Supervised Prototypical Contrastive Learning for Emotion Recognition in Conversation

Xiaohui Song[1,2⋆], **Longtao Huang**[3], **Hui Xue**[3] and **Songlin Hu**[1,2†]
[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]School of Cyber Security, University of Chinese Academy of Sciences
[3]Alibaba Group
{songxiaohui,husonglin}@iie.ac.cn, {kaiyang.hlt, hui.xueh}@alibaba-inc.com

Abstract

Capturing emotions within a conversation plays an essential role in modern dialogue systems. However, the weak correlation between emotions and semantics brings many challenges to emotion recognition in conversation (ERC). Even semantically similar utterances, the emotion may vary drastically depending on contexts or speakers. In this paper, we propose a **S**upervised **P**rototypical **C**ontrastive **L**earning (**SPCL**) loss for the ERC task. Leveraging the Prototypical Network, the SPCL targets at solving the imbalanced classification problem through contrastive learning and does not require a large batch size. Meanwhile, we design a difficulty measure function based on the distance between classes and introduce curriculum learning to alleviate the impact of extreme samples. We achieve state-of-the-art results on three widely used benchmarks. Further, we conduct analytical experiments to demonstrate the effectiveness of our proposed SPCL and curriculum learning strategy. We release the code at https://github.com/caskcsg/SPCL.

1 Introduction

With the development of online social networks, capturing emotions during conversations has gained increasing attention in both academia and industry(Li et al., 2020; Shen et al., 2021; Wang et al., 2020; Ghosal et al., 2020; Song et al., 2022; Zhu et al., 2021). Emotion recognition in conversation (ERC) is critical in many scenarios, such as chatbots, healthcare applications, mining opinions on social media, and so on(Poria et al., 2019b). The ERC task aims to identify different emotions at each turn within a conversation based on the transcript. A conversation often contains several speakers and runs several turns; thus, emotions can vary drastically during the conversation. Compared to traditional text classification tasks, figuring out emotions needs not only one turn of textual utterance but also contextual information. An example of ERC is illustrated in Figure 1.

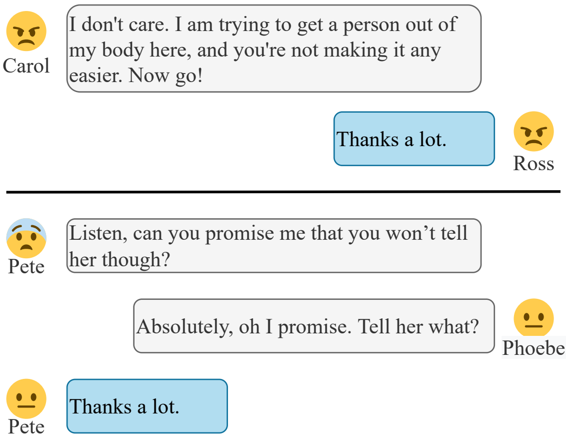

Carol
I don't care. I am trying to get a person out of my body here, and you're not making it any easier. Now go!
Thanks a lot.
Ross
Pete
Listen, can you promise me that you won't tell her though?
Absolutely, oh I promise. Tell her what?
Phoebe
Pete
Thanks a lot.

Figure 1: Examples of emotion recognition in conversation. The same utterance "Thanks a lot" can convey different emotions in different contexts.

Contrastive learning applied to self-supervised representation learning has seen a resurgence in recent years. Khosla et al. (2020) extended the self-supervised batch contrastive approach to the fully-supervised setting and show outperformance over cross-entropy loss in several benchmarks. Although CoG-BART(Li et al., 2021) has demonstrated the effectiveness of supervised contrastive learning (SCL) in the ERC task, there are still two issues worth to solve when building an ERC model with SCL: (1) As illustrated in Figure 2, existing ERC datasets are often class-imbalanced and samples may not be able to meet appropriate positive/negative samples within a mini-batch. (2) Existing ERC datasets are usually collected in a multi-modal manner. There are some conversations whose textual information is insufficient to distinguish emotions. Training a textual ERC model with those extreme samples may lead to performance degradation.

For the first issue, we propose a **S**upervised

[⋆]Work done during internship at Alibaba Group.
[†] Corresponding Author.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 5197–5206
December 7-11, 2022 ©2022 Association for Computational Linguistics

Prototypical **C**ontrastive **L**earning (**SPCL**) loss, which integrates Prototypical Network(Snell et al., 2017) and supervised contrastive learning. SPCL maintains a representation queue for each category. At each training step, SPCL samples a certain number of representations from these queues as the support set and calculates a temporary prototype vector for each emotion category. These prototype vectors are used as samples of the corresponding class to compute the loss. SPCL ensures that each sample has at least one positive sample of the same category and negative samples of all other categories within a mini-batch. Experiments show that SPCL can work well in class-imbalanced scenarios and is less sensitive to the training batch size.

