Semi-Supervised Semantic Segmentation Using Unreliable Pseudo-Labels

Yuchao Wang† Haochen Wang† Yujun Shen2 Jingjing Fei3 Wei Li3 Guoqiang Jin3 Liwei Wu3 Rui Zhao1,3‡ Xinyi Le†∗

1Shanghai Jiao Tong University  2The Chinese University of Hong Kong  3SenseTime Research

{44442222, wanghaochen0409, lexinyi}@sjtu.edu.cn  shenyujun0302@gmail.com
{feijingjing1, liwei1, jinguoqiang, wuliwei, zhaorui}@sensetime.com

Abstract

The crux of semi-supervised semantic segmentation is to assign adequate pseudo-labels to the pixels of unlabeled images. A common practice is to select the highly confident predictions as the pseudo ground-truth, but it leads to a problem that most pixels may be left unused due to their unreliability. We argue that every pixel matters to the model training, even its prediction is ambiguous. Intuitively, an unreliable prediction may get confused among the top classes (i.e., those with the highest probabilities), however, it should be confident about the pixel not belonging to the remaining classes. Hence, such a pixel can be convincingly treated as a negative sample to those most unlikely categories. Based on this insight, we develop an effective pipeline to make sufficient use of unlabeled data. Concretely, we separate reliable and unreliable pixels via the entropy of predictions, push each unreliable pixel to a category-wise queue that consists of negative samples, and manage to train the model with all candidate pixels. Considering the training evolution, where the prediction becomes more and more accurate, we adaptively adjust the threshold for the reliable-unreliable partition. Experimental results on various benchmarks and training settings demonstrate the superiority of our approach over the state-of-the-art alternatives.†

1. Introduction

Semantic segmentation is a fundamental task in the computer vision field, and has been significantly advanced along with the rise of deep neural networks [5,29,35,46]. Existing supervised approaches rely on large-scale annotated data, which can be too costly to acquire in practice. To alleviate this problem, many attempts [1,4,9,15,21,33,43,48] have been made towards semi-supervised semantic segmentation, which learns a model with only a few labeled samples and numerous unlabeled ones. Under such a setting, how to adequately leverage the unlabeled data becomes critical.

A typical solution is to assign pseudo-labels to the pixels without annotations. Concretely, given an unlabeled image, prior arts [27,41] borrow predictions from the model trained on labeled data, and use the pixel-wise prediction as the “ground-truth” to in turn boost the supervised model. To mitigate the problem of confirmation bias [2], where the model may suffer from incorrect pseudo-labels, existing approaches propose to filter the predictions with their confidence scores [42,43,50,51]. In other words, only the highly confident predictions are used as the pseudo-labels, while the ambiguous ones are discarded.

However, one potential problem caused by only using reliable predictions is that some pixels may never be learned
in the entire training process. For example, if the model cannot satisfyingly predict some certain class (e.g., chair in Fig. 1), it becomes difficult to assign accurate pseudo-labels to the pixels regarding such a class, which may lead to insufficient and categorically imbalanced training. From this perspective, we argue that, to make full use of the unlabeled data, every pixel should be properly utilized.

As discussed above, directly using the unreliable predictions as the pseudo-labels will cause the performance degradation [2]. In this paper, we propose an alternative way of Using Unreliable Pseudo-Labels. We call our framework as U²PL. First, we observe that, an unreliable prediction usually gets confused among only a few classes instead of all classes. Taking Fig. 2 as an instance, the pixel with white cross receives similar probabilities on class motorbike and person, but the model is pretty sure about this pixel not belonging to class car and train. Based on this observation, we reconsider the confusing pixels as the negative samples to those unlikely categories. Specifically, after getting the prediction from an unlabeled image, we employ the per-pixel entropy as the metric (see Fig. 2a) to separate all pixels into two groups, i.e., a reliable one and an unreliable one. All reliable predictions are used to derive positive pseudo-labels, while the pixels with unreliable predictions are pushed into a memory bank, which is full of negative samples. To avoid all negative pseudo-labels only coming from a subset of categories, we employ a queue for each category. Such a design ensures that the number of negative samples for each class is balanced. Meanwhile, considering that the quality of pseudo-labels becomes higher along with the model gets more and more accurate, we come up with a strategy to adaptively adjust the threshold for the partition of reliable and unreliable pixels.

We evaluate the proposed U²PL on PASCAL VOC 2012 [14] and Cityscapes [10] under a wide range of training settings, where our approach surpasses the state-of-the-art competitors. Furthermore, through visualizing the segmentation results, we find that our method achieves much better performance on those ambiguous regions (e.g., the border between different objects), thanks to our adequate use of the unreliable pseudo-labels.

2. Related Work

Semi-Supervised Learning has two typical paradigms: consistency regularization [3, 15, 33, 36, 42] and entropy minimization [4, 16]. Recently, a more intuitive but effective framework: self-training [27], has become the mainstream. Several methods [15, 43, 44] utilize strong data augmentation such as CutOut [13], CutMix [45], and ClassMix [31] based on self-training. However, these methods do not pay much attention to the characteristics of semantic segmentation, while our method focuses on those unreliable pixels which will be filtered out by most of self-training based methods [34, 43, 44].

Pseudo-Labeling is applied to prevent overfitting in incorrect pseudo-labels when generating predictions of input images from the teacher network [2, 27]. FixMatch [37] utilizes a confidence threshold to select reliable pseudo-labels. UPS [34], a method based on FixMatch [37], takes model uncertainty and data uncertainty into consideration. However, in semi-supervised semantic segmentation, our experiments show including unreliable pixels into training can boost performance.

Model Uncertainty in computer vision is mostly measured by Bayesian deep learning approaches [12, 23, 30]. In our settings, we do not focus on how to measure uncertainty. We simply use the entropy of pixel-wise probability distribution to be the metric.

