PLBR: A Semi-supervised Document Key Information Extraction via Pseudo-labeling Bias Rectification

Pengcheng Guo, Yonghong Song, *Member, IEEE,* Boyu Wang, *Member, IEEE,* Jiaohao Liu, Qi Zhang

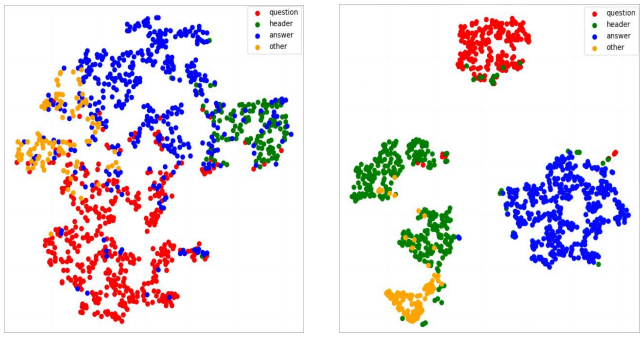
***Abstract*—Document key information extraction (DKIE) methods often require a large number of labeled samples, imposing substantial annotation costs in practical scenarios. Fortunately, pseudo-labeling based semi-supervised learning (PSSL) algorithms provide an effective paradigm to alleviate the reliance on labeled data by leveraging unlabeled data. However, the main challenges for PSSL in DKIE tasks: 1) context dependency of DKIE results in incorrect pseudo-labels. 2) high intra-class variance and low inter-class variation on DKIE. To this end, this paper proposes a similarity matrix Pseudo-Label Bias Rectification (PLBR) semi-supervised method for DKIE tasks, which improves the quality of pseudo-labels on DKIE benchmarks with rare labels. More specifically, the Similarity Matrix Bias Rectification (SMBR) module is proposed to improve the quality of pseudo-labels, which utilizes the contextual information of DKIE data through the analysis of similarity between labeled and unlabeled data. Moreover, a dual branch adaptive alignment (DBAA) mechanism is designed to adaptively align intra-class variance and alleviate inter-class variation on DKIE benchmarks, which is composed of two adaptive alignment ways. One is the intra-class alignment branch, which is designed to adaptively align intra-class variance. The other one is the inter-class alignment branch, which is developed to adaptively alleviate inter-class variance changes on the representation level. Extensive experiment results on two benchmarks demonstrate that PLBR achieves state-of-the-art performance and its performance surpasses the previous SOTA by $2.11\% \sim 2.53\%$, $2.09\% \sim 2.49\%$ F1-score on FUNSD and CORD with rare labeled samples, respectively. Code will be open to the public.**

***Index Terms*—Information extraction, semi-supervised, bias rectification, intra-class variance, inter-class variance.**

I. Introduction

DOCUMENT key information extraction (DKIE) aims at comprehending and extracting interesting information from digital documents like receipts, medical forms, scientific papers, etc. In contrast to the text-only named entity recognition task, visually rich documents in DKIE remain a major challenge since they typically contain a variety of modalities, including images, text and layout.

With the advances in deep learning techniques, extensive efforts have been developed to address this challenge, such as Grid-based methods [1]–[3], GNN-based methods [4]–[6] and transformer-based pre-training methods [7]–[10]. However, these efforts heavily depend on a pre-defined set of entity categories for each dataset, which makes it hard to apply the same DKIE model to different datasets. To address this challenge, some Transformer-based pre-training approaches are developed to learn cross-modal representations with self-attention from large-scale document scenarios. Despite achieving competitive results in DKIE, existing document pre-training models are expensive to collect sufficient high-quality labeled annotation for each entity category. Recently, pseudo-label based semi-supervised learning (PSSL) strategy [11]–[15] has become popular in PSSL since its promising performance in many tasks, such as object detection, image segmentation and data mining.



(a) W/o DBAA (b) W/ DBAA

Fig. 1. T-SNE visualization of PLBR with DBAA on the FUNSD. It shows that entity types have a high intra-class variance and dramatic changes in inter-class variance on DKIE benchmarks from the perspective of cluster structure.

However, existing PSSL methods [16]–[18] do not perform well on DKIE, as two main challenges that differ from traditional image classification tasks (e.g., CIFAR-10 and CIFAR-100). One is that context dependency of DKIE results in incorrect pseudo-labels. For example, the word "apple" may refer to different classes of entities in different tickets. On a digital product store receipt, "apple" may refer to an Apple brand electronic product like an iPhone or an iPad, while on a supermarket receipt, "apple" may refer to a fruit. This variation in meaning requires that the entity extraction model be able to accurately determine the meaning of words based on context and classify them correctly. Unreliable pseudo-labeling under a confidence thresholding mechanism would impair the model's understanding of lexical changes in different contexts of DKIE. The other critical problem is low inter-class variation and high intra-class variation on DKIE benchmarks, as shown

Pengcheng Guo, Yonghong Song, Jiahao Liu and Qi Zhang are with the School of Software Engineering, Xian Jiaotong University, Xian 710049, China. (Email: bedlexmunaxl@stu.xjtu.edu.cn; songyh@mail.xjtu.edu.cn; para15291067561@163.com

Boyu Wang is with the Department of Computer Science and the Brain Mind Institute, University of Western Ontario, London, ON N6A 3K7, Canada, and also with the Vector Institute, Toronto, ON M5G 1M1, Canada. E-mail: bwang@csd.uwo.ca

in Fig. 1. Dramatic change in inter-class variance manifests as different classes of textual information (e.g. invoice number, date, amount) being visually similar. Besides, high intra-class variance manifests as text information of the same class in different documents with different formats, such as date format may be different in different types of documents ("YYYY-MM-DD" or "DD/MM/YYYY", etc.). These challenges complicate DKIE in the SSL settings.

To address the aforementioned issues, this study proposes Pseudo-labeling Bias Rectification (PLBR), a novel PSSL method that improves the quality of pseudo-labels for context-dependent DKIE. Concretely, we propose a Similarity Matrix Bias Rectification (SMBR) module to leverage the context-dependent relationship between the prediction on labeled data and unlabeled data to produce more precisely determined pseudo-labels. Specifically, we explore the relationship between the prediction on labeled data and unlabeled data by building two matrixes, where these two matrixes are derived by computing the similarity between the prediction on labeled data and unlabeled data in the same class of entities. In this way, the model dynamically adapts and learns from complex linguistic structures and context dependencies to improve the quality of pseudo-labeling. Furthermore, we propose a Dual Branch Adaptive Alignment (DBAA) mechanism to adaptively align intra-class variance and enhance inter-class variation in learning cluster structure. In particular, DBAA learns the cluster structure from two perspectives. One is the intra-class alignment branch, which can adaptively compact samples of the same class to align intra-class variance. The other one is the inter-class alignment branch, which can adaptively disperse samples from different classes to enhance inter-class variance. Although simple, our method can significantly improve the performance of PSSL. The main contributions are summarized as follows:

- A simple yet effective PSSL method is developed to improve the quality of pseudo-labels for semi-supervised DKIE. To our best knowledge, this is the first semi-supervised method for DKIE task.
- The similarity matrix bias rectification (SMBR) module is proposed to improve the quality of pseudo-labels by leveraging context-dependent relationships. Moreover, the DBAA mechanism is designed to adaptively align intra-class variance and enhance inter-class variance in DKIE. This two-level alignment way interacts with each other to jointly evolve.
- Extensive experiments on two benchmark datasets demonstrate that our method can obtain new SOTA performance under the SSL setting.

II. Related Work

A. Document Key Information Extraction

Recently, document key information extraction (DKIE) has attracted a wide range of attention since it can reduce labor costs for companies dealing with large business files. DKIE is a significant task that aims to extract interesting information from various formats of PDF and document images such as purchase orders, business insurance, shopping bills, etc. Multi-modal pre-training techniques based on text, layout, and image information have greatly advanced the research of DKIE, and they have become a practical approach for various DKIE tasks. Some transformer-based pre-training approaches have been proven effective for a variety of DKIE tasks. For instance, LayoutLM [7] is a BERT-like transformer method, which is the first layout-aware pre-training model through incorporating the 2D spatial embedding into input for each token. Afterward, LayoutLMv2 [8] improves on LayoutLM by concatenating the visual information and text information into the pre-training stage. Recently, BROS [19] also employs the BERT model as an encoder, with a graph-based classifier based on SPADE [20], which is utilized to predict relationships between entities relations between different tokens for a document image. Although have achieved great progress, these existing pre-training models are limited by the lack of enough high-quality annotations for each entity class. To address this challenge, this work proposes a novel pseudo-labeling based semi-supervised learning method to explore the potential of limited samples.

B. Semi-supervised Learning

The core of semi-supervised learning (SSL) is to boost classification accuracy by exploiting information not only from limited labeled data but also from large amounts of unlabeled data. Existing strategies for SSL fall into three categories, pseudo-labeling [16], [21]–[23], consistency regularization [24]–[27], co-training [26], etc. Recently, pseudo-labeling based semi-supervised learning (PSSL) methods have achieved great progress, which use the model's predictions to explicitly produce a pseudo-labeling for the unlabeled samples. Some works [21], [22] reprocess the "soft" pseudo-labels through a sharpening function to decrease entropy, whereas FixMatch [16] generates "hard" pseudo-labels for unlabeled instances, whose maximum class probability drops over a fixed threshold. Recently, extensive approaches [17], [18], [23], [28] utilize a gradually growing threshold to improve the quality of pseudo-labels. Although improving the quality of pseudo-labels by a predefined or adaptive threshold, these existing threshold-based approaches limit the exploitation of pseudo-labels, which will hamper the generalization performance of the model [28]. In the context-dependent DKIE task, existing general SSL methods fail to effectively exploit the predictive relationship between labeled and unlabeled data to improve the accuracy of pseudo-labeling. Specifically, a fixed threshold would discard these unconfident but correct pseudo-labels, and an adaptive threshold would bring in incorrect pseudo-labels. To address this issue, our method exploits the context-dependent relationship between the prediction on labeled data and unlabeled data to obtain more accurate pseudo-labels. Different from [29] and [30], our method exploits the context-dependent relationship between the prediction on labelled data and unlabeled data to obtain more accurate pseudo-labels.

