Black-box Prompt Learning for Pre-trained Language Models

Shizhe Diao[♠], **Xuechun Li**[♡], **Yong Lin**[♠], **Zhichao Huang**[♠] **Tong Zhang**[♠]
[♠]The Hong Kong University of Science and Technology
{sdiaoaa, ylindf, zhuangbx, tongzhang}@ust.hk
[♡]Wuhan University
xuechun@whu.edu.cn

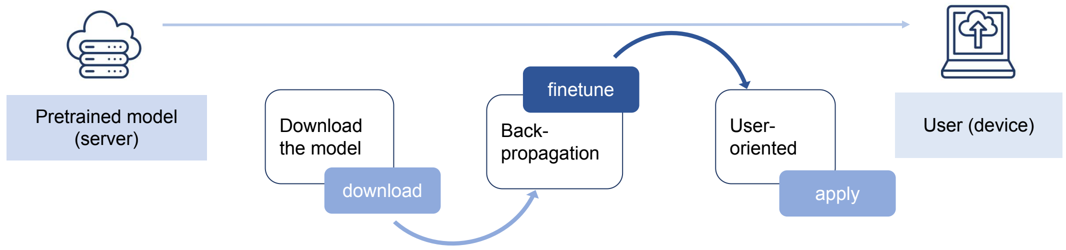
Abstract

Domain-specific fine-tuning strategies for large pre-trained models received vast attention in recent years. In previously studied settings, the model architectures and parameters are tunable or at least visible, which we refer to as *white-box* settings. This work considers a new scenario, where we do not have access to a pre-trained model, except for its outputs given inputs, and we call this problem *black-box* fine-tuning. To illustrate our approach, we first introduce the BLACK-BOX setting formally on text classification, where the pre-trained model is not only frozen but also invisible. We then propose our solution BLACK-BOX prompt, a new technique in the prompt-learning family, which can leverage the knowledge learned by pre-trained models from the pre-training corpus. Our experiments demonstrate that the proposed method achieved the state-of-the-art performance on eight datasets. Further analyses on different human-designed objectives, prompt lengths, and intuitive explanations demonstrate the robustness and flexibility of our method.

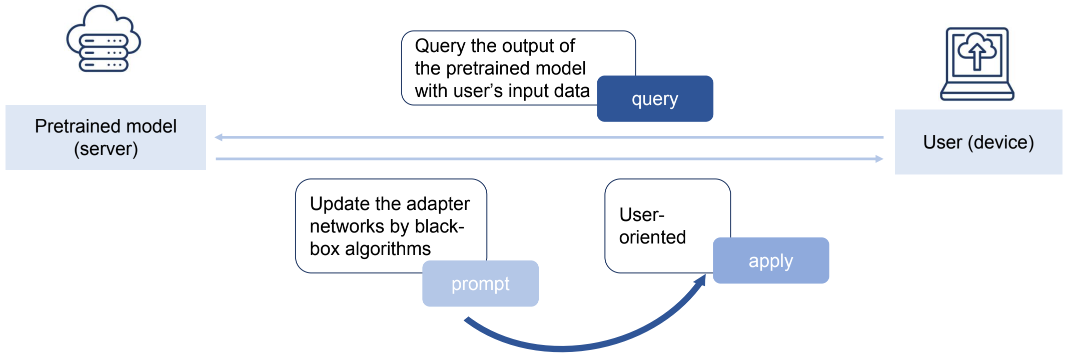
1 Introduction

Large pre-trained language models (PLMs) have achieved great success on natural language processing (NLP) and the *pre-train and fine-tune* approach has become a standard paradigm. PLMs have shown great benefits to various application scenarios across natural language understanding (NLU) (Devlin et al., 2019; Liu et al., 2019) and natural language generation (NLG) (Lewis et al., 2020; Zhang et al., 2020; Yang et al., 2020). In this paper, we focus on the text classification task, which is a well-explored but important task in NLU, aiming to identify the category of a given sentence. Previous studies were based on standard practice, i.e., given an input sentence, a pre-trained language model is fine-tuned with a labeled dataset (**fine-tune**). However, fine-tuning a large PLM requires both time and energy consumption, which is beyond the reach of many researchers. For example, GPT-3 (Brown et al., 2020) has 175 billion parameters, causing the out-of-memory issue when fine-tuning it. Even worse, we need to fine-tune a model for every different downstream task and save them on the disk, which is not only clumsy but also infeasible when we have tens of thousands of different tasks.