Contrastive Learning is applied by many successful works in self-supervised learning [7, 8, 17]. In semantic segmentation, contrastive learning has become a promising new paradigm [1, 28, 40, 47, 49]. However, these methods ignore the common false negative samples in semi-supervised
Negative Learning aims at decreasing the risk of incorrect information by lowering the probability of negative samples [24, 25, 34, 39], but those negative samples are selected with high confidence. In other words, these methods still utilize pixels with reliable predictions. By contrast, we propose to make sufficient use of those unreliable predictions for learning instead of filtering them out.

3. Method

In this section, we establish our problem mathematically and give an overview of our proposed method in Sec. 3.1 first. Our strategies about filtering reliable pseudo-labels are introduced in Sec. 3.2. Finally, we describe how to use unreliable pseudo-labels in Sec. 3.3.

3.1. Overview

Given a labeled set \( D_l = \{ (x_i^l, y_i^l) \}_{i=1}^{N_l} \) and a much larger unlabeled set \( D_u = \{ x_i^u \}_{i=1}^{N_u} \), our goal is to train a semantic segmentation model by leveraging both a large amount of unlabeled data and a smaller set of labeled data.

Fig. 3 gives an overview of U^2 PL, which follows the typical self-training framework with two models of the same architecture, named teacher and student respectively. The two models differ only when updating their weights. The student model’s weights \( \theta_s \) are updated consistent with the common practice while the teacher model’s weights \( \theta_t \) are exponential moving average (EMA) updated by the student model’s weights. Each model consists of a CNN-based encoder \( h \), a decoder with a segmentation head \( f \), and a representation head \( g \). At each training step, we equally sample \( B \) labeled images \( B_l \) and \( B \) unlabeled images \( B_u \). For every labeled image, our goal is to minimize the standard cross-entropy loss in Eq. (2). As for each unlabeled image, we first take it into the teacher model and get predictions. Then, based on pixel-level entropy, we ignore unreliable pixel-level pseudo-labels when computing unsupervised loss in Eq. (3). This part will be introduced in section Sec. 3.2 in detail. Finally, we use the contrastive loss to make full use of the unreliable pixels excluded in the unsupervised loss, which will be introduced in Sec. 3.3.

Our optimization target is to minimize the overall loss, which can be formulated as:

\[
\mathcal{L} = \mathcal{L}_s + \lambda_u \mathcal{L}_u + \lambda_c \mathcal{L}_c,
\]

where \( \mathcal{L}_s \) and \( \mathcal{L}_u \) represent supervised loss and unsupervised loss applied on labeled images and unlabeled images respectively, and \( \mathcal{L}_c \) is the contrastive loss to make full use of unreliable pseudo-labels. \( \lambda_u \) and \( \lambda_c \) are weights of unsupervised loss and contrastive loss respectively. Both \( \mathcal{L}_s \) and \( \mathcal{L}_u \) are cross-entropy (CE) loss:

\[
\mathcal{L}_s = \frac{1}{|B_l|} \sum_{(x_i^l, y_i^l) \in B_l} \ell_{ce}(f \circ h(x_i^l; \theta), y_i^l), \tag{2}
\]

\[
\mathcal{L}_u = \frac{1}{|B_u|} \sum_{x_i^u \in B_u} \ell_{ce}(f \circ h(x_i^u; \theta), \hat{y}_i^u), \tag{3}
\]

where \( y_i^l \) represents the hand-annotated mask label for the \( i \)-th labeled image, and \( \hat{y}_i^u \) is the pseudo-label for the \( i \)-th unlabeled image. \( f \circ h \) is the composition function of \( h \) and \( f \), which means the images are first fed into \( h \) and then \( f \) to get segmentation results. \( \mathcal{L}_c \) is the pixel-level InfoNCE [32] loss defined as:

\[
\mathcal{L}_c = -\frac{1}{C \times M} \sum_{c=0}^{C-1} \sum_{i=1}^{M} \log \left[ \frac{e^{z_{ci}}/\tau}{e^{z_{ci}}/\tau + \sum_{j=1}^{N} e^{z_{cij}}/\tau} \right], \tag{4}
\]

where \( M \) is the total number of anchor pixels, and \( z_{ci} \) denotes the representation of the \( i \)-th anchor of class \( c \). Each anchor pixel is followed with a positive sample and \( N \) negative samples, whose representations are \( z^+_i \) and \( z_{cij} \) respectively. Note that \( z = g \circ h(x) \) is the output of the representation head. \( \langle \cdot, \cdot \rangle \) is the cosine similarity between features from two different pixels, whose range is limited between \(-1\) to \(1\), hence the need of temperature \( \tau \). Following [28], we set \( M = 50, N = 256 \) and \( \tau = 0.5 \).

3.2. Pseudo-Labeling

To avoid overfitting incorrect pseudo-labels, we utilize entropy of every pixel’s probability distribution to filter high quality pseudo-labels for further supervision. Specifically, we denote \( p_{ij} \in \mathbb{R}^C \) as the softmax probabilities generated by the segmentation head of the teacher model for the \( i \)-th unlabeled image at pixel \( j \), where \( C \) is the number of classes. Its entropy is computed by:

\[
\mathcal{H}(p_{ij}) = -\sum_{c=0}^{C-1} p_{ij}(c) \log p_{ij}(c), \tag{5}
\]

where \( p_{ij}(c) \) is the value of \( p_{ij} \) at \( c \)-th dimension.