To alleviate the performance degradation caused by extreme samples, we combine curriculum learning(Bengio et al., 2009) with contrastive learning. We design a distance-based difficulty measure function. By sorting the training data via this function, we can schedule the training data in an *easy-to-hard* fashion. Experimental results demonstrate the effectiveness of our proposed curriculum learning strategy. Finally, we utilize SimCSE(Gao et al., 2021), a pretrained language model trained with contrastive learning as our backbone model. Combining our proposed SPCL loss and curriculum learning strategy, we reach state-of-the-art results on three widely used benchmarks. In summary, our contributions can be concluded as follows:

- We propose a **S**upervised **P**rototypical **C**ontrastive **L**earning (**SPCL**) loss for the ERC task, which can perform supervised contrastive learning efficiently on class-imbalanced data and has no need for large batch size.
- To the best of our knowledge, we are the first to combine supervised contrastive learning and curriculum learning for the ERC task.
- We achieve state-of-the-art results on three widely used benchmarks. Experimental results further demonstrate the effectiveness of our proposed SPCL loss and curriculum learning strategy.

2 Related Work

2.1 Emotion Recognition in Conversation

Most existing approaches focus on context modeling. They can be divided into sequence-based, graph-based, and knowledge-enhanced methods. Sequence-based methods consider contextual information as utterance sequences. ICON(Hazarika et al., 2018) and HiGRU(Jiao et al., 2019) both use the gated recurrent unit to capture the context information. DialogRNN(Majumder et al., 2019) is a recurrence-based method that models dialog dynamics with several RNNs. DialogueCRN(Hu et al., 2021) introduces multi-turn reasoning modules to model the ERC task from a cognitive perspective. CoMPM(Lee and Lee, 2021) models the context and speaker's memory via pretrained language models. For those graph-based methods, DialogGCN(Hu et al., 2021) and RGAT(Ishiwatari et al., 2020) build a graph upon the utterances nodes. ConGCN(Zhang et al., 2019) trades both speakers and utterances as nodes and builds a single graph upon the whole ERC dataset. DAG-ERC(Shen et al., 2021) uses a directed acyclic graph (DAG) to model the intrinsic structure within a conversation. Knowledge-enhanced methods(Zhong et al., 2019; Zhu et al., 2021; Ghosal et al., 2020; Zhang et al., 2020) usually utilize external knowledge from ATIMOC(Sap et al., 2019) or ConceptNet(Liu and Singh, 2004). Besides individual models, several frameworks have also been proposed. Yang et al. (2021) developed an ERC-oriented hybrid curriculum learn-

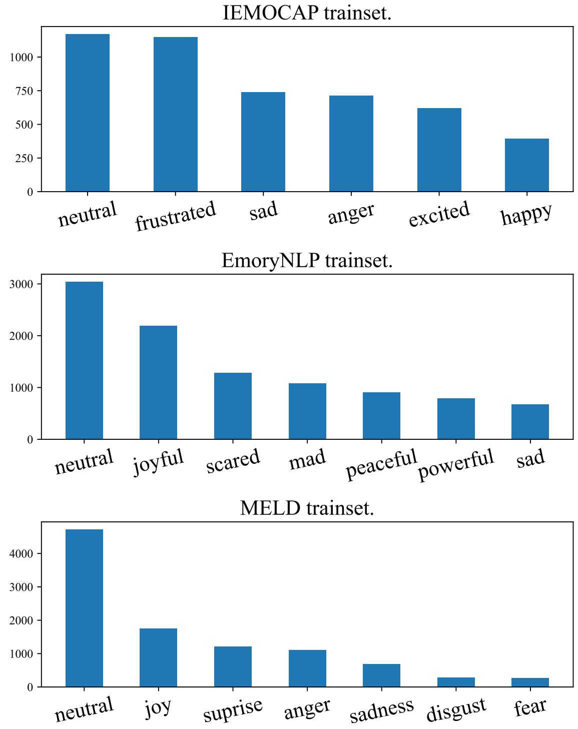


Figure 2: Emotion distributions of the three datasets.

ing framework and Bao et al. (2022) proposed a speaker-guided encoder-decoder framework, formulating the modeling of speaker interactions as a flexible component.

2.2 Contrastive Learning

In the field of natural language processing, SimCSE(Gao et al., 2021) is a state-of-the-art contrastive learning framework for generating sentence embeddings, it can learn from unlabeled sentences or annotated pairs from natural language inference datasets. Khosla et al. (2020) extend the self-supervised batch contrastive approach to the fully-supervised setting to make full use of label information. Yeh et al. (2021) let the contrastive learning get rid of the dependence on large batch size. CoG-BART(Li et al., 2021) adapts supervised contrastive learning to make different emotions mutually exclusive to identify similar emotions better.

3 Methodology

3.1 Definition

Given a collection of all speakers $\mathcal{S}$, an emotion label set $\mathcal{E}$ and a conversation $\mathcal{C}$, our goal is to identify speaker's emotion label at each conversation turn. A conversation is denoted as $[(s_1, u_1), (s_2, u_2), \cdots, (s_N, u_N)]$, where $s_i \in \mathcal{S}$ is the speaker and u_i is the utterance of i-th turn. In this paper, we focus on the real-time settings of ERC, in which model can only take previous turns $[(s_1, u_1), (s_2, u_2), \cdots, (s_t, u_t)]$ as input to predict the emotion label y_t of t-th turn.