Therefore, a new method called prompt-based learning (Gao et al., 2021; Liu et al., 2021b; Schick and Schütze, 2021; Li and Liang, 2021; Liu et al., 2021a) is recently proposed, mitigating the above issues with several benefits. First, we only need to tune a small portion of parameters instead of the entire PLM, which is much more cost-effective. Second, the prompt-based objectives could be designed to eliminate the gap between the pre-training and downstream tasks. Third, with very few tunable parameters, it can achieve performance comparable to those of the fine-tuning methods. Most of the current prompt-based studies focused on the design of prompt, namely prompt engineering. The proposed methods are under the WHITE-BOX setting, where the pre-trained model is tunable or at least visible so that the gradient could be back-propagated to update the prompts. There are several issues with the WHITE-BOX setting. First, the parameters of a pre-trained model cannot be seen in all cases. For example, some commercial products are not available to the researchers so that we are not able to conduct fine-tuning or WHITE-BOX prompt optimization. Even if we have access to pre-trained models, it may be too large to load it (e.g., GPT-3) into the memory for researchers and organizations with limited resources. Second, in many practical application scenarios, pre-trained models are deployed in the cloud, for which gradient computations are not available. Therefore, we propose a new setting called **BLACK-BOX prompt learning**, where the pre-trained models can neither be visi-



(a) The former white-box prompt optimization


Pretrained model (server)
Query the output of the pretrained model with user's input data
query
User (device)
Update the adapter networks by black-box algorithms
prompt
User-oriented
apply

(b) The proposed black-box prompt optimization

Figure 1: The comparison between the white-box prompt-learning and the black-box prompt-learning.

ble nor adjusted. The only tunable weights are the newly added classification layer and the prompts. To solve this problem, we design a BLACK-BOX prompt learning method that can be used without accessing the parameters of pre-trained models. The experimental results on two kinds of datasets, i.e., datasets without domain-shift and datasets with domain-shift, demonstrate the effectiveness of the proposed BLACK-BOX prompt learning, where significantly improves the performance over a generic pre-trained model and outperforms all baseline models on eight datasets. The results confirm that incorporating BLACK-BOX prompt-tuning for pre-trained models is an effective and efficient solution to the PLM adaptation. We also present further analyses by investigating the effects of different objectives and prompt lengths. Our analyses demonstrated the robustness and flexibility of the proposed method.

In addition, we simulate a practical setting, where there are N domains (edge devices) and BLACK-BOX prompts are maintained for each domain. At the same time, there is one pre-trained model deployed on the cloud, which is invisible and nonadjustable. Our method is shown to be effective under such setting by achieving good performance on four selected target datasets.

The contributions of the work can be summarized as follows:

- We propose a new setting of BLACK-BOX prompt learning, where we only have access to the input and output of pre-trained models without the need to access the model parameters or model gradients.

- We propose a new BLACK-BOX prompt learning method to solve this newly proposed problem, and demonstrate its effectiveness in dealing with domain shifts on various tasks.

- The BLACK-BOX prompt is optimized without the requirements of tuning pre-trained models, saving the fine-tuning costs. Moreover, we can perform fine-tuning in a much wider range of applications than previous methods, such as when the model is only accessible via prediction APIs in typical commercial products, or model personalization in the setting of device and cloud collaboration.

2 Related Work

2.1 Prompts for Pre-trained Models

Large pre-trained language models are of great importance and a standard paradigm is pre-training a language model on a large unlabeled corpus and then fine-tuning the pre-trained model on different supervised tasks. This approach shows great improvement on lots of downstream tasks but there

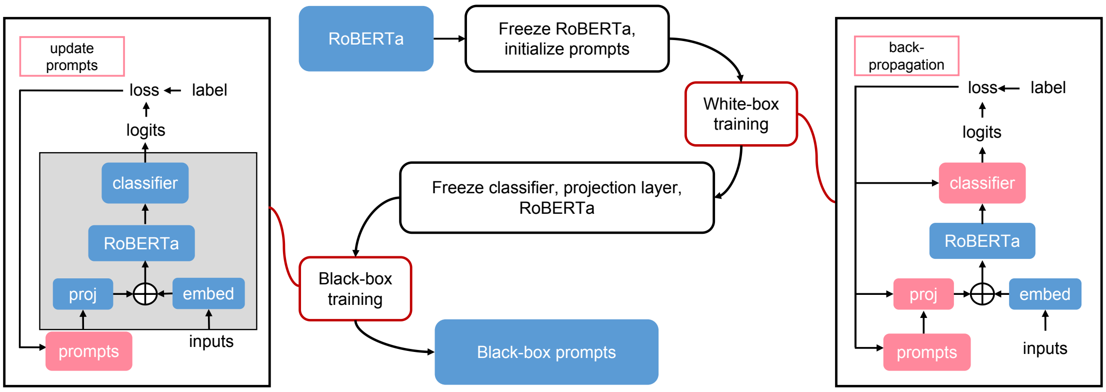


Figure 2: The overall architecture and training procedures of our model. The modules in red are tuned while the modules in blue are frozen.

