CRL-Prompt: Contrastive and Reinforcement Learning for Soft Prompt Tuning for Text Classification

Danila Lapokin[1], **Andrey Savchenko**[1,2,3]
[1]HSE University, [2]Sber AI Lab, [3]ISP RAS Research Center for Trusted Artificial Intelligence

Abstract

Prompt choice is crucial in adapting language models to text classification tasks, particularly under low-resource conditions. Manual prompt engineering is time-consuming, non-scalable, and brittle, while current auto-prompting techniques are still far from maturity. This paper presents a two-stage method for prompt learning of frozen language models, CRL-Prompt, based on soft prompt initialization followed by contrastive and reinforcement-based refinement. An experimental study demonstrates that our approach achieves consistent improvements in accuracy over baseline prompt tuning strategies, with gains of up to 2.2% while training fewer than 0.25% of model parameters.

1 Introduction

Nowadays, text classification is typically solved by BERT-like language models (LMs) due to their speed and efficiency (Siino et al., 2025; Savchenko and Savchenko, 2025). It is generally essential to use pre-trained models (Wu and Wan, 2025), as practical applications, such as smart speakers, may require solving multiple classification tasks, and only one LM should be used due to memory constraints (Edwards and Camacho-Collados, 2024; Stigall et al., 2024). Hence, a key factor in LM's adaptability is the choice of *prompt*, input sequence that conditions the model's behavior and outputs (Brown et al., 2020). Prompt engineering has emerged as a critical mechanism for controlling and adapting LMs in parameter-efficient manner (Chen et al., 2024; Marvin et al., 2023; Peng et al., 2024). Though prompt engineering today remains manual and heuristic in practice (Sahoo et al., 2024), automated prompt learning (Chang et al., 2024; Spiess et al., 2025; Xiao et al., 2025) has been recently studied from three major directions (Shin et al., 2020; Li et al., 2023; Zhuge et al., 2024). The first is *discrete prompt search*, where token-level prompt candidates are generated and scored using surrogate metrics, as seen in methods like TextGrad (Yuksekgonul et al., 2024) and Automatic Prompt Engineer (APE) (Zhou et al., 2022). Second, *continuous soft prompt tuning* directly learns embeddings and virtual prompt vectors that are optimized via gradient descent on a frozen language model; notable examples include P-Tuning v1 (Liu et al., 2021a) and v2 (Liu et al., 2021b), as well as MixtureSoft (Qin and Eisner, 2021). Finally, *Reinforcement Learning (RL)–based refinement* methods, such as ConsPrompt (Weng et al., 2024), adapt prompts by optimizing for task-specific rewards. While these techniques have demonstrated promise, they also exhibit limitations: discrete methods can be computationally expensive and brittle; continuous tuning may overfit in low-resource regimes due to proxy losses; and RL often requires complex reward shaping and unstable training dynamics.

We address these shortcomings by introducing a two-stage open-source CRL-Prompt framework[1] for automated soft prompt learning. In the first stage, P-Tuning v2 (Liu et al., 2021b) initializes trainable key/value vectors injected into all transformer layers. In the second stage, we refine the prompts using a combination of contrastive regularization (Chen et al., 2020; Weng et al., 2024; Yu et al., 2020) and reinforcement feedback-based policy optimization guided by task-level accuracy. The former term is designed to enhance the robustness of learned prompts by improving representation geometry. The latter term aligns optimization more closely with the classification task. Our method is fully compatible with frozen LMs and does not modify their parameters. The experimental results demonstrate that our framework consistently outperforms state-of-the-art baselines, achieving gains

[1] https://github.com/danilalapokinofficial/autoprompting

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop), pages 1391–1398

of up to 2.2% in accuracy while training less than 0.25% of model parameters.

2 Related Work

Discrete Prompt Search. Early work in prompt engineering explored discrete search over token templates to elicit knowledge from frozen LMs. AutoPrompt (Shin et al., 2020) uses gradient signals to select informative trigger tokens. APE (Zhou et al., 2022) employs a language model to generate and rank natural language instructions. While fully automatic, these methods are limited to template-level search and do not learn prompts to optimize the downstream metrics.