3.2 Context Modeling

We build a prompt-based context encoder upon SimCSE(Gao et al., 2021) to get speaker and context-aware emotion representations. The architecture of the context encoder is illustrated in Figure 3. To calculate representation for u_t, we use the most recent k turns of utterances and speakers as context.

$$C_t = [s_{t-k}, u_{t-k}, s_{t-k+1}, ..., s_t, u_t] \tag{1}$$

Kim and Vossen (2021) indicated that it is difficult for the pretrained language model to distinguish the "context" (i.e., $[s_{t-k}u_{t-k} \cdots s_{t-1}u_{t-1}]$) and target turn (i.e., s_t, u_t). Inspired by prompt learning(Liu et al., 2021), we construct a prompt for the t-th turn as follows.

$$P_t = \texttt{for } u_t, s_t \texttt{ fells <mask>} \tag{2}$$

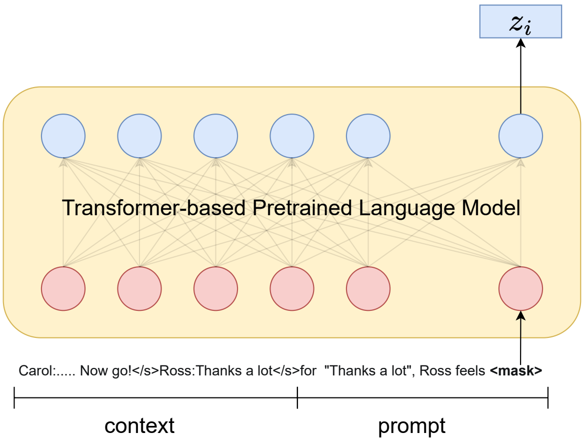

z_i
Transformer-based Pretrained Language Model
Carol:..... Now go!</s>Ross:Thanks a lot</s>for "Thanks a lot", Ross feels <mask>
context
prompt

Figure 3: The architecture of our prompt-based context encoder.

The full input of the encoder is $C_t \oplus P_t$, where $\oplus$ is the concatenation operation. In order to let the encoder realize that the prompt contains the target sentence, for the training pair of t-th turn $X_t^t = \{C_t \oplus P_t, y_t\}$, we construct an additional training pair $X_t^h = \{C_t \oplus P_h, y_h\}$, where h is randomly selected from $(t-k, \cdots, t)$. X_t^t and X_t^h shares the same context but has different prompts and labels. Training on such data helps the model to pay more attention to the target sentence and generate reasonable representations.

For a training pair X_t^k, we first feed $C_t \oplus P_k$ into the SimCSE model and get the last hidden states $H_t^k \in \mathbb{R}^{l \times d}$,

$$H_t^k = \mathrm{SimCSE}(C_t \oplus P_k) \tag{3}$$

where l is the number of tokens in $C_t \oplus P_k$, and d is the dimension of a token embedding. Then we use the embeddings of the special token <mask> from H_t^k as a representation of y_k-th emotion.

3.3 Supervised Prototypical Contrastive Learning for ERC

Supervised Contrastive Learning Supervised contrastive learning(Khosla et al., 2020) treats all examples with the same label in the batch as positive examples. A batch of N emotion representations generated via context encoder is denoted as $I = [z_1, z_2, \cdots, z_N]$. The vanilla supervised contrastive learning computes the loss $\mathcal{L}_i^{sup}$ for z_i as follows,

$$\mathcal{F}(z_i, z_j) = \exp(\mathcal{G}(z_i, z_j)/\tau) \tag{4}$$

$$\mathcal{N}_{sup}(i) = \sum_{z_j \in A(i)} \mathcal{F}(z_i, z_j) \tag{5}$$

$$\mathcal{P}_{sup}(i) = \sum_{z_p \in P(i)} \mathcal{F}(z_i, z_p) \tag{6}$$

$$\mathcal{L}_i^{sup} = -\log \frac{1}{|P(i)|} \frac{\mathcal{P}_{sup}(i)}{\mathcal{N}_{sup}(i)} \tag{7}$$

Here, $\mathcal{G}(z_i, z_j)$ is a score function that can be dot production, cosine similarity, etc. In our work, we use the cosine similarity for $\mathcal{G}(\cdot)$. $\tau \in \mathbb{R}^+$ is a scalar temperature parameter. $A(i) \equiv I \backslash \{z_i\}$ contains all representations in I except z_i, and $P(i)$ is the set of positive samples that have the same label with z_i in a batch.

Prototypical Contrastive Learning The object function $\mathcal{L}^{sup}$ introduces contrastive learning into supervised learning scenarios but suffers from class-imbalanced problem. Due to the limitations of batch size, samples from the majority class (e.g., `neutral` emotion) of the dataset may see insufficient negative samples within a batch. At the same time, it is hard for samples from the minority class to meet positive samples.

