Scribble-Supervised Semantic Segmentation by Uncertainty Reduction on Neural Representation and Self-Supervision on Neural Eigenspace

Zhiyi Pan[1] , **Peng Jiang**[1*] , **Yunhai Wang**[1] , **Changhe Tu**[1,2] , **Anthony G. Cohn**[3]
[1]Shandong University
[2]AICFVE, Beijing Film Academy
[3]University of Leeds
{panzhiyi1996, sdujump, cloudseawang, changhe.tu}@gmail.com, a.g.cohn@leeds.ac.uk

Abstract

Scribble-supervised semantic segmentation has gained much attention recently for its promising performance without high-quality annotations. Due to the lack of supervision, confident and consistent predictions are usually hard to obtain. Typically, people handle these problems to either adopt an auxiliary task with the well-labeled dataset or incorporate the graphical model with additional requirements on scribble annotations. Instead, this work aims to achieve semantic segmentation by scribble annotations directly without extra information and other limitations. Specifically, we propose holistic operations, including minimizing entropy and a network embedded random walk on neural representation to reduce uncertainty. Given the probabilistic transition matrix of a random walk, we further train the network with self-supervision on its neural eigenspace to impose consistency on predictions between related images. Comprehensive experiments and ablation studies verify the proposed approach, which demonstrates superiority over others; it is even comparable to some full-label supervised ones and works well when scribbles are randomly shrunk or dropped.

1 Introduction

In recent years, the use of neural networks, especially convolutional neural networks, has dramatically improved semantic classification, detection, and segmentation [LeCun *et al.*, 2015]. As one of the most fine-grained ways to understand the scene, typically, semantic segmentation demands large-scale data with high-quality annotations to feed the network. However, the pixel-level annotating process for semantic segmentation is costly and tedious, limiting its flexibility and usability on some tasks that require rapid deployment [Lin *et al.*, 2016]. As a consequence, the scribble annotations, which are more easily available, have become popular.

The main difficulty for scribble-supervised semantic segmentation lies in two aspects. (1) the scribble annotation is sparse and cannot provide enough supervision for the network to make confident predictions. (2) the scribble annotation varies from image to image, which is hard for the network to produce consistent results. As a consequence, Lin *et al.* adopted the classic graphical model as post-processing to obtain the final dense predictions. Some works [Vernaza and Chandraker, 2017; Wang *et al.*, 2019a] turn to auxiliary task (edge detection) with well-labeled dataset for help, so the heavy labor of annotation has not been relieved yet. To avoid the post-processing and dependence on another well-labeled dataset, Tang *et al.* design a graphical model regularized loss to make predictions consistency within the appearance similar neighborhood, but did not consider semantic similarity. Moreover, they require every object in an image to be labeled, which is too strict for dataset preparation.

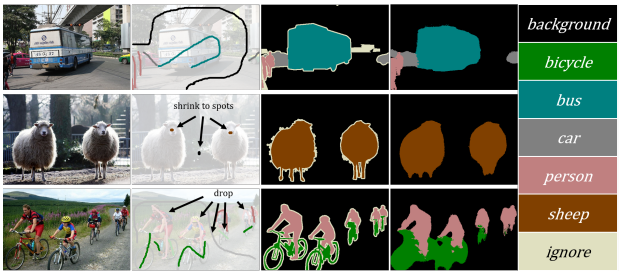


Figure 1: From left to right: image, scribble annotation, ground truth, our prediction. From top to bottom: sample with regular, shrunk and dropped scribble annotation, respectively.

We address the task by a more flexible approach without introducing auxiliary supervision and constraints in this work. The approach can work properly when scribbles on some objects are randomly dropped or even shrunk to spots. Several representative results are listed in Fig. 1. We propose two creative solutions for the problems of confidence and consistency mentioned above. To reduce the uncertainty when supervision is lacking, we take advantage of two facts related to semantic segmentation. The first one is that each pixel only belongs to one category (deterministic), and there is only one channel of output neural representation that plays the dominant role. The second one is neural representations should be uniform within internal object regions. Accordingly, we develop the first solution on neural representation, including two specific operations, minimizing entropy to encourage deterministic predictions and a network embedded random walk

*Corresponding Author

module to promote uniform intermediates. Besides, the transition matrix of a random walk will also be useful for consistency enhancement later. In general, we make up for the lack of supervision with scribble annotations by taking advantage of two priors, deterministic and uniform.

