PRCL: Probabilistic Representation Contrastive Learning for Semi-Supervised Semantic Segmentation

Haoyu Xie[1], Changqi Wang[2], Jian Zhao[3], Yang Liu[4], Jun Dan[5], Chong Fu[6], Baigui Sun[7]

1) First author, 2010643@stu.neu.edu.cn, School of Computer Science and Engineering, Northeastern University, Shenyang, 110819, China
2) First author, 2101668@stu.neu.edu.cn, School of Computer Science and Engineering, Northeastern University, Shenyang, 110819, China
3) First author, Corresponding author, zhaojian90@u.nus.edu, Intelligent Game and Decision Laboratory, Beijing 100191, China
4) ly261666@alibaba-inc.com, Alibaba DAMO Academy, Alibaba Group, Hangzhou 310000, China
5) danjun@zju.edu.cn, Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China
6) fuchong@mail.neu.edu.cn, School of Computer Science and Engineering, Northeastern University, Shenyang, 110819, China
7) Corresponding author, baigui.sbg@alibaba-inc.com , Alibaba DAMO Academy, Alibaba Group, Hangzhou 310000, China

Abstract Tremendous breakthroughs have been developed in Semi-Supervised Semantic Segmentation (S4) through contrastive learning. However, due to limited annotations, the guidance on unlabeled images is generated by the model itself, which inevitably exists noise and disturbs the unsupervised training process. To address this issue, we propose a robust contrastive-based S4 framework, termed the Probabilistic Representation Contrastive Learning (PRCL) framework to enhance the robustness of the unsupervised training process. We model the pixel-wise representation as Probabilistic Representations (PR) via multivariate Gaussian distribution and tune the contribution of the ambiguous representations to tolerate the risk of inaccurate guidance in contrastive learning. Furthermore, we introduce Global Distribution Prototypes (GDP) by gathering all PRs throughout the whole training process. Since the GDP contains the information of all representations with the same class, it is robust from the instant noise in representations and bears the intra-class variance of representations. In addition, we generate Virtual Negatives (VNs) based on GDP to involve the contrastive learning process. Extensive experiments on two public benchmarks demonstrate the superiority of our PRCL framework.

Key Word Semi-Supervised Semantic Segmentation; Contrastive Learning; Probabilistic Representation; Robust Learning;

Statements and Declarations The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Noname manuscript No.
(will be inserted by the editor)

PRCL: Probabilistic Representation Contrastive Learning for Semi-Supervised Semantic Segmentation

Haoyu Xie[1,3,*] · Changqi Wang[1,*] · Jian Zhao[2,*] ✉ · Yang Liu[3] · Jun Dan[4] · Chong Fu[1] · Baigui Sun[3] ✉

Received: date / Accepted: date

Abstract Tremendous breakthroughs have been developed in Semi-Supervised Semantic Segmentation (S4) through contrastive learning. However, due to limited annotations, the guidance on unlabeled images is generated by the model itself, which inevitably exists noise and disturbs the unsupervised training process. To address this issue, we propose a robust contrastive-based S4 framework, termed the Probabilistic Representation Contrastive Learning (PRCL) framework to enhance the robustness of the unsupervised training process. We model the pixel-wise representation as Probabilistic Representations (PR) via multivariate Gaussian distribution and tune the contribution of the ambiguous representations to tolerate the risk of inaccurate guidance in contrastive learning. Furthermore, we introduce Global Distribution Prototypes (GDP) by gathering all PRs throughout the whole training process. Since the GDP contains the information of all representations with the same class, it is robust from the instant noise in representations and bears the intra-class variance of representations. In addition, we generate Virtual Negatives (VNs) based on GDP to involve the contrastive learning process. Extensive experiments on two public benchmarks demonstrate the superiority of our PRCL framework.

Keywords Semi-Supervised Semantic Segmentation; Contrastive Learning; Probabilistic Representation; Robust Learning

* equal contribution

✉Corresponding author

1. School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China
2. Intelligent Game and Decision Laboratory, Beijing 100191, China
3. Alibaba DAMO Academy, Alibaba Group, Hangzhou 310000, China
4. Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China

1 Introduction

Semantic Segmentation is a fundamental task in computer vision, aiming to predict the class of each pixel. Significant progress has been made via training a segmentation model [20, 3, 61] on large-scale annotated images, which requires high labor costs. Semi-Supervised Semantic Segmentation (S4) leverages unlabeled images in the training process to further improve the performance of the segmentation model via adversarial training [23], consistency regularization [44], and self-training [51], which ease the thirsty of annotated images.

Self-training is a well-known paradigm extensively employed in S4 tasks. It involves leveraging a model pretrained on labeled images to generate predictions, *a.k.a*, pseudo-labels, for unlabeled images. These pseudo-labels in conjunction with annotations are subsequently used as guidance to retrain the model. Recent powerful methods introduce the concept of pixel-wise contrastive learning to the self-training paradigm, aiming to explore the semantic information not only in the local context of a single image but also in the images in a mini-batch or even the entire dataset. This is achieved by projecting pixels to representations in the latent space, where representations from the same class are aggregated around their class centroid, *a.k.a*, prototype, while those from different classes, *a.k.a*, negative representations, are separated. In the context of semi-supervised learning, the semantic guidance of unlabeled images for contrastive learning comes from the pseudo-labels during training. Consequently, the quality of pseudo-labels is critical in contrastive learning since inaccurate pseudo-labels lead to assigning representations to wrong classes and cause a disorder in latent space. Existing methods have attempted to refine pseudo-labels via their corresponding confidence [34] or entropy [11]. While these techniques have shown promise in enhancing the quality of pseudo-labels and eliminating inaccurate ones to some extent, they rely on delicate strategies and still struggle to fully

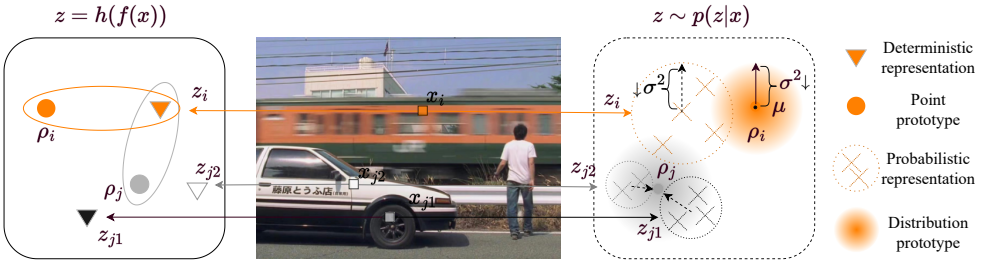

Fig. 1 Contradistinction between two types of representations and prototypes. Point prototype means the prototype of the deterministic representation and distribution prototype means the prototype of the probabilistic representation. Distinct from conventional representation, we introduce probability, and thus regarding representation as a multivariate Gaussian distribution. The probabilistic representation is able to demolish the ambiguity of representation prototype mapping to some extent and enhance the robustness of the model during training fuzzy pixels.

address the inherent noise and incorrectness in pseudo-labels. Motivated by these challenges, our goal is to improve the robustness of representations, enabling them to perform more effectively even in the presence of inaccurate pseudo-labels. By focusing on enhancing the robustness of representations to accommodate imperfect guidance, we are able to enhance the overall performance and reliability of contrastive-based S4 approaches.

In contrast to existing conventional *deterministic* representation modeling, which maps the representation to the deterministic point in the latent space, our proposed method introduces a novel perspective by treating representations as random variables with learnable parameters, termed Probabilistic Representation (PR). Specifically, we adopt a multivariate Gaussian distribution to model the representations, thereby obtaining distribution prototypes. As illustrated in Fig. 1, this probabilistic modeling is reflected in the expression $z \sim p(z|x)$. The pixel of the fuzzy train carriage x_i is mapped to z_i in the latent space which encompasses two components: the most likely representation μ (mean) and the probability σ^2 (variance) of the distribution. Similarly, the pixels of the car x_{j1} and x_{j2} are mapped to z_{j1} and z_{j2} respectively. For comparison, deterministic mapping is shown in $z = h(f(x))$. In scenarios where the distance between the representation z_i to prototype ρ_i is the same as the distance from z_i to ρ_j, deterministic representation encounters an ambiguity in mapping z_i to either ρ_i or ρ_j. On the contrary, in the latent space of probabilistic representations, z_i is mapped to ρ_i since ρ_i possesses a smaller value of σ^2 compared to ρ_j. It is worth noting that σ^2 is inversely proportional to the probability, indicating that the mapping from z_i to ρ_i is more reliable than mapping to ρ_j according to the probability.

Meanwhile, recent contrastive-based S4 works suffer from the limitation of solely considering the contrast in the current iteration. Specifically, the prototypes in those methods are obtained by aggregating the semantic information of representations belonging to the same class in the current iteration. This approach can result in prototype shifts across adjacent iterations due to discrepancies among representations caused by inaccurate pseudo-labels and intra-class variance. We argue that prototype consistency is crucial to establish a stable direction for representation aggregation. In addition, since negative representations are from the current mini-batch when solely considering the contrast in the current iteration, the distribution of negative representations is fragmentary due to the limited size of the mini-batch. Some approaches [1, 21] leverage the memory bank strategy to compensate for the fragmentary distribution, which stores representations in the external memory and sample from them when constructing negative distribution. However, dense pixel-wise representations lead to significant memory overhead and high computational cost. To overcome the above limitations, we rethink pixel-wise contrastive learning from the global perspective and build Global Distribution Prototypes (GDP) based on probabilistic representations. Distinct from the conventional prototypes which represent the semantic information of identical class representations in the current mini-batch, GDPs aggregate the representations along training iterations and are updated with the prototypes of the current iteration as an observation of Bayesian Estimation [52]. By employing GDPs to bridge iterations during the training process, our method enables prototypes to withstand instant noise in representations and accommodate intra-class variance among identical class representations. Therefore, the prototype is more consistent across iterations and provides a stable direction for identical representation aggregation. In contrast to the conventional approach of constructing a memory bank to provide a large number of negatives, we propose a novel strategy called Virtual Negatives (VN). By leveraging a reparameter trick from the GDPs, we generate VNs and facilitate a balance between compactness and diversity of them through a virtual radius. Notably, compared with the typical memory bank solution, our VN reduces the GPU memory from 2.63GB to 42KB, and also accelerates the training process for 22.23%. Most importantly, our introduced VN strategy performs better than the conventional memory bank strategy.