Soft Prompt Tuning. Continuous prompt tuning approaches replace discrete templates with learnable embeddings. P-Tuning (Liu et al., 2021a) and its extension P-Tuning v2 (Liu et al., 2021b) inject virtual key/value vectors into frozen transformer layers, achieving strong performance with fewer parameters. MixtureSoft (Qin and Eisner, 2021) learns multiple prompt variants and averages their outputs. However, such methods rely on cross-entropy as a proxy loss, which does not always align with end-task accuracy.

Reinforcement and Contrastive Methods. Recent work incorporates RL to refine prompt parameters using task-level rewards. DP2O (Li et al., 2023) and GPTSwarm (Zhuge et al., 2024) generate prompt policies or agent graphs via policy gradients. ConsPrompt (Weng et al., 2024) introduces contrastive loss to improve few-shot generalization. These approaches improve robustness but often involve complex architectures or require substantial reward engineering.

3 Proposed Approach

Let f_θ be a frozen LM with parameters θ, and let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ be a labeled text dataset for a classification task, i.e., $y_i \in \{1, ..., C\}$, where C is the number of classes. We aim to find prompt parameters P (e.g., soft embeddings or prefix vectors) that guide the LM toward accurate predictions, while keeping its weights θ fixed. It is required to choose P that maximizes classification accuracy $r(P)$ on a held-out validation set $\mathcal{D}_{\text{val}}$:

$$r(P) = \frac{1}{|\mathcal{D}_{\text{val}}|} \sum_{(x,y)\in\mathcal{D}_{\text{val}}} \mathbf{1}\left[f_\theta(P, x) = y\right], \quad (1)$$

where $\mathbf{1}[\cdot]$ is the indicator function, and $f_\theta(P, x) \in \mathbb{R}^C$ are the logits produced by the frozen LM under prompt P for input x_i.

Algorithm 1 Proposed CRL-Prompt approach

Require: Frozen LM f_θ, initial prompt P_0, training $\mathcal{D}$ and validation $\mathcal{D}_{\text{val}}$ sets, number of steps T, reward-update interval M, weights β, γ, noise variance σ^2

Phase 1: Prompt tuning via cross-entropy
for each batch (x_i, y_i) from $\mathcal{D}$ **do**
  Compute $L_{\text{CE}}(P, x_i, y_i)$
  Update P using gradient of L_{CE}
end for
Phase 2: Mixed optimization loop
for $t = 1$ **to** T **do**
  for each batch (x_i, y_i) from $\mathcal{D}$ **do**
    Compute $L_{\text{CE}}(P, x_i, y_i)$
    Compute $L_{\text{contrast}}(P, x_i, y_i)$
    if $t \bmod M = 0$ **then**
      Sample prompt $P' \sim \mathcal{N}(P, \sigma^2 I)$
      Evaluate accuracy $r(P')$ (1)
      Compute RL loss: L_{RL} (4)
    else
      $L_{\text{RL}} \leftarrow 0$
    end if
    $L_{\text{total}} \leftarrow L_{\text{CE}} + \beta\, L_{\text{contrast}} + \gamma\, L_{\text{RL}}$
    Update P using gradient of L_{total}
  end for
end for
return final prompt P

To solve this problem, we propose a two-stage automated prompt optimization framework (Algorithm 1). The first phase applies an arbitrary technique, e.g., P-Tuning v2 (Liu et al., 2021b), to initialize trainable soft prompts. They consist of virtual key/value vectors injected at each transformer layer of a frozen LM. Only these prompt embeddings are updated during training, keeping the model weights unchanged. The training objective in this phase is standard cross-entropy loss:

$$L_{\text{CE}} = -\log\Big[softmax\big(f_\theta(P, x_i)\big)_{y_i}\Big]. \quad (2)$$

