Probability-Density-aware Semi-supervised Learning

Shuyang Liu[*,1], **Ruiqiu Zheng**[*,1], **Yunhang Shen**[†,2], **Zhou Yu**[†,1,3],
Ke Li[2], **Xing Sun**[2], **Shaohui Lin**[1,3]
[1] East China Normal University
[2] Youtu Lab, Tencent, Shanghai, China
[3] Laboratory of Advanced Theory and Application in Statistics and Data Science - MOE, China
zyu@stat.ecnu.edu.cn, shenyunhang01@gmail.com

Abstract

Semi-supervised learning (SSL) assumes that neighbor points lie in the same category (*neighbor assumption*), and points in different clusters belong to various categories (*cluster assumption*). Existing methods usually rely on similarity measures to retrieve the similar neighbor points, ignoring *cluster assumption*, which may not utilize unlabeled information sufficiently and effectively. This paper first provides a systematical investigation into the significant role of probability density in SSL and lays a solid theoretical foundation for *cluster assumption*. To this end, we introduce a Probability-Density-Aware Measure (PM) to discern the similarity between neighbor points. To further improve Label Propagation, we also design a Probability-Density-Aware Measure Label Propagation (PMLP) algorithm to fully consider the *cluster assumption* in label propagation. Last, but not least, we prove that traditional pseudo-labeling could be viewed as a particular case of PMLP, which provides a comprehensive theoretical understanding of PMLP's superior performance. Extensive experiments demonstrate that PMLP achieves outstanding performance compared with other recent methods.

Introduction

Machine Learning's remarkable breakthroughs rely on constructing high-quality, complex labeled datasets. However, dataset annotations are increasingly costly or even infeasible in many professional areas (*e.g.*, medical and astronomical fields (Chen et al. 2020b; Xu et al. 2022)). Semi-supervised learning (SSL) has been proposed to mitigate the demand for large-scale labels by extracting information from unlabeled data with guidance from a few labeled data. The existing SSL methods are designed based on the widely accepted *assumption of consistency*, which assumes that close data points probably have the same label and data points lie in the same cluster tend to have the same labels (Zhou et al. 2003; Chapelle and Zien 2005; Iscen et al. 2019; Li et al. 2020; Zhao et al. 2022). The latter prior that points from the same category should lie in the same cluster is specifically called *cluster assumption* (Zhou et al. 2003; Iscen et al. 2019; Li et al. 2020).

In SSL, the *assumption of consistency* is usually considered by assigning the similarity between neighbor points. An enormous similarity indicates that the points tend to be in the same category. Some works use similarity to train a consistent encoder, considering that neighbor input should have neighbor features (Sohn et al. 2020). Contrastive learning learns consistent visual representations with unlabeled images (Chen et al. 2020a). Adversarial training (Miyato et al. 2018) and data augmentation (Berthelot et al. 2019; Sohn et al. 2020) are also widely applied to gain more training samples and improve the learning of representation. High-confidence pseudo-labels are propagated with close neighbor points' predictions and labels (Iscen et al. 2019; Zhao et al. 2022; Elezi et al. 2018). Combining consistent encoder and pseudo-labeling significantly improves the model performance (Iscen et al. 2019). Recently, the widely-used SSL framework often consists of supervised loss to take full consideration of labeled data, unsupervised loss to improve the quality of pseudo labels, and consistency loss to encourage a consistent encoder (Zhao et al. 2022; Berthelot et al. 2019; Zheng et al. 2022).

However, these works fail to fully consider *assumption of consistency*, for they do not use the prior of *cluster assumption*. When calculating the similarity between points, their measurements seldom consider the cluster prior. Considering that the points close to the decision boundary may be closer to points from other categories than some points in the same categories, these algorithms may be misled by wrongly assigning neighbors' labels. Our proposed solution, the Probability-Density-Aware Measurement(PM), has the potential to fully utilize the *cluster assumption* through the incorporation of density information and could address the limitations of existing algorithms. A natural thought goes that for two points with different clusters, their path tends to traverse low-density regions. We statistically prove the thought and demonstrate that the path probably goes through low-density areas. The theorem indicates that density information significantly helps to consider the *cluster assumption* and severs as the keystone of PM.

With PM, we can better measure the similarity between points. We deploy PM into Label Propagation to select better neighbors and generate higher-confidence pseudo-labels. The new algorithm is named Probability-Density-Aware Label-Propagation(PMLP). To show PM's superiority and

[*] The authors contribute equally to this paper.
[†] The corresponding author of this paper.

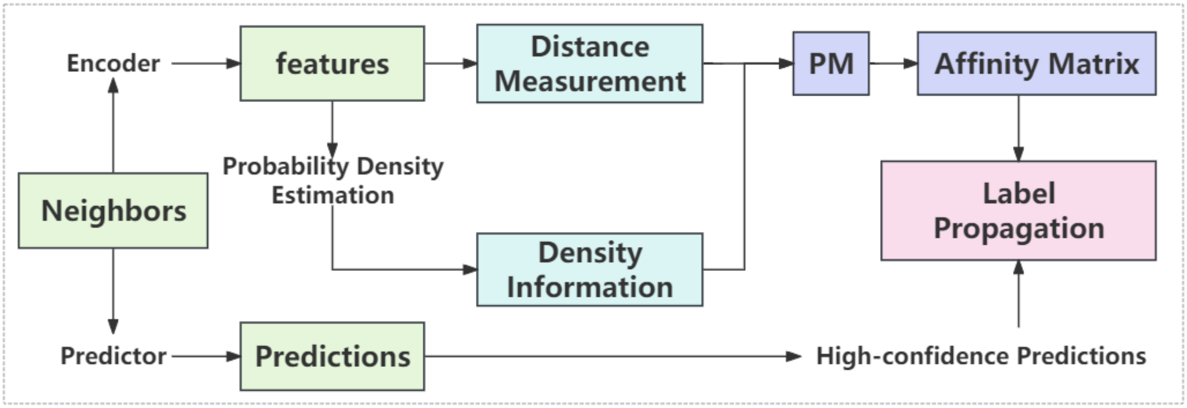

Figure 1: The procedure of PMLP. First, we select the neighbor points and extract their features. Then, we calculate the densities on the path; the density information is used to construct the Probability-density-aware Measure(PM). PM can fully consider the *cluster assumption*. Finally, high-confidence predictions are used for pseudo-labeling with an affinity matrix.

