Natural Language Processing (2026), **32**, pp. 322–356
doi:10.1017/nlp.2026.10013

CAMBRIDGE UNIVERSITY PRESS

ARTICLE

Supervised contrastive learning in few-shot soft prompt tuning

Ali Edalat and Yadollah Yaghoobzadeh

School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran
Corresponding author: Yadollah Yaghoobzadeh; Email: y.yaghoobzadeh@ut.ac.ir

(Received 10 August 2024; revised 11 July 2025; accepted 21 October 2025; first published online 16 April 2026)

Abstract

Soft prompt learning methods offer parameter-efficient tuning of pre-trained language models for few-shot scenarios. This study explores the integration of supervised contrastive learning (SCL) into two leading soft prompt tuning models: DifferentiAble pRompT (DART) and PTuning. By incorporating SCL as an auxiliary task, we observe consistent performance enhancements across 13 few-shot natural language understanding tasks, including benchmarks such as SST-2, TREC, MNLI, and real-world datasets such as Overruling, TC, and ADE. We also delve into SCL's impact in label-imbalanced settings, introducing a novel approach called balanced batch in SCL (BBSCL). BBSCL employs balanced mini-batches, sampling the majority class proportionally to the minority class to stabilize SCL calculations. Our results indicate that SCL and BBSCL significantly boost the performance and robustness of soft prompt learning models, especially on datasets with intricate label spaces. Experimentally, DART + SCL and PTuning + SCL outperform their base models by an average of 2.1% across the 13 tasks. Additionally, we find that SCL's contribution is more substantial in scenarios with complex and less separable label spaces. Compared to large language models such as GPT-3.5 and OpenChat, our enhanced soft prompt learning models with SCL and BBSCL extensions exhibit superior performance in both balanced and imbalanced few-shot settings. This research not only improves the effectiveness of few-shot tuning techniques but also deepens our understanding of this area.

Keywords: Supervised contrastive learning; few-shot learning; soft prompt learning; prompt tuning; imbalanced learning

1. Introduction

Large pretrained language models (PLMs) have revolutionized the field of few-shot learning in natural language understanding (NLU). Few-shot NLU involves the challenge of learning a new task with only a limited number of labeled examples. This scenario is practical and relevant, as obtaining a small number of annotations is often feasible and training on such data is efficient. To harness the power of PLMs for downstream tasks (domain adaptation), prompt-based paradigms have emerged. These paradigms involve transforming the task into an LM-like objective using prompts. Prompts can be categorized into two types: discrete and continuous.

Discrete prompts, or hard prompts, pose a challenge due to the need for manual expertise in crafting effective templates, especially for specific tasks. ChatGPT, an advanced language model developed by OpenAI, has showcased remarkable proficiency across various natural language tasks reliant on hard prompt learning strategies (OpenAI 2023). To streamline this process, researchers have delved into devising automatic methods for prompt design (Lampinen *et al.* 2022) and identifying the most informative few-shot examples (Wang, Yang, and Wei 2024a) within hard prompt learning frameworks.

In the realm of pre-trained masked language models, strategies like language model-based fine-tuning (LMBFF) (Gao, Fisch, and Chen 2021a), PET (Schick and Schütze 2021b) and AutoPrompt (Shin *et al.* 2020) have been introduced to unearth effective prompts. However, while these automated techniques offer promising avenues, they might not consistently pinpoint the optimal prompts and often necessitate a substantial amount of validation data to validate their efficacy.

Soft prompts, also referred to as continuous prompts (Liu *et al.* 2024), provide a means to simplify the search for effective prompts in downstream tasks. These prompts are learnable parameters that can be inserted into the input and optimized based on the task objectives and few-shot examples. Soft prompts do not require manual design or fixed input formats while they usually treat the underlying PLM as a black box. Using soft prompts gives us the extra advantage of avoiding the human bias injection that can be caused by manually designed prompts (Tian *et al.* 2023). Notably, two methods that leverage soft prompts that see PLM as a black box and achieve remarkable few-shot performance are differentiAble pRompT (DART) (Zhang *et al.* 2022) and PTuning (Liu *et al.* 2022). DART combines a DART mechanism with a fluency constraint objective, ensuring that the prompt tokens are compatible with the PLM's prior knowledge and preserve the LM information during task learning. We evaluate our contributions on DART and PTuning as our base models in this work because (i) their implementation is publicly available and (ii) their additional objectives are applied to the output representations and the inner layers of PLMs are not needed.

Contrastive learning is a technique that enhances the robustness of various models against biases and distribution shifts by creating robust representations (Chen *et al.* 2021; Choi *et al.* 2022; Pan *et al.* 2022; Sahoo *et al.* 2023; Du *et al.* 2024). Lately, supervised contrastive learning (SCL) (Gunel *et al.* 2020) has demonstrated improvements in supervised learning, especially in few-shot setups (Gunel *et al.* 2020; Jian, Gao, and Vosoughi 2022; Abaskohi, Rothe, and Yaghoobzadeh 2023). However, it has not yet been applied to soft prompt learning methods.

In this paper, we study the effects of incorporating SCL into our two soft prompt learning models, DART and PTuning. Our implementation of SCL does not require data augmentation but instead leverages the existing examples within a batch, following similar approaches to Gunel *et al.* (2020) and Zhang *et al.* (2022). Additionally, we propose a novel approach called balanced batch in SCL (BBSCL), which employs batch-balancing techniques to further enhance the robustness of SCL, particularly in label-imbalanced settings.

LLMs offer a valuable resource for data generation, particularly in tasks where human performance is outstanding and the models have been proficiently trained. However, they may struggle with generating suitable outputs for novel tasks (Møller *et al.* 2023). Data augmentation, while commonly used, can distort the original distribution of real data, potentially compromising the quality of model training (Suhaeni and Yong 2024). Considering the costs associated with LLM usage, including GPU and API expenses, as well as the limitations posed by models like ChatGPT, it becomes imperative to enhance model performance under imbalanced few-shot learning scenarios without resorting to data augmentation. This aspect underscores the significance of our work, as we seek to improve model robustness without resorting to conventional data augmentation techniques.

We conduct experiments on thirteen NLU tasks, namely SST-2, MR, CR, Subj, TREC, MNLI, SNLI, QNLI, MRPC, QQP, Overruling, TC, and ADE. Our experimental results consistently demonstrate that including SCL in soft prompt models leads to improved performance compared to our base models. To further evaluate the effectiveness of our contributions, we specifically assess the impact of adding SCL and the BBSCL extension on three few-shot label-imbalanced settings. Our findings indicate that SCL is effective in these scenarios as well, and the introduction of the BBSCL extension further enhances the robustness of the models. Moreover, we delve into the analysis of SCL's impact by examining the relationship between label complexity and its performance contribution. Notably, we observe that SCL tends to yield higher performance gains in cases where

https://doi.org/10.1017/nlp.2026.10013 Published online by Cambridge University Press

the label space is more complex and characterized by less separable classes. Our work makes the following key contributions:

- Integrating SCL into soft prompt tuning frameworks (DART and PTuning) for few-shot NLU tasks.
- Proposing BBSCL, a novel method to stabilize SCL in label-imbalanced settings through balanced mini-batches.
- Comprehensive empirical validation across 13 tasks, including real-world datasets, showing average gains of 2.1% over base models.
- Demonstrating that SCL's benefits are pronounced in complex label spaces and that small PLMs with our methods outperform large LLMs like GPT-3.5 in imbalanced scenarios.

2. Related work

2.1 Soft prompt learning

Soft prompt learning and its variations, such as prompt-tuning and residual prompt tuning, are emerging techniques aimed at enhancing the performance of PLMs with minimal parameter updates. These methods are designed to be both efficient and effective, leveraging the existing knowledge within large models to adapt to specific tasks.

Works like PTuningV2 (Liu *et al.* 2022), prefix-tuning (Li and Liang 2021), CodePrompt (Choi and Lee 2023), and SwitchPrompt (Goswami *et al.* 2023) use soft prompt learning, but they do not see PLMs as a black box. they add parameters in PLM layers for tuning.

Residual prompt tuning significantly enhances standard prompt tuning by incorporating a shallow network with a residual connection, improving performance and stability across various tasks, including the SuperGLUE benchmark. Additionally, it demonstrates reduced sensitivity to hyperparameters such as learning rate and prompt initialization (Razdaibiedina *et al.* 2023). However, this model does not assess robustness in imbalanced scenarios and is developed based on label text generation, differing from encoder PLMs. Therefore, our SCL extensions, which focus on input representation, are not accommodated by this model. Decomposed prompt tuning (DePT) (Shi and Lipani 2024) improves performance and reduces memory and time costs by over 20% compared to standard prompt tuning, while maintaining competitive performance with fewer trainable parameters. XPrompt (Ma *et al.* 2022), a novel prompt tuning model, bridges the performance gap between prompt tuning and fine-tuning at smaller model scales by eliminating negative prompt tokens through hierarchically structured pruning. Like residual prompt tuning, DePT and XPrompt do not evaluate model robustness in imbalanced scenarios and are also based on label text generation, which is distinct from encoder PLMs. Thus, it does not meet the requirements of our SCL extensions.

The transferability of trained soft prompts to similar tasks can accelerate training and improve performance, with cross-task transferability generally being more effective than cross-model transferability (Su *et al.* 2022). The performance of prompt tuning can be highly sensitive to the initialization of prompts, and traditional methods may not encode sufficient task-relevant information (Wu *et al.* 2023).

The meta soft prompt approach effectively generalizes PLMs across multiple domains, facilitating rapid adaptation to new domains through few-shot unsupervised domain adaptation (Chien, Chen, and Xue 2023). However, this work does not address the effects of SCL or the challenges posed by imbalanced scenarios on soft prompt learning.

MPrompt, a multilevel prompt tuning method, improves machine reading comprehension using task-specific, domain-specific, and context-specific prompts. This method achieves an average improvement over state-of-the-art methods (Chen *et al.* 2023a). While this method can be applied to text generation, our paper focuses on text classification.

https://doi.org/10.1017/nlp.2026.10013 Published online by Cambridge University Press

2.2 SCL in label-imbalanced few-shot text classification

SCL in label-imbalanced few-shot text classification is an approach that aims to improve the performance of models when only a small amount of labeled data is available, and there is an imbalance in the label distribution. This method leverages the limited data more effectively by focusing on learning representations that bring data points of the same class closer together while pushing apart data points from different classes.

SCL, as outlined by Lee and Yoo (2021), is a powerful tool for enriching few-shot learning. Refining the quality of feature extraction amplifies performance and curtails the necessity for expansive datasets, particularly amidst domain shifts, through the strategic use of data augmentation. Building upon this foundation, our paper pioneers an innovative method to mitigate the reliance on extensive datasets in the face of domain shifts while sidestepping the traditional route of data augmentation. In addition, we examine the impact of SCL on soft prompt learning which is not discussed in this paper.

Contrastive learning frameworks like ContrastNet offer a promising solution to the challenges of discriminative representation and overfitting in few-shot text classification. By strategically pulling similar class representations together while pushing apart dissimilar ones, ContrastNet effectively navigates the complexities of data distribution. Chen *et al.* (2022) demonstrated its efficacy, particularly in addressing overfitting through unsupervised regularization techniques. Notably, while ContrastNet utilized data augmentation strategies, our BBSCL extensions tailored for imbalanced datasets take a different approach, omitting augmentation while still achieving notable performance enhancements.

Semi-supervised contrastive learning (SSCL) uses pseudo-labels for data augmentation and a two-step contrastive scheme to improve deep model performance in few-shot text classification, mitigating the impact of pseudo-label noise (Wang *et al.* 2022) . Contrastive learning on web data for few-shot classification can be efficient by using pseudo-labels for data augmentation and a normalization strategy to enhance the generalization ability of learned representations (Li *et al.* 2023). Using a label-indicative component and hard negative mixing strategy, effectively addresses data imbalance in text classification tasks by constructing contrastive samples, resulting in superior text representations and noise-resistant representation learning (Chen *et al.* 2023b).

2.3 Contrastive learning in soft prompt learning

PromCSE trains small-scale Soft Prompts while keeping PLMs fixed, which helps mitigate performance issues under domain shift settings (Jiang, Zhang, and Wang 2022). This work does not investigate the model's robustness against label imbalances in the training data. Instead, this work adds soft prompts to each layer of the PLM, treating the PLM as a white-box architecture.

Soft prompt learning can be used for creating text representations in the hidden space for better planning long text generation. Contrastive soft prompt (CSP) (Chen *et al.* 2022a) model is proposed for improving the coherence of long text generation with this aspect. In this paper, we focus on NLU tasks that use soft prompts to relate input text and labels that is different from CSP model usage.

Soft and hard prompts can be used in the mixture. ContrastNER (Layegh *et al.* 2023) is a prompt-based NER framework that employs both discrete and continuous tokens in prompts and uses a contrastive learning approach to learn the continuous prompts and forecast entity types. In this paper, we focus on analyzing the impact of SCL on soft prompt learning, and we use only NLU tasks based on our interesting soft learning models' benchmarks in the literature. In addition, ContrastNER does not analyze the robustness of the model in imbalance scenarios.

2.4 Soft prompt learning in imbalanced dataset

Soft-prompt tuning presents an efficient alternative to standard model fine-tuning for predicting lung cancer using Dutch primary care medical notes, demonstrating its effectiveness in both balanced and imbalanced scenarios (Elfrink *et al.* 2023). However, this study does not aim to

https://doi.org/10.1017/nlp.2026.10013 Published online by Cambridge University Press

enhance the performance of soft prompts specifically in imbalanced scenarios; rather, it focuses on investigating the performance of soft prompts within the context of limited datasets.

Xu *et al.* (2024), focus on investigating and mitigating the impact of prompt bias, especially in imbalanced datasets, to enhance the reliability of factual knowledge extraction in PLMs. Their work specifically examines soft prompt learning in these imbalanced scenarios to address prompt bias, but it does not explore the robustness of soft prompts in such contexts.

LMPT (Xia *et al.* 2023) significantly enhances long-tailed multi-label visual recognition performance through the innovative use of prompt tuning and specialized loss functions. It maintains robustness against class imbalances with a class-specific embedding (CSE) loss and a distribution-balanced (DB) classification loss. Prompt tuning utilizes learnable prompt tokens with frozen pretrained model parameters, which enhances semantic feature interactions. The combination of CSE and DB losses effectively addresses label imbalances, improving performance for both head and tail classes. Notably, this approach achieves superior results without relying on contrastive learning, thereby preserving model robustness in the face of label imbalances.

Imbalanced learning algorithms, including focal loss, balanced SoftMax, and distribution alignment, have shown considerable promise in enhancing the performance of vision-language models on imbalanced datasets (Wang *et al.* 2024b). While this work highlights the limitations of soft prompt learning in label-imbalanced scenarios for vision-language models, it does not examine the influence of contrastive learning on these models. Furthermore, the impact of contrastive learning on the soft prompt learning of vision-language models within these imbalanced contexts remains unexplored.

