EMCRL: EM-Enhanced Negative Sampling Strategy for Contrastive Representation Learning

Kun Zhang, *Member, IEEE*, Guangyi Lv, Le Wu, *Member, IEEE*, Richang Hong, *Member, IEEE*, and Meng Wang, *Fellow, IEEE*

***Abstract*—As one representative framework of self-supervised learning (SSL), contrastive learning (CL) has drawn enormous attention in the representation learning area. By pulling together a "positive" example and an anchor, as well as pushing away many "negative" examples from the anchor, CL is able to generate high-quality representations for the data of different modalities. Therefore, the qualities of selected positive and negative examples are critical for the performance of CL-based models. However, due to the assumption of label unavailability, most existing work follows the paradigm of contrastive instance discrimination, which treats each input instance as an individual category. Therefore, they focused more on positive example generation and designed plenty of data augmentation strategies. For negative examples, they just leverage the in-batch negative sampling strategy. We argue that this negative sampling strategy will easily select false negatives and inhibit the capability of CL, which we also believe is one of the reasons why a large size of negatives is needed in CL. Apart from using annotated labels, we try to tackle this problem in an unsupervised manner. We propose to integrate expectation maximization (EM) into the selection of negative examples and develop a novel *EM-enhanced negative sampling strategy* (*EMCRL*) to distinguish false negatives from true ones for CL performance improvement. Specifically, *EMCRL* employs EM to estimate the distribution of ground-truth relations between each sample and corresponding in-batch negatives and then optimizes model parameters with the estimations. Considering the sensitivity of EM algorithm to the parameter initialization, we propose to add a random flip into the distribution estimation to enhance the robustness of the learning process. Extensive experiments over several advanced models on sentence representation and image representation tasks demonstrate the effectiveness of *EMCRL*. Our method is easy to implement, and the code is publicly available at https://github.com/zhangkunzk/EMCRL_pytorch.**

***Index Terms*—Contrastive learning (CL), expectation maximization (EM), negative examples, representation learning.**

Received 7 March 2024; revised 7 August 2024; accepted 27 August 2024. Date of publication 4 October 2024; date of current version 2 June 2025. This work was supported in part by National Science and Technology Major Project under Grant 2023ZD0121103, in part by the National Natural Science Foundation of China under Grant 62376086 and Grant U23B2031, and in part by the joint Funds of the National Natural Science Foundation of China under Grant U22A2094. *(Kun Zhang and Guangyi Lv contributed equally to this work.) (Corresponding author: Kun Zhang.)*

Kun Zhang, Le Wu, Richang Hong, and Meng Wang are with the School of Computer and Information, Hefei University of Technology, Hefei, Anhui 230029, China (e-mail: zhang1028kun@gmail.com; lewu.ustc@gmail.com; eric.mengwang@gmail.com; hongrc@hfut.edu.cn).

Guangyi Lv is with AI Laboratory, Lenovo Research, Beijing 100094, China (e-mail: lvgy1@lenovo.com).

Digital Object Identifier 10.1109/TCSS.2024.3454056

I. Introduction

CONTRASTIVE learning (CL) aims at learning a robust representation of input data in an unsupervised manner and has achieved impressive progress in natural language processing [1], [2] and computer vision [3], [4]. By treating each input as an anchor, and pulling together a "positive" sample and the anchor in embedding space as well as pushing apart many "negatives" from the anchor [5], CL is capable of reducing the performance gap with the supervised counterparts [6]. Plenty of frameworks, such as MoCo [7], SimCLR [8], BYOL [9], and SimCSE [10], have been proposed to unlock the potential of CL. Recent large language models (LLMs) progress also proves that CL is helpful for performance improvements, such as hallucination reduction [11], few-shot learning improvement [12], [13], and logical reasoning [14], [15]. According to the core idea, the sampling qualities of positives and negatives are critical for CL performance [16], [17], [18]. For positive sampling, data augmentations are developed to help models to learn the invariant features, such as crop, resize, and rotate for images [8], and synonym replacement, random insertion, and random swap for sentences [19]. Recently, the dropout operation has been verified effective for different data, such as sentences [10], images [20], and graphs [21].

As for negative sampling, in-batch negative sampling is the most commonly used strategy, which treats samples in the same batch other than the anchor as negative examples. Though simple and useful, *this strategy will easily select false negatives and is too strong to obtain in-class invariant features* [16]. Moreover, in-batch negative sampling will import sampling bias and hurt model performance [22], [23], [24]. As reported in Table I, in-batch negative sampling strategy will easily select false negatives. We can observe that the smaller the dataset categories, the more severe the false negative problem is (e.g., 50.4% for SciTail dataset with two categories and 19.3% for STS dataset with five categories). Meanwhile, we can observe that the smaller the batch size, the more severe the false negative problem is (87.1% for batch size $= 32$ and 43.2% for batch size $= 512$ on SciTail dataset). When pushing apart these negatives from the anchor example, it will easily cause information loss and hurt the quality of learned representations. All these phenomena demonstrate that the selection strategy of negative samples has much space to be further improved.

To overcome the shortcomings of in-batch negative sampling, various improvements are proposed. For example,

TABLE I
FALSE NEGATIVE PROBABILITY (%) OF IN-BATCH SAMPLING BATCH WITH DIFFERENT BATCH SIZES FROM DIFFERENT DATASETS

Dataset	Batch Size				
	32	64	128	256	512
SciTail (two classes)	0.871	0.714	0.504	0.667	0.432
SNLI (three classes)	0.452	0.397	0.346	0.317	0.337
SST (five classes)	0.355	0.230	0.193	0.201	0.194

Note: We repeat the random sampling five times and report the average score.

Chen et al. [8] pointed out that a larger batch size will help models to obtain more useful negatives, which is consistent with our finding in Table I. Khosla et al. [5] used annotated labels to recognize false negatives and proposed a supervised CL framework. Though simple and effective, these methods require additional computational resources or human resources (i.e., extra annotations for negatives), which are still far from satisfactory. Meanwhile, some unsupervised strategies are proposed. For example, Kalantidis et al. [25] selected K negatives with top-K similarity and employed MixUp operation to construct hard negatives. Zhang et al. [26] employed cluster methods to generate pseudolabels for the samples in one batch. Those with the same pseudolabel will be treated as positive examples. Despite the great progress, these methods usually employ specific assumptions and concrete implementations, and sometimes assumptions may even conflict with each other. For example, Kalantidis et al. [25] assumed that these negative samples with high similarity are hard negatives. In contrast, Zhou et al. [22] argued that those are more like false negatives. Therefore, *how to improve the quality of negative examples in CL at a relatively low cost with relaxed assumptions is very promising for CL performance improvement*, which is also the focus of our article.

Fortunately, inspired by the work [22], we conduct an experiment to explore the similarity distributions between the anchor example and the negatives, as well as corresponding factors. We select pretrained BERT [27] model and fine-tuned unsupervised SimCSE [10] to compare the generated similarity distributions. Fig. 1(a) reports the ground-truth relation distribution of one mini-batch data from SST datasets. We then randomly select one example as the anchor and calculate the cosine similarity between the anchor and the rest in-batch negatives. According to Fig. 1(b), we can observe that pretrained BERT suffers from the anisotropy problem [23] and tends to generate sentence representations in a small space, resulting in more concentrated and less distinguishable similarity scores (i.e., most similarity scores are between $[0.6, 0.9]$). As a comparison, unsupervised SimCSE uses CL to improve the quality of generated sentence representations. Therefore, we can obtain more diverse similarity scores from Fig. 1(b), indicating that the generated sentence representations have better discrimination. Moreover, we can observe that both models are cautious in generating extremely high-similarity scores (i.e., larger than 0.8). More powerful models are more reliable. Therefore, we can summarize that the quality of using similarity score to reveal the semantic relations

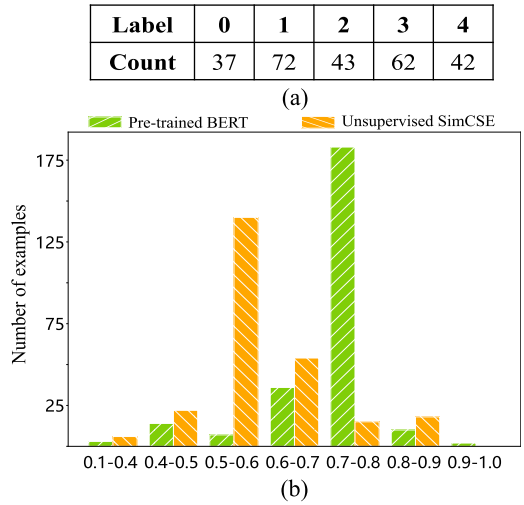


Fig. 1. Similarity distributions between one randomly sampled anchor example (labels: "1") and the rest in-batch negatives. The batch size is 256. (a) Ground-truth relation distribution. (b) Similarity scores from different models (i.e., pretrained BERT and unsupervised SimCSE).

is highly related to the model capability, which is intuitive but largely neglected.

