-entropy loss (4) and size constraint loss: size-target (2) or size-barrier (17). Seeds used in the experiments are obtained using the same method provided in [22]. The object and background barrier, a_{obj} and a_{bg} are set based on [22]. In the size-barrier experiments, similarly to [22], we suppress the non-tag classes, using the loss $L_{sup}(S) = (\bar{S}^{obj})^2$. Conversely, size-target loss automatically suppresses non-tag classes as discussed in Sec. 2. The left plot in Fig. 6 displays the segmentation accuracy against different levels of size target corruption. Our size-target loss consistently outperforms size-barrier loss, maintaining its superiority even when using highly noisy size targets. The peak in the accuracy curve aligns with the experimental results in Sec. 3.2 and Sec. 3.4. The accuracy of the model trained using size targets with relative errors of 8% surpasses the full supervision performance. Additionally, using a fixed average size target across all training images can yield performance comparable to the size-barrier method, see the dashed red line in the left plot of Fig. 6. The right image in Fig. 6 shows qualitative examples of both methods.

4 Conclusions

We proposed a new image-level supervision for semantic segmentation: size targets. Such targets could be approximate. In fact, our results suggest that some errors can benefit generalization. The size annotation by humans requires little extra effort compared to the standard image-level tags and it is much cheaper than the full pixel-accurate ground truth masks. We proposed an effective size-target loss based on forward KL divergence between the soft size targets and the average prediction. In combination with the standard CRF-based regularization loss, our approximate size-target supervision on standard segmentation architectures (DeepLab and ViT) achieves state-of-the-art performance. Our general easy-to-understand approach outperforms significantly more complex weakly-supervised techniques based on model modifications and multi-stage training procedures.

References

[1] Zifeng Wu, Chunhua Shen, and Anton van den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition, 2016.

[2] Nikita Araslanov and Stefan Roth. Single-stage semantic segmentation from image labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4253–4262, 2020.

[3] V. Kulharia, S. Chandra, A. Agrawal, P. Torr, and A. Tyagi. Box2seg: Attention weighted loss and discriminative feature learning for weakly supervised segmentation. In *ECCV'20*, 2020.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[5] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–308, 2009.

[6] Meng Tang, Federico Perazzi, Abdelaziz Djelouah, Ismail Ben Ayed, Christopher Schroers, and Yuri Boykov. On regularized losses for weakly-supervised cnn segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 507–522, 2018.

[7] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3159–3167, 2016.

[8] George Papandreou, Liang-Chieh Chen, Kevin P Murphy, and Alan L Yuille. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1742–1750, 2015.

[9] Alexander Kolesnikov and Christoph H Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 695–711. Springer, 2016.

[10] Xu Ji, Joao F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9865–9874, 2019.

[11] Jang Hyun Cho, Utkarsh Mall, Kavita Bala, and Bharath Hariharan. Picie: Unsupervised semantic segmentation using invariance and equivariance in clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16794–16804, 2021.

[12] Tianfei Zhou, Meijie Zhang, Fang Zhao, and Jianwu Li. Regional semantic contrast and aggregation for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4299–4309, 2022.

[13] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018.

[14] Jyh-Jing Hwang, Stella X Yu, Jianbo Shi, Maxwell D Collins, Tien-Ju Yang, Xiao Zhang, and Liang-Chieh Chen. Segsort: Segmentation by discriminative sorting of segments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7334–7344, 2019.

[15] Yuri Y Boykov and M-P Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *Proceedings eighth IEEE international conference on computer vision. ICCV 2001*, volume 1, pages 105–112. IEEE, 2001.

[16] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence*, 23(11):1222–1239, 2001.

[17] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected CRFs with Gaussian edge potentials. *Advances in neural information processing systems*, 24, 2011.

[18] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[19] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9268–9277, 2019.

[20] John Bridle, Anthony Heading, and David MacKay. Unsupervised classifiers, mutual information and'phantom targets. *Advances in neural information processing systems*, 4, 1991.