This work is based on our previous conference version [1] by tackling the limitations of merely considering

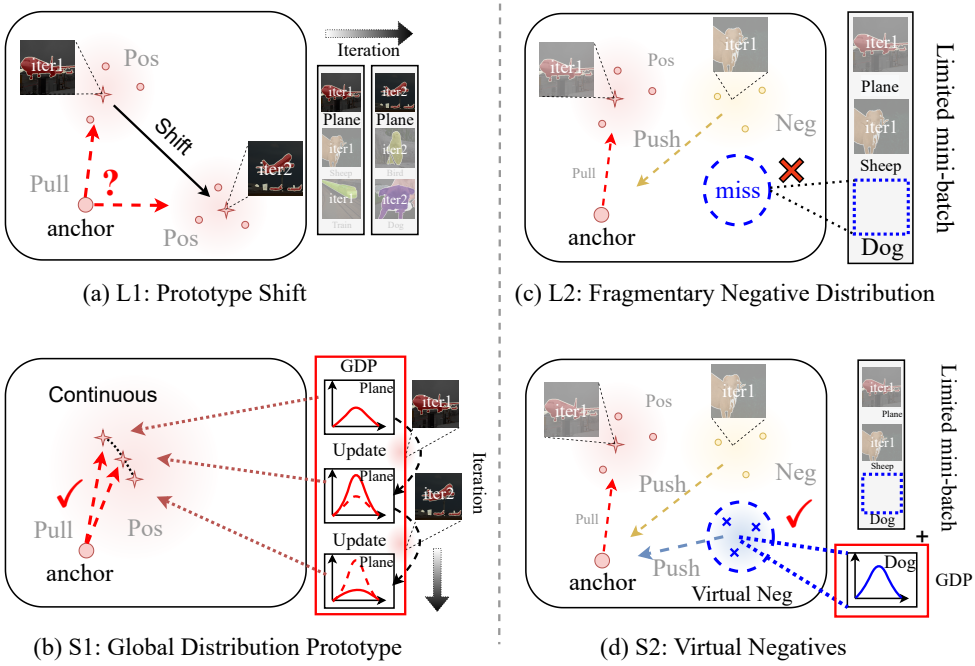

Fig. 2 Our PRCL framework tackles the negative effect brought by prototype shift and fragmentary negative distribution (L1 and L2) with our proposed global distribution prototype and virtual negatives. (S1 and S2).

the contrast in the current iteration, including the prototype shift between adjacent iterations and fragmentary negative distribution. Specifically, we propose GDP with an update strategy and VN to maintain the prototype consistency and compensate for the fragmentary negative distribution, respectively. To summarize, our main contributions are four-fold:

- We introduce the *probabilistic representation* and improve the robustness of representations in contrastive learning, which eases the negative effects of inaccurate pseudo-labels.
- We build *global distributional prototypes* and *virtual negatives* to make up for the defects brought by the limited size of the mini-batch, which is more memory efficient and faster than the conventional memory bank strategy.
- Extensive experiments on PASCAL VOC 2012 and Cityscapes demonstrate the effectiveness of our proposed method.
- We present comprehensive ablation studies and in-depth analysis of probabilistic representation, global distributional, and virtual negatives, which demonstrate that our method not only improves the robustness and performance of segmentation model in S4.

2 Related Work

2.1 Semi-supervised Semantic Segmentation

Semantic segmentation aims to classify each pixel in an entire image by class. Training models for this task typically requires a substantial amount of labeled data, involving meticulous manual annotations. Semi-supervised learning methods have emerged as effective approaches to leverage large volumes of unlabeled data, thereby reducing the dependency on extensive manual annotations. Self-training [23,27,64,45] and consistency regularization [42,44,66,10] are two widely-used paradigms. Self-training methods leverage high-dimensional perturbations [12,40,22] and refined pseudo-labels [50,11] to enhance their performance. Some methods [23,26,32,37] based on GANs [13] and adversarial learning [38] concentrate on generating more ground-truth like predictions. Additionally, methods focusing on balancing class distribution [15,19,67,57] have demonstrated competitiveness in specific scenarios. Recent works based on self-training [34,56,60] emphasize the regularization of representations in the latent space to maintain an ordered latent space. This improves the quality of features and ultimately boosts model performance, which is also our goal.

2.2 Pixel-wise Contrastive Learning

Distinct from instance-wise contrastive learning [58,65,4,17,14,59], which treats each image as an individual class and distinguishes it from other images through multiple views. In the case of pixel-wise contrastive learning [31,21,54,62,24,53,28], dense pixel-wise representations are distinguished by semantic guidance, *i.e.*, labels or pseudo-labels. However, in the semi-supervised setting, the availability of labeled images is limited, and the majority of pixel classifications rely on pseudo-labels, which

can introduce inaccuracies and disrupt the latent space. To address these challenges, previous methods [34, 1, 56] have attempted to refine pseudo-labels using threshold sampling strategies. In contrast, our approach focuses on improving the quality of representations and accommodating inaccurate pseudo-labels, rather than solely filtering them out. By emphasizing the enhancement of representation quality, we aim to ease the negative effects of inaccurate pseudo-labels and foster a more robust and ordered latent space.

The process of pixel-wise contrastive learning is to aggregate representations of the same class to their prototype (class centroid) and separate them away from negative representations (representations with different classes). Due to limited GPU memory, the prototype merely gathers the semantic information of representation in the current iteration in most methods [34, 56], thereby disregarding the global semantic information of the entire dataset. To address this limitation, some methods have proposed the use of a memory bank to store representations from past iterations [21] or update the prototype using Exponential Moving Average (EMA) [63, 68]. In our approach, we introduce a novel strategy that considers all historical representations, overcoming the limitation. As for negative representations, some methods [34] sample them from the current iteration. However, due to the limited batch size, representations in the current iteration may not cover all classes, leading to a fragmentary negative distribution problem. To mitigate this issue, some methods try to alleviate it by introducing the memory bank [55, 1, 56] or approximating the ideal negative distribution through a probabilistic way [60]. In our method, we compensate for the negative distribution by generating representations, which takes little memory consumption and minimal computational cost.

2.3 Probabilistic Embedding

Probabilistic Embedding (PE) extends the concept of conventional embeddings by predicting the overall distribution of embeddings *e.g.*, Gaussian [49] and von Mises-Fisher [33], instead of a single vector. The ability of neural networks to predict distributions stems from the work of Mixture Density Networks (MDN) [2]. Variable Auto-Encoders (VAE) [29] introduced the use of MLP to predict the mean and variance of a distribution, which serves as the foundation for many probabilistic embedding approaches [49, 48, 47, 43]. Hedged Instance emBeddings (HIB) [39] attempt to apply PE to image retrieval and verification tasks. Subsequently, PE is applied to face verification tasks. Probabilistic Face Embeddings (PFE) [49] maps each image to a Gaussian distribution in the latent space, with the mean predicted by a pretrained model and the variance estimated through an MLP (probability head). Sphere Confidence Face (SCF) [33] maps images to a von Mises-Fisher distribution with mean and concentration parameters. We adopt a similar architecture, but it is important to note that PFE and SCF optimize the mean and variance (concentration parameter) in two separate stages. Specifically, they pretrain a deterministic model to predict the mean and then freeze it while optimizing the variance. In our work, however, we optimize the mean and variance simultaneously, enabling them to interact with each other.

While conventional distance metrics measure the similarity between distributions based on their means, they are inadequate for capturing probabilistic similarity due to the variance component. To address this challenge, HIB employs the reparameter trick [29] to obtain two sets of samples from two distributions through Monte-Carlo sampling, and accumulates the similarity of samples to represent the similarity of distributions. In contrast, PFE and SCF directly compute distribution similarity using the mutual likelihood score. However, these methods are unable to optimize the mean and variance simultaneously since uncertainty/probability (variance) is only informative if representation (mean) is reasonable. PFE and SCF tackle this issue by training the mean and variance in two separate stages. In our approach, we address this by training the mean and variance separately with different learning rates.

3 Methodology

In the S4 task, we can achieve a small labeled set $\mathcal{D}_l = \{(\boldsymbol{x}_i^l, \boldsymbol{y}_i^l)\}_{i=1}^{N_l}$ and a large unlabeled set $\mathcal{D}_u = \{\boldsymbol{x}_i^u\}_{i=1}^{N_u}$, where labeled dataset $\mathcal{D}_l$ contains N_l pairs of images and corresponding pixel-wise labels $(\boldsymbol{x}^l, \boldsymbol{y}^l)$ and unlabeled dataset only contains N_u unlabeled images. Our goal is to train a segmentation model with $\mathcal{D}_l$ and $\mathcal{D}_u$. The base segmentation model contains an encoder $f(\cdot)$ and a segmentation head $g(\cdot)$. We adopt the teacher-student paradigm and pixel-wise contrastive learning to our framework, described in Sec. 3.1.

3.1 MT and Contrastive Learning

The standard teacher-student paradigm consists of two segmentation models with the same architecture, named the student model and the teacher model respectively. We denote $f(\cdot)$, $g(\cdot)$ as the encoder and segmentation head in the student model and $f'(\cdot)$, $g'(\cdot)$ as those of the teacher model. The student model parameters are optimized via Stochastic Gradient Descent (SGD) to minimize the loss function $\mathcal{L}$ while the parameters in the teacher model are updated by the Exponential Moving Average (EMA) of the parameters in the student model. The pseudo-labels $\boldsymbol{y}_i^u$ for training the student model during the unsupervised process are produced based on the output logits from the teacher model, *i.e.*, $\boldsymbol{p}_i^u = g'(f'(\boldsymbol{x}_i^u))$, formulated as:

$$\boldsymbol{y}_i^u = \mathbf{1}_c(\arg\max_c \{p_{i,c}^u\}_{c \in C}), \tag{1}$$

where $p^u_{i,c}$ denotes value of $\boldsymbol{p}^u_i$ on c^{th} dimension, $\mathbf{1}_c(\cdot)$ denotes the one-hot encoding of class c and C denotes the set of total classes in the dataset. In order to additionally regularize the model's output in the latent space, recent methods [1,34,56,54,21,25,55] introduce contrastive learning to the teacher-student paradigm. Concretely, a representation head $h(\cdot)$ is introduced to the student model to map pixels to representations, *i.e.*, $\boldsymbol{z_i} = h(f(\boldsymbol{x}_i))$. The representations of the identical class are aggregated to their prototype $\boldsymbol{\rho}$ and those of different classes are separated via a contrastive loss (*e.g.*, InfoNCE). The prototype $\boldsymbol{\rho}$ of each class is obtained by gathering the semantic information of the identical class representations. The semantic guidance of contrastive learning is also from $\boldsymbol{y}^l_i$ and $\boldsymbol{y}^u_i$.

Discussion. In this paper, we argue that recent S4 models based on pixel-wise contrastive learning suffer from two potential limitations: **1)** *The model suffers from poor robustness in the case of inaccurate pseudo-labels.* Since the pseudo-labels $\boldsymbol{y}^u_i$ are derived solely from the prediction of the teacher model's segmentation head $g'(\cdot)$, there exist inaccurate ones due to the limited cognitive ability of the teacher model. Even though the quality of pseudo-labels can be improved by adopting delicate sampling strategies, the essential errors in pseudo-labels are rather hard to be eliminated. Those inaccurate pseudo-labels will mislead the model if they are directly applied as the supervision during training on $\mathcal{D}_u$. The strategy of learning a robust contrastive-based S4 model under the inaccurate pseudo-labels has been overlooked and remains unexplored. **2)** *The prototype is shifted and the negative distribution is fragmentary.* In recent works, the prototype $\boldsymbol{\rho}$ is calculated as the mean of the identical class representations in the current iteration. However, due to incorrect pseudo-labels and intra-class variance, the representations of the same class can significantly vary across different iterations, resulting in the position of the prototype changing dramatically, termed prototype shift. We argue that this shift in the prototype hinders the provision of a consistent direction for aggregating identical representations and leads to a disordered latent space. Additionally, since the negative representations are from the current mini-batch, the limited mini-batch size leads to a fragmentary distribution of negatives for the contrastive learning process within the current iteration. While some approaches attempt to address this limitation by employing a memory bank to store representations from past iterations and sample negatives from it, this strategy incurs high memory usage and computational costs.

To mitigate these limitations, we build a framework that is robust against inaccurate pseudo-labels. The representations in our framework are modeled via Gaussian distribution, described in Sec. 3.2. We build Global Distribution Prototypes (GDP) to maintain the consistency of prototypes in Sec. 3.3 and obtain Virtual Negatives (VN) based on GDPs for compensating fragmentary negative distribution, described in Sec. 3.4.

3.2 Probabilistic Representation

In this section, we detail the process of building our probabilistic representations and the similarity measurement for probabilistic representations.

We denote the probability of mapping a pixel $\boldsymbol{x}_i$ to a representation $\boldsymbol{z}_i$ as $p(\boldsymbol{z}_i|\boldsymbol{x}_i)$ and define the representation as a random variable following it. For simplicity, we take the form of multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\sigma}^2_i \boldsymbol{I})$ as:

$$p(\boldsymbol{z}_i|\boldsymbol{x}_i) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_i, \boldsymbol{\sigma}^2_i \boldsymbol{I}), \tag{2}$$

where $\boldsymbol{I}$ represents the unit diagonal matrix. In this formulation, $\boldsymbol{\mu}$ represents the most likely values for the representation, while $\boldsymbol{\sigma}^2$ captures the associated probability. It is worth noting that $\boldsymbol{\sigma}^2$ is inversely related to the probability, meaning that larger $\boldsymbol{\sigma}^2$ indicate lower probabilities. Both $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$ have the same dimensions. The mean $\boldsymbol{\mu}$ is predicted by the representation head $h(\cdot)$. Meanwhile, we introduce a probability head $p(\cdot)$ in parallel to predict the probability $\boldsymbol{\sigma}^2$.

