Is Continuous Prompt a Combination of Discrete Prompts? Towards a Novel View for Interpreting Continuous Prompts

Tianjie Ju, Yubin Zheng, Hanyi Wang, Haodong Zhao, Gongshen Liu*
School of Electronic Information and Electrical Engineering
Shanghai Jiao Tong University
{jometeorie, zybhk21, why_820, zhaohaodong, lgshen}@sjtu.edu.cn

Abstract

The broad adoption of continuous prompts has brought state-of-the-art results on a diverse array of downstream natural language processing (NLP) tasks. Nonetheless, little attention has been paid to the interpretability and transferability of continuous prompts. Faced with the challenges, we investigate the feasibility of interpreting continuous prompts as the weighting of discrete prompts by jointly optimizing prompt fidelity and downstream fidelity. Our experiments show that: (1) one can always find a combination of discrete prompts as the replacement of continuous prompts that performs well on downstream tasks; (2) our interpretable framework faithfully reflects the reasoning process of source prompts; (3) our interpretations provide effective readability and plausibility, which is helpful to understand the decision-making of continuous prompts and discover potential shortcuts. Moreover, through the bridge constructed between continuous prompts and discrete prompts using our interpretations, it is promising to implement the cross-model transfer of continuous prompts without extra training signals. We hope this work will lead to a novel perspective on the interpretations of continuous prompts.

1 Introduction

Continuous prompts for pre-trained language models (PLMs) have shown remarkable performance on almost every NLP field (Li and Liang, 2021; Lester et al., 2021; Liu et al., 2021b). However, trained continuous prompts tend to improve performance at the sacrifice of interpretability and transferability relative to discrete prompts (Liu et al., 2021a), which causes mistrust in people and makes cross-model transfer challenging.

Recent advancements spiked interest in understanding how prompts work and found the counter-intuitive mechanism behind. (Webson and Pavlick, 2022) conducted numerous experiments on various discrete prompts, finding the improvement in downstream tasks does not originate from the model understanding task instructions in a manner similar to how humans use them. (Kavumba et al., 2022) presented the first investigation of the exploitation of superficial cues by prompt-based models, finding the presence of superficial cues which prompt-based models exploit. Continuous prompts, on the other hand, are more complicated and incomprehensible. Recent attempts for interpreting continuous prompts came from (Khashabi et al., 2022), which introduced the *Prompt Waywardness Hypothesis* to prove the infeasibility of interpreting a learned continuous prompt with a single discrete prompt. To the best of our knowledge, no general post-hoc interpretable framework is proposed to translate continuous prompts into a comprehensible form.

Towards filling this research gap, we propose the *Combination Hypothesis*, which argues the feasibility of utilizing combinations of discrete prompts

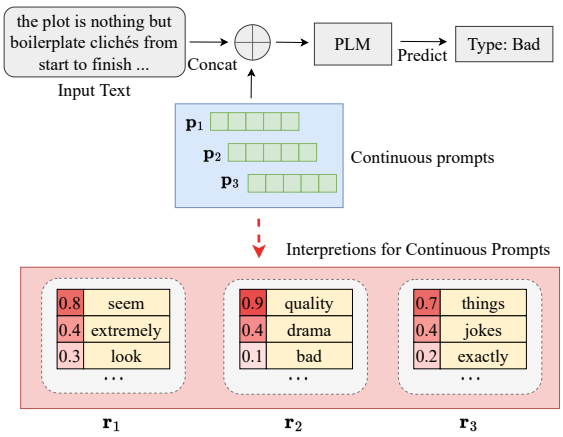

Figure 1: Interpreting continuous prompts for sentiment classification. Each continuous prompt ($\mathbf{p}_i$) can be regarded as the combination of discrete prompts ($\mathbf{r}_i$), which reflects the tokens utilized by continuous prompts in prompting the PLM to output expected labels.

*Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2023, pages 7804–7819
July 9-14, 2023

as faithful interpretations for continuous prompts (§3.2). In other words, we treat the continuous prompt as an embedding lookup table with the one-hot restriction removed. For instance, a well-trained continuous prompt for sentiment classification should contain task-related tokens such as "drama" or auxiliary tokens such as "seem", "look" to stimulate the PLM for desired outputs (Fig. 1). To find the effective interpretation, a joint optimization framework is proposed to ensure both prompt fidelity and downstream fidelity (§3.3).

Comprehensive experiments are conducted to support our hypothesis and framework. We first directly optimize parameters of the combination of discrete prompts to replace continuous prompts. Results show that the combination of discrete prompts has competitive performance in most scenarios (especially in few-shot learning), which verifies the feasibility of the *Combination Hypothesis* in practice (§5).

As a significant property of interpretations, faithfulness is comprehensively verified to check *how accurately it reflects the true reasoning process of the model* (Jacovi and Goldberg, 2020). We first verify the prompt fidelity and downstream fidelity of the interpretations using discrete prompts and continuous prompts as the content to be interpreted (§6.1), then we verify the tokens selected from interpretations can better restore the performance of source prompts on downstream tasks (§6.2).

Despite faithfulness, a high-quality interpretation should also contain plausibility, which refers to *how convincing the interpretation to humans* (Jacovi and Goldberg, 2020). By conducting a visual comparison with the nearest tokens to continuous prompts (§7.1), Our interpretations are shown to be more convincing and allow us to identify several "shortcuts" contained in the model's decision-making (§7.2).

Furthermore, inspired by the readability and transferability of discrete prompts, we investigate the feasibility of cross-model transfer for continuous prompts using our interpretations. We argue its breakthrough since no previous work to achieve cross-model transfer for continuous prompts without any training signals on target PLMs. Experiments show that even continuous prompts trained on a simple structured PLM with 100-shot settings can be transferred to large PLMs using our method and achieve competitive performance (§8).

2 Related Work

Prompt Engineering. Prompt engineering, as a crucial part of prompt learning, is the process of creating a prompt function that performs effectively on the downstream task (Liu et al., 2021a). It can be generally divided into discrete prompts and continuous prompts.

Discrete prompts usually search for templates, i.e., natural language tokens in discrete spaces as prompt functions. There is a line of work focused on manually-designed prompts (Petroni et al., 2019; Brown et al., 2020; Scao and Rush, 2021). These methods rely excessively on prior knowledge, while even experts have difficulty finding optimal templates (Jiang et al., 2020). Therefore, recent explorations devoted much attention to automatically searching for templates in discrete spaces (Jiang et al., 2020; Shin et al., 2020; Gao et al., 2021; Haviv et al., 2021).