C. Self-supervised Learning in Semi-supervised Learning

Recently, self-supervised learning learns effective representations from a large number of unlabeled samples on a variety

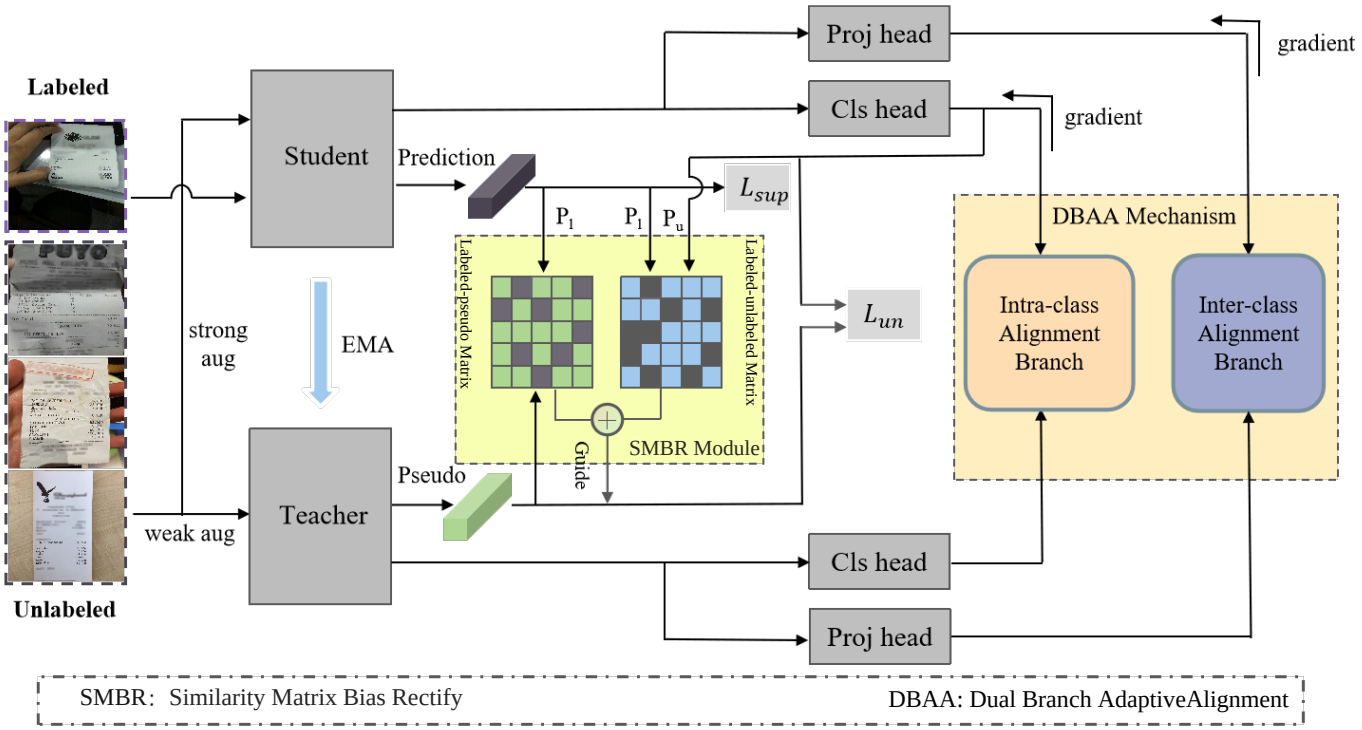


Fig. 2. Framework of the proposed PLBR with similarity matrix bias rectification (SMBR) module and Dual Branch Adaptive Alignment (DBAA) mechanism. Labeled and unlabeled samples from the training data mini-batch. Cls is a classification head and $Proj$ is the projection head. L_{sup} is a supervised classification loss and L_{un} is an unsupervised classification loss.

of tasks [31]–[36]. Contrastive learning has shown great success in self-supervised learning with good performance [32], [33], [35], [37], [38]. Contrastive learning is based on the idea that a good representation can identify the similarities between these different views. In the SSL methods, SimMatch [22] combines consistency regularization and contrastive learning, where consistency regularization is achieved on both semantic-level and instance-level to encourage similarity relationships between different augmented versions on the same data and other instances, which achieves the current state-of-the-art performance on semi-supervised learning. However, it cannot deal with the high intra-class variance and low inter-class variance on DKIE benchmarks. To address these issues, we design two alignment branches. The inter-class alignment branch can adaptively compact samples of the same class to reduce intra-class variance, and the intra-class alignment branch can adaptively disperse samples from different classes to enhance inter-class variance. Different from ConsEEGc [39] aims to improve the accuracy and reliability of Electroencephalogram data clustering by leveraging both labeled and unlabeled data while integrating constraints derived from different sources, our method aims to dynamically align intra-class variance and enhance inter-class variance.

III. Method

A. Problem Description

Let $X = \{(x_b, y_b) : b \in (1, \ldots, B)\}$ be a batch of labeled samples and their corresponding ground-truth annotations. Similarly, let $U = \{u_i : i \in (1, \ldots, B)\}$ represent a batch of unlabeled examples. Since X and U come from the same training set, X and U share the same label space.

The goal of semi-supervised DKIE is to refine entity extraction and sequence labeling accuracy by leveraging both labeled and unlabeled data. First, a shared encoder f is utilized by both the teacher and the student to extract the features. Then, a linear classification head p_{cls} generates the class probability Y. Meanwhile, an MLP projection head g is utilized to generate representation H. Subsequently, for an unlabeled instance, we obtain weakly augmented and strongly augmented representations, denoted as $Aug_w(\cdot)$ and strong augmentations $Aug_s(\cdot)$.

B. Overall Framework

In this section, we introduce the overall framework of our PLBR for semi-supervised DKIE. The complete training algorithm is shown in Algorithm 1. Following [16], [26], we adopt a teacher-student framework [26], where a teacher model is employed to yield pseudo-labels for weakly augmented unlabeled samples and a student model is used to produce predictions for strongly-augmented ones of the same samples. After each training iteration, the teacher model is updated according to an exponential moving average (EMA) [40] of the student model. The proposed PLBR consists of two novel components: a SMBR module and a DBAA mechanism, as shown in Fig. 2.

C. Similarity Matrix Bias Rectification Module

FixMatch [16] produces high-confidence pseudo-labels by using a predefined threshold in PSSL. And later Dash [18] proposed a flexible threshold to obtain a high-quality pseudo-label. However, they limit the exploitation of pseudo-labels

in the context-dependent DKIE benchmarks. On the one hand, a fixed threshold [16] would discard these unconfident but correct pseudo-labels, the flexible threshold [17], [18] would inevitably introduce incorrect pseudo-labels. To address this problem, this study proposes a novel SMBR module to provide more accurate pseudo-labels by efficiently utilizing the context-dependent relationship between predictions on labeled data and unlabeled data, as shown in Fig. 2. To fully explore the relationship between labeled predictions and unlabeled predictions, we build two matrixes under two different augmentations of unlabeled data: labeled-pseudo matrix and labeled-unlabeled matrix. The design of SMBR module follows **Motivation 1** and **Motivation 2**:

Motivation 1: When trained on noisy labels, deep neural networks have been observed to first fit the training data with clean labels during an "early learning" phase, before eventually memorizing the examples with false labels [41]. Compared to pseudo-labels on unlabeled data, predictions on labeled data are clean and reliable in earlier training phases.

Motivation 2: Labeled samples and unlabeled samples both follow the same data distribution and share the same label space [21]. Thus, prior knowledge of labeled predictions is beneficial to reduce the impact of noise on predictions of unlabeled samples. SMBR module constructs two similarity matrixes to fully mine the potential knowledge of the prediction on labeled samples: a labeled pseudo-label matrix and a labeled-unlabeled matrix.

The labeled-pseudo-label matrix aims to encourage pseudo-labels to learn more similar representations in the same class of entities from predictions of labeled samples. For a labeled sample x_i, we employ an encoder f to derive the prediction $p_i = p_{cls}(f(x_i))$. Similarly, for an unlabeled sample u_k, we utilize the same encoder f to obtain the pseudo-label $l_k = argmax(p_{cls}(f(u_k)))$. By collecting predictions $P = \{p_i\}_{i=1}^{B}$ for a batch of labeled samples and pseudo-labels $L = \{l_k\}_{k=1}^{B}$ for a batch of unlabeled samples, we construct the labeled-pseudo-label matrix by computing a similarity matrix W^{lp} between P and L.

$$W^{lp} = P^T \cdot L \tag{1}$$

The labeled-unlabeled matrix is designed to encourage labeled prediction to guide the prediction of unlabeled, which is based on **Motivation 2**. This matrix helps in identifying which unlabeled instances are more likely to share the same class as the labeled entities, facilitating a more accurate pseudo-labeling process. We use the matrix W^{lu} to represent the similarity between labeled prediction and unlabeled prediction in the same class of entities, where the similarity matrix is computed by predictions $P = \{p_i\}_{i=1}^{B}$ for the batch of labeled samples and predictions $U = \{p_{cls}(f(u_j))\}_{j=1}^{B}$ for the batch of unlabeled strong augmentation samples.

$$W^{lu} = P^T \cdot U \tag{2}$$

Then, to adaptability enhance the reliability and comprehensiveness of the blended matrix, we combine matrixes W^{lp} and W^{lu} using Gaussian distributions [42] to mix the controlled formulas.

$$W = \alpha W^{lp} + (1-\alpha) W^{lu} \tag{3}$$

$$\alpha \sim \text{Gamma}(\beta, \beta) \tag{4}$$

where blending factor α is sampled from a Gamma distribution parameterized by the β hyper-parameter, which is used to ensure the validity and appropriateness of the blended similarity matrix W. Then, we employ a standard graph Laplacian to normalize the similarity matrix to avoid the influence of noisy neighbors. Meanwhile, graph Laplacian provides supervision signals to optimize the similarity between two nodes. Specifically, for a symmetric adjacency matrix W with zero diagonal, we can compute the normalized counterpart of W:

$$W^{sys} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \tag{5}$$

where D is the degree matrix of W. Then we use W^{sys} to multiply the original pseudo-label to obtain the guided pseudo-label:

$$L' = L \cdot W^{sys} \tag{6}$$

SMBR module can well improve the quality of pseudo-labels. The experiment results demonstrate that the SMBR module is very important for DKIE to assign high-quality pseudo-labels for corresponding entities in Section IV-D.

D. Dual Branch Adaptive Alignment Mechanism

DBAA aims to adaptively align intra-class variance and enhance inter-class variance in DKIE task. Motivated by contrastive learning with positive samples only [37], we consider using contrastive learning to achieve inter-class separation and intra-class compactness in the classification representation and projection representation. However, the direct combination of comparative learning ignores the high intra-class variance and low inter-class variance on DKIE benchmarks. To address these issues, we design two alignment branches to adaptively align intra-class variance and enhance inter-class variance in the classification representation and projection representation, respectively. In this way, the intra-class alignment branch can adaptively compact samples of the same class to reduce intra-class variance, and the inter-class alignment branch can adaptively disperse samples from different classes to enhance inter-class variance.