are several issues with this approach: (1) fine-tuning needs to change all the parameters of the model, which causes computational costs in terms of money and time. (2) it has to fine-tune a model for different tasks and save them separately, which is clumsy and resource-intensive. Therefore, an approach that does not require tuning the large model is highly desired, which is termed as prompt-based learning. Based on the format of prompts, the prompt-based learning can be categorized into two kinds: discrete prompt (Jiang et al., 2020; Yuan et al., 2021; Haviv et al., 2021; Wallace et al., 2019; Shin et al., 2020; Gao et al., 2021; Ben-David et al., 2021; Davison et al., 2019) and continuous prompt (Zhong et al., 2021; Qin and Eisner, 2021; Hambardzumyan et al., 2021; Liu et al., 2021b; Han et al., 2021). The discrete prompt is usually a sequence of tokens or natural language phrases while the continuous prompt is designed as a sequence of vectors (embedding). However, all of these studies are limited to a WHITE-BOX setting, where it requires seeing all the parameters of a pre-trained model so that the gradients could be back-propagated. Our method, BLACK-BOX prompt learning, thus extends these studies and provides a black-box solution, which optimizes the prompts without accessing the pre-trained model.

2.2 Black-box Optimization

One of the applications of BLACK-BOX optimization is the score-based BLACK-BOX adversarial attack (Ilyas et al., 2018a,b; Huang and Zhang, 2019; Andriushchenko et al., 2020; Cheng et al., 2019; Guo et al., 2019), where the models are also invisible to the attacker. These studies use zeroth-order optimization methods such as natural evolution strategy (NES) (Wierstra et al., 2014) to optimize the input and increase the loss to fool the model. Instead of deteriorating the models' performance in the adversarial attack, our work uses NES to find better prompts and achieve higher accuracy. It is a new application of the BLACK-BOX optimization methods.

3 The Approach

We use ROBERTA as our backbone model, and the input is a sentence $X = x_1, x_2 \cdots x_i \cdots x_m$ with x_i indicating the i-th token. n prompt tokens $P = p_1, p_2, .., p_n$ are pre-pended to the input sentence to construct $[P, X]$, where the representation of each token in X is computed by ROBERTA's original embedding function. The embedding of $p_1, p_2, .., p_n$ are continuous free parameters to be learned. The whole approach is divided into two phases: the WHITE-BOX phase and the afterward BLACK-BOX phase. The overall architecture is shown in Figure 2. We first set up the WHITE-BOX phase to provide a solid initialization of parameters, and then introduce the BLACK-BOX training phase, in which by optimizing tens to hundreds of parameters in prompts, we can further improve the performance.

3.1 WHITE-BOX Optimization

In WHITE-BOX phase, we freeze ROBERTA, while the other parameters are tuned by back-propagation. As shown in Wierstra et al. (2014), most BLACK-BOX algorithms are effective when the objective function is in low dimensional space, it is not suitable to directly optimize the prompts whose dimen-

DATASET	MRPC	COLA	WNLI	RTE	CI	SE	AM	HP
TRAIN	3.7K	8.6K	635	2.5K	1.6K	3.2K	1.1K	516
DEV	408	1K	71	227	114	455	5K	64
TEST	1.7K	1K	146	3K	139	974	25K	65
CLASSES	2	2	2	2	6	7	2	2

Table 1: The statistics of eight datasets in different domains. AM, CI, SE, HP denote AMAZON, CITATIONINTENT, SCIERC, HYPERPARTISAN respectively. Notice that to limit the computational resources and balance between the eight tasks, we follow Diao et al. (2021) to randomly sample 1% AMAZON training set.

sion is as large as the hidden size of the LMs (e.g., 768). Therefore, we follow Li and Liang (2021) to initialize $p_i \in R^d$, and use a projection layer $\mathcal{F}$ to adapt it into R^D: $p_i = \mathcal{F}(p_i)$, where D is the hidden size of LMs and $d << D$. Notice that in order to limit the number of parameters within thousands, we only use the classical linear layer to implement $\mathcal{F}$. [1] After initialization and projection of prompts, we pre-pend $P = p_1, p_2, .., p_n$ and X to construct $[P, X]$, and train the model by:

$$\underset{\phi}{min}(\mathcal{L}(\mathcal{G}([P, X]), Y)) \tag{1}$$

where $\mathcal{G}$ is the main model, $\mathcal{L}$ is the loss function, X and Y are the inputs and labels, respectively. ϕ denotes all the parameters of $\mathcal{F}$, P, and the classification layer.

3.2 BLACK-BOX Optimization

When performing the BLACK-BOX training, we freeze the classifier and $\mathcal{F}$, and only further optimize the prompt initialized from WHITE-BOX training. We adopt natural evolution strategy (NES) (Wierstra et al., 2014) algorithm to accomplish BLACK-BOX training. In the BLACK-BOX phase, the gradients cannot be back-propagated to the prompts anymore, which means it is no longer possible to directly update prompts by calculating $\nabla\mathcal{L}(\mathcal{G}([p, x], y))$, where x and y are the inputs and labels respectively, p denotes the prompts. NES updates p using the following iteration:

$$\begin{cases} \mathcal{M}_i = \mathcal{L}(\mathcal{G}([w_i, x], y))\nabla_{p_t} \log \mathcal{N}(w_i|p_t, \sigma^2) \\ p_{t+1} = p_t - \eta \cdot (\frac{1}{I}\sum_{i=1}^{I} \mathcal{M}_i) \end{cases} \tag{2}$$

where η is the learning rate of prompts, I is the sample size , w_i is the samples initialized from Gaussian distribution $\mathcal{N}(p_t, \sigma^2)$. As shown in Huang and Zhang (2019), $\frac{1}{I}\sum_{i=1}^{I} \mathcal{M}_i$ provides an approximation of the gradient.