In conventional contrastive learning, the similarity between representations is typically measured using the ℓ_2 distance or cosine similarity, which does not possess the ability to quantify the similarity between two distributions. To solve this problem, we employ the Mutual likelihood Score (MLS) as the measurement of the similarity between two distributions $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$, as follows:

$$\begin{aligned} MLS(\boldsymbol{z}_i, \boldsymbol{z}_j) &= \log(p(\boldsymbol{z}_i = \boldsymbol{z}_j)) \\ &= -\frac{1}{2}\sum_{l=1}^{D}\Big(\frac{(\mu^{(l)}_i - \mu^{(l)}_j)^2}{\sigma^{2(l)}_i + \sigma^{2(l)}_j} \\ &\quad + \log(\sigma^{2(l)}_i + \sigma^{2(l)}_j)\Big) - \frac{D}{2} log 2\pi, \end{aligned} \tag{3}$$

where $\mu^{(l)}_i$ denotes to the l^{th} dimension of $\boldsymbol{\mu}_i$ and the same for $\sigma^{(l)}_i$. MLS combines a weighted ℓ_2 distance and a log regularization term, essentially. The conventional ℓ_2 distance solely considers the similarity between representations mapped in the latent space based on pseudo-labels, without taking into account their reliability. However, inaccurate pseudo-labels can lead to incorrect optimization directions and disrupt the latent space. To address this issue, the MLS incorporates the probabilities of $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ to account for inaccurate pseudo-labels from two perspectives: **(i)**: In the first term, the weight of ℓ_2 distance is reduced when the $\boldsymbol{\sigma}^2$ is large. This indicates that the similarity between $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ decreases due to the low probabilities, even if their ℓ_2 distance suggests they are similar. By considering the probabilities, the MLS incorporates a measure of similarity that accounts for the reliability of representations. **(ii)**: In the second term, the log regularization penalizes low probability representations, which encourages all representations to be more reliable. Additionally, $\boldsymbol{\sigma}^2$ and $\boldsymbol{\mu}$ can interact with each other. The learnable $\boldsymbol{\sigma}^2$ is associated with ℓ_2 distance, allowing it to be learned based on the relationships among

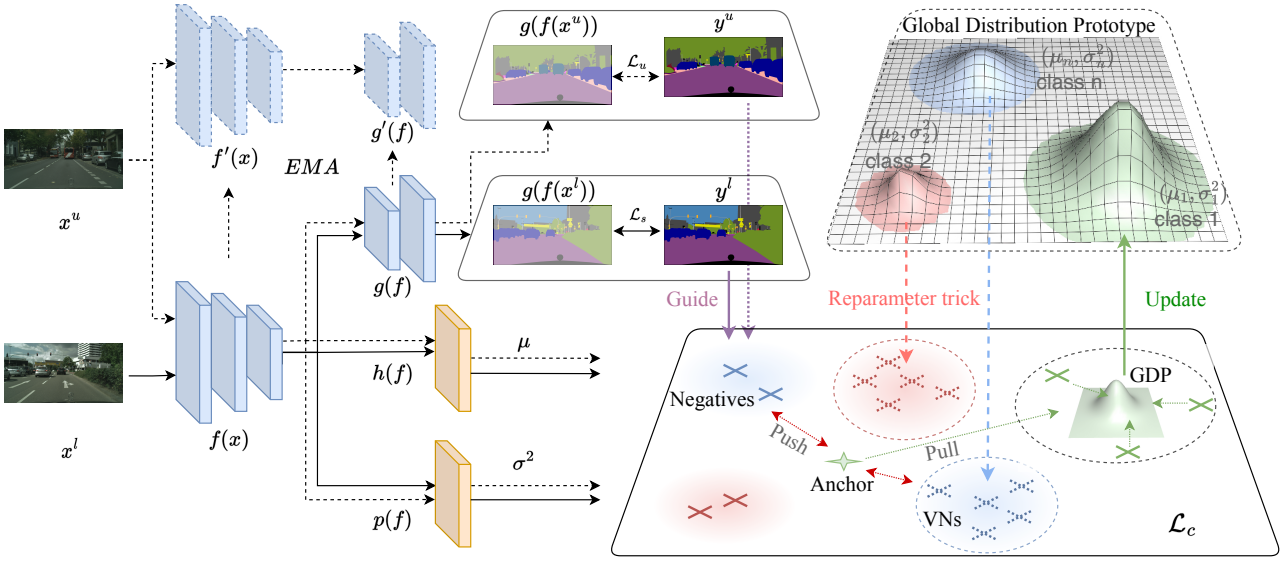

Fig. 3 Overall framework of PRCL. The training pipeline contains two input streams: labeled images (black arrows) and unlabeled images (black dash arrows). In the pixel space, the model is guided by the combination of ground-truth y^l and original pseudo-labels y^u. In the latent space, the model maps the pixels into probabilistic representations $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ via two heads: $h(\cdot)$ and $p(\cdot)$. And the GDP is stored in a prototype-level dictionary and is updated with the local prototype. We generate the virtual negatives (VN, dashed cross) from GDP for contrastive loss $\mathcal{L}_c$.

representations. Conversely, the $\boldsymbol{\mu}$ can also be optimized via the $\boldsymbol{\sigma}^2$. This mutual interaction between $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$ aligns with our intuitive understanding of representation learning.

3.3 Global Distribution Prototype

In conventional methods, the prototype $\boldsymbol{\rho}_c$ of class c is typically obtained by aggregating the semantic information from all representations that belong to class c in the current iteration. With probabilistic representations, this process can be formulated as:

$$\begin{aligned} &\boldsymbol{\rho}_c \sim \mathcal{N}(\hat{\boldsymbol{\mu}}_c, \hat{\boldsymbol{\sigma}}_c^2 \boldsymbol{I}), \\ &\frac{1}{\hat{\boldsymbol{\sigma}}_c^2} = \sum_{\boldsymbol{z}_{ci} \in \mathcal{Z}_c} \frac{1}{\boldsymbol{\sigma}_{ci}^2}, \\ &\hat{\boldsymbol{\mu}}_c = \sum_{\boldsymbol{z}_{ci} \in \mathcal{Z}_c} \frac{\hat{\boldsymbol{\sigma}}_c^2}{\boldsymbol{\sigma}_{ci}^2} \boldsymbol{\mu}_{ci}, \end{aligned} \tag{4}$$

where $\mathcal{Z}_c$ represents the set of the representations belong to class c in current iteration $\mathcal{Z}_c = \{\boldsymbol{z}_{c0}, \boldsymbol{z}_{c1}, ..., \boldsymbol{z}_{ci}\}$ and $\boldsymbol{z}_{ci} \overset{i.i.d}{\sim} \mathcal{N}(\boldsymbol{\mu}_{ci}, \boldsymbol{\sigma}_{ci}^2 \boldsymbol{I})$. Even though probabilistic representations offer the advantage of accommodating inaccurate pseudo-labels by incorporating probability into the representation, certain significant inaccuracies still affect the precision of prototypes, resulting in prototype shift. Additionally, the inherent intra-class variance introduces variations in the representations of identical classes across adjacent iterations, further causing the prototype shift. To address these challenges, we introduce an efficient strategy that sequentially aggregates representations across iterations from a global perspective. Specifically, we define the prototype calculated in the current iteration as *local* prototype and extend the local prototype to the Global Distribution Prototype (GDP). We introduce a variable t to represent the t^{th} iteration during training and use $\boldsymbol{\rho}_l(t)$ to represent the *local* prototype. For clarity, we omit class c here. We represent GDP as $\boldsymbol{\rho}_g(t)$, which can be formulated as:

$$p(\boldsymbol{\rho}_g(t) | \mathcal{Z}_g(t)) = \mathcal{N}(\hat{\boldsymbol{\mu}}_g(t), \hat{\boldsymbol{\sigma}}_g^2(t) \boldsymbol{I}), \tag{5}$$

where $\mathcal{Z}_g(t)$ represents the set of all identical class representations observed, and given $\mathcal{Z}_l(t)$ represents the set of identical class representations in the t^{th} iteration, $\mathcal{Z}_g(t) = \mathcal{Z}_l(0) \cup \mathcal{Z}_l(1) \ldots \cup \mathcal{Z}_l(t)$. Since each representation $\boldsymbol{z}_i \overset{i.i.d}{\sim} \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2 \boldsymbol{I})$, we have

$$\begin{aligned} \frac{1}{\hat{\boldsymbol{\sigma}}_g^2(t)} &= \sum_{\boldsymbol{z}_i \in \mathcal{Z}_g(t)} \frac{1}{\boldsymbol{\sigma}_i^2} = \sum_{\boldsymbol{z}_i \in \mathcal{Z}_g(t-1)} \frac{1}{\boldsymbol{\sigma}_i^2} + \sum_{\boldsymbol{z}_i \in \mathcal{Z}_l(t)} \frac{1}{\boldsymbol{\sigma}_i^2} \\ &= \frac{1}{\hat{\boldsymbol{\sigma}}_g^2(t-1)} + \frac{1}{\hat{\boldsymbol{\sigma}}_l^2(t)}, \end{aligned} \tag{6}$$

and

$$\begin{aligned} \hat{\boldsymbol{\mu}}_g(t) &= \sum_{\boldsymbol{z}_i \in \mathcal{Z}_g(t)} \frac{\hat{\boldsymbol{\sigma}}_g^2(t)}{\boldsymbol{\sigma}_i^2} \boldsymbol{\mu}_i \\ &= \hat{\boldsymbol{\sigma}}_g^2(t) \left(\sum_{\boldsymbol{z}_i \in \mathcal{Z}_g(t-1)} \frac{\boldsymbol{\mu}_i}{\boldsymbol{\sigma}_i^2} + \sum_{\boldsymbol{z}_i \in \mathcal{Z}_l(t)} \frac{\boldsymbol{\mu}_i}{\boldsymbol{\sigma}_i^2} \right) \\ &= \hat{\boldsymbol{\sigma}}_g^2(t) \left(\frac{1}{\hat{\boldsymbol{\sigma}}_g^2(t-1)} \sum_{\boldsymbol{z}_i \in \mathcal{Z}_g(t-1)} \frac{\hat{\boldsymbol{\sigma}}_g^2(t-1) \boldsymbol{\mu}_i}{\boldsymbol{\sigma}_i^2} + \frac{1}{\hat{\boldsymbol{\sigma}}_l^2(t)} \sum_{\boldsymbol{z}_i \in \mathcal{Z}_l(t)} \frac{\hat{\boldsymbol{\sigma}}_l^2(t) \boldsymbol{\mu}_i}{\boldsymbol{\sigma}_i^2} \right) \\ &= \hat{\boldsymbol{\sigma}}_g^2(t) \left(\frac{\hat{\boldsymbol{\mu}}_g(t-1)}{\hat{\boldsymbol{\sigma}}_g^2(t-1)} + \frac{\hat{\boldsymbol{\mu}}_l(t)}{\hat{\boldsymbol{\sigma}}_l^2(t)} \right). \end{aligned} \tag{7}$$

This means that we can get the current GDP from the last GDP $\boldsymbol{\rho}_g(t-1)$ and the current local prototype $\boldsymbol{\rho}_l(t)$:

$$p(\boldsymbol{\rho}_g|\mathcal{Z}_g(t)) = p(\boldsymbol{\rho}_g|\boldsymbol{\rho}_g(t-1), \boldsymbol{\rho}_l(t)), \tag{8}$$

precisely, the GDP can be updated as follow:

$$\begin{aligned} &\boldsymbol{\rho}_g(t) \sim \mathcal{N}(\hat{\boldsymbol{\mu}}_g(t), \hat{\boldsymbol{\sigma}}_g^2(t)\boldsymbol{I}), \\ &\frac{1}{\hat{\boldsymbol{\sigma}}_g^2(t)} = \frac{1}{\hat{\boldsymbol{\sigma}}_g^2(t-1)} + \frac{1}{\hat{\boldsymbol{\sigma}}_l^2(t)}, \\ &\hat{\boldsymbol{\mu}}_g(t) = \hat{\boldsymbol{\sigma}}_g^2(t)\left(\frac{\hat{\boldsymbol{\mu}}_g(t-1)}{\hat{\boldsymbol{\sigma}}_g^2(t-1)} + \frac{\hat{\boldsymbol{\mu}}_l(t)}{\hat{\boldsymbol{\sigma}}_l^2(t)}\right). \end{aligned} \tag{9}$$

The GDP has the following properties:

1. GDP considers all historical representations, which contribute to GDP according to their probability $\boldsymbol{\sigma}^2$.
2. In prototype calculation, GDP $\boldsymbol{\rho}_g(t-1)$ is equivalent to all historical representations $\mathcal{Z}_g(t-1)$ in previous iterations.