generality, we prove that traditional pseudo-labeling is a particular case of PMLP, which indicates that PMLP should always be better with just minor optimization. PM can be easily implemented to complement many popular SSL methods involving label propagation (Zhou et al. 2003; Iscen et al. 2019; Zhao et al. 2022; Elezi et al. 2018); we deploy PM to the best performance LPA algorithm (Zhao et al. 2022) to implement the experiment. Through comprehensive experiments on benchmark datasets, we show that PMLP achieves noticeable performance improvement compared to some of the most powerful approaches. In general, our contribution can be summarized into three points:

- We first propose a theoretical description to *cluster assumption*, which reveals that probability density can help make full use of SSL's prior;
- We introduce a new Probablity-Density-Aware similarity measurement(PM) to fully consider SSL's cluster prior. Probability-Density-Aware Label-Propagation(PMLP) is proposed based on PM. We mathematically prove PMLP's superiority and generality;
- Extensive experiments are conducted to validate the effectiveness of PMLP. For example, on CIFAR100 with 400 labeled samples, we can surpass the baseline model by 3.52% and surpass the second-best algorithm by 1.21%.

Related Works

Consistency Regularization

The *assumption of consistency* is widely accepted in SSL. Its foundation is generating well-consistency models that exhibit similar features for similar inputs. The works in (Hinton, Osindero, and Teh 2006; Bengio et al. 2006) improved consistency via self-training and finetuning the model with labeled data. $\Pi - model$ (Laine and Aila 2016) reduces the consistency loss between images and their augmentations, or between outputs and temporal average outputs. Meanteacher (Tarvainen and Valpola 2017) reduces consistency loss between current weights and temporal average weights. Data augmentation can help to extract more feature information (Miyato et al. 2018; Elezi et al. 2018). Recent algorithm (Chen et al. 2020b) involved contrastive learning to improve the model consistency, and LASSL (Zhao et al. 2022) generated class-aware contrastive learning by limiting the comparisons within the same class. However, these works fail to produce pseudo-labels directly.

Pseudo-labeling

Generating high-confidence pseudo-labels for unlabeled data is a widely used policy in SSL. A simple yet intuitive approach considers high-confidence probability(higher than a threshold τ) vectors to be accurate and assigns them pseudo-labels (Sohn et al. 2020). Freematch and Adsh (Wang et al. 2022; Guo and Li 2022) adaptive adjust threshold τ to improve the pseudo-label quality. Some early pseudo-labeling algorithms (Lee 2013; Sajjadi, Javanmardi, and Tasdizen 2016; Shi et al. 2018) rely heavily on predictions, while recent frameworks prefer graph-based label propagation yielding better consideration of *assumption of consistency* (Zhou et al. 2003; Iscen et al. 2019; Zhao et al. 2022; Douze et al. 2018). Graph transduction game (Elezi et al. 2018; **?**) fixes the graph and lacks a weighting mechanism. Some statistical methods (Zhou et al. 2003) take a reasonable consideration of *assumption of consistency* by calculating the affinities via the first-order similarity between features and iteratively spread labels to their neighbors. The strategy can extend to the high-dimensional feature space in deep learning (Iscen et al. 2019; Zhao et al. 2022). Comatch *et al.* (Li et al. 2021) introduces graphic information in contrastive learning and smooths pseudo-labels with a memory bank. However, these works fail to fully consider the *cluster assumption*.

Density-aware Methods

Although the above techniques effectively select close neighbors, the *cluster assumption* is not sufficiently considered, and they fail to distinguish neighbors' cluster affiliation directly. Some works intuitively point out that the decision boundary tends to be placed in a low-density region (Chapelle and Zien 2005; Li et al. 2020), and this prior helps consider *cluster assumption*. Wasserman *et al.* (Azizyan, Singh, and Wasserman 2013) proposes a density-sensitive distance metric that provides a lower value when the points are located in different clusters. Connectivity kernel (Chapelle and Zien 2005; Bousquet, Chapelle, and Hein 2003) can utilize the detection of density variations along a path to determine whether two points lie in the same cluster. These statistical methods can thoroughly consider *cluster assumption*, but their optimization in high-dimensional space needs expensive computation. Recently, DNA and DPLP (Li et al. 2020) prefer higher-density neighbors and construct a density-ascending path for pseudo-labeling. However, their density is cosine similarity, which harms the theoretical guarantee and statistical interpretability. Simmatch (Zheng et al. 2022) tries to optimize pseudo labels with marginal distribution, but labeled data in SSL is few.

Our algorithm aims at providing a general framework to fully consider the *cluster assumption* by density-aware distance, and thus produce high-quality pseudo-labels.

Methedology

A Statistical Explanation To *Cluster Assumption*

It has been widely accepted that with *cluster assumption*, points in the same cluster tend to lie in the same category. In the previous work, however, it still lacks mathematical investigation. This section presents an innovative statistical explanation demonstrating probability density is crucial in leveraging the *cluster assumption*. To lay a theoretical analysis, without loss of generality, we assume the points in the feature space are widely accepted sub-gaussian random vectors. Then we have the **Theorem1**:

Theorem1 *Suppose two sub-gaussian random vectors X_1, X_2 with mean μ_1, μ_2. Denote x_1, x_2 randomly sampled from X_1, X_2, and $\gamma(x_1, x_2)$ (Azizyan, Singh, and Wasserman 2013) is a continuous finite path from x_1 to x_2. For any $\tau \in R$, denote $p(x)$ is the probability density of point x estimated by Kernel Density Estimator (KDE), define event:*

$$C_\tau : \{\exists x \in \gamma(x_1, x_2), p(x) \leq \tau\}.$$

Then with $||\mu_1 - \mu_2||_2^2 \to \infty$, we have:

$$P(C_\tau) \to 1.$$

We can further prove that for two sufficiently dispersed X_1, X_2, almost every point on the straight line between x_1, x_2 traverses a low-density region.

