Potts Relaxations and Soft Self-labeling for Weakly-Supervised Segmentation

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

We consider weakly supervised segmentation where only a fraction of pixels have ground truth labels (scribbles) and focus on a self-labeling approach where soft pseudo-labels on unlabeled pixels optimize some relaxation of the standard unsupervised CRF/Potts loss. While WSSS methods can directly optimize CRF losses via gradient descent, prior work suggests that higher-order optimization can lead to better network training by jointly estimating pseudo-labels, e.g. using discrete graph cut sub-problems. The inability of hard pseudo-labels to represent class uncertainty motivates the relaxed pseudo-labeling. We systematically evaluate standard and new CRF relaxations, neighborhood systems, and losses connecting network predictions with soft pseudo-labels. We also propose a general continuous sub-problem solver for such pseudo-labels. Soft self-labeling loss combining the log-quadratic Potts relaxation and collision cross-entropy achieves state-of-the-art and can outperform full pixel-precise supervision on PASCAL.

1 Introduction

Full supervision for semantic segmentation requires thousands of training images with complete pixel-accurate ground truth masks. Their high costs explain the interest in weakly-supervised approaches based on image-level class *tags* [21, 4], pixel-level *scribbles* [26, 36, 35], or *boxes* [23]. This paper is focused on weak supervision with *scribbles*, which we also call *seeds* or *partial masks*. While only slightly more expensive than image-level class tags, scribbles on less than 3% of pixels were previously shown to achieve accuracy approaching full supervision without any modifications of the segmentation models. In contrast, tag supervision typically requires highly specialized systems and complex multi-stage training procedures, which are hard to reproduce. Our interest in the scribble-based approach is motivated by its practical simplicity and mathematical clarity. The corresponding methodologies are focused on the design of unsupervised or self-supervised loss functions and stronger optimization algorithms. The corresponding solutions are often general and can be used in different weakly-supervised applications.

1.1 Scribble-supervised segmentation

Assume that a set of image pixels is denoted by Ω and a subset of pixels with ground truth labels is $S \subset \Omega$, which we call *seeds* or *scribbles* as subset S is typically marked by mouse-controlled UI for image annotations, e.g. see seeds over an image in Fig.7(a). The ground truth label at any given pixel $i \in S$ is an integer

$$\bar{y}_i \in \{1, \dots, K\} \tag{1}$$

where K is the number of classes including the background. Without much ambiguity, it is convenient to use the same notation $\bar{y}_i$ for the equivalent *one-hot* distribution

$$\bar{y}_i \;\equiv\; (\bar{y}_i^1, \dots, \bar{y}_i^K) \in \Delta_{0,1}^K \qquad \text{for} \quad \bar{y}_i^k := [k = \bar{y}_i] \;\in \{0, 1\} \tag{2}$$

where $[\,\cdot\,]$ is the *True* operator for the condition inside the brackets. Set $\Delta_{0,1}^K$ represents K possible one-hot distributions, which are vertices of the K-class *probability simplex*

$$\Delta^K \;:=\; \{p = (p^1, \dots, p^K) \mid p^k \geq 0,\; \sum_{k=1}^{K} p^k = 1\}$$

representing all K-categorical distributions. The context of specific expressions should make it obvious if $\bar{y}_i$ is a class index (1) or the corresponding one-hot distribution (2).

Loss functions for weakly supervised segmentation with scribbles typically use *negative log-likelihoods* (NLL) over scribbles $i \in S \subset \Omega$ with ground truth labels $\bar{y}_i$

$$-\sum_{i \in S} \ln \sigma_i^{\bar{y}_i} \tag{3}$$

where $\sigma_i = (\sigma_i^1, \dots, \sigma_i^K) \in \Delta^K$ is the model prediction at pixel i. This loss is a standard in full supervision where the only difference is that $S = \Omega$ and usually, no other losses are needed for training. However, in a weakly supervised setting the majority of pixels are unlabeled, and unsupervised losses are needed for $i \notin S$.

The most common unsupervised loss in image segmentation is the Potts model and its relaxations. It is a pairwise loss defined on pairs of *neighboring* pixels $\{i, j\} \in \mathcal{N}$ for a given neighborhood system $\mathcal{N} \subset \Omega \times \Omega$, typically corresponding to the *nearest-neighbor* grid (NN) [6, 17], or other *sparse* (SN) [38] and *dense* neighborhoods (DN) [22]. The original Potts model is defined for discrete segmentation variables, e.g. as in

$$\sum_{\{i,j\} \in \mathcal{N}} P(\sigma_i, \sigma_j) \qquad \text{where} \quad P(\sigma_i, \sigma_j) = [\sigma_i \neq \sigma_j]$$

assuming integer-valued one-hot predictions $\sigma_i \in \Delta_{0,1}^K$. This *regularization* loss encourages smoothness between the pixels. Its popular *self-supervised* variant is

$$P(\sigma_i, \sigma_j) = w_{i,j} \cdot [\sigma_i \neq \sigma_j]$$

where pairwise affinities w_{ij} are based on local intensity edges [6, 17, 22]. Of course, in the context of network training, one should use relaxations of P applicable to (soft) predictions $\sigma_i \in \Delta^K$. Many types of its relaxation [33, 42] were studied in segmentation, e.g. *quadratic* [17], *bi-linear* [36], *total variation* [32, 8], and others [14].

Another unsupervised loss highly relevant for training segmentation networks is the entropy of predictions, which is also known as *decisiveness* [7, 18]

$$\sum_i H(\sigma_i)$$

where H is the Shannon's entropy function. This loss can improve generalization and the quality of representation by moving (deep) features away from the decision boundaries. Widely known in the context of unsupervised or semi-supervised classification, this loss also matters in weakly-supervised segmentation where it is used explicitly or implicitly[1].

Other unsupervised losses (e.g. contrastive), clustering criteria (e.g. K-means), or specialized architectures can be found in weakly-supervised segmentation [39, 31, 20, 9]. However, a lot can be achieved simply by combining the basic losses discussed above

$$L_{ws}(\sigma) \;:=\; -\sum_{i \in S} \ln \sigma_i^{\bar{y}_i} \;+\; \eta \sum_{i \notin S} H(\sigma_i) \;+\; \lambda \sum_{ij \in \mathcal{N}} P(\sigma_i, \sigma_j) \tag{4}$$

which can be optimized directly by gradient descent [36, 38] or using *self-labeling* techniques [26, 28, 27] incorporating optimization of auxiliary *pseudo-labels* as sub-problems.