Continuous prompts, on the other hand, relax the constraint that templates are natural language tokens (Li and Liang, 2021; Liu et al., 2021b; Lester et al., 2021; Zhong et al., 2021; Qin and Eisner, 2021; Zhang et al., 2022). These works effectively improve performance at the expense of interpretability. Khashabi et al. (2022) demonstrated the disconnection between continuous prompts and discrete prompts, In this paper, we investigate the feasibility of using discrete prompts to interpret continuous prompts from a novel view.

Cross-model Transfer. Benefiting from the readability of discrete prompts, we can easily transfer manually-designed prompts to any PLM (Perez et al., 2021). Nonetheless, since the embedding dimensions and semantic spaces of different PLMs are inconsistent, it is tricky for cross-model transfer of continuous prompts. Su et al. (2022) devoted the first attempt by *prompt projectors*, which trained on another task to project continuous prompts into the semantic space of target PLMs. As a post-hoc interpretable framework, this paper investigates the feasibility of cross-model transfer without the help of additional task data.

3 Prompt Decoupling

3.1 Setup and Formulation

Given a sequence with n continuous prompts $\mathcal{P} = \{\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_n\}$ trained on the dataset $\mathcal{D} = \{x, y\}$, we analyze the feasibility to interpret continuous prompts as a combination of discrete

prompts $\mathcal{R} = \{\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_n\}$, where $\mathbf{p}_i \in \mathbb{R}^d$ is a d-dimensional vector, $\mathbf{r}_i \in \mathbb{R}^v$ is a v-dimensional vector which decouples $\mathbf{p}_i$ into v discrete prompts (Fig. 1).

In this paper, we are interested in generating an interpretation $\mathcal{R}$ with both faithfulness and plausibility (Jacovi and Goldberg, 2020). In addition, as a side effect of the interpretation, it is also expected to utilize the results for cross-model transfer of continuous prompts.

3.2 The Combination Hypothesis

Continuous prompts are essentially trained on a large corpus of natural language. These incomprehensible prompts occupy the place of discrete prompts that are composed of natural language tokens, but better motivate the PLM to output desired results. Consequently, they are intuitively more likely to be associated with natural language tokens than to be isolated from them.

Considering the infeasibility of one-to-one mapping (Khashabi et al., 2022), we propose the idea that the continuous prompt may be a combination of multiple discrete prompts. It is known that the essence of discrete prompt $e(x)$ is a function of token x, which is parameterized by a one-hot embedding lookup table (Li et al., 2020a). If the one-hot restriction is removed, the continuous prompt can be seen as the output of a fully connected layer with all discrete prompts as input. We formalize the idea as the following hypothesis.

Hypothesis 1: *(Combination Hypothesis) For any continuous prompt* $\mathbf{p} \in \mathbb{R}^d$ *and a discrete prompt matrix* $\mathbf{E} \in \mathbb{R}^{v \times d}$ *of a large pre-trained model, there exists a vector* $\mathbf{r} \in \mathbb{R}^v$ *such that* $dist(\mathbf{r}^\top \mathbf{E}, \mathbf{p}^\top) \leq \Delta$*, where* $dist(\cdot)$ *is the Euclidean distance function,* Δ *is the shortest distance to* $\mathbf{p}$ *among all discrete prompts.*

In fact, it can almost be proved that the linear equation $\mathbf{r}^\top \mathbf{E} = \mathbf{p}^\top$ has infinitely many solutions. For general PLMs, it is always satisfied that $v \gg d$ (e.g., $v = 30522, d = 768$ in the $\text{BERT}_{\text{base}}$ model (Devlin et al., 2019)). Thus, for most cases, $R(\mathbf{E}^\top) = R(\mathbf{E}^\top, \mathbf{p}) < v$, where R denotes the rank of the matrix.

Nonetheless, although $v \gg d$, it is still not guaranteed that these discrete prompts can necessarily constitute a set of bases in the vector space, which implies the non-existence of an exact solution. Thus, we relax the restriction in our hypothesis, which only proves the existence of a more faithful interpretation than the nearest discrete prompt. We consider the following two cases.

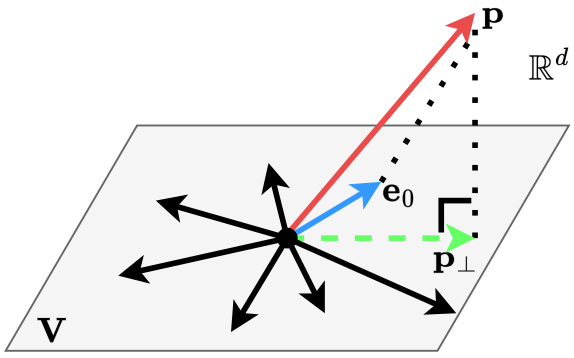

Figure 2: The case where discrete prompts fail to form a set of bases of the space and the continuous prompt $\mathbf{p}$ (red) is not in the linear subspace $\mathbf{V}$ they form. We can still find a linear combination of discrete prompts $\mathbf{p}_\perp$ (green) such that its distance to $\mathbf{p}$ is not greater than the distance from the nearest discrete prompt $\mathbf{e}_0$ (blue) to $\mathbf{p}$.

1. E constitutes a set of bases in the vector space. In this case, all vectors in the vector space can be represented by this set of bases. Therefore, there exists a solution $\mathbf{r}$ such that

$$\text{dist}(\mathbf{r}^\top \mathbf{E}, \mathbf{p}^\top) = 0 \leq \Delta. \tag{1}$$

2. E is not sufficient to constitute a set of bases in the vector space. Let $\mathbf{e}_0$ be the nearest discrete prompt to $\mathbf{p}$, V be the linear subspace composed of $\mathbf{E}$. If $\mathbf{p} \in \mathbf{V}$, then there exists a linear combination of discrete prompts that satisfies Eq.1. If $\mathbf{p} \notin \mathbf{V}$ (Fig. 2), we make a projection of $\mathbf{p}$ onto $\mathbf{V}$, denoted $\mathbf{p}_\perp$, then

$$\text{dist}(\mathbf{p}_\perp^\top, \mathbf{p}^\top) \leq \text{dist}(\mathbf{e}_0^\top, \mathbf{p}^\top) = \Delta. \tag{2}$$

Since $\mathbf{p}_\perp$ is in the linear subspace $\mathbf{V}$, it can be represented as a linear combination of discrete prompts. Therefore, in this case, the hypothesis also holds, which implies the existence of a more faithful interpretation than the discrete prompt.