The first property of GDP implies that it possesses robustness against instantaneous noisy pseudo-labels if $\boldsymbol{\sigma}^2$ is estimated effectively. And the second property suggests that utilizing GDP to bridge iterations incurs minimal memory cost since current GDP $\boldsymbol{\rho}_g(t)$ can be derived from the last GDP $\boldsymbol{\rho}_g(t-1)$ and the current local prototype $\boldsymbol{\rho}_l(t)$. Therefore, it is unnecessary to store all previous representations in memory; only the last GDP $\mathcal{Z}_g(t-1)$ needs to be stored.

3.4 Virtual Negatives

In order to compensate for the fragmentary distribution of negatives, instead of using the memory bank strategy, we propose an *efficient* strategy, which takes advantage of the distribution of GDP. We generate Virtual Negatives (VN) from GDP $\boldsymbol{\rho}_{(c)g}(t) \sim \mathcal{N}(\hat{\boldsymbol{\mu}}_{(c)g}(t), \hat{\boldsymbol{\sigma}}^2_{(c)g}(t)\boldsymbol{I})$ corresponding to class c via a modified reparameter trick [29]:

$$\boldsymbol{z}_c^{VN} = \hat{\boldsymbol{\mu}}_{(c)g}(t) + \beta\boldsymbol{\epsilon}^\top \boldsymbol{I}\hat{\boldsymbol{\sigma}}^2_{(c)g}(t), \tag{10}$$

where $\boldsymbol{\epsilon} = (\epsilon^{(1)}, \ldots, \epsilon^{(d)})$, $\epsilon^{(1)}, \ldots, \epsilon^{(d)} \sim \mathcal{N}(0,1)$ and β is a hyper-parameter we define to balance the compactness and the diversity of VNs, named *virtual radius*. The reparameter trick essentially samples some representations centred at the $\mu^{(d)}$, taking the Gaussian function $\mathcal{N}(\mu^{(d)}, \sigma^{2(d)})$ as the probability density function, and taking β as radius for each dimension d. VNs inherit the global features from GDP which covers the entire iterations. Moreover, VNs exhibit improved compactness compared to real representations, as they are not affected by intra-class variance. Additionally, VNs offer enhanced dispersion compared to GDP. In the current iteration, the limited size of the mini-batch restricts the coverage of real negative representations to only a subset of classes in the dataset. Consequently, this results in a fragmented negative distribution. However, our VNs have the advantage of encompassing representations from all classes in the dataset. As a result, they compensate for the fragmented distribution of negative representations and incorporate a broader range of global features.

Discussion about Memory Bank. The memory bank strategy, as employed in various studies [21,1,54,56], has been a conventional solution for bridging iterations and compensating for the fragmentary distribution of negatives. Originally designed to extend the coverage of images with a limited mini-batch size, the memory bank relies on an elaborate sampling strategy to enqueue and dequeue representations, which can be computationally expensive (31% slower processing) and memory-intensive (2.63 GB). This is because, compared with instance-level representations, pixel-wise representations are dense, and each pixel (or region) in the image is mapped to a corresponding representation. To overcome these limitations, we propose a new strategy that leverages GDP and VNs to complement the fragmentary distribution of negatives without the need for an intricate sampling strategy to enqueue and dequeue representations. Particularly, our method aggregates global information into GDP using only **42KB (v.s. 2.63 GB)** of memory and generates VNs with minimal computational cost (increase training time by **0.03 v.s. 0.47 GPU days)**. By using this approach, we can capture a larger amount of image information at scale without the significant memory and computational overhead associated with the memory bank strategy. The experimental proof can be found in Sec. 5.3

3.5 Training Objective

Cooperated with the conventional teacher-student framework and our introduced components, the total training object is composed of a supervised loss $\mathcal{L}_s$, an unsupervised loss $\mathcal{L}_u$, and a contrastive loss $\mathcal{L}_c$ as follows:

$$\mathcal{L} = \mathcal{L}_s + \mathcal{L}_u + \lambda_c(t)\mathcal{L}_c, \tag{11}$$

where $\lambda_c(t)$ is used to tune the contribution of contrastive loss and formulated as:

$$\lambda_c(t) = \lambda_{c0} \cdot exp\left(\alpha \cdot (\frac{t}{T_{total}})^2\right), \tag{12}$$

where λ_{c0} denotes the initial scaling parameter, α denotes a weight decay coefficient, t denotes the current t^{th} epoch and T_{total} denotes the total epochs.

$\mathcal{L}_s$ is constructed by standard Cross-Entropy (CE) loss ℓ_{ce} and formulated as:

$$\mathcal{L}_s = \frac{1}{|\mathcal{B}_l|} \sum_{(\boldsymbol{x}_i^l, \boldsymbol{y}_i^l) \in \mathcal{B}_l} \ell_{ce}(g(f(\boldsymbol{x}_i^l), \boldsymbol{y}_i^l)), \tag{13}$$

where $\mathcal{B}_l$ represent the batch of labeled images.

For $\mathcal{L}_u$, we first set a threshold δ_u to count the number $\hat{N}$ of training pixels whose corresponding confidence

p_i^u is higher than δ_u in the training set $\mathcal{B}_u$. With $\hat{N}$ and the total number N of the pixels in $\mathcal{B}_u$, the $\mathcal{L}_u$ is constructed by the weighted CE loss, formulated as:

$$\mathcal{L}_u = \frac{1}{|\mathcal{B}_u|} \sum_{\boldsymbol{x}_i^u \in \mathcal{B}_u} \omega \ell_{ce}(g(f(\boldsymbol{x}_i^u), \boldsymbol{y}_i^u)), \tag{14}$$

where $\boldsymbol{y}_i^u$ is the pseudo-labels from teacher model, and ω is the loss weight, formulated by $\omega = \frac{\hat{N}}{N}$.

While calculating contrastive loss, since the limited GPU memory, we sample valuable representations for the contrast, following prior works [34]. We adopt some sampling strategies according to confidence p_i and introduce strong threshold δ_s and weak threshold δ_w in these strategies. Our sampling strategies are as follows: **1) Valid representations sampling strategy**: We set δ_w for sampling valid representations whose p_i is higher than δ_w. Only valid representations will be considered in the contrast. **2) Anchors sampling strategy**: We set δ_s for sampling anchors whose corresponding p_i is lower than δ_s. **3) Negatives sampling strategy**: We non-uniformly sample negatives in different classes based on the similarity (i.e., MLS [49]) between GDPs of negative classes and the GDP of the current anchor class.

We apply InfoNCE [41] as our contrastive loss and introduce the PR, GDP, and VN (describe in Sec. 3.2, Sec. 3.3, and Sec. 3.4, respectively) to it, formulated as :

$$\mathcal{L}_c = -\frac{1}{|C| \times |\mathcal{Z}_c|} \sum_{c \in C} \sum_{\boldsymbol{z}_{ci} \in \mathcal{Z}_c} \log[\frac{e^{s(\boldsymbol{z}_{ci}, \boldsymbol{\rho}_{(c)g}(t))/\tau}}{e^{s(\boldsymbol{z}_{ci}, \boldsymbol{\rho}_{(c)g}(t))/\tau} + \sum_{\tilde{c} \in \tilde{C}_l} \sum_{\boldsymbol{z}_{\tilde{c}j} \in \mathcal{Z}_{\tilde{c}}} e^{s(\boldsymbol{z}_{ci}, \boldsymbol{z}_{\tilde{c}j})/\tau} + \sum_{\tilde{c} \in \tilde{C}_g} \sum_{\boldsymbol{z}_{\tilde{c}j}^{VN} \in \mathcal{Z}_{\tilde{c}}^{VN}} e^{s(\boldsymbol{z}_{ci}, \boldsymbol{z}_{\tilde{c}j}^{VN})/\tau}}], \tag{15}$$

where C represents the set of all anchor classes in the current iteration, $\mathcal{Z}_c$ represents the set of anchor representations $\boldsymbol{z}_{ci}$ belonging to class c, $\tilde{C}_l$ represents the negative classes in the current iteration, $\tilde{C}_g$ represents negative classes in the entire dataset, $\mathcal{Z}_{\tilde{c}}$ represents the set of real negative representations $\boldsymbol{z}_{\tilde{c}j}$ in the current iteration, $\mathcal{Z}_{\tilde{c}}^{VN}$ represents the set of virtual negatives $\boldsymbol{z}_{\tilde{c}j}^{VN}$ belonging to class c, $\boldsymbol{\rho}_{(\tilde{c})g}(t)$ represents the current GDP belonging to class c, τ represents the temperature parameter, s is MLS. And we pad zero to the probability of VNs empirically, due to the probability missing in the modified reparameter trick.

For training our probabilistic head, we adopt a strategy called soft freeze, following [1]. Specifically, we separate the training of the probability head from the training of the backbone and segmentation head and endow the probability head with a small learning rate. The reason for adopting this strategy is to guarantee the stability of the training process. At the beginning of training, the outputs of probability head climb dramatically since the representations are unreasonable and meaningless at that time. Therefore, it is essential to endow the probability head with a much smaller learning rate than the backbone so that its training process is able to keep pace with others and the different modules in the network can interact with each other. Our framework is in Fig. 3. Meanwhile, the process of training is demonstrated in Algorithm 1.

4 Experiments

4.1 Setup

Datasets. We conduct experiments on PASCAL VOC 2012 dataset [9] and Cityscapes dataset [6] to validate the effectiveness of our framework. The PASCAL VOC 2012 contains 1464 well-annotated images in `train` set and 1449 images in `val` set originally. We include 9118 images from SBD [16] as additional training images. Since the SBD dataset is coarsely annotated, we use both *classic* VOC `train` set (1464 candidate labeled images) and *blender* VOC `train` set (10582 candidate labeled images), following [56]. Cityscapes contains 2975 images in `train` set and 500 images in `val` set.

Network structure. We choose Deeplabv3+ [3] as our structure with ResNet [18] pre-trained on ImageNet [8] as the backbone. The segmentation and representation heads are composed of `Conv-BN-ReLU-Conv`. The probability head is composed of `Linear-BN-ReLU-Linear-BN`.

Implementation details. For both two datasets, we use stochastic gradient descent (SGD) optimizer. For training on PASCAL VOC 2012, the initial learning rate is 6.4×10^{-3} for the backbone, segmentation head, and representation head while 5×10^{-5} for the probability head. For training on Cityscapes, the initial learning rate is 6.4×10^{-3} for the backbone, segmentation head, and representation head while 5×10^{-5} for the probability head. We use the poly scheduling to decay the learning rate during training: $lr = lr_{base}(1 - \frac{iter}{total_{iter}})^{0.9}$. For PASCAL VOC 2012, we set the image crop size as 512 × 512, and the batch size as 16. For Cityscapes, we set the image crop size as 768 × 768, and the batch size as 16. The models are trained for 80,000 and 160,000 iterations on PASCAL VOC 2012 and Cityscapes when compared with SOTAs, respectively. Exceptionally, for the ablation study, we train models for 40,000 iterations on PASCAL VOC 2012 and the batch size is set to 8.