Then, we define pixels whose entropy on top \( \alpha_t \) as unreliable pseudo-labels at training epoch \( t \). Such unreliable pseudo-labels are not qualified for supervision. Therefore, we define the pseudo-label for the \( i \)-th unlabeled image at pixel \( j \) as:

\[
\hat{y}_{ij} = \begin{cases} \arg \max_c p_{ij}(c), & \text{if } \mathcal{H}(p_{ij}) < \gamma_t, \\ \text{ignore}, & \text{otherwise}, \end{cases}
\]
where $\gamma_t$ represents the entropy threshold at $t$-th training step. We set $\gamma_t$ as the quantile corresponding to $\alpha_t$, i.e., $\gamma_t = \text{np.percentile}(H.\text{flatten()}, 100 \times (1 - \alpha_t))$, where $H$ is per-pixel entropy map. We adopt the following adjustment strategies in the pseudo-labeling process for better performance.

**Dynamic Partition Adjustment.** During the training procedure, the pseudo-labels tend to be reliable gradually. Based on this intuition, we adjust unreliable pixels’ proportion $\alpha_t$ with linear strategy every epoch:

$$\alpha_t = \alpha_0 \cdot \left(1 - \frac{t}{\text{total epoch}}\right),$$  \hspace{1cm} (7)

where $\alpha_0$ is the initial proportion and is set to 20%, and $t$ is the current training epoch.

**Adaptive Weight Adjustment.**After obtaining reliable pseudo-labels, we involve them in the unsupervised loss in Eq. (3). The weight $\lambda_u$ for this loss is defined as the reciprocal of the percentage of pixels with entropy smaller than threshold $\gamma_t$ in the current mini-batch multiplied by a base weight $\eta$:

$$\lambda_u = \eta \cdot \frac{|B_u| \times H \times W}{\sum_i |B_u| \sum_{j=1}^H \sum_{k=1}^W \mathbb{I}(y_{ij}^u \neq \text{ignore})},$$  \hspace{1cm} (8)

where $\mathbb{I}(\cdot)$ is the indicator function and $\eta$ is set to 1.

**3.3. Using Unreliable Pseudo-Labels**

In semi-supervised learning tasks, discarding unreliable pseudo-labels or reducing their weights is widely used to prevent degradation of model’s performance [37,41,43,50]. We follow this intuition by filtering out unreliable pseudo-labels based on Eq. (6).

However, such contempt for unreliable pseudo-labels may result in information loss. It is obvious that unreliable pseudo-labels can provide information for better discrimination. For example, the white cross in Fig. 2, is a typical unreliable pixel. Its distribution demonstrates model’s uncertainty to distinguish between class person and class motorbike. However, this distribution also demonstrates model’s certainty to not to discriminate this pixel as class car, class train, class bicycle and so on. Such characteristic gives us the main insight to propose our U^2PL to use unreliable pseudo-labels for semi-supervised semantic segmentation.

U^2PL, with a goal to use the information of unreliable pseudo-labels for better discrimination, coincides with recent popular contrastive learning paradigm in distinguishing representation. But due to the lack of labeled images in semi-supervised semantic segmentation tasks, our U^2PL is built on more complicated strategies. U^2PL has three components, named anchor pixels, positive candidates and negative candidates. These components are obtained in a sampling manner from certain sets to alleviate huge computational cost. Next, we will introduce how to selecting: (a) anchor pixels (queries); (b) positive samples for each anchor; (c) negative samples for each anchor.

**Anchor Pixels.** During training, we sample anchor pixels (queries) for each class that appears in the current mini batch. We denote the set of features of all labeled candidate anchor pixels for class $c$ as $A_c^l$,

$$A_c^l = \{z_{ij} \mid y_{ij} = c, p_{ij}(c) > \delta_p\},$$  \hspace{1cm} (9)
where $y_{ij}$ is the ground-truth for the $j$-th pixel of labeled image $i$, and $\delta_p$ denotes the positive threshold for a particular class and is set to 0.3 following [28]. $z_{ij}$ means the representation of the $j$-th pixel of labeled image $i$. For unlabeled data, counterpart $A_u^c$ can be computed as:

$$A_u^c = \{z_{ij} | \hat{y}_{ij} = c, p_{ij}(c) > \delta_p\}.$$  

(10)

It is similar to $A_c^u$, and the only difference is that we use pseudo-label $\hat{y}_{ij}$ based on Eq. (6) rather than hand-annotated label, which implies that qualified anchor pixels are reliable, i.e., $\mathcal{H}(p_{ij}) \leq \gamma_t$. Therefore, for class $c$, the set of all qualified anchors is

$$A_c = A_c^l \cup A_c^u.$$  

(11)

Positive Samples. The positive sample is the same for all anchors from the same class. It is the center of all possible anchors:

$$z_c^+ = \frac{1}{|A_c|} \sum_{z \in A_c} z_c.$$  

(12)

Negative Samples. We define a binary variable $n_{ij}(c)$ to identify whether the $j$-th pixel of image $i$ is qualified to be negative samples of class $c$.

$$n_{ij}(c) = \begin{cases} n_{ij}^l(c), & \text{if image } i \text{ is labeled,} \\ n_{ij}^u(c), & \text{otherwise,} \end{cases}$$  

(13)

where $n_{ij}^l(c)$ and $n_{ij}^u(c)$ are indicators of whether the $j$-th pixel of labeled and unlabeled image $i$ is qualified to be negative samples of class $c$ respectively.

For $i$-th labeled image, a qualified negative sample for class $c$ should be: (a) not belonging to class $c$; (b) difficult to distinguish between class $c$ and its ground-truth category. Therefore, we introduce the pixel-level category order $O_{ij} = \text{argsort}(p_{ij})$. Obviously, we have $O_{ij}(\text{arg max } p_{ij}) = 0$ and $O_{ij}(\text{arg min } p_{ij}) = C - 1$.

$$n_{ij}^l(c) = \mathbb{I}[y_{ij} \neq c] \cdot \mathbb{I}[0 \leq O_{ij}(c) < r_l],$$  

(14)

where $r_l$ is the low rank threshold and is set to 3. The two indicators reflect feature (a) and (b) respectively.