Empirically, simply summing rather than concatenating prompts does not seem to make sense. Suppose we have two input vectors and their concatenation, denoted as $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^d$ and $\mathbf{x}_{\text{concat}} = [\mathbf{x}_1^\top \oplus \mathbf{x}_2^\top]^\top \in \mathbb{R}^{2d}$. Then we apply linear embedding projection e to x_{concat}:

$$\begin{aligned} e(\mathbf{x}_{\text{concat}}) &= \mathbf{W}\mathbf{x}_{\text{concat}} \\ &= [\mathbf{W}_1 \oplus \mathbf{W}_2] \cdot [\mathbf{x}_1^\top \oplus \mathbf{x}_2^\top]^\top \\ &= \mathbf{W}_1\mathbf{x}_1 + \mathbf{W}_2\mathbf{x}_2 \\ &= e(\mathbf{x}_1) + e(\mathbf{x}_2), \end{aligned} \tag{3}$$

where $\mathbf{W}_1 \in \mathbb{R}^{d\times d}, \mathbf{W}_2 \in \mathbb{R}^{d\times d}, \mathbf{W} \in \mathbb{R}^{d\times 2d}$ are parameters of the linear projection. This indicates that summing is somehow equivalent to concatenating, which also supports the rationality of decoupling continuous prompts into discrete prompts.

3.3 Finding Interpretations

The hypothesis indicates the existence of $\mathcal{R}$, but it does not consider how to find a solution that better represents the continuous prompt. In this section, we first introduce an optimization method to find the interpretations that both satisfy the hypothesis and ensure downstream fidelity, then we reduce the vocabulary size by traversing datasets and thus speed up the optimization.

Our post-hoc interpretable framework is similar to probes, which focus on simple linguistic properties of interest (Conneau et al., 2018). Therefore, following the view of Hewitt and Liang (2019), a simple model with only one linear layer is designed in our paper for interpreting continuous prompts. Since negative results can be confusing or controversial, the softplus activation function (Dugas et al., 2000) is applied in the output layer.

To satisfy the *Combination Hypothesis*, we minimize the distance between continuous prompts and the combination of discrete prompts:

$$\ell_1(\mathbf{r}; \mathbf{E}, \mathbf{p}) = \text{dist}(\mathbf{r}^\top \mathbf{E}, \mathbf{p}^\top). \tag{4}$$

It is not sufficient to find the most reasonable solution with the loss above. As a consequence, we introduce the following loss function to ensure downstream fidelity:

$$\ell_2(\mathbf{r}; \mathbf{E}, \mathbf{p}, \mathcal{D}) = \mathbb{E}_{x\sim\mathcal{D}}\mathbb{E}_{a\sim\mathbf{r},\mathbf{e}\sim\mathbf{E}} \tag{5}$$
$$[aD_{\text{KL}}(M(\mathbf{p}\oplus x), M(\mathbf{e}\oplus x))],$$

where $D_{\text{KL}}(\cdot)$ is the Kullback Leibler distance function, $M(\cdot)$ is the output of the PLM. This loss function helps to find a more meaningful combination, i.e., discrete prompts with larger values should have outputs on downstream tasks that are as consistent as possible with the continuous prompt.

We learn the interpretation $\mathbf{r}$ by jointly minimizing the loss $\ell_1(\cdot)$ for the *Combination Hypothesis* (Eq.4) and the loss $\ell_2(\cdot)$ for downstream fidelity (Eq.5):

$$\ell^{'}(\mathbf{r}; \mathbf{E}, \mathbf{p}, \mathcal{D}) = \ell_1(\mathbf{r}; \mathbf{E}, \mathbf{p}) + \gamma\ell_2(\mathbf{r}; \mathbf{E}, \mathbf{p}, \mathcal{D}), \tag{6}$$

$$\tilde{\mathbf{r}} = \arg\min_{\mathbf{r}\in\mathbb{R}^v} \ell^{'}(\mathbf{r}; \mathbf{E}, \mathbf{p}, \mathcal{D}), \tag{7}$$

where γ is a hyperparameter. In this paper, we find $\gamma = 0.09$ to achieve a reasonable trade-off between prompt fidelity and downstream fidelity (see §9).

Nonetheless, it is time-consuming since the second optimization requires traversing the vocabulary of the PLM. As a post-hoc interpretation, we argue that the decoupling result $\mathbf{r}$ should be *sparse*, i.e., most of the discrete prompts should correspond to 0. On the one hand, a dense interpretation is incomprehensible; on the other hand, as an effective prompt that motivates the PLM to output desired outputs, it should not have much useless token information.

We propose a simple method that traverses the full downstream dataset and selects the v tokens with the highest frequency into our new vocabulary since it is intuitive that critical tokens contained in continuous prompts tend to appear in the dataset to be trained already. Moreover, since the parameters of the PLM are fixed, $M(\mathbf{e}\oplus x)$ is invariant in different epochs. Thus, for a given discrete prompt $\mathbf{e}$ and sample x, we only need to compute the output once, which further speeds up the training.

4 Studying P-tuning: Experimental Setup

4.1 Model and Training Details

P-tuning (Liu et al., 2021b), as a typical representative of continuous prompts, is used in this paper to study our proposed framework. For PLMs, we use the base version of BERT (Devlin et al., 2019), which is broadly adopted in the NLP field.

We freeze the parameters of BERT and use the prompt template $T = \{\mathbf{x}, [\mathbf{p}_1], [\mathbf{p}_2], \mathbf{y}, [\mathbf{p}_3]\}$, where $[\mathbf{p}_1], [\mathbf{p}_2], [\mathbf{p}_3]$ are the only trainable parameters with a two-layer LSTM (Hochreiter and Schmidhuber, 1997) head, respectively. We use a batch size of 8, initial learning rate of 0.00001, AdamW optimizier (Loshchilov and Hutter, 2019), and 15 training epochs for P-tuning; initial learning rate of 0.01, L1 loss coefficient of 0.01 and 4000 steps for training our interpretations with early stopping based on the validation set. Unless otherwise stated, all experiments are conducted in the 100-shot scenario.