1) Intra-class Alignment Branch: To adaptively align intra-class variance in classification representation, this study proposes an intra-class alignment branch, which can adaptively encourage representations from the same category to cluster. More specifically, we obtain the outputs of two views via an encoder $f(\cdot)$ then transform them again with a classification head to yield P_w and P_s that are used for calculating the similarities between them. The subscripts s and w denote strong and weak broadening respectively. With a mini-batch of weak-augmented (Aug_w) (or strong-augmented (Aug_s)) examples, the resulting similarities can identify whether two distinct entities belong to the same class.

$$P_w = p_{cls}(f(Aug_w(u))) \tag{7}$$

$$P_s = p_{cls}(f(Aug_s(u))) \tag{8}$$

where f is a encoder, p_{cls} is a classification head. We utilize the encoder f and the classification head p_{cls} to obtain the high-level representation and the model's prediction.

To compute the similarity-based classifier for the prediction on each weak-augmented (or strong-augmented), we first build a dictionary of the momentum queue $\mathbf{Q} = \{Q_c\}_{c=1}^{C}$ where each key denotes a specific category and Q_c represents a memory queue for class c with the fixed size $|\mathrm{Q_c}|$. In addition, we update Q_c for all c at every step by pushing new predictions from labeled data in the batch and canceling the most old ones when Q_c is full. We compute the per-class similarity of query predictions P_w and P_s to the momentum queue Q:

$$S_w^c = \mathrm{sim}(P_w, \mathbf{Q})/T \tag{9}$$

$$S_s^c = \mathrm{sim}(P_s, \mathbf{Q})/T \tag{10}$$

$$Sim(m, n) = \frac{m \cdot n^T}{\|m\|_2 \|n\|_2} \tag{11}$$

where $c \in C$, T is a temperature parameter (Chen et al. 2020a), $\|\cdot\|_2$ represents the normalize operation. S_w^c and S_s^c denote the similarity of weak augmentation (w) and strong augmentation (s) in certain class.

To adaptively align intra-class variance, we design a weighting coefficient. In this way, a large weight is assigned to positive pairs with similar semantics (e.g., similar appearance, font color, or text size), and a small weight is assigned to positive pairs with less similarity.

$$W_w^c = exp(-sort(S_w^c)/\epsilon) \tag{12}$$

$$W_s^c = exp(-sort(S_s^c)/\epsilon) \tag{13}$$

where ϵ is another temperature parameter. $sort$ is a sorting operation. W_w^c and W_s^c denote the weighting coefficient in the weakly augmentation branch and the strong augmentation branch.

$$L_w = \frac{1}{BC} \sum_{i=1}^{B} \sum_{c=1}^{C} (W_w^{ci} \cdot (1 - S_w^{ci})) \tag{14}$$

$$L_s = \frac{1}{BC} \sum_{i=1}^{B} \sum_{c=1}^{C} (W_s^{ci} \cdot (1 - S_s^{ci})) \tag{15}$$

where the superscript c represents a certain class and $i \in (1, \ldots, B)$ denotes the indexes of the vector.

$$L_{intra} = L_w + L_s \tag{16}$$

To reduce the use of GPU memory and computation time, the gradient only flows back via the output of strong augmented samples Y_j. And gradient is implemented by forwarding strong augmented samples through the EMA model instead of the original model.

2) Inter-class Alignment Branch: To adaptively enhance inter-class variance in projection representation, this study proposes an inter-class alignment branch, which can adaptively facilitate inter-class representations away from each other. It maximizes consistency between different views of the same sample and away from representations of different classes. Concretely, we calculate that the two views are encoded via an encoder $f(\cdot)$ and then transformed again with a projection head $g(\cdot)$ to generate the corresponding representations H_w and H_s. They are employed to maximize consistency between different views of the same sample. With a mini-batch of augmented examples, the representations between a pair of positive examples (augmented from the same document) are given as follows:

$$H_w = g(f(Aug_w(u))) \tag{17}$$

$$H_s = g(f(Aug_s(u))) \tag{18}$$

where f is an encoder, g is a projection head. We use the model f and the projection head g to obtain the high-level representation and the non-linear representation.

Then, we define the similarity between the non-linear representation of weak augmentation and the non-linear representation of strong augmentation.

$$S = Sim(H_w, H_s) \tag{19}$$

To enhance inter-class variance, we build a weighting factor into the calculation of the loss. As a result, a large weight is assigned to positive entities with the same semantic class, and a small weight is assigned to entities of different classes.

$$W^b = exp(-sort(S)/\varepsilon) \tag{20}$$

Then the weighted loss function for a mini-batch of examples is defined as

$$L_{inter} = -\frac{1}{B} \sum_{i=1}^{B} (W^i \log(\frac{\exp(sim(H_w^{pos}, H_s^{pos}))}{\sum_{(H_w, H_s)} \exp(sim(H_w, H_s))})) \tag{21}$$

The optimizer objective of DBAA can be defined as:

$$L_{dbaa} = L_{intra} + L_{inter} \tag{22}$$

E. Overall Loss for Semi-supervision DKIE

Eventually, PLBR jointly optimizes three losses: (1) a supervised classification loss L_{sup} on the labeled sample, (2) an unsupervised classification loss L_{un} on the unlabeled sample, and (3) an optimizer objective of DBAA L_{dbaa} on unlabeled data. Specifically, L_{sup} is defined as the cross-entropy between the ground-truth labels and the model's predictions.

$$L_{total} = L_{sup} + L_{un} + \lambda L_{dbaa} \tag{23}$$

where λ is the balance factor, the details are described in Section IV-E. L_{un} and L_{sup} are then defined as:

$$L_{un} = \frac{1}{B} \sum_{i=1}^{B} \mathrm{Dist}\,(L_i', P_s) \tag{24}$$

TABLE I
EVALUATION RESULTS WITH DIFFERENT NUMBERS OF LABELED TRAINING SAMPLES ON TWO DKIE BENCHMARKS. BOLD/UNDERLINE INDICATES SOTA/THE SECOND BEST. "BASE MODEL" REPRESENTS DKIE MODEL (AS ENCODER IN SEMI-SUPERVISED LEARNING) AND "METHOD" REPRESENTS SEMI-SUPERVISED FRAMEWORK. WE REPORT THE AVERAGE RESULT WITH STANDARD DEVIATION ACROSS 5 RUNS. "FULLY-SUPERVISED" REFERS TO TRAINING WITH ALL TRAINING EXAMPLES.