Here we introduce the detailed procedures of updating prompts with NES algorithm. Assume the input data is divided into T batches and with each batch b_t, we perform I iterations. At the t-th batch and i-th iteration, we first randomly sample the perturbation w_i from $\mathcal{N}(p_t, \sigma^2)$, and use the projection layer $\mathcal{F}$ to map it into $R^{n\times D}$. After obtaining w_i, we pre-pend it to the embedding of b_t and feed $[w_i, b_t]$ to $\mathcal{G}$, which denotes the main model composed by ROBERTA and classification layer, then we use the loss function $\mathcal{L}$ to compute $loss_i$. $\mathcal{M}_i$ is computed by $loss_i \cdot (w_i - p_t)/\sigma$. At last, the final estimated gradients are computed by averaging all the $\mathcal{M}_i$ and the prompt p_t is updated by $p_{t+1} = p_t - \eta \cdot (\frac{1}{I}\sum_{i=1}^{I} \mathcal{M}_i)$. Algorithm 1 displays the training procedure of our proposed updating method.

Algorithm 1 The framework of BLACK-BOX training using NES Algorithm.

```
Input: Input batch $b_t$, Label batch $y_t$, prompt $p_t$, projection layer $\mathcal{F}$, main model $\mathcal{G}$, loss function $\mathcal{L}$
for $i \leq I$ do
    Sample $w_i \sim \mathcal{N}(p_t, \sigma^2)$
    $w_i = \mathcal{F}(w_i)$
    $loss_i = \mathcal{L}(\mathcal{G}([w_i, b_t], y_t))$
    $\mathcal{M}_i = loss_i \cdot (w_i - p_t)/\sigma$
$p_{t+1} = p_t - \eta \cdot (\frac{1}{I}\sum_{i=1}^{I} \mathcal{M}_i)$
return $p_{t+1}$
```

4 Experimental Settings

In this section, we first introduce the datasets (§4.1), followed by the baseline models (§4.2) and evaluation metrics (§4.3). Lastly we describe the implementation details (§4.4).

4.1 Datasets

In order to explore the model's ability in regular classification tasks as well as domain-specific classification tasks, we include four datasets from the GLUE benchmark (Wang et al., 2018) and four

[1] Our trial experiments show that other implementations such as MLP makes little difference.

Dataset	Mrpc	CoLA	Wnli	Rte	CI	SE	AM	HP
RoBERTa	82.93	30.96	56.34	59.57	30.17	35.77	57.37	81.99
RoBERTa + WP	85.22	49.93	57.75	63.18	49.88	49.73	61.49	90.25
RoBERTa + WP.P	86.29	54.02	63.38	64.62	51.18	49.68	62.01	88.54
RoBERTa + WP.P + black-box	**86.81**	**55.46**	**66.20**	**68.75**	**54.47**	**51.07**	**62.65**	**92.04**
black-box ↑	0.52	1.44	2.82	4.13	3.29	1.39	0.64	1.79

Table 2: The overall performance of black-box prompt and the comparison on eight tasks, where WP and WP.P stand for white-box prompt tuning and white-box prompt tuning with projection, respectively. The evaluations are performed on the development sets for Mrpc, CoLA, Wnli and Rte and on the test sets for the remaining. black-box ↑ denotes the uplift against the best result achieved by the baseline models.

datasets from the specific domains including computer science, reviews and news following Gururangan et al. (2020); Diao et al. (2021). The statistics of these datasets are shown in Table 1.

4.2 Baselines

In our experiments, we use the following three models as the baselines.

- **RoBERTa**: an off-the-shelf RoBERTa-base model with frozen weights. Only the newly added classification layer (classifier) is updated for downstream tasks.
- **RoBERTa + white-box Prompt** (WP): an off-the-shelf RoBERTa-base model with frozen weights. Only the newly added classification layer (classifier) and prompts are updated for downstream tasks. We follow the method PromptTuning Lester et al. (2021) where n prompt tokens $p_1, p_2, .., p_n$ are pre-pended to the input, and the embedding of $p_1, p_2, .., p_n$ are learned.
- **RoBERTa + white-box Prompt + Projection** (WP.P): an off-the-shelf RoBERTa-base model with frozen weights. The embedding of p_i are initialized $\in R^d$, and a projection function, i.e. a linear layer or MLP, is used to project p_i into R^D. The embedding of prompts $p_1, p_2, .., p_n$, the parameters of the projection function and classifier are learned.