4.2 Studied Datasets

Detailed experiments are conducted on the following 4 classification datasets: SST-2 (Socher et al., 2013), IMDB (Maas et al., 2011), Amazon Review Polarity (McAuley and Leskovec, 2013) and AG-News (Zhang et al., 2015). Statistics and target tokens for each dataset are attached in Appendix A

	SST-2			IMDB			Amazon			AGNews		
	50-shot	100-shot	Full	50-shot	100-shot	Full	50-shot	100-shot	Full	50-shot	100-shot	Full
P-tuning	71.11	78.36	**86.91**	68.08	71.50	**87.21**	72.26	78.08	**92.35**	84.49	**86.46**	**90.36**
Random	49.92	49.92	49.92	51.75	51.75	51.75	50.43	50.43	50.43	39.67	39.67	39.67
Discrete-500	73.86	71.28	83.25	70.15	75.31	83.06	77.97	**81.56**	85.28	83.74	84.50	87.21
Discrete-768	**77.70**	**78.42**	85.06	67.32	72.45	83.98	77.65	80.88	86.38	**84.74**	84.74	87.54
Discrete-1000	69.69	75.62	82.92	**71.72**	72.16	83.12	**79.17**	80.52	86.25	83.64	84.38	86.96
Discrete-1500	75.56	76.55	83.80	70.19	**78.16**	83.60	76.25	77.72	86.51	83.92	84.99	87.56

Table 1: Comparison of P-tuning and discrete prompts combinations (Discrete-v) on different tasks, where v is the vocabulary size, k-shot columns are trained on few-shot scenarios with k samples for each label, and Full columns are trained on the full datasets.

and B. Among all these datasets, test set accuracy is reported as our evaluation metric.

5 Hypothesis Verification

The *Combination Hypothesis* argues the existence of combinations of discrete prompts in fairly small neighborhoods as an alternative to continuous prompts. Therefore, it should also be feasible to train combinations of discrete prompts individually for downstream tasks. The amount of loss is quantified as follows:

$$\ell(\mathbf{r}; \mathbf{E}, \mathcal{D}) = \mathbb{E}_{x,y\sim\mathcal{D}}[\text{loss}(M(\mathbf{r}^\top \mathbf{E} \oplus x), y)], \quad (8)$$

where loss$(\cdot)$ is the loss function on downstream task. We then minimize the loss function to obtain a replacement for continuous prompts. The optimized performance is provided in Table 1. Our method performs competitively, especially in few-shot scenarios. Furthermore, we find that $v = 1500$ is sufficient for the model to obtain good performance, while a larger vocabulary size is more likely to introduce noisy tokens, which is not conducive to optimization. Therefore, we set $v = 1500$ in the following research.

Note that since the designed structure itself is difficult to optimize, we set the learning rate to 0.3 when trained on few-shot scenarios and 0.1 when trained on full datasets. Besides, the L1 loss function with a coefficient of 0.01 is added. This method does not aim to fully surpass P-tuning, but to verify the feasibility of the hypothesis that continuous prompt can be replaced by the full connection of discrete prompts without loss of precision and at the same time provide methods for faithfulness verification in §6. As an approximate alternative to continuous prompts, the loss of accuracy is unavoidable. For example, P-tuning is able to accurately find the simple connection between features and labels on full datasets, while it is more tricky for our method.

Prompt 1		Prompt 2		Prompt 3		Downstream
Token	PCT	Token	PCT	Token	PCT	ACC($\mathbf{p}$) $\xrightarrow{\Delta}$ ACC($\mathbf{r}^\top\mathbf{E}$)
What	41.30	exactly	0.91	things	30.10	67.98 $\xrightarrow{-4.22}$ 63.76
feeling	22.78	drama	4.53	quality	21.75	61.56 $\xrightarrow{+2.47}$ 64.03
cat	1.32	what	25.68	things	28.63	62.77 $\xrightarrow{-2.75}$ 60.02

Table 2: Performance of prompt fidelity and downstream fidelity on discrete prompts (SST-2). For prompt fidelity, the percentage (%) of corresponding tokens in the interpretations is reported (left). For downstream fidelity, comparisons of the accuracy (%) between continuous prompts $\mathbf{p}$ and our interpretations $\mathbf{r}^\top\mathbf{E}$ on downstream tasks are reported (right).

6 Faithfulness Verification

6.1 Do the Interpretations Faithfully Reflect the Source Prompts?

In this section, We verify the prompt fidelity and downstream fidelity of the interpretations, i.e., the proximity of the weighted discrete prompts to the source prompts and the similarity in performance on downstream tasks.

To obtain ground-truth labels, we first design three manual discrete templates on SST-2 and interpret them. The performance of prompt fidelity and downstream fidelity are shown in Table 2, where initial capitalization and plural forms are ignored. For most tokens, they account for more than 20% among the 1500 tokens. We consider this to be a fairly high value and the synonyms of the original tokens also achieve a high value. However, several tokens like "exactly", "drama" and "cat" still achieve a low value. For tokens like "exactly" and "drama", the interpretations discover their synonyms and give them an extremely high percentage (>20%), such as "completely" for "exactly" and "film" for "drama". For tokens like "cat", since they do not help with downstream tasks, the model can only attempt to optimize for the first objective (Eq. 4), leading to a jumbled interpretation. As for

	SST-2	IMDB	Amazon	AGNews
Nearest-1	0.0026	0.0030	0.0036	0.0047
Nearest-2	0.0027	0.0030	0.0037	0.0048
Ours	**0.0025**	**0.0027**	**0.0032**	**0.0043**

Table 3: Performance of prompt fidelity on continuous prompts (average squared distance reported).

	SST-2	IMDB	Amazon	AGNews
Nearest-1	49.86	50.12	50.06	49.55
Nearest-2	50.19	60.87	50.12	55.50
Ours	**75.18**	**66.77**	**69.72**	**74.21**

Table 4: Performance of downstream fidelity on continuous prompts.

downstream fidelity, the performance of the interpretations is similar to the source prompts in all 3 sets of experiments.

Furthermore, we verify the fidelity of the interpretations to continuous prompts. Performance of prompt fidelity and downstream fidelity is shown in Table 3 and Table 4, respectively. For comparison, the two nearest tokens in the Euclidean space are selected as interpretations for continuous prompts. Among all tasks, the distance of our results from the source prompts is closer compared to the nearest discrete token, indicating that our method has a higher fidelity in the restoration of source prompts. Moreover, simply taking the two nearest discrete tokens as a replacement for continuous prompts performs quite poorly on downstream tasks, even similar to random predictions in most cases, while our method achieves comparable performance to the source prompts on downstream tasks. In summary, our interpretations consistently maintain higher fidelity than the only existing method (select the nearest discrete prompts) and reflect the decision process of the source prompts well.