Once a stable initialization is established, we proceed to the second phase, where we utilize contrastive regularization and RL-based optimization to refine the soft prompts. While standard soft prompt tuning minimizes a cross-entropy loss over labeled examples, it does not explicitly encourage the prompt to separate semantically similar inputs from different classes. This can lead to overfitting or instability in few-shot settings, particularly when

initialization is noisy or training data is limited. To address this, we introduce a contrastive loss term that improves the representational geometry of the learned prompt space. Given a batch of examples $\{(x_i, y_i)\}$, negative prompt variants are generated for each batch by applying dropout or noise to the current prompt parameters. The InfoNCE-style loss (Chen et al., 2020) is used to encourage embeddings of correct predictions to cluster together, while pushing apart those of incorrect ones:

$$\mathcal{L}_{\text{contrast}} = -\log \frac{1}{1 + \sum_{j=1}^{k} e^{(\text{sim}(h_i, h_j^-)-1)/\tau}}, \quad (3)$$

where sim is a cosine similarity, τ is a temperature hyperparameter, h_i is the sequence-level embedding (e.g., the CLS or pooled output) of input x_i with prompt P, and h_j^- are negative examples generated by applying a simple dropout-inspired prompt perturbation: we create a masked variant of the prompt-encoder parameters by sampling a mask and zeroing out 10% of embedding values, then computing the LM representation under this perturbed prompt. This yields a nearby but corrupted prompt-derived embedding as a negative. The rationale here is that masked prompt variants emulate small, realistic prompt corruptions, so InfoNCE enforces local robustness of the mapping prompt to representation without requiring external data augmentations.

Periodically (every M steps), we run the REINFORCE algorithm (Williams, 1992). The soft prompt is perturbed with Gaussian noise ($\pi(P'|P) = \mathcal{N}(P, \sigma^2 I)$), and a sampled prompt P' is used to evaluate validation accuracy $r(P')$ (1). The policy-gradient loss is used to tune the prompt that maximizes this reward:

$$L_{\text{RL}}(P; P', \mathcal{D}_{\text{val}}) = -r(P') \log\big[\pi(P'|P)\big]. \quad (4)$$

The final loss combines cross-entropy, contrastive, and RL terms with scalar weights:

$$L_{\text{total}} = L_{\text{CE}} + \beta\, L_{\text{contrast}} + \gamma\, L_{\text{RL}}. \quad (5)$$

Here, the contrastive component can be viewed as a local dropout regularizer on the mapping from soft prompts to hidden representations: by pulling together clean and masked prompt variants and pushing them away from corrupted ones, it reduces sensitivity to small prompt perturbations. This stabilizes training and improves robustness in low-data regimes. The RL component treats the soft prompt as a policy that induces predictions from a frozen LM and directly optimizes task-level, potentially non-differentiable metrics, such as accuracy(1). In combination, contrastive smoothing reduces representation drift and gradient variance, while RL provides a mechanism to optimize objectives that cross-entropy alone cannot capture.

Unlike frameworks that rely on prompt embeddings or classifier-based scoring, our Algorithm 1 works entirely within the training loop of a frozen LM. This makes our method efficient and deployable in real-world few-shot scenarios with limited data and compute. Unlike discrete methods, our CRL-Prompt operates over continuous prompts; unlike pure soft tuning, it optimizes prompts directly for the downstream task metric; and unlike full RL pipelines, it maintains low compute cost by combining contrastive loss with lightweight REINFORCE updates. Compared to ConsPrompt (Weng et al., 2024), our framework does not require batch-level sampling or re-ranking and is compatible with standard PEFT libraries.

4 Experimental Results

We compare our method to *HandCraft* (manual) prompts and both "soft" and "discrete" prompt-engineering baselines that share the same constraints: frozen backbones and very small numbers of trainable parameters (0.15–0.25%): APE (Zhou et al., 2022), ConsPrompt (Weng et al., 2024), MixtureSoft (Qin and Eisner, 2021), Prompt v1 (Liu et al., 2021a), Prompt v2 (Liu et al., 2021b) and TextGrad (Yuksekgonul et al., 2024). We especially demonstrate the possibility for our method to work with different architectures, so all experiments use not only standard RoBERTa-base (Liu et al., 2019) with 125M parameters, but also decoder-only (Savchenko, 2013) falcon-rw-1b (Penedo et al., 2023) with 1B parameters with a conventional verbalizer. In contrast to our approach, ConsPrompt and MixtureSoft are only available for RoBERTa because they use maskedLM mode, which Falcon does not support.