Based on the observation, the problem turns to "How to use similarity score to identify the false negative automatically and improve the quality of negative examples in CL?" To this end, we propose to incorporate the expectation-maximization (EM) algorithm [28] and develop a novel *EM-enhanced negative sampling strategy* (*EMCRL*) to make better negative sampling. Specifically, *EMCRL* consists of two components: *representation learning* and *EM-enhanced optimization*. For the former, a commonly used CL framework is employed to generate representations for inputs, where data augmentation, encoder, and projection are used. For the latter, *EMCRL* focuses on in-batch negative sampling and treats the ground-truth similarity distribution between in-batch negatives and the anchor as a latent variable. Then, it uses the output of the current learning model to estimate the distribution, aiming to distinguish false negatives from true ones (E-step). Next, the estimation results are used to update the model parameters by minimizing the contrastive loss (M-step). One step further, since EM algorithm highly depends on the initialization, we propose to add an additional random flip when estimating the distribution of ground-truth relations for model robustness improvement. We have to note that *EMCRL* is model-agnostic, simple to implement, and stable to train. Empirical experiments over sentence representation and image representation tasks demonstrate the superiority and effectiveness of *EMCRL*. Moreover, the ablation study has demonstrated the effectiveness of our proposed random flip operation.

In summary, the main contributions of this article lie in the following parts.

1) We argue that negative sampling in CL should take the model ability into consideration.
2) We propose to employ EM algorithm and develop a simple but effective *EMCRL* to make better negative sampling for CL performance improvement.

3) We conduct extensive experiments on multiple representation learning models over different tasks to demonstrate the superiority and effectiveness of *EMCRL*.

The remainder of this article is organized as follows. Section II summarizes related work. Section III reports technical details of our proposed *EMCRL*. The experiments and detailed analysis are reported in Section IV. Finally, we discuss and conclude our work in Sections V and VI.

II. Related Work

Our work draws on the existing literature in CL. We mainly summarize the *positive generation* and *negative sampling* in CL.

A. Positive Generation in CL

Data augmentations are the most commonly used positive sampling methods. For example, Chen et al. [8] proposed a simple but representative framework: *SimCLR*, to achieve the contrastive between different examples, in which crop, resize, and rotate options are utilized to generate different views for a single image. Wang et al. [29] treated facial examples with the same emotion from the same group as the positives. You et al. [21] employed node dropping, edge perturbation, attribute masking, and subgraph to generate positives for graph data. For natural language, Wei and Zou [19] employed four simple but powerful operations: synonym replacement, random insertion, random swap, and random deletion to do sentence augmentations. To maintain the consistency of semantics, Yan et al. [30] used adversarial attack, token shuffling, cutoff, and dropout at the embedding matrix to generate positive examples, which will ensure maximum semantic invariance. Moreover, Gao et al. [10] demonstrated that dropout operation in PLMs can guarantee semantic invariance while efficiently generating positive samples. They sent one sample into PLMs twice to generate positive pairs and achieve impressive performance on sentence embedding. Further, Wu et al. [31] pointed out that positive sentences generated by dropout always have the same length, which is an unnecessary bias. They incorporated word repetition to improve the length diversity of positives. In summary, positive examples are used to describe input data from different views so that invariant features are obtained by models for better representation learning.

B. Negative Sampling in CL

In-batch negative sampling is the most commonly used strategy that treats the instances in the same batch other than the anchor as negatives. This type of contrastive is also named contrastive instance discrimination [16]. Though simple and useful, *this strategy will easily select false negatives and is too strong to obtain in-class invariant features* [16]. To overcome this problem and improve the quality of negatives, plenty of methods have been proposed.

One straightforward direction is to increase the batch size or use supervised signals. As revealed in Table I, a larger batch size can reduce the probability of false negatives effectively. Therefore, MoCo and its variants [7], [32] used momentum contrast and update to increase the number of available negative representations with low cost. Meanwhile, supervised signals can tackle this problem effectively, a large number of supervised sampling strategies are proposed to leverage existing labels to identify false negatives from in-batch negatives and then treat them as positive ones [5], [10], [33]. Though simple and useful, these methods require additional computational resources or human resources, which are not the optimal methods for improving the quality of negative examples.

Another direction is to use unsupervised methods to conduct negative sampling, which can be grouped into *false negative identification* and *better negative sampling*. For the former, Huynh et al. [34] used data augmentations to generate support views for the anchor, and negative examples that have top-k similarity with support views are treated as false negatives. Wang et al. [35] incorporated cluster methods to constrain the inherent nature of data when discriminating different examples. For the latter, Kalantidis et al. [25] selected top-K negatives based on the similarity with the anchor and employed MixUp operation to construct hard negatives. *Zhou* et al. [22] argued that in-batch negative sampling will import sampling bias and hurt model performance. They developed a noise-based negative sampling method and used learned instance weight to improve the quality of negative samples. Moreover, Robinson et al. [36] argued that hard negative examples should be paid more attention. They designed a new sampling strategy and verified its effectiveness on SimCLR [8]. Besides, there are other methods to improve the quality of negatives, such as conditional negative sampling [37], debiased contrastive loss construction [38], soft label generation [39], and curriculum learning-based sampling [40].

Our Distinction: Despite the achieved progress, most existing CL-based learning methods either rely on strong assumptions or require elaborate pseudolabel design, limiting the flexibility and generalization of CL. In contrast, by conducting a similarity distribution analysis, we observe that when considering the confidence of similarity scores between the anchor and the negatives, similarity scores can be used to improve the quality of in-batch negative sampling in CL. Thus, we propose adding EM algorithm to the optimization of CL. Thus, not only the inherent features of inputs but also the current performance of models are considered for negative sampling. Moreover, our proposed *EMCRL* is model-agnostic, simple to implement, and stable to train.

III. *EM-Enhanced Negative Sampling Strategy (EMCRL)*

Fig. 2 illustrates the overall architecture of our proposed *EMCRL*, including *representation learning* and *EM-enhanced optimization* components. For the former, we use a symmetric CL structure to generate sentence representation. For the latter, we develop a novel *EMCRL* strategy to boost the CL performance by incorporating EM algorithm. We also summarize the

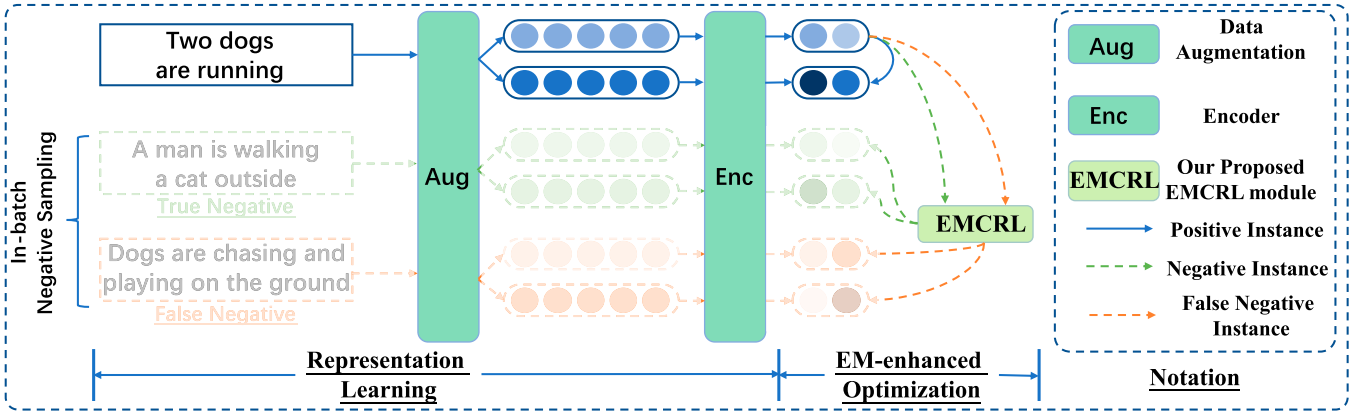


Fig. 2. Overall architecture of *EMCRL*. For simplicity, projection layer *Proj(·)* is not included.

