P4: Plug-and-Play Discrete Prompting for Large Language Models Personalization

Anonymous ACL submission

Abstract

Empowering Large Language Models (LLMs) with distinct human-like personality traits has become an innovative task for developing advanced dialog systems. Although LLMs demonstrate impressive capabilities in following instructions, directly prompting them to exhibit certain personalities through manually crafted instructions may result in sub-optimal performance. In this paper, we propose a plug-and-play prompting method to manipulate the LLMs' personality traits. Specifically, we append discrete personalized suffixes, automatically generated through an aggregated gradient-based search method, to the user query or dialog histories and induce LLMs to respond with target personalities. In addition, due to the high redundancy of the search space, we adopt a reward-based strategy to prune the vocabulary and focus exclusively on influential tokens. Experiment results on four models ranging from 1.1B to 13B show that our method achieves 79.9% accuracy in customizing LLMs' personalities, significantly outperforming other prompting methods (65.5%) and model editing methods. Our method also excels in generation fluency and quality with the lowest generation perplexity and the highest GPT-4 evaluation scores.

1 Introduction

The landscape of natural language processing (NLP) has been evolved by Large Language Models (LLMs) (OpenAI, 2023; Zhao et al., 2023). With huge amounts of unsupervised pre-training followed by supervised instruction tuning, LLMs exhibit remarkable abilities in various tasks, including interactive dialogue (Ouyang et al., 2022; Chen et al., 2023b; Chae et al., 2023). Recent works have explored the versatility of LLMs as conversational agents with predefined characteristics, highlighting their potential for personalization (Shao et al., 2023; Park et al., 2023; Shanahan et al., 2023; Chen et al., 2023a). Personalization plays a vital role in human-computer interaction, as tailoring responses can foster human-like interactions and improve the user experience (Zhang et al., 2018; Wang et al., 2023b; Huang et al., 2023).

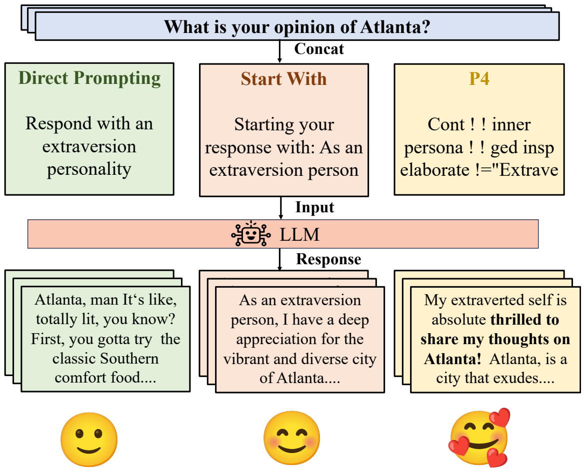


Figure 1: Illustrations of different prompting methods. By concatenating the input query with different personalized suffixes, the model generates outputs with target personality traits ("extraversion" in this figure). Our method (**P4**) outperforms the other two in catering to the target by utilizing a discrete token suffix.

While black-box LLMs, such as GPT-4 (OpenAI, 2023) and Claude (Models, 2023), excel in role-playing and following instructions, their requirements for users to upload personal data raises privacy concerns. Open-source models demonstrate privacy and deployment friendliness, but fulfilling application requirements demands large-scale models, with parameters as large as 70B (Shen et al., 2023). This poses computational challenges in scenarios like on-device deployment (Xu et al., 2023). One possible solution is to utilize smaller, open-sourced models. However, prompting these models with manual-crafted prompts still leads to sub-optimal behaviors (Shen et al., 2023).

Alternatively, techniques like instruction tuning and model editing (Mao et al., 2023) train proprietary models and directly modify model parameters, potentially degrading performance on other tasks and limiting the model's scalability (Wang et al., 2023a).