6.2 How Reductive are the Interpretations on Downstream Tasks?

As described in §3.3, the interpretations are intended to be sparse, which means that the top few tokens of interpretations are supposed to contain the majority of information from source prompts. In this section, We select the top five tokens of the interpretations as vocabulary and train the weighting of these tokens using the optimization method in §5. Comparison with baselines is shown in Table 5. In all scenarios, the tokens selected by our interpretations are more reductive than the randomly selected tokens and the tokens selected nearest to the continuous prompts, implying that these tokens do contain more information relevant to the downstream task from source continuous prompts.

Moreover, for a more visual demonstration of the ability of the selected five tokens to restore performance, we show the test set accuracy of several baselines under different training scenarios, including Manually, LM-BFF (Gao et al. (2021)) and P-tuning. For Manually, we report the best performance among the five manually designed templates (see Appendix E). For LM-BFF, we only use it to automatically generate templates without changing target tokens and making additional fine-tuning. It can be found that the performance of the five tokens selected by our method can outperform the templates selected by Manually and LM-BFF in all cases, and is even comparable to P-tuning in few-shot scenarios, while random selection and nearest neighbor selection are not. This further shows that our selected tokens are reliable and faithful.

7 Plausibility Verification

7.1 What do the Interpretations Look Like?

Still taking the 100-shot scenario as an example, we show our interpretations on different tasks in Table 6. For each prompt, five tokens with the largest values are selected for display. As a comparison, the five nearest tokens to each prompt in the Euclidean space are also selected for display.

As can be seen, our interpretations better reflect the decision-making of continuous prompts and output meaningful tokens compared to the *Nearest* baseline. For example, the continuous prompts in SST-2 induce the PLM to determine how great or terrible something in the input is, while prompts in IMDB and Amazon induce the PLM to judge how well someone thinks of something.

To our surprise, there contains a large number of task-independent tokens which also induce the PLM to output desired target tokens. For example, the interpretations on SST-2 contain tokens like "taste", "material" and "quality". These tokens are irrelevant to movie review sentiment classification, but can prompt the PLM to output the target tokens "terrible" or "great". We consider that continuous prompts may sneak in *shortcuts* (Geirhos et al., 2020) during training, which will be briefly verified in §7.2.

Nonetheless, there still remain several noisy to-

	SST-2			IMDB			Amazon			AGNews		
Manually	50.80			59.01			58.83			68.76		
Scenario	50-shot	100-shot	Full	50-shot	100-shot	Full	50-shot	100-shot	Full	50-shot	100-shot	Full
LM-BFF	64.85	64.85	-	57.91	57.91	-	59.05	59.05	-	55.37	55.37	-
P-tuning	71.11	78.36	86.91	68.08	71.50	87.21	72.26	78.08	92.35	84.49	86.46	90.36
Random	52.61	57.11	64.36	62.50	60.72	66.20	64.37	63.28	71.94	70.16	72.42	69.70
Nearest	59.75	53.98	67.49	64.88	65.34	67.76	66.90	71.84	67.28	69.50	58.22	70.05
Ours	**75.01**	**74.79**	**74.90**	**72.43**	**70.17**	**73.45**	**78.96**	**79.44**	**80.76**	**78.49**	**79.01**	**79.80**

Table 5: Performance of downstream tasks by training combinations of five selected tokens, including random selection, nearest to continuous prompts, and our method.

kens that are hard to understand for humans, especially in AGNews. These tokens seem irrelevant to the downstream task and it is difficult to spot potential shortcuts. We believe there are two reasons for the phenomenon. On the one hand, the tokens utilized by prompts are overcrowded in the semantic space, leading to the replacement of the interpreted tokens by irrelevant ones. On the other hand, the high complexity of the downstream task leads to a more difficult optimization of the interpretations. Future work will be conducted along these two directions.

7.2 Do Continuous Prompts Contain Shortcuts?

As shown in Table 6, our interpretations reveal the possibility of continuous prompts using shortcuts, which perform well on benchmarks but may fail to transfer on the anomaly test set (Geirhos et al., 2020). Taking the interpretation of SST-2 as an example, It contains unexpected tokens like "something", "taste", etc. to induce the PLM for desired target labels "terrible" or "great".

To test whether the model makes use of these shortcuts, we select several task-irrelevant texts containing shortcut tokens as suffixes to be added to the test set text and reverse the sentiment polarity of the added text to the test set labels on SST-2 (see Table 7.2). For example, "The food tastes delicious." is added if the ground-truth label is 0 (terrible), while "The food tastes unpalatable." is added if the ground-truth label is 1 (great). The significantly degraded performance suggests that the model utilizes a large number of shortcuts. To our surprise, these shortcuts do not disappear as the training data increases but are more fully exploited by the model, resulting in an accuracy of almost 0 after training on the full dataset. Obviously, continuous prompts of SST-2 are just baiting the PLM to output the target token terrible/great, not caring whether it is really a review of the movie or a review of food, cats, or something else. We present this phenomenon in the hope that it will attract more attention and research in the future.

8 Cross-Model Transfer

Due to the inconsistent embedding dimensions and semantic spaces of different PLMs, cross-model transfer of continuous prompts is tricky. With our proposed interpretable framework that establishes connections between continuous and discrete prompts, it becomes feasible to transfer continuous prompts from source PLMs to target PLMs without extra training signals on target PLMs. Considering a scenario to transfer continuous prompts of the source PLM M_a to the target PLM M_b, we can first get the decoupling results $\mathbf{r}$ using the method presented in §3.3. Then the continuous prompts transferred to M_b are $\mathbf{r}^\top \mathbf{E}_b$, where $\mathbf{E}_b$ is the discrete prompt matrix of M_b.

Following this idea, we investigate the feasibility of cross-model transfer from $\text{BERT}_{\text{base}}$ (Devlin et al., 2019) to $\text{BERT}_{\text{large}}$, $\text{RoBERTa}_{\text{base}}$ and $\text{RoBERTa}_{\text{large}}$ (Liu et al., 2019) respectively in Table 8. Considering that only discrete templates are capable of cross-model transfer without extra training signals on target PLMs in existing studies, we choose (1) select the nearest tokens to continuous prompts; (2) manually designed templates that perform best on $\text{BERT}_{\text{base}}$; and (3) automatically generated templates using LM-BFF (Gao et al. (2021)) as the baselines. For LM-BFF, we automatically generate templates using T5_{base} (Raffel et al., 2020) in the 100-shot scenario for cross-model transfer. Detailed results on the baseline (2) and (3) can be found in Appendix E.