Base Model	Method	FUNSD						CORD					
		5%			10%			5%			10%		
		Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
LayoutLM [7]	Fully-supervised	76.77±0.75	81.95±0.42	79.27±0.63	76.77±0.75	81.95±0.42	79.27±0.63	94.37±0.64	95.08±0.71	94.72±0.56	94.37±0.64	95.08±0.71	94.72±0.56
	Pseudo-Labeling [43]	43.56±1.71	32.47±1.53	37.13 ±1.35	48.43±1.42	39.40±1.34	43.37±1.76	52.32±1.41	56.86±1.23	54.12±1.28	63.71±1.03	65.34±1.07	64.39±1.16
	Mean Teacher [26]	44.61±2.77	34.57±1.46	38.88±1.31	50.38±1.43	42.32±1.35	45.87±1.21	54.34±1.34	63.17±1.63	58.35±1.28	64.98±1.41	66.45±1.05	65.58±1.38
	MixMatch [21]	46.71±2.77	37.53±1.46	41.48±1.29	52.71±1.84	45.23±1.77	48.62±1.38	58.17±1.41	60.78±1.56	59.37±1.32	67.34±1.25	68.47±1.33	67.77±1.22
	FixMatch [16]	49.49±2.03	43.14±1.74	45.88±1.71	53.91±2.09	49.23±2.27	51.37±2.06	60.19±2.25	62.03±2.08	61.02±1.98	69.98±1.64	70.84±2.01	70.29±1.21
	Dash [18]	51.09±0.97	45.91±0.88	48.24±0.91	54.64±0.75	52.89±0.72	53.67±0.73	63.59±0.78	64.84±0.84	64.17±0.88	70.37±0.76	73.97±0.72	72.03±0.79
	FlexMatch [17]	53.04±0.61	49.86±0.55	51.35±0.58	57.81±0.76	58.12±0.69	57.83±0.77	64.48±0.62	65.88±0.56	64.66±0.82	71.57±0.65	74.69±0.77	73.02±0.71
	SimMatch [22]	53.08±0.75	49.34±1.07	51.04±0.89	58.88±0.63	58.76±0.56	58.78±0.36	65.23±0.65	66.44±0.55	65.69±0.69	72.67±0.77	75.62±0.45	74.03±0.38
	SoftMatch [28]	53.73±0.51	50.04±0.63	51.69±0.61	59.81±0.81	58.88±0.75	58.54±0.78	66.34±0.73	67.53±0.46	66.82±0.75	74.45±0.93	76.92±0.75	75.58±0.92
	FreeMatch [23]	54.11±0.47	51.32±0.73	52.53±0.38	58.97±0.52	59.19±0.43	59.01±0.56	67.85±0.71	68.12±0.38	67.77±0.79	74.93±0.46	77.59±0.72	76.13 ±0.64
	PLBR (Ours)	**56.74±0.37**	**53.58±0.47**	**55.06±0.54**	**61.53±0.53**	**62.67±0.52**	**61.98±0.55**	**69.74±0.58**	**70.54±0.64**	**70.02±0.58**	**76.52±0.48**	**80.78±0.46**	**78.47±0.43**
LayoutLMv2 [8]	Fully-supervised	80.29±0.21	85.39±0.13	82.76±0.14	80.29±0.24	85.39±0.16	82.76±0.11	94.53±0.23	95.39±0.34	94.95±0.26	94.53±0.17	95.39±0.22	94.95±0.18
	Pseudo-Labeling [43]	44.15±2.04	37.87±1.76	40.56 ±2.13	53.67±1.87	44.23±1.66	48.37±1.38	55.43±1.41	63.74±1.74	59.18±1.56	65.89±1.37	69.67±1.58	67.67±1.38
	Mean Teacher [26]	46.74±1.24	44.98±1.35	45.77±1.49	56.48±1.55	54.66±1.47	55.39±1.58	57.43±1.86	65.60±1.42	61.15±1.33	73.89±1.17	72.18±1.51	72.87±1.60
	MixMatch [21]	49.14±1.25	47.56±0.91	48.29±1.43	60.35±1.76	58.11±1.99	59.15±2.08	60.97±2.11	67.69±1.62	64.12±2.28	74.25±1.11	74.17±2.01	74.12±1.81
	FixMatch [16]	51.43±1.78	48.76±1.55	50.01±1.48	63.71±1.55	59.48±1.65	61.45±1.76	61.57±1.72	66.99±1.52	64.07±1.52	75.02±1.35	75.86±1.65	75.35±1.81
	Dash [18]	52.78±0.83	51.53±0.71	52.04±0.53	64.13±0.65	60.43±0.34	62.08±0.48	64.89±0.67	68.91±0.57	66.76±0.29	76.64±0.35	77.04±0.72	76.67±0.80
	FlexMatch [17]	54.82±0.61	55.98±0.51	55.26±0.49	65.16±0.92	62.65±0.69	63.67±0.83	63.61±0.71	70.38±0.87	66.63±0.78	77.35±0.86	77.86±0.67	77.60±0.85
	SimMatch [22]	56.08±1.28	57.65±1.37	56.85±1.61	65.60±0.47	64.86±0.75	65.23±0.62	65.47±0.86	69.92±0.52	67.62±0.14	78.42±0.24	78.59±1.86	78.32±1.08
	SoftMatch [28]	56.48±0.73	59.34±0.78	57.79±0.78	66.21±0.85	67.36±0.63	66.70±0.78	66.59±1.21	73.33±0.46	69.75±0.89	79.36±1.02	79.92±0.34	79.58±0.63
	FreeMatch [23]	56.89±0.34	61.04±0.52	58.83±0.69	66.98±0.61	68.14±0.41	67.45±0.67	67.48±0.37	74.03±0.51	70.48±0.70	80.47±0.45	80.87±0.35	80.61±0.57
	PLBR (Ours)	**59.27±0.32**	**63.62±0.24**	**61.24±0.32**	**69.59±0.43**	**70.68±0.39**	**70.59±0.52**	**70.26±0.37**	**76.14±0.54**	**72.97±0.56**	**81.77±0.43**	**83.16±0.53**	**82.41±0.42**
Bros [19]	Fully supervised	81.16±0.33	85.02±0.32	83.05±0.26	81.16±0.33	85.02±0.32	83.05±0.26	95.58±0.65	95.14±0.54	95.36±0.61	95.58±0.65	95.14±0.54	95.36±0.61
	Pseudo-Labeling [43]	46.09±0.53	39.87±0.87	42.69±0.96	54.75±1.87	46.23±1.60	50.07±1.77	56.48±2.24	64.32±2.08	60.11±2.49	66.84±1.49	70.48±1.97	68.49±1.68
	Mean Teacher [26]	47.53±0.42	45.62±0.98	46.47±0.65	57.64±1.12	56.95±1.31	57.17±1.52	59.91±1.86	66.13±1.36	62.76±1.52	74.64±1.51	73.02±1.30	73.74±1.63
	Mixatch [21]	50.92±0.67	48.34±1.02	49.47±0.99	62.54±1.55	59.34±1.82	60.85±1.53	61.89±2.03	67.25±1.97	64.34±2.23	75.73±1.42	74.52±1.62	75.05±1.92
	FixMatch [16]	54.29±0.58	49.43±0.25	51.68±0.52	64.15±1.27	60.32±1.82	62.12±1.97	63.45±1.30	68.47±1.18	65.76±1.42	76.89±2.32	75.76±2.02	76.24±1.48
	Dash [18]	56.97±0.84	52.84±0.77	54.68±0.53	64.72±0.82	61.48±0.54	63.01±0.63	65.68±0.75	69.46±0.72	67.42±0.49	77.23±0.55	78.57±0.86	77.78±0.77
	FlexMatch [17]	57.53±0.67	56.47±0.88	56.79±0.74	66.21±0.73	63.59±0.65	64.69±0.65	64.63±0.78	71.37±0.61	67.67±0.73	77.89±0.84	78.74±0.57	78.27±0.96
	SimMatch [22]	58.68±0.32	59.37±0.47	58.92±0.43	65.24±0.41	65.37±0.34	64.78±0.43	67.64±0.53	72.65±0.47	69.87±1.11	79.56±0.87	79.85±0.94	79.57±0.85
	SoftMatch [28]	57.95±0.82	65.85±0.44	61.47±0.79	66.67±0.49	68.05±0.37	67.13±0.51	67.32±0.48	74.88±0.20	70.68±0.49	80.05±0.78	80.48±0.45	80.14±0.64
	FreeMatch [23]	58.95±0.37	65.85±0.52	62.13±0.47	67.69±0.37	69.18±0.49	68.35±0.44	68.45±0.58	75.08±0.11	71.45±0.43	81.03±0.78	81.37±0.58	81.11±0.69
	PLBR (Ours)	**60.89±0.18**	**68.23±0.27**	**64.24±0.38**	**72.24±0.15**	**70.97±0.14**	**71.47±0.18**	**71.18±0.24**	**76.33±0.35**	**73.54±0.31**	**83.87±0.46**	**83.67±0.43**	**83.70±0.37**

$$L_{sup} = \frac{1}{B}\sum_{b=1}^{B} \mathrm{H}\left(y_b, p\left(y \mid Aug_w\left(x_b\right)\right)\right) \tag{25}$$

where $H(\cdot)$ is a function to measure the distance between P_s and L'. Followed by [16], Cross Entropy (CE) is adopted as H in our framework by default. P_s and L' denote the prediction of strong augmentation (detailed in Eq. 8) and a guided pseudo-label (detailed in Eq. 6).

IV. EXPERIMENTS

A. Datasets

In our experiments, CORD, FUNSD and XFUNSD are employed to evaluate the performance of the proposed PLBR in DKIE.

CORD [44] contains 1,000 scanned receipts collected from wild scenes. It is released with 30 fields under 4 categories. It includes 800/100/100 receipts for training/validation/testing. FUNSD [45] is a noisy scanned form understanding documents. It consists of 199 receipts, which include 149 training samples and 50 testing samples. On the FUNSD dataset, each semantic entity can be labeled as one of "question", "answer", "header" or "other". As the two benchmarks of the DKIE task, the number of labeled samples in the semi-supervised is rare because the data set is very small. For example, divided by 5%, there are only 7 labeled samples in FUNSD. XFUND includes 7 languages (Chinese, Japanese, Spanish, French, Italian, German, and Portuguese) with 1,393 fully annotated forms. Each language includes 199 forms, where training set includes 149 forms, and testing set contains 50 forms. Entities in these forms are classified into four types: "question", "answer", "header" or "other". XFUNSD includes sub-tasks: semantic entity recognition and relation extraction. Relation extraction aims to predict the relationship between any two predicted semantic entities. Semantic entity recognition refers to the process of extracting semantic entities and classifying them into predefined entity types.

To verify the generalization of the proposed PLBR, we conducted experiments on CV, NLP and Audio tasks. More details about the datasets for these tasks can be found in Appendix B.

B. Implementation Details

Our experiments were carried out with NVIDIA Tesla V100 GPU. To demonstrate the effectiveness of our proposed PSSL framework, we chose three typical DKIE models as base models. For all methods, we apply the Adam optimizer with a learning rate of 2e-5 and a weight decay of 1e-2. We adopt a large batch size of 8 and train the networks for 80 epochs, with a linear learning rate warmup for the first 15 epochs. After the warmup, cosine scheduler is utilized. Following FixMatch, we randomly select 5% and 10% of the training samples as labeled data respectively. We adopt precision, recall, and F1-score as evaluation metrics for FUNSD and CORD. In the augmentation strategy, we employ the Random Insertion [46] strategy as a strong augmentation Aug_s and Synonym Replacement [46] as weak augmentation Aug_w. About parameters, The proposed approach introduces a new

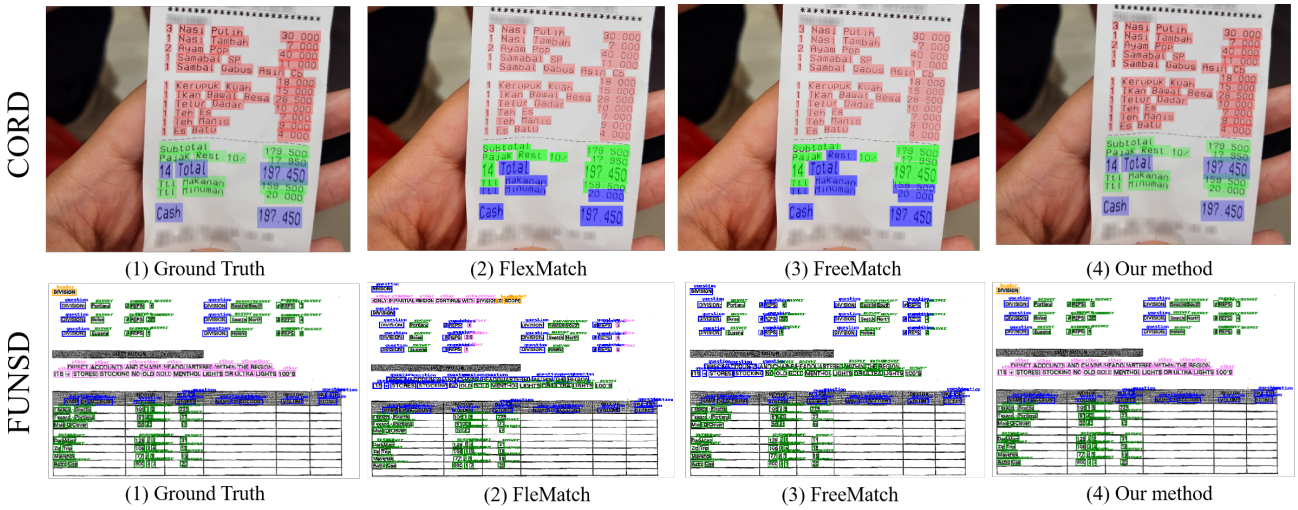


Fig. 3. Examples visualization of PLBR on the CORD and FUNSD datasets. In the *top* row of the figure, entities denoted with Red, Blue, Green for menu, total, and sub-total, respectively. In the *bottom* row of the figure, entities denoted with Blue box, Green box, Yellow box, Pink box, for the question, answer, header, and other, respectively. Best viewed in color.

hyper-parameter: the balance factor λ for the cascade adaptive alignment strategy. In this study, we set $\lambda = 0.2$ and evaluate the influence of parameter settings in Section 4.5. Followed by SimCLR [37], we simply set the two temperature parameters ϵ and T both to 0.05. We also conduct experiments on CV, NLP and Audio tasks, the hyper-parameters used for these tasks are shown in Appendix A.