To solve this issue, we design a supervised prototypical contrastive learning (SPCL) loss function, which introduces prototype vectors of each category into the $\mathcal{L}^{sup}$ loss. First, we maintain a fixed-size representation queue for each emotion category. A representation queue for i-th emotion with size M is denoted as $Q_i = [z_1^i, z_2^i, \cdots, z_M^i]$. When a new representation $\hat{z}^i$ of i-th emotion is generated, we will first remove the oldest element in Q_i if $|Q_i|$ equals M, then detach the gradient of $\hat{z}^i$ and push it into Q_i. Second, to calculate the prototype vector for i-th category, we randomly select K samples from Q_i as the support set S_K, then take the mean of support set as the prototype vector $\mathbf{T}_i$.

$$S_K = \textsc{RandomSelect}(Q_i, K) \tag{8}$$

$$\mathbf{T}_i = \frac{1}{K} \sum_{z_j^i \in S_K, j \in [1 \ldots K]} z_j^i \tag{9}$$

We can get at most C_M^K different prototypes through sampling even if the representation queue is not updated.

After obtaining the prototype vectors, we treat each of them as an example of the corresponding category, so the sum of negative scores of z_i can be calculated as follows,

$$\mathcal{N}_{spcl}(i) = \mathcal{N}_{sup}(i) + \sum_{k \in \mathcal{E} \backslash y_i} \mathcal{F}(z_i, \mathbf{T}_k) \tag{10}$$

where y_i is the emotion label of i-th sample. Simultaneously, the sum of positive scores of z_i is also computed with the corresponding prototype vector.

$$\mathcal{P}_{spcl}(i) = \mathcal{P}_{sup}(i) + \mathcal{F}(z_i, \mathbf{T}_{y_i}) \tag{11}$$

Based on Eq.(10) and Eq.(11), the SPCL loss can be formulated as follows,

$$\mathcal{L}_i^{spcl} = -\log \left(\frac{1}{|P(i)| + 1} \cdot \frac{\mathcal{P}_{spcl}(i)}{\mathcal{N}_{spcl}(i)} \right) \tag{12}$$

The total SPCL loss of a batch is as follows,

$$\mathcal{L}^{spcl} = \sum_{i=1}^{N} \mathcal{L}_i^{spcl} \tag{13}$$

In summary, by introducing the prototype vectors, the SPCL loss ensures that there are at least one positive pair and $|\mathcal{E}| - 1$ negative pairs for each sample in a batch.

3.4 Curriculum Learning

Existing ERC datasets are usually collected in a multi-modal fashion. When building a text-only ERC model, some utterances are not informative enough to judge the emotions. Training the model with these extreme samples will lead to performance degradation. In this paper, we try to use curriculum learning to alleviate this issue.

Difficulty Measure Function To combine with contrastive learning, we propose a difficulty measure function based on the distance between classes. Let the total size of training set $\mathcal{D}_{train}$ as L, the emotion representation of i-th data sample as z_i, and the label of i-th data sample as y_i. Before each training epoch, we first compute z_i for all samples, then the center of k-th emotion is computed as follows,

$$\mathbf{C}_k = \frac{1}{|\{z_j | \forall j, y_j = k\}|} \sum_{j=1}^{L} z_j \cdot \mathbb{I}(y_j = k) \tag{14}$$

The difficulty of i-th sample $\mathcal{DIF}(i)$ is calculate as follows,

$$\mathcal{DIF}(i) = \frac{\mathrm{dis}(z_i, \mathbf{C}_{y_i})}{\sum_{j=1}^{|\mathcal{E}|} \mathrm{dis}(z_i, \mathbf{C}_j)} \tag{15}$$

dis function here is cosine distance. This function has the following two properties:

- The closer the sample is to the category center, the lower the difficulty.
- For two samples with the same distance from the center within the category, the further away from the center of other categories, the lower the difficulty.

Curriculum Strategy After sorting the entire training set, instead of directly splitting the training set, we design a sampling-based approach to construct a series of subsets ranging from easy to hard. Let R as the number of training epochs, to train the model at k-th epoch, we first generate a arithmetic progression a with a length of L, where $a_1 = 1 - k/R$ and $a_L = k/R$. Then we initialize a Bernoulli distribution with a and draw a binary random array R_B from it. We use B to draw a subset $\mathcal{D}_{sub-k}$ from training set for the current epoch, where $\mathcal{D}_{sub-k} \equiv \{x_i \in \mathcal{D}_{train} | R_{Bi} = 1\}$. Obviously, $\mathcal{D}_{sub-0}$ mainly consists of easy samples and $\mathcal{D}_{sub-R}$ mainly consists of hard samples. Compared to splitting the training set sequentially, the sampling-based approach provides a smoother difficulty variation for the model. The curriculum strategy is illustrated in $line2 - line9$ of Algorithm 1. We conduct a qualitative analysis of our curriculum learning strategy in Section 5.5.

3.5 Training and Evaluation

Training The overall procedure of our proposed approach is illustrated in Algorithm 1. We first generate emotion representations for all samples in training set, then use them to compute difficulty for each sample. After sorting the training set based on difficulty, we sample a subset S_K and train the context encoder on S_K through the SPCL loss.

Evaluation Since we computed the center of each class $\mathbf{C}$ when calculating SPCL loss, we can directly obtain the prediction through matching the centers as follows,

$$p_m^{ic} = \frac{\mathcal{G}(z_i, \mathbf{C}_c)}{\sum_{k=1}^{|\mathcal{E}|} \mathcal{G}(z_i, \mathbf{C}_k)} \tag{16}$$

where p_m^{ic} indicates the probability that i-th sample belongs to category c, and the subscript m means p_m^{ic} is calculated through matching.