We propose to adopt self-supervision during training as the second solution for inconsistent results caused by varying scribble annotations from image to image, which imposes consistency on the neural representation before and after certain input transformation [Laine and Aila, 2016]. However, consistency over the whole neural representation usually is not necessary for semantic segmentation, especially for regions belonging to the background category, which usually are semantically heterogeneous. When these regions are distorted and changed heavily after transformation, it is hard for the network to generate consistent output and may confuse the network in some scenarios. With that in mind and given the transition matrix of a random walk, we propose to set self-supervision on the main parts of images by imposing consistent loss on the eigenspace of transition matrix. The idea is inspired by spectral methods [Von Luxburg, 2007], who observed that eigenvectors of a Laplacian matrix have the capability to distinguish the main parts in images, and some methods use this property for clustering [Ng *et al.*, 2002] and saliency detection [Jiang *et al.*, 2019]. Since the eigenspace of a transition matrix has a close relation to the one of a Laplacian matrix, our self-supervision on transition matrix's eigenspace will also focus on the main image parts.

The proposed approach demonstrates consistent superiority over others on the common scribble dataset and is even comparable to some fully supervised ones. Moreover, we further conduct experiments when scribbles are gradually shrunk and dropped. The proposed approach can still work reasonably, even the scribble shrunk to a spot or dropped significantly. Careful ablation studies are made to verify the effectiveness of every operation. Finally, in supplementary material, the code and dataset are open-sourced.

2 Related Work

Scribble-supervised semantic segmentation aims to produce dense predictions given only sparse scribbles. Existing deep learning-based works can usually be divided into two groups: 1) Two-stage approaches [Lin *et al.*, 2016; Vernaza and Chandraker, 2017], which first obtain full mask pseudo-labels by manipulating scribble annotations, and then train the network as usual using semantic segmentation with pseudo-labels. 2) Single-stage approaches [Tang *et al.*, 2018a; Tang *et al.*, 2018b], which directly train the network using scribble annotations by a specific loss function and network structure. While two-stage approaches can be formulated as regular semantic segmentation, single-stage approaches are usually defined to minimize the following function:

$$L = \sum_{p \in \Omega_{\mathcal{L}}} c(s(x)_p, y_p) + \lambda \sum_{p,q \in \Omega} u(s(x)_p, s(x)_q), \qquad (1)$$

where Ω is point (pixel) set, $\Omega_{\mathcal{L}}$ is point set with scribble annotations, $s(x)_i$ represents prediction at point i given input x, and y_i is the corresponding ground truth. The first term measures the error with scribble annotations and usually is in the form of cross-entropy. The second term is a pair-wise regularization to help generate uniform predictions. The two terms are harmonized by a weight parameter λ.

For scribble-supervised semantic segmentation, the graphical model has been prevalently adopted in either two-stage approaches for generating pseudo-label or one-stage approaches for loss design. Lin *et al.* iteratively conduct label refinement and network optimization through a graphical model. Vernaza and Chandraker generate high-quality pseudo-labels for full-label supervised semantic segmentation by optimizing graphical model with edge detector learned from another well-labeled dataset. These two works require iterative optimization or an auxiliary dataset. Instead, Tang *et al.* add soft graphical model regularization into the loss function and explicitly avoid graphical model optimization. Besides, some only work well on a dataset where every object is labeled by at least one scribble. In general, most existing works have not provided a flexible and efficient solution to scribble-supervised semantic segmentation yet.

3 Method

Scribble-supervised semantic segmentation usually suffers from uncertain and inconsistent predictions due to lack of supervision and varying annotations from image to image. In this work, we propose two solutions, *viz.* uncertainty reduction on neural representation and self-supervision on neural eigenspace to address these problems. Compared with others, we do not rely on auxiliary tasks with well-labeled datasets and additional requirements for annotation preparation.

3.1 Uncertainty Reduction on Neural Representation

To reduce the uncertainty on neural representation, we take advantage of priors that neural presentations should be deterministic and uniform for each semantic object. Thereby, holistic operations are developed and imposed on neural representation, including minimizing entropy and network embedded random walk.

Minimizing entropy

The entropy on neural representation is defined as:

$$E_\Omega = -\frac{1}{HW} \sum_{(i,j) \in \Omega} \sum_{c} s(x)_{i,j,c} \cdot log(s(x)_{i,j,c}), \qquad (2)$$

where $s(x)$ represents the prediction given the input x, and is of the size [*H,W,C*] (C is the number of categories). $s(x)_{i,j,c}$ represents the probability that the pixel at position (i, j) belongs to the c-th category.