As can be seen, our method outperforms baselines in most scenarios, especially on tasks like AGNews where it is tricky to construct discrete

	Prompt 1			Prompt 2			Prompt 3		
	Nearest	Ours		Nearest	Ours		Nearest	Ours	
SST-2	the	something	0.867	of	dark	0.245	the	involving	0.649
	his	those	0.010	the	taste	0.168	is	seem	0.130
	of	horror	0.004	his	material	0.047	of	things	0.046
	.	what	0.004	is	quality	0.330	was	touching	0.033
	him	bad	0.004	him	drama	0.275	several	working	0.018
IMDB	was	he	0.334	was	highly	1.295	was	anything	0.345
	.	during	0.309	.	how	0.220	were	##ness	0.229
	The	someone	0.087	were	particularly	0.144	the	atmosphere	0.200
	were	having	0.072	the	himself	0.051	of	##ful	0.163
	the	obviously	0.062	The	acted	0.048	The	theater	0.105
Amazon	.	he	0.856	seemed	completely	1.316	seemed	became	0.451
	,	guy	0.098	performance	heat	0.100	was	down	0.354
	the	having	0.086	seems	How	0.097	him	##le	0.238
	their	kid	0.080	him	nearly	0.075	became	seemed	0.089
	of	terrible	0.066	became	totally	0.059	would	scene	0.080
AGNews	National	Free	1.335	National	future	0.970	National	should	1.109
	2005	than	0.941	2005	senior	0.004	2004	control	0.046
	2004	toward	0.223	2004	Free	0.004	government	Department	0.008
	2006	Chief	0.031	Central	##ive	0.003	2006	might	0.006
	senior	likely	0.011	national	Top	0.002	2005	Research	0.004

Table 6: Intuitive comparison of interpretations on various continuous prompts, where *Nearest* selects the five nearest tokens to continuous prompts in the Euclidean space and *ours* selects the five largest values and their corresponding tokens using our proposed method.

	50-shot	100-shot	Full
(Raw Test Set)	71.11	78.36	86.91
The food *tastes* delicious/unpalatable.	57.50	52.28	2.53
Those cats *seem* to be great/terrible.	47.28	45.58	9.50
Something dark is of good/bad *quality*.	43.71	40.03	3.84

Table 7: Performance of continuous prompts on the SST-2 test set with shortcut tokens. Three task-irrelevant texts are selected to be added to the test set texts with sentiment polarity opposite to the ground-truth labels.

templates using prior knowledge. This enables zero-shot transfer of continuous prompts across arbitrary models without the restrictions of vector dimensionality and semantic space. For the poor performance on SST-2, we consider that the continuous prompts learned using $\text{BERT}_{\text{base}}$ inherently contain a large number of shortcuts, which may no longer be applicable after being captured by the interpretations and transferred to larger PLMs. Therefore, the performance of cross-model transfer is affected by the robustness of the source prompts. If continuous prompts are trained on larger PLMs and datasets, better performance will be obtained using our interpretations and is expected to be applied to areas such as model compression.

9 Further Analysis

Effect of Gamma. We analyze the effect of hyperparameter γ, i.e., the trade-off between prompt fidelity and downstream fidelity (Eq.6). Intuitively, as γ increases, the prompt fidelity decreases while the downstream fidelity goes up. When γ is 0, our method degenerates to use only prompt fidelity as the optimization objective. Fig. 3 shows the results of the grid search using the interpretations described in §3. As expected, the accuracy on $\text{BERT}_{\text{base}}$ improves as gamma increases since the interpretations are directly optimized on it. Nonetheless, when γ is larger than 0.09, the performance of the interpretations for cross-model transfer decreases. As a consequence, we choose $\gamma = 0.09$ in this paper.

10 Conclusion

In this paper, we present a novel view that interprets continuous prompts as a combination of discrete prompts. Contrary to the previous perspective which attempts to discover a one-to-one mapping between continuous prompts and discrete prompts, we demonstrate the continuous prompt to be an embedding lookup table with the one-hot restriction

	SST-2			IMDB			Amazon			AGNews		
P-tuning on M_a	78.36			71.50			78.08			86.46		
Transferred Model	M_b	M_c	M_d	M_b	M_c	M_d	M_b	M_c	M_d	M_b	M_c	M_d
P-tuning	74.52	87.70	73.09	77.02	80.70	88.84	85.25	92.21	84.07	86.66	85.63	82.82
Random	50.58	50.01	50.03	61.67	73.37	76.54	61.20	79.88	77.16	42.95	61.64	53.96
Nearest	50.19	50.52	49.97	57.02	59.29	62.38	51.41	87.03	75.80	56.37	51.38	44.34
Manually	54.20	72.54	83.91	53.68	56.58	70.12	51.78	53.30	78.23	71.29	46.22	48.64
LM-BFF	**75.12**	**81.38**	**86.05**	61.30	73.13	75.94	60.85	**83.30**	85.48	58.03	57.21	59.50
Ours	69.58	69.74	74.63	**72.52**	**75.69**	**80.18**	**75.02**	82.30	**90.33**	**76.04**	**69.91**	**68.75**

Table 8: Performance of cross-model transfer, including P-tuning on source PLMs, non-transferred baselines (P-tuning and random prompts), transferred baselines (Nearest, Manually Designed and LM-BFF) and our proposed method, where M_a, M_b, M_c, and M_d refer to $\text{BERT}_{\text{base}}$, $\text{BERT}_{\text{large}}$, $\text{RoBERTa}_{\text{base}}$, and $\text{RoBERTa}_{\text{large}}$, respectively. All experimental setups are similar to Table 6, with $\text{BERT}_{\text{base}}$ adopted as the source PLM in 100-shot scenario.