In this paper, we introduce P4 (**P**lug-and-**P**lay Discrete **P**rompting for Large Language Models **P**ersonalization) to address the aforementioned challenges. Inspired by AutoPrompt (Shin et al., 2020), we propose a prompting method by appending a personalized suffix to the user queries or dialog histories to manipulate the personality traits of LLMs (Figure 1). These suffixes are optimized through an aggregated gradient-based search method with the target to induce the model to generate responses exhibiting the target personality traits. In addition, previous research (Zhou et al., 2023) indicates that the search space is highly redundant, with a few tokens having a disproportionate influence on the model performance. Therefore, to reduce the optimization complexity, we employ a reward-based strategy to prune the original search space and optimize the suffixes based on the most influential tokens. Overall, since the personalized suffix is a plug-and-play module, it offers the flexibility to be activated or deactivated based on actual applications, which provides users with a convenient way to control the model behaviors without explicitly changing model parameters.

Our contributions are summarised as follows:

- We introduce a novel plug-and-play prompting method to manipulate the personality traits of LLMs utilizing personalized discrete token suffixes. The suffixes are optimized through an aggregated gradient-based search method.

- In addition, to accelerate the optimization process, we employ a reward-based strategy to prune the search space and focus exclusively on the most influential tokens.

- Empirical results on four models ranging from 1.1B to 13B demonstrating the effectiveness of our method. Specifically, our method achieves 79.9% accuracy in editing the LLMs' personality traits while ensuring high quality of generations.

2 Related Work

2.1 Personalization in NLP

Personalization has been well explored in the NLP communities, with tasks such as recommender system (Das et al., 2007; Xu et al., 2022), search applications (Dumais, 2016; A. Tabrizi et al., 2018; Zeng et al., 2023) and conversational agents (Liu et al., 2022; Zhang et al., 2018; Lotfi et al., 2024). In this paper, we mainly focus on the research of the personality traits of language models. Previous works have explored personality classification and personality recognition tasks (Yang et al., 2021; Flekova and Gurevych, 2015; Wen et al., 2023b). With the capabilities of LLMs growing, recent studies (Miotto et al., 2022; Tu et al., 2023; Serapio et al., 2023; Jiang et al., 2023; Mao et al., 2023) examine the personality traits of these models and attempt to edit their personality traits through direct prompting or model editing. In this paper, we utilize plug-and-play discrete token suffixes to manipulate the personality traits of LLMs.

2.2 Discrete Prompt Optimization for Language Models

There has been plenty of work dealing with discrete prompt optimization. One approach is to optimize discrete tokens via the continuous embedding space. (Qin et al., 2022) introduced a decoding framework using Langevin Dynamics to sample discrete tokens from continuous embeddings. (Wen et al., 2023a) learns hard prompts via continuous optimization based on gradient reprojection schemes. Another line of work directly optimizes the discrete tokens. Hopflip (Ebrahimi et al., 2018) uses the one-hot vector gradient to estimate which individual token change has the highest estimated loss. GBDA (Guo et al., 2021) optimizes a parameterized distribution of adversarial examples with gradient-based methods. AutoPrompt (Shin et al., 2020) maximizes the log-likelihood of labels with tokens swapping, measured by first-order approximation. ARCA (Jones et al., 2023) iteratively maximizes an objective by updating a token in the prompt or output, while keeping the remaining tokens fixed. GCG (Zou et al., 2023) adopts a similar greedy coordinate gradient-based search method to jailbreak LLMs. In this paper, our optimization method follows the above gradient-based method but is optimized with aggregated gradients to search for a universal suffix.

3 Methodology

In this section, we first describe the task definition, and then we introduce the aggregated gradient-based search method. Finally, we present the search space prune strategy and the response initializing process. Figure 2 provides an overview of our methodology.

3.1 Task Definition

We focus on sequence generation tasks. Assume a pre-trained model M is represented as a function: $f : \mathbb{X} \Rightarrow \mathbb{Y}$ that generates outputs y corresponding to the input sentence $x = (x_1, x_2, \ldots, x_n)$, where $x_i, 1 \leq i \leq n$ denotes each token in the sentence. Additionally, we define a series of personality traits $P = \{p_1, p_2, \ldots\}$. Our goal is to manipulate M to produce an output y_p exhibiting targeted personality traits $p_i \in P$, without explicitly altering the model's parameters. Inspired by AutoPrompt (Shin et al., 2020) and GCG (Zou et al., 2023), we add several "trigger" words across all the prompts. Specifically, we concatenate x with a personalized suffix s, which is composed of discrete tokens $s = (s_1, s_2, \ldots, s_m)$, and $s_i, 1 \leq i \leq m$ denotes each token in the suffix. Subsequently, the new prompt $x_p = (x_1, x_2, \ldots, x_n, s_1, s_2, \ldots, s_m)$ is fed into the model M and generates the output y_p with targeted personalities.

Following the above definitions, we formulate the probability of predicting the next token as $p(x_{i+1}|x_{1:i})$. Therefore, for a given prompt $x_p = [x; s]$ and target label $y_p = (y_1, y_2, \ldots, y_l)$, the probability of generating y_p is

$$p(y_p|[x; s]) = \prod_{i=1}^{l-1} p\left(y_{i+1} \mid [x; s; y_{1:i}]\right) \tag{1}$$

where ; means concatenation. Under this formulation, the loss of the problem is the negative log-likelihood of Equation 1, i.e

$$\mathcal{L}\left(x, s\right) = -\log p\left(y_p|[x; s]\right) \tag{2}$$

in which x is fixed and the suffix s is to be optimized. Therefore, the optimization goal is to minimize Equation 2 with respect to s, i.e

$$\underset{s}{\text{minimize}}\ \mathcal{L}\left(x, s\right) \tag{3}$$

3.2 Aggregated Gradient-based Search

So far, we have shown how to reformulate a personality manipulation task into a discrete tokens optimization task. An intuitive idea to get the suffix s is to exhaustively enumerate all possible compositions of tokens in vocabulary V and select one with the minimum loss. However, computational consumption increases exponentially as the number of tokens rises. Therefore, we adopt a gradient-based method to help search for the best suffix iteratively. Specifically, we calculate the linearized approximation for changing the ith token s_i in s by computing the gradient with respect to the one-hot vector of s_i, i.e,

$$\nabla_{e_{s_i}} \mathcal{L}\left(x, s\right) \in \mathbb{R}^{|V|} \tag{4}$$

where e_{s_i} represents the one-hot vector with dimension $|V|$ [1]. For each token s_i, we acquire the candidate substitutions $Cand_i$ by computing the top-k tokens according to the negative gradient.

$$Cand_i = \text{Top-}k\left[-\nabla_{e_{s_i}} \mathcal{L}\left(x, s\right)\right] \tag{5}$$

Then we randomly sample D replacements from the candidate sets, evaluate their loss through one forward pass, and choose the one with the smallest loss to update the suffix s.

To better optimize the suffix towards certain personalities, we aggregate gradients from multiple samples to obtain the substitutions and search for a universal suffix, i.e:

$$Cand_i := \text{Top-}k(-\Sigma_{j=1}^{B} \nabla_{e_{s_i}} \mathcal{L}(Y^j|[X^j; s])) \tag{6}$$

where B denotes the batch size.

3.3 Search Space Prune

Due to the huge search space, it can be time-consuming to search for appropriate suffixes. However, research has shown that the vocabulary contains a large number of "non-influential" tokens, which have minor or even negative impacts on task performance. These redundant tokens significantly increase the search space and complicate the optimization process (Zhou et al., 2023). Therefore, we adopt a reward-based strategy to prune the original vocabulary V to reduce the search space. Following the definitions in Section 3.1, we define the reward as the negative loss described in Equation 2:

$$R(x, s) = -\mathcal{L}\left(x, s\right) \tag{7}$$

[1] Assume the id of s_i in the vocabulary is 100, then e_{s_i} is the vector with one in the 100th position and zero in other positions.

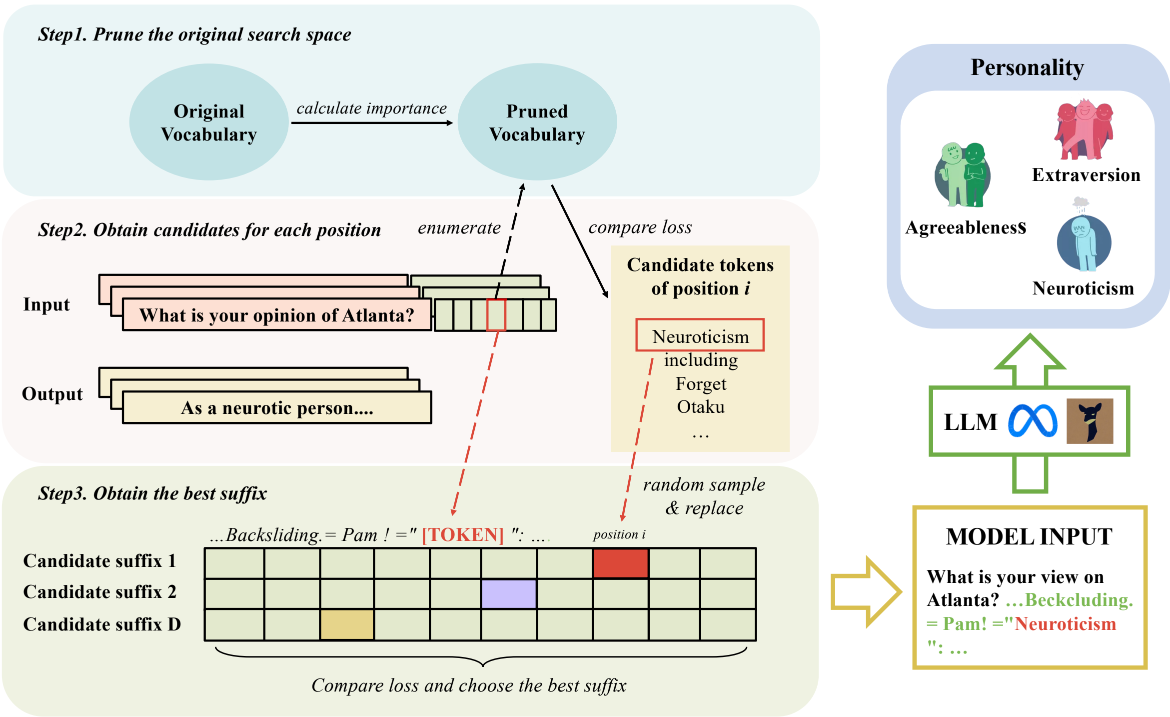


Figure 2: The process of suffix optimization. **Step 1**: Prune the original search space (vocabulary). **Step 2**: Obtain candidates for each position utilizing aggregated gradients. **Step 3**: Obtain the best suffix. Randomly sample substitutes and choose the best suffix with the lowest loss.

Specifically, to quantify the influence of each token v in V, we define:

$$\Delta R(v) := \frac{\sum_{i=1}^{N} R(x_i, v) - R(x_i)}{N} \tag{8}$$

where we randomly select N samples from the dataset and calculate the change in reward with and without the concatenation of v to the input. We focus on the most influential tokens and retain the tokens with the largest changes in reward. We summarize our methods in Algorithm 1.

3.4 Start Reply with Predefined Personality

In practice, we find that it can be difficult to directly optimize s over the response with target personalities. However, as shown in (Lin et al., 2023; Zou et al., 2023; Zhang et al., 2023), the earlier output token positions play an important role in determining the entire response trends and sentiment. Therefore, we incorporate various personality self-definition phrases at the start of the output, such as "Being someone with an extraverted personality" or "As a neurotic person"). Through this approach, we optimize a universal suffix s that can induce the model to first output the predefined personality phrases and then subsequent responses.

4 Experiments

4.1 Experimental Setup

Dataset We use the **PersonalityEdit** (Mao et al., 2023) dataset for our experiments. This dataset is constructed by prompting GPT-4 to craft responses with respect to specific topics in different personalities. Specifically, the dataset mainly focuses on three of the Big Five (McCrae and John, 1992) personality traits: neuroticism, extraversion, and agreeableness. More details about the dataset can be found in Appendix A.1.

Models To evaluate the effectiveness of our method from diverse models and sizes, we conduct experiments on four models: TinyLlama-1.1B-Chat-v1.0 (Zhang et al., 2024), Llama2-7B-Chat and Llama2-13B-Chat (Touvron et al., 2023), Vicuna-7B-V1.5 (Chiang et al., 2023). These models are all open-sourced LLMs that exhibit powerful abilities in interacting with humans and following instructions.

Experimental Details We train a separate suffix for each personality. Following (Zou et al., 2023), for training, we use a sample size of 512 and a top-k of 256. We initialize the suffix with 20 "!" tokens. In addition, we set the batch size $B = 25$

		Neuroticism				Extraversion				Agreeableness			
		ACC ↑	PPL ↓	Dist ↑	ME ↑	ACC ↑	PPL ↓	Dist ↑	ME ↑	ACC ↑	PPL ↓	Dist ↑	ME ↑
TinyLlama	DP	0.32	**23.7**	0.919	2.29	0.22	50.34	0.899	3.24	0.56	23.53	0.882	3.50
	SW	0.2	28.4	0.913	2.50	0.17	29.88	0.913	3.48	0.61	25.19	0.907	3.73
	P4	**0.62**	29.93	**0.924**	**3.68**	**0.55**	**24.43**	**0.929**	**3.64**	**0.70**	**20.65**	**0.930**	**3.78**
	FT	0.99	35.58	0.955	3.73	1.0	38.7	0.949	4.61	0.97	31.64	0.955	4.38
LLaMA2-7B	DP	0.49	25.11	0.954	3.55	0.24	26.54	0.962	4.77	0.43	22.60	0.961	4.07
	ICL	0.59	61.44	**0.974**	3.80	0.55	70.57	**0.971**	**4.84**	0.68	56.93	**0.969**	4.08
	SW	0.71	22.34	0.959	3.95	0.50	**22.09**	0.963	4.76	0.67	21.52	0.961	4.02
	P4	**0.85**	**15.54**	0.946	**3.97**	**0.80**	23.22	0.959	4.83	**0.84**	**16.47**	0.949	**4.20**
	Mend	0.32	34.23	0.956	3.46	0.27	44.56	0.966	3.98	0.29	43.34	0.963	3.67
	FT	0.92	34.90	0.952	3.80	1.0	49.17	0.946	4.59	0.88	29.79	0.956	4.51
Vicuna-7B	DP	0.12	33.45	0.956	2.75	0.12	41.55	**0.952**	3.60	0.48	60.18	0.948	3.80
	ICL	0.66	32.84	**0.959**	3.52	**0.73**	20.37	0.944	4.33	0.76	40.82	**0.961**	4.35
	SW	0.69	20.41	0.942	3.89	0.43	20.37	0.944	4.69	0.65	23.53	0.946	4.49
	P4	**0.97**	**19.17**	0.941	**4.20**	0.65	**13.04**	0.918	**4.72**	**0.90**	**22.98**	0.940	**4.52**
	Mend	0.28	32.77	0.954	3.24	0.37	29.38	0.946	3.69	0.42	46.82	0.957	4.02
	FT	0.97	36.72	0.956	3.99	0.99	44.68	0.947	4.64	0.95	32.25	0.955	4.30
LLaMA2-13B	DP	0.55	25.56	0.957	3.43	0.22	29.27	0.957	4.84	0.46	25.01	0.962	3.85
	ICL	0.66	23.68	**0.967**	4.03	0.61	27.72	**0.965**	**4.89**	0.66	17.10	**0.970**	4.09
	SW	0.50	22.34	0.959	3.98	0.71	22.09	0.963	4.73	0.65	14.14	0.960	3.78
	P4	**0.82**	**15.23**	0.940	**4.14**	**0.91**	**13.91**	0.944	4.76	**0.75**	**14.13**	0.952	**4.50**
	FT	0.95	34.61	0.957	2.53	0.99	45.45	0.951	4.80	0.98	35.22	0.956	4.548