Entropy indicates the randomness of a system. According to a classical thermodynamic principle: *Minimizing entropy results in minimum randomness of a system.* Thus, minimizing entropy on neural representation will reduce the uncertainty and force the network to produce deterministic predictions. However, uncertain predictions are inevitable in places such as object boundaries, and undifferentiated entropy minimization will cause network training conflict. Correspondingly, we propose to minimize entropy on neural representation excluding positions corresponding to object boundaries,

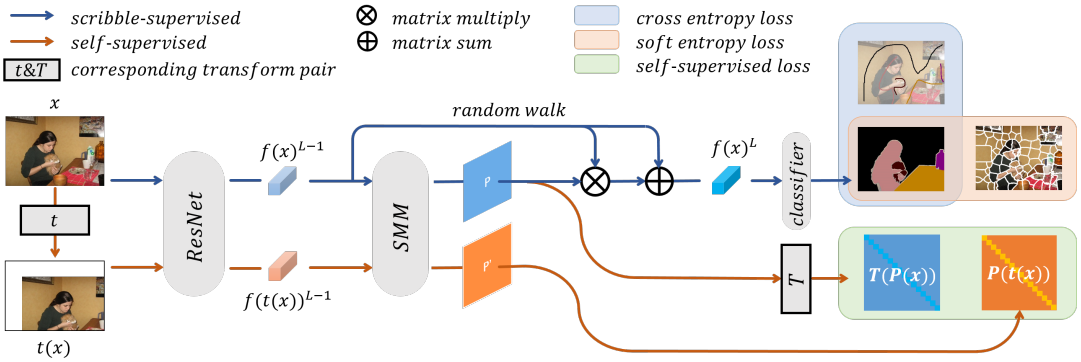


Figure 2: Network Pipeline. We use blue and orange flows to represent scribble-supervised training and self-supervised training, respectively. Given an image and its transform, we pass them to ResNet to extract neural representations $f(x)^{L-1}$, from which similarity measurement module (SMM) computes transition matrix. A random walk is then carried out on $f(x)^{L-1}$. The results $f(x)^{L}$ are used for classification. Simultaneously, soft entropy by pseudo-boundaries is minimized to reduce uncertainty of neural representation, and self-supervised loss is set between transition matrices to realize self-supervision on neural eigenspace. During inference, only the blue flow is activated.

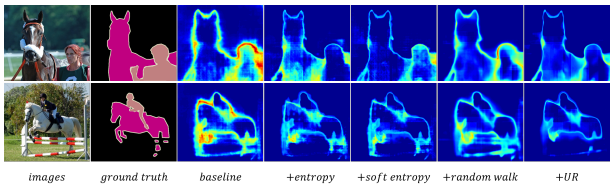


Figure 3: Entropy map. Colder color indicates smaller entropy.

leading to soft entropy:

$$E_{\Omega_{-\mathcal{B}}} = -\frac{1}{HW} \sum_{(i,j)\in\Omega_{-\mathcal{B}}} \sum_{c} s(x)_{i,j,c} \cdot log(s(x)_{i,j,c}) \quad (3)$$

where $\Omega_{-\mathcal{B}}$ is the point set that excludes object boundaries. In this way, minimizing soft entropy will reduce uncertainty and avoid potential conflicts on object boundaries. Since accurate boundaries are hard to acquire, we only use pseudo boundaries by the no-learning-based superpixel method, SLIC [Achanta *et al.*, 2012]. We note that entropy has been explored for some vision tasks, such as object detection [Wan *et al.*, 2018], but with different motivation and implementation. To our best knowledge, minimizing entropy is adopted to scribble-supervised semantic segmentation for uncertainty reduction for the first time.

Network embedded random walk

A random walk operation is defined as:

$$z = \alpha P y + (1-\alpha) y, \quad (4)$$

y is the initial state vector, α is a parameter that controls the degree of random walk, P is transition matrix that measures the transition possibilities between every two positions, and we usually set similar positions with a large possibility of transition [Von Luxburg, 2007]. By definition above, the output state z after random walk will have more similar states for similar positions, resulting in the uniform state within similar semantic/appearance regions.