Evaluation metric. We use mean intersection-over-union (mIoU) as our metric for evaluation. Meanwhile, following [5], we employ the slide window strategy to evaluate the performance of the Cityscapes dataset.

4.2 Comparing with Existing Methods

In this subsection, we conduct a comprehensive evaluation of our proposed method by comparing it against several baselines and state-of-the-art (SOTA) approaches. Specifically, we reproduce Mean Teacher (MT) [51] and ClassMix [40] on *classic* VOC `train` set and Cityscapes `train` set. It is worth noting that it is not hard to apply our components to most contrastive-based S4 works [34,

Algorithm 1 Pseudo-code of the training process in a Pytorch-like style

Network: Student encoder: f, student segmentation head: g, teacher encoder: f', teacher segmentation head: g', representation head: h, probability head: p
Input: Mini-batch B consists of (X^l, Y^l) and (X^u), last GDP $\boldsymbol{\rho}_g(t-1)$
Notation: Anchor class c, remaining classes in current iteration $\tilde{C}_l$, remaining classes in dataset $\tilde{C}_g$

for *epoch* in range($total\_epoch$) **do**
  $P^l = g(f(X^l))$ # Predict on labeled images
  $\mathcal{L}_s = ce\_loss(P^l, Y^l)$ # Calculate supervised loss $\mathcal{L}_s$
  $Y^u = max\_op(g'(f'(X^u)))$ # Generate pseudo-labels via max operation in Eq. 1
  $P^u = g(f(X^u))$ # Predict on unlabeled images
  $\mathcal{L}_u = ce\_loss(P^u, Y^u)$ # Calculate unsupervised loss $\mathcal{L}_u$
  $(X, Y) \leftarrow combine((X^l, Y^l), (X^u, Y^u))$ # Combine labeled and unlabeled training set
  for $(X, Y) \in B$ **do**
    $\mathcal{L}_c = 0$ # Initialize $\mathcal{L}_c$
    $\mu = h(f(X))$ # Calculate $\boldsymbol{\mu}$
    $\sigma^2 = p(f(X))$ # Calculate $\boldsymbol{\sigma}^2$
    $Z \leftarrow (\mu, \sigma^2)$ # Representations consisting of $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$
    $Z_{val}, Y_{val}, C_{val} \leftarrow mask(Z, Y)$ # Mask with δ_w according to sampling strategy
    for $c \in C_{val}$ **do**
      $\boldsymbol{\rho}_l^c(t) \leftarrow calculate\_prt(Z_{val}^c)$ # Calculate local prototype with Eq. 4
      $\boldsymbol{\rho}_g^c(t) \leftarrow update\_prt(\boldsymbol{\rho}_g^c(t-1), \boldsymbol{\rho}_l^c(t))$ # Update GDP with Eq. 9
    end for
    for $c \in C_{val}$ **do**
      $\mathcal{L} = 0$ # Initialize loss $\mathcal{L}$ in current class
      $Z_a^c \leftarrow sample\_a(Z_{val}^c, Y_{val}^c)$ # Sample anchor representations
      $neg\_dist \leftarrow [MLS(\boldsymbol{\rho}_g^c(t), \boldsymbol{\rho}_g^{\tilde{c}_l}(t))\ for\ \tilde{c}_l\ in\ \tilde{C}_l]$ # Calculate negative sampling distribution
      $Z_n \leftarrow sample\_n(Z_{val}^{\tilde{c}_l}, neg\_dist)$ # Sample real negative representations
      $Z_{VN} \leftarrow [generate\_VN(\boldsymbol{\rho}_g^{\tilde{c}_g}(t))\ for\ \tilde{c}_g\ in\ \tilde{C}_g]$ # Generate virtual negatives with Eq. 10
      $\mathcal{L} = contrast\_loss(Z_a^c, Z_n, Z_{VN}, \boldsymbol{\rho}_g^c(t))$ # Calculate $\mathcal{L}_c$ with Eq. 15
      $\mathcal{L}_c = \mathcal{L}_c + \mathcal{L}$
    end for
  end for
  $\mathcal{L}_{total} = \mathcal{L}_s + \mathcal{L}_u + \lambda \mathcal{L}_c$ # Calculate the total loss $\mathcal{L}_{total}$, λ is the current value of $\lambda_c(t)$ in Eq. 11
  optimizer.zero_grad()
  $\mathcal{L}_{total}$.backward()
  optimizer.step()
end for

56, 1, 54, 55]. For simplicity, we use a contrastive-based teacher-student framework as our **Baseline**. The baseline framework has the same architecture as our framework but without PR, GDP, prototype update strategy, and VN. In addition, we design **Baseline+** which adds the prototype update strategy. The prototype in **Baseline+** is updated by EMA, which is formulated by

$$\boldsymbol{\rho}_g^{EMA}(t) = \alpha \boldsymbol{\rho}_g^{EMA}(t-1) + (1-\alpha)\boldsymbol{\rho}_l(t), \tag{16}$$

where $\boldsymbol{\rho}_g^{EMA}(t)$ denotes the updated prototype, $\boldsymbol{\rho}_g^{EMA}(t-1)$ denotes the previous prototype, $\boldsymbol{\rho}_l^{EMA}(t)$ denotes the local prototype from the current mini-match, and α is hyper-parameter to control the update speed, set as 0.99.

Meanwhile, we compare our method on *blender* VOC `train` set with following recent SOTA S4 methods: CCT [42], CPS [5], U^{2}PL [56], ST++ [64], PSMT [35], ELN [30]. Besides, we compare our method on Cityscapes `train` set with the following methods: CCT [42], CPS [5], U^{2}PL [56], and PSMT [35]. For a fair comparison, following [56, 5], we use a modified ResNet-101 with the stem block to compare with SOTAs while using the original ResNet-101 to compare with baselines in both two datasets.

Results on PASCAL VOC 2012. Tab. 1 shows the results on *classic* VOC `train` set, comparing with baselines. Our method consistently outperforms baselines at all label rates. Tab. 2 compares our method with the SOTA methods, our method shows the competitive performance in a wide range of label rates. Since our contribution lies in contrastive learning derived from self-training, the performance is more advantageous in the case of the few labels.

Results on Cityscapes. Tab. 3 presents the comparison results on the Cityscapes dataset. Our proposed method exhibits a slight but consistent improvement in performance compared to all baseline methods and SOTA approaches. This improvement is observed across various label rates, indicating the robustness and effectiveness of our approach in different scenarios.

4.3 Other Results

The qualitative results of different methods on the PASCAL VOC 2012 are shown in Fig. 4. Our PRCL frame-

Table 1 Results on *classic* VOC `train` set using original ResNet-101. All approaches are reproduced. Labeled data is from the original VOC `train` set.

PASCAL VOC 2012 (*Classic*)					
Method	92	183	366	732	1464
Supervised	52.38	55.32	65.92	71.59	72.90
MT [51]	46.84	61.35	66.86	71.93	74.00
ClassMix [40]	63.83	67.20	71.23	73.98	76.91
Baseline	66.91	69.73	72.01	74.99	76.85
Baseline+	68.27	70.32	73.97	75.61	77.21
PRCL	**70.23**	**72.20**	**75.17**	**76.24**	**78.29**

Table 2 Results on *blender* VOC `train` set using a modified ResNet-101. All the results from the recent papers [56,64,30, 35]. Labeled data is from the augmented VOC `train` set.

PASCAL VOC 2012 (*Blender*)					
Method	Publication	662	1323	2646	5291
CCT [42]	CVPR 20	71.86	73.68	76.51	77.40
CPS [5]	CVPR 21	74.48	76.44	77.68	78.64
U^{2}PL [56]	CVPR 22	77.21	79.01	79.30	**80.50**
ST++ [64]	CVPR 22	74.70	77.90	77.90	-
PSMT [35]	CVPR 22	75.50	78.20	78.72	79.76
ELN [30]	CVPR 22	-	75.10	76.58	-
PRCL	-	**77.87**	**79.09**	**79.85**	80.11

Table 3 Results on Cityscapes. The model is trained on the Cityscapes `train` set, which consists of 2975 samples in total, and tested on the Cityscapes `val` set. And all the results from the recent papers [56,35]. † means we reproduce the approach.

Cityscapes					
Method	Publication	186	372	744	1488
Supervised†	-	62.04	65.71	69.48	70.06
MT† [51]	NeurIPS 17	67.06	68.43	70.05	70.64
ClassMix† [40]	WACV 21	67.98	69.58	72.21	72.82
CCT [42]	CVPR 20	69.32	74.12	75.99	77.40
CPS [5]	CVPR 21	69.78	74.31	74.58	76.82
U^{2}PL [56]	CVPR 22	70.30	74.37	76.47	79.05
PSMT [35]	CVPR 22	-	76.89	77.60	79.09
PRCL	-	**73.38**	**77.08**	**77.89**	**79.95**

work demonstrates superior performance compared to the other methods, benefiting from its enhanced robustness. While our baseline shows some improvement over the self-training framework with MT and Classmix, it still exhibits limitations in handling ambiguous regions, such as the boundaries between different classes. In contrast, our PRCL framework performs better in these ambiguous regions, visually showcasing the effectiveness of our approach. Fig. 5 shows the qualitative results on Cityscapes, further validating the superiority of our framework.

Additionally, to gain a more intuitive understanding of the advantages offered by our framework, we provide a quantitative comparison in representation space between both the baseline and our framework and two t-SNE plots to visualize the distribution of representations in the latent space for both the baseline and our framework. Tab. 4 shows the results of the quantitative comparison. We use the Silhouette score [46] (Silhouette) and Davies–Bouldin Index [7] (DBI) as metrics. These two measurements are both able to show the cohesion of a representation and its cluster and the separation of a representation and other clusters. As for the Silhouette Score, a larger value indicates better performance. In contrast, the smaller value of the Davies-Bouldin Index stands for better performance. The results show that our framework performs better than our baseline in representation space in two different datasets. Fig. 6 (a) and (b) show that our method results in a better representation distribution compared to the baseline on PASCAL VOC 2012. We can observe that our representation is more compact than the baseline, especially the region highlighted in red boxes. This is because our framework is more robust to inaccurate pseudo-labels and provides a consistent direction for representation aggregation. Fig. 6 (c) and (d) show the results on Cityscapes, which further prove the effectiveness of our framework.

Table 4 Performance of two frameworks in representation space.

Dataset	Pascal VOC 2012		Cityscapes	
Metrics	Silhouette	DBI	Silhouette	DBI
Baseline	0.3188	1.1221	0.2708	1.9472
PRCL	0.3836	1.0334	0.3014	1.7667

5 Ablative Study

The main contribution of our work lies in **1)** probabilistic representation, **2)** global distribution prototype, and **3)** virtual negatives. We conduct experiments to further explore the effectiveness and rationality of our components. We choose Deeplabv3+ with ResNet-101 pre-trained on ImageNet as our backbone and PASCAL VOC 2012 as our dataset. The baseline and baseline+ framework and other settings are the same as those in Sec. 4.

5.1 Effect of Probabilistic Representation

Behaviors of the probability. To visualize the relationship between probability and inaccurate pseudo-labels, we provide visualizations of the model's predictions along with their corresponding probabilities. In addition, we apply ℓ_1 normalization to $\boldsymbol{\sigma}^2$ for visualization purposes. In Fig. 7, columns from left to right represent input image, ground-truth, pseudo-label, and probability map, respectively. For the probability map, the red color represents the large $\boldsymbol{\sigma}^2$ (indicating low probability). In the visualizations, we use green boxes to highlight the mismatches caused by inaccurate pseudo-labels, such as instances mistakenly labeled as a person or a bottle. On the other hand, red boxes are used to mark fuzzy pixels, such as the furry edge of a bird. These cases are specifically identified by the $\boldsymbol{\sigma}^2$ values and are observed to

Fig. 4 Visualisation on PASCAL VOC 2012. All models are trained with 92 labeled images. The differences are highlighted in yellow boxes.