TABLE II
NOTATIONS AND EXPLANATIONS IN *EMCRL*

Notation	Explanation
$X_i = (x_{i1}, x_{i2})$	The ith input pair
$(\boldsymbol{h}_1, \boldsymbol{h}_2)$	Two learned representations of the input
$(\boldsymbol{z}_1, \boldsymbol{z}_2)$	Two projections used for similarity calculation
A_i^+, A_i^-	Positive set and negative set for the ith input
τ	Temperature used in loss function of CL
$R = r \in \{0, 1\}$	The relation of input pair
$\boldsymbol{E}$	The ground-truth relations of input pairs
$\boldsymbol{O}_t$	The tth predicted relations of input pairs
$\bar{\boldsymbol{O}}$	The random flip of the relations of input pairs

necessary notations in Table II for better illustration. Next, we will introduce the technical details of each component.

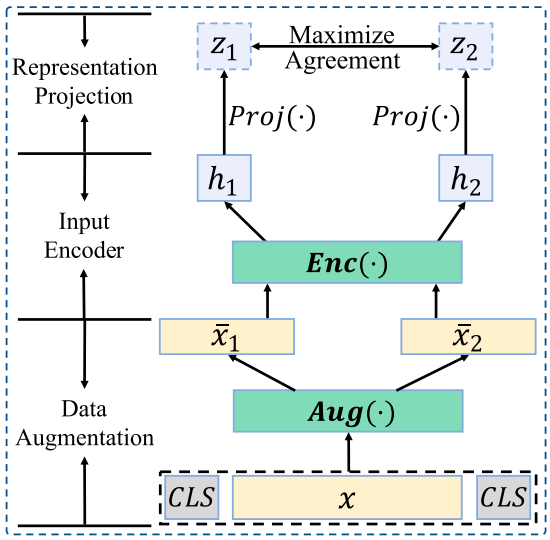


Fig. 3. Common structure of CL framework, which is also the representation learning framework of our proposed *EMCRL*.

A. Representation Learning

Since *EMCRL* focuses on negative sampling in CL, we first introduce CL-based representation learning framework, including *data augmentation*, *encoder*, *projection*, and *optimization*. Fig. 3 illustrates the common structure. Note that *projection* is not included in Fig. 2 for simplicity.

1) Data Augmentation: Let x denote the input sample. We leverage data augmentations $\text{Aug}(\cdot)$ to generate two different views of x, formulating as $\bar{x}_1 = \text{Aug}(\boldsymbol{x}), \bar{x}_2 = \text{Aug}(\boldsymbol{x})$. For image data, crop, resize, and rotate options are common augmentations. For sentences, synonym replacement, random insertion, and random swap are usually used. Meanwhile, dropout operation is also one commonly used augmentation.

2) Input Encoder: Next, we employ encoder $\text{Enc}(\cdot)$ to map augmentations into representation space, representing by $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$. For example, pretrained language models (PLMs), such as BERT [27], RoBERTa [41], and GPT-2 [42], can be selected as the backbone encoder. Moreover, inspired by [5], we also add a normalize operation on $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$ to obtain better representations. This process can be formulated as follows:

$$\boldsymbol{h}_1 = \text{Norm}(\text{Enc}(\bar{x}_1)), \quad \boldsymbol{h}_2 = \text{Norm}(\text{Enc}(\bar{x}_2)). \tag{1}$$

Note that *EMCRL* is model-agnostic. Experimental results reported in Section IV-C also show the effectiveness of *EMCRL* over different encoders.

3) Representation Projection: To calculate distances among anchor, positive, and negative examples, we further project representations into the same space with a projection function $\text{Proj}(\cdot)$. We select MLP with a nonlinear activation function to realize the projection function as: $\boldsymbol{z}_1 = \sigma(\text{MLP}(\boldsymbol{h}_1)), \boldsymbol{z}_2 = \sigma(\text{MLP}(\boldsymbol{h}_2))$. Usually, $\{\boldsymbol{z}_1, \boldsymbol{z}_2\}$ are used for contrastive loss. $\{\boldsymbol{h}_1, \boldsymbol{h}_2\}$ are treated as the learned representations and used for downstream tasks.

4) Optimization: To achieve the CL goal, the following contrastive loss is usually used:

$$L = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{f(\boldsymbol{z}_{i1}, \boldsymbol{z}_{i2})}{f(\boldsymbol{z}_{i1}, \boldsymbol{z}_{i2}) + \sum_{k \in A_i} f(\boldsymbol{z}_{i1}, \boldsymbol{z}_{k1})}$$
$$f(\boldsymbol{z}_i, \boldsymbol{z}_j) = \exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau) \tag{2}$$

where $\text{sim}(\cdot)$ is the similarity function, which can be realized by cosine similarity. A_i is the index set of negative examples for the ith anchor example. $\{\boldsymbol{z}_{i1}, \boldsymbol{z}_{i2}\}$ are augmentations of the ith negative example. τ is the temperature to control the sensitivity of models to negative examples. N is the batch size for in-batch negative sampling.

As discussed before, the quality of negative examples has a big impact on the CL performance. Therefore, we divide A_i into positive set A_i^+ and negative set A_i^- and reformulate (2) to better describe the ideal optimization target:

$$\hat{L} = -\frac{1}{N}\sum_{i=1}^{N}\frac{f(\boldsymbol{z}_{i1}, \boldsymbol{z}_{i2}) + \sum_{j\in A_i^+} f(\boldsymbol{z}_{i1}, \boldsymbol{z}_{j2})}{f(\boldsymbol{z}_{i1}, \boldsymbol{z}_{i2}) + \sum_{k\in A_i^+\cup A_i^-} f(\boldsymbol{z}_{i1}, \boldsymbol{z}_{k1})}. \quad (3)$$

To this end, the problem turns *reconstructing A_i^+ and A_i^- based on in-batch negative sampling*, which is very challenging due to the lack of supervised signals. In response, we introduce EM algorithm into the optimization and propose a novel *EMCRL* strategy to tackle this problem, which we will give a detailed introduction in the following section.

B. EM-Enhanced Optimization

As reported in Fig. 1, similarity scores can be used to help boost the learning process of CL-based models by identifying the false negatives. Meanwhile, a powerful model will generate similarity scores with high confidence, which will improve the quality of similarity-based false negative example identification. In other words, this is a dynamically interactive learning process. Thus, we can assume that there exists a latent variable Z, describing the ground-truth similarity distributions among in-batch negatives and the anchor. Since Z is unobserved under the unsupervised setting, we propose to employ the EM algorithm and design a novel *EMCRL* strategy to approximate this distribution (i.e., approximating A_i^+ and A_i^-).

First of all, we leverage the following equation to denote the output of the learning model for the ith input:

$$M(X_i, \boldsymbol{\theta}) = P(R = r|X_i, \boldsymbol{\theta}), \quad X_i = (x_{i1}, x_{i2}) \quad (4)$$

where $X_i = (x_{i1}, x_{i2})$ denotes the ith input sentence pair. $R = r$ represents the predicted relation of (x_{i1}, x_{i2}).

With this notation, we can revisit CL optimization target (3), which focuses on learning precise sentence representations and distinguishing semantic relations among different sentences accurately. Specifically, the optimization target is to obtain the best $\boldsymbol{\theta}$ for maximizing the joint probability between a given sentence pair and the corresponding similarity relation, which can be formulated as the following likelihood function:

$$\boldsymbol{\theta} = \arg\max_{\theta} L(\boldsymbol{\theta}) = \arg\max_{\theta} \sum_{i=1}^{N} \log P(X_i, R_i|\boldsymbol{\theta}). \quad (5)$$

For simplicity, in the following parts, we will ignore the summation notation and take the ith input X_i to introduce the details. Since the latent variable R_i is unobserved, we can follow EM algorithm and modify (5) as follows:

$$\begin{aligned}\boldsymbol{\theta} &= \arg\max_{\underset{(2)}{\theta}} \sum_{R} \underset{(1)}{P(R|X_i, \boldsymbol{\theta}_{\text{old}})} \log P(X_i, R|\boldsymbol{\theta}) \\ &= \arg\max_{\underset{(2)}{\theta}} \sum_{R} \underset{(1)}{M(X_i, \boldsymbol{\theta}_{\text{old}})} \log P(X_i, R|\boldsymbol{\theta}) \end{aligned} \quad (6)$$

where operation (1) $(M(X_i, \boldsymbol{\theta}_{\text{old}}))$ represents that using learned parameters $\boldsymbol{\theta}_{\text{old}}$ to estimate R_i for the input pair X_i. The motivation behind this operation is as follows: assuming that model parameter $\boldsymbol{\theta}$ has been learned well, we can use $\boldsymbol{\theta}$ to estimate the latent variable R, which can be used to describe the similarity distribution of the input data. To this end, operation (1) can be treated as the *E-step*: fixing the learning parameters $\boldsymbol{\theta}_{\text{old}}$ to calculate the joint distribution of input data.