For $i$-th unlabeled image, a qualified negative sample for class $c$ should: (a) be unreliable; (b) probably not belongs to class $c$; (c) not belongs to most unlikely classes. Similarly, we also use $O_{ij}$ to define $n_{ij}^u(c)$:

$$n_{ij}^u(c) = \mathbb{I}[\mathcal{H}(p_{ij}) > \gamma_t] \cdot \mathbb{I}[r_l \leq O_{ij}(c) < r_h],$$  

(15)

where $r_h$ is the high rank threshold and is set to 20. Finally, the set of negative samples of class $c$ is

$$\mathcal{N}_c = \{z_{ij} | n_{ij}(c) = 1\}.$$  

(16)

Category-wise Memory Bank. Due to the long tail phenomenon of the dataset, negative candidates in some particular categories are extremely limited in a mini-batch. In order to maintain a stable number of negative samples, we use category-wise memory bank $\mathcal{Q}_c$ (FIFO queue) to store the negative samples for class $c$. Finally, the whole process to use unreliable pseudo-labels is shown in Algorithm 1. All features of anchors are attached to gradient, come from student hence, while features of positive and negative samples are from teacher.

4. Experiments

4.1. Setup

Datasets. PASCAL VOC 2012 [14] Dataset is a standard semantic segmentation benchmark with 20 semantic classes of objects and 1 class of background. The training set and the validation set include 1,464 and 1,449 images respectively. Following [9, 21, 43], we use SBD [18] as the augmented set with 9,118 additional training images. Since the SBD [18] dataset is coarsely annotated, PseudoSeg [50] takes only the standard 1,464 images as the whole labeled set, while other methods [9, 21] take all 10,582 images as candidate labeled data. Therefore, we evaluate our method on both the classic set (1,464 candidate labeled images) and the blender set (10,582 candidate labeled images). Cityscapes [10], a dataset designed for urban scene understanding, consists of 2,975 training images with fine-annotated masks and 500 validation images. For each dataset, we compare U2PL with other methods under 1/2, 1/4, 1/8, and 1/16 partition protocols.

Network Structure. We use ResNet-101 [19] pre-trained on ImageNet [11] as the backbone and DeepLabv3+ [6] as the decoder. Both of the segmentation head and the rep-

---

**Algorithm 1: Using Unreliable Pseudo-Labels**

1. Initialize $L \leftarrow 0$;
2. Sample labeled images $B_l$ and unlabeled images $B_u$;
3. for $x_i \in B_l \cup B_u$ do
   4. Get probabilities: $p_i \leftarrow f \circ h(x_i; \theta_t)$;
   5. Get representations: $z_i \leftarrow g \circ h(x_i; \theta_s)$;
   6. for $c \leftarrow 0$ to $C - 1$ do
      7. Get anchors $A_c$ based on Eq. (11);
      8. Sample $M$ anchors: $B_A \leftarrow \text{sample}(A_c)$;
      9. Get negatives $\mathcal{N}_c$ based on Eq. (16);
     10. Push $\mathcal{N}_c$ into memory bank $\mathcal{Q}_c$;
     11. Pop oldest ones out of $\mathcal{Q}_c$ if necessary;
     12. Sample $N$ negatives: $B_N \leftarrow \text{sample}(\mathcal{Q}_c)$;
     13. Get $z^+$ based on Eq. (12);
     14. $L \leftarrow L + f(B_A, B_N, z^+)$ based on Eq. (4);
   end
4. end
5. Output: contrastive loss $L_c \leftarrow \frac{1}{|B| \times C} L$.
representation head consists of two Conv-BN-ReLU blocks, where both blocks preserve the feature map resolution and the first block halves the number of channels. The segmentation head can be seen as a pixel-level classifier, mapping the 512 dimensional features output from ASPP module into C classes. The representation head maps the same features into 256 dimensional representation space.

### Evaluation
Following previous methods [15, 21, 33, 48], the images are center cropped into a fixed resolution for PASCAL VOC 2012. For Cityscapes, previous methods apply slide window evaluation, so do we. Then we adopt the mean of Intersection over Union (mIoU) as the metric to evaluate these cropped images. All results are measured on the val set on both Cityscapes [10] and PASCAL VOC 2012 [14]. Ablation studies are conducted on the blender PASCAL VOC 2012 [14] val set under 1/4 and 1/8 partition protocol.

### Implementation Details
For the training on the blender and classic PASCAL VOC 2012 dataset, we use stochastic gradient descent (SGD) optimizer with initial learning rate 0.001, weight decay as 0.0001, crop size as 513 × 513, batch size as 16 and training epochs as 80. For the training on the Cityscapes dataset, we also use stochastic gradient descent (SGD) optimizer with initial learning rate 0.01, weight decay as 0.0005, crop size as 769 × 769, batch size as 16 and training epochs as 200. In all experiments, the decoder’s learning rate is ten times that of the backbone. We use the poly scheduling to decay the learning rate during the training process: \( lr = lr_{\text{base}} \times (1 - \frac{\text{iter}}{\text{total iter}})^{0.9}. \)

### 4.2. Comparison with Existing Alternatives
We compare our method with following recent semi-supervised semantic segmentation methods: Mean Teacher (MT) [38], CCT [33], GCT [22], PseudoSeg [50], CutMix [15], CPS [9], PC²Seg [48], AEL [21]. We re-implement MT [38], CutMix [45] for a fair comparison. For Cityscapes [10], we also reproduce CPS [9] and AEL [21]. All results are equipped with the same network architecture (DeepLabv3+ as decoder and ResNet-101 as encoder). It is important to note the classic PASCAL VOC 2012 Dataset and blender PASCAL VOC 2012 Dataset only differ in training set. Their validation set are the same common one with 1,449 images.