3. Background

In continuous prompt learning, trainable tokens are introduced to the input text (X) to form a prompted template (Equation 1). The objective is to guide the PLM in generating a textual output that corresponds to a specific class (label word w). The MLM prediction calculates the probability distribution $p([MASK] = y|X_{prompt})$. The trainable tokens are updated through backpropagation.

$$X_{prompt} = [CLS]X[T_1][T_2]\ldots[T_n][MASK][SEP] \tag{1}$$

Both the PTuning and DART models employ continuous prompt learning for their primary task, referred to as the class discrimination objective (L_{CE}), as illustrated in the "Class Discrimination" section of Figure 1. This primary task aims to maximize the probability of the correct class tokens and minimize the probability of the incorrect ones. Given an example (X) and its true label token ($y^\star$), we can express L_{CE} using cross entropy as follows:

$$L_{CE} = CE(q(y^\star|X), y^\star) - \sum_{y \neq y^\star} CE(q(y|X), y) \tag{2}$$

$$q(y|X) = p([MASK] = y|X_{prompt}) \tag{3}$$

Additionally, the DART model incorporates an auxiliary task called the fluency constraint objective (L_{MLM}). This objective ensures the coherence of the trainable tokens and preserves the NLU capability of the model. It is depicted in the "fluency constraint objective" section of Figure 1. The trainable prompt template tokens should be interdependent. In this auxiliary task, some tokens of the input text (X) are randomly masked (Equation 4). The gold label is used for filling in the label mask in X_{MLM} (X' is X with some masked tokens).

$$X_{MLM} = [CLS]X'[T_1][T_2]\ldots[T_n]Y.[SEP] \tag{4}$$

https://doi.org/10.1017/nlp.2026.10013 Published online by Cambridge University Press

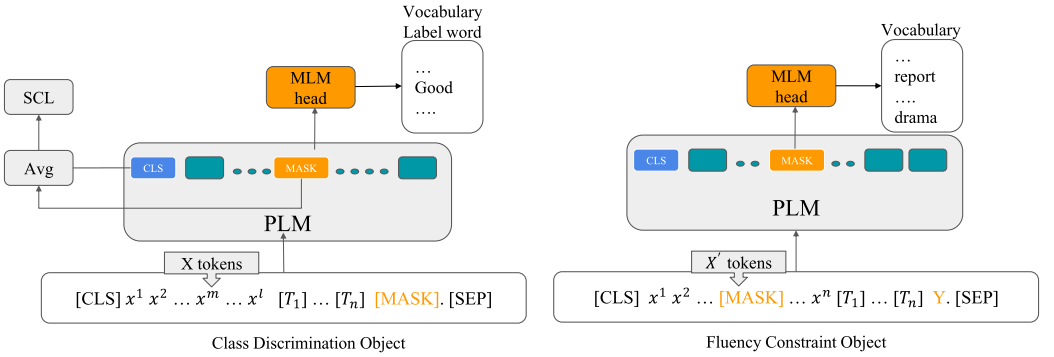

Figure 1. DART + SCL consists of two components: the DART model and the SCL module. The DART model optimizes two objectives: the "Class Discrimination Objective," which learns to distinguish between different text classes , and the "Fluency Constraint Objective." The SCL module uses the "Class Discrimination Objective" to encode text into a vector representation, by averaging the hidden state of the [CLS] token and the label. The left side of this figure shows the PTuning + SCL model, which consists of two components: the PTuning model with the "Class Discrimination Objective" and the SCL module.

The MLM prediction calculates the probability distribution $p([MASK] = x^m|X_{MLM})$, which determines the replacement of the mask with the original token (x^m denotes the masked token). The DART's auxiliary task aims to maximize the probability of choosing x^m for each mask token (L_{MLM}). We can define this task as:

$$L_{MLM} = \sum_{m \in M} BCE\big(p\big(x^m|X_{MLM}\big)\big) \tag{5}$$

4. Adding SCL to DART and PTuning

To enhance the DART and PTuning models, we introduce an SCL module, resulting in the DART + SCL and PTuning + SCL models. The SCL module is incorporated by adding an SCL loss as an additional objective function to optimize during training. SCL operates on the aggregated representation of an example, denoted as $\Phi(X)$, which is obtained by averaging the hidden states of the [CLS] token and the label mask tokens. In principle, it is not dependent on any specific architecture or prompt tuning method.

The SCL module is designed based on the approach outlined by Chen *et al.* (2022c). Its purpose is to bring examples of the same class closer in the representation space while pushing examples from different classes further apart. By incorporating SCL, we aim to enhance the discriminative capabilities of the models and improve their ability to capture class-related information. The SCL module is illustrated in Figure 1. We define the objective of this task L_{SCL} following Gunel *et al.* (2020). SCL loss mathematical definition is in Equation (6). We can follow Cui *et al.* (2021) and Zhu *et al.* (2022)' paper state about the imbalance impact on SCL by examining the SCL definition.

$$L_{SCL} = \sum_{i=1}^{N} -\frac{1}{N_{y_i} - 1} L_{SCL}(x_i) \tag{6}$$

The SCL loss aims to cluster representations of the same class while distancing different classes. For each example x_i, we compute similarities with other examples in the batch. Positive pairs (same class) are pulled closer, while negative pairs (different classes) are pushed apart. Formally, the loss for each example is defined as:

https://doi.org/10.1017/nlp.2026.10013 Published online by Cambridge University Press

$$L_{SCL}(x_i) = \sum_{j \in P(i)} \log \frac{\exp\left(\Phi(x_i) \cdot \Phi(x_j)/\tau\right)}{\sum_{k \in A(i)} \exp\left(\Phi(x_i) \cdot \Phi(x_k)/\tau\right)} \tag{7}$$

$$P(i) = \{j | 1 \leq j \leq N \wedge i \neq j \wedge y_i = y_j\} \tag{8}$$

$$A(i) = \{k | 1 \leq k \leq N \wedge i \neq k\} \tag{9}$$

where $P(i)$ and $A(i)$ denote positive and all pairs for x_i, respectively. BBSCL modifies this by grouping majority-class examples into balanced mini-batches to reduce bias. The total loss values are in Equations (10) and (11). We set $\alpha = \frac{\lambda}{2}$ and $\beta = \frac{1}{2}$ in our implementation. For multi-class tasks like TREC, we follow the one-vs-all approach as SCL is designed for binary tasks.

$$L_{DART+SCL} = L_{CE} + \lambda L_{MLM} + \alpha L_{SCL} \tag{10}$$

$$L_{PTuning+SCL} = L_{CE} + \beta L_{SCL} \tag{11}$$

4.1 Balanced batch in SCL (BBSCL)

Cui *et al.* (2021) and Zhu *et al.* (2022) show that high-frequency classes dominate the overall SCL loss value in imbalanced learning. In the original SCL, for each training batch, the distance between each pair of examples in the same class is computed. For a batch with two positives and four negatives, this gives two positive and six negative distances, which means four more contributions from the majority class here. The other eight positive to negative distances are balanced for each class. These distances then shape the SCL loss.

We propose BBSCL to solve the class imbalance problem in SCL. First, we divide the N examples in the majority class into K groups, each with the same number of examples as the minority class size (i.e., P). So K = N/P. These groups are therefore label-balanced. Then, we compute the scl loss for each of the K groups separately, between all the examples in the minority class and the group examples. Finally, we sum up the losses. These groups can be completely separate or have shared examples. The former creates the loss named "BBSCL (partition)" and the latter creates "BBSCL (random)". In creating BBSCL (random) balanced groups, we sample from the majority class with replacement. For our mentioned batch with two positives and four negatives, we make two balanced groups with two positives and two negatives each. The number of SCL distances is equal for each class in the new mini-batches, making the SCL objective unbiased and better for imbalanced learning. When the minority class size is zero, we use the average of all the minority class representations till that moment as the minority class data in the batch (comparison of SCL and BBSCL in Figures 2 and 3).

4.2 Clarification of sampling in BBSCL

In Section 4.1 , we proposed two variants of balanced-batch SCL (BBSCL): BBSCL (partition) and BBSCL (random). For clarity: BBSCL(partition) constructs balanced groups without replacement – the majority-class pool is partitioned into disjoint groups, each of size equal to the minority class (i.e., $K = \left\lceil \frac{N_{\text{maj}}}{P} \right\rceil$ groups), so that no majority example appears in more than one balanced group for a given epoch. By contrast, BBSCL(random) constructs each balanced group by sampling with replacement from the majority class: for every balanced group, we draw P majority examples uniformly (possibly repeating examples across different groups), then compute the SCL loss for that balanced pairwise comparison and sum across groups. The partition variant therefore guaranties disjointness of group membership (useful to avoid repeated emphasis on a small subset of majority examples), while the random (with-replacement) variant allows re-use of majority

https://doi.org/10.1017/nlp.2026.10013 Published online by Cambridge University Press

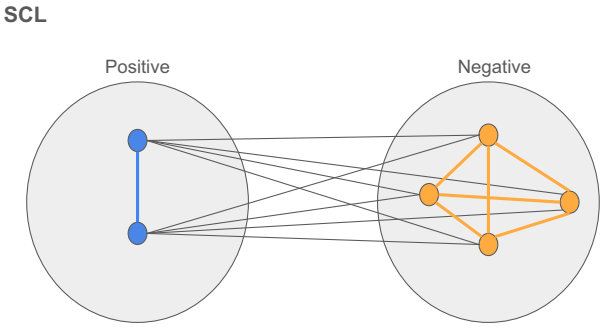

Figure 2. SCL loss calculation for imbalanced batch with two positive and four negative samples.

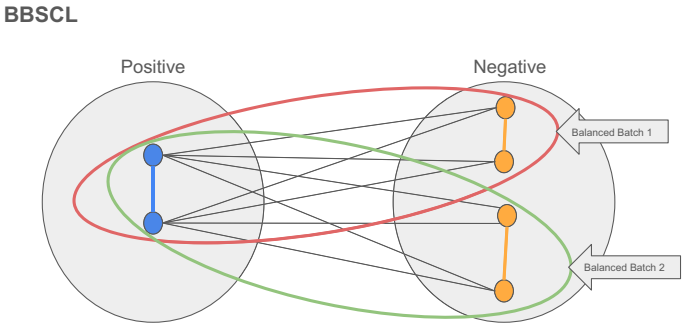

Figure 3. BBSCL loss calculation for imbalanced batch with two positive and four negative samples.

examples across groups and is particularly convenient when N_{maj} is not an integer multiple of P or when one wants to simulate repeated draws (this distinction is the same as the "partition vs. random" terminology used throughout the results section). Algorithmically, both variants implement the same loss aggregation (sum of contrastive losses computed on each balanced group); they only differ in how majority examples are selected for each group (see Section 4.1 for the loss formulation).

5. Experimental setup

5.1 Datasets

For our main evaluation, we utilize 10 sentence classification tasks, namely SST-2, MR, CR, Subj, TREC, MNLI, SNLI, QNLI, MRPC, and QQP. These tasks are evaluated in the balanced few-shot setting with $k = 16$, as defined by LMBFF (Gao, Fisch, and Chen 2021a). This evaluation allows us to assess the effectiveness of our proposed approach across a range of text classification benchmarks.

To further evaluate the performance of our approach in more challenging scenarios with complex label spaces and real-world datasets, we conduct experiments on three additional datasets: Overruling, TC, and ADE. These datasets are part of the RAFT benchmark. To ensure a comprehensive analysis, we create two splits for each dataset using random sampling. These splits are designed in the few-shot setting with k = 25.

https://doi.org/10.1017/nlp.2026.10013 Published online by Cambridge University Press

5.1.1 Real-world dataset construction

There is no development data in the raft datasets.[1] We take 100 balanced data from the test data as our development set for creating an imbalance setting and use the rest of the test data to evaluate the models. We use a combination of train data and our development set data (D_{com}) to generate different sampled train data. In our experiments on real-world datasets, we measure the average performance across two different sampled train data for each task. A fixed set of seeds ({*2021, 2022*}) is used for preparing two different sampled train data. For generating each sampled train data in the imbalanced setting $\rho^{+} = 0.25$ in these datasets, we sample twelve data from the positive class of D_{com} randomly. We also sample 38 data from the negative class in the same way. We use a seed value in the set of seeds for these two samplings. For balanced settings, we act like the imbalanced setting but we sample 25 data from each two positive and negative classes.

We convert these classification tasks to entailment problems (based on Wang *et al.* (2021)'s paper) for converting them to RTE tasks. We use the positive label's description (description based on work of Alex *et al.* (2021)) and input text for forming the RTE problem.

5.1.2 Create train data of SST-2 in the imbalanced few-shot setting

According to LMBFF (Gao, Fisch, and Chen 2021a), we generate training data for each seed in the few-shot setting for k = 8 (D_8) and $k = 24$ (D_{24}). For $\rho^{-} = 0.25$, we combine the data of the negative class in D_8 with the data of the positive class in D_{24} to obtain the training data for the desired seed in this imbalance condition.

5.2. Setup

We use the original DART[2] implementation for our DART + SCL, DART + BBSCL, PTuning + SCL, and PTuning+BBSCL models. We follow the same settings as DART (Zhang *et al.* 2022) for our few-shot experiments on 10 NLP datasets, as shown in Table 1. We average the performance over five sampled training datasets for each task. We select the best hyperparameters (i.e. learning rate, weight decay, and batch size) based on the development set using grid search. The search space is in Section 5.2.1. We use RoBERTa-large (Liu *et al.* 2019) and BERT-large-uncased (Devlin *et al.* 2019) as the base PLMs model for all tasks and train them on Kaggle's[3] free P100 GPU. For the Overruling, TC, and ADE datasets, we average the performance over two sampled training datasets. The sampling and entailment conversion methods are given in Section 5.1.1. We define ρ as the ratio of positive (or negative) examples to the total to measure the class imbalance.

To examine the effects of SCL and BBSCL on cross-entropy (CE) loss in standard fine-tuning of PLMs for domain adaptation, we utilize the implementation from Dual Contrastive Learning[4] paper (Chen *et al.* 2022). We report average performance on 5 different runs w.r.t different run random seed. In the few-shot balanced setting, we employ 16 examples per class, while in the imbalanced setting, we use 8 examples for the positive class and 24 examples for the negative class.