Inspired by the characteristic of a random walk, we propose to embed this operation into the network for the uniform neural representation,

$$f(x)^{L} = \alpha P f(x)^{L-1} + f(x)^{L-1}, \quad (5)$$

where $f(x)^{L-1}$ is the neural representation of input x in layer L-1 and $f(x)^{L}$ is the neural representation after random walk in layer L, they are both of dimension [*M,N,K*]. We set α as a learnable parameter to be trained during training and define the probabilistic transition matrix P as:

$$P = softmax(f(x)^{L-1^T} f(x)^{L-1}), \quad (6)$$

where $f(x)^{L-1}$ is flattened to [*MN,K*], and $f(x)^{L-1^T} f(x)^{L-1}$ will produce a matrix of dimension [*MN,MN*]. By $softmax$ in the horizontal direction, we generate a suitable probabilistic transition matrix P with all units are positive and every row of the matrix is summed to 1.

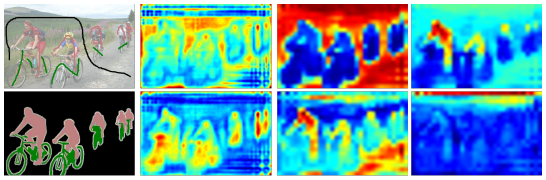
A random walk has been frequently used for semantic segmentation tasks [Bertasius *et al.*, 2017; Jiang *et al.*, 2018; Ahn and Kwak, 2018; Araslanov and Roth, 2020]. However, most of them use a random walk to diffuse the pseudo-label or refine the initial predictions. Instead, we use a random walk on neural representation for uniform and uncertainty reduction when given only scribble annotations.

Uncertainty reduction verification

In this part, we verify how the uncertainty is reduced by the two operations mentioned above. Given several randomly selected samples, we measure pixel-level entropy maps for predictions obtained by *baseline*, networks with proposed operations individually (*+entropy*,*+soft entropy*,*+random walk*) and together (*+uncertainty reduction (UR)*). The results are visualized in Fig. 3. As expected, the entropy is decreased by the proposed two operations, and using them both leads to minimum entropy, albeit object boundaries remaining uncertain. The detailed setting of networks will be stated later.

3.2 Self-Supervision on Neural Eigenspace

Self-supervision computes the misfit between the network's intermediates of the input and its transform, which forces the network to produce consistent outputs. Self-supervision has been utilized for unsupervised learning tasks to provide unsupervised loss [Laine and Aila, 2016; Mittal *et al.*, 2019].



(a) (b) (c)

Figure 4: (a) From top to bottom: scribble-annotation and ground-truth. (b) From top to bottom: Neural representation before and after a random walk. (c) Leading eigenvectors of the transition matrix.

We adopt self-supervision to address the issue of inconsistent results caused by varying scribble annotations from image to image. Several issues need to be considered when applying self-supervision loss in this work: (1) where the self-supervision is involved; (2) how the self-supervision loss is calculated; (3) what kinds of the transform will be used. We address these issues in the following.

Self-supervision

The most straight forward way to implement self-supervision is to compute the difference between neural representations of the input and its transform:

$$ss(x, \phi) = l(T_\phi(f(x)), f(t_\phi(x))), \tag{7}$$

where t_ϕ denotes the transform operation on x with ϕ parameter, while T_ϕ corresponds to the transform operation on $f(x)$ (t_ϕ and T_ϕ are a pair of corresponding transforms for self-supervision). l is the metric to measure difference. We denote this kind of consistency as $ss(x, \phi)$. Given operations in Sec. 3.1, There are several obvious places to apply self-supervision, *e.g.* neural representations $f(x)^{L-1}$ and $f(x)^L$.

Self-supervision on neural eigenspace

However, as for the semantic segmentation task with self-supervision, we argue that directly calculating loss on the whole neural representation is not necessary and may not be optimal. When the image is distorted heavily after the transform, some parts of its neural representation will change greatly, so minimizing Eq. 7 will be hard and even ambiguous. In this work, given the transition matrix P of random walk in Sec. 3.1, we propose to apply self-supervision on the neural eigenspace of P.