### Results on classic PASCAL VOC 2012 Dataset
Tab. 1 compares our method with the other state-of-the-art methods on classic PASCAL VOC 2012 Dataset. \( \text{U}^2\text{PL} \) outperforms the supervised baseline by +22.21%, +14.23%, +7.78% and +4.47% under 1/16, 1/8, 1/4 and 1/2 partition protocols respectively. For a fair comparison, we only list the methods tested on classic PASCAL VOC
2012. Our method U²PL outperform PC²Seg under all partition protocols by $+10.98\%$, $+2.87\%$, $+3.88\%$ and $+3.11\%$ under $1/16$, $1/8$, $1/4$ and $1/2$ partition protocols respectively. Even under full supervision, our method outperform PC²Seg by $+5.34\%$.

**Results on blender PASCAL VOC 2012 Dataset.** Tab. 2 shows the comparison results on blender PASCAL VOC 2012 Dataset. Our method U²PL outperforms all the other methods under most partition protocols. Compared with the baseline model (trained with only supervised data), U²PL achieves all improvements of $+9.34\%$, $+7.46\%$, $+3.50\%$ and $+3.37\%$ under $1/16$, $1/8$, $1/4$ and $1/2$ partition protocols respectively. Compared with the existing state-of-the-art methods, U²PL surpasses them under all partition protocols. Especially under $1/8$ protocol and $1/4$ protocol, U²PL outperforms AEL by $+1.44\%$ and $+1.24\%$.

**Results on Cityscapes Dataset.** Tab. 3 illustrates the Results on Cityscapes Dataset. To prove our core insight, i.e., using unreliable pseudo-labels promotes semi-supervised semantic segmentation, we conduct experiments about selecting negative candidates (Sec. 3.3) with different reliability. Tab. 4 demonstrates the mIoU results on PASCAL VOC 2012 val set. “Unreliable” outperforms other options, proving using unreliable pseudo-labels does help. Appendix B shows the effectiveness of using unreliable pseudo-labels on Cityscapes.

### 4.3. Ablation Studies

**Effectiveness of Using Unreliable Pseudo-Labels.** To prove our core insight, i.e., using unreliable pseudo-labels promotes semi-supervised semantic segmentation, we conduct experiments about selecting negative candidates (Sec. 3.3) with different reliability. Tab. 4 demonstrates the mIoU results on PASCAL VOC 2012 val set. “Unreliable” outperforms other options, proving using unreliable pseudo-labels does help. Appendix B shows the effectiveness of using unreliable pseudo-labels on Cityscapes.

**Effectiveness of Probability Rank Threshold.** Sec. 3.3 proposes to use probability rank threshold to balance informativeness and confusion caused by unreliable pixels. Tab. 5 provides a verification that such balance promotes the performance. $r_l = 3$ and $r_h = 20$ outperform other options by a large margin. When $r_l = 1$, false negative candidates would not be filtered out, causing the intra-class features of pixels incorrectly distinguished by $\mathcal{L}_c$. When $r_l = 10$, negative candidates tend to become irrelevant with corresponding anchor pixels in semantic, making such discrimination less informative. Appendix D.2 studies PRT and $\alpha_0$ on Cityscapes.

### Table 4. Ablation study on using pseudo pixels with different reliability, which is measured by the entropy of pixel-wise prediction (see Sec. 3.3). “Unreliable” denotes selecting negative candidates from pixels with top 20% highest entropy scores. “Reliable” denotes the bottom 20% counterpart. “All” denotes sampling regardless of entropy.

|               | Unreliable | Reliable | All  |
|---------------|------------|----------|------|
| 1/8 (1323)    | 79.01      | 77.30    | 77.40|
| 1/4 (2646)    | 79.30      | 78.35    | 77.57|

### Table 5. Ablation study on the probability rank threshold, which is described in Sec. 3.3.

| $r_l$ | $r_h$ | 1/8 (1323) | 1/4 (2646) |
|-------|-------|------------|------------|
| 1     | 3     | 78.57      | 79.03      |
| 1     | 20    | 78.64      | 79.07      |
| 3     | 10    | 78.27      | 78.91      |
| 3     | 20    | 79.01      | 79.30      |
| 10    | 20    | 78.62      | 78.94      |

### Table 6. Ablation study on the effectiveness of various components in our U²PL, including unsupervised loss $\mathcal{L}_u$, contrastive loss $\mathcal{L}_c$, category-wise memory bank $\mathcal{Q}_c$, Dynamic Partition Adjustment (DPA), Probability Rank Threshold (PRT), and high entropy filtering (Unreliable).

| $\mathcal{L}_c$ | $\mathcal{Q}_c$ | DPA | PRT | Unreliable | 1/4 (2646) |
|-----------------|-----------------|-----|-----|------------|------------|
| ✓               |                 | ✓   | ✓   | ✓          | 73.02      |
| ✓               | ✓               | ✓   | ✓   | ✓          | 77.08      |
| ✓               | ✓               | ✓   | ✓   | ✓          | 78.49      |
| ✓               | ✓               | ✓   | ✓   | ✓          | 79.07      |
| ✓               | ✓               | ✓   | ✓   | ✓          | 77.57      |
| ✓               | ✓               | ✓   | ✓   | ✓          | 79.30      |

### Table 7. Ablation study on $\alpha_0$ in Eq. (7), which controls the initial proportion between reliable and unreliable pixels.

|       | 40% | 30% | 20% | 10%  |
|-------|-----|-----|-----|------|
| 1/8 (1323) | 76.77 | 77.34 | 79.01 | 77.80 |
| 1/4 (2646) | 76.92 | 77.38 | 79.30 | 77.95 |

**Effectiveness of Components.** We conduct experiments in Tab. 6 to ablate each component of U²PL step by step. For a fair comparison, all the ablations are under $1/4$ partition protocol on blender PASCAL VOC 2012 Dataset. Above all, we use no $\mathcal{L}_c$ trained model as our baseline, achieving mIoU of 73.02% (MT in Tab. 2). Simply adding $\mathcal{L}_c$ without DPA strategy improves the baseline by $+4.06\%$. Category-wise memory bank $\mathcal{Q}_c$, along with PRT and high entropy filtering brings an improvement by $+5.47\%$ to baseline. Dynamic Partition Adjustment (DPA) together with high entropy filtering, brings an improvement by $+6.05\%$ to baseline. Note that DPA is a linear adjustment without tuning (refer to Eq. (7)), which is simple yet efficient.
For Probability Rank Threshold (PRT) component, we set corresponding parameter according to Tab. 5. Without high entropy filtering, the improvement decreased significantly at +4.55%. Finally, when adding all the contribution together, our method achieves state-of-the-art result under 1/4 partition protocol with mIoU of 79.30%. Following this result, we apply these components and corresponding parameters in all experiments on Tab. 2 and Tab. 1.