We also analyze the influence of training set size in imbalanced few-shot settings on our results. To achieve this, we conduct experiments using training set sizes that are significantly different from typical few-shot settings (According to the experiments from Schick and Schütze 2021b), the performance of fine-tuning and prompt learning is similar when the training size is 256. This suggests that this training size is significantly different from a few-shot setting, where prompt learning

[1] Train and unlabeled test: https://huggingface.co/datasets/ought/raft test labels: https://huggingface.co/datasets/ought/raft-submission

[2] https://github.com/zjunlp/DART

[3] https://www.kaggle.com/

[4] https://github.com/hiyouga/Dual-Contrastive-Learning

https://doi.org/10.1017/nlp.2026.10013 Published online by Cambridge University Press

Table 1. Results on ten few-shot datasets used in the DART paper which are standard in prompt tuning literature and for three real-world datasets (ADE, TC, and Overruling) in few-shot settings Mean ± std performances are reported. Models use RoBERTa-large. Adding SCL makes consistent improvements to the DART and PTuning model. We report the PTuning and DART from their original papers to ground our reproduced DART and PTuning results better. Our experimental setting GPU type is different from the DART paper. We focus on our reproduced results here to fairly evaluate our SCL and BBSCL extensions. Avg: average performance

Model	MRPC	TREC	SST-2	QNLI	CR	ADE	TC
PET (Schick and Schütze 2021b)	61.9	32.0	83.6	50.8	79.5	-	-
LMBFF (Gao, Fisch, and Chen 2021)	$76.2_{\pm 2.3}$	$88.2_{\pm 2.0}$	$92.3_{\pm 1.0}$	$68.3_{\pm 7.4}$	$89.0_{\pm 1.4}$	-	-
PTuning (Liu *et al.* 2022)	$78.6_{\pm 1.1}$	$90.5_{\pm 1.6}$	$92.4_{\pm 0.6}$	$67.6_{\pm 7.3}$	$91.1_{\pm 0.6}$	-	-
DART (Zhang *et al.* 2022)	$78.3_{\pm 4.5}$	$87.1_{\pm 3.8}$	$93.5_{\pm 0.5}$	$66.7_{\pm 3.7}$	$91.8_{\pm 0.5}$	-	-
PTuning	$78.2_{\pm 3.0}$	$88.9_{\pm 4.6}$	$88.8_{\pm 2.5}$	$67.8_{\pm 2.6}$	$89.7_{\pm 1.9}$	$49.6_{\pm 2.3}$	$47.5_{\pm 4.1}$
PTuning + SCL	$79.5_{\pm 3.3}$	$90.1_{\pm 3.7}$	$89.8_{\pm 2.6}$	$71.0_{\pm 1.6}$	$90.3_{\pm 1.1}$	$52.1_{\pm 0.6}$	$50.5_{\pm 0.2}$
DART	$78.3_{\pm 5.1}$	$87.8_{\pm 3.3}$	$92.0._{\pm 1.2}$	$66.4_{\pm 3.6}$	$91.1_{\pm 0.3}$	$52.0_{\pm 0.1}$	$47.1_{\pm 0.7}$
DART + SCL	$79.0_{\pm 3.8}$	$88.9_{\pm 3.1}$	$93.7_{\pm 0.3}$	$67.4_{\pm 3.4}$	$92.1_{\pm 0.8}$	$58.3_{\pm 0.9}$	$52.7_{\pm 5.5}$
Model	MR	SUBJ	SNLI	MNLI	QQP	Overruling	Avg
PET (Schick and Schütze 2021b)	80.8	51.4	49.5	50.8	49.7	-	-
LMBFF (Gao, Fisch, and Chen 2021)	$85.5_{\pm 2.8}$	$91.2_{\pm 1.1}$	$77.1_{\pm 2.1}$	$68.3_{\pm 2.5}$	$67.0_{\pm 3.0}$	-	-
PTuning (Liu *et al.* 2022)	$86.4_{\pm 1.5}$	$91.8_{\pm 0.8}$	$68.3_{\pm 7.3}$	$65.7_{\pm 4.0}$	$65.8_{\pm 3.9}$	-	-
DART (Zhang *et al.* 2022)	$88.2_{\pm 1.0}$	$90.7_{\pm 1.4}$	$75.8_{\pm 1.6}$	$67.5_{\pm 2.6}$	$67.8_{\pm 3.2}$	-	-
PTuning	$86.3_{\pm 1.9}$	$91.5_{\pm 1.2}$	$69.9_{\pm 2.6}$	$62.9_{\pm 2.4}$	$62.4_{\pm 3.3}$	$53.0_{\pm 0.6}$	72.0
PTuning + SCL	$87.3_{\pm 0.7}$	$91.9_{\pm 1.6}$	$73.1_{\pm 2.5}$	$68.1_{\pm 2.6}$	$64.7_{\pm 2.9}$	$54.4_{\pm 3.4}$	74.1
DART	$87.3_{\pm 0.7}$	$90.5_{\pm 0.7}$	$75.1_{\pm 1.5}$	$66.7_{\pm 2.7}$	$67.0_{\pm 1.7}$	$57.5_{\pm 1.0}$	73.8
DART + SCL	$88.3_{\pm 1.0}$	$93.0_{\pm 1.0}$	$76.6_{\pm 1.8}$	$69.8_{\pm 3.7}$	$68.2_{\pm 2.6}$	$58.1_{\pm 0.7}$	75.9

is expected to outperform fine-tuning in few-shot settings). When investigating the impact of SCL and BBSCL on fine-tuning, we utilize 256 training data in an imbalanced few-shot setting. Similarly, when examining the effect of SCL and BBSCL on soft prompt learning models, we employ 400 training data in an imbalanced few-shot setting.

5.2.1. Soft prompt learning's hyperparameter search space

Hyper-parameter search space of 10 NLP datasets (SST-2, MR, CR, Subj, TREC, MNLI, SNLI, QNLI, MRPC, and QQP) is shown below (the optimal set of parameters may vary across different tasks and data splits). Other hyperparameters are similar to values that are used in the DART model implementation.[5] Hyperparameters for SST-2, MR, CR, Subj, TREC, QNLI, MRPC, and QQP datsets:

- **Learning Rate**: We experimented with the following values: 1×10^{-5}, 5×10^{-5}, 1×10^{-4}, and 2×10^{-4}.
- **Weight Decay**: The weight decay values considered were 0.0, 0.01, 0.05, and 0.10.
- **Batch Size**: We evaluated batch sizes of 4, 8, 16, 24, and 32.

[5] https://cl.gy/fTiRK

https://doi.org/10.1017/nlp.2026.10013 Published online by Cambridge University Press

what is the sentiment of below 10 texts? (0: negative or 1: positive)

output format:

write only 0 or 1 for each text. write only all predictions for below texts in Python list. write only Python list.

texts

Figure 4. Evaluation prompt for SST2 dataset in zero-shot setting.

what is the sentiment of below 10 Customer review texts? (0: negative or 1: positive)

output format:

write only 0 or 1 for each Customer review text. write only all predictions for below texts in Python list. write only Python list.

texts

Figure 5. Evaluation prompt for CR dataset in zero-shot setting.

detect subjective and objective text in below 10 texts. (0: subjective or 1: objective)

output format:

write only 0 or 1 for each text. write only all predictions for below texts in Python list. write only Python list.

texts

Figure 6. Evaluation prompt for SUBJ dataset in zero-shot setting.

The hyperparameters utilized for the MNLI and SNLI datasets are specified as follows:

- **Learning Rate**: $\{1 \times 10^{-5}, 5 \times 10^{-5}, 1 \times 10^{-4}, 2 \times 10^{-4}\}$
- **Weight Decay**: {0.0, 0.01, 0.05, 0.10}
- **Batch Size**: {4, 8, 16}

Due to the lack of development data for ADE, TC, and Overruling datasets, we refrained from searching to find the best hyperparameters. For two tasks, we used "batch size" equal to 4, "learning rate" equal to 5e-5, and "weight decay equal" to 0.01.

5.3 LLMs' prompts in different settings

To evaluate the performance of LLMs in different settings for the examined datasets, we need different hard prompts that are shown in Figures 4, 5, and 6. As discussed in the introduction, finding the optimal version of these types of prompts requires searching. The used prompts were obtained after several trial and error steps through chat with these LLMs. In these prompts, we utilize batch inference by sending 10 texts as 'texts' and receiving LLMs answers. In few-shot settings, we include some examples at the beginning of these prompts. An illustrative example displaying "texts" and few-shot examples is depicted in Figure 7. For visualization purposes in this example, we use a 2-shot balanced setting and a batch inference size of 2 for the questionnaire texts.

https://doi.org/10.1017/nlp.2026.10013 Published online by Cambridge University Press

examples:

text: "most new movies have a bright sheen"
output: 1

text: "i do n ' t have an i am sam clue"
output: 0

what is the sentiment of below 10 texts? (0: negative or 1: positive)

output format:

write only 0 or 1 for each text. write only all predictions for below texts in Python list. write only Python list.

text 0:
"no movement , no yuks , not much of anything"

text 1:
"this is one of polanski ' s best films"

Figure 7. Complete evaluation prompt for SST2 dataset in the few-shot setting. For visualization purposes, we utilize a 2-shot balanced setting and a batch inference size of 2 for the questionnaire texts.

5.4 Computational cost and training time

Because BBSCL requires computing contrastive terms across balanced groups rather than a single unbalanced batch, it introduces additional compute compared to vanilla in-batch SCL. Formally, if a standard SCL update on a batch of size B requires $O(B^2)$ pairwise similarity computations (dominant cost is the $B * B$ similarity matrix in representation dimension d), then forming K balanced groups of size B_{grp} incurs roughly $K \cdot O(B_{grp}^2)$ similarity operations per update. In our implementation B_{grp} is selected so that the number of positive and negative examples per group is balanced (see Section 4.1), and for typical few-shot batches, the resulting values K are small (e.g. $K = 2$ in many of our imbalanced experiments), so the extra cost is modest in practice. Concretely, end-to-end training reported in this work was performed on NVIDIA P100 hardware; the full grid of experiments and seeds required approximately 110 GPU-hours total on a P100. A single training run for one specific hyperparameter configuration (one seed, one model, one data split) executes on the order of 4 minutes on that same hardware; therefore, the reported 110 GPU-hours reflect the aggregate cost across the grid and seeds.

6. Results

Statistical significance analysis of all tables is reported in Appendix A (Tables A1–A12).

6.1 Hard versus soft prompt learning

Table 1 compares methods based on hard prompt learning (such as PET and LMBFF) with those using soft prompt learning (such as PTuning, DART, PTuning + SCL, and DART + SCL). The results consistently show that soft prompt approaches outperform hard prompt methods across all datasets, both individually and on average. This performance gap highlights the limitations of hard prompts, which rely on manually designed or searched templates, and underscores the effectiveness of learnable, continuous prompts in few-shot scenarios. Notably, these comparisons are

https://doi.org/10.1017/nlp.2026.10013 Published online by Cambridge University Press

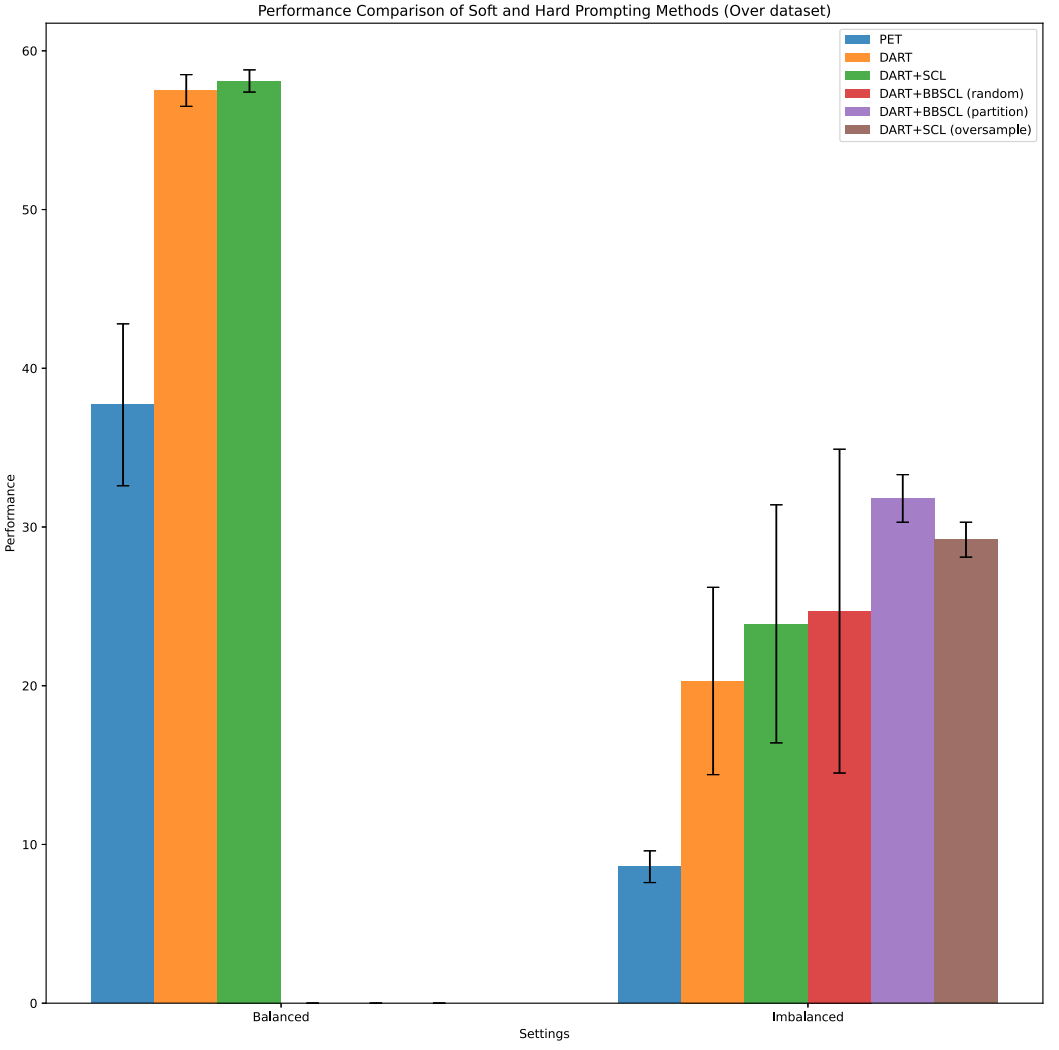

Figure 8. Comparison of hard and soft prompting methods in balanced and imbalanced settings for the Overruling dataset. For hard prompting, we consider the PET method. For soft prompting, we consider DART and our proposed extension in this paper when the PLM model is RoBERTa-large (the setting that shows the best performance in the previous). Results are reported based on different data folds. Mean $\pm$ as bar's height and standard deviation as error bar are reported. (ρ is an imbalance Ratio. For the imbalanced setting, we consider $\rho^{+} = 0.25$.

conducted under balanced data conditions, ensuring a fair evaluation of prompting types without the confounding effects of label imbalance. In Figure 8 for overruling dataset, we observe that under the balanced setting, the DART method outperforms PET; however, its performance is comparable to the our proposed method DART + SCL. In contrast, under the imbalanced setting, PET, which relies on hard prompting, yields the weakest performance. Meanwhile, the DART method, which employs soft prompting, achieves better results. Notably, all of our proposed methods consistently outperform hard prompting approaches in the imbalanced scenario. As shown in Figure 9 for the TC dataset, PET outperforms DART and DART + SCL in the balanced scenario. However, in the imbalanced scenario, all our proposed methods based on soft prompt learning (DART) surpass PET, which relies on hard prompt learning.

https://doi.org/10.1017/nlp.2026.10013 Published online by Cambridge University Press

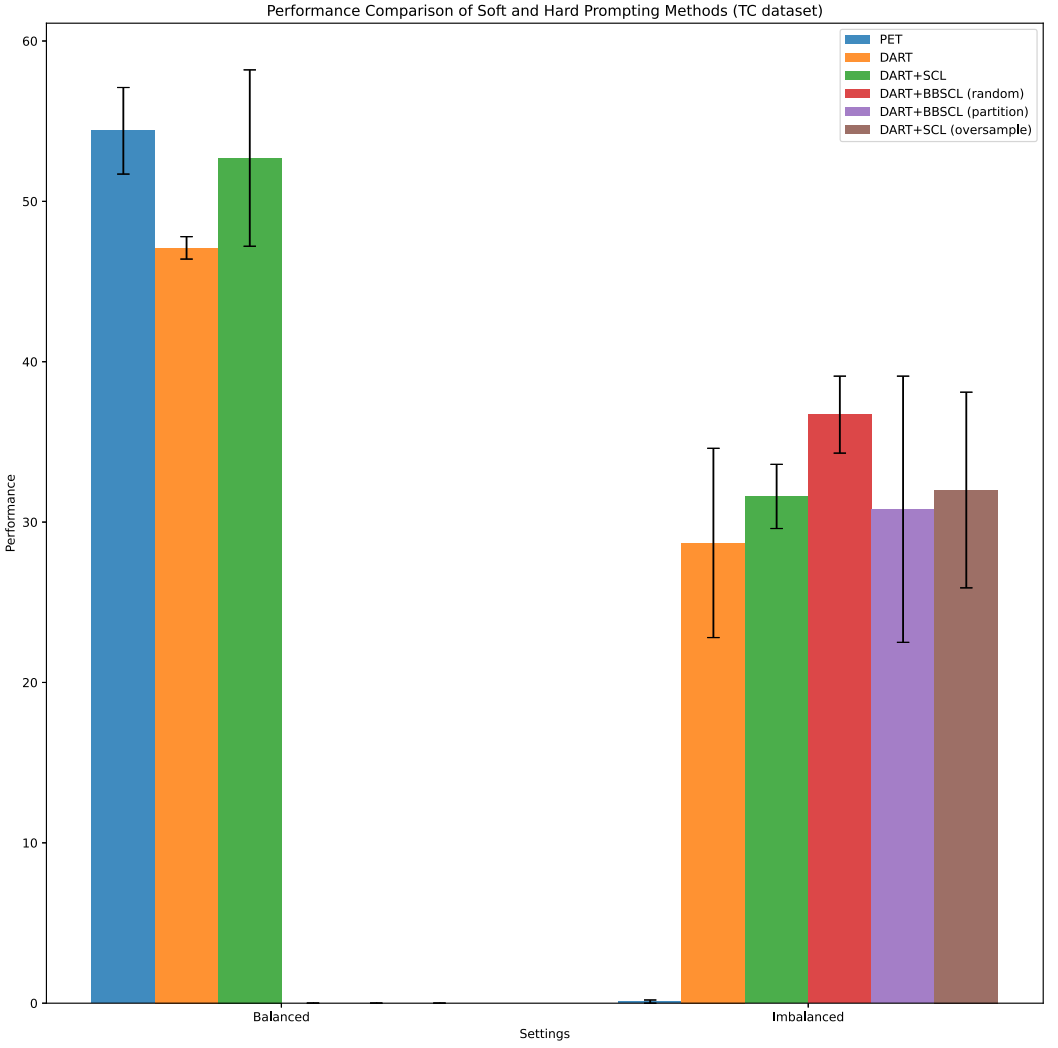

Figure 9. Comparison of hard and soft prompting methods in balanced and imbalanced settings for the TC dataset. For hard prompting, we consider the PET method. For soft prompting, we consider DART and our proposed extension in this paper when the PLM model is RoBERTa-large (the setting that shows the best performance in the previous). Results are reported based on different data folds. Mean $\pm$ as bar's height and standard deviation as error bar are reported. (ρ is an imbalance Ratio. For the imbalanced setting, we consider $\rho^{+} = 0.25$.