C. Compared Methods

To evaluate the efficiency of PLBR, we compare it against the following several state-of-the-art SSL approaches:

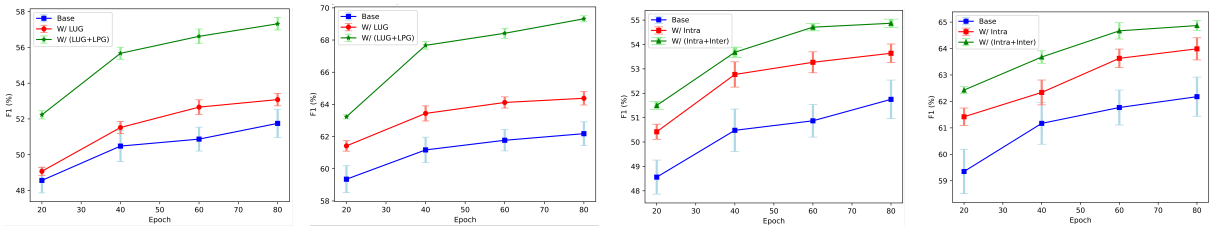
TABLE II
EVALUATION RESULTS WITH 10% LABELED TRAINING SAMPLES ON XFUNSD, WHERE "SER" DENOTES THE SEMANTIC ENTITY RECOGNITION AND "RE" DENOTES THE RELATION EXTRACTION. BOLD/UNDERLINE INDICATES SOTA/THE SECOND BEST. WE USE LAYOUTLMV2 AS BASE MODEL IN ALL SEMI-SUPERVISED LEARNING METHODS AND WE REPORT THE AVERAGE RESULTS FOR ALL LANGUAGES ON XFUNSD WITH STANDARD DEVIATION ACROSS 5 RUNS.

Model	XFUNSD (SRE)	XFUNSD (RE)
Fully-supervised	79.40±0.64	60.33±0.26
Pseudo-Labeling [43]	54.01±0.34	27.70±0.47
Mean Teacher [26]	56.23±0.51	30.14±0.48
Mixatch [21]	59.27±0.29	32.49±0.14
FixMatch [16]	59.69±0.48	33.85±0.74
Dash [18]	60.78±0.46	35.46±0.43
FlexMatch [17]	62.15±0.44	36.37±0.58
SimMatch [22]	62.43±0.43	38.09±0.52
SoftMatch [28]	<u>65.32±0.42</u>	39.92±0.47
FreeMatch [23]	64.78±0.36	<u>41.09±0.42</u>
PLBR (Ours)	**66.56±0.27**	**42.77±0.37**

Pseudo Labeling [43] converts unlabeled data's probabilities into 'one-hot' labels for training. Mean Teacher [26] takes the exponential moving average (EMA) of the neural model as the teacher model. MixMatch [21] operates by producing pseudo-labels for each unlabeled sample and the pseudo-label is sharpened by adjusting temperature scaling to get the final pseudo-label. FixMatch [16] combines consistency regularization and pseudo-labels generated by a fixed threshold. Unlike these methods using fixed threshold to improve pseudo-labels [21], [26], [43], our work proposes a SMBR module to improve the quality of pseudo-labels. Dash [18] algorithm improves the FixMatch by using a progressively increased threshold rather than a fixed threshold, which allows more unlabeled data to participate in the training during the early stage. FlexMatch [17] firstly introduces the class-specific thresholds into SSL by considering the different learning difficulties of different classes. FreeMatch [23] goes further by having the thresholds adjusted in a self-adaptive manner. Distinct from aforementioned methods, our approach not only considers the quality of pseudo-label but also the high intra-class variance and low inter-class variance on DKIE tasks. SimMatch [22] considers both semantic-level and instance-level consistency regularization to encourage similarity relationships between different augmented versions of the same data and other instances. Unlike SimMatch, our method designs a DBAA mechanism to adaptively align intra-class variance and enhance inter-class variance in DKIE. SoftMatch [28] utilizes unconfident but correct pseudo-labels to fit a truncated Gaussian function as the distribution of confidence and proposes uniform alignment to solve the imbalance of pseudo-labels. Different from SoftMatch, our method presents a SMBR module to improve the quality of pseudo-labels.

D. Results

To validate the effectiveness of the proposed approach, we performed experiments on two widely used DKIE benchmark datasets. Table I shows comparative results on the FUNSD and CORD datasets. For the 5% and 10% protocols, our PLBR surpasses all the SOTA methods. From Table I, we can see that the proposed method outperforms other PSSL methods in terms of performance on two benchmarks. In particular, for 5% and 10% FUNSD protocols, our PLBR improves $2.11\% \sim 2.53\%$, $2.97\% \sim 3.14\%$ F1-score over the sub-optimal model FreeMatch in three base models. For 5% and



(a) Study on each component of SMBR on 5% labeled samples (b) Study on each component of SMBR on 10% labeled samples (c) Study on each component of DBAA on 5% labeled samples (d) Study on each component of DBAA on 10% labeled samples

Fig. 4. How PLBR works in FUNSD on 5% and 10% training samples.

Algorithm 1 PLBR algorithm.

Input: $\mathcal{X} = \{(x_b, y_b) : b \in (1, \ldots, B)\}$, $\mathcal{U} = \{u_i : i \in (1, \ldots, B)\}$ a batch of labeled and unlabeled samples. $Aug_w(\cdot)$ and $Aug_s(\cdot)$: weak and strong augmentation function. f: Teacher or student encoder. p_{cls}: classifier. $g(\cdot)$: projection head. λ: balance factor. L_{dbaa}: DBAA loss. H: cross entropy.
while not reach the maximum iteration **do**
  for $c = 1$ to step **do**
    $P_w = p_{cls}(f(\mathrm{Aug_s}(u)))$ $P_s = p_{cls}(f(\mathrm{Aug'_s}(u)))$
    $H_w = g(f(Aug_w(u)))$ $H_s = g(f(Aug_s(u)))$
    Compute labeled-pseudo-label matrix W_{lp} by Eq. (1)
    Compute labeled-unlabeled matrix W_{lu} by Eq. (2)
    Compute the normalized multi-level matrix W^{sys} by Eq. (3), Eq. (4) and Eq. (5)
    Compute the guided pseudo-label L' using matrix W^{sys} by Eq. (6)
    Compute the intra-class weighting coefficient W_w^c of the weakly augmentation branch by Eq. (12)
    Compute the intra-class weighting coefficient W_s^c of the strong augmentation branch by Eq. (13)
    Compute the inter-class weighting coefficient W_b by Eq. (20)
    Compute the inter-class weighted contrastive loss L_r using W_b by Eq. (21)
    Compute the intra-class weighted contrastive loss L_{intra} using W_s^c and W_w^c by Eq. (14), Eq. (15) and Eq. (16)
    $L_{sup} = \frac{1}{B}\sum_{b=m}^{B} H\left(y_m, p\left(y \mid \mathrm{Aug_w}\left(x_m\right)\right)\right)$
    $L_{dbaa} = L_{con}^c + L_r$
    $L_{un} = \frac{1}{B}\sum_{i=1}^{B} H\left(L_i', Y_s\right)$
    $L_{total} = L_{sup} + L_{un} + \lambda L_{dbaa}$
    Optimize f and g by $\mathcal{L}_{total}$
  end for
end while
Return: The well trained model f.

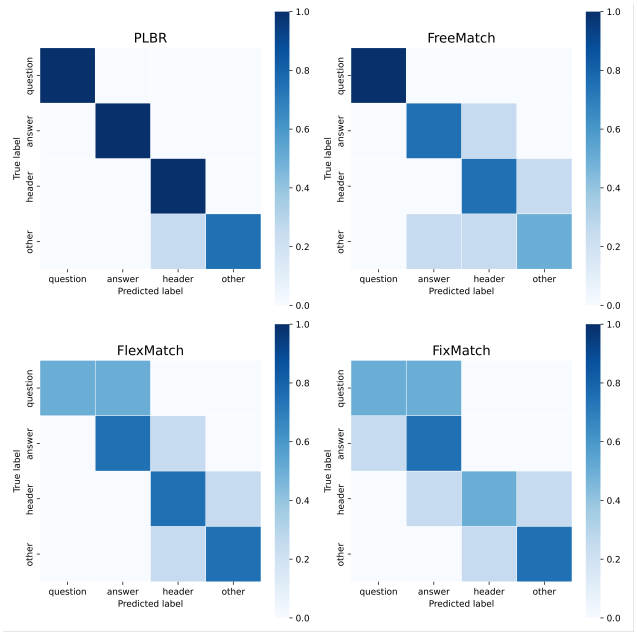
10% CORD protocols, our PLBR surpasses the second best by a margin of 2.09% $\sim$ 2.49%, 1.8% $\sim$ 2.59% in three base models respectively. Furthermore, we can find that the F1-score of compared methods enhances with an increasing number of labeled samples, yet PLBR consistently surpasses other baselines. We also draw the F1-score curves of recent PSSL methods to analyze convergence speed. From Fig. 7, we can observe that PLBR with improved pseudo-labels during early phases helps better guide the learning process and accelerate convergence speed than other existing SSL algorithms such as FreeMatch and SimMatch. Meanwhile, these results demonstrate that is consistent with **Motivation 1** in Section III-C. To validate the effectiveness of DBAA, we provide a qualitative comparison by t-SNE, as in Fig. 1 (right), PLBR with DBAA can cluster most classes better. From Fig. 3, we observe that PLBR can obtain better performance than several excellent PSSL methods such as FreeMatch and FlexMatch, which means PLBR with SMBR can help model produce high-quality pseudo-labels. Moreover, Figure 5 displays the confusion matrix of the PLBR model on the whole test set containing both CORD and FUNSD data. Based on these analyses, we can safely draw the following conclusion that SMBR and DBAA interact with each other to jointly improve performance in DKIE tasks with rare labeled samples.

In addition, we also validate the effectiveness of the proposed approach on a larger dataset like XFUNSD. The results are shown in Table II. We can see that our PLBR surpasses all the SOTA methods. Meanwhile, To verify the impact of additional unlabeled data on the experiments, we explored the interplay between the two datasets by adding the additional dataset XFUNSD on top of the FUNSD dataset, shown in Table XIII in Appendix D. From Table I, Table XIII, Table XIV and Table II, we can see that a significant improvement in model performance by including more unlabeled data for all SSL methods.

