Single-Stage Semantic Segmentation from Image Labels

Nikita Araslanov Stefan Roth
Department of Computer Science, TU Darmstadt

Abstract

Recent years have seen a rapid growth in new approaches improving the accuracy of semantic segmentation in a weakly supervised setting, i.e. with only image-level labels available for training. However, this has come at the cost of increased model complexity and sophisticated multi-stage training procedures. This is in contrast to earlier work that used only a single stage – training one segmentation network on image labels – which was abandoned due to inferior segmentation accuracy. In this work, we first define three desirable properties of a weakly supervised method: local consistency, semantic fidelity, and completeness. Using these properties as guidelines, we then develop a segmentation-based network model and a self-supervised training scheme to train for semantic masks from image-level annotations in a single stage. We show that despite its simplicity, our method achieves results that are competitive with significantly more complex pipelines, substantially outperforming earlier single-stage methods.

1. Introduction

Many applications of scene understanding require some form of semantic localisation with pixel-level precision, hence semantic segmentation has enjoyed enormous popularity. Despite the successes of supervised learning approaches [8, 36], their general applicability remains limited due to their reliance on pixel-level annotations. We thus consider the task of learning semantic segmentation from image-level annotations alone, aiming to develop a practical approach. This problem setup is especially challenging compared to other weakly supervised scenarios that assume available localisation cues, such as bounding boxes, scribbles, and points [4, 10, 27, 33, 51].

Attention mechanisms, such Class Activation Maps (CAM) [63], offer a partial solution: they localise the most discriminative regions in the image using only a pre-trained classification network. Such masks, however, are quite coarse – they violate object boundaries, tend to be incomplete for large-scale objects and imprecise for small ones. This is not surprising, since attention maps were not designed for segmentation in the first place. Nevertheless, most methods for weakly supervised segmentation from image labels adopt attention maps (e.g., CAMs) as initial seeds for further refinement. Yet the remarkable progress these methods have achieved – currently reaching more than 80% of fully supervised accuracy [3, 57] – has come along with increased model and training complexity. While early methods consisted of a single stage, i.e. training one network [17, 38, 40, 41], they were soon superseded by more advanced pipelines, employing multiple models, training cycles, and off-the-shelf saliency methods [24, 55, 57, 60].

In this work, we develop an effective single-stage approach for weakly supervised semantic segmentation that streamlines previous multi-stage attempts, and uses neither saliency estimation nor additional data. Our key insight is to enable segmentation-aware training for classification. Consider Fig. 2 depicting typical limitations of attention maps: (i) two areas in local proximity with similar appearance may be assigned different classes, i.e. the semantic labelling may be locally inconsistent; (ii) attention maps tend to be in-
complete in terms of covering the whole extent of the object; (iii) while the area of the attention maps dominates for the correct object class, parts of the map may still be mislabelled (i.e., are semantically inaccurate). These observations lead us to define three properties a segmentation-aware training should encompass: (a) local consistency implies that neighbouring pixels with similar appearance share the same label; (b) semantic fidelity is exhibited by models producing segmentation masks that allow for reliable classification decisions (e.g., with good generalisation); (c) completeness means that our model identifies all visible class occurrences in the image. Note that since classification requires only sufficient evidence, CAMs neither ensure completeness nor local consistency.

Using these concepts as our guidelines, we design an approach that significantly outperforms CAMs in terms of segmentation accuracy. First, we propose normalised Global Weighted Pooling, a novel process for computing the classification scores, which enables concurrent training for the segmentation task. Second, we encourage masks to heed appearance cues with Pixel-Adaptive Mask Refinement. These masks are supplied to our model as pseudo ground truth for self-supervised segmentation. Third, to counter the compounding effect of inaccuracies in the pseudo mask annotation (a common problem of self-supervised methods), we introduce a Stochastic Gate that mixes feature representations with varying receptive field sizes. As our experiments demonstrate, the resulting single-stage model offers segmentation quality on par or outperforming the state of the art, while being simple to train and to use.

2. Related Work

Methods for weakly supervised semantic segmentation have evolved rapidly from simple single-stage models to more complex ones, employing saliency estimation methods, additional data (e.g., videos), and fully supervised “fine-tuning” (cf. supplemental material, Sec. A).