6.2 Adding SCL to DART and PTuning

Table 1 reveals the results of adding SCL to DART and PTuning models in 13 sentence classification tasks. DART + SCL and PTuning + SCL consistently outperform DART and PTuning, respectively, by small or large margins. This reveals that SCL is generally helpful in these few-shot scenarios.

To investigate the impact of PLM on SCL extension performance difference, we do these experiments for TC, Overruling, and SST-2 datasets with 2 different PLMs. the results of these experiments in balanced settings are reported in Tables 1 and 3. Results show that DART + SCL and PTuning + SCL consistently outperform DART and PTuning independent of PLM type in balanced settings.

https://doi.org/10.1017/nlp.2026.10013 Published online by Cambridge University Press

Table 2. Performance in few-shot balanced and imbalanced settings. Models use BERT-large-uncased. (ρ is a imbalance ratio; Avg: average performance; $\frac{pos}{all} = \rho^+$; $\frac{neg}{all} = \rho^-$

Model	SST-2 (balanced)	Overruling (balanced)	TC (balanced)	Avg
PTuning	$74.5_{\pm 7.2}$	$44.6_{\pm 0.1}$	$50.8_{\pm 7.3}$	56.6
PTuning + SCL	$\mathbf{82.0}_{\pm 1.8}$	$\mathbf{51.8}_{\pm 2.6}$	$\mathbf{54.7}_{\pm 4.6}$	**62.8**
DART	$81.4_{\pm 3.8}$	$47.8_{\pm 0.4}$	$41.4_{\pm 4.5}$	56.9
DART + SCL	$\mathbf{83.0}_{\pm 3.4}$	$\mathbf{52.3}_{\pm 5.4}$	$\mathbf{46.2}_{\pm 8.8}$	**60.5**
Model	SST-2	Overruling	TC	Avg
	($\rho^- = 0.25$)	($\rho^+ = 0.25$)	($\rho^+ = 0.25$)	
PTuning	$64.5_{\pm 5.2}$	$10.0_{\pm 0.5}$	$8.8_{\pm 1.8}$	27.8
PTuning + SCL	$64.8_{\pm 4.2}$	$10.8_{\pm 4.4}$	$6.6_{\pm 1.0}$	27.4
PTuning + BBSCL (random)	$65.9_{\pm 3.0}$	$11.7_{\pm 5.5}$	$11.6_{\pm 3.3}$	29.7
PTuning + BBSCL (partition)	$\mathbf{66.1}_{\pm 1.8}$	$17.6_{\pm 0.4}$	$9.0_{\pm 3.0}$	30.9
PTuning + SCL (oversample)	$65.0_{\pm 4.6}$	$\mathbf{20.4}_{\pm 4.8}$	$\mathbf{16.8}_{\pm 6.5}$	**34.1**
DART	$69.1_{\pm 2.8}$	$16.7_{\pm 4.1}$	$7.9_{\pm 0.6}$	31.2
DART + SCL	$68.4_{\pm 4.2}$	$22.1_{\pm 6.3}$	$11.9_{\pm 4.4}$	34.1
DART + BBSCL (random)	$\mathbf{71.1}_{\pm 5.2}$	$24.7_{\pm 7.9}$	$14.6_{\pm 3.9}$	36.8
DART + BBSCL (partition)	$69.3_{\pm 5.0}$	$\mathbf{25.5}_{\pm 4.3}$	$\mathbf{20.8}_{\pm 1.8}$	**38.5**
DART + SCL (oversample)	$68.4_{\pm 10.8}$	$23.9_{\pm 2.7}$	$14.4_{\pm 1.5}$	35.6

6.3 Robustness in imbalanced few-shot setting

In Tables 2 and 3, we analyze the impact of SCL and BBSCL extensions in imbalanced few-shot settings. To construct the training data, we sample examples from two classes based on the specified imbalance ratios (details in Sections 5.1.1 and 5.1.2). The data size remains the same across different settings of a task.

We observe that the performance is generally weakened when training on imbalanced data compared to the balanced setting. However, incorporating SCL improves the robustness of the models on average. there is an exception about the TC dataset when PLM is BERT-large-uncased and the soft prompt learning model is PTuning. In other cases, SCL improves the robustness of the models. Furthermore, using BBSCL (random) and BBSCL (partition) instead of SCL provides additional benefits in terms of performance on average. BBSCL(random) and BBSCL(partition) outperform SCL except for a few cases. In all datasets, PLMs, and soft prompt models, one of our BBSCL extensions outperforms SCL.

In the comparison of our BBSCL extensions and adding the oversample augmentation method to SCL (SCL (oversample)), we observe SCL (oversample) outperforms our BBSCL on average except in DART + BBSCL (partition) with BERT-large-uncased as PLM. With BERT-large-uncased PLM and in all datasets, one of our BBSCL extensions outperforms SCL (oversample). With RoBERTa-large as PLM and in all datasets except the SST-2 dataset, one of our BBSCL extensions outperforms SCL (oversample).

To examine the impact of train size in imbalanced settings, we do Tables 2 and 3 experiments for Overruling and TC datasets for larger train size far from few-shot settings. Table 6 shows these

https://doi.org/10.1017/nlp.2026.10013 Published online by Cambridge University Press

Table 3. Performance in few-shot imbalanced settings. Models use RoBERTa-large. (ρ is a imbalance ratio; $\frac{pos}{all} = \rho^+$; $\frac{neg}{all} = \rho^-$

Model	SST-2 ($\rho^- = 0.25$)	Overruling ($\rho^+ = 0.25$)	TC ($\rho^+ = 0.25$)	Avg
PTuning	$78.3_{\pm 8.3}$	$17.8_{\pm 4.1}$	$30.8_{\pm 7.5}$	42.3
PTuning + SCL	$83.5_{\pm 3.8}$	$25.0_{\pm 6.0}$	$35.0_{\pm 3.1}$	47.8
PTuning + BBSCL (random)	$85.6_{\pm 4.1}$	$25.6_{\pm 10.6}$	$35.6_{\pm 2.6}$	48.9
PTuning + BBSCL (partition)	$\mathbf{86.0}_{\pm 2.5}$	$29.8_{\pm 4.0}$	$28.3_{\pm 2.1}$	48.0
PTuning + SCL (oversample)	$85.9_{\pm 2.3}$	$\mathbf{33.4}_{\pm 3.3}$	$\mathbf{38.6}_{\pm 7.6}$	**52.6**
DART	$82.8_{\pm 5.5}$	$20.3_{\pm 5.9}$	$28.7_{\pm 5.9}$	43.9
DART + SCL	$83.3_{\pm 8.4}$	$23.9_{\pm 7.5}$	$31.6_{\pm 2.0}$	46.3
DART + BBSCL (random)	$84.5_{\pm 9.9}$	$24.7_{\pm 10.2}$	$\mathbf{36.7}_{\pm 2.4}$	48.6
DART + BBSCL (partition)	$85.0_{\pm 10.3}$	$\mathbf{31.8}_{\pm 1.5}$	$30.8_{\pm 8.3}$	49.2
DART + SCL (oversample)	$\mathbf{87.7}_{\pm 4.1}$	$29.2_{\pm 1.1}$	$32.0_{\pm 6.1}$	**49.6**

experiments results. Only in the PTuning model with RoBERTa-large as PLM, incorporating SCL improves the robustness of the models on average. It shows that high-frequency class domination of SCL loss is more visible in larger train data. one of our BBSCL extensions outperforms SCL on average except in the PTuning model with BERT-large-uncased as PLM. It shows our BBSCL approach can decrease the SCL problem in imbalanced settings. In the comparison of our BBSCL extensions and SCL (oversample), we observe that SCL (oversample) outperforms our BBSCL on average except in the PTuning model with RoBERTa-large as PLM. To investigate the effect of imbalance ratio value on performance improvement w.r.t adding SCL and BBSCL (random), we compare DART, DART + SCL, and DART + BBSCL for different imbalance ratios for a dataset.

In Figure 10 and Table 7, the comparison of DART, DART + SCL, and DART + BBSCL (random) for different imbalance ratios (ρ^+) for the Overruling dataset is shown. We see that adding SCL is beneficial in imbalanced settings for different imbalance ratio values, especially for $\rho^+ = 0.25$. The use of BBSCL (random) instead of SCL has resulted in robustness improvement. It shows that BBSCL (random) has successfully addressed the problem of SCL which focuses more on the majority class.

6.4 Impact of proposed approaches on base transformer without learnable prompt

We expressed that our extensions can be applied to models with learnable prompt tokens and they only need input representation from the main task. The input representation is not specific to prompt learning and can be achieved from other tuning models like standard fine-tuning frozen PLMs. In this part, we investigate the impact of our BBSCL extensions on the standard fine-tuning of PLMs. Table 4 shows the impact of adding SCL and our BBSCls to CE loss of fine-tuning in balanced and imbalanced few-shot settings. Results show that CE + SCL consistently outperforms CE independent of PLM type in balanced and imbalanced settings.

CE + BBSCL (random) and CE + BBSCL (partition) outperform CE + SCL in imbalanced settings, on average, and across all datasets and PLMs, with only a few exceptions. This demonstrates that our BBSCL approaches effectively address the SCL problem. However, in imbalanced settings, CE + SCL (oversample) outperforms CE + BBSCL (random) and CE + BBSCL (partition)

https://doi.org/10.1017/nlp.2026.10013 Published online by Cambridge University Press

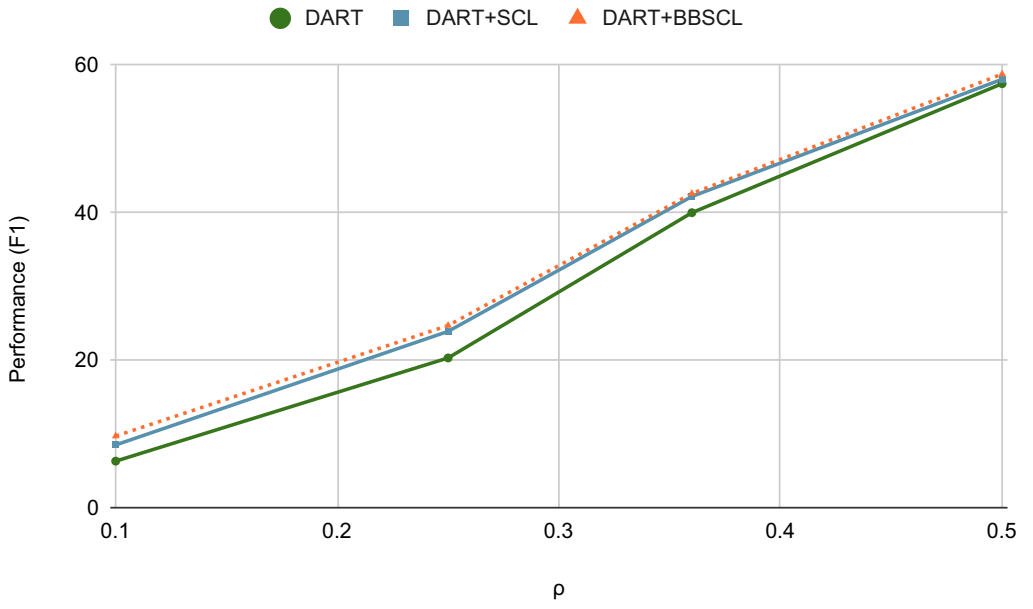

Figure 10. F1 performance of DART, DART + SCL, and DART + BBSCL (random) in Overruling dataset wrt four different imbalance ratio ($\rho^+ = \{0.1, 0.25, 0.36, 0.5\}$). $\rho^+ = 0.5$ is the balanced setting and lower ρ^+ means higher label-imbalance.

on average and across all datasets and PLMs, except in the case of CE + BBSCL (random) with all PLMs when the dataset is CR.

To examine the impact of train size in imbalanced settings, we do Table 4 experiments for a larger train size far from few-shot settings. Table 5 shows these experiments results. One of the CE + BBSCL models on average and across all datasets and PLMs outperforms CE + SCL. CE + SCL (oversample) on average outperforms our CE + BBSCL models except in some cases.

6.5 Analysis of data distribution for label complexity

We focus on the real-world datasets (ADE, TC, Overruling) presented in Table 1. The difference in performance between the DART and DART + SCL models in these datasets can be attributed to the unique characteristics of the data in each set. To better understand this, we need to analyze the data distribution in each set, which is displayed in Figures 11, 12, and 13. TC and ADE datasets' label space is more complex than Overruling. In Overruling, classes are more separable. The SCL improves performance more when there is more overlap between the concentrations of the two classes because it can create representations that better separate the two classes.

The vector representation of text generated by the language model has high dimensions that are not easily understandable by humans. To better comprehend the distribution of texts, we display this distribution in two-dimensional space. To achieve this, we utilize the UMAP (McInnes *et al.* 2018) dimension reduction method, which is specifically designed for this purpose. By using this method, we obtain a two-dimensional representation for each text, and the data distribution is displayed in this two-dimensional space.