[1] Interestingly, a unary decisiveness-like term is the difference between convex quadratic and *tight*, but non-convex, bi-linear relaxations [33, 27] of the discrete pairwise Potts model.

1.2 Soft pseudo-labels: motivation and contributions

We observe that self-labeling with hard pseudo-labels y_i, which is discussed in the Appendix A, is inherently limited as such labels can not represent the uncertainty of class estimates at unlabeled pixels $i \in \Omega \backslash S$. Instead, we focus on *soft* pseudo-labels

$$y_i \;\; = \;\; (y_i^1, \ldots, y_i^K) \, \in \Delta^K \tag{5}$$

which are general categorical distributions p over K-classes. It is possible that the estimated pseudo-label y_i in (5) could be a one-hot distribution, which is a vertex of Δ^K. In such a case, one can treat y_i as a class index, but we avoid this in the main part of our paper starting Section 2. However, the ground truth labels $\bar{y}_i$ are always hard and we use them either as indices (1) or one-hot distributions (2), as convenient.

Soft pseudo-labels can be found in prior work on weakly-supervised segmentation [25, 41] using the "soft proposal generation". In contrast, we formulate soft self-labeling as a principled optimization methodology where network predictions and soft pseudo-labels are variables in a joint loss, which guarantees convergence of the training procedure. Our pseudo-labels are auxiliary variables for ADM-based [5] splitting of the loss (4) into two simpler optimization sub-problems: one focused on the Potts model over unlabeled pixels, and the other on the network training. While similar to [28], instead of hard, we use soft auxiliary variables for the Potts sub-problem. Our work can be seen as a study of the relaxed Potts sub-problem in the context of weakly-supervised semantic segmentation. The related prior work is focused on discrete solvers fundamentally unable to represent class estimate uncertainty. Our contributions can be summarized as follows:

- convergent *soft self-labeling* framework based on a simple joint self-labeling loss
- systematic evaluation of Potts relaxations and (cross-) entropy terms in our loss
- state-of-the-art in scribble-based semantic segmentation that does not require any modifications of semantic segmentation models and is easy to reproduce
- using the same segmentation model, our self-labeling loss with 3% scribbles may outperform standard supervised cross-entropy loss with full ground truth masks.

2 Our soft self-labeling approach

First, we apply ADM splitting [5] to weakly supervised loss (4) to formulate our self-labeling loss (6) incorporating additional soft auxiliary variables, i.e. pseudo-labels (5). It is convenient to introduce pseudo-labels y_i on all pixels in Ω even though a subset of pixels (seeds) $S \subset \Omega$ have ground truth labels $\bar{y}_i$. We will simply impose a constraint that pseudo-labels and ground truth labels agree on S. Thus, we assume the following set of pseudo-labels

$$Y_\Omega := \{y_i \in \Delta^K \,|\, i \in \Omega, \text{ s.t. } y_i = \bar{y}_i \text{ for } i \in S\}.$$

We split the terms in (4) into two groups: one includes NLL and entropy H terms keeping the original prediction variables σ_i and the other includes the Potts relaxation P replacing σ_i with auxiliary variables y_i. This transforms loss (4) into expression

$$-\sum_{i \in S} \ln \sigma_i^{\bar{y}_i} \; + \; \eta \sum_{i \notin S} H(\sigma_i) \; + \; \lambda \sum_{ij \in \mathcal{N}} P(y_i, y_j)$$

equivalent to (4) assuming equality $\sigma_i = y_i$. The standard approximation is to incorporate constraint $\sigma_i \approx y_i$ directly into the loss, e.g. using KL-divergence. For simplicity, we use weight η for $KL(\sigma_i, y_i)$ to combine it with $H(\sigma_i)$ into a single cross-entropy term

$$-\sum_{i \in S} \ln \sigma_i^{\bar{y}_i} \; + \; \underbrace{\eta \sum_{i \notin S} H(\sigma_i) \; + \; \eta \sum_{i \notin S} KL(\sigma_i, y_i)}_{\eta \sum_{i \notin S} H(\sigma_i, y_i)} \; + \lambda \sum_{ij \in \mathcal{N}} P(y_i, y_j)$$

defining joint *self-labeling loss* for both predictions σ_i and pseudo-labels y_i

$$L_{self}(\sigma, y) \;\; := \;\; -\sum_{i \in S} \ln \sigma_i^{\bar{y}_i} \; + \; \eta \sum_{i \notin S} H(\sigma_i, y_i) \; + \; \lambda \sum_{ij \in \mathcal{N}} P(y_i, y_j) \tag{6}$$

bi-linear $\sim$ “graph cut”	**quadratic** $\sim$ “random walker”
$P_{\text{BL}}(p,q) \;:=\; 1 - \; p^\top q$	$P_{\text{Q}}(p,q) \;:=\; \frac{1}{2}\|p-q\|^2$
normalized quadratic	
$P_{\text{NQ}}(p,q) \;:=\; 1 - \frac{p^\top q}{\|p\|\|q\|}$	$\equiv \; \frac{1}{2}\left\| \frac{p}{\|p\|} - \frac{q}{\|q\|} \right\|^2$

Table 1: Second-order Potts relaxations, see Fig.1(a,b,c)

approximating the original weakly supervised loss (4).

Iterative minimization of this loss w.r.t. predictions σ_i (model parameters training) and pseudo-labels y_i effectively breaks the original optimization problem for (4) into two simpler sub-problems, assuming there is a good solver for optimal pseudo-labels. The latter seems plausible since the unary term $H(\sigma_i, y_i)$ is convex for y_i and the Potts relaxations were widely studied in image segmentation for decades.

Section 2.1 discusses standard and new relaxations of the Potts model P. Section 2.2 discusses several robust variants of cross-entropy H for connecting predictions with uncertain (soft) pseudo-labels y_i estimated for unlabeled points $i \in \Omega \backslash S$. Appendix B proposes an efficient general solver for the corresponding pseudo-labeling sub-problems.

2.1 Second-order relaxations of the Potts model

We focus on second-order relaxations for two reasons. First, to manage the scope of this study. Second, this includes several important baseline cases (see Table 1): *quadratic*, the simplest convex relaxation popularized by the *random walker* algorithm [17], and *bi-linear*, which is non-convex but *tight* [33] w.r.t. the original discrete Potts model. The latter implies that optimizing it over relaxed variables will lead to a solution consistent with a discrete Potts solver, e.g. *graph cut* [6]. On the contrary, the quadratic relaxation will produce a significantly different soft solution. We investigate such soft solutions.