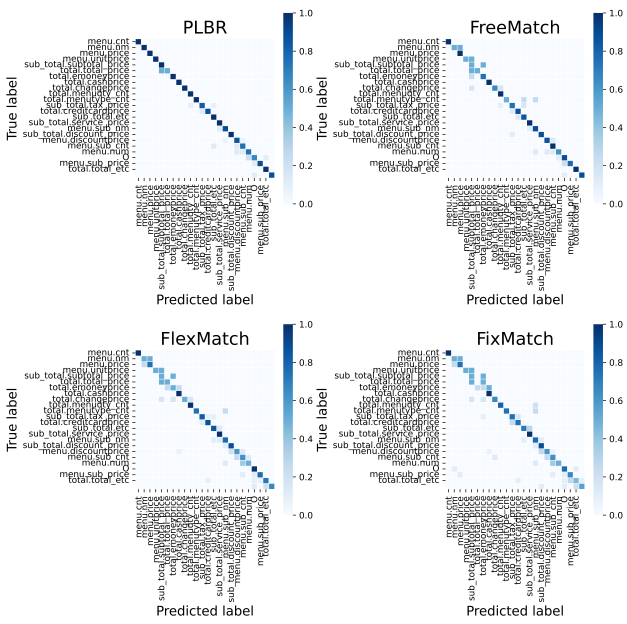
TABLE III
ABLATION STUDY OF PLBR ON 10% LABELED SAMPLES FROM FUNSD AND CORD. "LUG","LPG", "INTRA" AND "INTER" MEAN LABELED-UNLABELED MATRIX, LABELED-PSEUDO-LABEL MATRIX, INTRA-CLASS ALIGNMENT BRANCH AND INTER-CLASS ALIGNMENT BRANCH, RESPECTIVELY. WE USE BROS AS THE ENCODER IN OUR PROPOSED PLBR. THE EVALUATION METRIC IN EXPERIMENTS IS THE F1-SCORE.

n	LUG	LPG	Intra	Inter	FUNSD	CORD
0					62.18	76.32
1	√				64.38	78.17
2	√		√		66.05	79.84
3	√	√			69.61	81.77
4	√	√	√		70.49	82.86
5	√	√	√	√	71.47	83.70

Additionally, to validate the generalization of our proposed PLBR, we also conducted experiments in CV, NLP and audio



(a)



(b)

Fig. 5. (a) Confusion matrix of PLBR on FUNSD on 10% labeled samples. (b) Confusion matrix of PLBR on CORD on 10% labeled samples. As we can see, there are some labels that contain very few examples. Let's replace them with the "neutral" label "O" (which stands for "outside").

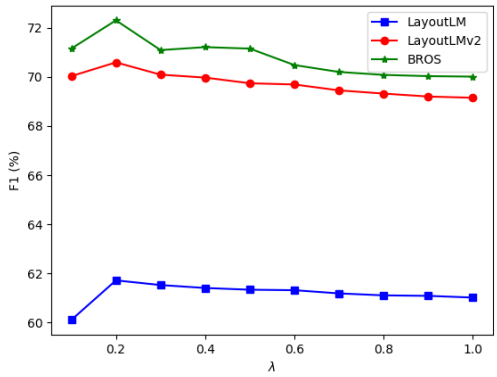


Fig. 6. The weight factor λ with different base models on the FUNSD dataset on 10% labeled training samples.

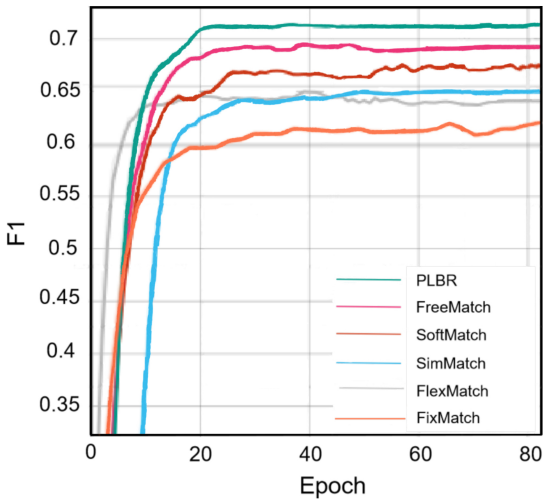


Fig. 7. F1 curves of PLBR on FUNSD with 10% labeled samples, compared to other methods.

TABLE IV
EVALUATION RESULTS ON DIFFERENT BLENDING STRATEGIES BETWEEN W^{lp} AND W^{lu}.

Dataset	No Distribution	Uniform Distribution	Gamma Distribution
CORD	79.45	80.37	81.77
FUNSD	67.52	68.24	69.57

tasks, as shown in Appendix C. From Table X, XI and XII, we can see that PLBR achieves small gains in some metrics, compared to existing general PSSL methods. These results confirm that our method is specifically designed and optimized for the DKIE task.

E. Ablation study

To verify the effectiveness of each component of PLBR (Bros as the base model), we performed extensive ablation studies on FUNSD and CORD in Table III.

Effectiveness of the Similarity Matrix Bias Rectification Module. To validate the effectiveness of the similarity matrix bias rectification (SMBR) module, we performed experiments on FUNSD and CORD and the results can be seen in Table III. From Table III, we can see that the SMBR module can improve performance. Specifically, compared to the base model, PLBR improves F1-score by 7.43% and 5.45% at FUNSD and CORD, respectively. From Fig. 4-(a) and Fig. 4-(b), the results show that the dual matrix is better than the single matrix for both 5% and 10% labeled samples. This indicates that the SMBR module can better employ the reliable prediction on labeled data to reduce the influence of noise and therefore improve the quality of pseudo-labels. In addition, we validate the effectiveness of Gamma distribution in Eq. 4 on Table IV. The results reveal that gamma distribution can create

TABLE V
ABLATION STUDIES OF DIFFERENT β VALUES IN GAMMA DISTRIBUTION ON THE FUNSD AND CORD ON 10% LABELED SAMPLES.

Dataset	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
CORD	78.66	80.21	80.56	81.08	81.31	81.77	80.43	80.02
FUNSD	67.61	67.99	68.76	69.01	69.43	69.57	68.85	68.09

TABLE VI
ABLATION STUDY OF SMBR WITH LABELED PREDICTION AND GROUND TRUTH ON DIFFERENT-SIZED LABELED INSTANCES. WE REPORT THE AVERAGE RESULT WITH STANDARD DEVIATION ACROSS 5 RUNS.

Dataset	FUNSD		CORD	
# Label	5%	10%	5%	10%
Ground truth	54.68±0.21	65.77±0.16	70.78±0.41	78.56±0.45
Labeled prediction (Ours)	57.32±0.12	69.32±0.13	72.08±0.15	81.78±0.25

a more comprehensive representation.

Effectiveness of the Dual Branch Adaptive Alignment Mechanism. To better understand the effectiveness of the Dual Branch Adaptive Alignment (DBAA) mechanism, we conducted ablation studies on the FUNSD and CORD in Table III. From Table III, DBAA mechanism improves F1 scores by 1.86% on the FUNSD and 1.93% on the CORD over the base model. The reason is that DABB mechanism can adaptively align intra-class variance and enhance inter-class variance on DKIE benchmarks. To verify the impact of different components of DBAA, experiments with different adaptive ways are conducted in Fig. 4-(c) and Fig. 4-(d). Intra-class alignment branch has a greater impact than inter-class alignment branch, and this two-level alignment way interacts with each other to jointly evolve.

Study on the selection of weight factor λ and the parameter β of SMBR. We evaluate the influence of the weight factor λ, we conduct experiments on the FUNSD dataset, and the results are presented in Figure 6. The results demonstrate that our proposed PLBR can achieve the best performance in three base models when λ is set to 0.2. From Table V, we can find that SMBR can achieve the best performance when β is 0.6.

F. Additional Analysis of SMBR

We also analyze why labeled prediction is used to guide pseudo-labeled and unlabeled prediction instead of ground truth since labeled prediction has more knowledge and is more robust to noise than hard labeled (ground truth) [47]. We also conducted an ablation study on CORD and FUNSD to verify this idea, as shown in Table VI.

V. CONCLUSION

In this paper, we propose a novel similarity matrix Pseudo-Label Bias Rectification (PLBR) semi-supervised method for DKIE tasks. Specifically, based on two motivations in Section III-C, the similarity matrix bias rectification module (SMBR) is proposed to improve the quality of pseudo-labels on DKIE benchmarks by leveraging context-dependent relationships. Moreover, the dual branch adaptive alignment (DBAA) mechanism is designed to adaptively align intra-class variance and enhance inter-class variance on DKIE benchmarks, which consists of two adaptive alignment ways. Specifically, the intra-class alignment branch is designed to adaptively align intra-class variance and the inter-class alignment branch is developed to adaptively disperse samples from different classes to enhance inter-class variance. This two-level alignment way interacts with each other to jointly evolve. Extensive experimental results and ablation studies have demonstrated that the proposed PLBR surpasses other existing state-of-the-art methods on the two benchmarks. Our findings suggest that the proposed model can obtain high-quality pseudo-labels based on predictions on labeled samples, which provides a good solution for PSSL.

We will explore our approach to practical applications such as legal documents, financial statements, and health insurances. Facing limitations on the availability of extensive labeled datasets due to strict confidentiality norms in these applications, our semi-supervised learning paradigm utilizes available unlabeled documents to enhance the precision and operational efficiency of essential information extraction.

In the future, we will explore a class-missing semi-supervised DKIE model, which aims to identify known and unknown class entities on DKIE.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China under Grant 61973245 and the Key RD Plan of Shaanxi Province (Program No.2023-YBGY-029).

REFERENCES

[1] A. R. Katti, C. Reisswig, C. Guder, S. Brarda, S. Bickel, J. Höhne, and J. B. Faddoul, "Chargrid: Towards understanding 2d documents," *arXiv preprint arXiv:1809.08799*, 2018.
[2] T. I. Denk and C. Reisswig, "Bertgrid: Contextualized embedding for 2d document representation and understanding," *arXiv preprint arXiv:1909.04948*, 2019.
[3] M. Kerroumi, O. Sayem, and A. Shabou, "Visualwordgrid: Information extraction from scanned documents using a multimodal approach," in *International Conference on Document Analysis and Recognition*. Springer, 2021, pp. 389–402.
[4] W. Yu, N. Lu, X. Qi, P. Gong, and R. Xiao, "Pick: processing key information extraction from documents using improved graph learning-convolutional networks," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 4363–4370.
[5] Y. Qian, E. Santus, Z. Jin, J. Guo, and R. Barzilay, "Graphie: A graph-based framework for information extraction," *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019.
[6] G. Tang, L. Xie, L. Jin, J. Wang, J. Chen, Z. Xu, Q. Wang, Y. Wu, and H. Li, "Matchvie: Exploiting match relevancy between entities for visual information extraction," pp. 1039–1045, 2021.
[7] Y. Xu, M. Li, L. Cui, S. Huang, F. Wei, and M. Zhou, "Layoutlm: Pre-training of text and layout for document image understanding," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 1192–1200.
[8] Y. Xu, Y. Xu, T. Lv, L. Cui, F. Wei, G. Wang, Y. Lu, D. Florencio, C. Zhang, W. Che *et al.*, "Layoutlmv2: Multi-modal pre-training for visually-rich document understanding," pp. 2579–2591, 2021.
[9] S. Appalaraju, B. Jasani, B. U. Kota, Y. Xie, and R. Manmatha, "Docformer: End-to-end transformer for document understanding," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 993–1003.
[10] P. Li, J. Gu, J. Kuen, V. I. Morariu, H. Zhao, R. Jain, V. Manjunatha, and H. Liu, "Selfdoc: Self-supervised document representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5652–5660.