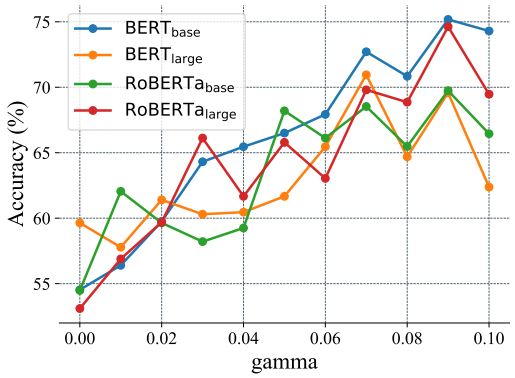

Figure 3: Effect of γ for downstream fidelity (blue) and cross-model transfer (others) on SST-2.

removed. Detailed experiments are conducted to verify that our interpretations faithfully reflect the reasoning of source prompts with both prompt fidelity and downstream fidelity. Furthermore, our interpretations exhibit promising readability and plausibility, which not only provides a tool for understanding model decisions but also offers a chance for discovering potential shortcuts contained in the prompts. Finally, with the bridge between continuous prompts and discrete prompts, we analyze the feasibility of cross-model transfer for continuous prompts with the proposed method. Results show that even trained on a small PLM ($\text{BERT}_{\text{base}}$) and 100-shot scenario, continuous prompts maintain good performance after transferring to various large PLMs. We hope that this work will bring a novel view for interpreting continuous prompts and encourage more research to explore the internal mechanisms of continuous prompts.

Acknowledgements

This work is partly supported by the Joint Funds of the National Natural Science Foundation of China under No. U21B2020 and the Shanghai Science and Technology Plan under No. 22511104400.

Limitations

Although the proposed method provides interpretations for continuous prompts with both faithfulness and plausibility, it can still only be used as an approximation to find the most likely combination, since the process of combining discrete prompts to continuous prompts is irreversible. Moreover, the output layer of PLMs tends to degenerate and occupy an anisotropic cone in the vector space (Wang et al., 2020; Li et al., 2020b), which significantly increases the difficulty of finding the correct interpretations. We encourage future research to take the magnitude of token vectors and the tokens in their neighborhoods into consideration for a more robust interpretation.

Due to space and time constraints, we only perform detailed experiments on P-tuning and the bidirectional language models like BERT and RoBERTa, which ignored numerous SOTA works such as Prefix Tuning (Li and Liang, 2021), Prompt Tuning (Lester et al., 2021) for continuous prompts and GPT (Radford et al., 2019), T5 (Raffel et al., 2020) for PLMs. We encourage future research to conduct experiments on more prompt methods and PLMs to investigate the generalizability of our method.

Ethical Statement

We propose a novel view to interpret continuous prompts, which have been considered "black boxes", as combinations of human-understandable discrete tokens. Since the method itself is unbiased and faithful, and all experiments are conducted on publicly available datasets, we believe that our work does not create any potential ethical risk.

Further, we discover shortcuts latent in continuous prompts, implying that systematic biases or discrimination may also exist in continuous prompts. These biases may originate from training datasets which are exploited by continuous prompts as a shortcut to the acquisition of true labels, or even originate from artificially implanted backdoors. We hope this work will provide the possibility to detect these potential biases in continuous prompts.

Our created artifacts are intended to provide researchers or users with a tool for understanding decision-making and detecting possible unexpected shortcuts of continuous prompts, while at the same time offering the feasibility of cross-model transfer without extra training signals on target PLMs. They are compatible with the original access conditions. All use of existing artifacts is consistent with their intended use in this paper.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher 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 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single \$&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2126–2136. Association for Computational Linguistics.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Haitao Zheng, and Maosong Sun. 2022. Openprompt: An open-source framework for prompt-learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, ACL 2022 - System Demonstrations, Dublin, Ireland, May 22-27, 2022*, pages 105–113. Association for Computational Linguistics.

Charles Dugas, Yoshua Bengio, François Bélisle, Claude Nadeau, and René Garcia. 2000. Incorporating second-order functional knowledge for better option pricing. In *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA*, pages 472–478. MIT Press.

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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3816–3830. Association for Computational Linguistics.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard S. Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. 2020. Shortcut learning in deep neural networks. *Nat. Mach. Intell.*, 2(11):665–673.

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, EACL 2021, Online, April 19 - 23, 2021*, pages 3618–3623. Association for Computational Linguistics.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 2733–2743. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable NLP systems: How should we

define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4198–4205. Association for Computational Linguistics.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know. *Trans. Assoc. Comput. Linguistics*, 8:423–438.

Pride Kavumba, Ryo Takahashi, and Yusuke Oda. 2022. Are prompt-based models clueless? In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 2333–2352. Association for Computational Linguistics.

Daniel Khashabi, Xinxi Lyu, Sewon Min, Lianhui Qin, Kyle Richardson, Sean Welleck, Hannaneh Hajishirzi, Tushar Khot, Ashish Sabharwal, Sameer Singh, and Yejin Choi. 2022. Prompt waywardness: The curious case of discretized interpretation of continuous prompts. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 3631–3643. Association for Computational Linguistics.

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, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3045–3059. Association for Computational Linguistics.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020a. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 9119–9130. Association for Computational Linguistics.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020b. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 9119–9130. Association for Computational Linguistics.

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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4582–4597. Association for Computational Linguistics.

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

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. GPT understands, too. *CoRR*, abs/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. *CoRR*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 142–150. The Association for Computer Linguistics.

Julian J. McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Seventh ACM Conference on Recommender Systems, RecSys '13, Hong Kong, China, October 12-16, 2013*, pages 165–172. ACM.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 11054–11070.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. 2019. Language models as knowledge bases? 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 2019, Hong Kong, China, November 3-7, 2019, pages 2463–2473. Association for Computational Linguistics.

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, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 5203–5212. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Teven Le Scao and Alexander M. Rush. 2021. How many data points is a prompt worth? In *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*, pages 2627–2636. Association for Computational Linguistics.

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 the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4222–4235. Association for Computational Linguistics.

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 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1631–1642. ACL.

Yusheng Su, Xiaozhi Wang, Yujia Qin, Chi-Min Chan, Yankai Lin, Huadong Wang, Kaiyue Wen, Zhiyuan Liu, Peng Li, Juanzi Li, Lei Hou, Maosong Sun, and Jie Zhou. 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, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 3949–3969. Association for Computational Linguistics.

Lingxiao Wang, Jing Huang, Kevin Huang, Ziniu Hu, Guangtao Wang, and Quanquan Gu. 2020. Improving neural language generation with spectrum control. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Albert Webson and Ellie Pavlick. 2022. Do prompt-based models really understand the meaning of their prompts? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 2300–2344. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 38–45. Association for Computational Linguistics.

Ningyu Zhang, Luoqiu Li, Xiang Chen, Shumin Deng, Zhen Bi, Chuanqi Tan, Fei Huang, and Huajun Chen. 2022. Differentiable prompt makes pre-trained language models better few-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. Factual probing is [MASK]: learning vs. learning to recall. In *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*, pages 5017–5033. Association for Computational Linguistics.