Corollary1 *Specifically take $\gamma(x_1, x_2)$ in **Theorem1** as a straight line and $x \in \gamma(x_1, x_2)$ distributes uniformly. For $\forall \tau \in R$, with sufficient large $||\mu_1 - \mu_2||_2^2$, we have:*

$$\lim_{||\mu_1 - \mu_2||_2^2 \to \infty} P(\{x : p(x) \leq \tau \,, x \in \gamma(x_1, x_2)\}) \to 1.$$

We have statistically proved that density information helps consider the *cluster assumption* besides traditional distance measures. Specifically, their path probably traverses a low-density region when the points lie in different clusters. We simply choose $\gamma(x_1, x_2)$ as the connecting line $l(x_1, x_2)$ to save computation and equidistant select K samples $\{x_{i,j}^1, \ldots, x_{i,j}^k\}$ along $l(x_1, x_2)$. We estimate their densities $\{p(x_{i,j}^1), ..., p(x_{i,j}^k)\}$ with Kernel Density Estimator(KDE) and the generated density information is denoted as $\{I(p_{i,j}) | \{p(x_{i,j}^1), ..., p(x_{i,j}^k)\}\}$. We hope $I(p_{i,j})$ tends large when $l(x_i, x_j)$ traverses inside one cluster and tends small when $l(x_i, x_j)$ traverses among different clusters.

With any traditional distance measure $D(x_i, x_j)$ (such as first-order similarity, Euclidean distance, and cosine similarity), we propose our density-aware distance measure:

$$\begin{cases} d(x_i, x_j) = I(p_{i,j}) D^{-1}(x_i, x_j), & \text{if } i \neq j; \\ d(x_i, x_j) = 0, & \text{if } i = j. \end{cases}$$

We name $d(x_i, x_j)$ as Probability-Density-Aware Measure(PM). The **Theorem2** presents four choices of $I(p_{i,j})$ and proves PM's superiority. In PM, the distance measure $D(x_i, x_j)$ reflects the *neighbor assumption* that neighbor points tend to have the same label, and the density information $I(p_{i,j})$ reflects the *cluster assumption* that features lie in different clusters tend to have different labels, thus takes full consideration of *assumption of consistency*.

Probability-Density-Aware Measure Label-Propagation

Traditional Label Propagation Algorithm (LPA) (Zhou et al. 2003; Iscen et al. 2019; Zhao et al. 2022) search neighbor points by a distance measure $D(x_i, x_j)$ including first-order similarity, cosine similarity, and Euclidean distance, which lack consideration of *cluster assumption*. We propose Probability-Density-Aware Measure Label-Propagation (PMLP) by introducing the PM $d(x_i, x_j) = I(p_{i,j}) D^{-1}(x_i, x_j)$ into pseudo-labeling. Theorem 2 lays a theoretical foundation for the good performance of PMLP.

Suppose we have observations O and labeled data (X^L, Y^L), unlabeled data and its prediction $(X^U, p(X^U))$. As assumed in LPA, we propagate high-confidence predictions to low-confidence predictions by similarity measures between neighbor points.

As a familiar setting in LPA, we use KNN to choose K nearest neighbors in feature space O, and then we get new $O^{LP} = \{o_1^{LP}, ..., o_K^{LP}\}$, and the corresponding $Y^{LP} = \{y_1^{LP}, ..., y_K^{LP}\} \in \{p(X^U)\} \cup \{Y^L\}$. Then we reweight Y^{LP} by threshold τ:

$$\begin{aligned} y_t^{LP,\text{high}} &= I(\max(y_t^{LP}) \geq \tau) y_t^{LP}, \\ y_t^{LP,\text{low}} &= I(\max(y_t^{LP}) < \tau) y_t^{LP}, \end{aligned} \tag{1}$$

where $Y^{LP,\text{high}}$ includes high-confidence predictions $y_t^{LP,\text{high}}$ and ground-truth labels Y^L. Denote any distance measurements between features as $D(o_i^{LP}, o_j^{LP})$ in traditional LPA. To fully consider *cluster assumption*, we introduce PM $d(x_i, x_j)$ to introduce the density information between two features o_i^{LP}, o_j^{LP}. Then we get the reweighted

Algorithm 1: The algorithm for density-aware label propagation

```
Input:
$K$, $Y^{LP}$, $O^{LP}$, threshold $\tau$, k, n, $\alpha$, $\eta$.
Output: density-aware pseudo-label $\widehat{Y}^{\star LP}$

Choose $K$ nearest neighbor points in features $O^{LP}$;
Seperate $Y^{LP,\text{high}}$ and $Y^{LP,\text{low}}$ by Eq. 1 with $\tau$;
For points in $Y^{LP,\text{low}}$ and $Y^{LP,\text{high}}$, calculate the distance $D(x_i, x_j)$.
Get k-equal division points on the connecting line, for each point in $o^1_{i,j}, ..., o^k_{i,j}$, select $n$ neighbors to calculate density $p_{i,j}$ with KDE;
Reweight $W$ with $I(p_{i,j})$ as Eq. 2, get $W'$;
Propagating the label with $W'$, $\alpha$ by Eq. 3, get $Y^{'LP}$;
Mix the propagated label $Y^{'LP}$ and $Y^{LP}$ by Eq. 4 with $\eta$, get final density-aware pseudo-label $\widehat{Y}^{*LP}$.
```

adjacent matrix W':

$$W'_{i,j} = \begin{cases} 0, & \text{if } i = j, \\ I(p_{i,j}) D^{-1}(o_i^{LP}, o_j^{LP}), & \text{if } i \neq j, \end{cases} \tag{2}$$