The data distribution of the three datasets reveals that the data in TC and ADE are concentrated in many areas where two classes of data are present side by side and on top of each other, resulting in zero separation between them. The ADE dataset is entirely filled with overlapped concentrations of the two classes, while in the TC dataset, there are areas with less concentrated data, and the separation of the two classes is visible in those areas. In some areas of the data distribution, the

https://doi.org/10.1017/nlp.2026.10013 Published online by Cambridge University Press

Table 4. Impact of SCL and BBSCL on DART and PTuning models. Performances are in imbalanced settings (100 examples for positive and 300 examples for negative class) far from few-shot settings. Results are reported based on different data folds. Mean ± std performances are reported. (ρ is an imbalance Ratio; $\frac{pos}{all} = \rho^+$; $\frac{neg}{all} = \rho^-$; Avg: average performance)

Model	Overruling ($\rho^+ = 0.25$)	TC ($\rho^+ = 0.25$)	Avg
RoBERTa-large			
PTuning	$27.9_{\pm 1.0}$	$23.8_{\pm 0.9}$	25.9
PTuning + SCL	$26.0_{\pm 0.2}$	$35.8_{\pm 7.9}$	30.9
PTuning + BBSCL (random)	$\mathbf{28.7}_{\pm 0.5}$	$\mathbf{44.2}_{\pm 22.3}$	**36.5**
PTuning+BBSCL (partition)	$26.3_{\pm 1.0}$	$28.5_{\pm 2.8}$	27.4
PTuning + SCL (oversample)	$28.0_{\pm 10.8}$	$38.2_{\pm 1.0}$	33.1
DART	$28.7_{\pm 1.0}$	$28.2_{\pm 6.8}$	28.5
DART + SCL	$33.2_{\pm 1.5}$	$20.9_{\pm 5.8}$	27.1
DART + BBSCL (random)	$24.8_{\pm 2.3}$	$36.1_{\pm 6.0}$	30.5
DART + BBSCL (partition)	$32.1_{\pm 1.8}$	$25.0_{\pm 6.5}$	28.6
DART + SCL (oversample)	$\mathbf{37.6}_{\pm 4.0}$	$\mathbf{36.4}_{\pm 2.0}$	**37.0**
BERT-large-uncased			
PTuning	$21.2_{\pm 1.4}$	$23.3_{\pm 0.1}$	22.3
PTuning + SCL	$23.0_{\pm 1.9}$	$16.5_{\pm 2.2}$	19.8
PTuning + BBSCL (random)	$25.2_{\pm 2.2}$	$17.6_{\pm 1.1}$	21.4
PTuning + BBSCL (partition)	$19.0_{\pm 1.5}$	$21.5_{\pm 0.0}$	20.3
PTuning + SCL (oversample)	$\mathbf{31.1}_{\pm 1.0}$	$\mathbf{25.8}_{\pm 1.0}$	**28.5**
DART	$27.1_{\pm 1.5}$	$23.7_{\pm 9.5}$	25.4
DART + SCL	$22.2_{\pm 2.6}$	$11.2_{\pm 4.3}$	16.7
DART + BBSCL (random)	$28.3_{\pm 3.4}$	$15.6_{\pm 1.5}$	22.0
DART + BBSCL (partition)	$\mathbf{30.5}_{\pm 1.6}$	$20.6_{\pm 1.0}$	25.6
DART + SCL (oversample)	$29.1_{\pm 0.3}$	$\mathbf{31.0}_{\pm 6.5}$	**30.1**

Table 5. F1 of DART, DART + SCL, and DART + BBSCL (random) in overrulling dataset w.r.t four different imbalance ratio ($\rho^+ = \{0.1, 0.25, 0.36, 0.5\}$). $\rho^+ = 0.5$ is the balanced setting and lower ρ^+ means higher label-imbalance

	Imbalance ratio			
Model	0.1	0.25	0.36	0.5
DART	$6.3_{\pm 0.04}$	$20.3_{\pm 5.9}$	$40_{\pm 1.8}$	$57.5_{\pm 1.0}$
DART + SCL	$8.5_{\pm 1.5}$	$23.9_{\pm 7.5}$	$42.2_{\pm 3.5}$	$58.1_{\pm 0.7}$
DART + BBSCL	$\mathbf{9.7}_{\pm 1.5}$	$\mathbf{24.7}_{\pm 10.2}$	$\mathbf{42.6}_{\pm 0.2}$	$\mathbf{58.8}_{\pm 0.3}$

https://doi.org/10.1017/nlp.2026.10013 Published online by Cambridge University Press

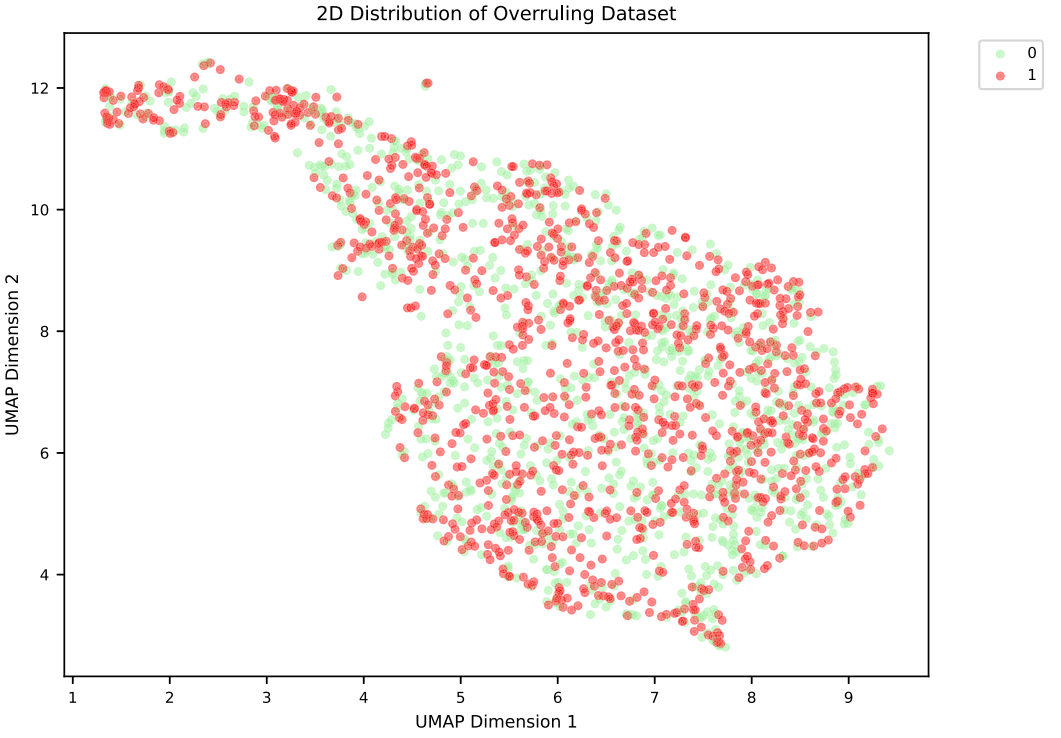

Figure 11. Test data distribution of Overruling dataset in 2D space. Test data are a sample of the entire data space. The vector representation of texts is prepared with the RoBERTa-large language model. We used the UMAP (McInnes, Healy, and Saul 2018) dimension reduction method to present this distribution in two-dimensional space. Classes are well-separated (low $R_D = 0.44$), explaining smaller SCL gains here.

data of the two classes are not close to each other, and there is a visible gap between them. In the Overruling data distribution, overlapping in classes' concentrations is less common, and in most areas, the concentrations of the data of these classes are separated from each other. In most areas of the data, there is a visible separation between the two classes. These distributions correspond to the difference in performance of the two models.

6.6 When does SCL cause more performance gain?

We analyze how SCL's contribution is related to the label complexity (analyze of data distribution for label complexity is explained in Section 6.5). To assess the separability of classes in the datasets, we utilize the R_D metric, adopted from DART (Zhang *et al.* 2022). This metric represents the ratio of the average intraclass distance to the average interclass distance. Each data point is represented by the vector of the mask token in the "Class Discrimination Objective". A lower (higher) value of R_D indicates higher (lower) separability between classes, implying lower (higher) label complexity in the dataset.

For the Overruling, ADE, and TC datasets, the R_D values are 0.44, 0.85, and 0.84, respectively. The lowest R_D value (0.44) is observed in the Overruling dataset, where the performance difference between the DART and DART + SCL models is the smallest (0.6). For the other two datasets, the R_D values are higher and so are the the performance differences. Based on these results, it is evident that the SCL objective provides greater benefit in datasets with higher R_D values, indicating greater label complexity.

https://doi.org/10.1017/nlp.2026.10013 Published online by Cambridge University Press

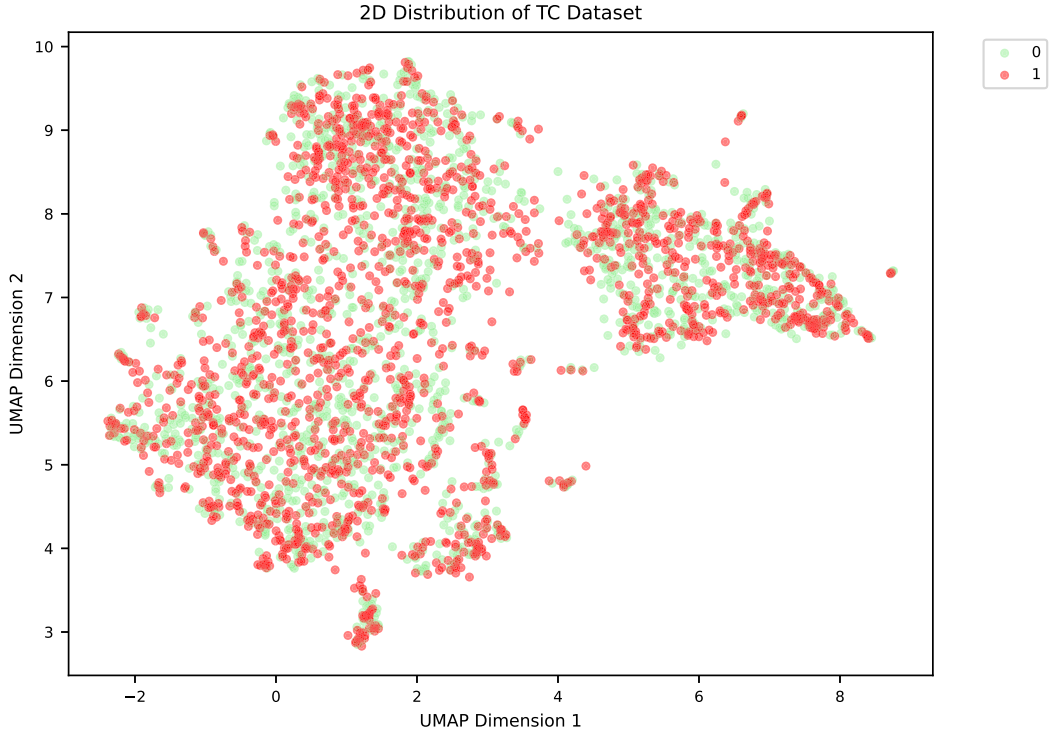

Figure 12. Test data distribution of TC dataset in 2D space. Test data are a sample of the entire data space. The vector representation of texts is prepared with the RoBERTa-large language model. We used the UMAP (McInnes, Healy, and Saul 2018) dimension reduction method to present this distribution in two-dimensional space. TC dataset shows moderate overlap ($R_D = 0.84$), where SCL improves separability.

6.7 LLMs vs soft prompt learning models in our settings

In this section, we will evaluate the performance of LLMs, focusing on ChatGPT (OpenAI 2023). We will compare models based on soft prompt learning discussed in this paper with GPT-3.5-turbo (GPT-3.5) (OpenAI 2023) as a closed-source LLM and OpenChat (Wang *et al*. 2024) as an open-source LLM. Comparison results are reported in Table 8.

In the table results, it is evident that in balanced settings, GPT-3.5 performs better in few-shot scenarios compared to zero-shot. However, in the case of OpenChat, an LLM with fewer parameters, incorporating few-shot examples from the training data does not enhance performance. This indicates that increasing the size of the input prompt poses a challenge for LLMs with fewer parameters.

In the imbalance settings, it is evident that OpenChat faces significant challenges, leading to a considerable reduction in its performance compared to the balanced few-shot examples. Conversely, GPT-3.5 shows an average performance decrease of 1.1. Analyzing OpenChat's performance, we observe that due to the imbalance of examples, the model's performance has declined compared to scenarios where no examples are provided.

The noteworthy observation is that in the SUBJ dataset, LLMs achieve an accuracy of nearly 50 when no examples are provided, and upon adding examples, the GPT-3.5 model attains an accuracy of 71.5 in a balanced setting. This observation is made in the context of our study about soft prompt learning models, which have demonstrated notably higher performance with significantly smaller PLMs. This case demonstrates that LLMs are not proficient in all tasks, and small language models can outperform these large models with high inference and API costs in specific

https://doi.org/10.1017/nlp.2026.10013 Published online by Cambridge University Press

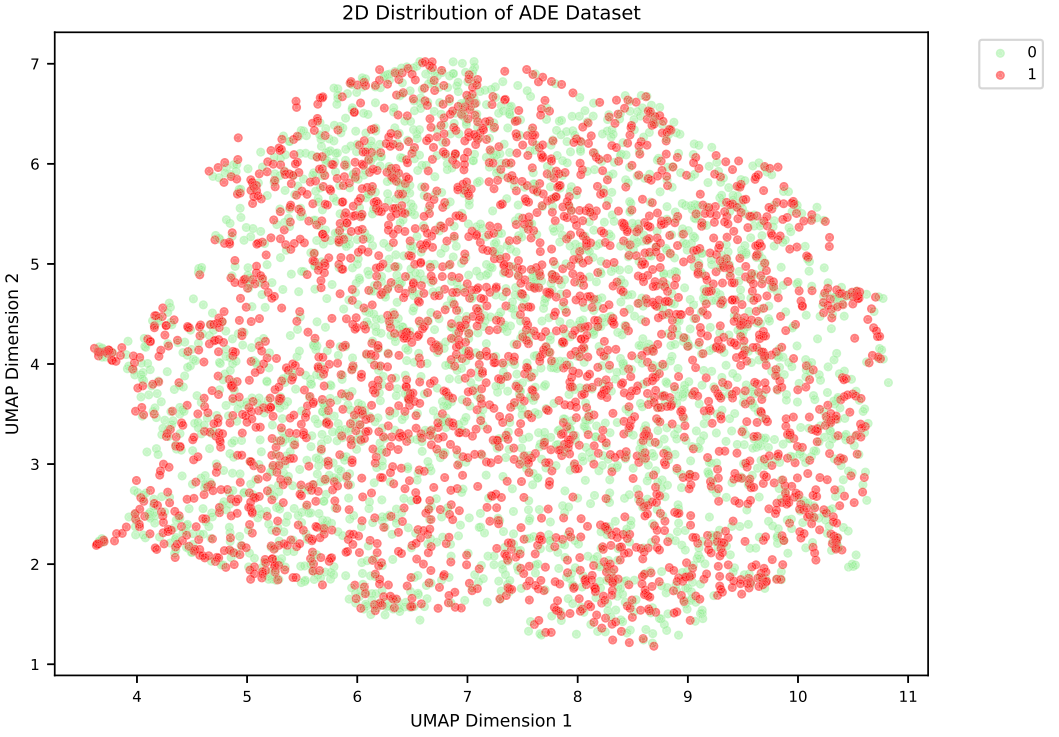

Figure 13. Test data distribution of ADE dataset in 2D space. Test dataare a sample of the entire data space. The vector representation of texts is prepared with the RoBERTa-large language model. We used the UMAP (McInnes, Healy, and Saul 2018) dimension reduction method to present this distribution in two-dimensional space. ADE exhibits high class overlap ($R_D = 0.85$), leading to significant SCL benefits.

tasks. Additionally, our review indicates that, on average, the soft prompt learning models with SCL and BBSCL extensions perform better than these models in both balanced and imbalanced data conditions.

6.8 Impact of batch size on BBSCL method

To quantify how batch size influences performance under severe label imbalance ($\rho^+ = 0.25$) on the TC dataset, we trained RoBERTa-large with five configurations – DART, DART + SCL, DART + SCL (oversample), DART + BBSCL (random), and DART + BBSCL (partition) – using batch sizes of 4, 8, and 16. The averaged F_1 scores (with one-sigma error bars) are plotted in Figure 14.