Ablation Study on Hyper-parameters. We ablate following important parameter for U²PL. Tab. 7 studies the impact of different initial reliable-unreliable partition. This parameter $\alpha_0$ have a certain impact on performance. We find $\alpha_0 = 20\%$ achieves the best performance. Small $\alpha_0$ will introduce incorrect pseudo labels for supervision, and large $\alpha_0$ will make the information of some high-confidence samples underutilized. Other hyper-parameters are studied in Appendix D.1.

4.4. Qualitative Results

Fig. 4 shows the results of different methods on the PASCAL VOC 2012 val set. Benefiting from using unreliable pseudo-labels, U²PL outperforms other methods. Note that using contrastive learning without filtering those unreliable pixels, sometimes does harm to the model (see row 2 and row 4 in Fig. 4), leading to worse results than those when the model is trained only by labeled data.

Furthermore, through visualizing the segmentation results, we find that our method achieves much better performance on those ambiguous regions (e.g., the border between different objects). Such visual difference proves that our method finally makes the reliability of unreliable prediction labels stronger.

5. Conclusion

We propose a semi-supervised semantic segmentation framework U²PL by including unreliable pseudo-labels into training, which outperforms many existing state-of-the-art methods, suggesting our framework provide a new promising paradigm in semi-supervised learning research. Our ablation experiments proves the insight of this work is quite solid. Qualitative result gives a visual proof for its effectiveness, especially the better performance on borders between semantic objects or other ambiguous regions.

The training of our method is time-consuming compared with fully-supervised methods [5, 6, 29, 35, 46], which is a common disadvantage for semi-supervised learning tasks [9, 20, 21, 33, 43, 48]. Due to the extreme lack of labels, the semi-supervised learning frameworks commonly need to pay a price in time for higher accuracy. More in-depth exploration could be conducted on their training optimization in the future.
References

[1] Inigo Alonso, Alberto Sabater, David Férstl, Luis Montesano, and Ana C Murillo. Semi-supervised semantic segmentation with pixel-level contrastive learning from a class-wise memory bank. In Int. Conf. Comput. Vis., 2021. 1, 2

[2] Eric Araoz, Diego Ortega, Paul Albert, Noel E O’Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In IJCNN, 2020. 1, 2

[3] Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. In Adv. Neural Inform. Process. Syst., 2014. 2

[4] Huaihan Chen, Yi Jin, Guoqiang Jin, Changan Zhu, and Enhong Chen. Semi-supervised semantic segmentation by improving prediction confidence. IEEE Transactions on Neural Networks and Learning Systems, 13(1), 2021. 1, 2

[5] 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 Trans. Pattern Anal. Mach. Intell., 40(4):834–848, 2017. 1, 8

[6] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Eur. Conf. Comput. Vis., 2018. 5, 8

[7] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. In Adv. Neural Inform. Process. Syst., 2020. 2

[8] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In Int. Conf. Comput. Vis., 2021. 2

[9] Xiaokang Chen, Yuhui Yuan, Gang Zeng, and Jingdong Wang. Semi-supervised semantic segmentation with cross pseudo supervision. In IEEE Conf. Comput. Vis. Pattern Recogn., 2021. 1, 5, 6, 8, 10

[10] 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 IEEE Conf. Comput. Vis. Pattern Recogn., 2016. 2, 5, 6, 10, 12

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In IEEE Conf. Comput. Vis. Pattern Recogn., 2009. 5

[12] Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? Structural Safety, 31(2):105–112, 2009. 2

[13] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552, 2017. 2

[14] 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(2):303–338, 2010. 1, 2, 5, 6, 10, 12

[15] Geoff French, Samuli Laine, Timo Aila, Michal Mackiewicz, and Graham Finlayson. Semi-supervised semantic segmentation needs strong, varied perturbations. In Brit. Mach. Vis. Conf., 2020. 1, 2, 6

[16] Yves Grandvalet, Yoshua Bengio, et al. Semi-supervised learning by entropy minimization. CAP, 367:281–296, 2005. 2

[17] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Dorsch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheislahgi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. arXiv preprint arXiv:2006.07733, 2020. 2

[18] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In Int. Conf. Comput. Vis., 2011. 5, 6

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In IEEE Conf. Comput. Vis. Pattern Recog., 2016. 5

[20] Ruifei He, Jihan Yang, and Xiaojuan Qi. Re-distributing biased pseudo labels for semi-supervised semantic segmentation: A baseline investigation. In Int. Conf. Comput. Vis., 2021. 8

[21] Hanzhe Hu, Fangyun Wei, Han Hu, Qiwei Ye, Jinshi Cui, and Liwei Wang. Semi-supervised semantic segmentation via adaptive equalization learning. In Adv. Neural Inform. Process. Syst., 2021. 1, 5, 6, 8, 10

[22] Zhanghan Ke, Di Qiu, Kaican Li, Qiong Yan, and Rynson WH Lau. Guided collaborative training for pixel-wise semi-supervised learning. In Eur. Conf. Comput. Vis., 2020. 6