Then, the pseudo-labels are generated by iteratively propagating the high-confidence predictions to their close and same-cluster neighbors based on the affinity matrix W'. Denote $\widehat{Y}'^{LP}(i)$ implies the propagated pseudo-label in i-th iteration. Note D the diagonal matrix with (i, i)-th equal to the sum of the i-th row of W', and iteratively propagate the label information to unlabeled points:

$$\widehat{Y}'^{LP}(i) = \alpha D^{-\frac{1}{2}} W' D^{-\frac{1}{2}} \widehat{Y}^{LP}(i-1) + (1-\alpha) Y^{LP,\text{high}}. \tag{3}$$

We get the final optimal form $\widehat{Y}'^{LP}$:

$$\widehat{Y}'^{LP} = (I - \alpha D'^{-\frac{1}{2}} W' D'^{-\frac{1}{2}})^{-1} Y^{LP,\text{high}}.$$

And the final pseudo-label $\widehat{Y}^{*LP}$ is combined by $\widehat{Y}'^{LP}$ and our prediction $Y^{LP,low}$:

$$\widehat{Y}^{*LP} = \eta \widehat{Y}'^{LP} + (1-\eta)(Y^{LP,\text{low}}). \tag{4}$$

The superiority and generalization of PMLP lie in its good theoretical guarantee: whatever the choice of $D(o_i^{LP}, o_j^{LP})$ and loss function $L(Y^P, \widehat{Y}^{*LP})$, we prove that, when $I(p_{i,j})$ is chosen as:

$$\begin{cases} I(p_{i,j}) = \max\{p_t(i,j), t = 1, 2, ..., k\}, & \text{if } i = j, \\ I(p_{i,j}) = \min\{p_t(i,j), t = 1, 2, ..., k\}, & \text{if } i \neq j, \\ I(p_{i,j}) = \text{avg}(\{p_t(i,j), t = 1, 2, ..., k\}), & \text{if } i \neq j, \\ I(p_{i,j}) = Q_t(\{p_t(i,j), t = 1, 2, ..., k\})), & \text{if } i \neq j, \end{cases}$$

where $\text{avg}(\cdot)$ denotes the mean and $Q_t(\cdot)$ denotes the $t-quantile$, in our implemention, we take it as the median. The procedure of PMLP is in pseudo code 1.

We prove that PMLP can consistently outperform traditional LPA. Traditional LPA is a particular case of PMLP, indicating both PMLP's superiority and reasonableness.

Theorem2. *Denote the model parameter in one iteration as θ. And let $\mathcal{L}_U$ be the unsupervised loss between predictions and pseudo-labels. Denote Y^P as the set of predictions, $\widehat{Y}^{*LP}$ and $\widehat{Y}^{LP}$ be the pseudo-labels generated by PMLP and traditional LPA with any $D(\cdot,\cdot)$ (Zhou et al. 2003; Iscen et al. 2019; Zhao et al. 2022). Suppose we calculate the probability density by KDE with exponential kernel and bandwidth h. Then $\mathcal{L}_U$ satisfies:*

$$\min_{\theta\in\Theta, h\in R} \mathcal{L}_U(Y^P, \widehat{Y}^{*LP}) \leq \min_{\theta\in\Theta} \mathcal{L}_U(Y^P, \widehat{Y}^{LP}).$$

The Competitive Pseudo-Labeling

Recent pseudo-labeling algorithms combine a consistent encoder, considering that a better encoder can generate better representations and facilitate the pseudo-labeling (Iscen et al. 2019; Zhao et al. 2022; Berthelot et al. 2019; Sohn et al. 2020; Zheng et al. 2022). We inherit this framework in the implementation of PMLP. We choose the Label-guided Self-training approach to Semi-supervised Learning (LASSL) (Zhao et al. 2022) as our baseline. As the best-performing label-propagation algorithm in recent years, LASSL still behaves dissatisfiedly compared with other recent algorithms. We hope deploying PMLP with the LASSL framework will allow it to surpass recent algorithms, thus demonstrating the full potential of label propagation under *cluster assumption*.

In our implementation, we update the model with three losses: the supervised cross-entropy loss $\mathcal{L}_S$, the class-aware contrastive (CACL) loss $\mathcal{L}_C$ (Zhao et al. 2022), and the unsupervised cross-entropy loss $\mathcal{L}_U$.

Denote θ as the model's parameters, labeled dataset $X^L = \{X_L, Y_L\}$, the model takes a consideration of labeled data by minimizing $\mathcal{L}_S$:

$$\mathcal{L}_S(X^L; Y^L; \theta) := CE(X^L; Y^L; \theta).$$

Denote the weak and strong augmentation of input x as $a(x), A(x)$, denote $z_i^L = G(a(x_i))$ and $z_j^U = G(A(u_j))$ as the output features for both labeled and unlabeled data. In the t-th iteration, note $Y_{t-1} = \{y_1^L, ..., y_B^L\} \cup \{y_1^U, ..., y_{\mu B}^U\}$ all predictions of labeled and unlabeled data, for $y_i, y_j \in Y_{t-1}$, introduce the instance relationships to help the contrastive learning:

$$w_{i,j} = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j \text{ and } y_i \cdot y_j \leq \epsilon, \\ y_i \cdot y_j, & \text{if } i \neq j \text{ and } y_i \cdot y_j \geq \epsilon, \end{cases}$$

where ϵ is a hyper-parameter determining the confidence that two instances belong to the same category.

Then we get $\mathcal{L}_C$ to help learning a good representation:

$$\mathcal{L}_C = -\Sigma_{i=1}^{|Y_{t-1}|} \log(\frac{\Sigma_{j=1}^{|Y_{t-1}|} w_{i,j} \exp(\frac{z_i \cdot z_j}{T})}{\Sigma_{j=1, j\neq i}^{|Y_{t-1}|} \exp(\frac{z_i \cdot z_j}{T})}),$$

where T is temperature hyper-parameter (Chen et al. 2020b).