The eigenspace of transition matrix P and that of the normalized Laplacian matrix L have close relationships [Von Luxburg, 2007]. It can be proved that Λ_P=1 $-\ \Lambda_L$ and U_P=U_L (Λ denotes a diagonal matrix with eigenvalues as entries, U denotes a matrix with eigenvectors as columns). According to [Jiang *et al.*, 2015; Jiang *et al.*, 2019], columns of U_L have the capability to distinguish the main parts of the images. So, U_P will also inherit this property. We visualize several eigenvectors of P in Fig. 4. As can be seen, compared with the original neural representations $f(x)^{L-1}$ and $f(x)^L$, the eigenvectors of P are better able to distinguish the main parts from others and neglect some details, though P is also computed from neural representation. Based on the above analysis, we define the self-supervision as,

$$\begin{aligned} ss(x, \phi) = & \ l(T_\phi(U_P(x)), U_P(t_\phi(x))) \\ & + l(T_\phi(\Lambda_P(x)), \Lambda_P(t_\phi(x))). \end{aligned} \tag{8}$$

Soft eigenspace self-supervision

Eq. 8 requires explicit eigendecomposition, which is time-consuming, especially within the deep neural network context. Though there are some approximation methods [Dang *et al.*, 2018; Wang *et al.*, 2019b; Sun and Xu, 2019] proposed, their efficiency and stability are still far from satisfactory. To this end, we develop soft eigenspace self-supervision, which avoids explicit eigendecomposition. Firstly, in view of the fact that the matrix's trace is equal to the sum of its eigenvalues, we measure the consistency on Λ by computing the difference to the trace of P, $tr(P)$. Secondly, given the consistency on the Λ, we propose to measure the consistency on the P to obtain consistent U indirectly. In other words, the soft eigenspace self-supervision loss is defined as:

$$\begin{aligned} ss_P(x, \phi) = & \ l_1(T_\phi(P(x)), P(t_\phi(x))) \\ & + \gamma * l_2(T_\phi(tr(P(x))), tr(P(t_\phi(x)))), \end{aligned} \tag{9}$$

where $P(x)$ denotes P for input x, $tr(P(x))$ is the trace of $P(x)$. Since $P(x)$ is a probabilistic transition matrix, we define l_1 as Kullback-Leibler Divergence, and use the L_2 norm for l_2. γ is the weight to control the two terms.

Computing efficiency

We set two linear transform operations for self-supervision, horizontal flip, and translation, $\phi \in$(horizontal flip, translation). Compared with the transform that impacts neural representation, any transform will lead to a complex change on P and complicate the computation. However, since all transform operations are linear, the probabilistic transition matrix after transform can be expressed as the multiplication of the original P with predefined computing matrices to facilitate Eq. 9 computation. $T_\phi(P(x))$ can be defined as:

$$T_\phi(P(x)) = T_\phi r \cdot P(x) \cdot T_\phi c, \tag{10}$$

where $T_\phi r$ and $T_\phi c$ are predefined computing matrices for transform ϕ. Details are in the supplementary material.

Uncertainty Reduction			mIoU
entropy	boundary	random walk	
			66.8
		✓	69.3
✓			69.4
✓	✓		70.0
✓	✓	✓	**70.9**

Table 1: Ablation study for uncertainty reduction.

4 Implementation

The network is illustrated in Fig. 2, including two modules (ResNet to extract features, and Similarity Measurement Module (*SMM*) to compute probabilistic transition matrix), one specific process (random walk), and three loss functions (soft entropy, self-supervision, and cross-entropy). These components realize uncertainty reduction on neural representation and self-supervision on neural eigenspace.

A random walk is embedded in the network's computation flow and conducted on the final layer right before the classifier, strictly following Eq. 5 with learned α that controls

Self-Supervision		mIoU
transform operation	location	
flip	$f(x)^{L-1}$	72.5
	$f(x)^{L}$	72.5
	$Eigenspace$	**72.9**
translation	$f(x)^{L-1}$	71.9
	$f(x)^{L}$	70.0
	$Eigenspace$	**72.6**
random	$f(x)^{L-1}$	72.4
	$f(x)^{L}$	71.6
	$Eigenspace$	**73.0**

Table 2: Ablation study for self-supervision.

the degree of random walk. Similarity Measurement Module computes the inner product distance between any pairs of neural representation elements and forms the probabilistic transition matrix P as Eq. 6.