Algorithm 1 Training Process with SPCL and Curriculum Learning

```
Inputs: $\mathcal{D}_{train}$:the training set with size $L$
    $R$: the number of total epochs
    $K$: the size of support set $S_K$
    $M$: the context encoder
    $\mathcal{E}$: the label set
Outputs: the optimal model $M^*$
    category centers $\mathbf{C}_j, j \in 1..|\mathcal{E}|$
 1: for $k$=0 to $R$ do
 2:   $I = \{M(x_i), \forall x_i \in \mathcal{D}_{train}\}$
 3:   compute $\mathbf{C}_j, j \in 1..|\mathcal{E}|$ (Eq.14)
 4:   compute $\mathcal{DIF}(i), i = 1..L$ (Eq.15)
 5:   $\mathcal{D}_{train}$ = sort($\mathcal{D}_{train}$, $\mathcal{DIF}$)
 6:   $st = k/R, ed = 1 - k/R$
 7:   $a_1 = st, a_n = a_1 + (n-1) \cdot \frac{ed-st}{L-1}$
 8:   $R_B \sim \text{Bernoulli}(p = a)$
 9:   $\mathcal{D}_{sub-k} \equiv \{x_i \in \mathcal{D}_{train} | R_{Bi} = 1\}$
10:   $Q_j$ = [],$j \in 1..|\mathcal{E}|$
11:   for $batch \in \mathcal{D}_{sub-k}$ do
12:     UPDATE($Q_j$), $j \in 1..|\mathcal{E}|$
13:     $S_{Kj} = \text{RANDOMSELECT}(Q_j, K)$
14:     compute prototype $\mathbf{T}_j$, $j \in 1..|\mathcal{E}|$ (Eq.9)
15:     compute SPCL loss (Eq.10-Eq.13)
16:     optimize($M$)
17:   end for
18: end for
19: return $M^*, \mathbf{C}_j, j \in 1..|\mathcal{E}|$
```

For comparison, we train an additional linear layer to predict the labels using cross-entropy loss,

$$p_l^i = \mathrm{W} \cdot z_i + b \tag{17}$$

$$\mathcal{L}_{CE} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{\mathcal{E}} y_{ic} \cdot \log p_l^{ic} \tag{18}$$

where $\mathrm{W} \in \mathbb{R}^{dim \times |\mathcal{E}|}$ is a trainable parameter. The gradient of z_i is detached so the model is only optimized via contrastive learning loss.

In this paper, we use p_m^i to predict labels when SPCL is the loss function and use p_l^i for other cases.

4 Experimental Settings

4.1 Experimental Setup

The code framework and initial weight of SimCSE come from Huggingface's Transformers(Wolf et al., 2020). We use the AdamW optimizer and cosine learning rate schedule strategy. When constructing training samples, we restrict their length to less than 256. We search the hyper-parameters on the develop set. For all experiments in this paper, we keep the best checkpoint on the develop set, then report the results on the test set using the

Models	IEMOCAP	MELD	EmoryNLP
COSMIC(Ghosal et al., 2020)	65.28	65.21	38.11
DialogueCRN (Hu et al., 2021)	66.46	63.42	38.91
DAG-ERC (Shen et al., 2021)	68.03	63.65	39.02
TODKAT (Zhu et al., 2021)	61.33	65.47	38.69
Cog-BART (Li et al., 2021)	66.18	64.81	39.04
TUCORE-GCN_RoBERTa(Lee and Choi, 2021)	-	65.36	39.24
SGED + DAG-ERC(Bao et al., 2022)	68.53	65.46	40.24
EmotonFlow-Large (Song et al., 2022)	-	66.50	-
CoMPM (Lee and Lee, 2021)	69.46	66.52	38.93
SPCL-CL-ERC(Ours)	**69.74**	**67.25**	**40.94**

Table 1: Performance comparisons on three datasets.

	MELD	IEMOCAP	EmoryNLP
No.Dials	1,432	151	827
Train	1,038	100	659
Dev	114	20	89
Test	280	31	79
No.Uttrs	13,708	7,333	9,489
Train	9,989	4,810	7,551
Dev	1,109	1,000	954
Test	2,610	1,523	984
No.CLS	7	6	7

Table 2: Statistics of the three datasets.

kept checkpoint. All experiments are conducted on Nvidia V100 GPU.

4.2 Datasets

We conduct experiments on three widely used benchmarks: MELD(Poria et al., 2019a), EmoryNLP(Zahiri and Choi, 2018) and IEMOCAP(Busso et al., 2008).

MELD This dataset has more than 1400 dialogues and 13000 utterances from Friends TV series. Multiple speakers participated in the dialogues. Each utterance in a dialogue has been labeled by any of these seven emotions – Anger, Disgust, Sadness, Joy, Neutral, Surprise and Fear.

EmoryNLP This dataset comprises 97 episodes, 897 scenes, and 12,606 utterances, where each utterance is annotated with one of the seven emotions borrowed from the six primary emotions in the Willcox's feeling wheel(Willcox, 1982), i.e., Sad, Mad, Scared, Powerful, Peaceful, Joyful, and a default emotion of Neutral.