After obtaining the similarity distribution of input data, we then use the predicted distribution to retrain the model parameters for better similarity score calculation. This is consistent with the operation (2) in (6). According to total probability, we can rewrite the latter part as follows:

$$\begin{aligned}\log P(X_i, R|\boldsymbol{\theta}) &= \log\left(P(R|X_i, \boldsymbol{\theta})P(X_i|\boldsymbol{\theta})\right) \\ &\approx \log P(R|X_i, \boldsymbol{\theta}) \\ &= \log M(X_i, \boldsymbol{\theta}) \end{aligned} \quad (7)$$

since sentence pair X_i is sampled randomly from entire training data with equal probability in CL framework, it is independent of model parameters $\boldsymbol{\theta}$. In other words, $P(X_i|\boldsymbol{\theta})$ is determined by the dataset. When the dataset is confirmed, $P(X_i|\boldsymbol{\theta})$ can be treated as a constant term. We can leverage $\log M(X_i, \boldsymbol{\theta})$ to approximate operation (2) in (6). In this process, we keep other information fixed and update model parameters $\boldsymbol{\theta}$. Therefore, operation (2) can be treated as the **M-step**: maximizing (7) to update the model.

Moreover, since input pair X_i only has a similar or dissimilar relation in CL framework, we can align (6) and (7) with the *cross-entropy* target as follows:

$$\begin{aligned}\boldsymbol{\theta} &= \arg\max_{\underset{(\text{M-step})}{\theta}} \sum_{r\in\{0,1\}} \underset{(\text{E-step})}{M(X_i, \boldsymbol{\theta}_{\text{old}})} \log M(X_i, \boldsymbol{\theta}) \\ &= \arg\max_{\theta} \{[(\boldsymbol{E} + \boldsymbol{O}_{t-1} + \bar{\boldsymbol{O}})\log(\boldsymbol{O}_t)]_i \\ &\qquad + [(1 - (\boldsymbol{O} + \boldsymbol{O}_{t-1} + \bar{\boldsymbol{O}}))\log(1 - \boldsymbol{O}_t)]_i\} \end{aligned} \quad (8)$$

where $\{\boldsymbol{E}, \boldsymbol{O}_{t-1}, \bar{\boldsymbol{O}}\}$ represent ground-truth relation, model prediction in the $(t-1)$th iteration, and random flip, which we use to approximate $\{A_i^+, A_i^-\}$ in (3). $\boldsymbol{O}_t$ is the predicted sentence pair relation (also known as the similarity score distribution) from the learning model in the tth iteration. $[\cdot]_i$ denotes that the calculation is focused on the ith example. If we use the equation $\boldsymbol{Y} = \boldsymbol{E} + \boldsymbol{O}_{t-1} + \bar{\boldsymbol{O}}$, this formulation is a standard *cross-entropy* target: $\boldsymbol{Y}_i * \ln(\hat{\boldsymbol{Y}}_i) + (1 - \boldsymbol{Y}_i) * (1 - \ln(\hat{\boldsymbol{Y}}_i))$.

One step further, we provide an example in Fig. 4 to visualize the learning target in (8). In CL framework, we have known that input data $\bar{x}_1$ have a positive relation with its augmentation $\bar{x}_2$. Therefore, we can use diagonal matrix $\boldsymbol{E}$ to add this prior information into the learning process. Moreover, if

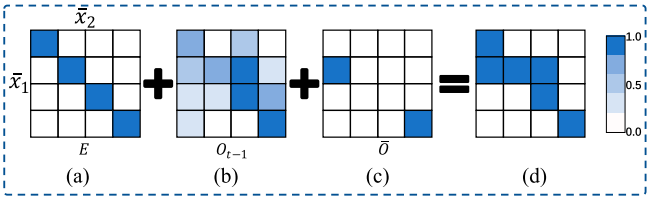


Fig. 4. Visualization of optimization target of our proposed *EMCRL*. (a) The diagonal matrix E for the original contrastive result. (b) The model prediction in the (t-1)th iteration. (c) The random flip result. (d) The total optimization target in EMCRL.

we have additional side information about the similarity distribution among the anchor and the in-batch negatives (e.g., supervised signals), we can also add them into $\boldsymbol{E}$. Next, it is hard to obtain the ground-truth similarity distribution among the anchor and in-batch negative examples. Therefore, following the settings of EM algorithm, we treat the distribution as a hidden variable, which is approximated by the learning model. Thus, we leverage $\boldsymbol{O}_{t-1}$ to denote the model capability in the previous iteration and to guide the learning process in the current iteration. Moreover, since EM algorithm is very sensitive to the initialization, we add $\bar{\boldsymbol{O}}$ with the probability η to improve the robustness of *EMCRL*. Along this line, we can effectively approximate the ground-truth similarity distribution between the anchor and the in-batch negatives, improve the quality of $\{A_i^+, A_i^-\}$ in CL framework, and boost the performance of CL-based models.

We have to note that $\bar{\boldsymbol{O}}$ is imported to prevent the learning model from being affected by initial parameters and improve the model robustness. To implement this design, we first initialize a square matrix of dimension equal to the batch size. Then, we randomly assign some element to 1 with the probability η, which can be treated as the pseudopositive relations of the anchor and corresponding negatives. Thus, we obtain the random flip matrix $\bar{\boldsymbol{O}}$ for model learning.

C. Discussion

Different from the existing in-batch negative sampling modification work, *EMCRL* has following desirable properties: first, *EMCRL* is isolated from the original data type and only focuses on semantic representations, which ensures its flexibility. Second, *EMCRL* employs EM algorithm to identify false negative examples with similarity scores and update parameters of learning models iteratively. The identification process is dynamic and related to the model capability so that it can estimate the ground-truth relations among in-batch negatives and the anchor, which is essential for the CL performance improvement. Moreover, this process does not require label information. Therefore, *EMCRL* is adaptive and has broader applications. Third, *EMCRL* modifies M-step by adding a random flip during optimization, which is helpful to alleviate the dependency of EM algorithm on the initialization and improve the model robustness. However, *EMCRL* still has some weaknesses. Since the optimization target in M-step is highly affected by learned embeddings. The selection of the representation model is a key factor in the final performance. Moreover, label information should also be considered when available, which we leave as our future work.

Algorithm 1: Pseudocode of *EMCRL*

input : $\boldsymbol{X}$: a batch of data with N examples;
η: the flip probability in M-step;
ϵ: threshold to verify whether the model is optimized;
output: $\boldsymbol{H}$: learned representations for the input

Data augmentation method $Aug(\cdot)$, the encoder $Enc(\cdot)$, and the projection $Proj(\cdot)$;
repeat
  for *each example* x *in current batch* **do**
    Use $Aug(\cdot)$ to generate $\bar{x}_1, \bar{x}_2$;
    Use $Enc(\cdot)$ with parameter $\boldsymbol{\theta}^{t-1}$ to generate $\boldsymbol{h}_1, \boldsymbol{h}_2$; Use $Proj(\cdot)$ to obtain $\boldsymbol{z}_1, \boldsymbol{z}_2$;
    Generate diagonal matrix $\boldsymbol{E}$ for current batch;
    `// E-step`
    Use Cosine Similarity to calculate the similarity score between x and $2(N-1)$ negative examples and obtain $\boldsymbol{O}_{t-1}$;
    `// M-step`
    Initialize a square matrix and randomly assign some element to 1 with probability η and obtain $\bar{\boldsymbol{O}}$;
    Construct the optimization target $\boldsymbol{Y} = \boldsymbol{E} + \boldsymbol{O}_{t-1} + \bar{\boldsymbol{O}}$;
    Calculate the Cross-Entropy loss with Eq.(8) and use AdamW to update the learning model and obtain parameters $\boldsymbol{\theta}^t$.
until $Loss_t - Loss_{(t-1)} < \epsilon$;

IV. Experiments

In this section, we first introduce datasets and experiment settings. Then, we provide a detailed analysis of experiment results and all methods. Moreover, we will provide a case study to better demonstrate the effectiveness of *EMCRL*.

A. Datasets

To better demonstrate the effectiveness and adaptiveness of *EMCRL*, we select sentence representation and image representation tasks to verify the model performance.