Figure 2 shows two examples illustrating local minima for (a) the bi-linear and (b) quadratic relaxations of the Potts loss. In (a) two neighboring pixels attempt to jointly change the common soft label from $y_i = y_j = (1,0,0)$ to $y_i'' = y_j'' = (0,1,0)$, which corresponds to a “move” where the whole object is reclassified from A to B. This move does not violate smoothness within the region represented by the Potts model. But, the soft intermediate state $y_i' = y_j' = (\frac{1}{2}, \frac{1}{2}, 0)$ will prevent this move in bi-linear case

$$P_{\text{BL}}(y_i', y_j') = \frac{1}{2} \quad > \quad 0 = P_{\text{BL}}(y_i, y_j) = P_{\text{BL}}(y_i'', y_j'')$$

while quadratic relaxation assigns zero loss for all states during this move. On the other hand, the example in Figure 2(b) shows a move problematic for the quadratic relaxation. Two neighboring pixels have labels $y_i = (1,0,0)$ and $y_j = (0,0,1)$ corresponding to the boundary of objects A and C. The second object attempts to change from C to B. This move does not affect the discontinuity between two pixels, but quadratic relaxation prefers that the second object is stuck in the intermediate state $y_j' = (0, \frac{1}{2}, \frac{1}{2})$

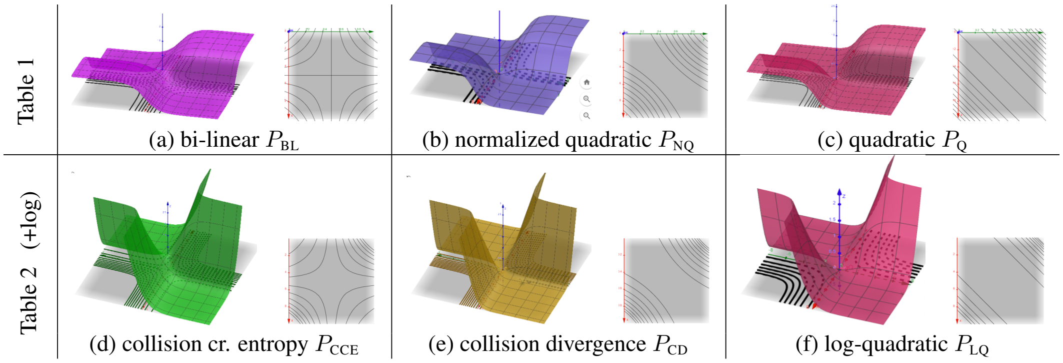
$$P_{\text{Q}}(y_i, y_j') = \frac{3}{4} \quad < \quad 1 = P_{\text{Q}}(y_i, y_j) = P_{\text{Q}}(y_i, y_j'')$$

while bi-linear relaxation $P_{\text{BL}}(y_i, y_j) = 1$ remains constant as y_j changes.

We propose a new relaxation, *normalized quadratic* in Table 1. Normalization leads to equivalence between quadratic and bi-linear formulations combining their benefits. As easy to check, normalized

collision cross entropy	**log-quadratic**
$P_{\text{CCE}}(p,q) \;:=\; -\ln p^\top q$	$P_{\text{LQ}}(p,q) \;:=\; -\ln\left(1 - \frac{\|p-q\|^2}{2}\right)$
collision divergence	
$P_{\text{CD}}(p,q) \;:=\; -\ln \frac{p^\top q}{\|p\|\|q\|}$	$\equiv \; -\ln\left(1 - \frac{1}{2}\left\| \frac{p}{\|p\|} - \frac{q}{\|q\|} \right\|^2\right)$

Table 2: Log-based Potts relaxations, see Fig.1(d,e,f)

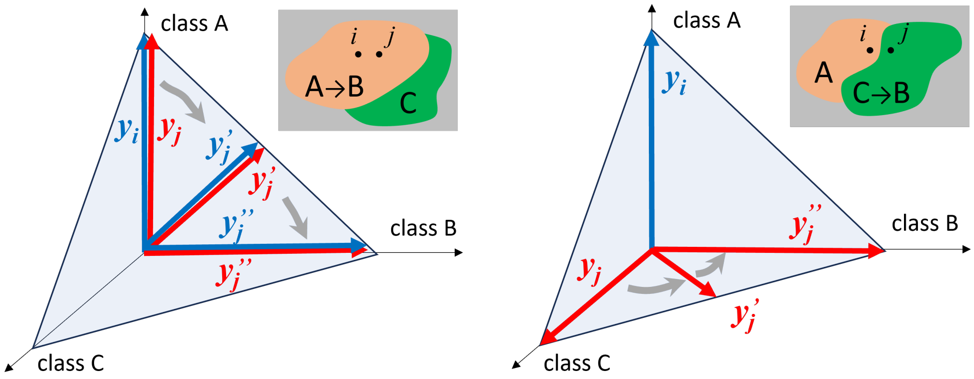


(a) bi-linear P_{BL} (b) normalized quadratic P_{NQ} (c) quadratic P_{Q}

(d) collision cr. entropy P_{CCE} (e) collision divergence P_{CD} (f) log-quadratic P_{LQ}

Figure 1: Second-order Potts relaxations in Tables 1 and 2: interaction potentials P for pairs of predictions (σ_i, σ_j) in (4) or pseudo-labels (y_i, y_j) in (6) are illustrated for $K = 2$ when each prediction σ_i or label y_i, i.e. distribution in Δ^2, can be represented by a single scalar as $(x, 1-x)$. The contour maps are iso-levels of $P((x_i, 1-x_i), (x_j, 1-x_j))$ over domain $(x_i, x_j) \in [0,1]^2$. The 3D plots above illustrate the potentials P as functions over pairs of "logits" $(l_i, l_j) \in \mathbb{R}^2$ where each scalar logit l_i defines binary distribution $(x_i, 1-x_i)$ for $x_i = \frac{1}{1+e^{-2l_i}} \in [0,1]$.

quadratic relaxation P_{NQ} does not have local minima in both examples of Figure 2. Table 2 also proposes "logarithmic" versions of the relaxations in Table 1 composing them with function $-\ln(1-x)$. As illustrated by Figure 1, the logarithmic versions in (d-f) addresses the "vanishing gradients" evident in (a-c).