IEMOCAP This dataset consists of 151 videos of recorded dialogues, with 2 speakers per session for a total of 302 videos across the dataset. Each segment is annotated for the presence of 9 emotions (Angry, Excited, Fear, Sad, Surprised, Frustrated, Happy, Disappointed and Neutral). The dataset is recorded across 5 sessions with 5 pairs of speakers.

The Statistics of these datasets are listed in Table 2. No.dials stands for the number of dialogues while No.uttrs stands for the total number of utterances in the dataset. No.CLS is the number of different emotions in the dataset.

4.3 Metrics

From Figure 2 we can see class-imbalance in all three benchmarks, so we use weighted-F1 score as the metric for all experiments in this paper.

5 Results and Analysis

5.1 Main Results

We compare our proposed approach with state-of-the-art text-based ERC methods, and the results are presented in Table 1. We can see that combining our proposed SPCL and curriculum learning strategy, we achieves state-of-the-art results on three benchmarks, which outperform previous SOTAs by 0.28%(CoMPM on IEMOCAP), 0.73%(CoMPM on MELD) and 0.7%(SGED + DAG-ERC on EmoryNLP).

5.2 Ablation Study

To evaluate the individual effects of SPCL and CL, we conducted a series of ablation experiments, and the results are shown in Table 3. The first line in Table 3 shows the performances of our proposed

	IEMOCAP	MELD	EmoryNLP
CE	68.35	65.33	38.72
+ CL	67.40	65.63	39.00
SupCon	68.13	65.67	39.20
+ CL	68.64	66.15	39.49
SPCL	69.03	66.56	40.14
+ CL	69.74	67.25	40.94

Table 3: Results of ablation study. Here, CE means Cross-entropy loss, SupCon is the vanilla supervised contrastive learning loss and SPCL is our proposed supervised prototypical contrastive learning loss. CL is our proposed curriculum strategy.

prompt-based context encoder trained with cross-entropy loss, which is our baseline model. We notice that curriculum learning didn't help a lot with cross-entropy loss. We believe that it is because we use the cosine distance in the difficulty measure function $\mathcal{DIF}$. However, it is unreasonable to compute cosine distance directly on representations optimized via cross-entropy loss.

The SupCon loss performs better than cross-entropy loss on MELD and EmoryNLP datasets but slightly worse than on the IEMOCAP dataset. Combining the three results, we can see no significant performance gap between SupCon and cross-entropy losses. But the combination of curriculum learning and SupCon(SupCon+CL) shows consistent superiority since SupCon uses cosine similarity as the score function. For the representations generated by SupCon, the cosine distance between representations of the same category will be closer, and the distance between representations of different categories will be distant. Therefore, the difficulty measure function of CL can be more faithful, resulting in better performance.

The SPCL loss outperforms SupCon and cross-entropy losses on all three datasets. Meanwhile, it also has consistent performance improvements in combination with curriculum learning.

To summary, both our proposed SPCL and curriculum learning strategy contribute significantly to the results.

5.3 Using SPCL on Class Imbalanced Data

To demonstrate the superiority of SPCL on imbalanced data, we construct an imbalanced subset from MELD training set, as illustrated in Figure 4. We sample 1024, 128, 64, 32, 32, 32, and 32 samples of `neutral`, `joy`, `surprise`, `anger`, `sadness`, `disgust`, and `fear`, respectively. We train the model using these two loss functions on the imbalanced training set. Since a small batch size will aggravate the impact of class-imbalance on contrastive learning, we conducted experiments with batch size of 4, 8, 16, and 32, respectively. The results on MELD test set are shown in Table 4. We notice that using SPCL outperforms using SupCon in all four sets of experiments. As the batch size decreases from 32 to 4, the weighted-F1 score of SupCon loss drops 6.95% while SPCL drops 4.1%. We can conclude that in the class-imbalance scenarios: 1) both SupCon and SPCL need a larger batch size to reach satisfied performances; 2) introducing the prototypical network into contrastive learning can alleviate the impact of class-imbalance.

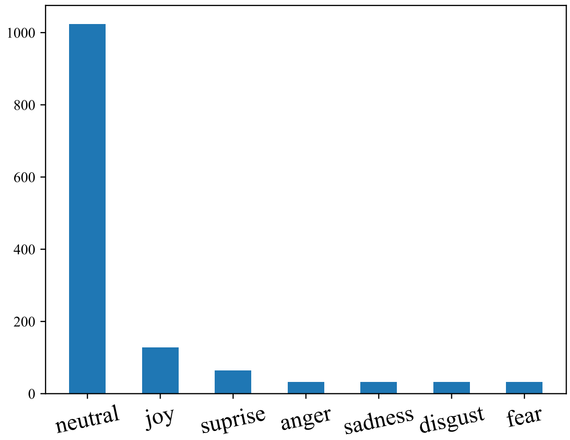


Figure 4: Emotion distribution of the extreme class-imbalanced training set. We construct it from MELD training set.

	4	8	16	32
SupCon	53.14	57.36	58.50	60.09
SPCL	57.27	58.85	59.47	61.38

Table 4: Results of different loss functions and different batch sizes trained on the imbalanced training set.