Single-stage methods. Following [40], Pinheiro & Collobert [41] used a Multiple Instance Learning (MIL) formulation, but applied a LogSumExp-aggregation of the pixel-level predictions in the output layer to produce the class scores, and refined the segments with image-level priors. Papandreou et al. [38] took an expectation-maximisation (EM) approach, where masks are inferred from intermediate predictions and used as pseudo ground truth. [43] combines top-down attention masks with bottom-up segmentation cues in an end-to-end model using CRF-RNN [62]. The attention-based model [17] allows for joint classification and segmentation training in a cross-domain setting. Despite their simplicity, single-stage models have fallen out of favour, owing to their inferior segmentation accuracy.

Seed and expand. Kolesnikov & Lampert [28] introduced the idea of expanding high-precision localisation cues, such as CAMs [63], to align with segment boundaries. In this framework, a segmentation network can be trained end-to-end, but the localisation cues are pre-computed from a standalone classification network. Consequently, higher-quality localisation [32] can further improve the segmentation accuracy. Following the seed-and-expand principle, Huang et al. [24] employed a seeded region growing algorithm [1] to encourage larger coverage of the initial localisation seeds.

Erasing. One common observation with CAMs [63] is their tendency to identify only the most discriminative class evidence. Wei et al. [55] explored the idea of “erasing” these high-confidence areas from the images and re-training the network for classification using the left-over regions to mine for additional cues. Similarly, SeeNet [21] implements erasing using two decoder branches aided by saliency [20]: the first branch removes the peak CAM-response and feeds into the second. To avoid re-training or modifying the decoder structure, Chaudhry et al. [5] iteratively applied an off-the-shelf saliency detector [35] to a progressively erased image in order to accumulate the foreground mask.

Multiple training rounds. Another line of work trains a chain of segmentation networks, each learning the predictions of its predecessor [27]. Wei et al. [56] sequentially trained three networks in increasing order of task difficulty. Following Khoreva et al. [27], Jing et al. [26] used multiple training rounds and refine intermediate results with GrabCut [42], yet without the use of bounding-box annotations. Similarly, Wang et al. [54] iteratively train a network with the seed-and-expand principle, refine the intermediate results with saliency maps [53], and provide them as supervision to the segmentation network.

Additional data. Hong et al. [18] sought additional data in videos for training a class-agnostic decoder: at inference time the class-specific attention map has to pass through the decoder individually. More recently, Lee et al. [31] aggregated additional attention maps from videos by merging the detected masks from consecutive frames via warping.

Saliency and further refinement. Zeng et al. [60] showed that joint training with saliency ground truth sig-
nificantly improves the mask accuracy. Fan et al. [14] combined the saliency detector of [13] with attention maps to partition the features within each detection window into a segment. To increase the recall of attention maps, Wei et al. [57] added multiple dilation rates in the last layer. Towards the same goal, Lee et al. [30] stochastically selected hidden units and supplied the improved initial seeds to DSRG [24].

Image-level labels only. In this work we adhere to the early practice of relying only on image-label annotation. Following this setup, Ahn & Kwak [3] modeled the pixel-level affinity distance from initial CAMs and employed a random walk to propagate individual class labels at the pixel level. Instance-aware affinity encoding provides additional benefits [2]. Both methods require training a standalone segmentation network on the masks for the final result – a common practice (e.g., [24, 55, 56, 57]). Shimoda & Yanai [45] proposed to post-refine these masks with a cascade of three additional “difference detection” modules.

In contrast to these works, we develop a competitive alternative with practicality in mind: a single network for weakly supervised segmentation, trained in one cycle.

3. Model

3.1. Overview

Our network model, illustrated in Fig. 3, follows the established design of a fully convolutional segmentation network with a softmax output and skip connections [36]. This allows for a straightforward extension of any segmentation-based network architecture and exploiting pre-trained classification models for parameter pre-conditioning. Inference requires only a single forward pass, analogous to fully supervised segmentation networks. By contrast, however, our model allows to learn for segmentation in a self-supervised fashion from image labels alone.