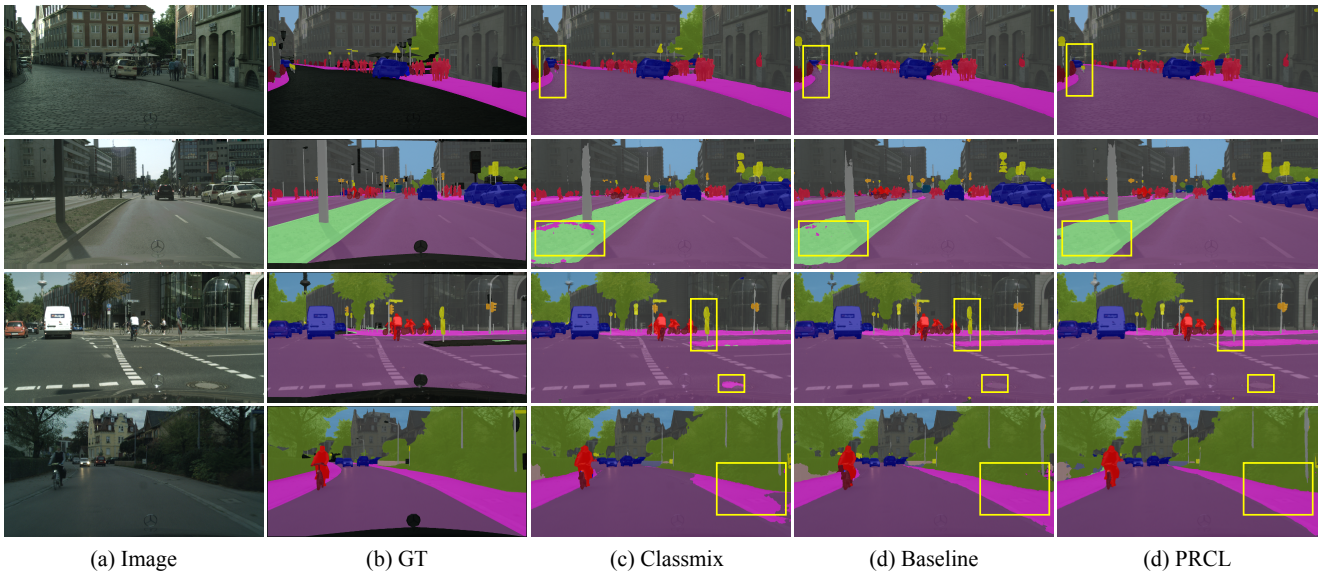

Fig. 5 Visualisation on Cityscapes. All models are trained with 186 labeled images. The differences are highlighted in yellow boxes.

have a relatively low contribution during the contrastive training process.

Results of using probabilistic representation. To explore the impact of probabilistic representation, we conduct the experiments as follows: baseline **1)** without probabilistic representation (w/o PR), and **2)** with probabilistic representation (w/ PR). Tab. 5 shows the effectiveness of probabilistic representation across various label rates. This can be attributed to introducing probability in our representation, which allows us to reduce the negative effect of inaccurate pseudo-labels during contrastive learning. Consequently, PR endows our framework the greater robustness compared to conventional contrastive-based teacher-student frameworks with deterministic representations. Meanwhile, our approach can be easily applied to other contrastive-based S4 frame-

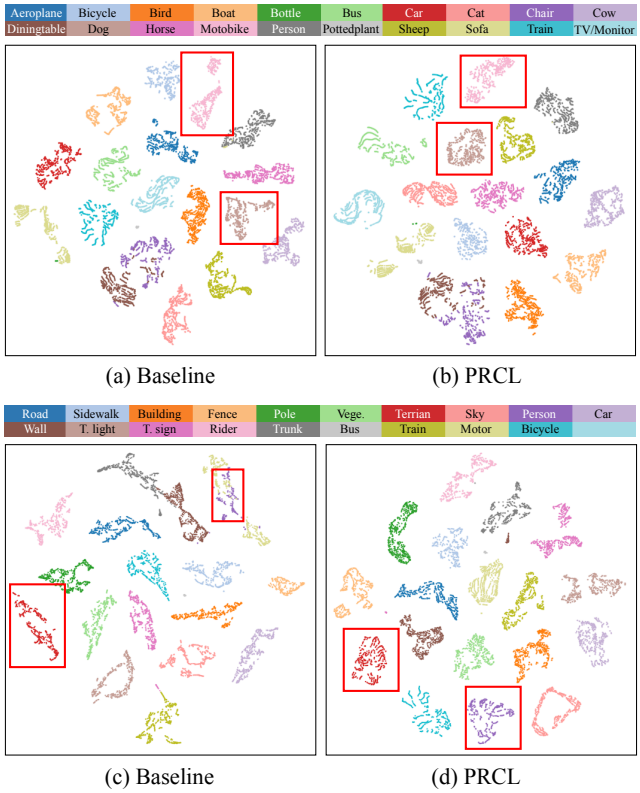

Fig. 6 Representation visualizations. The differences are highlighted in red boxes.

works by substituting probabilistic representations for deterministic representations.

Table 5 Results on the effect of the probabilistic representation.

PASCAL VOC 2012				
Label rates	92	183	366	732
w/o PR	66.91	69.73	72.01	74.99
w/ PR	68.49	71.79	74.36	76.00

5.2 Effect of Global Distribution Prototype

To investigate the influence of the prototype update strategy, we conduct experiments **a)** baseline without update strategy, **b)** baseline with EMA update strategy (baseline+), **c)** baseline without update strategy but with probabilistic representation, and **d)** baseline with update strategy and probabilistic representation.

Behavior of prototype. We visualize the changes in prototypes during the training process in the conducted experiments through t-SNE [36]. As shown in Fig. 8 (a) and Fig. 8 (c), without any strategy, there is a noticeable prototype shift observed between prototypes in two adjacent iterations. This significant shift in distance leads to inconsistent directions for the aggregation of anchor representations, thereby hindering the effective aggregation of representations. Although the EMA strategy also updates prototypes based on previous ones, its behavior exhibits instability and discontinuity throughout the training process, as depicted in Fig. 8 (b). Due to its limited global property, the EMA strategy is sensitive to instant noisy pseudo-labels, also leading to prototype shifts resulting from incorrect representation assignments. In contrast, our GDP strategy demonstrates greater stability and robustness against instant noisy pseudo-labels. It benefits from its fully global property, which encompasses all historical information. As shown in Fig. 8 (d), the GDP strategy maintains stable and continuous behavior throughout the training process.

Results of different update strategies. Tab. 6 shows the impact of two different update strategies on two types of representations: EMA (Eq. 16) update strategy on deterministic representations (Vanilla), and our GDP (Eq. 9) update strategy on probabilistic representation (PR) on PASCAL VOC 2012 with a wide range of label rate. Both update strategies contribute to performance improvement by ensuring consistency in the representation space. However, our GDP approach yields the best outcomes, demonstrating its superiority over the conventional EMA approach.

Table 6 Impact of Update Strategy (US)

Vanilla Rep.	92	183	366	732
w/o US	66.91	69.73	72.01	74.99
w/ US	68.27	70.32	73.97	75.61
PR	92	183	366	732
w/o US	68.49	71.79	74.36	76.00
w/ US	69.52	72.20	75.17	76.24

5.3 Effect of Virtual Negatives

Visualization of the virtual negatives. To investigate how virtual negatives work, we utilize t-SNE to visualize them. Fig. 9 illustrates the results. The VNs are observed to form clusters around the global distribution prototype, while the real representations also exhibit clustering around the global distribution prototype, but with notable displacements. This observation indicates that virtual negatives are more effective in capturing global features compared to real negatives, which primarily represent local-level features. Furthermore, virtual negatives occupy positions that should have been filled by real negatives but were lost due to the limited capacity of the mini-batch We visualize two sets of VNs with different virtual radius β, enabling a balance between the compactness and diversity of VNs, as Fig. 9 (b) details.

Results of using virtual negatives. Our VN introduces two critical hyper-parameters: its number and virtual ra-

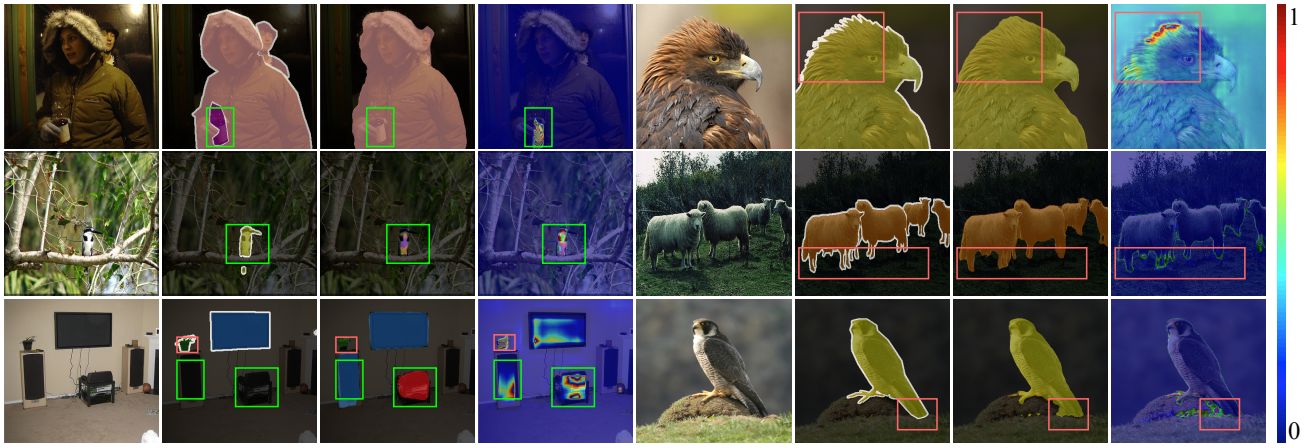

Fig. 7 Visualization of probability behavior.

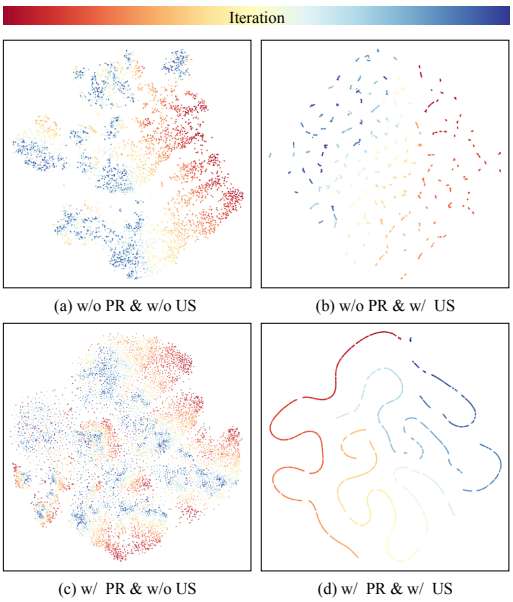

Fig. 8 Visualization of prototype behaviour. w/o PR and w/ PR mean without probabilistic representation and with probabilistic representation, respectively. Similarly, w/o US and w/ US mean without prototype update strategy and with prototype update strategy, respectively.

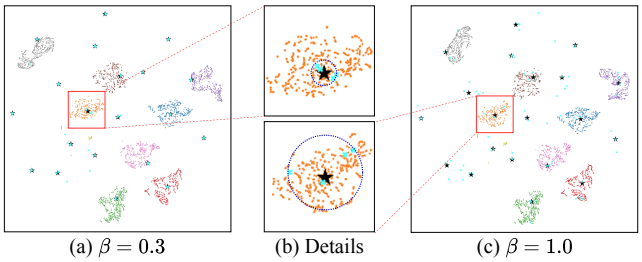

Fig. 9 Visualization of virtual negatives.