[23] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In Adv. Neural Inform. Process. Syst., 2017. 2

[24] Youngdong Kim, Junho Yim, Juseung Yun, and Junmo Kim. Nnl: Negative learning for noisy labels. In IEEE Conf. Comput. Vis. Pattern Recog., 2019. 3

[25] Youngdong Kim, Juseung Yun, Hyounguk Shon, and Junmo Kim. Joint negative and positive learning for noisy labels. In IEEE Conf. Comput. Vis. Pattern Recog., 2021. 3

[26] Geoffrey Hinton Laurens Van der Maaten. Visualizing data using t-sne. In JMLR, 2008. 12, 13

[27] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In IEEE Conf. Comput. Vis. Pattern Recog., 2021. 1, 2

[28] Shikun Liu, Shuaifeng Zhi, Edward Johns, and Andrew J Shelhamer. Guided collaborative training for pixel-wise semi-supervised learning. In Int. Conf. Mach. Learn., 2020. 1, 8

[29] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In IEEE Conf. Comput. Vis. Pattern Recog., 2015. 1, 8

[30] Jishnu Mukhoti and Yarin Gal. Evaluating bayesian deep learning methods for semantic segmentation. arXiv preprint arXiv:1811.12709, 2018. 2

[31] Viktor Olsson, Wilhelm Tranaheden, Juliano Pinto, and Lennart Svensson. Classmix: Segmentation-based data augmentation for semi-supervised learning. In IEEE Conf. Comput. Vis. Pattern Recog., 2021. 2
Appendix

We organize the Appendix as follows. Above all, more details for reproducing the results will be given in Appendix A. Then we will give more results on Cityscapes from two perspectives in Appendix B. We also provide an alternative of contrastive learning to prove our main insight does not only rely on contrastive learning in Appendix C. Besides, ablation studies on both PASCAL VOC 2012 and Cityscapes for more hyper-parameters are given in Appendix D. Finally, visualization on feature space gives a visual proof for the effectiveness of U²PL in Appendix E.

A. More Details for Reproducibility

For Cityscapes [10], we utilize OHEM which is the same as previous methods [9, 21]. The temperature τ is set to 0.5 for both PASCAL VOC 2012 [14] and Cityscapes [10]. We use SGD optimizer for all experiments. For experiments in PASCAL VOC 2012 [14], the initial base learning rate is 0.001 and the weight decay is 0.0001. For experiments in Cityscapes [10], the initial base learning rate is 0.01 and the weight decay is 0.0005. In our experiments, we find if we train the model only with supervised loss for the initial a few epochs then apply U²PL, it can achieve better performance. We define such epoch as the warm start epoch, and the weight decay is 0.0001.

For Cityscapes [10], the initial base learning rate is 0.01 and the weight decay is 0.0005. In our experiments, we find if we train the model only with supervised loss for the initial a few epochs then apply U²PL, it can achieve better performance. We define such epoch as the warm start epoch, and the corresponding warm start epochs for PASCAL VOC 2012 and Cityscapes are 1 and 20 respectively.

To prevent overfitting, we apply random cropping, random horizontal flipping, and random scaling with the range of [0.5, 2.0] for both PASCAL VOC 2012 [14] and Cityscapes [10] following previous methods [9, 21, 48, 51].
Figure A1. Qualitative results on Cityscapes val set. All models are trained under the 1/2 partition protocol, which contains 1,488 labeled images and 1,487 unlabeled images. (a) Input images. (b) Hand-annotated labels for the corresponding image. (c) Only labeled images are used for training. (d) The vanilla contrastive learning framework, where all pixels are used as negative samples without entropy filtering. (e) Predictions from our U\textsuperscript{2}PL. Yellow rectangles highlight the promotion by adequately using unreliable pseudo-labels.

Our memory queue is category-specific. For the background category, the length of the queue is set to be 50,000. For foreground categories, the length of the queue is all 30,000. All baselines i.e., “SupOnly”, “MT”, and “CutMix” are re-implemented by ourselves, where the only difference between “MT” and “CutMix” is that the latter applies CutMix [45] augmentation for unlabeled images.

The hyper-parameters used in this work are listed in Tab. A1. Among them, \( M, N, \delta_p \) are used for contrastive learning, for which we simply follow [28]. \( \lambda_c, \eta, \tau \) are training-related, while \( \alpha_0, r_l, r_h \) are additionally introduced by our U\textsuperscript{2}PL.

Table A1. Summary of hyper-parameters used in U\textsuperscript{2}PL.

| Symbol | Description | Default Value |
|--------|-------------|---------------|
| \( (M, N) \) | contrastive learning settings | (50, 256) |
| \( \delta_p \) | confidence threshold of positive samples | 0.3 |
| \( (\lambda_c, \eta) \) | loss weights | (0.1, 1) |
| \( \tau \) | loss temperature | 0.5 |
| \( \alpha_0 \) | initial proportion of unreliable pixels | 20\% |
| \( (r_l, r_h) \) | probability rank thresholds | (3, 20) |

Table A2. Ablation study on using pseudo pixels with different reliability, which is measured by the entropy of pixel-wise prediction. “Unreliable” denotes selecting negative candidates from pixels with top 20\% highest entropy scores. “Reliable” denotes the bottom 20\% counterpart. “All” denotes sampling regardless of entropy. We prove this effectiveness under 1/2 and 1/4 partition protocol on Cityscapes val set.

| | Unreliable | Reliable | All |
|---|-------------|----------|-----|
| 1/2 (1488) | 79.05 | 77.19 | 76.96 |
| 1/4 (744) | 76.47 | 75.16 | 74.51 |

B. More Results on Cityscapes

Quantitative Results. Tab. A2 demonstrates the mIoU results on Cityscapes val set. “Unreliable” outperforms other options, proving using unreliable pseudo-labels does help. U\textsuperscript{2}PL fully mines the information of all pixels.