Denote $Y^U = \{y_1^U, \ldots, y_{\mu B}^U\}$ the predictions of unlabeled data Unsupervised loss $\mathcal{L}_U$ encourages predictions to be consistent with PMLP's confidence pseudo-labels Y^{*LP},

Method		CIFAR10		CIFAR100		SVHN		STL-10	
		40 labels	250 labels	400 labels	2500 labels	40 labels	250 labels	40 labels	1000 labels
Pseudo-label	(2013)	25.39±0.26	53.51±2.20	12.55±0.85	42.26±0.28	25.39±5.6	79.79±1.09	25.31±0.99	67.36±0.71
Mean-Teacher	(2017)	29.91±1.60	62.54±3.30	18.89±1.44	54.83±1.06	33.81±3.98	96.43±0.11	29.28±1.45	66.10±1.37
MixMatch	(2019)	63.81±6.48	86.37±0.59	32.41±0.66	60.24±0.48	69.40±3.39	96.02±0.23	45.07±0.96	88.3±0.68
UDA	(2019)	89.33±3.75	94.84±0.06	53.61±1.59	72.27±0.21	94.88±4.27	98.01±0.02	62.58±3.44	93.36±0.17
ReMixMatch	(2019)	90.12±1.03	93.70±0.05	57.25±1.05	73.97±0.35	79.96±3.13	97.08±0.48	67.88±6.24	93.26±0.14
FixMatch	(2020)	92.53±0.28	95.14±0.05	53.58±0.82	71.97±0.16	96,19±1.18	98.03±0.01	64.03±4.14	93,75±0.33
ReFixMatch	(2023)	95.06±0.01	95.17±0.05	53.88±1.07	72.72±0.22	71.40±4.21	94.26±0.3	97.85±1.23	98.11±0.03
Softmatch	(2023)	94.89±0.14	95.04±0.09	62.40±024	73.61±0.38	97.54±0.24	97.99±0.01	77.78±3.82	94.21±0.15
EPASS	(2023)	94.69±0.1	94.92±0.05	61.12±0.24	74.32±0.33	97.02±0.02	97.96±0.02	84.39±2.48	94.06±1.42
Freematch	(2023)	95.1±0.04	95.12±0.18	62.02±0.42	73.53±0.2	**98.03±0.02**	98.03±0.01	84.46±0.55	94.37±0.15
Shrinkmatch	(2023)	94.92	95.26	64.64	74.83	97.49	98.04	**85.98**	94.18
LASSL	(2022,baseline)	95.07± 0.78	95.71 ±0.46	62.33±2.69	74.67± 0.65	96.91±0.52	97.85± 0.13	81.57±0.36	94.23±0.26
PMLP+LASSL	(Ours)	**95.42±0.32**	**95.76±0.14**	**65.85±0.8**	**75.27±0.19**	97.85±0.31	**98.10±0.05**	85.53±1.92	**94.53±0.53**
Outperform than LASSL		+ 0.35	+ 0.05	+ **3.52**	+ **0.60**	+ **0.94**	+ 0.25	+ **3.96**	+0.30

Table 1: Performance comparison on CIFAR10, CIFAR100, SVHN, and STL-10. We show the mean accuracy from 5 experiments and the standard deviation.

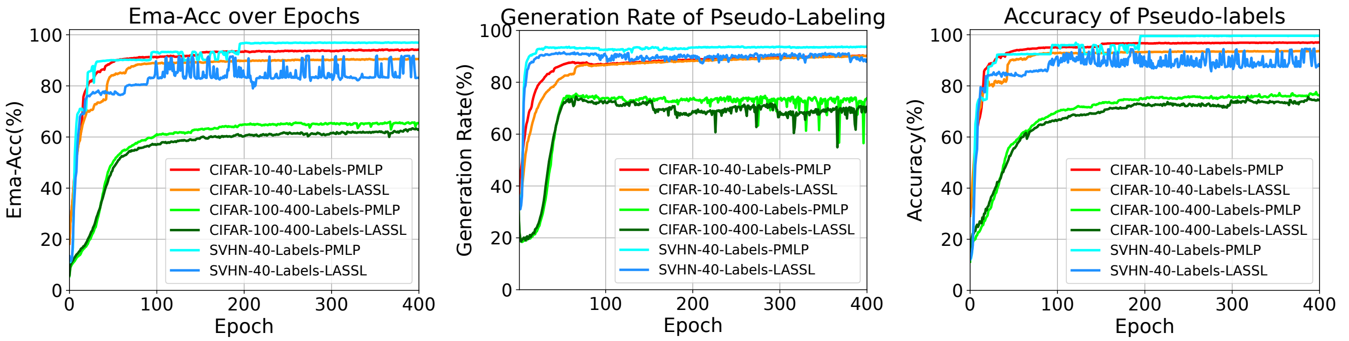

Figure 2: Left: ema-accuracy of models. Middle: rate of high-quality pseudo-labels. Right: rate of correct high-quality labels.

Epoch	0-4	5-9	10-14	15-19
Time(s), PMLP	748.4	747.3	747.1	747.0
Time(s), KDE	3733.0	4244.0	4166.6	4189.0

Table 2: Running time for using KDE and our PMLP to compute the density.

which in turn encourages consistency encoder and helps to improve the pseudo-labeling:

$$\mathcal{L}_U = I(y_j^U \geq \tau) H(Y^U, \widehat{Y}^{*LP}).$$

Combine the three losses, we aim to update the model's parameter θ to minimize the loss function L:

$$\min_{\theta \in \Theta} \mathcal{L} = \min_{\theta \in \Theta} (\mathcal{L}_S + \mathcal{L}_U + \mathcal{L}_C).$$

Experiments

Experiment Setup

[1] We present comprehensive experiments of PMLP across extensive datasets, including SVHN (Krizhevsky and Hinton 2009), CIFAR10 (Krizhevsky and Hinton 2009), and CIFAR100 (Netzer et al. 2011), STL-10 (Adam Coates 2011). Following the standard protocol of SSL (Zhao et al. 2022; Zheng et al. 2022; Wang et al. 2022; Chen et al. 2023; Yang et al. 2023), we randomly select 40 and 250 labeled samples from SVHN and CIFAR10. For CIFAR100, 400 and $2,500$ labeled samples are randomly selected. For STL-10, we select $40, 1000$ labeled samples.