We propose three novel components relevant to our task: (i) a new class aggregation function, (ii) a local mask refinement module, and (iii) a stochastic gate. The purpose of the class aggregation function is to leverage segmentation masks for classification decisions, i.e. to provide semantic fidelity as defined earlier. To this end, we develop a normalised Global Weighted Pooling (nGWP) that utilises pixel-level confidence predictions for relative weighting of the corresponding classification scores. Additionally, we incorporate a focal mask penalty into the classification scores to encourage completeness. We discuss these components in more detail in Sec. 3.2. Next, in order to comply with local consistency, we propose Pixel-Adaptive Mask Refinement (PAMR), which revises the coarse mask predictions w.r.t. appearance cues. The updated masks are further used as pseudo ground truth for segmentation, trained jointly along with the classification objective, as we explain in Sec. 3.3. The refined masks produced by PAMR may still contain inaccuracies w.r.t. the ground truth, and self-supervised learning may further compound these errors via overfitting. To alleviate these effects, we devise a Stochastic Gate (SG) that combines a deep feature representation susceptible to this phenomenon with more robust, but less expressive shallow features in a stochastic way. Sec. 3.4 provides further detail.

3.2. Classification scores

CAMs. It is instructive to briefly review how the class score is normally computed with Global Average Pooling (GAP), since this analysis builds the premise for our aggregation mapping. Let \( x_{k,:,:} \) denote one of \( K \) feature channels of size \( h \times w \) preceding GAP, and \( a_{c,:} \) be the parameter vector for class \( c \) in the fully connected prediction layer. The class score for class \( c \) is then obtained as

\[
y_{c}^{\text{GAP}} = \frac{1}{hw} \sum_{k=1}^{K} a_{c,k} \sum_{i,j} x_{k,i,j}.
\]

Next, we can compute the Class Activation Mapping (CAM) [63] for class \( c \) as

\[
m_{c,:,:}^{\text{CAM}} = \max \left( 0, \sum_{k=1}^{K} a_{c,k} x_{k,:,:} \right).
\]

Fig. 4a illustrates this process, which we refer to as CAM-GAP. From Eq. (1) we observe that it encourages all pixels in the feature map to identify with the target class. This may disadvantage small segments and increase the reliance of the classifier on the context, which can be undesirable due to a loss in mask precision. Also, as becomes evident from Eq. (2), there are two more issues if we were to adopt CAM-GAP to provide segment cues for learning. First, the mask value is not bounded from above, yet in segmentation we seek a normalised representation (e.g. \( \in (0,1) \)) that can be interpreted as a confidence by downstream applications. Second, GAP does not encode the notion of pixel-level competition from the underlying segmentation task where each pixel can assume only one class label (i.e. there is no softmax or a related component). We thus argue that CAM-GAP is ill-suited for the segmentation task.

![Figure 3. Architecture overview. Our model shares the design of a segmentation network, but additionally makes use of normalised Global Weighted Pooling (nGWP, Sec. 3.2) and Pixel-Adaptive Mask Refinement (PAMR, Sec. 3.3) to enable self-supervised learning for segmentation from image labels.](image-url)
Going beyond CAMs. To address this, we propose a novel scheme of score aggregation, see Fig. 4b for an overview, which allows for seamless integration into existing backbones, yet does not inherit the shortcomings of CAM-GAP. Note that the following discussion is orthogonal to the loss function applied on the final classification scores, which we keep from our baseline model.

Given features \( x_{c:i} \) of size \( C \times h \times w \) for each pixel, we then add a background channel (with a constant value) and compute a pixelwise softmax to obtain masks with confidence values \( m_{c:i} \) — this is a standard block in segmentation. To compute a classification score, we propose normalised Global Weighted Pooling (nGWP), defined as

\[
y_{c}^{\text{nGWP}} = \frac{\sum_{i,j} m_{c,i,j} y_{c,i,j}}{\sum_{i,j} m_{c,i,j} + \epsilon},
\]

Here, a small \( \epsilon > 0 \) tackles the saturation problem often observed in practice (cf. supplemental material, Sec. B).