A Dataset

General descriptions and Statistics of the datasets we mentioned above are shown in Table 9 and 10. For few-shot scenarios, We randomly sample the same dataset for all tasks with the random seed set to 123.

Dataset	Language	Domain
SST-2	English	Moive Review
IMDB	English	Moive Review
Amazon	English	Product Review
AGNews	English	News Report

Table 9: General descriptions of datasets.

Dataset	Train	Valid	Test
SST-2	6920	872	1821
IMDB	20000	5000	25000
Amazon	2000000	1600000	400000
AGNews	80000	40000	7600

Table 10: Statistics of datasets.

B Target Tokens

Manual verbalizers are adopted in this paper. We rank the target tokens by their likelihoods and select the target token with the maximum likelihood as the classification output. The used target tokens for each task are shown in Table 11.

C Usage of Existing Packages

The pre-processing steps and prompt-based methods are all implemented in OpenPrompt (Ding et al., 2022), an open-source framework for deploying prompt learning. Our interpretable method is implemented in PyTorch (Paszke et al., 2019), an open-source framework for deploying deep learning algorithms. For PLMs, we use "bert-base-cased" as the base model, "bert-large-cased", "roberta-base", "roberta-large" for cross-model transfer, and "T5-base" for generating templates in LM-BFF from Huggingface transformers (Wolf et al., 2020). All licenses of these packages allow us for normal research use.

Identical hyperparameters are adopted regardless of the dataset. Detailed setups for P-tuning and our interpretable method are already shown in §4.1.

Dataset	Target Tokens
SST-2	terrible, great
IMDB	bad, good
Amazon	bad, good
AGNews	politics, sports, business, technology

Table 11: Target tokens of classification tasks.

For the LM-BFF baseline, we fix the target tokens and only use $T5_{base}$ to search for the best discrete template with the training epochs of 10, learning rate of 0.00001, batch size of 2, and beam width of 100.

D Experimental Details

For all the experiments mentioned in this paper, we use 2 NVIDIA GeForce GTX 1080 Ti GPUs with 11G memory each. For training our interpretable framework, an additional linear layer with $n \times v$ parameters is introduced besides the source PLM, where n denotes the number of continuous prompts and v denotes the vocabulary size. In this paper, we set $n = 3, v = 1500$, which means only 4,500 extra parameters are introduced. Compared to large-scale PLMs such as BERT or RoBERTa, these parameters are almost negligible.

E Performance of Discrete Templates

The performance of the manually designed templates (the first five rows of each table) and the templates generated by LM-BFF (the last row of each table) on each task and PLM is shown in Table 12-15. For manually designed templates, the best-performing templates on $BERT_{base}$ are selected as the baseline templates for cross-model transfer.

Templates	$\text{BERT}_{\text{base}}$	$\text{BERT}_{\text{large}}$	$\text{RoBERTa}_{\text{base}}$	$\text{RoBERTa}_{\text{large}}$
{x}The sentiment :{y}.	50.25	50.36	56.07	74.74
{x}Terrible or great :{y}.	50.69	49.92	59.69	69.52
{x}Overall, it is a{y}film .	50.14	58.05	73.15	71.94
{x}It feels{y}about the film .	50.63	53.27	73.15	84.68
{x}The feeling of the review is{y}.	**50.80**	**54.20**	**72.54**	**83.91**
{x}It's{y}.	64.85	75.12	81.38	86.05

Table 12: Performance of discrete templates on SST-2.

Templates	$\text{BERT}_{\text{base}}$	$\text{BERT}_{\text{large}}$	$\text{RoBERTa}_{\text{base}}$	$\text{RoBERTa}_{\text{large}}$
{x}The sentiment:{y}.	50.35	50.92	72.98	79.18
{x}Bad or good:{y}.	**59.01**	**53.68**	**56.58**	**70.12**
{x}Overall, it is a{y}film.	57.04	63.17	77.48	83.95
{x}It feels{y}about the film.	50.54	51.75	72.22	72.46
{x}The feeling of the review is{y}.	50.48	57.40	73.51	72.58
{x}Very{y}.	57.91	61.30	73.13	75.94

Table 13: Performance of discrete templates on IMDB.

Templates	$\text{BERT}_{\text{base}}$	$\text{BERT}_{\text{large}}$	$\text{RoBERTa}_{\text{base}}$	$\text{RoBERTa}_{\text{large}}$
{x}The sentiment:{y}.	50.25	50.73	53.29	85.27
{x}Bad or good:{y}.	**58.83**	**51.78**	**53.30**	**78.23**
{x}Overall, it is a{y}product.	50.17	57.60	77.67	78.35
{x}It feels{y}about the product.	50.72	56.29	79.05	83.13
{x}The feeling of the review is{y}.	50.09	53.99	73.09	66.54
{x}Very{y}.	59.05	60.85	83.30	85.48

Table 14: Performance of discrete templates on Amazon.

Templates	$\text{BERT}_{\text{base}}$	$\text{BERT}_{\text{large}}$	$\text{RoBERTa}_{\text{base}}$	$\text{RoBERTa}_{\text{large}}$
{x}The topic is about{y}.	**68.76**	**71.29**	**46.22**	**48.64**
{x}The type of the news is{y}.	41.57	51.66	50.70	59.96
{x}News category:{y}.	51.95	75.29	80.78	79.64
{x}Overall, it is{y}news.	45.75	46.14	52.96	37.72
{x}What type is the news?{y}.	64.95	63.21	69.79	77.63
{x}in{y}.	55.37	58.03	57.21	59.50

Table 15: Performance of discrete templates on AGNews.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section Limitations.

☑ A2. Did you discuss any potential risks of your work?
Section Ethical Considerations.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section Abstract and Section 1.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Appendix C.

☑ B1. Did you cite the creators of artifacts you used?
Section References.

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Appendix C.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section Ethical Considerations.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
All the data we use is derived from widely used open-source datasets, which have undergone public scrutiny. Since our paper focuses only on the interpretability and transferability of continuous prompts, potentially privacy-invasive or offensive content contained in these datasets is not further discussed.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Appendix A.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Appendix A.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

C ☑ Did you run computational experiments?

Section 4.1, Section 5, Section 9 and Appendix D.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix D.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4.1, Section 5 and Section 9.

☒ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Since all experiments were run once with the same random seed 123, we did not report descriptive statistics such as error bars and summary statistics.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 4.1 and Appendix C.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.