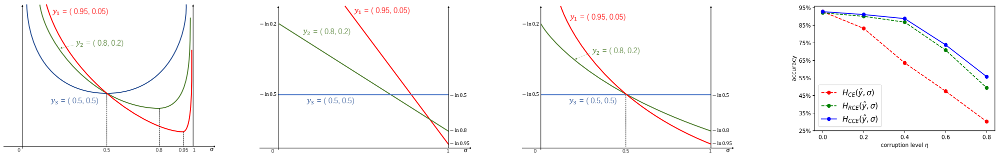


(a) both pixels change (A $\rightarrow$ B) (b) only pixel q changes (C $\rightarrow$ B)

Figure 2: Examples of "moves" for neighboring pixels $\{i, j\} \in \mathcal{N}$. Their (soft) pseudo-labels y_i and y_j are illustrated on the probability simplex Δ^K for $K = 3$. In (a) both pixels i and j are inside a region/object changing its label from A to B. In (b) pixels i and j are on the boundary between two regions/objects; one is fixed to class A and the other changes from class C to B.

2.2 Cross-entropy and soft pseudo-labels

Shannon's cross-entropy $H(y, \sigma)$ is the most common loss for training network predictions σ from ground truth labels y in the context of classification, semantic segmentation, etc. However, this loss may not be ideal for applications where the targets y are soft categorical distributions representing various forms of class uncertainty. For example, this paper is focused on scribble-based segmentation where the ground truth is not known for most of the pixels, and the network training is done jointly with estimating *pseudo-labels* y for the unlabeled pixels. In this case, soft labels y are distributions representing class uncertainty. We observe that if such y is used as a target in $H(y, \sigma)$, the network is trained to reproduce the uncertainty, see Figure 3(a). This motivates the discussion of alternative "cross-entropy" functions where the quotes indicate an informal interpretation of this information-theoretic concept. Intuitively, such functions should encourage decisiveness, as well as proximity



(a) standard $H_{\text{CE}}(y, \sigma)$ (b) reverse $H_{\text{RCE}}(y, \sigma)$ (c) collision $H_{\text{CCE}}(y, \sigma)$ (d) empirical comparison

Figure 3: Illustration of cross-entropy functions: (a) standard (7), (b) reverse (8), and (c) collision (9). (d) shows the empirical comparison on the robustness to label uncertainty. The test uses ResNet-18 architecture on fully-supervised *Natural Scene* dataset [30] where we corrupted some labels. The horizontal axis shows the percentage η of training images where the correct ground truth labels were replaced by a random label. All losses trained the model using soft target distributions $\hat{y} = \eta * u + (1-\eta) * y$ representing the mixture of one-hot distribution y for the observed corrupt label and the uniform distribution u, following [29]. The vertical axis shows the test accuracy. Training with the reverse and collision cross-entropy is robust to much higher levels of label uncertainty.

between the predictions and pseudo-labels, but avoid mimicking the uncertainty in both directions: from soft pseudo-labels to predictions and vice-versa. We show that the last property can be achieved in a probabilistically principled manner. The following three paragraphs discuss different cross-entropy functions that we study in the context of our self-labeling loss (6).

Standard cross-entropy provides the obvious baseline for evaluating two alternative versions that follow. For completeness, we include its mathematical definition

$$H_{\text{CE}}(y_i, \sigma_i) \quad = \quad H(y_i, \sigma_i) \quad \equiv \quad -\sum_k y_i^k \ln \sigma_i^k \tag{7}$$

and remind the reader that this loss is primarily used with hard or one-hot labels, in which case it is also equivalent to NLL loss $-\ln \sigma_i^{y_i}$ previously discussed for ground truth labels (3). As mentioned earlier, Figure 3(a) shows that for soft pseudo-labels like $y = (0.5, 0.5)$, it forces predictions to mimic or replicate the uncertainty $\sigma \approx y$. In fact, label $y = (0.5, 0.5)$ just tells that the class is unknown and the network should not be supervised by this point. This problem manifests itself in the poor performance of the standard cross-entropy (7) in our experiment discussed in Figure 3 (d) (red curve).

Reverse cross-entropy switches the order of the label and prediction in (7)

$$H_{\text{RCE}}(y_i, \sigma_i) \quad = \quad H(\sigma_i, y_i) \quad \equiv \quad -\sum_k \sigma_i^k \ln y_i^k \tag{8}$$

which is not too common. Indeed, Shannon's cross-entropy is not symmetric and the first argument is normally the *target* distribution and the second is the *estimated* distribution. However, in our case, both distributions are estimated and there is no reason not to try the reverse order. It is worth noting that our self-labeling formulation (6) suggests that reverse cross-entropy naturally appears when the ADM approach splits the decisiveness and fairness into separate sub-problems. Moreover, as Figure 3(b) shows, in this case, the network does not mimic uncertain pseudo-labels, e.g. the gradient of the blue line is zero. The results for the reverse cross-entropy in Figure 3 (d) (green) are significantly better than for the standard (red). Unfortunately, now pseudo-labels y mimic the uncertainty in predictions σ.

Collision cross-entropy resolves the problem in a principled way. We define it as

$$H_{\text{CCE}}(y_i, \sigma_i) \quad \equiv \quad -\ln \sum_k \sigma_i^k y_i^k \quad \equiv \quad -\ln \sigma^\top y \tag{9}$$

which is symmetric w.r.t. pseudo-labels and predictions. The dot product $\sigma^\top y$ can be seen as a probability that random variables represented by the distribution σ, the prediction class C, and the distribution y, the unknown true class T, are equal. Indeed,

$$\Pr(C = T) = \sum_k Pr(C = k) \Pr(T = k) = \sigma^\top y.$$

Loss (9) maximizes this "collision" probability rather than the constraint $\sigma = y$. Figure 3(c) shows no mimicking of uncertainty (blue line). However, unlike reverse cross-entropy, this is also valid when y

is estimated from uncertain predictions σ since (9) is symmetric. This leads to the best performance in Figure 3 (d) (blue). Our extensive experiments are conclusive that collision cross-entropy is the best option for H in self-labeling loss (6).