Qualitative Results. Fig. A1 shows the results of different methods on the Cityscapes val set. Benefiting by using unreliable pseudo-labels, U\textsuperscript{2}PL outperforms other methods. Note that using contrastive learning without filtering those unreliable pixels, sometimes does harm to the model (see the 1-st row and the 4-th row in Fig. A1), leading to worse results than those when the model is trained only by labeled data. Such visual difference proves that our method finally makes the reliability of unreliable prediction labels stronger.

C. Alternative of Contrastive Learning

Our proposed U\textsuperscript{2}PL is not limited by contrastive learning. Binary classification is also a sufficient way to use unreliable pseudo-labels, i.e., using binary cross-entropy loss (BCE) \( L_0 \) other than contrastive loss. For \( i \)-th anchor \( z_{ci} \) belongs to class \( c \), we simply use its negative samples
\{z_{cij}\}_{i=1}^{N}
and positive sample \(z_c^+\) to compute the BCE loss:

\[
\mathcal{L}_b = -\frac{1}{C \times M \times N} \sum_{c=0}^{C-1} \sum_{i=1}^{M} \sum_{j=1}^{N} \log \left( \frac{e^{\langle z_{ci}, z_{cj}^+ \rangle} / \tau}}{e^{\langle z_{ci}, z_{cj}^+ \rangle} / \tau + e^{\langle z_{ci}, z_{cj} \rangle} / \tau}} \right),
\]

where \(C\), \(M\), and \(N\) are the total number of classes, anchor pixels, and negative samples, respectively. \(\langle \cdot, \cdot \rangle\) is the cosine similarity of two features, and \(\tau\) represents the temperature.

Tab. A3 and Tab. A4 are results of using unreliable pseudo-labels based on binary classification on Cityscapes [10] and PASCAL VOC 2012 [14] val set respectively. From Tab. A3 and Tab. A4, we can tell that our U^2PL is not restricted by contrastive learning, a basic binary classification also does help. On Cityscapes val set, U^2PL with \(\mathcal{L}_b\) can outperforms supervised only baseline by +3.77%, +0.40%, +1.48%, and +0.53% under 1/16, 1/8, 1/4, and 1/2 partial protocols. U^2PL with \(\mathcal{L}_b\) can outperforms supervised only baseline by +7.49%, +5.07%, +3.84%, and +2.67% under 1/16, 1/8, 1/4, and 1/2 partial protocols on PASCAL VOC 2012 val set. Note that under the 1/4 partition protocol of blender VOC PASCAL 2012, the binary classification based U^2PL (w/ \(\mathcal{L}_b\)) outperforms the contrastive learning based U^2PL (w/ \(\mathcal{L}_c\)) by +0.34%, which proves that contrastive learning is not the only efficient way of using unreliable pseudo-labels.

### D. More Ablation Studies

#### D.1. More Hyper-parameters on VOC

**Base Learning Rate.** The impact of the base learning rate is shown in Tab. A5. Results are based on U^2PL on blender VOC PASCAL 2012 Dataset. We find that 0.001 outperforms other alternatives.

**Temperature.** Tab. A6 gives a study on the effect of temperature \(\tau\). Temperature \(\tau\) plays an important role to adjust the importance to hard samples. When \(\tau = 0.5\), our U^2PL achieves best results. Too large or too small of \(\tau\) will have an adverse effect on overall performance.

**D.2. Ablation Studies on Cityscapes**

**Probability Rank Threshold.** Tab. A7 provides a verification that such balance promotes the performance. \(r_t = 3\) and \(r_h = 20\) outperform other options by a large margin.

**Initial Reliable-Unreliable Partition.** Tab. A8 studies the impact of different \(\alpha_0\). When \(\alpha_0 = 20\%\), the model achieves the best performance.
Table A5. Ablation study on base learning rate under 1/4 partition protocol (2646) in blender VOC PASCAL 2012 Dataset.

| $l_{r_{\text{base}}}$ | $10^{-1}$ | $10^{-2}$ | $10^{-3}$ | $10^{-4}$ | $10^{-5}$ |
|----------------------|-----------|-----------|-----------|-----------|-----------|
| mIoU                 | 3.49      | 77.82     | **79.30** | 74.58     | 65.69     |

Table A6. Ablation study on temperature under 1/4 partition protocol (2646) in blender VOC PASCAL 2012 Dataset.

| $\tau$  | 10  | 1   | 0.5 | 0.1 | 0.01 |
|---------|-----|-----|-----|-----|------|
| mIoU    | 78.88 | 78.91 | **79.30** | 79.22 | 78.78 |

Table A7. Ablation study on PRT on Cityscapes val set.

| $r_l$ | 1 | 1 | 3 | 3 | 10 |
|-------|---|---|---|---|----|
| $r_h$ | 3 | 20 | 10 | 20 | 20 |

|       | 1/8 (372) | 1/4 (744) |
|-------|-----------|-----------|
| mIoU  | 71.41     | 76.27     |
|       | 72.08     | 76.04     |
|       | 72.60     | 76.01     |
|       | **74.37** | **76.47** |
|       | 72.24     | 76.18     |

Table A8. Ablation study on $\alpha_0$ on Cityscapes val set.

| $\alpha_0$ | 40% | 30% | 20% | 10% |
|------------|-----|-----|-----|-----|
| mIoU       | 72.07 | 72.93 | **74.37** | 71.63 |
|           | 75.20 | 76.08 | **76.47** | 76.40 |

E. Visualization on Feature Space

To have a better understanding of U$^2$PL, we give an illustration on visualization of feature space. Two t-SNE [26] plots are given respectively on the supervised only method and U$^2$PL.

We can observe from Fig. A2 that decision boundaries of features generated by the supervised only method are quite confusing, while U$^2$PL has much more clear ones. This explains why U$^2$PL works from a feature point of view.