For sentence representation task, we conduct experiments over seven semantic textual similarity (STS) tasks in an unsupervised manner, similar to SimCSE work [10]. Note that no STS training sets are considered for evaluation. Moreover, clustering semantically similar sentences together is also employed to evaluate the quality of sentence embeddings in our work. The evaluation is conducted with SentEval Toolkit.[1]

For image representation task, we conduct experiments on STL10 image classification task [43] in the linear evaluation setting [8] to verify model generalization performance.

All experiments are conducted *five times*, and *the best results* are reported for model evaluation.

B. Model Implementation

For sentence representation task, we select advanced representative model: SimCSE [10] to verify the effectiveness of *EMCRL*. $\text{BERT}_{\text{base}}$ and $\text{RoBERTa}_{\text{base}}$ are considered as backbones, separately. Meanwhile, BERT, IS-BERT, RoBERTa, and IS-RoBERTa are also selected as baselines. The output of the

[1]https://github.com/facebookresearch/SentEval

TABLE III
EXPERIMENTAL RESULTS (SPEARMAN'S CORRELATION) ON STS TASKS IN AN UNSUPERVISED MANNER

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
GloVe embeddings♣	55.14	70.66	59.73	68.25	63.66	58.02	53.76	61.32
$\text{BERT}_{\text{base}}$ (first-last avg)♣	39.70	59.38	49.67	66.03	66.19	53.87	62.06	56.70
IS-$\text{BERT}_{\text{base}}$♣	56.77	69.24	61.21	75.23	70.16	69.21	64.25	66.58
SimCSE-$\text{BERT}_{\text{base}}$-dropout	67.74	81.97	74.10	78.89	78.42	77.14	72.54	75.83
w/ *EMCRL*-dropout	**68.82**↑	82.17↑	75.36↑	77.83↓	**78.76**↑	78.62↑	**73.60**↑	76.45↑
SimCSE-$\text{BERT}_{\text{base}}$-EDA	68.71	82.12	74.66	**80.87**	78.52	77.83	72.20	76.41
w/ *EMCRL*-EDA	68.43↓	**82.68**↑	**75.72**↑	80.69↓	78.6↑	**79.93**↑	73.54↑	**77.08**↑
$\text{RoBERTa}_{\text{base}}$ (first-last avg)♣	40.88	58.74	49.07	65.63	61.48	58.55	61.63	56.57
IS-$\text{RoBERTa}_{\text{base}}$	57.13	71.45	61.62	75.03	72.12	67.30	65.44	67.15
SimCSE-$\text{RoBERTa}_{\text{base}}$-dropout	70.16	82.11	72.67	80.32	80.5	80.79	68.91	76.49
w/ *EMCRL*-dropout	71.09↑	82.37↑	73.41↑	80.46↑	**80.71**↑	**81.26**↑	68.75↓	76.86↑
SimCSE-$\text{RoBERTa}_{\text{base}}$-EDA	70.42	81.95	73.13	80.44	80.62	81.12	69.57	76.75
w/ *EMCRL*-EDA	**71.38**↑	**82.4**↑	**73.65**↑	**81.85**↑	80.36↓	80.89↓	**70.15**↑	**77.24**↑

Note: ♣: RESULTS FROM [10]. $\text{BERT}_{\text{base}}$: USING $\text{BERT}_{\text{base}}$ AS THE EMBEDDING BACKBONE. W/*EMCRL*: using *EMCRL* to improve negative sampling. **boldface** and underline denote the best and the second best results. ↑ (↓) Indicates performance improvement (decrease) compared with original SimCSE.

first token "[CLS]" is treated as the sentence representation. For data augmentations, we not only select dropout operation as SimCSE did but also select EDA [19] to generate augmented samples from different views.

During model training, we randomly select 10^6 sentence pairs from English Wikipedia, similar to what Gao et al. [10] did. We leverage AdamW [44] as the optimizer with the learning rate as 3×10^{-4}. The batch size is 640. The embedding size (i.e., dimension of $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$) is 768. The projection size (i.e., dimension of $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$) is 300. The dropout is set as 0.3. The noise probability for $\bar{\boldsymbol{O}}$ is set as 4×10^{-4}. The temperature τ in (2) and (3) is set as 0.4

For image representation task, SimCLR [8] is selected as the representative model. Similar to sentence evaluation, we also choose different backbones (i.e., VGG16, ViT-b-16, and ResNet-50) for generalization evaluation. A linear classifier is trained on top of frozen backbones for the classification task. For data augmentations, *crop, color distort, flip*, and *grayscale* are used to generate different image views.

For model training, we also select AdamW as the optimizer. The corresponding learning rate is 1×10^{-3}. The batch size is set as 256. The embedding size and projection size are set as 512 and 128. The dropout is set as 0.3. The noise probability for $\bar{\boldsymbol{O}}$ is set as 1×10^{-5}. The temperature τ in (2) and (3) is set as 0.3

For negative sampling, since *EMCRL* focuses on in-batch negative sampling strategy and aims to improve the quality of selected samples, we choose all the other samples except for the ith example as the negatives of the ith example.

C. Experiments on Sentence Representations

Table III reports overall experiments on STS tasks. To make a fair comparison, we reproduce SimCSE with different backbones and data augmentations. Therefore, the results of SimCSE reported in Table III are slightly different from the results from the original paper [10]. From these results, we conclude the following observations.

First of all, CL can effectively improve model performance by helping models learn invariant features of input data. Moreover, when using *EMCRL* to improve the quality of negative sampling, model performance has been further improved to varying degrees (1.7% and 2.6% improvements for SimCSE-$\text{BERT}_{\text{base}}$ on STS14 and STS-B), proving the effectiveness of *EMCRL*. By iteratively using the learning model to estimate the ground-truth sample similarity distributions and using the estimation results to update the model parameters, *EMCRL* can dynamically improve the quality of in-batch negative sampling in an unsupervised manner and boost the model performance.

By comparing different backbones and data augmentations, we observe that $\text{RoBERTa}_{\text{base}}$ and EDA-based methods achieve better performance. For one thing, RoBERTa can generate better representations than BERT, which helps to achieve better performance. Furthermore, EDA can generate more diverse samples and provide a comprehensive view for input sentences, which not only is essential for representation learning but also indicates the necessity of better data augmentations. Moreover, we observe that improvement of *EMCRL* on EDA operation is more obvious than dropout. The possible reason is that EDA operation is more likely to generate false positive samples, although generating more diverse samples. On the contrary, by randomly dropping some learned features, dropout can achieve competitive performance with minimal risk of generating false positive samples. This observation is consistent with the conclusion in [10].

Apart from the improvement that *EMCRL* has made, we can also observe that integrating *EMCRL* might cause a degradation of model performance on some tasks (e.g., 1.34% decrease of SimCSE-BERT-dropout on STS15 and 0.32% decrease of SimCSE-RoBERTa-EDA on STS16). After a thorough analysis of these datasets, we speculate the following possible reasons. Though *EMCRL* is able to estimate the ground-truth sample similarity distributions, it is still functional in an unsupervised manner. The quality of *learned representations, different data augmentations*, and *parameter initialization* will have different impacts on its effectiveness. Moreover, we leverage randomly

TABLE IV
IMAGE CLASSIFICATION RESULTS (ACCURACY) OF LINEAR CLASSIFIERS TRAINED ON REPRESENTATIONS LEARNED OF SIMCLR WITH DIFFERENT BACKBONES

Backbones	SimCLR	SimCLR-*EMCRL*
VGG-16	68.91%	72.16% (+3.25%) ↑
ResNet-18	79.50%	80.49% (+0.99%)↑
ResNet-34	79.96%	81.96% (+2.0%)↑
ResNet-50	81.88%	82.73% (+0.85%)↑
ViT-b-16	81.79%	**82.83%** (+1.04%)↑

Note: The boldface and underline denote the best and second best results.

TABLE V
RESULTS OF DIFFERENT POOLING STRATEGIES ON SIMCSE-BERT$_{\text{base}}$ WITH *EMCRL*

Strategy	STS-B	SICK-R	Avg.
[CLS]	78.62	**73.6**	76.11
Last layer			
Mean Pooling	78.56	73.51	70.04
attention pooling	77.04	71.53	74.29
First-last			
Mean Pooling.	**78.75**	73.56	**76.16**
Attention Pooling	69.45	67.63	68.54

Note: The boldface and underline denote the best and second best results.

flip $\bar{O}$ to introduce some noise to help *EMCRL* be insensitive to the initialization. However, this introduced noise may hurt the model performance when the data are difficult. To this end, choosing an instance-level CL framework would be a safe choice. Thus, we can observe that SimCSE can achieve better performance without our proposed *EMCRL* on these datasets. Meanwhile, this phenomenon also inspires us to explore *EMCRL* in a supervised manner, which is also the future direction of our work.