4.3 Evaluation Metrics

For the tasks from the GLUE benchmark, we adopt Matthews Correlation Coefficient for CoLA, F1-score for Mrpc, and accuracy for Rte and Wnli following their original metric choices. We adopt macro-F1 for Amazon, CitationIntent, SciERC, HyperPartisan as evaluation metrics following Diao et al. (2021).

4.4 Implementation

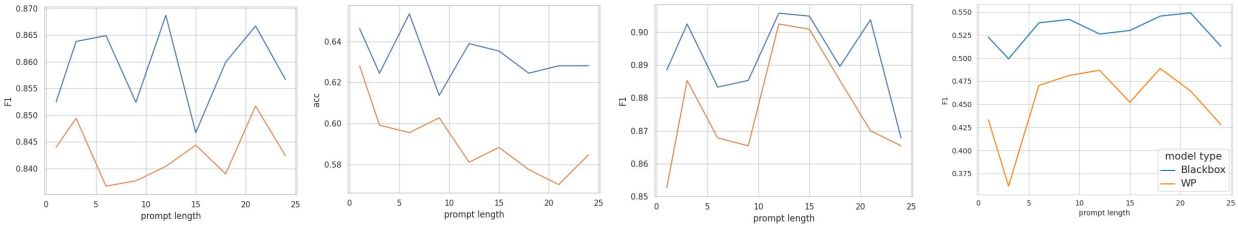
For all experiments, we implement RoBERTa-base architecture and initialize it with pre-trained weights by Huggingface's Transformers library[2]. The batch size of training and evaluation is set to 16 and 32, respectively. We train both the baseline models and white-box phase of our model for 30 epochs with learning rate 5×10^{-4}. The adopted optimizer is AdamW (Loshchilov and Hutter, 2019). For the RoBERTa+WP baseline, we follow the implementation of Lester et al. (2021), and the length of prompts is 6. For white-box phase, we randomly initialize the Prompt, whose dimension is in $\{4, 8, 16, 32\}$ and then project it into a 768-d vector with a linear layer. The prompts have the same dimension with the hidden size of RoBERTa. For the black-box phase, we reload the saved model which achieves the highest score on development set in white-box phase, and train it by only optimizing the Prompts. The other parameters contained in our model can be found in the Appendix.

5 Experimental Results

5.1 Overall Performance

We compare the RoBERTa model with prompt-based RoBERTa, under both white-box and black-box settings. The overall results on eight datasets are reported in Table 2. First, the models with prompt tuning outperform that without it, demonstrating that prompt tuning is effective on all eight tasks. In detail, two white-box prompt learning models, RoBERTa+WP and RoBERTa+WP.P achieve an average of 9.04% and 10.57% improvement across eight datasets, respectively. This observation is consistant with the previous studies on prompt learning (Li and

[2] https://github.com/huggingface/transformers



(a) MRPC (b) RTE (c) HP (d) CI

Figure 3: The effects of prompt length.

DATASET	SST2	IMDB	CR	MR	MPQA
TRAIN	67K	2K	1.8K	8.6K	8.6K
DEV	873	5K	500	500	500
TEST	1.8K	25K	2K	2K	2K

Table 3: The statistics of five sentiment analysis datasets.

Liang, 2021; Schick and Schütze, 2020; Liu et al., 2021b). Second, compared with the model with WHITE-BOX optimization, BLACK-BOX optimization brings further gains based on it. The BLACK-BOX optimization helps to improve the performance by approximately 2.22% on average, which shows that the BLACK-BOX optimization plays a synergistic role with the WHITE-BOX. Across eight tasks, it is observed that the BLACK-BOX optimization on datasets with domain-shift is as effective as the datasets in general domain. While it is known that domain shift is more difficult for models to deal with, BLACK-BOX optimization offers an efficient solution to domain-specific datasets. We provide two main reasons for the further uplift in performance brought by BLACK-BOX training. First, the WHITE-BOX training is usually not sufficient due to the time and resource limitation. Second, it is technically impossible to find the best set of hyper-parameters for each dataset. Therefore, the actual performance of WHITE-BOX training could be lower than the theoretical assumption, leaving future optimization space for BLACK-BOX training.

In summary, our proposed two-stage optimization (i.e., WHITE-BOX and then BLACK-BOX) is an effective and efficient solution on tuning large pretrained models. Compared with the ROBERTA, the final model (WHITE-BOX + BLACK-BOX) brings an average improvement of performance by around 12.79%, illustrating the effectiveness on both general datasets and domain-specific datasets.

DATASET	IMDB	CR	MR	MPQA
ROBERTA	82.43	85.58	82.27	81.34
ROBERTA+WP.P	86.54	86.74	87.35	80.40
+ BLACK-BOX	**87.02**	**88.29**	**87.75**	**85.25**

Table 4: The performance of BLACK-BOX training on transfer learning. ROBERTA denotes training the classifier on each target task's own training data, and WP.P refers to WHITE-BOX training on SST-2 and directly evaluating on each target task's test set.