We conduct experiments on standard benchmarks with the train/test splits provided by their authors: 1) **AG News** (Zhang et al., 2015) – 4-class news topic classification, 120K training and 7.6K testing examples; 2) **TREC** (Li and Roth, 2002) – 6-class question type classification with 5.5K training and 500 testing examples; and 3) **SST–2** (Stanford Sentiment Treebank) (Socher et al., 2013) –

Table 1: Main results: accuracy on the test set.

	AG News		TREC		SST-2		EmphateticDialogues	
Method	**RoBERTa-base**	**falcon-rw-1b**	**RoBERTa-base**	**falcon-rw-1b**	**RoBERTa-base**	**falcon-rw-1b**	**RoBERTa-base**	**falcon-rw-1b**
HandCraft	0.470±0.00	0.450±0.000	0.500±0.000	0.390±0.000	0.730±0.000	0.790±0.000	0.272±0.000	0.203±0.000
TextGrad	0.700 ±0.004	0.880±0.003	0.550±0.003	0.395±0.002	0.734 ±0.005	0.810±0.004	0.363±0.006	0.306±0.005
APE	0.550±0.003	0.520±0.003	0.546 ±0.003	0.410±0.004	0.845±0.002	0.827±0.002	0.372±0.005	0.318±0.004
MixtureSoft	0.888±0.004	–	0.656±0.005	–	0.922±0.001	–	–	–
ConsPrompt	0.839±0.005	–	0.691±0.003	–	0.892±0.002	–	–	–
Prompt v1	0.927±0.003	0.929±0.002	0.922±0.004	0.672±0.007	0.898±0.001	0.948±0.001	0.437±0.005	0.385±0.007
Prompt v2	0.932±0.001	0.932±0.001	0.938±0.002	0.928±0.001	0.922±0.002	0.903±0.003	0.454±0.004	0.391±0.006
Our CRL-Prompt	**0.940±0.002**	**0.946±0.003**	**0.960 ±0.002**	**0.962±0.001**	**0.940 ±0.003**	**0.943 ±0.001**	**0.474±0.007**	**0.402±0.005**

binary sentiment classification with 67K training and 1.8K testing examples.

For the proposed CRL-prompt, we used a small held-out split (10% of the original training set) as the validation/development set to estimate the accuracy of our RL procedure, while the remaining 90% was used for training. This validation set is distinct from the untouched test set used for final reporting. Thus, it is simply a part of the original training set, so the comparison with other techniques is fair. The validation signal is used as the RL reward proxy (i.e., CRL-Prompt treats the prompt as a policy and evaluates expected validation reward), which is conceptually similar to using a development metric for hyperparameter selection or early stopping rather than tapping test labels.

In addition, we consider a more complicated **EmpatheticDialogues** dataset (Rashkin et al., 2019), with 84170, 12078, 10973 phrases in train, validation, and test sets labeled by 32 emotion classes. Appendix A contains additional details.

Table 1 summarizes mean accuracy on test sets after 10 runs. We ran all experiments on a single Nvidia A100 GPU. The total training time for each method was restricted by 2 hours. On the AG News dataset, our method improves upon the best-performing baseline (Prompt v2) by 0.8% and 1.4% for RoBERTa and Falcon, respectively. The gains are significant given Prompt v2's strong performance. For SST-2, a binary sentiment task, our approach yields 94.0% with RoBERTa and 94.3% with Falcon. Although absolute gains over Prompt v2 and MixtureSoft are slightly smaller (1.8–2.0%), they are consistent across architectures. The results also show that MixtureSoft, while effective for RoBERTa, does not generalize to larger models like Falcon, highlighting a strength of our approach. Preliminary study on SST-2 using Qwen3-Embedding-4B (Zhang et al., 2025) also demonstrated that our CRL-Prompt achieves an accuracy of 95.5%, which is 3% higher than the result (92.32%) of the best baseline (Prompt v2).

TREC is a small, fine-grained 6-way question classification task. In such low-resource settings, results are inherently more variable. In particular, Prompt v2 baseline injects soft prompts into every transformer layer, significantly increasing its capacity and stability on small or challenging tasks by utilizing layer-wise prompts. In contrast, P-Tuning v1 only augments the input layer and may overfit or underperform on larger models or limited data. According to Table 1, v1's shallower design did not generalize well to the 1B-param model under scarce training data, whereas v2's richer prompts mitigated that effect. However, our method significantly outperforms Prompt v2 by 2.2% and 3.4% for RoBERTa and Falcon, respectively. This gap is the largest among the datasets, which we attribute to TREC's small size and fine-grained nature. In low-resource settings, the robustness introduced by contrastive regularization and alignment with end-task rewards proves especially beneficial.