D. Experiments on Image Representation

To better evaluate the generalization of *EMCRL*, we conduct experiments on image classification over STL10 in a linear evaluation setting. Similar to previous experiments, we also select different backbones for comprehensive evaluation. Results are summarized in Table IV. We observe that deeper backbones achieve better performance, proving the importance of backbone encoders. This observation is consistent with the results in Section IV-C. Moreover, *EMCRL* can further improve the performance (e.g., 2.0% and 3.25% absolute improvement for ResNet-34 and VGG-16 backbones). Furthermore, *EMCRL* can help SimCLR with a relatively simpler backbone to achieve better performance (e.g., 80.49% for ResNet-18 backbone is better than 79.96% for ResNet-34 backbone). These phenomena also provide solid proof for the effectiveness and superiority and the generalization of *EMCRL*.

E. Ablation Study

To make a deeper analysis about *EMCRL*, we further compare the effect of different pooling strategies, different data augmentations, and different flip probability η. We also report the uniformity and alignment results for better illustration. We select SimCSE-BERT$_{\text{base}}$ with *EMCRL* as the baseline and compare the performance with different settings. Results are summarized in Tables V and VI and Figs. 5 and 6.

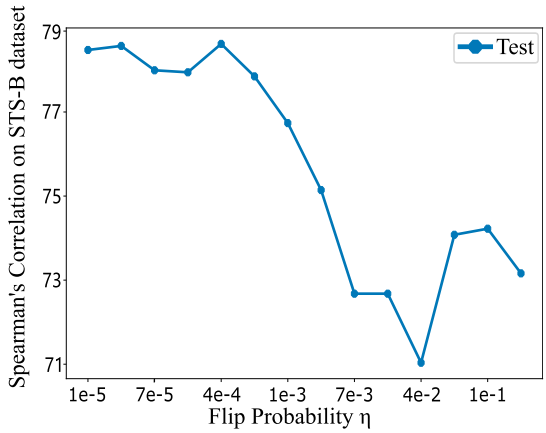


Fig. 5. Impact of additional flip probability η in M-step on *EMCRL* over STS-B test.

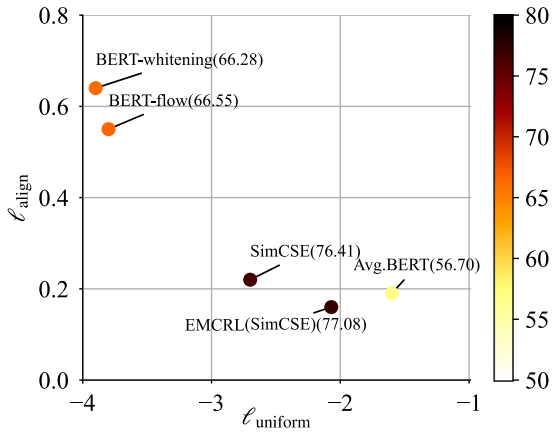


Fig. 6. ℓ_{align}-ℓ_{uniform} plot for different sentence representation methods based on BERT$_{\text{base}}$.

1) Pooling Strategy: Inspired by attention pooling in [45], we also verify the effectiveness of attention mechanism, as well as mean pooling. According to Table V, attention mechanism is not effective in this scenario, even compared with the straightforward "[CLS]" representation. The possible reason is that attention pooling requires extra parameters, which increases the model complexity. The work [45] obtained the best parameters in a supervised manner, where parameters are determined by downstream tasks. However, in unsupervised settings, these parameters are hard to optimize, which leads to a performance decrease. Second, mean pooling has better performance when using the first-last outputs of PLMs, demonstrating that different layers in PLMs measure sentences from different aspects. Selecting a suitable combination of different layers may provide better access to sentence semantics.

2) Data Augmentation: We mainly focus on EDA and examine different augmentation strategies. Results are summarized in Table VI. Note that *SR* denotes synonym replacement; *RI* denotes random insertion; *RS* denotes random swap; and

TABLE VI
RESULTS OF DIFFERENT AUGMENTATION STRATEGIES

Strategy	STS-B	SICK-R	Avg.
Dropout	78.62	73.60	76.11
EDA			
SR	78.65	73.57	76.11
RI	78.62	73.63	76.13
RS	**78.74**	**73.68**	**76.21**
RD	77.25	72.07	74.66

Note: SR: synonym replacement; RI: random insertion; RS: random swap; RD: random deletion. The boldface and underline denote the best and second best results.

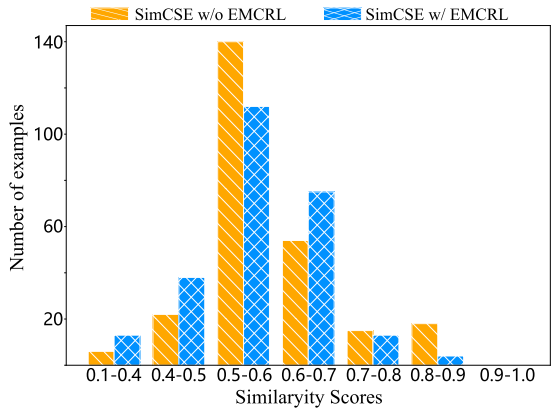


Fig. 7. Learned similarity distributions generated by unsupervised SimCSE and the one with our proposed *EMCRL*. Note that the mini-batch data are the same as the data used in Fig. 1.

RD denotes random deletion. Among these strategies, SR has a similar performance to dropout, which demonstrates that SR is a relatively safe way to generate semantically similar sentences. Moreover, other strategies (i.e., *RI, RS,* and *RD*) have better performance than SR. For one thing, these results prove that better data augmentations can provide more diverse examples for CL to learn invariant features of input data, which is in favor of improving the capability of CL. Furthermore, stronger data augmentation strategies are more likely to generate false negative examples, which will mislead the model training process and hurt the model performance. Thus, a better negative sampling strategy is essential for the utilization of strong data augmentation strategies. By employing EM algorithm to use learning models to appropriate ground-truth similarity distribution for the positive and negative sets construction, as well as use constructed sets to update parameters of learning models iteratively, our proposed *EMCRL* is effective in helping CL to exploit higher quality samples for representation learning and improve model performance.

3) Sensitive Test of Hyperparameter η: In Section III-B, we propose to add an additional random flip for the model robustness improvement. Therefore, we conduct experiments with different flip ratios η to verify its impact. As shown in Fig. 5, overall performance first increases and then decreases. The best results are achieved with $\eta = 0.0004$. This phenomenon is in line with our expectations. For increasing part, the random flip will enforce *EMCRL* to better estimate the ground-truth similarity distributions of the in-batch negatives and the anchor, which will help to improve model performance. For the decreasing part, when the flip ratio is too large, the M-step in *EMCRL* will impose too much noise. Based on (8), $\bar{\boldsymbol{O}}$ will mislead *EMCRL* to estimate the ground-truth similarity distributions, resulting in an incorrect approximation. As for the increase in model performance when $\eta > 0.04$, we speculate the possible reason is that the random flip may have some overlap with the high-confidence false negative samples, resulting in a smaller proportion of effective flips.

4) Uniformity and Alignment: Following [46], we also select uniformity and alignment metrics to verify the contrastive representation quality of different embedding models. Results are reported in Fig. 6. Since *EMCRL* leverages SimCSE as the base model, we only compare the results of SimCSE with and without *EMCRL* strategy. From the figure, we can observe that by using *EMCRL* strategy, the alignment of SimCSE is improved, whereas uniformity is still kept in a relatively good performance. This phenomenon proves that *EMCRL* strategy is effective in improving the quality of positive and negative samples, which is in favor of CL performance improvement.

F. Case Study

In this section, we provide some intuitionistic examples to explain the effectiveness of *EMCRL*. We first reported the similarity distributions and updated positive and negative sets in one mini-batch data. Then, we give more intuitionistic examples to demonstrate the effectiveness of *EMCRL*.