5.4 Using SPCL with Small Batch Size

Contrastive learning approaches usually need a large batch size to ensure more positive/negative pairs within a batch, which leads high computational cost. In the Section 5.3, we find that both SupCon and SPCL relay on large batch sizes. We conjecture that SPCL's dependence on batch size may be because we sample too few data samples for some categories(i.e., 32 for `anger`, `sadness`,`disgust`, and `fear`) to compute reasonable prototypes at the beginning of the training.

In order to further investigate the effect of batch sizes on SPCL, we apply SPCL to a more general scenario. From Figure 2 we can see the IEMOCAP dataset is not extreme class-imbalanced. Even the smallest category (happy) still has hundreds of samples. We conduct experiments on the IEMOCAP dataset, and the results are illustrated in Table 5. Experimental results show that given enough samples for each category, the SupCon loss still needs a large batch size, but the SPCL doesn't. With the batch size decreasing from 32 to 4, the performance of SupCon drops 5.63%, while SPCL only drops 0.82%.This demonstrates that the SPCL loss is less sensitive to the training batch size.

	4	8	16	32
SupCon	62.50	65.01	67.04	68.13
SPCL	68.21	68.41	68.48	69.03

Table 5: Results of SupCon and SPCL with different batch sizes on IEMOCAP dataset.

5.5 Qualitative Analysis of Curriculum Learning

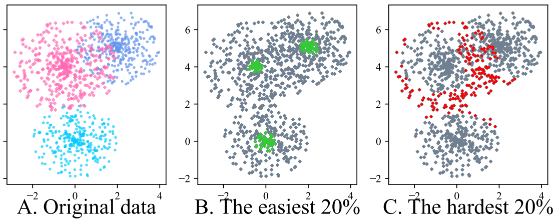
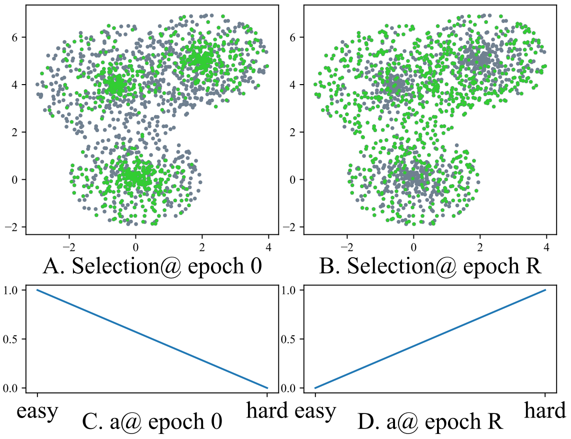

A. Original data B. The easiest 20% C. The hardest 20%

Figure 5: Visualizations of how the difficulty measure function $\mathcal{DIF}$ in Eq.(15) ranks the data.

To conduct a qualitative analysis of our proposed curriculum learning strategy, we generate a toy dataset that contains three classes and visualize it in Figure 5(A).

As illustrated in Figure 5(B-C), the difficulty measure function $\mathcal{DIF}$ ranks the samples in a reasonable way. The easiest 20% samples(in green) are distributed in the center of their respective categories, while the hardest 20% samples(in red) are mainly on the boundaries between classes.

In practice, we found that directly sorting the data with $\mathcal{DIF}$ cannot obtain satisfactory results. The model will overfit on simple samples in the early stage of training and produce large losses in the later stage, so we design the sampling-based curriculum learning strategy described in Section 3.4 to provide a smoother difficulty variation for the model. We control the difficulty of training subsets based on the sampling probability of samples with different difficulties. As illustrated in Figure 6, the arithmetic progression a we used to sample downs from 1 to 0 at the first epoch, the sampling results are shown in Figure 6(A), we can find that the most selected samples(in green) are around the centers while a few samples are away from the centers. When running to the last epoch, a grows from 0 to 1, so hard samples are in the majority of the subset, as shown in Figure 6(B).


A. Selection@ epoch 0 B. Selection@ epoch R

C. a@ epoch 0 D. a@ epoch R

Figure 6: The sampling-based curriculum strategy.

6 Conclusion

In this paper, we propose a novel loss function called **S**upervised **P**rototypical **C**ontrastive **L**earning (**SPCL**) loss for the emotion recognition in conversation task. Combining with Prototypical Network, the SPCL loss outperforms the traditional supervised contrastive learning loss. It also works well on class-imbalanced data and is less sensitive to the training batch size, which reduces the requirement of computing resource. To further exploit the power of contrastive learning on ERC tasks, we design a distance-based difficulty measure function and introduce curriculum learning to alleviate the impact of extreme samples. We conduct experiments on three widely used benchmarks: IEMOCAP, MELD, and EmoryNLP. Results show that our approach achieves state-of-the-art performance on all three datasets.

Limitations

This work has three limitations: 1) We introduce too many hyperparameters, which requires additional computing resources to search. 2) Our proposed difficulty measure function can not be combined with most existing ERC methods since it requires the emotion representations produced by the ERC model to be distance-aware. 3) We used multiple random sampling, resulting in unstable performance. The results in this paper are averaged with multiple seeds. In practice, we found that the results generated by different seeds may have significant variance.