With a batch size of 4, vanilla DART achieves $28.7_{\pm 1.0}$, while adding the standard supervised contrastive loss (DART + SCL) raises performance to $33.2_{\pm 1.5}$. Oversampling the minority class before SCL yields an even higher F_1 of $37.6_{\pm 4.0}$, indicating that simple duplication can help when batches are very small. Both BBSCL variants also improve over DART: the random-balanced batch strategy reaches $24.8_{\pm 2.3}$ and the partitioned strategy $32.1_{\pm 1.8}$, demonstrating that explicit balancing pays off even at tiny batch sizes.

At batch size 8, DART's performance drops to $19.8_{\pm 1.9}$, and DART + SCL to $24.1_{\pm 0.9}$. Here, BBSCL(random) peaks at $32.0_{\pm 2.4}$ and BBSCL(partition) at $31.7_{\pm 3.1}$ – both significantly outpacing

https://doi.org/10.1017/nlp.2026.10013 Published online by Cambridge University Press

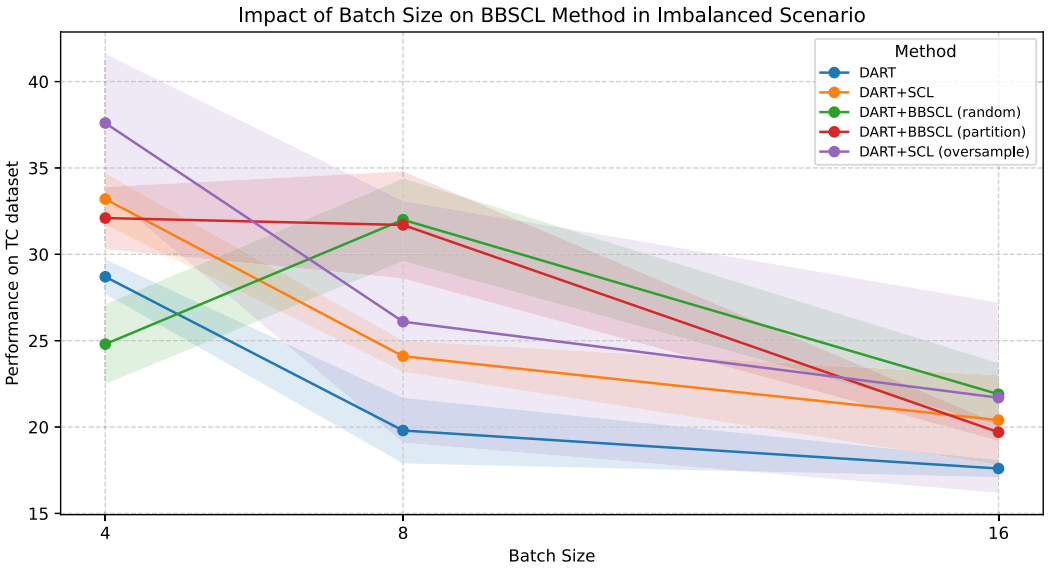

Figure 14. Impact of batch size on BBSCL method in the imbalanced setting.

oversampling ($26.1_{\pm 7.0}$) and vanilla SCL. This confirms that our batch balancing is most effective in moderate-sized batches, where the natural class distribution would otherwise dominate.

When the batch size grows to 16, all methods converge toward lower F_1 scores: DART at $17.6_{\pm 0.5}$, DART + SCL at $20.4_{\pm 2.6}$, oversampling at $21.7_{\pm 5.5}$, BBSCL(random) at $21.9_{\pm 1.8}$, and BBSCL(partition) at $19.7_{\pm 0.5}$. Although the absolute gains shrink under large batches, BBSCL(random) still slightly outperforms standard SCL, suggesting that balanced contrastive sampling continues to mitigate imbalance bias even when many examples are seen per update.

Small batches (size 4) benefit most from oversampling, but at batch size 8 our BBSCL methods deliver the largest and most consistent improvements. Even at size 16, BBSCL(random) retains a modest edge over both standard SCL and oversampling, confirming that balancing each mini-batch is a robust strategy across a wide range of batch-size settings (see Figure 14).

7. Discussion

7.1 When to use SCL and BBSCL

The integration of SCL into soft prompt tuning frameworks like DART and PTuning offers significant advantages in specific scenarios. Our experiments demonstrate that SCL is particularly effective in low-data regimes, where labeled examples are scarce. Using limited labels to create robust representations, SCL enhances the discriminative power of the model, leading to consistent performance improvements in 13 tasks of a few shots. For instance, in the SST-2 dataset, DART + SCL achieved a 2.1% improvement over the base DART model, highlighting the value of SCL in few-shot settings.

In imbalanced datasets, our proposedBBSCL method addresses the challenge of class imbalance by ensuring that each mini-batch contains a proportional representation of minority and majority classes. This approach stabilizes the contrastive loss calculations, preventing the majority class from dominating the learning process. For example, in the Overruling dataset with a 25% positive class ratio, BBSCL improved F1 scores by 4.2% compared to standard SCL. This makes BBSCL particularly suitable for real-world applications, such as medical text classification, where label imbalances are common.

https://doi.org/10.1017/nlp.2026.10013 Published online by Cambridge University Press

Table 6. Impact of SCL and BBSCL on text classification with fine-tuning frozen PLMs. Performances are in few-shot balanced (16 examples per class) and imbalanced settings (8 examples for positive and 24 examples for negative class). For PLMs, we use RoBERTa-large and BERT-large-uncased. Results are reported based on 5 run with different random seeds. Mean ± std performances are reported. (ρ is an imbalance Ratio; $\frac{pos}{all} = \rho^+$; $\frac{neg}{all} = \rho^-$; CE is cross entropy; Avg: average performance; in CE + SCL (oversample), we use oversample to balance batches in training; in CE + BBCL (random), we sample negative nodes from majority class with replacement to construct label-balanced groups and it is possible to have shared negative nodes between groups; In CE + BBCL (partition), we sample negative nodes from majority class without replacement to construct label-balanced groups and there are no shared nodes between groups.)

Loss	SST-2 (balanced)	SUBJ (balanced)	CR (balanced)	Avg
RoBERTa-large				
CE	$77.5_{\pm 4.3}$	$92.1_{\pm 0.5}$	$75.7_{\pm 2.9}$	81.8
CE + SCL	$\mathbf{80.7}_{\pm 5.7}$	$\mathbf{92.4}_{\pm 0.7}$	$\mathbf{76.9}_{\pm 6.1}$	**83.3**
BERT-large-uncased				
CE	$80.0_{\pm 5.9}$	$91.3_{\pm 0.6}$	$68.7_{\pm 3.4}$	80.0
CE + SCL	$\mathbf{81.5}_{\pm 1.5}$	$\mathbf{92.3}_{\pm 0.2}$	$\mathbf{70.6}_{\pm 3.6}$	**81.5**
Loss	SST-2	SUBJ	CR	Avg
	($\rho^+ = 0.25$)	($\rho^+ = 0.25$)	($\rho^+ = 0.25$)	
RoBERTa-large				
CE	$67.5_{\pm 5.8}$	$83.6_{\pm 9.7}$	$66.4_{\pm 7.4}$	72.5
CE + SCL	$72.5_{\pm 5.3}$	$88.4_{\pm 4.5}$	$72.8_{\pm 4.6}$	77.9
CE + BBSCL (random)	$74.1_{\pm 6.3}$	$91.1_{\pm 1.1}$	$\mathbf{77.2}_{\pm 3.5}$	80.8
CE + BBSCL (partition)	$72.1_{\pm 2.4}$	$91.6_{\pm 0.5}$	$66.6_{\pm 3.4}$	76.8
CE + SCL (oversample)	$\mathbf{78.2}_{\pm 4.7}$	$\mathbf{91.7}_{\pm 1.0}$	$73.0_{\pm 7.3}$	**81.0**
BERT-large-uncased				
CE	$64.3_{\pm 6.0}$	$86.5_{\pm 2.2}$	$50.2_{\pm 3.0}$	67.0
CE + SCL	$65.0_{\pm 3.2}$	$87.9_{\pm 1.8}$	$56.4_{\pm 3.8}$	69.8
CE + BBSCL (random)	$75.4_{\pm 5.8}$	$89.7_{\pm 0.7}$	$\mathbf{62.1}_{\pm 8.8}$	75.7
CE + BBSCL (partition)	$73.0_{\pm 2.6}$	$91.0_{\pm 0.7}$	$59.5_{\pm 5.4}$	74.5
CE + SCL (oversample)	$\mathbf{80.1}_{\pm 4.0}$	$\mathbf{91.4}_{\pm 1.0}$	$61.2_{\pm 7.5}$	**77.6**

Finally, SCL excels in complex label spaces, where classes are less separable. Our analysis using the R_D metric (ratio of intraclass to interclass distances) revealed that SCL provides greater benefits in datasets with higher R_D values, such as ADE ($R_D = 0.85$) and TC ($R_D = 0.84$). In these cases, SCL helps create more distinct representations for overlapping classes, improving overall model performance. For example, in the ADE dataset, DART + SCL achieved a 6.3% improvement over the base model, demonstrating the effectiveness of SCL in challenging classification tasks.

7.2 Comparison with large language models (LLMs)

Our comparison with large language models (LLMs) like GPT-3.5 and OpenChat reveals important insights into the trade-offs between fine-tuning small PLMs and using large, pretrained LLMs.

https://doi.org/10.1017/nlp.2026.10013 Published online by Cambridge University Press

Table 7. Impact of SCL and BBSCL on text classification with fine-tuning frozen PLMs. Performances are in imbalanced settings (64 examples for positive and 192 examples for negative class) far from few-shot settings. Results are reported based on 5 run with different random seeds. Mean ± std performances are reported. (ρ is an imbalance Ratio; $\frac{pos}{all} = \rho^+$; $\frac{neg}{all} = \rho^-$; CE is cross entropy; Avg: average performance)

Loss	SST-2 ($\rho^+ = 0.25$)	SUBJ ($\rho^+ = 0.25$)	CR ($\rho^+ = 0.25$)	Avg
RoBERTa-large				
CE	$90.2_{\pm 1.0}$	$93.3_{\pm 1.0}$	$89.5_{\pm 0.5}$	91.0
CE + SCL	$90.0_{\pm 2.0}$	$93.7_{\pm 0.4}$	$89.7_{\pm 1.7}$	91.1
CE + BBSCL (random)	$\mathbf{91.3}_{\pm 1.2}$	$93.7_{\pm 0.6}$	$89.6_{\pm 0.4}$	91.5
CE + BBSCL (partition)	$90.5_{\pm 1.3}$	$94.1_{\pm 0.6}$	$\mathbf{89.8}_{\pm 0.3}$	91.5
CE + SCL (oversample)	$90.8_{\pm 1.3}$	$\mathbf{94.2}_{\pm 0.3}$	$89.7_{\pm 0.5}$	**91.6**
BERT-large-uncased				
CE	$91.0_{\pm 0.5}$	$92.8_{\pm 0.5}$	$87.3_{\pm 1.4}$	90.4
CE + SCL	$91.0_{\pm 0.5}$	$94.2_{\pm 0.8}$	$88.5_{\pm 1.4}$	91.2
CE + BBSCL (random)	$\mathbf{91.1}_{\pm 0.2}$	$94.4_{\pm 0.2}$	$88.6_{\pm 1.4}$	91.4
CE + BBSCL (partition)	$90.0_{\pm 0.6}$	$93.8_{\pm 0.3}$	$87.2_{\pm 2.2}$	90.3
CE + SCL (oversample)	$90.5_{\pm 0.3}$	$\mathbf{94.6}_{\pm 0.4}$	$\mathbf{89.5}_{\pm 0.8}$	**91.5**

While GPT-3.5 performs well in balanced few-shot settings (e.g., 98.5% accuracy on SST-2), its performance degrades in imbalanced scenarios, and it incurs high computational and API costs. In contrast, our enhanced soft prompt learning models with SCL and BBSCL consistently outperform GPT-3.5 in both balanced and imbalanced settings, while using significantly smaller PLMs like RoBERTa-large. For instance, DART + SCL achieved 93.7% accuracy on SST-2, compared to GPT-3.5's 98.0%, but with much lower resource requirements.

This suggests that small, fine-tuned models are often more practical for domain-specific tasks, especially when computational resources are limited. However, LLMs remain valuable for zero-shot scenarios or tasks requiring broad generalization. Our findings highlight the importance of choosing the right approach based on the task requirements: fine-tuning with SCL/BBSCL for few-shot, imbalanced, or domain-specific tasks, and leveraging LLMs for zero-shot or general-purpose applications.

7.3 Bias mitigation and fairness

The use of SCL and BBSCL also has implications for mitigating biases in language models. By ensuring a balanced representation of classes during training, BBSCL reduces the risk of majority-class bias, which is particularly important in sensitive applications like medical diagnosis or legal text analysis. For example, in the Overruling dataset, BBSCL improved the F1 score for the minority class by 9.7%, compared to standard SCL, demonstrating its ability to handle imbalanced data more fairly.

However, it is important to note that while BBSCL addresses class imbalance, it does not eliminate other forms of bias, such as those stemming from the underlying pre-trained language model. Future work could explore combining BBSCL with debiasing techniques, such as adversarial training or fairness constraints, to enhance model fairness further.

https://doi.org/10.1017/nlp.2026.10013 Published online by Cambridge University Press

Table 8. Comparison of LLMs and soft prompt learning models in balanced and imbalanced settings. Based on the results in Tables 1 to 6, and considering the superior performance of models utilizing RoBERTa-large as the pretrained language model (PLM) and the DART as soft prompt model in most times, we have opted to exclusively employ the RoBERTa-large model in our experiments for the DART model. Results are reported based on 3 different data folds in the few-shot and 3 different run seeds in the zero-shot. We sample 200 data points from the test dataset for evaluation to manage the API costs of GPT-3.5. Mean ± std performances are reported. GPT-3.5 and OpenChat results are based on their APIs on 8 February. (ρ is an imbalance ratio; $\frac{pos}{all} = \rho^+$; $\frac{neg}{all} = \rho^-$; Avg: average performance)

Model	SST-2 (balanced)	CR (balanced)	SUBJ (balanced)	Avg
LLMs				
gpt-3.5-turbo (OpenAI 2023) (zero-shot)	**98.5**$_{\pm 0.0}$	96.0$_{\pm 0.0}$	52.5$_{\pm 0.4}$	82.3
gpt-3.5-turbo (OpenAI 2023)	98.0$_{\pm 0.0}$	**97.0**$_{\pm 0.4}$	**71.5**$_{\pm 2.5}$	**88.8**
OpenChat (Wang *et al.* 2024)(zero-shot)	85.0$_{\pm 0.7}$	94.6$_{\pm 0.5}$	54.7$_{\pm 3.9}$	78.1
OpenChat (Wang *et al.* 2024)	82.7$_{\pm 6.5}$	93.8$_{\pm 1.5}$	53.2$_{\pm 2.7}$	76.6
RoBERTa-large				
DART	89.7$_{\pm 1.4}$	90.5$_{\pm 1.8}$	90.5$_{\pm 1.9}$	90.2
DART + SCL	**91.5**$_{\pm 1.1}$	**91.8**$_{\pm 0.6}$	**92.7**$_{\pm 1.2}$	**92.0**
Model	SST-2	CR	SUBJ	Avg
	($\rho^+ = 0.25$)	($\rho^+ = 0.25$)	($\rho^+ = 0.25$)	
LLMs				
gpt-3.5-turbo (OpenAI 2023)	**97.8**$_{\pm 0.2}$	**97.0**$_{\pm 0.8}$	**68.3**$_{\pm 3.8}$	**87.7**
OpenChat (Wang *et al.* 2024)	50.6$_{\pm 2.0}$	92.3$_{\pm 1.3}$	51.2$_{\pm 3.6}$	64.7
RoBERTa-large				
DART	87.8$_{\pm 3.0}$	82.8$_{\pm 5.2}$	84.5$_{\pm 5.0}$	85.0
DART + SCL	88.0$_{\pm 3.9}$	86.2$_{\pm 2.5}$	90.2$_{\pm 1.5}$	88.1
DART + BBSCL (random)	88.2$_{\pm 3.0}$	**87.8**$_{\pm 4.1}$	91.0$_{\pm 0.7}$	89.0
DART + BBSCL (partition)	89.2$_{\pm 2.4}$	85.7$_{\pm 3.1}$	**91.3**$_{\pm 1.6}$	88.7
DART + SCL (oversample)	**89.3**$_{\pm 1.0}$	87.2$_{\pm 0.5}$	91.0$_{\pm 1.5}$	**89.2**

7.4 Generalizing BBSCL to multiclass and multilabel settings

While the main paper focused on binary imbalanced scenarios (minority vs. majority), balanced-batch supervised contrastive learning (BBSCL) generalizes naturally to both multiclass and multilabel problems. In this subsection, we present the principal extensions, give a compact algorithmic sketch for the multiclass (random sampling) variant, and note practical complexity trade-offs.