We use the pre-trained ResNet [He *et al.*, 2016] with dilation [Chen *et al.*, 2017] as the backbone to extract initial neural representations. The total loss in our work is defined as:

$$L = \sum_{p \in \Omega_{\mathcal{L}}} c(s(x)_p, y_p) + \omega_1 E_{\Omega_{-\mathcal{B}}} + \omega_2 * ss_P(x, \phi), \quad (11)$$

where ω_1 and ω_2 are predefined weights. The first term measures the divergence of prediction to ground truth at positions with scribbles. The second term computes entropy within regions excluding pseudo-boundaries. The third term is self-supervision on eigenspace. By minimizing Eq. 11, we are training the network to approximate scribbles when they are available, produce confident predictions and consistent outcome eigenspace. The random walk embedded in the network will also help generate uniform intermediates. Consequently, we overcome difficulties of scribble-supervised semantic segmentation when given sparse and random annotations.

The training process has two steps. In the beginning, only the first two terms participate. At this time, the network may not perform well initially, and self-supervision will not bring benefits but prevent the optimization. After the network gets reasonable performance, the whole Eq. 11 is activated.

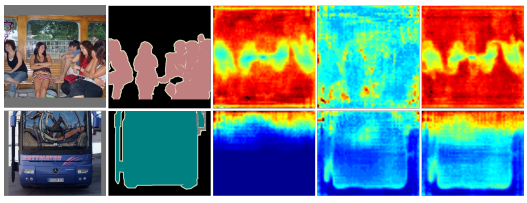


Figure 5: Variation by different self-supervision. (a) input image, (b) ground truth, (c) (d) (e) variation on $f(x)^{L-1}$, $P(x)$ and $f(x)^{L}$, respectively. The first row shows variations for the flip operation, while the second row is for the translation operation.

	$f(x)^{L-1}$	$f(x)^{L}$	$P(x)$
flip	52.8%	37.5%	6.7%
translation	7.5%	12.0%	3.5%

Table 3: Variation comparison under the same transform operation.

5 Experiment

5.1 Experiment Setting

Datasets

We make comparison on the common scribble-annotated dataset, *scribblesup* [Lin *et al.*, 2016]. This dataset has 21 classes (including an ignore category) with every object in the image labeled by at least one scribble. However, our approach can work without preconditions. To verify our advantages, we further prepare two variants of *scribblesup* with the same training and validation partition. The first one is *scribble-drop*, where every object in image randomly drops (*i.e.* deletes) all scribbles. The second one is *scribble-shrink*, where every scribble in the image is shrunk randomly (even to a spot). We test many settings of the drop and shrink rate.

Compared methods

We compare with recently proposed scribble-supervised methods including scribblesup [Lin *et al.*, 2016], RAWKS [Vernaza and Chandraker, 2017], NCL [Tang *et al.*, 2018a], KCL [Tang *et al.*, 2018b], BPG-PRN [Wang *et al.*, 2019a], and also point supervised methods (What'sPoint [Bearman *et al.*, 2016]). Besides, the full-label supervised method (DeepLabV2 [Chen *et al.*, 2017]) is also compared. We use mIoU as the main metric for evaluation. When comparing with others, we refer to their reported scores if available.

Hyper-parameters

All training images are randomly scaled (0.5 to 2), rotated (-10 to 10), blurred, and flipped for data augmentation, then cropped to [465,465] before feeding to the network. $f(x)^{L-1}$ and $f(x)^{L}$ are of spatial dimension [59,59]. All the computations are carried out on NVIDIA TITAN RTX GPUs. The supplementary material details the setting of γ, ω_1 and ω_2.

Method	Ann.	Backbone	wo/ CRF	w/ CRF
What'sPoint	$\mathcal{P}$	$VGG16$	46.0	-
DeepLabV2	$\mathcal{F}$	$ResNet101$	76.4	77.7
scribblesup	$\mathcal{S}$	$VGG16$	-	63.1
RAWKS	$\mathcal{S}$	$ResNet101$	59.5	61.4
NCL	$\mathcal{S}$	$ResNet101$	72.8	74.5
KCL	$\mathcal{S}$	$ResNet101$	73.0	75.0
BPG-PRN	$\mathcal{S}$	$ResNet101$	71.4	-
ours-ResNet50	$\mathcal{S}$	$ResNet50$	73.0	74.7
ours-ResNet101	$\mathcal{S}$	$ResNet101$	**74.4**	**76.1**

Table 4: Performance on validation set of $Scribblesup$. The annotation type (Ann.) indicates: $\mathcal{P}$–point, $\mathcal{S}$–scribble and $\mathcal{F}$–full label.