For our most complex dataset, EmphateticDialogues, the proposed CRL-Prompt is again the most accurate technique, achieving accuracies over 47% and 40% with RoBERTa and Falcon, respectively. It is worth noting that the former metric is higher than the results of specialized techniques on this dataset: 36.57% of KEMP (Li et al., 2022) and 36.84% of CEM (Sabour et al., 2022). Although these methods are designed to generate empathetic responses, they also perform emotion classification as part of their training. As a result, their authors report emotion recognition accuracy on the EmphateticDialogues dataset and compare their results with those of other methods.

Thus, across all datasets, our approach consistently outperforms discrete search (APE, TextGrad), pure soft tuning (Prompt v1/v2), and reinforcement-only methods (ConsPrompt). Moreover, it achieves high parameter efficiency by updating only the soft prompt parameters while keeping the backbone LM frozen. Indeed, we use 20 learnable virtual tokens per transformer layer. For

Table 2: Ablation results: test accuracy for different loss variants.

	AG News		TREC		SST-2	
Method Variant	**RoBERTa-base**	**falcon-rw-1b**	**RoBERTa-base**	**falcon-rw-1b**	**RoBERTa-base**	**falcon-rw-1b**
P-Tuning v2 only ($\beta = 0, \gamma = 0$)	0.932±0.001	0.932±0.001	0.938±0.002	0.928±0.001	0.922±0.002	0.903±0.003
+ Contrastive only ($\beta > 0, \gamma = 0$)	0.936±0.002	0.940±0.002	0.953±0.003	0.951±0.001	0.931±0.003	0.930±0.002
+ RL only ($\beta = 0, \gamma > 0$)	0.935±0.001	0.938±0.003	0.948±0.002	0.946±0.001	0.927±0.002	0.928±0.001
Full: Contrastive + RL ($\beta > 0, \gamma > 0$)	**0.940±0.002**	**0.946±0.003**	**0.960±0.002**	**0.962±0.001**	**0.940±0.003**	**0.943±0.001**

instance, RoBERTa-base consists of 12 layers with a hidden size of 768, resulting in $20 \times 12 \times 768 = 184{,}320$ trainable parameters, which is less than 0.15% of the 125M parameters of the whole model. For Falcon-RW-1B, the number of updated parameters $32 \times 20 \times 4544 = 2{,}908{,}160$ remains approximately equal to 0.25% out of 1B weights in the model. Thus, our method is well-suited for settings with limited computational resources or constraints on model modification.

Though we tried to keep the comparison focused on the prompt-as-adapter design space and frozen LM, it is possible to compare the results with PEFT baselines, such as LoRA, or even SFT (Supervised Fine-Tuning). In our preliminary experiment, we fixed the number of trainable parameters to approximately match our prompt learning (185K, i.e., 0.15% of the total number of parameters) by setting rank 2 for only part of the layers. As a result, LoRA's maximal accuracy for RoBERTa is 88.9% for SST-2 and 87.6% for TREC, i.e., 4-8% lower than our method (94% for SST-2 and 96% for TREC). For sure, if we increase the number of trainable parameters, we can exceed our results. For instance, setting rank to 32 (1.8M trainable parameters, an order of magnitude higher than for our method) leads to accuracy 94.29% for SST-2 and 98.6% for TREC. Full SFT (125M trainable parameters) is even more accurate: 94.45% for SST-2 and 98.8% for TREC (Wang and Azman, 2025).

To better understand the contribution of each component in our hybrid framework, we perform ablation experiments by selectively disabling parts of the loss function. Table 2 summarizes the impact of each component. Here, combining gradient-based soft prompt tuning with contrastive and RL refinements yields consistent and significant accuracy improvements across tasks and model architectures. The two-stage design is practical: the initial P-Tuning phase provides a strong starting point for optimization, while contrastive regularization enhances generalization by structuring the prompt embedding space via pulling together same-class examples. Although applied infrequently, reinforcement-based updates align the optimization process with the true end-task objective (classification accuracy, Eq. 1), yielding additional performance gains.