For SVHN and CIFAR10, we use WideResNet-28-2 as the encoder to generate representations. For CIFAR100, encoder is WideResNet-28-8. For STL-10, encoder WideResNet-37-2. The predictor is a one-linear layer network. In the CACL, we calculate z_i with a 2-layer MLP projector. The batch size includes 64 unlabeled and 448

[1] Code: https://github.com/sdagfgaf/Probability-Density-aware-Semi-supervised-Learning

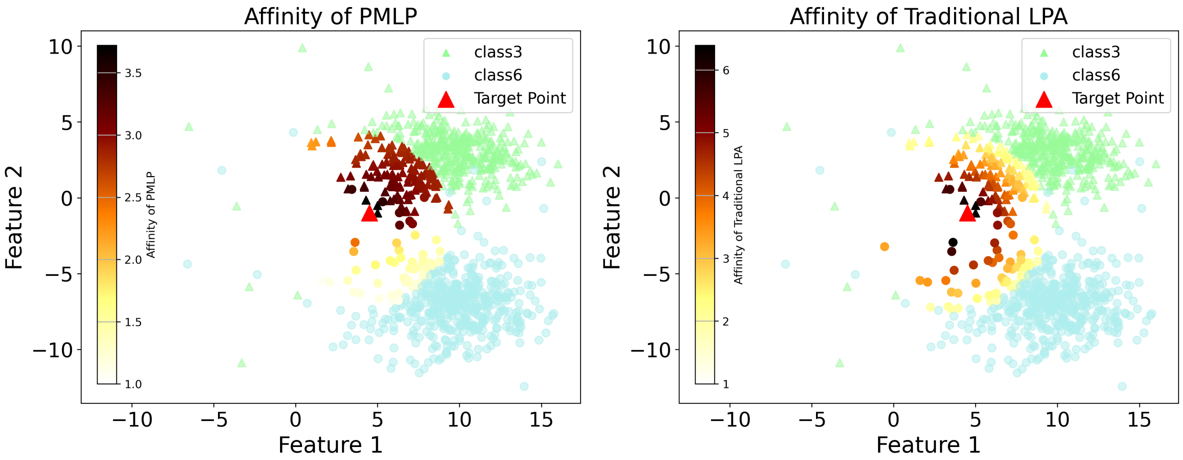

Figure 3: Different colors represent different distances between the target point and neighbor points. Black represents a close distance and a higher affinity. The left one chooses PM as the distance measure, and the right one chooses traditional first-order similarity. PMLP tends to choose neighbors within one cluster, and LPA equally chooses neighbors with different clusters.

Dataset	Method	Average Time (s/epoch)	Average Time Across Different Iterations (s/epoch)			
			1-256	257-512	513-768	769-1024
STL-10	LASSL	615.21	593.81	625.56	618.04	623.40
	LASSL+PMLP	648.08(**+5.34%**)	734.59(**+23.71%**)	652.42(**+4.29%**)	606.64(**-1.85%**)	598.69(**-3.96%**)
CIFAR10	LASSL	211.41	255.52	196.10	196.66	197.35
	LASSL+PMLP	217.94 (**+3.09%**)	260.05 (**+1.77%**)	203.72 (**+3.89%**)	203.52 (**+3.89%**)	204.46 (**+3.60%**)

Table 3: Overall average time and average time of different iterations for different methods on STL-10 and CIFAR-10 datasets.

labeled points. In the label-propagation process, we select $1.5 \times N$ nearest neighbors, where N represents the dataset category. In PMLP, we use $\alpha = 0.8$, $\eta = 0.2$, and $\tau = 0.95$. In CACL, ϵ is set to 0.7. Optimization is performed using the SGD optimizer with a momentum of 0.9 and a weight decay of 5×10^{-4}. The learning rate follows a cosine decay schedule. For KDE, we chose a bandwidth of $h = 5$ for SVHN, CIFAR10, and CIFAR100, and $h = 3$ for STL-10. In KDE, we select 512 points for CIFAR10 and SVHN, and the nearest 45 points for CIFAR100 and STL-10.

Results On SVHN, CIFAR10, CIFAR100, STL-10

Tab. 1 compares the average testing accuracy of PMLP over 5 runs against both classical algorithms and recent SOTA SSL approaches (Zheng et al. 2022; Wang et al. 2022; Chen et al. 2023; Yang et al. 2023; Nguyen 2024; Zhao et al. 2022; Nguyen 2023). Here we set $I(p_{i,j}) = \text{avg}(\{p_t(i,j), t = 1, 2, ..., k\})$, $K = 1$. PMLP consistently outperforms other SOTA approaches on CIFAR10 and CIFAR100 under all settings. With 400 labeled samples in CIFAR100, PMLP achieves an accuracy of 65.85%, surpassing the second-best ShrinkMatch (Yang et al. 2023) by 1.21% and improving our baseline LASSL with 3.52%. We also improved our baseline LASSL with 0.6% thus surpassing the second-best ShrinkMatch with 0.44%. On CIFAR10, PMLP always surpasses ShrinkMatch by more than 0.5%. With 40 labeled samples in STL-10, PMLP reaches an accuracy of 85.53%, improving our baseline model LASSL with 3.96%; for SVHN with 40 labels, our PMLP can outperform the baseline LASSL with 0.94%, while still competitive compared with the SOTA.

PMLP Generates Better Pseudo-labels

In this section, we design experiments to show that PMLP generates better pseudo-labels. Fig. 2 shows that LASSL and PMLP generate almost the same number of high-confidence pseudo-labels, but PMLP's pseudo-labels are more accurate. It directly explains PMLP's superior performance while implying that traditional LPA brings more misleading wrong labels into training, harming the model's performance.