Notation and objectives. Let C denote the number of classes and let n_c be the number of available examples for class $c \in \{1, \ldots, C\}$. For multilabel data, each example i has a label set $L_i \subseteq \{1, \ldots, C\}$. We denote the encoder (including optional projection head) by $\Phi(\cdot)$ and use $\mathrm{sim}(\cdot, \cdot)$ for a similarity function (e.g., cosine). The supervised contrastive loss for an example i in a batch with

https://doi.org/10.1017/nlp.2026.10013 Published online by Cambridge University Press

positive index set $P(i)$ is the usual form

$$\mathscr{L}_i = -\frac{1}{|P(i)|} \sum_{j \in P(i)} \log \frac{\exp\left(\mathrm{sim}(\Phi_i, \Phi_j)/\tau\right)}{\sum_{k \neq i} \exp\left(\mathrm{sim}(\Phi_i, \Phi_k)/\tau\right)}, \tag{12}$$

where τ is the temperature parameter. For multilabel settings, one can weight each positive by label-overlap (e.g., Jaccard):

$$w_{ij} = \frac{|L_i \cap L_j|}{|L_i \cup L_j|}, \qquad \mathscr{L}_i = -\frac{1}{\sum_{j \in P(i)} w_{ij}} \sum_{j \in P(i)} w_{ij} \log \frac{\exp\left(\mathrm{sim}(\Phi_i, \Phi_j)/\tau\right)}{\sum_{k \neq i} \exp\left(\mathrm{sim}(\Phi_i, \Phi_k)/\tau\right)}. \tag{13}$$

Multiclass extension (direct balanced grouping). A straightforward extension is to construct per-class groups of size p (where p may be chosen as $\min_c n_c$ or another target sample count). For each class c we form $K_c = \lceil n_c/p \rceil$ groups; groups may be formed either *without replacement* (partitioning the class pool across groups) or *with replacement* (random sampling). Balanced mini-batches are assembled by concatenating one or more per-class groups so that each selected class contributes approximately p examples to the batch. Positives for an example are other examples in the batch that share the same class; negatives are examples of other classes. This construction removes per-update dominance by any single class and directly generalizes the binary BBSCL grouping mechanism.

Multilabel extension (label-aware grouping). For multilabel tasks, define positive pairs as those that share at least one label. To balance label influence we maintain per-label reservoirs (partitioned or randomly sampled) and form groups such that each label is represented roughly equally within a group. When examples carry multiple labels, their positive contributions are distributed according to overlap weight w_{ij} (Eq. 13). This preserves the intuition of balanced contributions while respecting multilabel structure.

Pseudocode: multiclass BBSCL (random sampling variant). The following pseudocode gives an implementation recipe for the multiclass random-sampling variant of BBSCL. Replace the sampling line with a partitioning step to implement the 'partition' (without replacement) variant.

Algorithm 1: BBSCL – Multiclass (random sampling variant)

Input: Dataset partitioned by class $\mathscr{D}_1, \ldots, \mathscr{D}_C$; target per-class group size p; number of classes per batch $m \leq C$; encoder Φ; temperature τ.

for *each epoch* **do**
  for *each mini-step* **do**
    for *each class* $c \in \{1, \ldots, C\}$ **do**
      Sample (with replacement) a group G_c of p examples from $\mathscr{D}_c$;
    end
    Select a subset $S \subseteq \{1, \ldots, C\}$ of m classes;
    Form a balanced batch: Batch $\leftarrow$ $_{c \in S}\, G_c$;
    Compute embeddings: Φ(Batch);
    For each sample i in Batch, define $P(i) = \{j : y_j = y_i\}$;
    Compute loss $\mathscr{L} = \sum_i \mathscr{L}_i$ using Eq. (12) (or Eq. (13) for multilabel weighting);
    Update model parameters by backpropagation.
  end
end

https://doi.org/10.1017/nlp.2026.10013 Published online by Cambridge University Press

Complexity and practical modifications. The dominant cost per update is the in-batch pairwise similarity computation, which grows as $O(B^2)$ for a batch of size B. Because balanced batches typically increase the number of classes represented, the squared-term may grow; in practice one can reduce cost while preserving balance objectives by:

(1) **Prototype-based contrastive terms:** compute a class prototype (mean of class embeddings) and contrast examples against prototypes instead of all in-batch examples, reducing cost to $O(B \cdot C)$ per update.
(2) **Importance-weighted sampling:** sample classes with probabilities that upweight rare classes (instead of enforcing exact per-batch equality), yielding a smoother trade-off between computation and class-coverage.
(3) **Smaller projection heads or lower projection dimension:** to reduce memory and matrix-multiplication cost when computing similarity matrices.

Evaluation recommendations. When reporting results for multiclass or multilabel tasks, we recommend providing both micro- and macro-averaged metrics (e.g., accuracy/F1/mAP). Macro-averaged metrics are particularly informative in imbalanced settings because they give equal weight to each class and therefore better reflect performance improvements on rare classes induced by BBSCL-style balancing.

7.5 Limitations and future work

While our study demonstrates the effectiveness of SCL and BBSCL in text classification tasks, there are several limitations and opportunities for future research. First, our experiments are limited to text classification, and it remains to be seen whether SCL can be effectively applied to other NLP tasks, such as text generation, summarization, or machine translation. For example, extending SCL to generation tasks could involve contrasting different paraphrases of the same input text to improve output diversity and quality.

Second, our study focuses on few-shot settings, and while BBSCL shows promise in imbalanced scenarios, its performance in larger datasets with extreme class imbalances (e.g., 1:100 ratios) warrants further investigation. Additionally, exploring the scalability of BBSCL with larger batch sizes and more powerful hardware could reveal further performance gains.

Finally, our work primarily uses RoBERTa-large and BERT-large as base PLMs. Testing SCL and BBSCL with even larger models, such as GPT-2 or T5, could provide insights into the generalizability of our methods across different architectures and scales.

7.6 Practical recommendations

Based on our findings, we offer the following recommendations for practitioners:

- **Use SCL for Few-Shot Learning**: When labeled data are limited, integrating SCL into soft-prompt tuning frameworks like DART or PTuning can significantly improve performance.
- **Apply BBSCL for Imbalanced Data**: In datasets with skewed class distributions, BBSCL stabilizes training and ensures fair representation of minority classes.
- **Leverage Small PLMs for Domain-Specific Tasks**: Fine-tuned small PLMs with SCL/BBSCL often outperform large LLMs in specific tasks while being more resource-efficient.
- **Consider LLMs for Zero-Shot or General Tasks**: For tasks requiring broad generalization or zero-shot capabilities, LLMs like GPT-3.5 remain a strong choice despite their higher costs.

https://doi.org/10.1017/nlp.2026.10013 Published online by Cambridge University Press

8. Conclusion

In this paper, we explored the role of SCL in enhancing soft-prompt tuning methods for few-shot learning. We proposed a novel approach that incorporates an SCL objective into DART and PTuning and demonstrated its effectiveness on various few-shot datasets. Our results showed that our approach not only improved the performance on standard benchmarks but also increased the robustness and generalization ability on label-imbalanced and real-world datasets. Furthermore, we revealed a positive relationship between the label complexity of a few-shot dataset and the benefit of SCL.

Supplementary material. The supplementary material for this article can be found at https://doi.org/10.1017/nlp.2026.10013.

Funding statement. No funding was received for conducting this study.

Competing interests. The authors declare that they have no competing interests in the field of the research presented. There are no financial conflicts or other competing interests that might have influenced the outcomes or interpretations of the findings. This ensures that the research was conducted and reported without bias.

Authors contributions. Ali Edalat and Yadollah Yaghoobzadeh jointly conceptualized the study and developed the methodology. Ali Edalat was primarily responsible for software development, data curation, formal analysis, investigation, and visualization. He also wrote the initial draft of the manuscript. Yadollah Yaghoobzadeh provided resources, supervised the project, administered the overall research activities, and contributed to reviewing and editing the manuscript. Both authors reviewed and approved the final manuscript.

Availability of data and materials. The data, PLM models, and open-source LLM supporting the findings of this study are publicly available. Our research involves both a proprietary closed-source large language model and an open-source large language model. The API for the closed-source LLM is also accessible to the public. The data and code for this work are provided as supplementary material alongside this paper. They will also be available in a GitHub[6] repository to ensure reproducibility.

Limitations. Our SCL modules are based on in-batch contrastive learning, which may be sensitive to the batch size, but we had memory constraints and could not test larger batch sizes. Finally, our study focused mainly on text classification tasks, following the previous works on soft-prompt tuning, and did not consider other types of natural language processing tasks.

Ethics. Our work relies on PLMs and LLMs, which have been shown to be biased in prior works (Liang *et al.* 2021; He *et al.* 2023). This should be considered when using such models.

References

Abaskohi A., **Rothe S.** and **Yaghoobzadeh Y.** (2023). LM-CPPF: Paraphrasing-Guided data augmentation for contrastive prompt-based few-shot fine-tuning. In Rogers, A., Boyd-Graber, J. L. and Okazaki, N. (eds), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics* (Volume 2: Short Papers). Toronto, Canada: Association for Computational Linguistics, pp. 670–681.

Alex N., **Lifland E.**, **Tunstall L.**, **Thakur A.**, **Maham P.**, **Riedel C.J.**, **Hine E.**, **Ashurst C.**, **Sedille P.**, **Carlier A.**, **Noetel M.** and **Stuhlmüller A.** (2021). Raft: A real-world few-shot text classification benchmark. In Vanschoren, J. and Yeung, S.-K. (eds), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021*, December 2021, virtual. Association for Computational Linguistics, pp. 1752–1767.

Chen G., **Pu J.**, **Xi Y.** and **Zhang R.** (2022a). Coherent long text generation by contrastive soft prompt. In Bosselut, A., Chandu, K., Dhole, K., Gangal, V., Gehrmann, S., Jernite, Y., Novikova, J. and Perez-Beltrachini, L. (eds), *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*. Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics, pp. 445–455.

Chen G., **Qian Y.**, **Wang B.** and **Li L.** (2023a). MPrompt: Exploring multi-level prompt tuning for machine reading comprehension. In Bouamor, H., Pino, J. and Bali, K. (eds), *Findings of the Association for Computational Linguistics: EMNLP 2023*, Singapore, December 6–10, 2023. Singapore: Association for Computational Linguistics, pp. 5163–5175.

[6] https://github.com/AliEdalat/Supervised-Contrastive-Learning-in-Few-Shot-Soft-Prompt-Tuning.git

Chen J., **Zhang R.**, **Mao Y.** and **Xu J.** (2022). ContrastNet: A contrastive learning framework for few-shot text classification. In *Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI 2022)*. AAAI Press, pp. 10492–10500.

Chen Q., **Zhang R.**, **Zheng Y.** and **Mao Y.** (2022c). Dual contrastive learning: Text classification via label-aware data augmentation.

Chen X., **Xie X.**, **Bi Z.**, **Ye H.**, **Deng S.**, **Zhang N.** and **Chen H.** (2021). Disentangled contrastive learning for learning robust textual representations. In Fang, L., Chen, Y., Zhai, G., Wang, Z.J., Wang, R. and Dong, W. (eds), *Artificial Intelligence - First CAAI International Conference*, CICAI 2021, Hangzhou, China, June 5–6, 2021, Proceedings, Part II. Lecture Notes in Computer Science, Vol. 13070. Berlin: Springer, pp. 215–226.

Chen X., **Zhang W.**, **Pan S.** and **Chen J.** (2023b). Solving data imbalance in text classification with constructing contrastive samples. *IEEE Access* **11**, 90554–90562.

Chien J.-T., **Chen M.-Y.** and **Xue J.-H.** (2023). Learning meta soft prompt for few-shot language models. In *2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 57–62.

Choi S., **Jeong M.**, **Han H.** and **Hwang S.-W.** (2022). C2L: Causally contrastive learning for robust text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 10526–10534.

Choi Y. and **Lee J.-H.** (2023). CodePrompt: Task-agnostic prefix tuning for program and language generation. In Rogers, A., Boyd-Graber, J. and Okazaki, N. (eds), *Findings of the Association for Computational Linguistics: ACL 2023*. Toronto, Canada: Association for Computational Linguistics, pp. 5282–5297.

Cui J., **Zhong Z.**, **Liu S.**, **Yu B.** and **Jia J.** (2021). Parametric contrastive learning. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 695–704.

Devlin J., **Chang M.**, **Lee K.** and **Toutanova K.** (2019). BERT: Pre-Training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C. and Solorio, T. (eds), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT 2019, Volume 1 (Long and Short Papers). Minneapolis, MN, USA: Association for Computational Linguistics, pp. 4171–4186.

Du M., **He F.**, **Zou N.**, **Tao D.** and **Hu X.** (2024). Shortcut learning of large language models in natural language understanding. *Communications of the ACM* **67**(1), 110–120.

Elfrink A., **Vagliano I.**, **Abu-Hanna A.** and **Calixto I.** (2023). Soft-prompt tuning to predict lung cancer using primary care free-text dutch medical notes.

Gao T., **Fisch A.** and **Chen D.** (2021a). Making pre-trained language models better few-shot learners. 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)*. Online. Association for Computational Linguistics, pp. 3816–3830.

Goswami K., **Lange L.**, **Araki J.** and **Adel H.** (2023). SwitchPrompt: Learning domain-specific gated soft prompts for classification in low-resource domains. In Vlachos, A. and Augenstein, I. (eds), In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, Dubrovnik, Croatia: Association for Computational Linguistics, pp. 2689–2695.

Gunel B., **Du J.**, **Conneau A.** and **Stoyanov V.** (2020). Supervised contrastive learning for pre-trained language model fine-tuning.

He J., **Lin N.**, **Shen M.**, **Zhou D.** and **Yang A.** (2023). Exploring bias evaluation techniques for quantifying large language model biases. In *2023 International Conference on Asian Language Processing (IALP)*. IEEE, pp. 265–270.

Jian Y., **Gao C.** and **Vosoughi S.** (2022). Contrastive learning for prompt-based few-shot language learners. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Seattle, United States: Association for Computational Linguistics, pp. 5577–5587.

Jiang Y., **Zhang L.** and **Wang W.** (2022). Improved universal sentence embeddings with prompt-based contrastive learning and energy-based learning. In Goldberg, Y., Kozareva, Z. and Zhang, Y. (eds), *Findings of the Association for Computational Linguistics: EMNLP 2022*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 3021–3035.