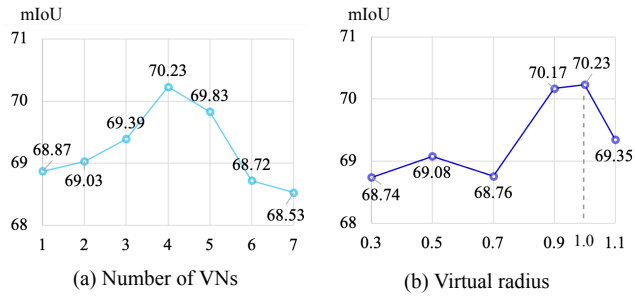

Fig. 10 Effect on the number of virtual negatives and the virtual radius.

dius β. Both of these parameters have a significant impact on the properties of the negative distribution and consequently affect the overall performance. To investigate the influence of the number of VNs on performance, we conducted several experiments with varying numbers of VNs. Fig. 10 (a) demonstrates that using 4 VNs for each class performs best in the current setting. We argue that an excessive number of VNs may excessively focus on the global representation while disregarding the local representation. This imbalance may hinder local contrasts, thereby impeding the update of global distribution prototypes based on local prototypes. On the other hand, too few VNs may not fully leverage the benefits of global representations, resulting in a diminished impact on the learning process. To explore the impact of the virtual radius on performance, we conducted multiple experiments with different values of β. As shown in Fig. 10 (b), our VN achieves optimal performance with a virtual radius of $\beta = 1.0$. The β controls the diversity and noise in generated VNs, and there also will be a dilemma between diversity and noise in VNs. Specifically, too large β will introduce too much noise in VNs, even though it endows VNs with more diversity. In contrast, too small β loses diversity when reducing noise in VNs. The number of VNs and the virtual radius both exert non-linear effects on the properties of the virtual negatives, which will influence the overall performance of the framework.

Overall, carefully selecting the number of VNs and the virtual radius is crucial to strike a balance between local and global representations, ultimately enhancing the performance of the model.

Comparisons to the memory bank strategy. To compare different compensation strategies, we conduct experiments based on a single NVIDIA Tesla V100 GPU with three different strategies: no compensation strategy (w/o strategy), memory bank strategy (MB), and our virtual negatives (VN). It is worth noting that we use our baseline framework with PR and GDP to conduct experiments. To ensure a fair comparison and mitigate the impact of an inconsistent number of negative representations, we included an equal number of real representations in the experiments without the update strategy. Since the memory bank strategy consumes a substantial amount of memory, necessitating the release of memory and reduction in batch size. Consequently, this leads to a degradation in the performance of pixel-wise contrastive learning. As Tab. 7 shows, the experiment without any compensation strategy achieves 66.89% mIoU. Introducing the memory bank strategy improves the mIoU by 0.22%, while our VN strategy achieves a more substantial improvement of **1.12**% mIoU. It is important to consider the trade-off between the performance gain and the associated resource costs. The memory bank strategy consumes a considerable amount of memory, approximately 2.63 GB, while our VN strategy requires only **42** KB of memory. Additionally, the training time for our VN strategy increases by only **0.03** GPU days, whereas the memory bank strategy requires an additional 0.47 GPU days. These results suggest that the memory bank strategy may not be the optimal solution for pixel-wise contrastive learning. Due to the limited storage capacity, the number of representations stored in memory bank is limited and the update in memory bank is frequent. Therefore, the memory bank approach discards most historical representations and loses global features. In contrast, our VNs, generated by GDPs, consider all historical representations because the update strategy for GDP utilizes all representations in the training process. As a result, our VNs are more effective in capturing global features.

Table 7 Comparison under different compensation strategies. All experiments are performed on Pascal VOC 2012 with 92 labeled images. The measurement of times is GPU days.

w/o strategy		mIoU	Memory	Times
		66.89	0M	1.51
MB	size	mIoU	Memory	Times
	30000	66.05	1.20GB	1.67
	65536	67.11	2.63GB	1.98
VN	β	mIoU	Memory	Times
	0.7	66.95	42KB	1.54
	0.8	**68.01**		
	0.9	66.04		
	1.0	66.23		
	1.1	67.98		
	1.2	67.06		

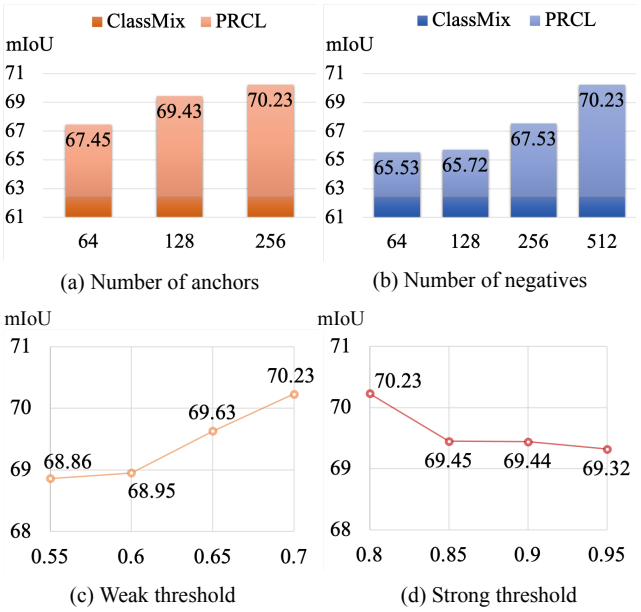

Fig. 11 Ablation study on the number of samples and thresholds.

5.4 Other Ablative Studies

In this section, we conduct some other ablative studies to further explore our framework.

Impact of sampling numbers. We conducted two sets of experiments to investigate the impact of different sampling numbers on the performance of our framework. Firstly, we explored the effect of varying the number of anchors. As Fig. 11 (a) shows, increasing the number of anchors led to improved performance within the constraints of our limited GPU memory. The number of anchors exhibited a significant influence on the overall performance, suggesting that a careful selection of anchor representations is crucial for achieving optimal results. Similarly, we examined the influence of different numbers of negatives in our experiments. Fig. 11 (b) illustrates that increasing the number of negatives resulted in improved performance. This can be attributed to the fact that a larger set of negatives provides a more comprehensive representation of the ideal negative distribution, which in turn has a positive impact on the contrastive learning process. These findings also support our motivation for compensating for the distribution of negative representations at a global level.

Impact of thresholds. We conduct multiple experiments with different strong thresholds δ_s and weak thresholds δ_w by varying one threshold while keeping the other constant. As shown in Fig. 11 (c) and (d), the strategy with $\delta_s = 0.80$ and $\delta_w = 0.70$ yields the best performance. This outcome can be primarily attributed to the sampling of more ambiguous and challenging anchor representations achieved by using a lower δ_s. And we argue that more ambiguous and hard anchors are valuable in

Table 8 Ablation study on the effectiveness of components in our framework, including Probabilistic Representation (PR), Global Distribution Prototype (GDP) and Virtual Negatives (VN)

component	mIoU	obtain
baseline	66.91	-
PR	68.49	1.58
PR+GDP	69.52	2.61
PR+VN	69.07	2.16
PR+GDP+VN	70.23	3.32

contrast since the model does not fully grasp the information of these corresponding pixels.

Impact of components. We conduct experiments in Tab. 8 to ablate each component of our framework on PASCAL VOC 2012 with 92 labeled images. Our baseline achieves mIoU of 66.91%. Simply substituting vanilla representation for probabilistic representation (PR) improves the baseline by 1.58% mIoU. On the basis of this, we introduce the global distribution prototype (GDP), which additionally boosts the performance of 1.03% mIoU. This improvement demonstrates the necessity and effectiveness of prototype consistency. Meanwhile, we introduce virtual negatives (VN), which compensate for the fragmentary negative distribution and include more global features in the contrast. This strategy also boosts the performance of 0.58% mIoU. Finally, combining these two increases performance by 1.74% mIoU. Collectively, these experiments demonstrate the significant and distinct contributions of each component in our proposed method, culminating in improved performance and highlighting the effectiveness of our framework.

6 Conclusion

In this paper, we present a novel framework, termed PRCL, which enhances the robustness of contrastive learning by incorporating probabilistic representations, thus effectively addressing the challenges posed by inaccurate pseudo-labels. Furthermore, our method introduces two key components, namely the global distribution prototype and the virtual negative, to overcome the limitations arising from the limited size of the mini-batch. Comprehensive experiments conducted on various datasets validate the efficacy of our proposed components, as they significantly improve the model's robustness and enhance overall performance.

A Proof of Equation 4

Generally, we regard the prototype as the posterior distribution after the n^{th} observations of representations $\{z_1, z_2, \ldots, z_n\}$. Meanwhile, we assume that all the observations are conditionally independent, the distribution prototype can be derived as $p(\boldsymbol{\rho}|\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_n)$. Without loss of generality, we only consider a one-dimensional case here. It is easy to extend the proof to all dimensions since each dimension of the feature is supposed to be independent. We assume that the distribution prototype $p(\boldsymbol{\rho}|\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_n)$ is with $\hat{\mu}_n$ and $\hat{\sigma}_n^2$ as mean and variance, respectively. Now we need to add a new representation as the observation to obtain a new prototype $p(\boldsymbol{\rho}|\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_{n+1})$, if we take log on this prototype, we have:

$$\begin{aligned}
&\log p(\boldsymbol{\rho}|\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_{n+1}) \\
=&\log p(\boldsymbol{\rho}|\boldsymbol{z}_{n+1}) + \log p(\boldsymbol{\rho}|\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_n) - \log p(\boldsymbol{\rho}) + const \\
=&-\frac{(\boldsymbol{\rho}-\boldsymbol{\mu}_{n+1})^2}{2\boldsymbol{\sigma}_{n+1}^2} - \frac{(\boldsymbol{\rho}-\hat{\boldsymbol{\mu}}_n)^2}{2\hat{\boldsymbol{\sigma}}_n^2} + \frac{(\boldsymbol{\rho}-\boldsymbol{\mu}_0)^2}{2\boldsymbol{\sigma}_0^2} + const \\
=&-\frac{(\boldsymbol{\rho}-\hat{\boldsymbol{\mu}}_{n+1})^2}{2\hat{\boldsymbol{\sigma}}_{n+1}^2} + const,
\end{aligned} \tag{17}$$

where "const" means the constant which is irrelevant to the prototype $\boldsymbol{\rho}$ and

$$\hat{\boldsymbol{\mu}}_{n+1} = \hat{\boldsymbol{\sigma}}_{n+1}^2\left(\frac{\boldsymbol{\mu}_{n+1}}{\boldsymbol{\sigma}_{n+1}^2} + \frac{\hat{\boldsymbol{\mu}}_n}{\hat{\boldsymbol{\sigma}}_n^2} - \frac{\boldsymbol{\mu}_0}{\boldsymbol{\sigma}_0^2}\right), \tag{18}$$

$$\frac{1}{\hat{\boldsymbol{\sigma}}_{n+1}^2} = \frac{1}{\boldsymbol{\sigma}_{n+1}^2} + \frac{1}{\boldsymbol{\sigma}_n^2} - \frac{1}{\boldsymbol{\sigma}_0^2}. \tag{19}$$

$\boldsymbol{\sigma}_0$ is the $\boldsymbol{\sigma}$ of the first representation. At the beginning of the training process, the representation is unreasonable, so we consider that the reliability is quite low and $\boldsymbol{\sigma}_0 \to \infty$ (corresponds to an extremely large value in experiments). We have

$$\hat{\boldsymbol{\mu}}_{n+1} = \frac{\hat{\boldsymbol{\sigma}}_n^2\boldsymbol{\mu}_{n+1} + \boldsymbol{\sigma}_{n+1}^2\hat{\boldsymbol{\mu}}_n}{\boldsymbol{\sigma}_{n+1}^2 + \hat{\boldsymbol{\sigma}}_n^2}, \tag{20}$$

$$\frac{1}{\hat{\boldsymbol{\sigma}}_{n+1}} = \frac{\boldsymbol{\sigma}_{n+1}^2 + \hat{\boldsymbol{\sigma}}_n^2}{\boldsymbol{\sigma}_{n+1}^2\hat{\boldsymbol{\sigma}}_n^2}. \tag{21}$$