5.2 Ablation Study

In this part, we investigate all operations involved in Sec. 3. We use *Scribblesup* dataset for training and validation.

drop rate	0.1	0.2	0.3	0.4	0.5	shrink rate	0.2	0.5	0.7	1
baseline	66.3	65.4	65.1	63.9	63.8	baseline	66.2	64.8	63.8	57.0
+Uncertainty Reduction	69.6	69.1	68.0	67.3	67.2	+Uncertainty Reduction	69.7	67.3	65.3	59.0
+Self-Supervision	72.2	71.4	71.4	69.8	69.5	+Self-Supervision	72.1	70.4	68.8	62.8

Table 5: mIoU scores on *scribble-drop* and *scribble-shrink* with different drop and shrink rates.

Firstly, we do an ablation study for operations in Sec. 3.1. Starting with baseline (*ResNet50*), we gradually add entropy minimization, boundary exclusion, and random walk, obtaining networks with different combinations. We report mIoU in Tab. 1. The first row is the baseline. We can observe that all operations can obtain better performance, and using them all leads to the best performance. We get 4.1% improvement by uncertainty reduction on neural representation in total.

Secondly, we do an ablation study for self-supervision in Sec. 3.2. Starting with the network getting best score (70.9%) in Tab. 1, we compare networks after further training by self-supervision on $f(x)^{L-1}$, $f(x)^{L}$ and eigenspace of $P(x)$ with different transform operations. We report mIoU in Tab. 2. We observe that not all of them would bring improvement, self-supervision on $f(x)^{L}$ with translation operation even deteriorates the performance. However, self-supervision on the eigenspace of $P(x)$ improves the performance consistently, and randomly selecting transform operations achieves the best performance. We get 2.1% (from 70.9% to 73.0%) improvement by self-supervision on the neural eigenspace.

The eigenspace of P is the feature located behind $f(x)^{L-1}$ and in front of $f(x)^{L}$. To delve into the reason why self-supervision on eigenspace outperforms others, in Tab. 3, we show mean variations of $f(x)^{L-1}$, $f(x)^{L}$ and $P(x)$ under the same transform (no self-supervision applied yet). The variation is measured by the relative error. As can be seen, the same transform will always lead to less variation on $P(x)$. In Fig. 5, we visualize variation for self-supervision on different places. Compared with $f(x)^{L-1}$ and $f(x)^{L}$, variation on $P(x)$ is smaller in background regions. This phenomenon indicates that self-supervision on $P(x)$ can focus on the main parts and avoid training conflicts on the background category with unstable semantics, relieving the training burden.

5.3 Quantitative Results

We get mIoU 73.0% with ResNet50 and 74.4% with ResNet101 on the *Scribblesup* dataset. When comparing with others, we also report performance with CRF as others. Tab. 4 lists all scores for compared methods. In addition to scribble-supervised methods, we also show methods with other label types, such as point and full-label. The proposed method reaches state-of-the-art performance compared with other scribble-supervised methods and is even comparable to the full-label one. The reported full-label method (DeepLabV2) was additionally pre-trained on COCO dataset. The supplementary material has more results by different label types.

It should be noted that some methods, such as [Lin *et al.*, 2016; Vernaza and Chandraker, 2017; Tang *et al.*, 2018b], require every object is labeled. However, ours does not have this limit. In Tab. 5, we show the performance under different drop and shrink rates on *scribble-drop* and *scribble-shrink* datasets. As can be seen, with our proposed solutions, we perform well when the drop rate and shrink rate increase, even when all scribbles were shrunk to spots.

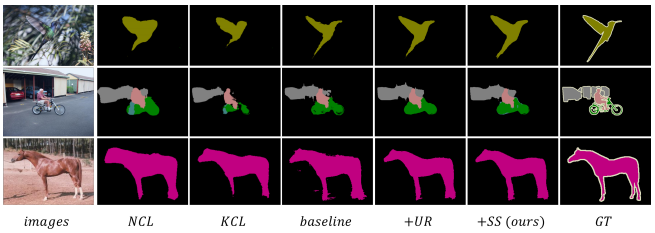


Figure 6: Visual comparison between ours and others.

5.4 Qualitative Results

Fig. 6 shows the visual comparison between NCL, KCL, and ours on three images from the test set of *scribblesup*. With the proposed uncertainty reduction (UR) and self-supervision (SS), results are gradually refined and show significant promotion over the baseline and others. (The results and scores in this section are all from the validation set. The supplementary material has more results on *scribble-drop* and *scribble-shrink* datasets.)