5 Conclusion

This paper introduces a novel Algorithm 1 for soft prompt learning without modifying the backbone LM, making it suitable for resource-constrained scenarios and simultaneously solving multiple classification tasks with the same LM. Known prompt-tuning methods typically use either gradient-based methods (e.g., P-Tuning) or RL-based refinement (e.g., ConsPrompt), but not both, as in the proposed approach. Our ablation (Table 2) shows that each component helps: contrastive learning provides representational robustness and intra-class cohesion, and policy-gradient tuning directly maximizes end-task reward. Their combination yields synergistic gains that neither approach achieves on its own. Empirically, CRL-Prompt outperforms both pure soft-tuning (Prompt v1/v2) and RL-only baselines (ConsPrompt) on all datasets (Table 1).

The proposed CRL-Prompt is most beneficial for practical usage when (a) it is required to adapt a frozen backbone with strict parameter-efficiency constraints, (b) the evaluation metric is non-differentiable or asymmetric (F1, EM, user preference), or (c) training samples are small, so the representation robustness matters. Our approach is usually better than existing prompt-tuning techniques, as they can be incorporated into the first stage of our method. Other alternatives for modifying the backbone via PEFT or SFT are sensible in large-data regimes. Thus, our method's simplicity, modularity, and effectiveness across datasets and models may make it a practical foundation for scalable LM adaptation in various settings.

Limitations

Due to focus on classification problems, we chose relatively small language models (RoBERTa-base, Falcon-1B) that are widely used in practice for such tasks. Moreover, as shown in our experiment

with the EmphateticDialogues dataset, our results with RoBERTa are even better than hand-crafted prompt design for GPT-4 (Mozikov et al., 2024). Nevertheless, for more complex tasks, it may be worthwhile to consider more sophisticated LMs in the future.

Moreover, our algorithm incurs additional training costs due to the online computation of accuracy in the RL loss (1), which can be time-consuming when the validation set is large. One promising direction is to use off-policy or bandit-based RL algorithms (Li et al., 2023) to reduce the overhead of online reward computation.

Ethical Considerations

It is recognized that the development of language models entails inherent risks that require a deliberate examination of their ethical implications. The experimental framework has incorporated pretrained models, such as RoBERTa-base and falcon-rw-1b, and public datasets, including AG News, TREC, SST-2, and EmphateticDialogues. Their respective publishers have carefully processed these models and datasets, addressing potential ethical concerns. Moreover, using text classification algorithms may pose potential societal risks, none of which we feel need to be specifically highlighted here. However, ethical risks associated with deployment should be carefully analyzed: overconfidence in CRL-Prompt's high accuracy could lead to unchecked text classification outputs in high-stakes scenarios (e.g., healthcare).

Acknowledgments

The work of A. Savchenko was supported by a grant, provided by the Ministry of Economic Development of the Russian Federation in accordance with the subsidy agreement (agreement identifier 000000C313925P4G0002) and the agreement with the Ivannikov Institute for System Programming of the Russian Academy of Sciences dated June 20, 2025 No. 139-15-2025-011.

References

Younes Belkada, Sylvain Gugger, Omar Sanseviero, and 1 others. 2023. Peft: Parameter-efficient fine-tuning. https://github.com/huggingface/peft. Hugging Face.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Kaiyan Chang, Songcheng Xu, Chenglong Wang, Yingfeng Luo, Xiaoqian Liu, Tong Xiao, and Jingbo Zhu. 2024. Efficient prompting methods for large language models: A survey. *arXiv preprint arXiv:2404.01077*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*.

Xiaojun Chen, Ting Liu, Philippe Fournier-Viger, Bowen Zhang, Guodong Long, and Qin Zhang. 2024. A fine-grained self-adapting prompt learning approach for few-shot learning with pre-trained language models. *Knowledge-Based Systems*, 299:111968.

Aleksandra Edwards and Jose Camacho-Collados. 2024. Language models for text classification: Is in-context learning enough? *arXiv preprint arXiv:2403.17661*.