As we observe from Eq. (3), nGWP is invariant to the mask size. This may bring advantages for small segments, but lead to inferior recall compared to the more aggressive GAP aggregation. To encourage completeness, we encourage increased mask size for positive classes with a penalty term:

\[
y_{c}^{\text{size}} = \log \left( \lambda + \frac{1}{hw} \sum_{i,j} m_{c,i,j} \right).
\]

The magnitude of this penalty is controlled by a small \( \lambda > 0 \). The logarithmic scale ensures that we incur a large negative value of the penalty only when the mask is near zero. Since we decouple the influence of the class scores (captured by Eq. (3)) from that of the mask size (through Eq. (4)), we can apply difficulty-aware loss functions. We generalise the penalty term in Eq. (4) to the focal loss [34], used in our final model:

\[
y_{c}^{\text{size-focal}} = (1 - m_{c})^p \log(\lambda + m_{c}), \quad \bar{m}_{c} = \frac{1}{hw} \sum_{i,j} m_{c,i,j},
\]

Note that as the mask size approaches zero, \( \bar{m}_{c} \to 0 \), the penalty retains its original form, i.e. Eq. (4). However, if the mask is non-zero, \( p > 0 \) discounts the further increase in mask size to focus on the failure cases of near-zero masks. We compute our final classification scores as \( y_{c} \equiv y_{c}^{\text{nGWP}} + y_{c}^{\text{size-focal}} \) and use the multi-label soft-margin loss function [39] used in previous work [3, 57] as the classification loss,

\[
\mathcal{L}_{\text{cls}}(y, z) = -\frac{1}{C} \sum_{c=1}^{C} z_{c} \log \left( \frac{1}{1 + e^{-y_{c}}} \right) + \frac{e^{-y_{c}}}{1 + e^{-y_{c}}},
\]

where \( z \) is a binary vector of ground-truth labels. The loss encourages \( y_{c} < 0 \) for negative classes (i.e. when \( z_{c} = 0 \)) and \( y_{c} > 0 \) for positive classes (i.e. when \( z_{c} = 1 \)).

3.3. Pixel-adaptive mask refinement

While our classification loss accounts for semantic fidelity and completeness, the task of local mask refinement is to fulfil local consistency: nearby regions sharing the same appearance should be assigned to the same class.

The mapping formalising this idea takes the pixel-level mask predictions \( m_{c:i} \in (0, 1)^{(C+1) \times h \times w} \) (note +1 for the background class) and considers the image \( I \) to produce refined masks \( m_{c}^{t+1} \). Such a mapping has to be efficient, since we will use it to produce self-supervision for segmentation trained concurrently with the classification objective. Therefore, a naive choice of GrabCut [42] or dense CRFs [29] would slow down the training process. Instead, our implementation derives from the Pixel-Adaptive Convolution (PAC) [49]. The idea, illustrated in Fig. 5, is to iteratively update pixel mask \( m_{c,i,j} \) using a convex combination of the labels of its neighbours \( \mathcal{N}(i,j) \), i.e.
at the $t$th iteration we have

$$m_{t,i,j}^c = \sum_{(l,n) \in N(i,j)} \alpha_{i,j,l,n} \cdot m_{t-1}^c,$$

(7)

where the pixel-level affinity $\alpha_{i,j,l,n}$ is a function of the image $I$. To compute $\alpha$, we use a kernel function on the pixel intensities $I$,

$$\kappa(I_{i,j}, I_{n,l}) = \frac{\|I_{i,j} - I_{n,l}\|}{\sigma_{i,j}},$$

(8)

where we define $\sigma_{i,j}$ as the standard deviation of the image intensity computed locally for the affinity kernel. We apply a softmax to obtain the final affinity distance $\alpha_{i,j,l,n}$ for each neighbour $(l,n)$ of $(i,j)$, i.e. $\alpha_{i,j,l,n} = e^{\kappa(I_{i,j}, I_{n,l})} / \sum_{(q,r) \in N(i,j)} e^{\kappa(I_{i,j}, I_{r,q})}$, where $\kappa$ is the average affinity value across the RGB channels.