Lampinen A., **Dasgupta I.**, **Chan S.**, **Mathewson K.**, **Tessler M.**, **Creswell A.**, **McClelland J.**, **Wang J.** and **Hill F.** (2022). Can language models learn from explanations in context?. In Goldberg, Y., Kozareva, Z. and Zhang, Y. (eds), *Findings of the Association for Computational Linguistics: EMNLP 2022*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 537–563.

Layegh A., **Payberah A.H.**, **Soylu A.**, **Roman D.** and **Matskin M.** (2023). Contrastner: Contrastive-based prompt tuning for few-shot ner. arXiv preprint arXiv: 2305.17951.

Lee T. and **Yoo S.** (2021). Augmenting few-shot learning with supervised contrastive learning. *IEEE Access* **9**, 61466–61474.

Li X.L. and **Liang P.** (2021). Prefix-tuning: Optimizing continuous prompts for generation.

Li Z., **Wang H.**, **Świstek T.**, **Yu E.** and **Wang H.** (2023). Efficient few-shot classification via contrastive pretraining on web data. *IEEE Transactions on Artificial Intelligence* **4**, 522–533.

Liang P. P., **Wu C.**, **Morency L.-P.** and **Salakhutdinov R.** (2021). Towards understanding and mitigating social biases in language models. In *International Conference on Machine Learning*. PMLR, pp. 6565–6576.

Liu X., **Ji K.**, **Fu Y.**, **Tam W.**, **Du Z.**, **Yang Z.** and **Tang J.** (2022). P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Dublin, Ireland: Association for Computational Linguistics, pp. 61–68.

https://doi.org/10.1017/nlp.2026.10013 Published online by Cambridge University Press

Liu X., **Zheng Y.**, **Du Z.**, **Ding M.**, **Qian Y.**, **Yang Z.** and **Tang J.** (2024). GPT understands, too. *AI Open* **5**, 208–215.

Liu Y., **Ott M.**, **Goyal N.**, **Du J.**, **Joshi M.**, **Chen D.**, **Levy O.**, **Lewis M.**, **Zettlemoyer L.** and **Stoyanov V.** (2019). Roberta: A robustly optimized bert pretraining approach.

Ma F., **Zhang C.**, **Ren L.**, **Wang J.**, **Wang Q.**, **Wu W.**, **Quan X.** and **Song D.** (2022). XPrompt: Exploring the extreme of prompt tuning. In Goldberg, Y., Kozareva, Z. and Zhang, Y. (eds), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 11033–11047.

McInnes L., **Healy J.**, **Saul N.** and **Großberger L.** (2018). UMAP: Uniform Manifold Approximation and Projection. *Journal of Open Source Software* **3**(29), 861.

Møller A.G., **Dalsgaard J.A.**, **Pera A.** and **Aiello L.** (2023). Is a prompt and a few samples all you need? using GPT-4 for data augmentation in low-resource classification tasks. ArXiv, abs/2304.13861.

OpenAI (2023). ChatGPT. Available at https://openai.com/blog/chatgpt.

Pan L., **Hang C.-W.**, **Sil A.** and **Potdar S.** (2022). Improved text classification via contrastive adversarial training. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event*, February 22 - March 1, 2022. AAAI Press, pp. 11130–11138.

Razdaibiedina A., **Mao Y.**, **Khabsa M.**, **Lewis M.**, **Hou R.**, **Ba J.** and **Almahairi A.** (2023). Residual prompt tuning: improving prompt tuning with residual reparameterization. In Rogers, A., Boyd-Graber, J. and Okazaki, N. (eds), *Findings of the Association for Computational Linguistics: ACL 2023*. Toronto, Canada: Association for Computational Linguistics, pp. 6740–6757.

Sahoo A., **Panda R.**, **Senapati A.**, **Das A.**, **Kim Y.** and **Feris R.** (2023). Contrastive prompt tuning improves generalization in vision-language models.

Schick T. and **Schütze H.** (2021c). Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Online. Association for Computational Linguistics, pp. 255–269.

Schick T. and **Schütze H.** (2021b). It's not just size that matters: Small language models are also few-shot learners. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tür, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T. and Zhou, Y. (eds), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT 2021, Online, June 6-11, 2021. Association for Computational Linguistics, pp. 2339–2352.

Shi Z. and **Lipani A.** (2024). DePT: Decomposed prompt tuning for parameter-efficient fine-tuning.

Shin T., **Razeghi Y.**, **Logan R.L.**IV, **Wallace E.** and **Singh S.** (2020). AutoPrompt: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online. Association for Computational Linguistics, pp. 4222–4235.

Su Y., **Wang X.**, **Qin Y.**, **Chan C.-M.**, **Lin Y.**, **Wang H.**, **Wen K.**, **Liu Z.**, **Li P.**, **Li J.**, **Hou L.**, **Sun M.** and **Zhou J.** (2022). On transferability of prompt tuning for natural language processing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Suhaeni C. and **Yong H.-S.** (2024). Enhancing imbalanced sentiment analysis: A GPT-3-based sentence-by-sentence generation approach. *Applied Sciences* **14**(2), 622.

Tian J.-J., **Emerson D.**, **Miyandoab S.Z.**, **Pandya D.**, **Seyyed-Kalantari L.** and **Khattak F.K.** (2023). Soft-prompt tuning for large language models to evaluate bias. arXiv preprint arXiv: 2306.04735.

Wang F., **Chen L.**, **Xie F.**, **Xu C.** and **Lu G.** (2022). Few-shot text classification via semi-supervised contrastive learning. In 2022 4th International Conference on Natural Language Processing (ICNLP), pp. 426–433.

Wang G., **Cheng S.**, **Zhan X.**, **Li X.**, **Song S.** and **Liu Y.** (2024). OpenChat: Advancing open-source language models with mixed-quality data. In *The Twelfth International Conference on Learning Representations (ICLR 2024)*. Vienna, Austria: OpenReview.net.

Wang L., **Yang N.** and **Wei F.** (2024a). Learning to retrieve in-context examples for large language models. In Graham, Y. and Purver, M. (eds), *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*. St. Julian's, Malta: Association for Computational Linguistics, pp. 1752–1767.

Wang S., **Fang H.**, **Khabsa M.**, **Mao H.** and **Ma H.** (2021). Entailment as few-shot learner.

Wang Y., **Yu Z.**, **Wang J.**, **Heng Q.**, **Chen H.**, **Ye W.**, **Xie R.**, **Xie X.** and **Zhang S.** (2024b). Exploring vision-language models for imbalanced learning. *International Journal of Computer Vision* **132**(1), 224–237.

Wu J., **Yu T.**, **Wang R.**, **Song Z.**, **Zhang R.**, **Zhao H.**, **Lu C.**, **Li S.** and **Henao R.** (2023). Infoprompt: Information-Theoretic soft prompt tuning for natural language understanding. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M. and Levine, S. (eds), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023*, New Orleans, LA, USA, December 10-16, 2023.

Xia P., **Xu D.**, **Ju L.**, **Hu M.**, **Chen J.** and **Ge Z.** (2023). LMPT: Prompt tuning with class-specific embedding loss for long-tailed multi-label visual recognition.

https://doi.org/10.1017/nlp.2026.10013 Published online by Cambridge University Press

Xu Z., **Peng K.**, **Ding L.**, **Tao D.** and **Lu X.** (2024). Take care of your prompt bias! investigating and mitigating prompt bias in factual knowledge extraction. In Calzolari, N., Kan, M.-Y., Hoste, V., Lenci, A., Sakti, S. and Xue, N. (eds), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024*, 20-25 May, 2024, Torino, Italy: ELRA and ICCL, pp. 15552–15565.

Zhang N., **Li L.**, **Chen X.**, **Deng S.**, **Bi Z.**, **Tan C.**, **Huang F.** and **Chen H.** (2022). Differentiable prompt makes pre-trained language models better few-shot learners. In *The Tenth International Conference on Learning Representations (ICLR 2022)*. Virtual Event: OpenReview.net.

Zhang S., **Ji T.**, **Ji W.** and **Wang X.** (2022). Zero-shot event detection based on ordered contrastive learning and prompt-based prediction. In *Findings of the Association for Computational Linguistics: NAACL 2022*. Seattle, United States: Association for Computational Linguistics, pp. 2572–2580.

Zhu J., **Wang Z.**, **Chen J.**, **Chen Y.-P. P.** and **Jiang Y.-G.** (2022). Balanced contrastive learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6908–6917.

A. Statistical significance analysis

Table A1. Significance levels of *p*-values for comparisons between DART and its extensions, as well as *P*-Tuning and its extensions, using RoBERTa-large as the pretrained language model in Table 1. Only the average performance across all datasets is considered. Significance markers are assigned as follows: "$***$" for p-value < 0.001, "$**$" for p-value < 0.01, "$*$" for p-value < 0.05, and an empty string otherwise. A hyphen ("–") indicates no comparison was possible

Method	PTuning + SCL	DART + SCL
PTuning	***	-
DART	-	**

Table A2. Significance levels of *p*-values for comparisons between DART and its extensions, as well as P-Tuning and its extensions, using RoBERTa-large as the pretrained language model in Table 2. Only the average performance across all datasets is considered. Significance markers are assigned as follows: "$***$" for p-value < 0.001, "$**$" for p-value < 0.01, "$*$" for p-value < 0.05, and an empty string otherwise. A hyphen ("–") indicates no comparison was possible

Method	PTuning	DART
PTuning + SCL	*	-
PTuning + BBSCL (random)	*	-
PTuning + BBSCL (partition)	*	-
PTuning + SCL (oversample)	***	-
DART + SCL	-	
DART + BBSCL (random)	-	
DART + BBSCL (partition)	-	*
DART + SCL (oversample)	-	**

https://doi.org/10.1017/nlp.2026.10013 Published online by Cambridge University Press

Table A3. Significance levels of *p*-values for comparisons between DART and its extensions, as well as P-Tuning and its extensions, using BERT-large-uncased as the pretrained language model in Table 3, in the imbalanced setting. Only the average performance across all datasets is considered. Significance markers are assigned as follows: "$***$" for p-value < 0.001, "$**$" for p-value < 0.01, "$*$" for p-value < 0.05, and an empty string otherwise. A hyphen ("–") indicates no comparison was possible

Method	PTuning	DART
PTuning + SCL		-
PTuning + BBSCL (random)	**	-
PTuning + BBSCL (partition)	*	-
PTuning + SCL (oversample)	**	-
DART + SCL	-	*
DART + BBSCL (random)	-	***
DART + BBSCL (partition)	-	***
DART + SCL (oversample)	-	***

Table A4. Significance levels of *p*-values for comparisons between DART and its extensions, as well as P-Tuning and its extensions, using BERT-large-uncased as the pretrained language model in Table 3, in the balanced setting. Only the average performance across all datasets is considered. Significance markers are assigned as follows: "$***$" for p-value < 0.001, "$**$" for p-value < 0.01, "$*$" for p-value < 0.05, and an empty string otherwise. A hyphen ("–") indicates no comparison was possible

Method	PTuning	DART
PTuning + SCL	***	-
DART + SCL	-	***

https://doi.org/10.1017/nlp.2026.10013 Published online by Cambridge University Press

Table A5. Significance levels of p-values for comparisons between DART and its extensions, as well as P-Tuning and its extensions in Table 6 when the setting is imbalanced and PLM is RoBERTa-large. Only the average performance across all datasets is considered. Significance markers are assigned as follows: "$***$" for p-value < 0.001, "$**$" for p-value < 0.01, "$*$" for p-value < 0.05, and an empty string otherwise. A hyphen ("–") indicates no comparison was possible

Method	PTuning	DART
PTuning + SCL	**	-
PTuning + BBSCL (random)	**	-
PTuning + BBSCL (partition)	**	-
PTuning + SCL (oversample)	***	-
DART + SCL	-	
DART + BBSCL (random)	-	
DART + BBSCL (partition)	-	
DART + SCL (oversample)	-	*

Table A6. Significance levels of p-values for comparisons between DART and its extensions, as well as P-Tuning and its extensions in Table 6 when the setting is imbalanced and PLM is BERT-large-uncased. Only the average performance across all datasets is considered. Significance markers are assigned as follows: "$***$" for p-value < 0.001, "$**$" for p-value < 0.01, "$*$" for p-value < 0.05, and an empty string otherwise. A hyphen ("–") indicates no comparison was possible

Method	PTuning	DART
PTuning + SCL	***	-
PTuning + BBSCL (random)	*	-
PTuning + BBSCL (partition)	***	-
PTuning + SCL (oversample)	***	-
DART + SCL	-	
DART + BBSCL (random)	-	
DART + BBSCL (partition)	-	
DART + SCL (oversample)	-	

https://doi.org/10.1017/nlp.2026.10013 Published online by Cambridge University Press

Table A7. Significance levels of p-values for comparisons between DART and its extensions in Table 7 when the setting is imbalanced with different ratios. Significance markers are assigned as follows: "$***$" for p-value < 0.001, "$**$" for p-value < 0.01, "$*$" for p-value < 0.05, and an empty string otherwise. A hyphen ("–") indicates no comparison was possible

Ratio	+SCL	+BBSCL
0.1	*	*
0.25		
0.36		*
0.5		*

Table A8. Significance of improvements over base model (Roberta and Bert) in Table 4 when we use balanced setting

Method	Roberta	Bert
SCL	**	**

Table A9. Significance of improvements over base model (Roberta and Bert) in Table 4 when we use an imbalanced setting

Method	Roberta	Bert
SCL	***	***
SCL (over)	***	***
BBSCL (random)	***	***
BBSCL (partition)	***	***

Table A10. Significance of improvements over base model (Roberta and Bert) in Table 5 when we use an imbalanced setting

Method	Roberta	Bert
SCL		*
SCL (over)		*
BBSCL (random)		*
BBSCL (partition)		

https://doi.org/10.1017/nlp.2026.10013 Published online by Cambridge University Press

Table A11. Significance levels of p-values for comparisons between GPT-3.5, openchat, DART, and DART + SCL in Table 8 when the setting is balanced. For GPT-3.5 and openchat, we consider the best performance in different settings. Only the average performance across all datasets is considered. Significance markers are assigned as follows: "***" for p-value < 0.001, "**" for p-value < 0.01, "*" for p-value < 0.05, and an empty string otherwise. A hyphen ("–") indicates no comparison was possible

Method	GPT-3.5	openchat	DART	DART + SCL
GPT-3.5	-	***	**	***
openchat	***	-	***	***
DART	**	***	-	***
DART + SCL	***	***	***	-

Table A12. Significance levels of p-values for comparisons between GPT-3.5, openchat, DART, DART + SCL, DART + BBSCL (random), DART + BBSCL (partition), and DART + SCL (over) in Table 8 when the setting is imbalanced. Only the average performance across all datasets is considered. Significance markers are assigned as follows: "***" for p-value < 0.001, "**" for p-value < 0.01, "*" for p-value < 0.05, and an empty string otherwise. A hyphen ("–") indicates no comparison was possible

Method	GPT-3.5	openchat	DART	DART + SCL	DART + BBSCL (random)	DART + BBSCL (partition)	DART + SCL (over)
GPT-3.5	-	***	***		**	*	***
openchat	***	-	***	***	***	***	***
DART	***	***	-	***	***	***	***
DART + SCL		***	***	-	*		*
DART + BBSCL (random)	**	***	***	*	-		
DART + BBSCL (partition)	*	***	***			-	
DART + SCL (over)	***	***	***	*			-

Cite this article: Edalat A and Yaghoobzadeh Y (2026). Supervised contrastive learning in few-shot soft prompt tuning. *Natural Language Processing* **32**, 322–356. https://doi.org/10.1017/nlp.2026.10013