Qintong Li, Piji Li, Zhaochun Ren, Pengjie Ren, and Zhumin Chen. 2022. Knowledge bridging for empathetic dialogue generation. In *Proceedings of the AAAI conference on Artificial Intelligence*, volume 36, pages 10993–11001.

Xiang Lisa Li, Ping Yu, Chunting Zhou, and Timo Shen. 2023. Dialogue for prompting: Policy–gradient–based discrete prompt generation (dp2o). *arXiv preprint arXiv:2308.07272*.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING)*.

Xiao Liu, Kaixuan Ji, Yicheng Fu, and Weng Lam. 2021a. P-tuning: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2103.10385*.

Xiao Liu, Kaixuan Ji, Yicheng Fu, and Weng Lam. 2021b. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ggaliwango Marvin, Nakayiza Hellen, Daudi Jjingo, and Joyce Nakatumba-Nabende. 2023. Prompt engineering in large language models. In *International conference on data intelligence and cognitive informatics*, pages 387–402. Springer.

Mikhail Mozikov, Nikita Severin, Maria Glushanina, Mikhail Baklashkin, Andrey Savchenko, and Ilya Makarov. 2024. InsideOut: Unifying emotional llms

to foster empathy. In *ECAI 2024*, pages 4499–4502. IOS Press.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Hamza Alobeidli, Alessandro Cappelli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The RefinedWeb dataset for Falcon LLM: Outperforming curated corpora with web data only. *Advances in Neural Information Processing Systems*, 36:79155–79172.

Cheng Peng, XI Yang, Kaleb E Smith, Zehao Yu, Aokun Chen, Jiang Bian, and Yonghui Wu. 2024. Model tuning or prompt tuning? a study of large language models for clinical concept and relation extraction. *Journal of Biomedical Informatics*, 153:104630.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying lms with mixtures of soft prompts. *arXiv preprint arXiv:2104.06599*.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381.

Sahand Sabour, Chujie Zheng, and Minlie Huang. 2022. CEM: Commonsense-aware empathetic response generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11229–11237.

Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. 2024. A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*.

Andrey Savchenko and Lyudmila Savchenko. 2025. Leveraging lightweight facial models and textual modality in audio-visual emotional understanding in-the-wild. In *Proceedings of the Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 5778–5788.

Andrey V Savchenko. 2013. Phonetic words decoding software in the problem of russian speech recognition. *Automation and Remote Control*, 74(7):1225–1232.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of EMNLP*.

Marco Siino, Ilenia Tinnirello, and Marco La Cascia. 2025. From foundations to gpt in text classification: A comprehensive survey on current approaches and future trends. *Foundations and Trends® in Information Retrieval*, 19(5):557–711.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment TreeBank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1631–1642.

Claudio Spiess, Mandana Vaziri, Louis Mandel, and Martin Hirzel. 2025. AutoPDL: Automatic prompt optimization for llm agents. *arXiv preprint arXiv:2504.04365*.

William Stigall, Md Abdullah Al Hafiz Khan, Dinesh Attota, Francis Nweke, and Yong Pei. 2024. Large language models performance comparison of emotion and sentiment classification. In *Proceedings of the 2024 ACM Southeast Conference*, pages 60–68.

Xiaoxu Wang and Azreen Azman. 2025. LoRA-based efficient fine-tuning of transformer models for short text classification. In *International Visual Informatics Conference (IVIC)*, pages 168–179. Springer.

Jinta Weng, Yifan Deng, Donghao Li, Hao You, Yue Hu, and Heyan Huang. 2024. Consprompt: Exploiting contrastive samples for few-shot prompt learning. *arXiv preprint arXiv:2211.04118*.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256.

Yujia Wu and Jun Wan. 2025. A survey of text classification based on pre-trained language model. *Neurocomputing*, 616:128921.

Zehao Xiao, Shilin Yan, Jack Hong, Jiayin Cai, Xiaolong Jiang, Yao Hu, Jiayi Shen, Qi Wang, and Cees GM Snoek. 2025. DynaPrompt: Dynamic test-time prompt tuning. *arXiv preprint arXiv:2501.16404*.