This local refinement, termed Pixel-Adaptive Mask Refinement (PAMR), is implemented as a parameter-free recurrent module, which iteratively updates the labels following Eq. (7). Clearly, the number of required iterations depends on the size and shape of the affinity kernel (e.g. $3 \times 3$ in Fig. 5). In practice, we combine multiple $3 \times 3$-kernels with varying dilation rates. We study the choice of the affinity structure in more detail in our ablation study (cf. Sec. 4.2). Note that since we do not back-propagate through PAMR, it is always in “evaluation” mode, hence memory-efficient. In practice, one iteration adds less than 1% of the baseline’s GPU footprint, and we empirically found 10 refinement steps to provide a sufficient trade-off between the efficiency and the delivered accuracy boost from PAMR.

**Self-supervised segmentation loss.** We generate pseudo ground-truth masks from PAMR by considering pixels with confidence $> 60\%$ of the maximum value ($> 70\%$ for the background class). Conflicting pixels and pixels with low confidence are ignored by the loss function. We fully discard images for which some of the ground-truth classes do not give rise to any confident pseudo ground-truth pixels. Following the fully supervised case [8], we use pixelwise cross-entropy, but balance the loss distribution across the classes, i.e. the loss for each individual class is normalised w.r.t. the number of corresponding pixels contained in the pseudo ground truth. The intermediate results for segmentation self-supervision at training time are illustrated in Fig. 6.

### 3.4. Stochastic gate

The fundamental premise of self-supervised learning is the idea of bootstrapping: we expect the model to “average out” inaccuracies (manifested by their irregular nature) in the pseudo ground truth, thereby improving the predictions and thus the pseudo supervision. However, this is at odds with the representational power of the model, since a powerful model may just as well learn to mimic these errors. Strong evidence from previous work indicates that the large receptive field of the deep features enables the model to learn such complex phenomena in segmentation [8, 58, 61].

To counter the compounding effect of the errors in self-supervision, we propose a type of regularisation, referred to as Stochastic Gate (SG). The underlying idea, shown in Fig. 7, is to stochastically combine deep features (with a large receptive field) with features from the preceding layers, where the size of receptive field is moderate. Formally, let $x^{(d)}$ and $x^{(s)}$ represent the activation in the deep and shallow feature map, respectively (omitting the tensor subscripts for brevity). Applying SG for each pixel at training time is reminiscent of Dropout [48]:

$$x_{SG} = (1 - r)x^{(d)} + rx^{(s)} \quad \text{with} \quad r \sim \text{Bernoulli}(\psi),$$

(9)

where the mixing rate $\psi \in [0, 1]$ regulates the proportion of the two feature representations in the output tensor. At inference time, we deterministically combine the two streams using $\psi$ to match the first moment of the output, i.e.

$$x_{SG} = (1 - \psi)x^{(d)} + \psi x^{(s)}.$$

(10)

Shallow features alone may be too limited in terms of the semantic information they contain. To enrich their representation, yet preserve their original receptive field, we devise **Global Cue Injection (GCI)** via Adaptive Instance Normalisation (AdIN) [23]. As shown in Fig. 7, we first apply a $1 \times 1$ convolution to the deep feature tensor to double the number of channels. Then, we extract two vectors with global information (i.e. without spatial cues) via Global Max Pooling (GMP). Shown as the left (unshaded) and right (shaded) half of the 1D vector after GMP in Fig. 7, let $z^{(a)}$ and $b^{(a)}$ denote two parts of such a representation, which will be shared by each site in a shallow feature channel. We compute the **augmented** shallow activation $x^{(s)}_{\ast}$ as

$$x^{(s)}_{\ast} = \text{ReLU} \left( z^{(a)} \left( \frac{x^{(s)} - \mu(x^{(s)})}{\sigma(x^{(s)})} \right) + b^{(a)} \right),$$

(11)
where $\mu(\cdot)$ and $\sigma(\cdot)$ are the mean and the standard deviation of each channel of $x^{(s)}$. The updated activation, $x^{(s)^+}$, goes through a $1 \times 1$-convolution and replaces the original $x^{(s)}$ in Eq. (9) and Eq. (10) in the final form of SG. Following [8], the output from SG then passes through a 3-layer decoder.

4. Experiments

4.1. Setup

Dataset. Pascal VOC 2012 [12] is an established benchmark for weakly supervised semantic segmentation and contains 20 object categories. Following the standard practice [3, 28, 57], we augment the original VOC training data with an additional image set provided by Hariharan et al. [15]. In total, we use 10,582 images with image-level annotation for training and 1449 images for validation.