3 Experiments

We conducted comprehensive experiments to demonstrate the choice of each element (cross-entropy, pairwise term, and neighborhood) in the loss and compare our method to the state-of-the-art. In Section 3.1, quantitative results are shown to compare different Potts relaxations. The qualitative examples are shown in Figure 7. Then we compare several cross-entropy terms in Section 3.2. Besides, we also compare our soft self-labeling approach on the nearest and dense neighborhood systems in Section 3.3. We summarized the results in Section 3.4. In the last section, we show that our method achieves the SOTA and even can outperform the fully-supervised method. More details on the dataset, implementation, and additional experiments are given in Appendix C.

3.1 Comparison of Potts relaxations

To compare different Potts relaxations under the self-labeling framework, we need to choose one cross-entropy term. Motivated by the properties and empirical results shown in Section 3.2, we use H_{CCE}. The neighborhood system is the nearest neighbors. The quantitative results are in Table 3. First, One can see that the pairwise terms with logarithm are better than those without the logarithm because the logarithm may help with the gradient vanishing problem in softmax operation. Moreover, the logarithm does not like abrupt change across the boundaries, so the transition across the boundaries is smoother (see Figure 7 in the appendix.). Note that it is reasonable to have higher uncertainty around the boundaries. Second, the results prefer the normalized version, which confirms the points made in Figure 2. Third, the simplest quadratic formulation P_{Q} can be a fairly good starting point to obtain decent results. Additionally, we specifically test $H_{\text{Q}} + P_{\text{Q}}$ due to the existing closed-form solution [1, 17]. Since the pseudo-labels generated from this formula tend to be overly soft, we explicitly add entropy terms during the training of network parameters and the mIoU goes up to 68.97% from 67.8%.

	scribble length ratio				
	0	0.3	0.5	0.8	1.0
P_{BL}	56.42	61.74	63.81	65.73	67.24
P_{NQ}	59.01	65.53	67.80	70.63	71.12
P_{Q}	58.92	65.34	67.81	70.43	71.05
P_{CCE}	56.40	61.82	63.81	65.81	67.41
P_{CD}	59.04	65.52	67.84	70.93	71.22
P_{LQ}	59.03	65.44	67.81	70.80	71.21

Table 3: Comparison of Potts relaxations with self-labeling. mIoUs on validation set are shown here.

3.2 Comparison of cross-entropy terms

In this section, we compare different cross-entropy terms while fixing the pairwise term to P_{Q} due to its simplicity and using the nearest neighborhood system. The results are shown in Figure 4. One can see that H_{CCE} performs the best consistently across different supervision levels, i.e. scribble lengths. Both H_{CCE} and H_{RCE} are consistently better than standard H_{CE} with a noticeable margin because they are more robust, as explained in Section 2.2, to the uncertainty in soft pseudo-labels when optimizing network parameters. We also test the performance of using $H_{\text{CCE}} + P_{\text{Q}}$ with hard pseudo-labels obtained via the $argmax$ operation on the soft ones. The mIoU on the validation set is 69.8% under the full scribble-length supervision.

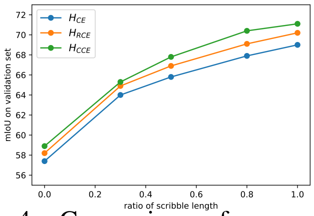


Figure 4: Comparison of cross-entropy terms.

3.3 Comparison of neighborhood systems

Until now, we only used the four nearest neighbors for the pairwise term. In this section, we also use the dense neighborhood and compare the results under the self-labeling framework.

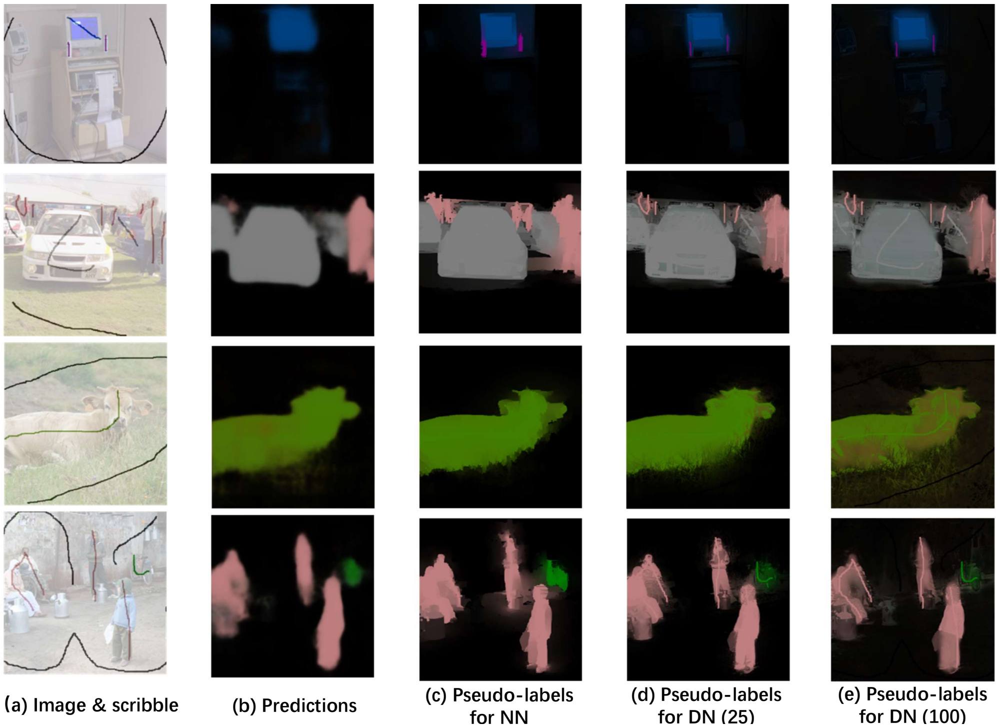

(a) Image & scribble
(b) Predictions
(c) Pseudo-labels for NN
(d) Pseudo-labels for DN (25)
(e) Pseudo-labels for DN (100)

Figure 5: Pseudo-labels generated from given network predictions using different neighborhoods.

Firstly, to optimize the pseudo-labels for the dense neighborhood, we still use the gradient descent technique as detailed in Appendix B. The gradient computation employs the bilateral filtering technique following [35]. For the pairwise term, we use P_{Q}. The cross-entropy term is H_{CCE}. Note that the bilateral filtering technique only supports quadratic pairwise terms, i.e. P_{BL} and P_{Q}. Since P_{BL} leads to hard solutions, P_{Q} is the only practical choice for soft self-labeling. We obtained 71.1% mIoU on nearest neighbors while only getting 67.9% on dense neighborhoods (bandwidth is 100). Some qualitative results are shown in Figure 5. Clearly from this figure one can see that a larger neighborhood size induces lower-quality pseudo-labels. A possible explanation is that the Potts model gets closer to cardinality/volume potentials when the neighborhood size becomes larger [37]. The nearest neighborhood is better for edge alignment and thus produces cleaner results.