1) Similarity Score Distribution: We select the same data with Fig. 1 and report the similarity distribution of SimCSE after using *EMCRL*. Compared with unsupervised SimCSE [Fig. 7(a)], when employing our proposed *EMCRL*, SimCSE [Fig. 7(b)] can generate more diverse similarity scores. Since the anchor example belongs to the category "1," according to Fig. 1(a), we can conclude that *EMCRL* is able to identify false negative examples effectively. By putting these false negatives into the positive set, *EMCRL* helps to boost the capability of CL to generate high-quality semantic representations, which is essential for model performance improvement. These results clearly demonstrate how *EMCRL* achieves CL-based model performance improvement.

2) Sentence Representation Visualization: We randomly sample 30 sentences from English Wikipedia to conduct this experiment. Since supervised SimCSE-BERT$_{\text{base}}$ can leverage labels to better distinguish relations among sentences [10], we select its generated representations to calculate similarity as the ground truth. Then, unsupervised SimCSE and the one with *EMCRL* strategy are tested to generate corresponding representations, respectively. Next, we calculate the cosine similarity of any two sentences to draw heat maps to visualize corresponding results and report results in Fig. 8. The left part in Fig. 8(a) reports the corresponding sentences.

First of all, we observe that for easy samples, all methods generate accurate representations, as illustrated by the diagonal line and the red box in the upper left corner in heat maps. Second, Fig. 8(a) demonstrates that there exist positive examples

Index	Sentence
1	Both studies find that aerobic fitness levels do not improve the predictive power of fat free mass for resting metabolic rate.
2	An important game mechanic is the construction of a road network so as to allow for an efficient transportation system, as any settlers transporting goods must use roads.
3	He has a small, bucket-like head and carries a trident.
4	For example, in some systems, the interatomic spacing and the atomic charge of a material could allow only electrons of certain wavelengths to exist.
5	The Second Schleswig War was the second military conflict over the Schleswig-Holstein Question of the nineteenth century.
6	The 2014 Danish TV series "1864" depicts the Second Schleswig War
7	A second conflict, the Second Schleswig War, erupted in 1864.
8	His book describes many of his notable techniques used in his film editing.
9	All of it is contained in the body as important parts of tissues, blood hormones, and enzymes.
10	With her career stalled, she supported her husband during the production of "Salt of the Earth" (1954).
11	Major road networks within Warri Metropolis has been improved upon by the state government to improve the image of the city.
12	"One of the Hollywood Ten" (2000) chronicled Sondergaard's relationship with Biberman and her role in the making of "Salt of the Earth".
13	All communes and wards are connected to the road network.
14	It has been translated into many languages, including English, Russian, German, French, Italian, Turkish and Arabic.
15	The effectiveness of the system and the weapon were demonstrated by a new, long-range record as well as a successful two-missile salvo shot.
16	The other assumption of the dilemma is that there is a universal right and wrong, against which a god either creates or is defined by.
17	Salt has been extracted from rock salt beds underneath the Cheshire Plain since Roman times.
18	Its smoky breath blankets the surrounding area like a thick, black fog with an aroma of salt and pepper.
19	Heart rate is important for basal metabolic rate and resting metabolic rate because it drives the blood supply, stimulating the Krebs cycle.
20	This is a demographic history of Quebec chronicling the evolution of the non-indigenous population in Quebec.
21	Salts are frequently used for de-icing, but salt solutions are not used for cooling systems because they induce corrosion of metals.
22	The following figure shows examples of minimum vertex covers in the previous graphs.
23	In other systems, the crystalline structure of a material could allow waves to propagate in one direction, while suppressing wave propagation in another direction.
24	The hill's formerly steep slopes had become flattened in months of intense shelling and bombardment.
25	The minimum vertex cover problem is the optimization problem of finding a smallest vertex cover in a given graph.
26	Horses could not carry or pull their loads properly because of the snow and ice; riders had to dismount and lead their horses.
27	Men and horses had trouble standing.
28	The earth on the hill had remained black in the winter, as the snow kept melting in the many fires and explosions.
29	By measuring heart rate we can then derive estimations of what level of substrate utilization is actually causing biochemical metabolism in our bodies at rest or in activity.
30	But salt attacks both the concrete and steel rebar used in girders, pylons, and other parts.

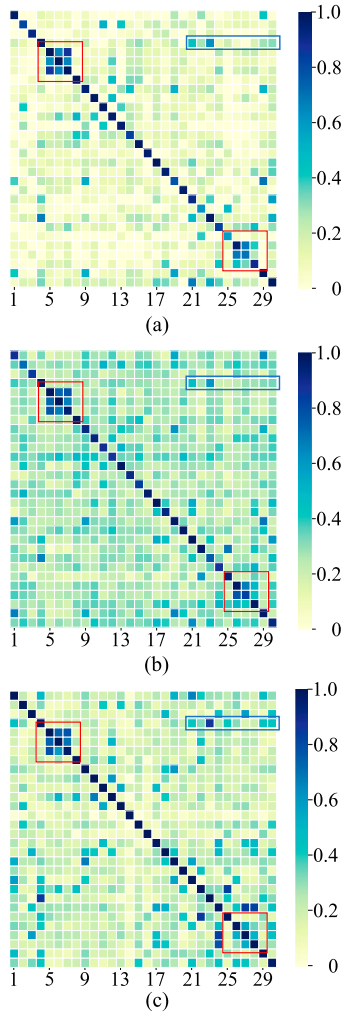


Fig. 8. Similarity visualization heatmaps of sentence embeddings in the left part from (a) Supervised SimCSE; (b) Unsupervised SimCSE; (c) Unsupervised SimCSE with our proposed EMCRL.

for the anchor in one batch (dark square in the blue rectangle in the upper right corner in heat maps). In-batch negative sampling strategy will easily treat them as negatives, which will harm the capability of CL. Thus, it is essential to distinguish these false negative samples. Third, Fig. 8(c) is more differentiated and closer to Fig. 8(a). Its lower right corner, especially the red box in the lower right corner, is much clearer and more distinguishable than Fig. 8(b). These phenomena demonstrate that similarity scores in Fig. 8(c) are more accurate than in Fig. 8(b). By introducing EM algorithm into the optimization process, *EMCRL* can use the training model to generate similarity scores for false negative identification and use the modified sample sets to train the model iteratively. Therefore, the sample quality can be improved in an unsupervised manner, which is helpful for models to generate more accurate and differentiated representations.

V. Conclusion

In this article, we argued that in-batch negative sampling in CL suffered from a false negative selection problem, which has compromised the capability of CL. To tackle this problem, we proposed to incorporate EM algorithm into the optimization of CL and developed a novel *EMCRL*. Specifically, we employed a commonly used CL framework to generate representations. Then, we focused on improving the quality of in-batch negatives with the consideration of model capability. By iteratively using the learning model to estimate the ground-truth sample similarity distributions for false negative identification and using the estimation results to update the model parameters, *EMCRL* could help to approximate ground-truth similarity distributions among in-batch negatives and the anchor. Moreover, to improve the robustness of *EMCRL*, we proposed to add a random flip into the optimization target in M-step of *EMCRL*, which has been proven useful. We have to note that *EMCRL* is model-agnostic, simple to implement, and stable to train. Finally, we have conducted extensive experiments to demonstrate the effectiveness of our proposed *EMCRL*.

VI. Limitations

To inspire future work, we summarize some limitations of our proposed *EMCRL* as follows.

1) Large language models (LLMs) have not been considered in our work. How to exploit the advantages of LLMs

to further assist *EMCRL* for the quality improvement of examples in CL needs to be investigated.

2) The optimization in M-step is highly related to the learning embeddings. How to select the best embedding learning model should be discussed and verified.
3) *EMCRL* achieved impressive performance on sentence embedding and image classification tasks. The potential of our proposed method on other downstream tasks, such as NLI and QA, still needs to be investigated.

REFERENCES

[1] H. Liu, I. Chatterjee, M. Zhou, X. S. Lu, and A. Abusorrah, "Aspect-based sentiment analysis: A survey of deep learning methods," *IEEE Trans. Comput. Social Syst.*, vol. 7, no. 6, pp. 1358–1375, Dec. 2020.

[2] W. An, F. Tian, P. Chen, and Q. Zheng, "Aspect-based sentiment analysis with heterogeneous graph neural network," *IEEE Trans. Comput. Social Syst.*, vol. 10, no. 1, pp. 403–412, Feb. 2023.

[3] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 776–794.

[4] W. Wang, W. Zhou, J. Bao, D. Chen, and H. Li, "Instance-wise hard negative example generation for contrastive learning in unpaired image-to-image translation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 14020–14029.

[5] P. Khosla et al., "Supervised contrastive learning," 2020, *arXiv:2004.11362*.