[11] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," *Advances in neural information processing systems*, vol. 17, 2004.

[12] H. Pham, Z. Dai, Q. Xie, and Q. V. Le, "Meta pseudo labels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 557–11 568.

[13] W. Shi, Y. Gong, C. Ding, Z. M. Tao, and N. Zheng, "Transductive semi-supervised deep learning using min-max features," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 299–315.

[14] Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le, "Unsupervised data augmentation for consistency training," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6256–6268, 2020.

[15] B. Zoph, G. Ghiasi, T.-Y. Lin, Y. Cui, H. Liu, E. D. Cubuk, and Q. Le, "Rethinking pre-training and self-training," *Advances in neural information processing systems*, vol. 33, pp. 3833–3845, 2020.

[16] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *Advances in neural information processing systems*, vol. 33, pp. 596–608, 2020.

[17] B. Zhang, Y. Wang, W. Hou, H. Wu, J. Wang, M. Okumura, and T. Shinozaki, "Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling," *Advances in Neural Information Processing Systems*, vol. 34, pp. 18 408–18 419, 2021.

[18] Y. Xu, L. Shang, J. Ye, Q. Qian, Y.-F. Li, B. Sun, H. Li, and R. Jin, "Dash: Semi-supervised learning with dynamic thresholding," in *International Conference on Machine Learning*. PMLR, 2021, pp. 11 525–11 536.

[19] T. Hong, D. Kim, M. Ji, W. Hwang, D. Nam, and S. Park, "Bros: A pre-trained language model focusing on text and layout for better key information extraction from documents," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 10, 2022, pp. 10 767–10 775.

[20] W. Hwang, J. Yim, S. Park, S. Yang, and M. Seo, "Spatial dependency parsing for semi-structured document information extraction," in *The Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021)*. Association for Computational Linguistics, 2021.

[21] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," *Advances in neural information processing systems*, vol. 32, 2019.

[22] M. Zheng, S. You, L. Huang, F. Wang, C. Qian, and C. Xu, "Simmatch: Semi-supervised learning with similarity matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 471–14 481.

[23] Y. Wang, H. Chen, Q. Heng, W. Hou, M. Savvides, T. Shinozaki, B. Raj, Z. Wu, and J. Wang, "Freematch: Self-adaptive thresholding for semi-supervised learning," *arXiv preprint arXiv:2205.07246*, 2022.

[24] M. Sajjadi, M. Javanmardi, and T. Tasdizen, "Regularization with stochastic transformations and perturbations for deep semi-supervised learning," *Advances in neural information processing systems*, vol. 29, 2016.

[25] T. Miyato, S.-I. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: A regularization method for supervised and semi-supervised learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1979–1993, 2019.

[26] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," *Advances in neural information processing systems*, vol. 30, 2017.

[27] D. Shi, L. Zhu, J. Li, Z. Cheng, and Z. Liu, "Binary label learning for semi-supervised feature selection," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[28] H. Chen, R. Tao, Y. Fan, Y. Wang, M. Savvides, J. Wang, B. Raj, X. Xie, and B. Schiele, "Softmatch: Addressing the quantity-quality tradeoff in semi-supervised learning," in *Eleventh International Conference on Learning Representations*. OpenReview. net, 2023.

[29] T. Zhang, T. Zhu, J. Li, M. Han, W. Zhou, and S. Y. Philip, "Fairness in semi-supervised learning: Unlabeled data help to reduce discrimination," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 4, pp. 1763–1774, 2020.

[30] F. Zhou, G. Wang, K. Zhang, S. Liu, and T. Zhong, "Semi-supervised anomaly detection via neural process," *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[31] J. Giorgi, O. Nitski, B. Wang, and G. Bader, "Declutr: Deep contrastive learning for unsupervised textual representations," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 879–895.

[32] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8748–8763.

[33] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 297–304.

[34] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," *arXiv preprint arXiv:2003.04297*, 2020.

[35] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9729–9738.

[36] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," *Advances in neural information processing systems*, vol. 33, pp. 21 271–21 284, 2020.

[37] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[38] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 15 750–15 758.

[39] C. Dai, J. Wu, J. J. Monaghan, G. Li, H. Peng, S. I. Becker, and D. McAlpine, "Semi-supervised eeg clustering with multiple constraints," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[40] F. Klinker, "Exponential moving average versus moving exponential average," *Mathematische Semesterberichte*, vol. 1, no. 58, pp. 97–107, 2011.

[41] S. Liu, J. Niles-Weed, N. Razavian, and C. Fernandez-Granda, "Early-learning regularization prevents memorization of noisy labels," *Advances in neural information processing systems*, vol. 33, pp. 20 331–20 342, 2020.

[42] E. W. Stacy, "A generalization of the gamma distribution," *The Annals of mathematical statistics*, pp. 1187–1192, 1962.

[43] D.-H. Lee *et al.*, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on challenges in representation learning, ICML*, vol. 3, no. 2, 2013, p. 896.

[44] S. Park, S. Shin, B. Lee, J. Lee, J. Surh, M. Seo, and H. Lee, "Cord: a consolidated receipt dataset for post-ocr parsing," in *Workshop on Document Intelligence at NeurIPS 2019*, 2019.

[45] G. Jaume, H. K. Ekenel, and J.-P. Thiran, "Funsd: A dataset for form understanding in noisy scanned documents," in *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, vol. 2. IEEE, 2019, pp. 1–6.

[46] J. Wei and K. Zou, "Eda: Easy data augmentation techniques for boosting performance on text classification tasks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 6382–6388.

[47] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[48] Y. Wang, H. Chen, Y. Fan, W. Sun, R. Tao, W. Hou, R. Wang, L. Yang, Z. Zhou, L.-Z. Guo *et al.*, "Usb: A unified semi-supervised learning benchmark for classification," *Advances in Neural Information Processing Systems*, vol. 35, pp. 3938–3961, 2022.

[49] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 702–703.

[50] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, "fairseq: A fast, extensible toolkit for sequence modeling," in *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

[51] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 449–12 460, 2020.

[52] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[53] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.
[54] A. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, 2011, pp. 142–150.
[55] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: understanding rating dimensions with review text," in *Proceedings of the 7th ACM conference on Recommender systems*, 2013, pp. 165–172.
[56] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 1041–1044.

TABLE IX
HYPER-PARAMETERS OF AUDIO TASKS.

Dataset	GTZAN	UrbanSound8k
Sampling Rate	16,000	
Max Length	3.0	4.0
Weight Decay	1e-4	
Model	Wav2Vecv2-Base	HuBERT-Base
Labeled Batch size	8	
Unlabeled Batch size	8	
Learning Rate	2e-5	5e-5
Layer Decay Rate	1.0	0.75
Scheduler	$\eta = \eta_0 \cos(\frac{7\pi k}{16K})$	
Model EMA Momentum	0.0	
Prediction EMA Momentum	0.999	
Weak Augmentation	Random Sub-sample	
Strong Augmentation	Random Sub-sample, Random Gain, Random Pitch, Random Speed	

TABLE VII
HYPER-PARAMETERS OF CLASSIC IMAGE CLASSIFICATION TASKS.

Dataset	CIFAR-10	CIFAR-100
Model	WRN-28-2	
Weight Decay	4e-5	
Labeled Batch size	64	
Unlabeled Batch size	128	
Learning Rate	0.002	
Scheduler	$\eta = \eta_0 \cos(\frac{7\pi k}{16K})$	
Optimizer	Adam	
Model EMA Momentum	0.999	
Prediction EMA Momentum	0.999	
Weak Augmentation	Random Crop, Random Horizontal Flip	
Strong Augmentation	RandAugment [49]	

TABLE VIII
HYPER-PARAMETERS OF NLP TASKS.

Dataset	AG News	IMDb
Model	Bert-Base	
Weight Decay	1e-4	
Labeled Batch size	16	
Unlabeled Batch size	16	
Learning Rate	1e-5	
Scheduler	$\eta = \eta_0 \cos(\frac{7\pi k}{16K})$	
Model EMA Momentum	0.0	
Prediction EMA Momentum	0.999	
Weak Augmentation	None	
Strong Augmentation	Back-Translation [14]	

APPENDIX

A. Experiment Details

To validate the effectiveness of our method on other tasks, we conducted experiments on audio processing (AUDIO), natural language processing (NLP), and computer vision (CV). Note that hyper-parameters setting of these experiments followed by USB [48].

1) Classic Image Classification: The hyper-parameters setting for the classic image classification is shown in Table VII. We use NVIDIA V100 for training classic image classification and Adam as optimizer. For faster training, WRN-28-2 is used for both CIFAR-10 and CIFAR-100. Note that for strong augmentation, we use back-translation similar to [14]. We carry out back-translation offline before training, using EN-DE and EN-RU with models provided in fairseq [50]. All text classification models were trained to employ NVIDIA V100, requiring roughly 20 hours for the entire process.

2) NLP Tasks: For NLP tasks, we random split a validation set from the training set of each dataset used. For IMDb and AG News, we randomly sample 1,000 data and 2,500 data per-class respectively as validation set, and other data is used as training set. The hyper-parameters set- ting for the NLP is shown in Table VIII.

3) Setup for Audio Tasks: For Audio tasks, we adopt Wav2Vec 2.0 [51] and HuBert [52] as the pre-trained model. The batch size of labeled data and unlabeled data is set to 8. We keep the sampling rate of audios as $16,000$. We adopt AdamW optimizer with a weight decay of $5\mathrm{e}{-4}$, and search the learning rate and layer decay as before (detailed in Table IX). Other hyper-parameter settings are the same as NLP tasks. Following RandAugment, we randomly select 2 augmentations from the pool and set their magnitudes randomly for enhanced audio task training.