[21] Andreas Krause, Pietro Perona, and Ryan Gomes. Discriminative clustering by regularized information maximization. *Advances in neural information processing systems*, 23, 2010.

[22] Hoel Kervadec, Jose Dolz, Meng Tang, Eric Granger, Yuri Boykov, and Ismail Ben Ayed. Size-constraint loss for weakly supervised CNN segmentation. In *Medical Imaging with Deep Learning*, 2018.

[23] Deepak Pathak, Philipp Krahenbuhl, and Trevor Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1796–1804, 2015.

[24] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4981–4990, 2018.

[25] Zhaozheng Chen and Qianru Sun. Extracting class activation maps from non-discriminative features as well. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3135–3144, 2023.

[26] Lianghui Zhu, Yingyue Li, Jieming Fang, Yan Liu, Hao Xin, Wenyu Liu, and Xinggang Wang. Weaktr: Exploring plain vision transformer for weakly-supervised semantic segmentation. *arXiv preprint arXiv:2304.01184*, 2023.

[27] Xiaobo Yang and Xiaojin Gong. Foundation model assisted weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 523–532, 2024.

[28] Lixiang Ru, Heliang Zheng, Yibing Zhan, and Bo Du. Token contrast for weakly-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3093–3102, 2023.

[29] Zhiwei Yang, Kexue Fu, Minghong Duan, Linhao Qu, Shuo Wang, and Zhijian Song. Separate and conquer: Decoupling co-occurrence via decomposition and representation for weakly supervised semantic segmentation. *arXiv preprint arXiv:2402.18467*, 2024.

[30] Xinyu Yang, Hossein Rahmani, Sue Black, and Bryan M Williams. Weakly supervised co-training with swapping assignments for semantic segmentation. *arXiv preprint arXiv:2402.17891*, 2024.

[31] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[32] Olivier Bernard, Alain Lalande, Clement Zotti, Frederick Cervenansky, Xin Yang, Pheng-Ann Heng, Irem Cetin, Karim Lekadir, Oscar Camara, Miguel Angel Gonzalez Ballester, et al. Deep learning techniques for automatic mri cardiac multi-structures segmentation and diagnosis: is the problem solved? *IEEE transactions on medical imaging*, 37(11):2514–2525, 2018.

[33] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *2011 international conference on computer vision*, pages 991–998. IEEE, 2011.

[34] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[35] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[36] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7262–7272, 2021.

[37] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[38] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

[39] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[40] Sanghyun Jo, In-Jae Yu, and Kyungsu Kim. Mars: Model-agnostic biased object removal without additional supervision for weakly-supervised semantic segmentation. *arXiv preprint arXiv:2304.09913*, 2023.

[41] Changwei Wang, Rongtao Xu, Shibiao Xu, Weiliang Meng, and Xiaopeng Zhang. Treating pseudo-labels generation as image matting for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 755–765, 2023.

[42] Lian Xu, Wanli Ouyang, Mohammed Bennamoun, Farid Boussaid, and Dan Xu. Multi-class token transformer for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4310–4319, 2022.

[43] Zesen Cheng, Pengchong Qiao, Kehan Li, Siheng Li, Pengxu Wei, Xiangyang Ji, Li Yuan, Chang Liu, and Jie Chen. Out-of-candidate rectification for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23673–23684, 2023.

[44] Zhaozheng Chen, Tan Wang, Xiongwei Wu, Xian-Sheng Hua, Hanwang Zhang, and Qianru Sun. Class re-activation maps for weakly-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 969–978, 2022.

[45] Bingfeng Zhang, Jimin Xiao, Yunchao Wei, Mingjie Sun, and Kaizhu Huang. Reliability does matter: An end-to-end weakly supervised semantic segmentation approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12765–12772, 2020.

[46] A. Bearman, O. Russakovsky, V. Ferrari, and F. Li. Semantic segmentation with point supervision. In *ECCV*, 2015.