Above is the process of obtaining ρ_{n+1} through ρ_n. Next, we briefly give the solution of obtaining ρ_{n+1} using $n+1$ representations.

$$\begin{aligned}
&\log p\left(\rho \mid \boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_n\right) \\
=&\log\left[\alpha p\left(\rho \mid \boldsymbol{z}_1\right)\prod_{i=2}^{n}\frac{p\left(\rho \mid \boldsymbol{z}_i\right)}{p(\rho)}\right] \\
=&(n-1)\log p(\rho) - \sum_{i=1}^{n}\log p\left(\rho \mid \boldsymbol{z}_i\right) + const \\
=&(n-1)\frac{(\rho-\mu_0)^2}{2\sigma_0^2} - \sum_{i=1}^{n}\frac{(\rho-\mu_i)^2}{2\sigma_i^2} + const \\
=&-\frac{(\rho-\hat{\mu}_n)^2}{2\hat{\sigma}_n^2} + const,
\end{aligned} \tag{22}$$

where $\alpha = \frac{\prod_{i=1}^{n} p(\boldsymbol{\rho}_i)}{p(\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_n)}$ and

$$\hat{\boldsymbol{\mu}}_n = \sum_{i=1}^{n}\frac{\hat{\boldsymbol{\sigma}}_n^2}{\boldsymbol{\sigma}_i^2}\boldsymbol{\mu}_i - (n-1)\frac{\hat{\boldsymbol{\sigma}}_n^2}{\boldsymbol{\sigma}_0^2}\boldsymbol{\mu}_0, \tag{23}$$

$$\frac{1}{\hat{\boldsymbol{\sigma}}^2} = \sum_{i=1}^{n}\frac{1}{\boldsymbol{\sigma}_i^2} - (n-1)\frac{1}{\boldsymbol{\sigma}_0^2}. \tag{24}$$

Because $\boldsymbol{\sigma}_0 \to \infty$, we have

$$\hat{\boldsymbol{\mu}}_n = \sum_{i=1}^{n}\frac{\hat{\boldsymbol{\sigma}}_n^2}{\boldsymbol{\sigma}_i^2}\boldsymbol{\mu}_i, \tag{25}$$

$$\frac{1}{\hat{\boldsymbol{\sigma}}^2} = \sum_{i=1}^{n}\frac{1}{\boldsymbol{\sigma}_i^2}. \tag{26}$$

References

1. Alonso, I.n., Sabater, A., Ferstl, D., Montesano, L., Murillo, A.C.: Semi-supervised semantic segmentation with pixel-level contrastive learning from a class-wise memory bank. In: ICCV (2021)
2. Bishop, C.M.: Mixture density networks. Neural Computing Research Group Report (1994)
3. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: ECCV (2018)
4. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: ICML (2020)
5. Chen, X., Yuan, Y., Zeng, G., Wang, J.: Semi-supervised semantic segmentation with cross pseudo supervision. In: CVPR (2021)
6. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR (2016)
7. Davies, D.L., Bouldin, D.W.: A cluster separation measure. TPAMI (1979)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009)
9. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV (2010)
10. Fan, J., Gao, B., Jin, H., Jiang, L.: Ucc: Uncertainty guided cross-head co-training for semi-supervised semantic segmentation. In: CVPR (2022)
11. Feng, Z., Zhou, Q., Gu, Q., Tan, X., Cheng, G., Lu, X., Shi, J., Ma, L.: Dmt: Dynamic mutual training for semi-supervised learning. Pattern Recognition (2022)
12. French, G., Aila, T., Laine, S., Mackiewicz, M., Finlayson, G.: Semi-supervised semantic segmentation needs strong, high-dimensional perturbations (2020)
13. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Communications of the ACM (2020)
14. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al.: Bootstrap your own latent-a new approach to self-supervised learning. NeurIPS (2020)
15. Guan, D., Huang, J., Xiao, A., Lu, S.: Unbiased subclass regularization for semi-supervised semantic segmentation. In: CVPR (2022)
16. Hariharan, B., Arbeláez, P., Bourdev, L., Maji, S., Malik, J.: Semantic contours from inverse detectors. In: IJCV (2011)
17. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR (2020)
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)

19. He, R., Yang, J., Qi, X.: Re-distributing biased pseudo labels for semi-supervised semantic segmentation: A baseline investigation. In: ICCV (2021)
20. Hoffman, J., Wang, D., Yu, F., Darrell, T.: Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. arXiv (2016)
21. Hu, H., Cui, J., Wang, L.: Region-aware contrastive learning for semantic segmentation. In: CVPR (2021)
22. Hu, H., Wei, F., Hu, H., Ye, Q., Cui, J., Wang, L.: Semi-supervised semantic segmentation via adaptive equalization learning. NeurIPS (2021)
23. Hung, W.C., Tsai, Y.H., Liou, Y.T., Lin, Y.Y., Yang, M.H.: Adversarial learning for semi-supervised semantic segmentation. arXiv (2018)
24. Jabri, A., Owens, A., Efros, A.: Space-time correspondence as a contrastive random walk. NeurIPS (2020)
25. Jiang, Z., Li, Y., Yang, C., Gao, P., Wang, Y., Tai, Y., Wang, C.: Prototypical contrast adaptation for domain adaptive semantic segmentation. In: ECCV (2022)
26. Kalluri, T., Varma, G., Chandraker, M., Jawahar, C.: Universal semi-supervised semantic segmentation. In: ICCV (2019)
27. Ke, R., Aviles-Rivero, A.I., Pandey, S., Reddy, S., Schönlieb, C.B.: A three-stage self-training framework for semi-supervised semantic segmentation. TIP (2022)
28. Ke, T.W., Hwang, J.J., Yu, S.X.: Universal weakly supervised segmentation by pixel-to-segment contrastive learning. In: ICLR (2021)
29. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv (2013)
30. Kwon, D., Kwak, S.: Semi-supervised semantic segmentation with error localization network. In: CVPR (2022)
31. Lai, X., Tian, Z., Jiang, L., Liu, S., Zhao, H., Wang, L., Jia, J.: Semi-supervised semantic segmentation with directional context-aware consistency. In: CVPR (2021)
32. Li, D., Yang, J., Kreis, K., Torralba, A., Fidler, S.: Semantic segmentation with generative models: Semi-supervised learning and strong out-of-domain generalization. In: CVPR (2021)
33. Li, S., Xu, J., Xu, X., Shen, P., Li, S., Hooi, B.: Spherical confidence learning for face recognition. In: CVPR (2021)
34. Liu, S., Zhi, S., Johns, E., Davison, A.: Bootstrapping semantic segmentation with regional contrast. In: ICLR (2022)
35. Liu, Y., Tian, Y., Chen, Y., Liu, F., Belagiannis, V., Carneiro, G.: Perturbed and strict mean teachers for semi-supervised semantic segmentation. In: CVPR (2022)
36. van der Maaten, L., Hinton, G.: Visualizing data using t-sne. JMLR (2008)
37. Mittal, S., Tatarchenko, M., Brox, T.: Semi-supervised semantic segmentation with high-and low-level consistency. TPAMI (2019)
38. Miyato, T., Maeda, S.i., Koyama, M., Ishii, S.: Virtual adversarial training: a regularization method for supervised and semi-supervised learning. TPAMI (2018)
39. Oh, S.J., Gallagher, A.C., Murphy, K.P., Schroff, F., Pan, J., Roth, J.: Modeling uncertainty with hedged instance embeddings. In: ICLR (2019)
40. Olsson, V., Tranheden, W., Pinto, J., Svensson, L.: Classmix: Segmentation-based data augmentation for semi-supervised learning. In: WACV (2021)
41. Van den Oord, A., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv (2018)
42. Ouali, Y., Hudelot, C., Tami, M.: Semi-supervised semantic segmentation with cross-consistency training. In: CVPR (2020)
43. Park, J., Lee, J., Kim, I.J., Sohn, K.: Probabilistic representations for video contrastive learning. In: CVPR (2022)
44. Peng, J., Estrada, G., Pedersoli, M., Desrosiers, C.: Deep co-training for semi-supervised image segmentation. PR (2020)
45. Qiao, P., Wei, Z., Wang, Y., Wang, Z., Song, G., Xu, F., Ji, X., Liu, C., Chen, J.: Fuzzy positive learning for semi-supervised semantic segmentation. In: CVPR (2023)
46. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics (1987)
47. Scott, T.R., Gallagher, A.C., Mozer, M.C.: von mises-fisher loss: An exploration of embedding geometries for supervised learning. In: ICCV (2021)
48. Scott, T.R., Ridgeway, K., Mozer, M.C.: Stochastic prototype embeddings. arXiv (2019)
49. Shi, Y., Jain, A.: Probabilistic face embeddings. In: ICCV (2019)
50. Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C.A., Cubuk, E.D., Kurakin, A., Li, C.L.: Fixmatch: Simplifying semi-supervised learning with consistency and confidence. NeurIPS (2020)
51. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. NeurIPS (2017)
52. Vaseghi, S.V.: Advanced digital signal processing and noise reduction. John Wiley & Sons (2008)
53. Wang, C., Xie, H., Yuan, Y., Fu, C., Yue, X.: Space engage: Collaborative space supervision for contrastive-based semi-supervised semantic segmentation. In: ICCV (2023)
54. Wang, W., Zhou, T., Yu, F., Dai, J., Konukoglu, E., Van Gool, L.: Exploring cross-image pixel contrast for semantic segmentation. In: ICCV (2021)
55. Wang, X., Zhang, B., Yu, L., Xiao, J.: Hunting sparsity: Density-guided contrastive learning for semi-supervised semantic segmentation. In: CVPR (2023)
56. Wang, Y., Wang, H., Shen, Y., Fei, J., Li, W., Jin, G., Wu, L., Zhao, R., Le, X.: Semi-supervised semantic segmentation using unreliable pseudo labels. In: CVPR (2022)
57. Wei, C., Sohn, K., Mellina, C., Yuille, A., Yang, F.: Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. In: CVPR (2021)
58. Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: CVPR (2018)
59. Xiao, T., Liu, S., De Mello, S., Yu, Z., Kautz, J., Yang, M.H.: Learning contrastive representation for semantic correspondence. IJCV (2022)
60. Xie, B., Li, S., Li, M., Liu, C.H., Huang, G., Wang, G.: Sepico: Semantic-guided pixel contrast for domain adaptive semantic segmentation. TPAMI (2023)
61. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: Segformer: Simple and efficient design for semantic segmentation with transformers. NeruIPS (2021)
62. Xie, Z., Lin, Y., Zhang, Z., Cao, Y., Lin, S., Hu, H.: Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In: CVPR (2021)
63. Xu, H.M., Liu, L., Bian, Q., Yang, Z.: Semi-supervised semantic segmentation with prototype-based consistency regularization. NeurIPS (2022)

64. Yang, L., Zhuo, W., Qi, L., Shi, Y., Gao, Y.: St++: Make self-training work better for semi-supervised semantic segmentation. In: CVPR (2022)
65. Ye, M., Zhang, X., Yuen, P.C., Chang, S.F.: Unsupervised embedding learning via invariant and spreading instance feature. In: CVPR (2019)
66. Zheng, X., Luo, Y., Wang, H., Fu, C., Wang, L.: Transformer-cnn cohort: Semi-supervised semantic segmentation by the best of both students. arXiv (2022)
67. Zhou, B., Cui, Q., Wei, X.S., Chen, Z.M.: Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In: CVPR (2020)
68. Zhou, T., Wang, W., Konukoglu, E., Van Gool, L.: Rethinking semantic segmentation: A prototype view. In: CVPR (2022)