B. Details of Datasets in PLBR

1) CV Tasks:

a) CIFAR-100: The CIFAR-100 [53] dataset is a natural image (32×32 pixels) recognition dataset consisting of 100 classes. There are 500 training samples and 100 test samples per class.

b) CIFAR-10: The CIFAR-10 contains 60,000 32x32 color images in 10 classes, with 6,000 images per class. The dataset includes 50,000 training images and 10000 test images.

2) NLP Tasks:

a) IMDB: The IMDB [54] dataset is a popular and publicly available datasets used in natural language processing tasks such as sentiment analysis. It consists of 50,000 movie reviews from the internet movie database (IMDB), with an equal number of positive and negative reviews. This dataset is often used for binary sentiment classification tasks where the goal is to predict whether a review expresses a positive or negative sentiment about the movie. In our experiments, we select 12,500 samples for the training dataset and 1,000 samples per class for the validation dataset from the available training samples. The test dataset remain unchanged.

b) Amazon Review: The Amazon Review [55] dataset is a sentiment classification dataset. There are 5 classes (scores). Each class (score) contains 600,000 training samples and 130,000 test samples. We randomly drew 50,000 and 5,000 samples per class from the training set to create the training and validation datasets for our experiments, respectively. The test dataset remains unchanged.

TABLE X
TOP-1 ERROR RATE (%) ON CIFAR-10 AND CIFAR-100 OF 3 DIFFERENT RANDOM SEEDS. THE BEST NUMBER IS IN BOLD.

Dataset	CIFAR-10			CIFAR-100		
# Label	40	250	4,000	400	2,500	10,000
Fully-Supervised	4.62±0.05	4.62±0.05	4.62±0.05	8.44±0.09	8.44±0.09	8.44±0.09
Pseudo-Labeling [43]	74.61±0.26	46.49±2.20	15.08±0.19	87.45±0.85	57.74±0.28	36.55±0.24
MeanTeacher [26]	70.09±1.60	37.46±3.30	8.10±0.21	81.11±1.44	45.17±1.06	31.75±0.23
MixMatch [21]	36.19±6.48	13.63±0.59	6.66±0.26	67.59±0.66	39.76±0.48	27.78±0.29
FixMatch [16]	7.47±0.28	4.86±0.05	4.21±0.08	46.42±0.82	28.03±0.16	22.20±0.12
Dash [18]	8.93±3.11	5.16±0.23	4.36±0.11	44.82±0.96	27.15±0.22	21.88±0.07
FlexMatch [17]	4.97±0.06	4.98±0.09	4.19±0.01	39.94±1.62	26.49±0.20	21.90±0.15
SoftMatch [28]	4.91±0.12	**4.82±0.09**	4.04±0.02	37.10±0.77	26.66±0.25	22.03±0.03
SimMatch [22]	5.60±1.37	4.84±0.39	3.96±0.01	37.81±2.21	**25.07±0.32**	**20.58±0.11**
FreeMatch [23]	4.90±0.04	4.88±0.18	4.10±0.02	37.98±0.42	26.47±0.20	21.68±0.03
PLBR (Ours)	**4.60±0.79**	4.90±0.39	**3.72±0.03**	**37.07±0.79**	26.02±0.13	21.23±0.01

TABLE XI
ERROR RATE (%) OF NLP TASKS WITH VARYING LABELED SET SIZES LABELED INSTANCES. WE REPORT THE AVERAGE RESULT WITH STD ACROSS 5 RUNS.

Dataset	IMDB		AG News	
# Label	20	100	40	200
Fully-Supervised	5.87±0.01	5.84±0.12	5.74±0.30	5.64±0.05
Pseudo-Labeling [43]	26.38±4.04	21.38±1.34	23.86±7.63	12.29±0.40
Mean Teacher [26]	21.27±3.72	14.11±1.77	14.98±1.10	13.23±1.12
MixMatch [21]	-	-	-	-
FixMatch [16]	8.20±0.29	7.36±0.07	22.80±5.18	11.43±0.65
Dash [18]	8.93±1.27	7.97±0.53	19.30±6.73	11.20±1.12
FlexMatch [17]	7.35±0.10	7.80±0.24	16.90±6.76	11.43±0.91
SoftMatch [28]	-	7.48±0.12	**12.68±0.34**	11.34±0.13
SimMatch [22]	**7.24±0.02**	7.44±0.20	14.80±0.57	11.12±0.15
FreeMatch [23]	7.43±0.34	7.47±0.21	13.08±0.48	11.79±0.42
PLBR (Ours)	7.29±0.32	**7.12±0.20**	13.82±0.57	**11.01±0.45**

TABLE XII
ERROR RATE (%) OF AUDIO TASKS WITH VARYING LABELED SET SIZES LABELED INSTANCES. WE REPORT THE AVERAGE RESULT WITH STD ACROSS 5 RUNS. THE BEST NUMBER IS IN BOLD.

Dataset	GTZAN		UrbanSoun8k	
# Label	100	400	250	500
Fully-Supervised	5.98±0.32	5.98±0.32	16.65±1.71	16.61±1.71
Pseudo-Labeling [43]	57.29±2.80	33.93±0.69	42.09±2.41	27.00±1.34
Mean Teacher [26]	51.40±3.48	31.60±1.46	41.70±3.39	28.91±0.93
MixMatch [21]	-	-	-	-
FixMatch [16]	36.04±4.57	22.09±0.65	36.12±4.26	21.43±2.88
Dash [18]	47.00±3.65	23.42±0.83	42.02±5.02	22.26±0.89
FlexMatch [17]	36.93±1.23	22.20±1.39	40.82±5.02	22.26±0.89
SoftMatch [28]	32.86±3.44	21.87±1.08	34.14±7.10	21.47±1.35
SimMatch [22]	32.42±2.18	20.80±0.77	31.70±6.05	**19.55±1.89**
FreeMatch [23]	**31.36±0.37**	20.69±0.59	34.85±6.24	21.11±1.97
PLBR (Ours)	32.29±0.45	**20.51±0.37**	**31.05±5.64**	20.42±1.45

3) Audio Tasks:

a) GTZAN: The GTZAN dataset is collected for music genre classification of 10 classes and 100 audio recordings for each class. The maximum length of the recordings is 30 seconds and the original sampling rate is 22,100 Hz. We split 7,000 samples for training, 1,500 for validation, and 1,500 for testing. All recordings are re-sampled at 16,000 Hz.

b) UrbanSound8k: The UrbanSound8k dataset [56] contains 8,732 labeled sound events of urban sounds of 10 classes, with a maximum length of 4 seconds. The original sampling rate of the audio recordings is 44,100 and we re-sample it to 16,000. It is originally divided into 10 folds, where we use the first 8 folds of 7,079 samples as training sets, and the last two folds as validation sets of size 816 and testing sets of size 837 respectively.

C. Benchmark Results

To validate the effectiveness of our proposed PLBR, we also conduct experiments in CV, NLP and audio tasks.

To ensure fair comparisons of various tasks, we perform experiments in the USB framework[1]. On the other hand, we

[1]We present the full tuning results in: https://github.com/microsoft/Semi-supervised-learning.

TABLE XIII
MULTITASK FINE-TUNING F1-SCORE ON XFUND AND FUNSD DATASET (TRAINING ON ALL LANGUAGES, TESTING ON EACH LANGUAGE OR FUNSD), WHERE "SER" DENOTES THE SEMANTIC ENTITY RECOGNITION AND "RE" DENOTES THE RELATION EXTRACTION. BOLD/UNDERLINE INDICATES SOTA/THE SECOND BEST. AVG DENOTES THE AVERAGE RESULTS FOR SEVEN LANGUAGES ON XFUNSD.

Task	Model	FUNSD	XFUNSD(Avg)
SER	Fully-supervised	79.59±0.48	83.74±0.22
	Pseudo-Labeling [43]	50.38±0.53	56.33±0.26
	Mean Teacher [26]	56.79±1.01	57.31±0.34
	Mixatch [21]	60.52±0.63	60.56±0.20
	FixMatch [16]	62.67±0.74	61.81±0.31
	Dash [18]	63.56±0.72	62.95±0.35
	FlexMatch [17]	65.19±0.58	64.01±0.37
	SimMatch [22]	66.41±0.28	64.69±0.28
	SoftMatch [28]	67.38±0.42	66.08±0.30
	FreeMatch [23]	68.15±0.76	65.39±0.21
	PLBR (Ours)	**71.46±0.67**	**68.08±0.27**
RE	Fully-supervised	53.75±0.45	62.61±0.21
	Pseudo-Labeling [43]	29.25±0.19	32.84±0.36
	Mean Teacher [26]	31.18±1.66	33.30±0.37
	Mixatch [21]	31.89±0.81	33.92±0.03
	FixMatch [16]	32.54±0.17	36.45±0.23
	Dash [18]	34.73±0.35	37.19±0.33
	FlexMatch [17]	35.87±0.45	39.23±0.26
	SimMatch [22]	37.08±0.38	40.02±0.29
	SoftMatch [28]	39.67±0.19	42.43±0.30
	FreeMatch [23]	40.85±0.73	43.40±0.13
	PLBR (Ours)	**43.67±0.25**	**44.36±0.22**

adopt the best model based on validation datasets which were then assessed on test datasets. Additionally, the error rate is employed as an evaluation metric in various tasks.

The experimental setup is detailed in Appendix A. Note that 'fully-supervised' refers to training using all data with full annotations in our reported results.

TABLE XIV
EVALUATION RESULTS WITH 10% LABELED TRAINING SAMPLES ON FUNSD, WHERE "RE" DENOTES THE RELATION EXTRACTION. BOLD/UNDERLINE INDICATES SOTA/THE SECOND BEST. WE USE LAYOUTLMV2 AS BASE MODEL IN ALL SEMI-SUPERVISED LEARNING METHODS AND WE REPORT THE AVERAGE RESULTS ON FUNSD WITH STANDARD DEVIATION ACROSS 5 RUNS.

Model	FUNSD (RE)
Fully-supervised	51.87±0.45
Pseudo-Labeling [43]	28.24±0.38
Mean Teacher [26]	30.21±0.58
Mixatch [21]	30.06±0.28
FixMatch [16]	31.43±0.52
Dash [18]	32.87±0.36
FlexMatch [17]	33.62±0.48
SimMatch [22]	35.09±0.36
SoftMatch [28]	36.99±0.72
FreeMatch [23]	38.16±0.57
PLBR (Ours)	**41.88±0.59**

D. Addition Experiments

We also extend experiments about practical applications, which explore using extra unlabeled data such as XFUNSD into FUNSD. The results described in Table XIII. In order to validate the performance of our approach for relational extraction on FUNSD, we have added related experiments on Table XIV.