Implementation details. Our model is implemented in PyTorch [39]. We use a WideResNet-38 backbone network [58] provided by [3] (see supplemental material, Sec. F, for experiments with VGG16 [47] and ResNet backbones [16]). We further extend this model to DeepLabv3+ by adding Atrous Spatial Pyramid Pooling (ASPP), a skip connection (with our Stochastic Gate), and the 3-layer decoder [8]. We train our model for 20 epochs with SGD using weight decay $5 \times 10^{-4}$ with momentum 0.9, a constant learning rate of 0.01 for the new (randomly initialised) modules and 0.001 for WideResNet-38 parameters, initialised from ImageNet [11] pre-training. We first train our model for 5 epochs using only the classification loss and switch on the self-supervised segmentation loss for the remaining 15 epochs. We use inference with multi-scale inputs [7] and remove masks for classes with classifier confidence < 0.1.

Data augmentation. Following common practice [3, 57], we use random rescaling (in the (0.9, 1.0) range w.r.t. the original image area), horizontal flipping, colour jittering, and train our model on random crops of size $321 \times 321$.

where $\mu(\cdot)$ and $\sigma(\cdot)$ are the mean and the standard deviation of each channel of $x^{(s)}$. The updated activation, $x^{(s)^+}$, goes through a $1 \times 1$-convolution and replaces the original $x^{(s)}$ in Eq. (9) and Eq. (10) in the final form of SG. Following [8], the output from SG then passes through a 3-layer decoder.

4. Experiments

4.1. Setup

Dataset. Pascal VOC 2012 [12] is an established benchmark for weakly supervised semantic segmentation and contains 20 object categories. Following the standard practice [3, 28, 57], we augment the original VOC training data with an additional image set provided by Hariharan et al. [15]. In total, we use 10,582 images with image-level annotation for training and 1449 images for validation.

Implementation details. Our model is implemented in PyTorch [39]. We use a WideResNet-38 backbone network [58] provided by [3] (see supplemental material, Sec. F, for experiments with VGG16 [47] and ResNet backbones [16]). We further extend this model to DeepLabv3+ by adding Atrous Spatial Pyramid Pooling (ASPP), a skip connection (with our Stochastic Gate), and the 3-layer decoder [8]. We train our model for 20 epochs with SGD using weight decay $5 \times 10^{-4}$ with momentum 0.9, a constant learning rate of 0.01 for the new (randomly initialised) modules and 0.001 for WideResNet-38 parameters, initialised from ImageNet [11] pre-training. We first train our model for 5 epochs using only the classification loss and switch on the self-supervised segmentation loss for the remaining 15 epochs. We use inference with multi-scale inputs [7] and remove masks for classes with classifier confidence < 0.1.

Data augmentation. Following common practice [3, 57], we use random rescaling (in the (0.9, 1.0) range w.r.t. the original image area), horizontal flipping, colour jittering, and train our model on random crops of size $321 \times 321$.

4.2. Ablation study

Focal mask penalty. Following the intuition from [34], the focal mask penalty emphasises training on the current failure cases, i.e. small (large) masks for the classes present (absent) in the image. Recall from Eq. (5) that $\lambda$ controls the penalty magnitude, while $p$ is the discounting rate for better-off image samples. We aim to verify if the “focal” aspect of the mask penalty provides advantages over the baseline penalty (i.e. $p = 0$). Table 1 summarises the results.

First, we find that the focal version of the mask penalty improves the segmentation quality of the baseline. This improvement, maximised with $p = 5$ and $\lambda = 0.1$, is tangible, yet comes at a negligible computational cost. Second, we observe that increasing $\lambda$ tends to increase the segmentation accuracy. While changing $\lambda$ from 0.01 to 0.001 leads to higher recall on average, it has a detrimental effect on precision. Lastly, we also find that moderate positive values of $p$ in conjunction with CRF refinement lead to more sizeable gains in mask quality: with $p = 3$, $\lambda = 0.01$ we achieve 62.2% IoU, whereas the highest IoU with $p = 0$ is only 60.5% (reached with $\lambda = 0.01$). However, higher values of $p$ do not benefit from CRF processing (e.g. 50.8% with $p = 5$, $\lambda = 0.1$). Hence, $p = 3$ strikes the best balance between the model accuracy with and without using a CRF. Note that removing the mask penalty, $\psi_{\text{unfocal}}$, leads to an expected drop in recall, reaching only 56.6% IoU.