[6] T.-S. Chen, W.-C. Hung, H.-Y. Tseng, S.-Y. Chien, and M.-H. Yang, "Incremental false negative detection for contrastive learning," 2021, *arXiv:2106.03719*.

[7] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9729–9738.

[8] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[9] J.-B. Grill et al., "Bootstrap your own latent: A new approach to self-supervised learning," 2020, *arXiv:2006.07733*.

[10] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple contrastive learning of sentence embeddings," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2021, pp. 6894–6910.

[11] W. Sun, Z. Shi, S. Gao, P. Ren, M. de Rijke, and Z. Ren, "Contrastive learning reduces hallucination in conversations," in *Proc. AAAI Conf. Artif. Intell.*, vol. 37, no. 11, 2023, pp. 13618–13626.

[12] Y. Jian, C. Gao, and S. Vosoughi, "Contrastive learning for prompt-based few-shot language learners," in *Proc. Conf. North Amer. Chap. Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2022, pp. 5577–5587.

[13] Q. Cheng, X. Yang, T. Sun, L. Li, and X. Qiu, "Improving contrastive learning of sentence embeddings from AI feedback," in *Findings of the Association for Computational Linguistics: ACL 2023*. Toronto, Canada: Association for Computational Linguistics, 2023, pp. 11122–11138.

[14] S. O'Brien and M. Lewis, "Contrastive decoding improves reasoning in large language models," 2023, *arXiv:2309.09117*.

[15] Q. Bao et al., "Contrastive learning with logic-driven data augmentation for logical reasoning over text," 2023, *arXiv:2305.12599*.

[16] T. T. Cai, J. Frankle, D. J. Schwab, and A. S. Morcos, "Are all negatives created equal in contrastive instance discrimination?" 2020, *arXiv:2010.06682*.

[17] J. T. Ash, S. Goel, A. Krishnamurthy, and D. Misra, "Investigating the role of negatives in contrastive representation learning," 2021, *arXiv:2106.09943*.

[18] K. Xu, F. Wang, H. Wang, Y. Wang, and Y. Zhang, "Mitigating the impact of data sampling on social media analysis and mining," *IEEE Trans. Comput. Social Syst.*, vol. 7, no. 2, pp. 546–555, Apr. 2020.

[19] J. Wei and K. Zou, "Eda: Easy data augmentation techniques for boosting performance on text classification tasks," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 6382–6388.

[20] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 16000–16009.

[21] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 5812–5823.

[22] K. Zhou, B. Zhang, X. Zhao, and J.-R. Wen, "Debiased contrastive learning of unsupervised sentence representations," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics* (vol. 1 Long Papers), 2022, pp. 6120–6130.

[23] K. Ethayarajh, "How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 55–65.

[24] P. Shao et al., "Average user-side counterfactual fairness for collaborative filtering," *ACM Trans. Inf. Syst.*, vol. 42, no. 5, pp. 1–26, 2024.

[25] Y. Kalantidis, M. B. Sariyildiz, N. Pion, P. Weinzaepfel, and D. Larlus, "Hard negative mixing for contrastive learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 21798–21809, 2020.

[26] D. Zhang et al., "Supporting clustering with contrastive learning," 2021, *arXiv:2103.12953*.

[27] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chap. Assoc. Comput. Linguistics: Hum. Language Technol.*, vol. 1, 2019, pp. 4171–4186.

[28] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal Process. Mag.*, vol. 13, no. 6, pp. 47–60, Nov. 1996.

[29] X. Wang, D. Zhang, H.-Z. Tan, and D.-J. Lee, "A self-fusion network based on contrastive learning for group emotion recognition," *IEEE Trans. Comput. Social Syst.*, vol. 10, no. 2, pp. 458–469, Apr. 2023.

[30] Y. Yan, R. Li, S. Wang, F. Zhang, W. Wu, and W. Xu, "ConSERT: A contrastive framework for self-supervised sentence representation transfer," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.* (vol. 1 Long Papers), 2021, pp. 5065–5075.

[31] X. Wu, C. Gao, L. Zang, J. Han, Z. Wang, and S. Hu, "ESimCSE: Enhanced sample building method for contrastive learning of unsupervised sentence embedding," 2021, *arXiv:2109.04380*.

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

[33] V. Suresh and D. Ong, "Not all negatives are equal: Label-aware contrastive loss for fine-grained text classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2021, pp. 4381–4394.

[34] T. Huynh, S. Kornblith, M. R. Walter, M. Maire, and M. Khademi, "Boosting contrastive self-supervised learning with false negative cancellation," 2020, *arXiv:2011.11765*.

[35] Y. Wang et al., "ClusterSCL: Cluster-aware supervised contrastive learning on graphs," in *Proc. ACM Web Conf.*, 2022, pp. 1611–1621.

[36] J. Robinson, C.-Y. Chuang, S. Sra, and S. Jegelka, "Contrastive learning with hard negative samples," 2020, *arXiv:2010.04592*.

[37] M. Wu, M. Mosse, C. Zhuang, D. Yamins, and N. Goodman, "Conditional negative sampling for contrastive learning of visual representations," 2020, *arXiv:2010.02037*.

[38] C.-Y. Chuang, J. Robinson, Y.-C. Lin, A. Torralba, and S. Jegelka, "Debiased contrastive learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 8765–8775, 2020.

[39] J. Wang, T. Zhu, L. Chen, H. Ning, and Y. Wan, "Negative selection by clustering for contrastive learning in human activity recognition," 2022, *arXiv:2203.12230*.

[40] Y. Su et al., "Dialogue response selection with hierarchical curriculum learning," 2020, *arXiv:2012.14756*.

[41] Y. Liu et al., "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.

[42] A. Radford et al., "Improving language understanding by generative pre-training," pp. 1–12, 2018.

[43] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 215–223.

[44] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–18.

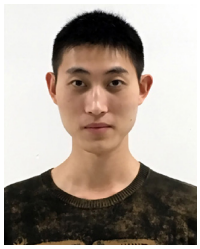
[45] K. Zhang, L. Wu, G. Lv, M. Wang, E. Chen, and S. Ruan, "Making the relation matters: Relation of relation learning network for sentence semantic matching," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 14411–14419.

[46] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2020, pp. 9929–9939.

Kun Zhang (Member, IEEE) received the Ph.D. degree in computer science and technology from the University of Science and Technology of China, Hefei, China, in 2019.

He is currently an Associate Professor with Hefei University of Technology (HFUT), Hefei, China. His research interests include natural language understanding, large language model alignment and debiasing, and recommender system. He has published several papers in refereed journals and conferences, such as IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, ACL, SIGIR, WWW, AAAI, KDD, and ICDM.

Dr. Zhang received the KDD 2018 Best Student Paper Award.



Guangyi Lv received the Ph.D. degree in computer science and technology from the University of Science and Technology of China, Hefei, China, in 2019.

He is currently a Faculty Member with AI Laboratory, Lenovo Research, Beijing, China. His research interests include natural language processing, lifelong machine learning, and few-shot learning. He has published several papers in refereed journals and conferences, such as IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, IEEE TRANSACTIONS ON BIG DATA, AAAI, IJCAI, and ICDM.

Le Wu (Member, IEEE) received the Ph.D. degree in computer science from the University of Science and Technology of China (USTC), Hefei, China, in 2015.

She is a Professor with Hefei University of Technology (HFUT), Hefei, China. Her research interests include data mining, recommender system, and social network analysis. She has published several papers in referred journals and conferences, such as IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, *ACM Transactions on Intelligent Systems and Technology*, AAAI, IJCAI, KDD, SDM, and ICDM.

Dr. Wu was the recipient of the Best of SDM 2015 Award.

Richang Hong (Member, IEEE) received the Ph.D. degree in signal and information processing from the University of Science and Technology of China (USTC), Hefei, China, in 2008.

He is currently a Professor with Hefei University of Technology, Hefei, China. He has published more than 100 publications in the areas of his research interests, which include multimedia question answering, video content analysis, and pattern recognition.

Dr. Hong is a member of the Association for Computing Machinery. He was a recipient of the Best Paper Award in the ACM Multimedia 2010.

Meng Wang (Fellow, IEEE) received the B.E. and Ph.D. degrees in signal and information processing from the University of Science and Technology of China (USTC), Hefei, China, in 2003 and 2008, respectively.

He is a Professor with Hefei University of Technology (HFUT), Hefei. His research interests include multimedia content analysis, computer vision, and pattern recognition. He has authored more than 200 book chapters, journal, and conference papers in these areas.

Dr. Wang was the recipient of the ACM SIGMM Rising Star Award 2014. He is an Associate Editor of IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS.