GPS: Genetic Prompt Search for Efficient Few-shot Learning

Hanwei Xu*, **Yujun Chen***, **Yulun Du***,
Nan Shao, Yanggang Wang, Haiyu Li, Zhilin Yang†
Recurrent AI
{xuhanwei, chenyujun, duyulun, kimi_yang}@rcrai.com

Abstract

Prompt-based techniques have demostrated great potential for improving the few-shot generalization of pretrained language models. However, their performance heavily relies on the manual design of prompts and thus requires a lot of human efforts. In this paper, we introduce Genetic Prompt Search (GPS) to improve few-shot learning with prompts, which utilizes a genetic algorithm to automatically search for high-performing prompts. GPS is gradient-free and requires no update of model parameters but only a small validation set. Experiments on diverse datasets proved the effectiveness of GPS, which outperforms manual prompts by a large margin of 2.6 points. Our method is also better than other parameter-efficient tuning methods such as prompt tuning.

1 Introduction

Pretrained language models, such as BERT (Devlin et al., 2019), XLNet (Yang et al., 2019), T5 (Raffel et al., 2020), and GPT (Radford et al., 2018), are often finetuned for downstream natural language processing tasks, which has been shown to improve performance over non-pretrained models. However, this pretraining-finetuning paradigm still relies on a relatively large set of labeled data for each downstream task to obtain competitive performance. Although GPT-3 (Brown et al., 2020) shows promising performance for zero-shot and few-shot learning by prompting on an extremely-large pretrained language models with 175B parameters, finding out the optimum prompt for each given task could be difficult (Mishra et al., 2021; Wang et al., 2022).

To improve the performance of prompting on pretrained language models, recent works focus on supervised pretraining with carefully designed or crowdsourced manual prompts (Gao et al., 2021a; Wei et al., 2021; Sanh et al., 2021; Ouyang et al., 2022). Diverse prompts are collected to enhance the robustness and performance of prompting (Sanh et al., 2021). Ouyang et al. (2022) introduced a dataset of labeler demonstrations and used it to finetune GPT-3. Despite all these efforts, the challenge to obtain high-performing prompts for few-shot learning still exists. As pointed out by previous works (Liu et al., 2021b; Gao et al., 2021b; Liu et al., 2021a), manual prompts are usually suboptimal and suffers a high variance on performance.

To address this challenge, we propose a novel Genetic Prompt Search (GPS) algorithm that gradually mutates the prompts with a generative model and selects candidates according to their performance on a small development set. This evolutionary procedure relies on a tiny set of labeled data, only used for validation but not training. As illustrated in Figure 1, GPS does not require updating any parameter, but only searches for the optimal hard prompts for every downstream task. Similar to prompt tuning, GPS allows the pretrained model to serve a large number of applications simultaneously. Meanwhile, GPS is even easier to deploy than prompt tuning, because it does not need to store the tuned continuous soft prompts. Empirically, GPS achieves substantial improvement over the baseline of manual prompts, and it also outperforms other parameter-efficient few-shot tuning methods.

Our contributions can be summarized as follows.

- We propose a tuning-free Genetic Prompt Search method that only requires a small validation set to automatically search for high-performing prompts.
- Our experiments demonstrate that manual prompts are usually suboptimal. Using the proposed search method, some simple aug-

* Equal contribution
† Corresponding author
Code is available at https://github.com/hwxu20/GPS

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 8162 - 8171
December 7-11, 2022 ©2022 Association for Computational Linguistics

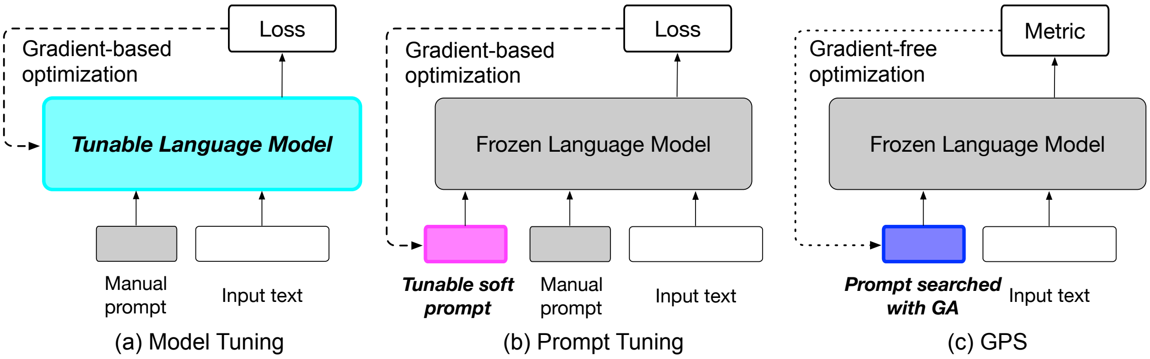

Figure 1: The paradigms of Model Tuning, Prompt Tuning, and GPS. Model Tuning requires the pretrained model to be tunable, and the tuned model can be only used for a single task. Prompt tuning needs extra tunable soft prompts. Our proposed GPS is tuning-free.

mentation technique such as back translation can lead to considerable improvement over manual prompts. Further, our best practice outperforms parameter-efficient few-shot tuning baselines. We conduct an overall comparison of different few-shot learning methods, and our proposed method stands out with the best performance as well as serving efficiency.

- We carefully studied the effects of hyper-parameters and prompt generation strategies in the proposed algorithm. Different from previous work (Shin et al., 2020), we find that our searched prompts are semantically fluent just as human-written templates.

2 Related Work

Recently, thanks to the prompt-based learning method, pretrained language models (PLMs) have been widely explored under zero-shot and few-shot scenarios for language understanding and generation tasks (Schick and Schütze, 2021; Gao et al., 2021b; Le Scao and Rush, 2021). Prompt-based learning bridges the gap between pretraining and finetuning objectives by stitching the text input X with a prompt template and augmenting the label output y as a text string, such that the input and the output can be constructed in a sentence completion task form. Previous few-shot learning methods can be generally categorized into two types, few-shot tuning methods that require updating parameters (Liu et al., 2021b; Han et al., 2021) and prompt enchancement methods that have no learnable parameter but optimize the discrete prompts directly (Shin et al., 2020; Mishra et al., 2021).

2.1 Few-Shot Tuning

Some few-shot tuning methods focus on template design and update all the parameters of pretrained language models. PET(Schick and Schütze, 2021) exploited the simple manual template and unified different classification tasks with pattern-verbalizer pairs. LM-BFF (Gao et al., 2021b) proposed several simple techniques for better few-shot learning including automatic verbalizer search and automatic prompt search. Han et al. (2021) applied rules in prompt tuning to deal with the hard many-class text classification tasks.

Another line of work is parameter-efficient few-shot learning, which aims at reducing the number of tunable parameters to improve deployment efficiency. Adapters (Houlsby et al., 2019) proposed to add an adpater module integrated in the original language model for each downstream task and only this module is tunable. Lester et al. (2021) showed the effectiveness of tuning the prompt embeddings only especially for large-scale models. P-tuning (Liu et al., 2021b) applied continuous prompt embedding optimization for GPT and made it comparable to BERT on NLU tasks. BitFit (Ben Zaken et al., 2022) tuned only the bias terms of the original model. A low-rank decomposition approach named LoRA was proposed in Hu et al. (2021), which injected trainable matrices in parallel with the original forward pass into each layer. Black-Box Tuning (Sun et al., 2022a) is a gradient-free optimization method for prompt tuning and thus it is suitable to use language models as a service. However, all these aforementioned methods require updating parameters, which is computationally expensive and costly in storage capacity for serving every task at hand. Our GPS, instead,

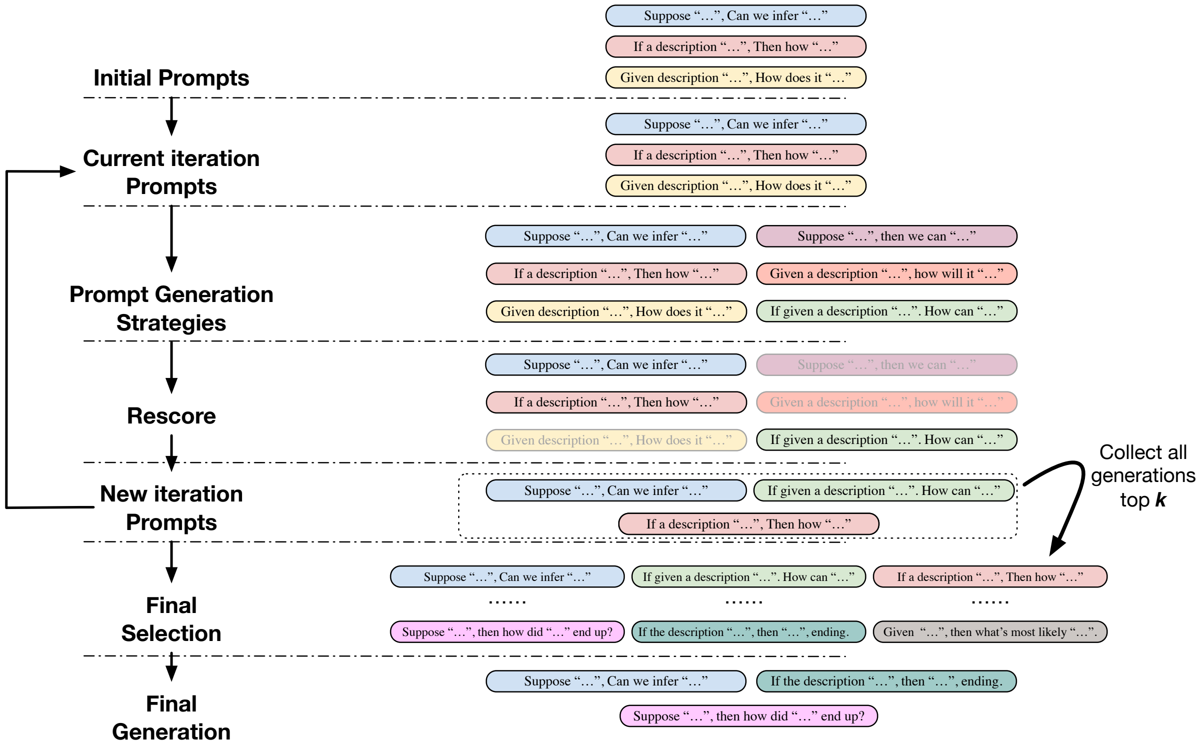
Initial Prompts
Suppose “…”, Can we infer “…”
If a description “…”, Then how “…”
Given description “…”, How does it “…”
Current iteration Prompts
Suppose “…”, Can we infer “…”
If a description “…”, Then how “…”
Given description “…”, How does it “…”
Prompt Generation Strategies
Suppose “…”, Can we infer “…”
Suppose “…”, then we can “…”
If a description “…”, Then how “…”
Given a description “…”, how will it “…”
Given description “…”, How does it “…”
If given a description “…”. How can “…”
Rescore
Suppose “…”, Can we infer “…”
Suppose “…”, then we can “…”
If a description “…”, Then how “…”
Given a description “…”, how will it “…”
Given description “…”, How does it “…”
If given a description “…”. How can “…”
New iteration Prompts
Suppose “…”, Can we infer “…”
If given a description “…”. How can “…”
If a description “…”, Then how “…”
Collect all generations top *k*
Final Selection
Suppose “…”, Can we infer “…”
If given a description “…”. How can “…”
If a description “…”, Then how “…”
……
……
……
Suppose “…”, then how did “…” end up?
If the description “…”, then “…”, ending.
Given “…”, then what’s most likely “…”.
Final Generation
Suppose “…”, Can we infer “…”
If the description “…”, then “…”, ending.
Suppose “…”, then how did “…” end up?

Figure 2: Overall pipeline of our GPS algorithm. The idea of GPS is borrowed from the genetic algorithm. Prompts are initialized from handcrafted prompts. Better prompts are searched for over each iteration. Finally, all generated prompts are reranked and selected as the final prompts.

is tuning-free and aims at searching for the optimal prompts.

2.2 Prompting Enhancement

GPT-3 (Brown et al., 2020) shows the effectiveness of In-Context Learning. However, discrete prompting requires human efforts to provide manual prompts, and its sensitivity to labeled examples makes it hard to obtain stable performance. AutoPrompt (Shin et al., 2020) proposed to search discrete prompts with a gradient-guide method, but the generated prompts are literally uninterpretable. GPTk (Mishra et al., 2021) explored several ways to manually reframe task instructions.

GRIPS (Prasad et al., 2022) is a concurrent work, which also applies iterative prompt search to improve the few-shot performance. Several operations, including add, deletion, swap and paraphrase, are defined to edit the manual prompts. Compared to GRIPS, our method uses different prompt reproduction approaches including back translation as well as cloze and sentence continuation by using generative language models. These methods do not need any human-defined edit rule and the generated prompts are semantically fluent. We conduct experiments comparing the performance of GRIPS and our method in Sec 4.

In this paper, we follow T0 (Sanh et al., 2021), which is a very powerful zero-shot baseline of multitask prompted training. Different from T0 and other methods, we regard the crowd-sourced manual prompts as seed prompts, and focus on parameter-free and gradient-free prompt search to further improve prompting performance under the few-shot learning setting.

3 Genetic Prompt Search

In this section, we will introduce the algorithm of Genetic Prompt Search (GPS) and various prompt generation strategies we have studied. Note that the prompt search here refers to the search for a high-performing hard prompt in the discrete word space as shown in Fig. 1, and the formulation does not include soft prompts.

3.1 Genetic Prompt Search Algorithm

It is challenging to automatically find high-performing prompts for a new unseen task. Inspired by Genetic Algorithms (Mitchell, 1980), we propose Genetic Prompt Search (GPS) for this purpose.

In GPS, we will first sample a tiny number of

Algorithm 1 Genetic Prompt Search

```
Require: $G^0$; $D_{dev}$; $f_{GPS}$; $g_{GPS}$; $T$; $K$;
Ensure: Final optimized prompts, $G^{T+1}$
obtain handcrafted prompts $G^0$ as initialization
for each $t \in [0, T]$ do
    store $G^t$
    calculate score for each prompt in $G^t$ using $f_{GPS}$,
    from $G^t$, select top $K$ prompts as reproductive group $G^t_*$,
    generate $G^{t+1}$ based on $G^t_*$ using $g_{GPS}$,
end for
from stored $\{G^0_*, ..., G^T_*\}$, select top $K$ prompts as optimal prompts group $G^{T+1}$ using $g_{GPS}$ .
return $G^{T+1}$;
```

data as a development set D_{dev} for each downstream task. Then, we will design two genetic functions, where f_{GPS} is the metric function to decide which prompts will be reserved or eliminated at each iteration, and g_{GPS} represents the genetic function to generate new prompts. The process of Genetic Prompt Search is described in Fig. 2. According to the algorithm, GPS is firstly initialized with a set of handcrafted prompts, G^0. And the key process of GPS is to reproduce the current generation of prompts and use re-scoring to select prompts iteratively. For each iteration, we calculate the scores of prompts in G^t using f_{GPS}, and select the top-K prompts as G^t_*. Then we generate G^{t+1} using g_{GPS} based on G^t_*. After several steps of genetic search, we will collect all the top-K prompts in each generation, and rescore all these prompts to make the final decision on which prompts are optimal.

Now we discuss several strategies to generate the candidates at each iteration.

3.2 Prompt Generation Strategies

Back Translation: Back Translation (BT), a common technique for data augmentation in NLP, is applied for prompt reproduction. Here we first translate the manual prompts from English to 11 other languages including Chinese, Japanese, Korean, French, Spanish, Italian, Russian, German, Arabic, Greek, Cantonese, and then translate them back to English.

Cloze: We introduce a prompt generation approach making use of the cloze task form and pretrained language models. Firstly, we follow previous work LM-BFF (Gao et al., 2021b), which is a suite of simple techniques for few-shot learning, and exploit its automatic template generation method. Specifically, we use the large pretrained text-to-text transformer (T5) (Raffel et al., 2020) to generate templates. For each input example and its verbalizer, we compose the template with placeholders as prefix and suffix, and let T5 to fill in the placeholders. We apply beam search to generate multiple prompt candidates. More details can be found in Gao et al. (2021b). However, this approach does not work well since our setting conducts no parameter update, which is different from the few-shot training setting in the original paper. Therefore, we instead use manual prompts as initial templates, replace some random tokens with placeholders, and then let T5 fill in the blanks to generate new prompts. We select the best prompt according to the average logits across all the validation samples.

Sentence Continuation: Another alternative for prompt augmentation is Sentence Continuation (SC). Inspired by DINO (Schick and Schütze, 2021), we use a pretrained language model to generate new prompts. Specifically, we use the template "Write two sentences that mean the same thing. Sentence 1: *Manual Prompt*, Sentence 2:" to the pretrained model, and let it generate continuations as a new prompt. We conducted experiments with GPT2-XL (1.5B) and T5LM-XXL (11B) as our prompt generation models.

Prompt Scoring: For Cloze, we follow previous work (Gao et al., 2021b) to score the prompts with average logits on the validation set D_{dev}. For Back Translation and Sentence Continuation, since averaging logits is not applicable, we score each prompt using accuracy on D_{dev}.

4 Experiments

In this section, we conduct extensive experiments to study the effectiveness of GPS, and reveal the way to obtain the best prompt for Genetic Prompt Search. We also study several possible impact factors and hyper-parameters in GPS.

4.1 Experimental Setups

To match with the real few-shot scenario, we use a small validation set randomly sampled from each task. Empirically, for every task, only 32 data samples are needed to build the validation set, and we keep the number of samples for each task the same to make the data balanced. The actual shot number will be 32 divided by the number of classes. For example, we will have 8 shots for each class if there are 4 classes. Our few-shot setting follows the "true few-shot" setting (Perez et al., 2021). For all the tuning-free methods that do not require a training set, we use the validation set to search for the optimal prompt. For all the methods that require tuning parameters, we split the validation set into two halves as a training set and a validation set. Therefore all the experiments use the same number of data for fair comparison. We repeat the experiments of few-shot methods with 3 different data splits and report the average performance across all prompts.

4.2 Datasets

We use the 10 test tasks of T0, which are not included in the prompted training tasks, to evaluate the performance of our GPS and other methods. There are various kinds of NLP tasks in the test set including natural language inference (ANLI R1, ANLI R2, ANLI R3, CB, RTE), coreference resolution (WSC, Winogrande), sentence completion (COPA, HellaSwag) and word sense disambiguation (WiC). We report the average accuracy of different prompts for all the tasks.

4.3 Baselines

We compare GPS under the few-shot learning setting with state-of-the-art methods. Here we categorize the baselines to three groups: the manual prompt baseline, methods with tunable parameters, and methods without tunable parameters.

Manual prompt baseline: T0 (Sanh et al., 2021) is a multitask pretrained encoder-decoder model, which is on the basis of T5 and further pretrained on different types of downstream tasks with diverse manually designed prompts.

Methods w. tunable parameters: 1) Model Tuning (MT) is the common paradigm to finetune the entire pretrained language model on each task. 2) Prompt Tuning (Webson and Pavlick, 2021) (PT) is a gradient-guided tuning method, which only trains the extra continuous soft prompts while the pretrained language model is frozen. 3) Black-Box Tuning (Sun et al., 2022b) (BBT) is a gradient-free few-shot tuning method. Rather than searching for discrete text prompts, Black-Box Tuning aims at searching for the best soft prompt embedding in the continuous space.

Methods w.o. tunable parameters: 1) In-Context Learning (Brown et al., 2020) (ICL) is a common method of few-shot learning for large-scale pretrained language models. Demonstrations composed of labeled samples and manual templates are used to help the model understand the meaning of the test tasks. 2) GRIPS (Prasad et al., 2022) is a concurrent work which introduced a gradient-free edit-based method for optimal prompt search, but GRIPS mostly focus on simple rule-based editing operations such as add, deletion and swap.

We conduct experiments on English natural language processing tasks of which the manual prompts are introduced in T0 (Sanh et al., 2021). Note that all the seed prompts we used in our experiments were taken from T0. To make fair comparison, we used the same suite of prompts for other baseline approaches including Model Tuning, Prompt Tuning, Black-Box Tuning and GRIPS.

For Prompt Tuning, we use the Adafactor Optimizer and set the learning rate as 0.05. For Model Tuning, we use the same Optimizer as Prompt Tuning and set the learning rate as 5e-5. The batch size is set as 4 for both prompt tuning and model tuning. For Black-Box Tuning, we take 500 as the intrinsic dimension, 20 as the pop size and the cross entropy loss. We report the best results with 1 and 50 soft prompt tokens. For In-Context Learning, we randomly select 2 examples from the training set for each task. For GRIPS, we try to keep all hyper-parameters the same as Prasad et al. (2022). The only difference is that the initial prompts are from T0.

4.4 Implementation Details

In practice, we assume there is only a few-shot validation set to conduct our experiments, which means, we will not tune any hyper-parameter in the method according to the performance on test set. Specifically, we set K as the number of initial prompts for each. It is a reasonable setup, because if the K is too low, the method may simply drop all prompts of low quality and keep the rest prompts as the final result. In the main experiment, we run the genetic prompt search for 6 steps. To generate diverse candidate prompts at each step, we perform top-p sampling, where the p is set as 0.9. Besides, we filter out all prompts that are the same as the existing prompts or do not have a valid input placeholder, such as "premise" in SuperGLUE CB.

Dataset	Zero-Shot		Few-Shot Parameter Tuning			Few-Shot Parameter Frozen		
	T0†	T0 ‡	BBT	PT	MT	ICL	GRIPS	Our GPS
ANLI R1	43.56	43.16	$42.97_{0.32}$	$43.21_{0.15}$	$\underline{46.73_{3.25}}$	$37.87_{0.80}$	$\mathbf{44.41_{0.01}}$	$44.06_{2.78}$
ANLI R2	38.68	38.68	$38.89_{0.05}$	$37.36_{1.69}$	$\underline{39.12_{3.17}}$	$34.51_{1.53}$	$\mathbf{39.57_{0.24}}$	$38.10_{1.68}$
ANLI R3	41.26	41.87	$41.32_{0.05}$	$40.85_{0.36}$	$\underline{42.20_{2.11}}$	$35.98_{2.80}$	$\mathbf{42.96_{0.42}}$	$41.51_{2.33}$
CB	70.12	70.12	$72.06_{0.14}$	$71.31_{2.39}$	$\underline{83.97_{5.40}}$	$59.21_{5.65}$	$76.55_{0.41}$	$\mathbf{80.12_{1.61}}$
RTE	80.83	80.97	$81.73_{0.49}$	$\underline{82.47_{0.86}}$	$79.51_{2.08}$	$64.86_{10.29}$	$81.71_{0.12}$	$\mathbf{84.22_{1.02}}$
WSC	61.45	61.06	$60.74_{0.31}$	$63.30_{1.82}$	$\underline{64.65_{1.79}}$	$61.03_{3.86}$	$61.47_{1.44}$	$\mathbf{63.62_{1.68}}$
Winogrande	59.94	59.70	$59.46_{0.32}$	$58.63_{0.70}$	$\underline{59.76_{1.47}}$	$53.22_{1.58}$	$58.11_{0.26}$	$\mathbf{59.59_{2.06}}$
COPA	90.02	90.02	$90.51_{0.74}$	$92.33_{0.39}$	$\underline{92.54_{0.98}}$	$82.82_{1.39}$	$91.75_{0.42}$	$\mathbf{93.50_{0.14}}$
HellaSwag	33.55	33.52	$33.47_{0.20}$	$37.28_{0.29}$	$\underline{49.75_{4.98}}$	$27.35_{2.28}$	$33.09_{0.20}$	$\mathbf{38.85_{5.54}}$
WiC	56.68	56.13	$57.09_{0.40}$	$58.86_{1.22}$	$\underline{59.04_{1.09}}$	$50.35_{0.89}$	$57.03_{0.67}$	$\mathbf{57.65_{1.18}}$
Avg	57.60	57.52	$57.82_{0.03}$	$58.56_{0.20}$	$\underline{61.73_{0.09}}$	$51.28_{1.66}$	$58.66_{0.35}$	$\mathbf{60.12_{1.40}}$

Table 1: Main results (accuracy) on the test benchmark of different methods. Black-Box Tuning (BBT), Prompt Tuning (PT), Model Tuning (MT) tune the continuous prompt embedding or the full model for few-shot learning, while In-Context Learning (ICL), GRIPS (Prasad et al., 2022) and our GPS do not update any parameter. We repeat the experiments of few-shot methods with 3 different data splits and report the average performance across all prompts. The number of subscript is the standard deviation across different data splits. T0, BBT, PT, MT, and ICL all use the same manual prompt set, while GRIPS and GPS use their final searched prompt sets, where the number of prompts are the same in different prompts sets. Underlined results are the best of few-shot parameter tuning methods and **bold** results are the best of few-shot parameter frozen methods. †: The original results from Sanh et al. (2021). ‡: The results we reproduced.

4.5 Main Results

In this section, we compare GPS with the aforementioned baselines on the 10 unseen tasks.

As shown in Table 1, GPS outperforms other few-shot learning methods that do not require parameter updating. Compared to GRIPS, another discrete prompt search method, GPS wins 1.4 points on average, and the performance of GRIPS is close to that of PT while worse than MT. This reveals that the way to conduct prompt generation is important to obtaining the optimal prompts, and we also conduct further ablation studies on different prompt generation strategies in Sec 4.6.1. Meanwhile, we find that the result of ICL with T0 is significantly worse than the zero-shot baseline, which differs from the results in previous works using GPT. We suppose this is because the prompts used in T0 at the multitask pretraining stage are quite different from the ICL demonstrations with labeled examples. Even compared with parameter-efficient tuning methods like black-box tuning and prompt tuning, GPS also achieves considerably better performance.

4.6 Ablation Study

In this section, we conduct several ablation experiments on various hyper-parameters. To control experimental variables, we explore the effect of each hyper-parameter while keeping the other hyper-parameters fixed as the default value.

Dataset	T0†	BT	Cloze	SC (GPT2)	SC (T5LM)
ANLI R1	43.16	44.95	42.10	44.64	**46.47**
ANLI R2	38.68	**40.13**	38.95	39.14	39.91
ANLI R3	41.87	42.83	42.00	42.75	**43.01**
CB	70.12	79.38	73.21	79.71	**80.00**
RTE	80.97	82.60	82.64	82.53	**83.86**
WSC	61.06	65.38	64.38	63.65	**65.48**
Winogrande	59.70	61.09	53.50	59.76	**61.96**
COPA	90.02	93.12	89.77	93.31	**93.43**
HellaSwag	33.52	36.34	33.83	36.72	**44.29**
WiC	56.13	**60.72**	56.15	57.91	58.82
Avg	57.52	60.65	57.65	60.01	**61.72**

Table 2: Ablation results with different prompt generation strategies including Cloze, Back Translation (BT) and Sentence Continuation (SC). For SC, we include two different pretrained models, GPT2 and T5LM. †: The results we reproduced.

4.6.1 Prompt Generation Strategies

In Table 2, we compare different automatic prompt generation strategies, including Cloze, Back Translation (BT), and Sentence Continuation (SC) as described in section 3.2. Among all the prompt generation strategies, SC with T5LM obtains the best results. And SC with T5LM shows that it outperforms SC with GPT2 by a significant margin. This reveals that the pretrained model for SC is vital to obtaining the optimal prompts, and larger language models usually generate better prompts.

BT achieves good results and has the best score

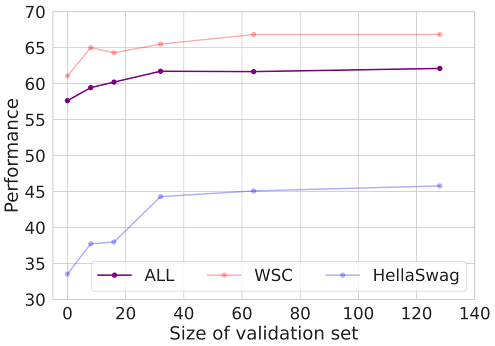

Figure 3: Ablation results on the size of the validation set. The performance with an empty validation set is zero-shot T0.

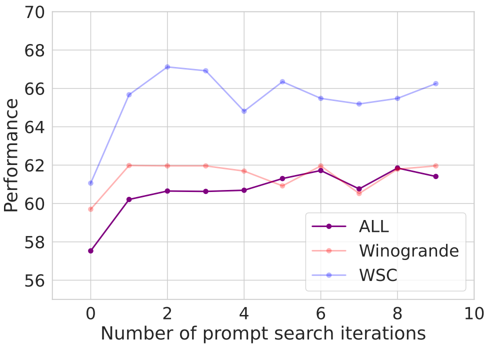

Figure 4: Ablation results on the number of prompt search iterations. The performance with 0 iterations is zero-shot T0.

on ANLI R2 and WiC. Cloze does not work well and its overall performance is no better than the zero-shot baseline. The results suggest that these simple strategies cannot provide enough variance of prompts for search, and we will discuss this with a few examples in section 4.7.

4.6.2 The Size of Validation Set

The validation set plays an important role in GPS for scoring each prompt, and its size matters to give credible feedback for the prompt selection. The results of GPS on validation sets of different sizes from 8 to 128 are presented in Fig. 3. Generally, the gain of prompt search rises with more validation samples. Most datasets follow this trend, such as WSC and HellaSwag. Although more examples lead to further improvement, we set the default size of validation set to 32 in our experiments because we focus on the few-shot scenario with limited labeled points. The size of the prompt pool might be important as well for continuous improvement, especially with a large validation set. Here we set the prompt pool size to be 30 in consideration of the computational costs. On the other hand, GPS is still much better than manual prompts when there is only 8 examples for validation.

4.6.3 The Number of Prompt Search Iterations

Another critical hyperparameter is the number of iterations for genetic prompt search. We experimented with up to 9 prompt sesarch iterations and the results are given in Fig. 4. It can be seen that the performances on some datasets such as WSC and Winogrande achieve the best at an early iteration. However, the overall performance on all the datasets improves on more search iterations. The default iteration number is set to 6 in the paper for the trade-off between the performance and costs.

4.7 Case Study

In Table 3, we present cases of prompts selected by GPS using different strategies. As we can see, GPS modifies the original prompts to optimize performance for unseen tasks while not changing the major meanings. However, using simple strategies like back translation can only provide minor prompt modifications, while SC with T5LM shows larger prompt modifications and dramatically better performance. For example, in WSC, GPS firstly removes the less informative adverb "In the passage above", and then adds a hint "the person of" to help the model navigate the answer. This modification obtains a significant 10 points gain compared to the original prompt. Overall, Table 3 illustrates the necessity and effectiveness of GPS, especially when SC is used on a large language model.

4.8 Overall Comparison of Different Few-Shot Methods

Table 4 compares different methods on serving efficiency, tunable parameters, performance, and computational cost.

Serving Efficiency. MT lacks serving efficiency due to the huge storage cost to store the full model for each new task. Although ICL does not have any tunable parameter, the long sequence length makes it expensive for inference, especially when the number of demonstrations is large. PT, BBT and GPS have few or zero tunable parameters, and thus they are cheaper for deployment.

Tunable Parameters. MT needs to tune the full model for each task, and it requires large resources

Task	Generated Prompts	Metric
Hellaswag	**Origin**: If a description of a situation begins like this: {{ ctx }}... Then how does it continue?	34.00
	BT: If the description of a situation begins like this: {{ ctx }}... Then how will it continue?	34.74
	SC(GPT2): -	34.00
	SC(T5LM): If a description of a situation begins like this: {{ ctx }}... then what is the most likely thing to happen next?	**47.63**
SuperGLUE COPA	**Origin**: Select the most plausible {% if question == "cause" %} cause: {% else %} effect: {% endif %}	93.00
	BT: Select the most believable {% if question == "cause" %} cause: {% else %} effect: {% endif %}	95.00
	SC(GPT2): What is the most plausible {% if question == "cause" %} cause: {% else %} effect:{% endif %}	95.00
	SC(T5LM): Select the most agreeable {% if question == "cause" %} cause: {% else %} effect:{% endif %}	**96.00**
SuperGLUE CB	**Origin**: {{premise}} Are we justified in saying that "{{hypothesis}}"? Yes, no, or maybe?	78.57
	BT: {{premise}} Do we have reason to say this "{{hypothesis}}"? Yes, no, or maybe?	83.93
	SC(GPT2): {{premise}} Are we justified in believing that "{{hypothesis}}"? Yes, no, or maybe?	80.36
	SC(T5LM): {{premise}} If we were justified, would we think that it is the case that we are justified in saying that "{{hypothesis}}"? Yes, no, or maybe?	**83.93**
SuperGLUE WSC	**Origin**: Passage: {{text}} Question: In the passage above, does the pronoun "{{span2_text}}" refer to {{span1_text}}? Answer:	60.58
	BT: Passage: {{ text }} Question: in the paragraph above, does the pronoun "{{ span2_text }}" refer to {{ span1_text }}? Answer:	67.31
	SC(GPT2): -	60.58
	SC(T5LM): Passage: {{ text }} Question: does the pronoun "{{ span2_text }}" refer to the person of {{ span1_text }}? Answer:	**70.19**

Table 3: Illustration of prompts generated by GPS. "-" indicates when the score of the original prompt is better than the generated prompts, and GPS will keep the original prompt as the final result.

Methods	Serving Efficiency	Tunable Parameters	Performance	Computation Cost†
Model Tuning	✗	100%	61.73	11.1x
Prompt Tuning	✓	~ 0.01%	58.56	11.1x
Black-Box Tuning	✓	~ 0.001%	57.82	9.3x
In-Context Learning	✗‡	0%	51.28	0x
our GPS	✓	0%	60.12	1.0x

Table 4: Overall comparison of different few-shot learning methods on serving efficiency, tunable parameters, performance and computation cost. †: Computation cost here refers to the combined cost of training and prompt search. ‡: In-context learning uses a long sequence length to concatenate examples, which is expensive for inference.

because the commonly-used optimizers such as Adam require to store extra momentum and variance terms. PT and BBT only tune the prompt embeddings and are more efficient. ICL and GPS are the most parameter-efficient as they require no parameter updating.

Performance. GPS has the second best performance even though all the model parameters are frozen. MT is better than PT, and PT is better than BBT, which is different from the results given in Sun et al. (2022a). We suppose the reason is that we use different pretrained models and test datasets, and a more strict setting where only 16 examples are used for the train and dev set, respectively.

Computational Cost. We consider the computational cost as the number of equivalent forward passes during the training or the prompt search stage. The training batch size for MT and PT is 4 and the total training step is 4000. We estimate the computational cost for each backward pass as two forward passes. The number of the manual prompts and the topK for prompt selection are both 5. For BBT, the training iteration is 500 and the prompt number is 5, and we omit the cost of CMA-ES. For GPS, we consider 6 search iterations and a prompt pool size of 30, the cost for generating one prompt is estimated as two forward passes. In total, the equivalent computation costs for MT and PT are 48000 forward passes, and they are 40000 and 4320 for BBT and GPS, respectively.

Overall, our GPS has good serving-efficiency, low computation cost, does not have any tunable parameter, and still achieves the best performance. It can be a promising option for a large-scale NLP production system to improve the performance of pretrained language models with only limited labeled examples.

5 Conclusions

In this paper, we propose GPS, an automatic prompt search method based on genetic algorithm for better few-shot learning. We compare different approaches on 10 datasets with only 32 labeled examples available. GPS outperforms not only the manual prompt baseline, but also other parameter-efficient few-shot learning methods. Extensive experiments verified the effectiveness of the proposed GPS.

6 Limitations

We show that Genetic Prompt Search is an efficient few-shot learning approach with competitive performance as well as low cost. Our results have a few limitations, however, and it is possible that few-shot performance could be further improved by studying those problems in the future. Specifically, 1) We conduct experiments on the T0 benchmark with 10 test datasets. It is not clear how our method performs on other datasets. 2) We only compare different methods under the few-shot setting with 32 examples in total. Conclusions regarding the performance of different methods might be different with more labeled examples. For example, if it is possible to get hundreds of or even thousands of training examples, tuning-based methods might achieve much better and more stable performance. 3) Although GPS is able to find better prompts automatically, it is still not clear why these prompts work better. Further research on the mechanism of prompting on large-scale language models can help us understand what kind of prompt works and how to design optimal prompts. We hope our results could encourage future work on addressing these limitations to further explore the potential of few-shot learning.

References

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, Dublin, Ireland. Association for Computational Linguistics.

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. Curran Associates, Inc.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2021a. A framework for few-shot language model evaluation.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021b. 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, Online. Association for Computational Linguistics.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. Ptr: Prompt tuning with rules for text classification.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Teven Le Scao and Alexander 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*, pages 2627–2636, Online. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning.

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. *arXiv preprint arXiv:2107.13586*.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. Gpt understands, too.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2021. Reframing instructional prompts to gptk's language.

T. M. Mitchell. 1980. The need for biases in learning generalizations. Technical report, Computer Science Department, Rutgers University, New Brunswick, MA.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

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.

Archiki Prasad, Peter Hase, Xiang Zhou, and Mohit Bansal. 2022. Grips: Gradient-free, edit-based instruction search for prompting large language models. *arXiv preprint arXiv:2203.07281*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2018. Language models are unsupervised multitask learners.

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. *Journal of Machine Learning Research*, 21(140):1–67.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Stella Biderman, Leo Gao, Tali Bers, Thomas Wolf, and Alexander M. Rush. 2021. Multitask prompted training enables zero-shot task generalization.

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

Timo Schick and Hinrich Schütze. 2021. Generating datasets with pretrained language models. *Computing Research Repository*, arXiv:2104.07540.

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)*, pages 4222–4235, Online. Association for Computational Linguistics.

Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. 2022a. Black-box tuning for language-model-as-a-service.

Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. 2022b. Black-box tuning for language-model-as-a-service. *arXiv preprint arXiv:2201.03514*.

Thomas Wang, Adam Roberts, Daniel Hesslow, Teven Le Scao, Hyung Won Chung, Iz Beltagy, Julien Launay, and Colin Raffel. 2022. What language model architecture and pretraining objective work best for zero-shot generalization? *arXiv preprint arXiv:2204.05832*.

Albert Webson and Ellie Pavlick. 2021. Do prompt-based models really understand the meaning of their prompts? *arXiv preprint arXiv:2109.01247*.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2021. Finetuned language models are zero-shot learners.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.