We also visually show the PMLP's selected neighbors in the feature space. By *Theorem1*, with the same distance, we tend to choose neighbors in the same cluster and reject the neighbors in the different clusters. We implement PMLP and classical LPA with 40 CIFAR10 labeled samples and output 64-dimensional features. Principal component analysis (PCA) is applied, and we select the two most important principal components and plot them in Fig. 3. We select the target point from class 3 and its nearest 200 neighbors. In Fig. 3, we deploy PMLP and the classical LPA (Zhou et al. 2003; Iscen et al. 2019; Zhao et al. 2022) and color the neighbors according to the value of $d(x_i, x_j)$ and $D(x_i, x_j)$.

Bandwidth	$I(p_{i,j})$	K=1	K=3	K=5
h=5	$\min(p_k(i,j))$	94.75	**88.82**	**93.70**
	$\max(p_k(i,j))$	95.42	95.19	**87.56**
	$\text{avg}(p_k(i,j))$	95.61	95.35	95.19
	$Q_t(p_k(i,j))$	94.97	95.62	95.45
$h = \infty$, LASSL	None	95.30		

Table 4: Comparison of different K and $I(p_{i,j})$ on CIFAR10 dataset with 40 labeled data.

Bandwidth	h=0.01	h=0.5	h=5	h=100	h=∞	LASSL
Acc	63.34	65.42	**66.5**	64.32	62.50	62.33
Density Ratio	1.73	1.39	1.09	1.009	1	1

Table 5: Ablation study on CIFAR100 with 400 labeled data.

The deeper color represents a closer distance and more significant affinity. It shows that PMLP tends to choose neighbors in the same cluster. In PMLP, we introduce density information $I(p_{i,j})$ into $d(x_i, x_j)$; some neighbors in the other clusters are chosen but assigned with a lower affinity and tend to be diminished in the LPA process.

Acceleration Results Of PMLP

We introduce the density information $I(p_{i,j})$, and the probability density is calculated with KDE. A natural thought goes that KDE is expensive to compute. Considering we choose N samples on the connecting line and pick up n support points each time for KDE, $O(N^2kn)$ computation is needed. The KDE from Sklearn runs only on the CPU, leading to slower calculations. We use the exponential kernel $K(\cdot)$ in KDE to promote the divergence and design a GPU-based KDE, which can reach $\times 5$ acceleration Tab. 2. The details can be seen in the Appendix[2]. We further compare the average time consumption in training between PMLP and our baseline LASSL in Tab. 3, indicating that PMLP does not cost too much time(less than about 5%) than traditional LPA.

Ablation Studies

The Effect Of $I(p_{i,j})$ In Tab. 5, we compare PMLP's performance with different $I(p_{i,j})$ and different K points on the connecting line. We deploy the experiment with the CIFAR10 dataset, 40 labeled samples. We use simplify notations for $I(p_{i,j})$, such as $\text{avg}(\{p_t(i,j), t = 1, 2, ..., k\}) = \text{avg}(p_t(i,j))$. However, it can be seen that $\max(\{p_t(i,j)\})$ and $\min(p_t(i,j))$ are unstable with K. Meanwhile, the $\text{avg}(p_t(i,j))$ and $Q_t(p_k(i,j))$ keeps a stable performance. A thought goes that $\min(p_t(i,j))$ and $\max(p_t(i,j))$ are easily affected by singular points with an outlier density: some high-confidence points may be diminished by $\min(p_t(i,j))$, and incorrect neighbors from the other cluster may be enhanced by $\max(p_t(i,j))$. The mean and median can sufficiently use the density information to get a stable result.

Neighbors(N')	N'=3	N'=8	N'=10	N'=15	N'=20
Acc	82.71	94.21	95.04	**95.55**	95.21

Table 6: Ablation study with N' neighbors on CIFAR10.

We point out that Tab. 4 does not conflict with our **Theorem2**, considering that we forgo sufficiently fine-tuning the bandwidth h for $\max(p_t(i,j)), \min(p_t(i,j))$ considering $\text{avg}(p_t(i,j))$ is more efficient and convenient options.

A Mild Density-aware Strategy Inspired by Tab. 5, a thought goes that $I(p_{i,j})$ reflects the *cluster assumption*, but it does not serve as a sufficient statistic. Thus an incorrect prior at the beginning may severely mislead the training. It motivates us that it's better to take PMLP as a mild punishment that does not reject the far neighbors firmly and control the PM $d(x_i, x_j)$ with a comparably slight disparity.

We choose the $I(p_{i,j}) = \text{avg}(\{p_t(i,j), t = 1, 2, ..., k\})$ to deploy PMLP and fine-tune the bandwidth h to attenuate the influence of density information, as h significantly impacts the KDE. Tab. 4 presents the accuracy of the PMLP on CIFAR100 with different bandwidth h, $K = 1$. When h is relatively small, the density of the midpoints shows significant differentiation. Thus, the affinity matrix W' is notably influenced by the density information $I(p_{i,j})$ and efficiently diminishes the far neighbors. The densities become similar when the h is relatively large, and the W' mildly obsoletes the low-confidence neighbors. By **Theorem2**, setting $h \to \infty$ degenerates PMLP into a classical LPA (Zhou et al. 2003; Iscen et al. 2019; Zhao et al. 2022). When $h = \infty$, the accuracy of PMLP indeed converges to LASSL, which indicates that PMLP degenerates to classical LPA.

Effect of Neighbor Point Numbers *N'* For a fair comparison with our baseline LASSL, we keep the same choice of $1.5N$ neighbors in LPA, here N represents the categories of the data. It is still a matter of whether more or fewer neighbors will bring better performance. Then we try other choices of the neighbors, N', in Tab. 6 for the ablation study. For CIFAR10 with 40 labeled samples, $N' = 1.5N = 15$ performs best over the 5 choices. The result aligns with our intuition that fewer neighbors lack the information and too many neighbors may burden the LPA, for some may come from the other categories.