Yue Yu, Simiao Zuo, Haoming Jiang, Wendi Ren, Tuo Zhao, and Chao Zhang. 2020. Fine-tuning pre-trained language model with weak supervision: A contrastive-regularized self-training approach. *arXiv preprint arXiv:2010.07835*.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Zhi Huang, Carlos Guestrin, and James Zou. 2024. Textgrad: Automatic "differentiation" via text. *arXiv preprint arXiv:2406.07496*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, and 1 others. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910*.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. 2024. Gptswarm: Language agents as optimizable graphs. *arXiv preprint arXiv:2402.16823*.

A Experimental Setup

All methods were implemented using the HuggingFace Transformers and PEFT libraries (Belkada et al., 2023). To tune the hyperparameters, we used a brute-force search on our validation set (10% of the initial training set). For our approach, we extended the standard training pipeline with a custom Trainer to support the two-phase optimization scheme described in Algorithm 1. In the first phase, we initialize the soft prompt with 20 virtual tokens per layer, following standard practice in prompt tuning literature. We optimize this prompt using cross-entropy loss with a fixed learning rate of 1×10^{-3} and a batch size 32. This setup proved sufficient to yield stable convergence in the initial prompt embedding. In the second phase, the model is refined via contrastive regularization and RL-style updates. We perform a small-scale grid search over learning rates $\{1 \times 10^{-3},\ 2 \times 10^{-3},\ 5 \times 10^{-4},\ 1 \times 10^{-4}\}$ to ensure stable performance during joint optimization. In all experiments, we use contrastive dropout at 10% and set the temperature parameter in the InfoNCE loss to $\tau = 0.1$.

To reduce computational overhead, we compute RL-based reward signals every $M = 100$ steps using a 10% hold-out subset of the training data. Although this introduces additional cost, it helps align prompt updates with the true downstream metric (classification accuracy). The loss components (5) are weighted as follows. For β, we performed a small grid search over 0.1, 0.3, 0.5. We found $\beta = 0.3$ to be a robust default across datasets. For γ, rather than relying on a single fixed value, we selected it based on the relative contribution of the RL branch: γ was chosen such that the effective magnitude of the RL loss contributes approximately 30% of the total loss during early training. This heuristic consistently stabilized training while preserving the benefits of RL feedback. Extremely large γ destabilizes optimization, while extremely small γ renders RL ineffective; our selection strategy avoids both regimes. As a result, we used $\gamma = 2 \times 10^{-5}$ for reinforcement loss in all experiments. Overall, these settings represent a trade-off between stability, efficiency, and expressiveness, and they generalize well across datasets and model architectures in our experiments.

We use the following *HandCraft* prompts:

- **AG News:** "`Read the following news: {text}. What is the category of news (World, Sport, Business, or Science)? Answer:`"
- **TREC:** "`Question: {text}. Class of question (Entity, Abbreviation, Description, Human, Location, Number):`"
- **SST–2:** "`Review: {text}. Sentiment of review (positive or negative):`"

Here, `{text}` is replaced by the input example, and the model's top-scoring output token was mapped to the corresponding label.

B Use of scientific artifacts and AI assistants

AG News dataset[2] was provided by the academic community for research purposes. TREC dataset[3] is available under CC0: Public Domain license. SST–2[4] was also released under CC0: Public Domain license. Finally, EmpatheticDialogues[5] was provided under a CC BY-NC-SA 4.0 license.

RoBERTa-base[6] is available under the MIT License, while falcon-rw-1b[7] is distributed under the Apache License 2.0. We used all the artifacts as intended by their creators. No personal information or offensive content is contained in the considered datasets.

The original text of this paper was spell- and grammar-checked and slightly smoothed out using Grammarly.

[2] https://www.kaggle.com/datasets/amananandrai/ag-news-classification-dataset

[3] https://www.kaggle.com/datasets/thedevastator/the-trec-question-classification-dataset-a-longi

[4] https://www.kaggle.com/datasets/atulanandjha/stanford-sentiment-treebank-v2-sst2

[5] https://www.kaggle.com/datasets/atharvjairath/empathetic-dialogues-facebook-ai

[6] https://huggingface.co/FacebookAI/roberta-base

[7] https://huggingface.co/tiiuae/falcon-rw-1b