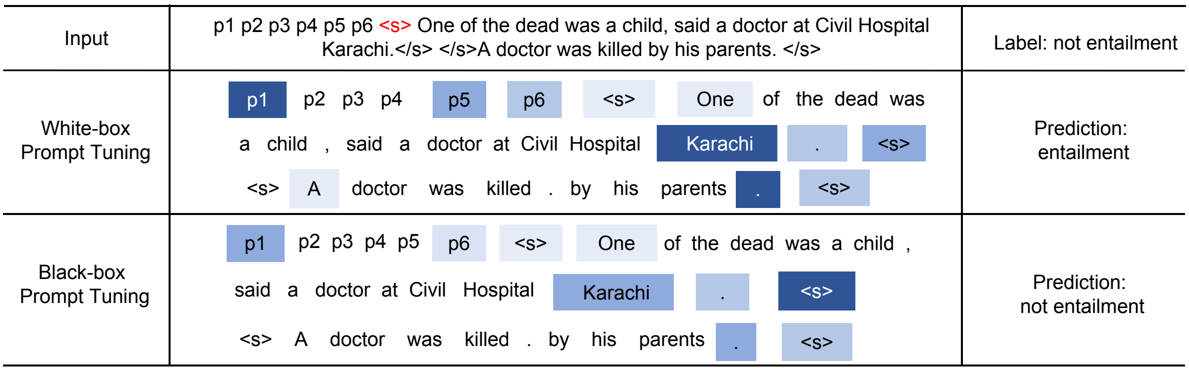
DATASET	MRPC	RTE	CI	HP
ROBERTA+WP.P	84.44	65.34	54.02	87.00
+ BLACK-BOX-ce	84.63	66.43	**54.89**	88.54
+ BLACK-BOX-hinge	**84.96**	**66.79**	54.57	**90.09**
Cross-entropy ↑	0.19	1.09	0.87	1.54
Hinge ↑	0.52	1.45	0.55	3.09

Table 5: The performance of BLACK-BOX prompt with hinge loss and cross-entropy loss on four selected tasks. -ce and -hinge represent using cross-entropy loss and hinge loss, respectively.

5.2 Performance in Transfer Learning

In this section, we conduct experiments on four sentiment analysis datasets (i.e., IMDB, CR, MR, MPQA) to verify the ability of BLACK-BOX training in transfer learning. First, we use SST-2 as the source dataset following Vu et al. (2021) and perform WHITE-BOX training on it. After the WHITE-BOX phase, we freeze the main model and perform BLACK-BOX training on each target task to update the prompts only. Following Wang et al. (2021), for CR, MR and MPQA, we randomly sample 2,000 instances as the test set and use the rest as training set. For IMDB, to reduce the size of training data, we follow Diao et al. (2021) to randomly sample 10% of the training set. We conduct experiments on two baseline models. The first one is training the classification layer on each target task's train set directly, and the second baseline is training classification layer and prompts on source dataset then testing on each target task's test set.

The results are shown in Table 4. It is ob-



Input	p1 p2 p3 p4 p5 p6 <s> One of the dead was a child, said a doctor at Civil Hospital Karachi.</s> </s>A doctor was killed by his parents. </s>	Label: not entailment
White-box Prompt Tuning	p1 p2 p3 p4 p5 p6 <s> One of the dead was a child , said a doctor at Civil Hospital Karachi . <s> <s> A doctor was killed . by his parents . <s>	Prediction: entailment
Black-box Prompt Tuning	p1 p2 p3 p4 p5 p6 <s> One of the dead was a child , said a doctor at Civil Hospital Karachi . <s> <s> A doctor was killed . by his parents . <s>	Prediction: not entailment

Figure 4: The attention distribution of the token <s> in RTE.

served that BLACK-BOX training outperforms both ROBERTA and ROBERTA+WP.P by a large margin, demonstrating that our BLACK-BOX method is robust under transfer learning settings. The experimental results display the expansion potential of our approach to server-device deployment. In application scenarios, large pre-trained models are too large to save on the device (e.g., phones), so they are deployed on the cloud. To adapt the model for user's habit, we need a tiny model deployed on the device and update it with gradients from the cloud. With our proposed BLACK-BOX prompt, we save the cost of transmitting gradients between cloud and device and still achieve a great performance. This is a promising application approach practically, especially when there are N edge devices (domains) and one large, invisible and nonadjustable pre-trained model deployed on the cloud. We can simply maintain and update the BLACK-BOX prompt for each device.

6 Analysis

6.1 Effects of Different Loss Types

In our previous experiments, we apply NES algorithm with cross-entropy (CE) loss to optimize the prompt. In this section, we explore the performance of our model with different loss types and show that our our BLACK-BOX training are flexible enough to work with human-designed objectives. We conducted further experiments with hinge loss on four datasets: MRPC, COLA, CI, SE, selected from the eight datasets and find that other loss can achieve comparable results with cross-entropy loss. The results are shown in Table 5. It is observed that cross-entropy and hinge loss bring an average of 0.92% and 1.40% improvements over ROBERTA-WP.P, respectively. Therefore, we conclude that, with hinge loss, our approach could work as well as that with cross-entropy loss and we hope to extrapolate to any other kinds of human-designed objectives.