Conclusion

We first introduce **Theorem1** to prove that probability density helps to consider the *cluster assumption* comprehensively. Then, we introduce a probability-density-aware measurement, PM. We design a new algorithm with PM, the Probability-Density-Aware Label-Propagation algorithm(PMLP), that propagates pseudo-labels based on clusters' geometric priors. **Theorem2** demonstrates PMLP's exceptional performance, attributing it to integrating density information: LPA relying on traditional distance measurements is a particular case of PMLP. Extensive experiments are designed to show PMLP's superiority.

[2] Appendix: https://arxiv.org/pdf/2412.17547

Acknowledgements

The research is supported by the National Key R&D Program of China (Grant No. 2023YFA1008700 and 2023YFA1008703), the National Natural Science Foundation of China (NO. 62102151), the Open Research Fund of Key Laboratory of Advanced Theory and Application in Statistics and Data Science, Ministry of Education (KLATASDS2305), the Fundamental Research Funds for the Central Universities.

References

Adam Coates, H. L., Andrew Ng. 2011. An analysis of single-layer networks in unsupervised feature learning. In *In Proceedings of the fourteenth international conference on artificial intelligence and statistics, pp. 215–223.*

Azizyan, M.; Singh, A.; and Wasserman, L. 2013. Density-sensitive semisupervised inference.

Bengio, Y.; Lamblin, P.; Popovici, D.; and Larochelle, H. 2006. Greedy layer-wise training of deep networks. In *NeurIPS*.

Berthelot, D.; Carlini, N.; Cubuk, E. D.; Kurakin, A.; Sohn, K.; Zhang, H.; and Raffel, C. 2019. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring.

Bousquet, O.; Chapelle, O.; and Hein, M. 2003. Measure based regularization. In *NeurIPS*.

Chapelle, O.; and Zien, A. 2005. Semi-supervised classif cation by low density separation. In *AISTATS*.

Chen, H.; Tao, R.; Fan, Y.; Wang, Y.; Wang, J.; Schiele, B.; and Savvides, M. 2023. Softmatch: Addressing the quantity-quality trade-off in semi-supervised learning.

Chen, T.; Kornblith, S.; Swersky, K.; Norouzi, M.; and Hinton, G. E. 2020a. Big self-supervised models are strong semi-supervised learners. In *NeurIPS*.

Chen, X.; Chen, W.; Chen, T.; Yuan, Y.; Gong, C.; Chen, K.; and Wang, Z. 2020b. Self-pu: Self-boosted and calibrated positive-unlabeled training. In *ICML*.

Douze, M.; Szlam, A.; Hariharan, B.; and Jégou, H. 2018. Low-shot learning with large-scale diffusion. In *CVPR*.

Elezi, I.; Torcinovich, A.; Vascon, S.; and Pelillo, M. 2018. Transductive label augmentation for improved deep network learning. In *ICPR*.

Guo, L. Z.; and Li, Y. F. 2022. Class-imbalanced semi-supervised learning with adaptive thresholding. In *ICML*.

Hinton, G. E.; Osindero, S.; and Teh, Y. W. 2006. A fast learning algorithm for deep belief nets.

Iscen, A.; Tolias, G.; Avrithis, Y.; and Chum, O. 2019. Label propagation for deep semi-supervised learning. In *CVPR*.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical report.

Laine, S.; and Aila, T. 2016. Temporal ensembling for semi-supervised learning.

Lee, D. H. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML*.

Li, J.; et al. 2021. Comatch: Semi-supervised learning with contrastive graph regularization. In *ICCV*.

Li, S.; Liu, B.; Chen, D.; Chu, Q.; Yuan, L.; and Yu, N. 2020. Density-aware graph for deep semi-supervised visual recognition. In *CVPR*.

Miyato, T.; Maeda, S. I.; Koyama, M.; and Ishii, S. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading digits in natural images with unsupervised feature learning. In *NeurIPS Workshop*.

Nguyen, K. B. 2023. Boosting Semi-Supervised Learning by bridging high and low-confidence prediction. In *ICCV*.

Nguyen, K. B. 2024. Debiasing, calibrating, and improving Semi-supervised Learning performance via simple Ensemble Projector. In *WACV*.

Sajjadi, M.; Javanmardi, M.; and Tasdizen, T. 2016. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *NeurIPS*.

Shi, W.; Gong, Y.; Ding, C.; Tao, Z. M.; and Zheng, N. 2018. Transductive semi-supervised deep learning using min-max features. In *ECCV*.

Sohn, K.; Berthelot, D.; Carlini, N.; Zhang, Z.; Zhang, H.; Raffel, C. A.; Cubuk, E. D.; Kurakin, A.; and Li, C. L. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*.

Tarvainen, A.; and Valpola, H. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*.

Wang, Y.; Chen, H.; Heng, Q.; Hou, W.; Fan, Y.; Wu, Z.; et al. 2022. Freematch: Self-adaptive thresholding for semi-supervised learning.

Xu, N.; Zhang, X.; Wang, W.; Wang, J.; Guo, M.; and Cai, D. 2022. One positive label is sufficient: Single-positive multi-label learning with label enhancement. In *NeurIPS*.

Yang, L.; Zhao, Z.; Qi, L.; Qiao, Y.; Shi, Y.; and Zhao, H. 2023. Shrinking class space for enhanced certainty in semi-supervised learning. In *ICCV*.

Zhao, Z.; Zhou, L.; Wang, L.; Shi, Y.; and Gao, Y. 2022. LASSL: Label-guided self-training for semi-supervised learning. In *AAAI*.

Zheng, M.; You, S.; Huang, L.; Wang, F.; Qian, C.; and Xu, C. 2022. Simmatch: Semi-supervised learning with similarity matching. In *CVPR*.

Zhou, D.; Bousquet, O.; Lal, T.; Weston, J.; and Schölkopf, B. 2003. Learning with local and global consistency. In *NeurIPS*.