Pixel-Adaptive Mask Refinement (PAMR). Recall from Sec. 3.3 that PAMR aims to improve the quality of the original coarse masks w.r.t. local consistency to provide self-supervision for segmentation. Here, we verify (i) the im-

Figure 7. Concept illustration of the Stochastic Gate. All rectangular blocks are tensors of the same size. The baseline model from DeepLabv3+ [8] is shown in red: the output from ASPP is augmented via a skip connection from conv3 and the result, $x^{(d)}$, passes directly through the decoder. Shown in blue, our modification (GCI) infuses global cues extracted from the deep features into the shallow features via AdIN [23]. The enriched shallow and the deep features are then combined using Eq. (9) at training and Eq. (10) at inference time.

Table 1. Ablation study on Pascal VOC. We study the role of (a) the focal mask penalty, (b) the Pixel-Adaptive Mask Refinement, and (c) the Stochastic Gate.

<table>
<thead>
<tr>
<th>Config</th>
<th>IoU</th>
<th>IoU (+ CRF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\psi = 0.5$</td>
<td>59.4</td>
<td>62.2</td>
</tr>
<tr>
<td>$\psi = 0.5$ w/o GCI</td>
<td>59.8</td>
<td>60.9</td>
</tr>
<tr>
<td>$\psi = 0.3$</td>
<td>59.7</td>
<td>62.7</td>
</tr>
<tr>
<td>$\psi = 0.3$ w/o GCI</td>
<td>57.7</td>
<td>60.3</td>
</tr>
<tr>
<td>w/o SG</td>
<td>55.6</td>
<td>57.5</td>
</tr>
<tr>
<td>Deterministic Gate</td>
<td>57.5</td>
<td>57.7</td>
</tr>
</tbody>
</table>

(a) IoU (val.%) w.r.t. focal mask penalty. We fix $\psi = 0.5$ w/ GCI for SG and use PAMR kernel [1, 2, 4, 8, 12, 24].

(b) IoU (val.%) w.r.t. Pixel-Adaptive Mask Refinement. We fix $\psi = 0.5$ w/ GCI for SG and set $p = 3$, $\lambda = 0.01$.
show that PAMR is a crucial component in our self-supervised model, as the segmentation accuracy drops markedly from 59.4% to 31.8% without it. We find further that the size of the kernel also affects the accuracy. This is expected, since small receptive fields (dilations 1–2–4–8 in Table 1b) are insufficient to revise the boundaries of the coarse masks that typically exhibit large deviations from the object boundaries. The results with larger receptive fields of the affinity kernel further support this intuition: increasing the dilation of the largest 3×3-kernel to 24 attains the best mask quality compared to the smaller affinity kernels. Furthermore, we observe that varying the kernel shape does not have such a drastic effect; the change from 1–3–6–9–12–16 to 1–2–4–8–12–16 only leads to small accuracy changes. This is desirable in practice as sensitivity to these minor details would imply that our architecture overfits to particularities in the data [52].

Stochastic Gate (SG). The intention of the SG, introduced in Sec. 3.4, is to counter overfitting to the errors contained in the pseudo supervision. Here, there are four baselines we aim to verify: (i) disabling SG; (ii) combining \( x^{(d)} \) and \( x^{(s)} \) deterministically (i.e., \( r \equiv \psi \)) in Eq. (9); (iii) the role of the Global Cue Injection (GCI); and (iv) the effect of the mixing rate \( \psi \). These results are summarised in Table 1c. Evidently, SG is crucial, since disabling it substantially weakens the mask accuracy (from 59.8% to 55.6% IoU). The stochastic nature of SG is also important: simply summing up \( x^{(d)} \) and \( x^{(s)} \) (we used \( r \equiv \psi = 0.5 \)) yields inferior mask IoU (57.5% vs. 59.8%). In our model comparison with both \( \psi = 0.5 \) and \( \psi = 0.3 \), we find that the model with GCI tends to provide superior results. However, the model without GCI can be as competitive given a particular choice of \( \psi \) (e.g., 0.5). In this case, the model with GCI usually has higher recall, while the model without it has higher precision. Since CRFs tend to increase the precision provided sufficient mask support, the model with GCI should therefore profit more from this refinement. We confirmed this and observed a more sizeable improvement of the model with GCI (59.4 vs. 62.2% IoU). Additionally, we found GCI to deliver more stable results for different \( \psi \), which can alleviate parameter fine-tuning in practice.