	P (M)	M (MB)	S (it/s)
Baseline	41.72	2015.5	4.21
+Uncertainty Reduction	+1.24	+37.2	+0.32
+Self-Supervision	+0	+10.8	+0

Table 6: Parameters (P), memory (M), inference speed (S).

5.5 Computational Resources Consumed

The consumed resources are measured in Tab. 6. Self-supervision does not need to preserve intermediates nor has extra parameters and is only adopted during training. Random walk only requires moderate space to save the transition matrix. Consequently, the cost of the proposed solutions is acceptable.

6 Conclusion

In this work, we recognize that semantic segmentation given only scribble annotations will cause uncertain and inconsistent predictions. Accordingly, we develop two creative solutions, uncertainty reduction on neural representation to produce confident results, and self-supervision on neural eigenspace for consistency in output. No additional information and requirement for annotation preparation is needed.

Thorough ablation studies and intermediate visualization have verified the effectiveness of the proposed solutions. Finally, we reach state-of-the-art performance compared with others, even comparable to the full-label supervised ones. Moreover, the proposed approach still works when scribbles are randomly dropped or shrunk.

References

[Achanta *et al.*, 2012] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

[Ahn and Kwak, 2018] 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.

[Araslanov and Roth, 2020] Nikita Araslanov and Stefan Roth. Single-stage semantic segmentation from image labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[Bearman *et al.*, 2016] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In *European conference on computer vision*. Springer, 2016.

[Bertasius *et al.*, 2017] Gedas Bertasius, Lorenzo Torresani, Stella X Yu, and Jianbo Shi. Convolutional random walk networks for semantic image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 858–866, 2017.

[Chen *et al.*, 2017] 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.

[Dang *et al.*, 2018] Zheng Dang, Kwang Moo Yi, Yinlin Hu, Fei Wang, Pascal Fua, and Mathieu Salzmann. Eigendecomposition-free training of deep networks with zero eigenvalue-based losses. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 768–783, 2018.

[He *et al.*, 2016] 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.

[Jiang *et al.*, 2015] Peng Jiang, Nuno Vasconcelos, and Jingliang Peng. Generic promotion of diffusion-based salient object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[Jiang *et al.*, 2018] Peng Jiang, Fanglin Gu, Yunhai Wang, Changhe Tu, and Baoquan Chen. Difnet: Semantic segmentation by diffusion networks. In *Advances in Neural Information Processing Systems*, pages 1630–1639, 2018.

[Jiang *et al.*, 2019] Peng Jiang, Zhiyi Pan, Changhe Tu, Nuno Vasconcelos, Baoquan Chen, and Jingliang Peng. Super diffusion for salient object detection. *IEEE Transactions on Image Processing*, 29:2903–2917, 2019.

[Laine and Aila, 2016] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

[LeCun *et al.*, 2015] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553), 2015.

[Lin *et al.*, 2016] 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.

[Mittal *et al.*, 2019] Sudhanshu Mittal, Maxim Tatarchenko, and Thomas Brox. Semi-supervised semantic segmentation with high-and low-level consistency. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[Ng *et al.*, 2002] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002.

[Sun and Xu, 2019] Jian Sun and Zongben Xu. Neural diffusion distance for image segmentation. In *Advances in Neural Information Processing Systems*, pages 1443–1453, 2019.

[Tang *et al.*, 2018a] 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.

[Tang *et al.*, 2018b] 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.

[Vernaza and Chandraker, 2017] Paul Vernaza and Manmohan Chandraker. Learning random-walk label propagation for weakly-supervised semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7158–7166, 2017.

[Von Luxburg, 2007] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[Wan *et al.*, 2018] Fang Wan, Pengxu Wei, Jianbin Jiao, Zhenjun Han, and Qixiang Ye. Min-entropy latent model for weakly supervised object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1297–1306, 2018.

[Wang *et al.*, 2019a] 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*, pages 3663–3669, 2019.

[Wang *et al.*, 2019b] Wei Wang, Zheng Dang, Yinlin Hu, Pascal Fua, and Mathieu Salzmann. Backpropagation-friendly eigendecomposition. In *Advances in Neural Information Processing Systems*, pages 3162–3170, 2019.