References

Yinan Bao, Qianwen Ma, Lingwei Wei, Wei Zhou, and Songlin Hu. 2022. Speaker-guided encoder-decoder framework for emotion recognition in conversation. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 4051–4057.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. 2008. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42(4):335–359.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Deepanway Ghosal, Navonil Majumder, Alexander Gelbukh, Rada Mihalcea, and Soujanya Poria. 2020. Cosmic: Commonsense knowledge for emotion identification in conversations. *arXiv preprint arXiv:2010.02795*.

Devamanyu Hazarika, Soujanya Poria, Rada Mihalcea, Erik Cambria, and Roger Zimmermann. 2018. Icon: Interactive conversational memory network for multimodal emotion detection. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2594–2604.

Dou Hu, Lingwei Wei, and Xiaoyong Huai. 2021. Dialoguecrn: Contextual reasoning networks for emotion recognition in conversations. 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)*, pages 7042–7052.

Taichi Ishiwatari, Yuki Yasuda, Taro Miyazaki, and Jun Goto. 2020. Relation-aware graph attention networks with relational position encodings for emotion recognition in conversations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7360–7370.

Wenxiang Jiao, Haiqin Yang, Irwin King, and Michael R Lyu. 2019. Higru: Hierarchical gated recurrent units for utterance-level emotion recognition. In *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)*, pages 397–406.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673.

Taewoon Kim and Piek Vossen. 2021. Emoberta: Speaker-aware emotion recognition in conversation with roberta. *arXiv preprint arXiv:2108.12009*.

Bongseok Lee and Yong Suk Choi. 2021. Graph based network with contextualized representations of turns in dialogue. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 443–455.

Joosung Lee and Wooin Lee. 2021. Compm: Context modeling with speaker's pre-trained memory tracking for emotion recognition in conversation. *arXiv preprint arXiv:2108.11626*.

Jingye Li, Meishan Zhang, Donghong Ji, and Yijiang Liu. 2020. Multi-task learning with auxiliary speaker identification for conversational emotion recognition. *arXiv preprint arXiv:2003.01478*.

Shimin Li, Hang Yan, and Xipeng Qiu. 2021. Contrast and generation make bart a good dialogue emotion recognizer. *arXiv preprint arXiv:2112.11202*.

Hugo Liu and Push Singh. 2004. Conceptnet—a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Navonil Majumder, Soujanya Poria, Devamanyu Hazarika, Rada Mihalcea, Alexander Gelbukh, and Erik Cambria. 2019. Dialoguernn: An attentive rnn for emotion detection in conversations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6818–6825.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2019a. Meld: A multimodal multi-party dataset for emotion recognition in conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 527–536.

Soujanya Poria, Navonil Majumder, Rada Mihalcea, and Eduard Hovy. 2019b. Emotion recognition in conversation: Research challenges, datasets, and recent advances. *IEEE Access*, 7:100943–100953.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3027–3035.

Weizhou Shen, Siyue Wu, Yunyi Yang, and Xiaojun Quan. 2021. Directed acyclic graph network for conversational emotion recognition. *arXiv preprint arXiv:2105.12907*.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.

Xiaohui Song, Liangjun Zang, Rong Zhang, Songlin Hu, and Longtao Huang. 2022. Emotionflow: Capture the dialogue level emotion transitions. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8542–8546. IEEE.

Yan Wang, Jiayu Zhang, Jun Ma, Shaojun Wang, and Jing Xiao. 2020. Contextualized emotion recognition in conversation as sequence tagging. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 186–195.

Gloria Willcox. 1982. The feeling wheel: A tool for expanding awareness of emotions and increasing spontaneity and intimacy. *Transactional Analysis Journal*, 12(4):274–276.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Lin Yang, Yi Shen, Yue Mao, and Longjun Cai. 2021. Hybrid curriculum learning for emotion recognition in conversation. *arXiv preprint arXiv:2112.11718*.

Chun-Hsiao Yeh, Cheng-Yao Hong, Yen-Chi Hsu, Tyng-Luh Liu, Yubei Chen, and Yann LeCun. 2021. Decoupled contrastive learning. *arXiv preprint arXiv:2110.06848*.

Sayyed M Zahiri and Jinho D Choi. 2018. Emotion detection on tv show transcripts with sequence-based convolutional neural networks. In *Workshops at the thirty-second aaai conference on artificial intelligence*.

Dong Zhang, Liangqing Wu, Changlong Sun, Shoushan Li, Qiaoming Zhu, and Guodong Zhou. 2019. Modeling both context-and speaker-sensitive dependence for emotion detection in multi-speaker conversations. In *IJCAI*, pages 5415–5421.

Duzhen Zhang, Xiuyi Chen, Shuang Xu, and Bo Xu. 2020. Knowledge aware emotion recognition in textual conversations via multi-task incremental transformer. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4429–4440.

Peixiang Zhong, Di Wang, and Chunyan Miao. 2019. Knowledge-enriched transformer for emotion detection in textual conversations. 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)*, pages 165–176.

Lixing Zhu, Gabriele Pergola, Lin Gui, Deyu Zhou, and Yulan He. 2021. Topic-driven and knowledge-aware transformer for dialogue emotion detection. *arXiv preprint arXiv:2106.01071*.