4.3. Comparison to the state of the art

Setup. Here, our model uses SG with GCI, \( \psi = 0.3 \), the focal penalty with \( p = 3 \) and \( \lambda = 0.01 \), and PAMR with 10 iterations and a 1–2–4–8–12–24 affinity kernel.

Mask quality. Recall that the majority of recent work, e.g., [24, 30, 31, 57], additionally trains a separate fully supervised segmentation network from the pseudo ground truth. To evaluate the quality of such pseudo supervision generated by our model, we use image-level ground-truth labels to remove any masks for classes that are not present in the image (for this experiment only). The results in Table 2 show that using our single-stage mask output as pseudo supervision gener-
Figure 8. **Qualitative results on PASCAL VOC.** We show example segmentations from our method (left), the result of CRF post-processing (middle), and the ground truth (right). Our method produces masks of high quality under a variety of challenging conditions.

### Segmentation accuracy

Table 3 provides a comparative overview w.r.t. the state of the art. Since image-level labels are generally not available at test time, we do not perform any mask pruning here (unlike Table 2). In the setting of image-level supervision, IRN [2] and SSDD [45] are the only methods with higher IoU than ours. Both methods are multi-stage; they are trained in at least three stages. IRNet [2] trains an additional segmentation network on pseudo labels to eventually outperform our method by only 0.5% IoU. Recall that SSDD is essentially a post-processing approach: it refines the masks from AffinityNet [3] (with 63.7% IoU) using an additional network and further employs a cascade of two networks to revise the masks. This strategy improves over our results by only 1.2% IoU, yet at the cost of a considerable increase in model complexity.

Our single-stage method is also competitive with JointSaliency [60], which uses a more powerful backbone [22] and saliency supervision. The recent Frame-to-Frame system [31] is also supervised with saliency and is trained on additional 15K images mined from videos, which requires state-of-the-art optical flow [50]. By contrast, our approach is substantially simpler since we train one network in a single shot. Nevertheless, we surpass a number of multi-stage methods that use additional data and saliency supervision [5, 24, 28, 37, 56]. We significantly improve over previous single-stage methods [38, 41, 43], as well as outperform the single-stage WebCrawl [18], which relies on additional training data and needs multiple forward passes through its class-agnostic decoder. Our model needs neither and infers masks for all classes in one pass.

Note that training a standalone segmentation network on our pseudo labels is a trivial extension, which we omit here in view of our practical goals. However, we still provide these results in the supplemental material (Sec. E), in fact, achieving state of the art in a multi-stage setup as well.

### Qualitative analysis

From the qualitative results in Fig. 8, we observe that our method produces segmentation masks that align well with object boundaries. Our model exhibits good generalisation to challenging scenes with varying object scales and semantic content. Common failure modes of our segmentation network are akin to those of fully supervised methods: segmenting fine-grained details (e.g., bicycle wheels), mislabelling under conditions of occlusions (e.g., leg of the cyclist vs. bicycle), and misleading appearance cues (e.g., low contrast, similar texture).

### 5. Conclusion

In this work, we proposed a practical approach to weakly supervised semantic segmentation, which comprises a single segmentation network trained in one round. To ensure local consistency, semantic fidelity, and completeness of the segmentation masks, we introduced a new class aggregation function, a local mask refinement module, and a stochastic gate. Our approach is astonishingly effective despite its simplicity. Specifically, it yields segmentation accuracy on par with the state of the art and outperforms a range of recent multi-stage methods relying on additional training data and saliency supervision. We expect that our model can also profit from auxiliary supervision and suit the needs of downstream tasks without considerable deployment effort.
References