DATASET	RTE	WNLI	CI	HP
WP	1.13	5.59	1.69	18.82
WP.P	38.33	7.35	**3.42**	21.49
+ BLACK-BOX	**40.52**	**8.59**	3.35	**23.01**

Table 6: The cosine similarity of prompts and frequent entities in four datasets.

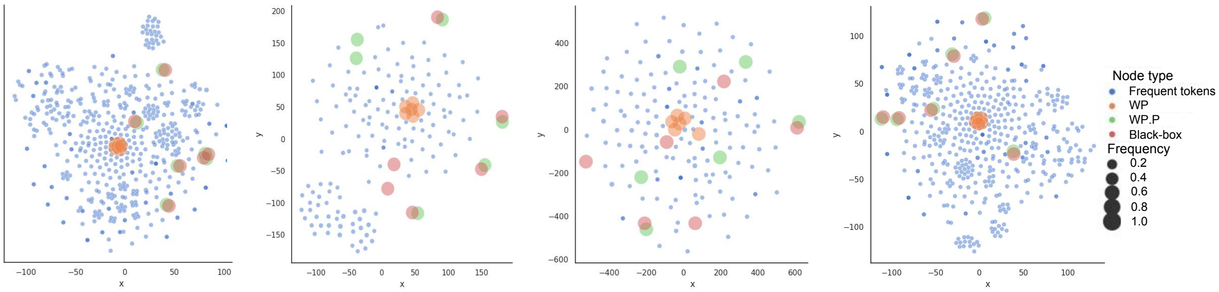
6.2 Effects of Prompt Length

It is well known that prompt-based methods are sensitive to the many aspects of prompts including contexts (Jiang et al., 2020; Shin et al., 2020), orders (Lu et al., 2021) and length (Lester et al., 2021), and inappropriate prompts will cause bad performance. In this section, we study the effects of different prompt length $\in \{1, 3, 6, 9, 12, 15, 18, 21, 24\}$ on four selected datasets. As for WHITE-BOX tuning, with the increase of prompt length, the performance is increased and then decreased on MRPC, HP and CI, while on RTE, the performance drops. As for the BLACK-BOX tuning, similar phenomenon is observed except that the performance of RTE is increased and then decreased as well.

6.3 Visualization of Attention Maps

In this section, we visualize the attention map of the token $< s >$ at the beginning of inputs $X = x_1, x_2 \cdots x_i \cdots x_T$, which is past into the classifier to get the sentence classification results. We choose the dataset RTE since the sentences contained in this dataset are easy for human interpretation. The RTE task is to classify each input, which contains two sentences separated with tap, as entailment or not entailment.

As shown in Figure 4, in the results of BLACK-



(a) RTE (b) WNLI (c) CI (d) HP

Figure 5: The tSNE visualization of prompts and the top k entities. Given that the size and number of entities varies, we choose $k = 500, 155, 150, 500$ for RTE, WNLI, CI, HP respectively. The prompt length is fixed as 6.

BOX model, the first token $< s >$ attributes more to the $< s >$ token in the middle of the inputs, which indicates the end to one sentence thus might referring to less coherence between the two sentences, thus the prediction is not entailment. While in the WP results, the first token $< s >$ attributes more to the first prompt p_1 and the last period, which indicates the model tends to consider the two sentences as a whole, resulting to the prediction: entailment. More visualized examples are not shown here due to space limitation. However, based on considerable empirical evidence, we conclude that BLACK-BOX prompts capture more reasonable information to guide the prediction.

6.4 Prompt Explanation

In this part, we explore the explanation of different prompting methods. We first use spaCy [3] to extract named entities from the original dataset, and select top k entities by ranking their frequency. Then we compute the similarity of prompts and the top k entities by: $Similarity = \sum_{i=1}^{n} \sum_{j=1}^{k} \mathcal{S}(p_i, e_j)$, where n is the length of prompts, $\mathcal{S}$ is cosine similarity function, and e_j is the ROBERTA embedding of the first token in an entity. As shown in Table 6, in all datasets, BLACK-BOX prompts are more similar to the top k entities than WP, showing that more accurate prediction scores could be related to prompts' similarity to the dataset.

Moreover, we use the tSNE method (Haviv et al., 2021) to visualize $p_1, ..., p_n$ and $e_1, ..., e_k$. As shown in Figure 5, WP tends to result in prompts constructed by similar p_i while the BLACK-BOX training generates more separated p_i. For instance, for the CI dataset, the embeddings of $p_1, ..., p_6$ gather in the center of the whole embedding space, while both WP.P and BLACK-BOX generate more separated prompts, scattering in the marginal of the embedding space. We conclude that the concentrated prompts tend to encode homogeneous information from the dataset during training, overlooking the variety. Besides, such phenomenon appears in all four datasets regardless of their size and entity number, showing that BLACK-BOX training encodes more diversity in prompts, leading to more accurate prediction.