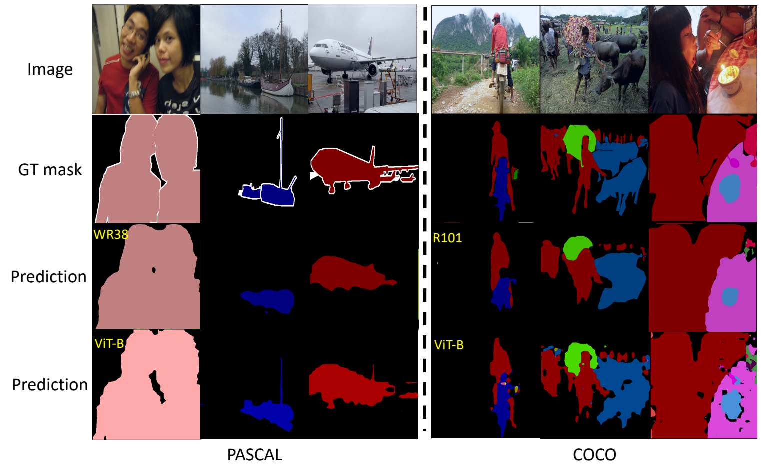

Figure 7: Segmentation examples using size-target supervision ($mRE = 8\%$). Model backbones are shown in the top-left corner of the predictions, see Table 2 for decoders.

A Appendix / supplemental material

A.1 Labeling costs and accuracies reported in Figure 1

Labelling costs. Figure 1 in the paper shows labeling speed and accuracy for different forms of supervision on PASCAL VOC. The table at the bottom of Figure 1 shows ballpark estimates of average labeling time per image in the whole dataset. We use the data in [46], as well as Table 1 in the paper, and aggregate all labeling speeds from "per class", "per instance", or "per point" to "per image" using the average number of instances or classes in each image and the aggregation rules formulated in [46], see their Section 4. The top-left corner in each picture shows the corresponding estimated labeling times for the representative multi-instance image. All the labeling times are only rough estimates, but they are intuitive. The relative costs for point supervision seem underestimated, but they follow evaluation conventions detailed in [46].

Accuracies. The values of "point", "size target" and "full supervision" accuracy (mIOU%) are based on the experiments in the paper (Figure 4). We follow the learning rate scheme in DeepLabV3+ [18] for the training with full supervision. For fairness, we compare these with end-to-end methods using similar ResNet backbones in *tag*- [2] and *box*-supervision [3]. Typical SOTA methods for tag and box supervision use special architectural modifications, unlike our generic size-target loss, cannot be seamlessly plugged into any segmentation model.

A.2 Qualitative results

Figure 7 presents the qualitative examples of our method on PASCAL (left) and COCO (right) validation sets. Despite size targets providing only image-level information, segmentation models can precisely identify object locations, eliminating the need for localization methods like CAM.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Contributions are included in the abstract and listed in Sec. 1.3 in the introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [No]

   Justification: Although the limitations were not explicitly detailed in the paper, we mentioned that only a subset of the PASCAL dataset was labeled due to resource constraints, see Sec. 3.4. To address this, we generated approximate synthetic size targets by corrupting the exact size targets. This allowed us to evaluate our method on the entire PASCAL dataset, as well as on COCO and ACDC datasets.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer:[NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Our size-target loss function is discussed in the 2. The experimental settings are discussed in the 3.1

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: To preserve anonymity, the code will be released in the final version.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental setting is detailed in the Sec. 3.1

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Error bars are not reported because it would be too computationally expensive. Our plots in Figure 4, 5, 6 are smooth enough to verify our method.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The information on the computer resources is detailed in Sec. 3.1

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research in the paper conforms with the code of ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: Our research on weakly-supervised semantic segmentation is a purely technical advancement to improve image segmentation, with no direct societal impacts or associated ethical concerns.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The owners of assets used in this paper are credited and the license is mentioned and respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.