3.4 Soft self-labeling vs. hard self-labeling vs. gradient descent

In this section, we give a summary in Table 4 as to what is the best framework for the WSSS based on losses regularized by the Potts model. Firstly, to directly optimize the network parameters via stochastic gradient descent on the regularized loss, one needs a larger neighborhood size. One possible explanation is that a larger neighborhood size induces a smoother Potts model and it helps the gradient descent [28]. However, larger neighborhood size is not preferred in the self-labeling framework. If we use Potts model on nearest neighborhoods, the self-labeling optimization should be applied and one should use soft pseudo-labels instead of hard ones. Note that with proper optimization the advantage of the Potts model on small neighborhood size can show up. In Figure 6, we also compare these approaches across different scribble lengths.

		$\mathcal{N}$	
		NN	DN
GD		67.0	69.5* [36]
SL	hard	69.6* [27]	63.1 [26]
	soft	**71.1**	67.9

Table 4: Summary of comparisons. "$*$" stands for the reproduced results from their code repository.

3.5 Comparison to SOTA

In this section, we use a different network architecture, ResNet101, to fairly compare our method with the current state-of-the-art. We only compare the results before applying any post-processing steps. The results are shown in Table 5. Note that our results can outperform the fully-supervised method when using 12 as the batch size. We also observe that a larger batch size usually improves the results quite a lot. Our results with 12 batch size can outperform several SOTA methods which use 16 batch size.

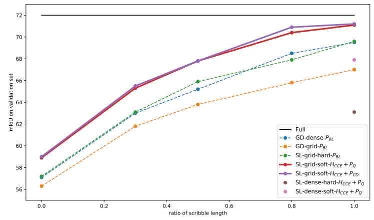


Figure 6: Comparison of different methods using Potts relaxations. The architecture is DeeplabV3+ with the backbone MobileNetV2.

Method	Architecture	Batchsize	Optimization			$\mathcal{N}$	mIoU
			GD	SL			
				hard	soft		
Full supervision							
Deeplab* [12]	V3+	16	✓	-	-	-	78.9
Deeplab* [12]	V3+	12	✓	-	-	-	76.6
Deeplab [11]	V2	12	✓	-	-	-	75.6
Scribble supervision							
Architectural modification							
BPG [39]	V2	10	✓	-	-	-	73.2
URSS [31]	V2	16	✓	-	-	-	74.6
SPML [20]	V2	16	✓	-	-	-	74.2
PSI [41]	V3+	-	-	-	✓	-	74.9
SEMINAR [9]	V3+	12	✓	-	-	-	76.2
TEL [25]	V3+	16	-	-	✓	-	77.1
Loss modification - Potts relaxations							
ScribbleSup [26]	VGG16(V2)	8	-	✓	-	DN	63.1
DenseCRF loss* [36]	V3+	12	✓	-	-	DN	75.8
GridCRF loss* [27]	V3+	12	-	✓	-	NN	75.6
NonlocalCRF loss* [38]	V3+	12	✓	-	-	SN	75.7
$\mathbf{H}_{\text{CCE}} + \mathbf{P}_{\text{Q}}$	V3+	12	-	-	✓	NN	77.5
$\mathbf{H}_{\text{CCE}} + \mathbf{P}_{\text{CD}}$	V3+	12	-	-	✓	NN	**77.7**
$\mathbf{H}_{\text{CCE}} + \mathbf{P}_{\text{CD}}$ (no pretrain)	V3+	12	-	-	✓	NN	76.7
$\mathbf{H}_{\text{CCE}} + \mathbf{P}_{\text{CD}}$	V3+	16	-	-	✓	NN	**78.1**
$\mathbf{H}_{\text{CCE}} + \mathbf{P}_{\text{CD}}$ (no pretrain)	V3+	16	-	-	✓	NN	77.6

Table 5: Comparison to SOTA methods (without CRF postprocessing) on scribble-supervised segmentation. The numbers are mIoU on the validation dataset of Pascal VOC 2012 and use full-length scribble. The backbone is ResNet101 unless stated otherwise. V2: deeplabV2. V3+: deeplabV3+. $\mathcal{N}$: neighborhood. "$*$": reproduced results. GD: gradient descent. SL: self-labeling. "no pretrain" means the segmentation network is not pretrained using cross-entropy on scribbles.

4 Conclusions

This paper proposed a convergent soft self-labeling framework based on a simple well-motivated loss (6) for joint optimization of network predictions and soft *pseudo-labels*. The latter were motivated as auxiliary optimization variables simplifying optimization of weakly-supervised loss (4). Our systematic evaluation of the cross-entropy and the Potts terms in self-labeling loss (6) provides clear recommendations based on the discussed conceptual advantages empirically confirmed by our experiments. Specifically, our work recommends the collision cross-entropy, log-quadratic Potts relaxations, and the earest-neighbor neighborhood. They achieve the best result that may even outperform the fully-supervised method with full pixel-precise masks. Our method does not require any modifications of the semantic segmentation models and it is easy to reproduce. Our general framework and empirical findings can be useful for other weakly-supervised segmentation problems (boxes, class tags, etc.).

References

[1] Multilabel random walker image segmentation using prior models. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, pages 763–770. IEEE, 2005.

[2] 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.

[3] Jiwoon Ahn, Sunghyun Cho, and Suha Kwak. Weakly supervised learning of instance segmentation with inter-pixel relations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2209–2218, 2019.

[4] 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.

[5] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[6] 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*, pages 105–112. IEEE, 2001.

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

[8] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*, 40:120–145, 2011.

[9] Hongjun Chen, Jinbao Wang, Hong Cai Chen, Xiantong Zhen, Feng Zheng, Rongrong Ji, and Ling Shao. Seminar learning for click-level weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6920–6929, 2021.

[10] 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.

[11] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4): 834–848, 2017.

[12] 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.

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

[14] Camille Couprie, Leo Grady, Laurent Najman, and Hugues Talbot. Power watershed: A unifying graph-based optimization framework. *IEEE transactions on pattern analysis and machine intelligence*, 33(7):1384–1399, 2010.