7 Conclusion

This paper proposes a novel setting for text categorization namely BLACK-BOX prompt learning, where a large pre-trained model is invisible so that the gradients can not be back-propagated to update the prompts. Compared with the standard pretrain then fine-tune paradigm, our approach only requires updating very few parameters. Compared with the previous prompt-based methods, our approach does not require the visibility of pre-trained models, and thus it provides more flexibility in practical applications. We propose a BLACK-BOX prompt learning method which employs the NES algorithm to approximate the gradients, and then update the prompts. Experimental results demonstrate that our approach brings large gains over the base method without prompt learning. Compared with the WHITE-BOX prompt-based methods, our approach achieves further improvements, illustrating the effectiveness of BLACK-BOX optimization. Experiments on the transfer learning show the potential of our approach in the realistic scenarios, where the pre-trained model is deployed on the cloud, and the prompt learning can be implemented on each device. Moreover, our approach does not require back-propagating the gradients, and thus saves computation and communication costs.

[3] https://spacy.io/

References

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. 2020. Square Attack: a Query-efficient Black-box Adversarial Attack via Random Search.

Eyal Ben-David, Nadav Oved, and Roi Reichart. 2021. PADA: A Prompt-based Autoregressive Approach for Adaptation to Unseen Domains. *arXiv preprint arXiv:2102.12206*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Shuyu Cheng, Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. 2019. Improving Black-box Adversarial Attacks with a Transfer-based Prior. *Advances in Neural Information Processing Systems*, 32:10934–10944.

Joe Davison, Joshua Feldman, and Alexander M Rush. 2019. Commonsense Knowledge Mining From Pretrained Models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1173–1178.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Shizhe Diao, Ruijia Xu, Hongjin Su, Yilei Jiang, Yan Song, and Tong Zhang. 2021. Taming Pre-trained Language Models with N-gram Representations for Low-Resource Domain Adaptation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3336–3349.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. 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)*, pages 3816–3830. Association for Computational Linguistics.

Chuan Guo, Jacob Gardner, Yurong You, Andrew Gordon Wilson, and Kilian Weinberger. 2019. Simple Black-box Adversarial Attacks. In *International Conference on Machine Learning*, pages 2484–2493. PMLR.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360. Association for Computational Linguistics.

Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. 2021. WARP: Word-level Adversarial ReProgramming. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4921–4933. Association for Computational Linguistics.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. PTR: Prompt Tuning with Rules for Text Classification. *arXiv preprint arXiv:2105.11259*.

Adi Haviv, Jonathan Berant, and Amir Globerson. 2021. BERTese: Learning to speak to BERT. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3618–3623, Online. Association for Computational Linguistics.

Zhichao Huang and Tong Zhang. 2019. Black-box Adversarial Attack with Transferable Model-based Embedding. *arXiv preprint arXiv:1911.07140*.

Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. 2018a. Black-box Adversarial Attacks with Limited Queries and Information. In *International Conference on Machine Learning*, pages 2137–2146. PMLR.

Andrew Ilyas, Logan Engstrom, and Aleksander Madry. 2018b. Prior Convictions: Black-box Adversarial Attacks with Bandits and Priors. In *International Conference on Learning Representations*.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How Can We Know What Language Models Know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer.

2020. BART: Denoising Sequence-to-Sequence Pretraining for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pretrain, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *arXiv preprint arXiv:2107.13586*.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. GPT Understands, Too. *arXiv preprint arXiv:2103.10385*.

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*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021. Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity. *arXiv preprint arXiv:2104.08786*.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying LMs with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5203–5212, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2020. Exploiting Cloze Questions for Few Shot Text Classification and Natural Language Inference. *arXiv preprint arXiv:2001.07676*.

Timo Schick and Hinrich Schütze. 2021. It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. *arXiv preprint arXiv:2010.15980*.

Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou, and Daniel Cer. 2021. Spot: Better Frozen Model Adaptation Through Soft Prompt Transfer. *arXiv preprint arXiv:2110.07904*.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2153–2162, Hong Kong, China. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

Sinong Wang, Han Fang, Madian Khabsa, Hanzi Mao, and Hao Ma. 2021. Entailment as Few-Shot Learner. *arXiv preprint arXiv:2104.14690*.

Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber. 2014. Natural Evolution Strategies. *The Journal of Machine Learning Research*, 15(1):949–980.

Ze Yang, Wei Wu, Can Xu, Xinnian Liang, Jiaqi Bai, Liran Wang, Wei Wang, and Zhoujun Li. 2020. StyleDGPT: Stylized Response Generation with Pretrained Language Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 1548–1559.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. BARTScore: Evaluating Generated Text as Text Generation. *arXiv preprint arXiv:2106.11520*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and William B Dolan. 2020. DIALOGPT: Large-Scale Generative Pre-training for Conversational Response Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278.

Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. Factual Probing is [mask]: Learning vs. Learning to Recall. *arXiv preprint arXiv:2104.05240*.