[15] 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.

[16] 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.

[17] Leo Grady. Random walks for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 28(11):1768–1783, 2006.

[18] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. *Advances in neural information processing systems*, 17, 2004.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[20] Tsung-Wei Ke, Jyh-Jing Hwang, and Stella X Yu. Universal weakly supervised segmentation by pixel-to-segment contrastive learning. *arXiv preprint arXiv:2105.00957*, 2021.

[21] 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.

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

[23] 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*.

[24] Jungbeom Lee, Eunji Kim, Sungmin Lee, Jangho Lee, and Sungroh Yoon. Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5267–5276, 2019.

[25] Zhiyuan Liang, Tiancai Wang, Xiangyu Zhang, Jian Sun, and Jianbing Shen. Tree energy loss: Towards sparsely annotated semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16907–16916, 2022.

[26] 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.

[27] Dmitrii Marin and Yuri Boykov. Robust trust region for weakly supervised segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6608–6618, 2021.

[28] Dmitrii Marin, Meng Tang, Ismail Ben Ayed, and Yuri Boykov. Beyond gradient descent for regularized segmentation losses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10187–10196, 2019.

[29] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019.

[30] NSD. Natural Scenes Dataset [NSD]. `https://www.kaggle.com/datasets/nitishabharathi/scene-classification`, 2020.

[31] Zhiyi Pan, Peng Jiang, Yunhai Wang, Changhe Tu, and Anthony G Cohn. Scribble-supervised semantic segmentation by uncertainty reduction on neural representation and self-supervision on neural eigenspace. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7416–7425, 2021.

[32] Thomas Pock, Antonin Chambolle, Daniel Cremers, and Horst Bischof. A convex relaxation approach for computing minimal partitions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 810–817, 2009.

[33] Pradeep Ravikumar and John Lafferty. Quadratic programming relaxations for metric labeling and Markov Random Field MAP estimation. In *The 23rd International Conference on Machine Learning*, page 737–744, 2006.

[34] 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.

[35] Meng Tang, Abdelaziz Djelouah, Federico Perazzi, Yuri Boykov, and Christopher Schroers. Normalized cut loss for weakly-supervised cnn segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1818–1827, 2018.

[36] 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.

[37] Olga Veksler. Efficient graph cut optimization for full crfs with quantized edges. *IEEE transactions on pattern analysis and machine intelligence*, 42(4):1005–1012, 2019.

[38] Olga Veksler and Yuri Boykov. Sparse non-local crf. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4493–4503, 2022.

[39] Bin Wang, Guojun Qi, Sheng Tang, Tianzhu Zhang, Yunchao Wei, Linghui Li, and Yongdong Zhang. Boundary perception guidance: A scribble-supervised semantic segmentation approach. In *IJCAI International joint conference on artificial intelligence*, 2019.

[40] 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.

[41] Jingshan Xu, Chuanwei Zhou, Zhen Cui, Chunyan Xu, Yuge Huang, Pengcheng Shen, Shaoxin Li, and Jian Yang. Scribble-supervised semantic segmentation inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15354–15363, 2021.

[42] Christopher Zach, Christian Häne, and Marc Pollefeys. What is optimized in tight convex relaxations for multi-label problems? In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1664–1671, 2012.

[43] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017.

[44] Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. *ProQuest Number: INFORMATION TO ALL USERS*, 2002.

A Self-labeling and hard pseudo-labels

One argument motivating self-labeling approaches to weakly-supervised segmentation comes from well-known limitations of gradient descent when optimizing the Potts relaxatons, e.g. [28]. But even when using convex Potts relaxations [17, 32, 8], they are combined with the concave entropy term in (4) making their optimization challenging.

Typical self-labeling methods, including one of the first works on scribble-based semantic segmentation [26], introduce a sub-problem focused on the estimation of *pseudo-labels* over unlabeled points, separately from the network training by such labels. Pseudo-labeling is typically done by optimization algorithms or heuristics balancing unsupervised or self-supervised criteria, e.g. the Potts, and proximity to current predictions. Then, network fine-tuning from pseudo-labels and pseudo-labeling steps are iterated.

We denote pseudo-labels y_i slightly differently from the ground truth labels $\bar{y}_i$ by omitting the bar. It is important to distinguish them since the ground truth labels $\bar{y}_i$ for $i \in S$ are given, while the pseudo-labels y_i for $i \in \Omega \backslash S$ are estimated. The majority of existing self-labeling methods [26, 2, 28, 3, 24, 27, 40] estimate *hard* pseudo-labels, which could be equivalently represented either by class indices

$$y_i \in \{1, \dots, K\} \tag{10}$$

or by the corresponding one-hot categorical distributions

$$y_i \;\; \equiv \;\; (y_i^1, \dots, y_i^K) \, \in \Delta_{0,1}^K \qquad \text{for} \quad y_i^k := [k = y_i] \; \in \{0,1\} \tag{11}$$

analogously with the hard ground truth labels in (1) and (2). In part, hard pseudo-labels are motivated by the network training where the default is NLL loss (3) assuming discrete labels. Besides, there are powerful discrete solvers for the Potts model [6, 32, 8]. We discuss the potential advantages of soft pseudo-labels in the next Section 1.2.

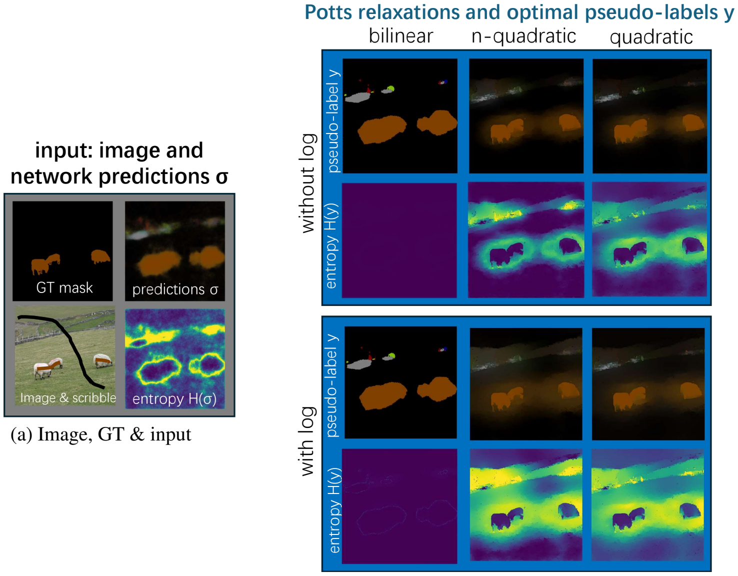
Joint loss vs "proposal generation": The majority of self-labeling approaches can be divided into two groups. One group designs pseudo-labeling and the network training sup-problems that are not formally related, e.g. [26, 25, 41]. While pseudo-labeling typically depends on the current network predictions and the network fine-tuning uses such pseudo-labels, the lack of a formal relation between these sub-problems implies that iterating such steps does not guarantee any form of convergence. Such methods are often referred to as *proposal generation* heuristics.

Alternatively, the pseudo-labeling sub-problem and the network training sub-problem can be formally derived from a weakly-supervised loss like (4), e.g. by ADM *splitting* [28] or as high-order *trust-region* method [27]. Such methods often formulate a *joint loss* function w.r.t network predictions and pseudo-labels and iteratively optimize it in a convergent manner that is guaranteed to decrease the loss. We consider this group of self-labeling methods as better motivated, more principled, and numerically safer.

B Optimization Algorithm

In this section, we will focus on the optimization of (6) in steps iterating optimization of y and σ. The network parameters are optimized by standard stochastic gradient descent in all our experiments. Pseudo-labels are also estimated online using a mini-batch. To solve y at given σ, it is a large-scale constrained convex problem. While there are existing general solvers to find global optima, such as projected gradient descent, it is often too slow for practical usage. Instead, we reformulate our problem to avoid the simplex constraints so that we can use standard gradient descent in PyTorch library accelerated by GPU. Specifically, instead of directly optimizing y, we optimize a set of new variables $\{l_i \in \mathbb{R}^K, i \in \Omega\}$ where y_i is computed by $softmax(l_i)$. Now, the simplex constraint on y will be automatically satisfied. Note that the hard constraints on scribble regions still need to be considered because the interaction with unlabeled regions through pairwise terms will influence the optimization process. Inspired by [44], we can reset $softmax(l_i)$ where $i \in S$ back to the ground truth at the beginning of each step of the gradient descent.

However, the original convex problem now becomes non-convex due to the Softmax operation. Thus, initialization is important to help find better local minima or even the global optima. Empirically, we observed that the network output logit can be a fairly good initialization. The quantitative comparison



(a) Image, GT & input

(b) Pseudo-labels using different Potts relaxation

Figure 7: Illustration of the difference among Potts relaxations. The visualization of soft pseudo-labels uses the convex combination of RGB colors for each class weighted by pseudo-label itself.

uses a special quadratic formulation where closed-form solution and efficient solver [1, 17] exist. We compute the standard soft Jaccard index for the pseudo-labels between the solutions given by our solver and the global optima. The soft Jaccard index is 99.2% on average over 100 images. Furthermore, our experimental results for all other formulations in Figure 7, 5, and Section 3 confirm the effectiveness of our optimization solver. In all experiments, the number of gradient descent steps for solving y is 200 and the corresponding learning rate is 0.075. To test the robustness of the number of steps here, we decreased 200 to 100 and the mIoU on the validation set just dropped from 71.05 by 0.72. This indicates that we can significantly accelerate the training without much sacrifice of accuracy. When using 200 steps, the total time for the training will be about 3 times longer than the SGD with dense Potts [36].

C Experimental settings

Dataset and evaluation We mainly use the standard PASCAL VOC 2012 dataset [16] and scribble-based annotations for supervision [26]. The dataset contains 21 classes including background. Following the common practice [10, 35, 36], we use the augmented version which has 10,582 training images and 1449 images for validation. We employ the standard mean Intersection-over-Union (mIoU) on validation set as the evaluation metric. We also test our method on two additional datasets in Section 3.5. One is Cityscapes [13] which is built for urban scenes and consists of 2975 and 500 fine-labeled images for training and validation. There are 19 out of 30 annotated classes for semantic segmentation. The other one is ADE20k [43] which has 150 fine-grained classes. There are 20210 and 2000, images for training and validation. Instead of scribble-based supervision, we followed [25] to use the block-wise annotation as a form of weak supervision.

Implementation details We adpoted DeepLabv3+ [12] framework with two backbones, ResNet101 [19] and MobileNetV2 [34]. We use ResNet101 in Section 3.5, and use MobileNetV2 in other sections for efficiency. All backbone networks (ResNet-101 and MobileNetV2) are pre-trained on Imagenet [15]. Unless stated explicitly, we use batch 12 as the default across all the experiments. Following [35], we adopt two-stage training, unless otherwise stated, where only the cross-entropy loss on scribbles is used in the first stage. The optimizer for network parameters is SGD. The learning rate is scheduled by a polynomial decay with a power of 0.9. Initial learning is set to 0.007 in the first stage and 0.0007 in the second phase. 60 epochs are used to train the model with different losses where hyperparameters are tuned separately for them. For our best result, we use $\eta = 0.3, \lambda = 6,$

H_{CCE} and P_{CD}. The color intensity bandwidth in the Potts model is set to 9 across all the experiments on Pascal VOC 2012 and 3 for Cityscapes and ADE20k datasets.

Method	Architecture	Cityscapes	ADE20k
Full supervision			
Deeplab [12]	V3+	80.2	44.6
Block-scribble supervision			
DenseCRF loss [36]	V3+	69.3	37.4
GridCRF loss* [27]	V3+	69.5	37.7
TEL [25]	V3+	71.5	39.2
$\mathbf{H}_{\text{CCE}} + \mathbf{P}_{\text{CD}}$	V3+	72.4	39.7

Table 6: Comparison to SOTA methods (without CRF postprocessing) on segmentation with block-scribble supervision. The numbers are mIoU on the validation dataset of cityscapes [13] and ADE20k [43] and use 50% of full annotations for supervision following [25]. The backbone is ResNet101. "$*$": reproduced results. All methods are trained in a single-stage fashion.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading “NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification:

   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: The training time is longer and more details can be found in the end of Appendix B.

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:

   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: All the details are given in the Appendix C.

   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: The code will be released upon acceptance.

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: See Appendix C.

   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: We reported the best following everyone else.

   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: See end of Appendix B.

   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:

   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:

    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:

    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: [NA]

    Justification:

    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:

    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:

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:

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.