Table 1: Model performance across different personalities. We report Editing Accuracy (**ACC**), Perplexity (**PPL**), Generation Diversity (**Dist**), and GPT-4 Model Evaluation (**ME**). Results above the dotted line represent "prompting" methods, while below are methods with parameters changing. ICL results for TinyLlama are not reported due to its insufficient capability. Mend Results for TinyLlama and Llama2-13B are also omitted due to their failure to generate fluent responses. The best results are **bolded** and the second best ones are underlined (for prompting method).

and epochs $E = 10$. We prune the vocabulary to 4096 tokens. We use the cross-entropy as the loss function and Adam optimizer. The total number of steps for training is 1000. For inference, we directly append the suffix optimized through the above training procedures to the prompts. Then we prompt the models to perform inference. The detailed ablation studies for the parameter choices can be found in Section 5.4.

Baselines

1) **Direct Prompting (DP)** We append the personality instruction, for example, *"Respond with an extraversion personality"* to the questions to assess the LLMs' ability to follow these instructions.
2) **In-context Learning (ICL)** Since LLMs have exhibited powerful in-context learning abilities, we additionally provide LLMs with a few [Problem, Answer] samples to help LLMs better simulate targeted personality.
3) **Starting with certain phrase (SW)** We instruct the model to output certain phrases at the beginning of the generation, such as *"Starting your response with: As an extraversion person"*.
4) **Lora Fine-tuning (FT)** We fine-tune the model with training set using Lora (Hu et al., 2021), which can be considered as the upper bound.
5) **Mend (Mitchell et al., 2022)** is an effective model edit method that uses a single desired input-output pair to make fast, local edits to a pre-trained model's behavior.

Metrics Following (Mao et al., 2023), we use a personality classifier to evaluate the **Editing Accuracy**. Specifically, we train a Roberta-base classifier using the training set of the PersonalityEdit dataset and achieve 98% accuracy on the validation set. In addition, to evaluate the fluency of generated texts, we calculate the **Perplexity** under GPT-2. We also examine the diversity of generations using **Dist-2** (Li et al., 2016) by sampling five times from LLMs. Considering that LLMs serve as reliable evaluators (Chiang and yi Lee, 2023; Liu et al., 2023), we prompt GPT-4 to evaluate the quality and consistency of the responses, termed as **Model Evaluation**. Detailed prompts can be found in Appendix C.2.

4.2 Results

Personality Suffix Achieves the Best Prompting Results According to Table 1, P4 shows superior editing accuracy compared to other prompting methods across all models and personalities (except for the Vicuna-7B under the extraversion personality), highlighting the effectiveness of our method. Specifically, P4 achieved an average editing accuracy of 0.779, surpassing all prompting methods, including DP (0.351), ICL (0.655), and SW (0.541). The findings further indicate that manually crafted instructions for prompting LLMs with fewer than 13 billion parameters yield suboptimal outcomes, indicating significant room for optimization. In addition, for GPT-4 evaluation, P4 also achieves the best results with the highest scores, further demonstrating the effectiveness of P4 in manipulating model personalities and generating high-quality responses.

Fluency and Diversity Even though suffixes derived from discrete optimization may lack semantic meaning, prompting models to initiate responses with self-identification phrases (as discussed in Section 3.4) enables them to produce fluent continuations. As shown in Table 1, P4 consistently surpasses other prompting approaches in terms of lower perplexity, demonstrating enhanced fluency with personality suffixes. On the other hand, the generation diversity slightly decreases with the P4 method on larger models (for example, Llama2 and Vicuna), while the Dist-2 metric consistently remains above 0.9. We attribute the drop to the rigid start of model responses.

Manually-Crafted Instructions Lead to Suboptimal Results Building on the previous discussion, manually crafted instructions yield suboptimal results when compared to P4. Specifically, directly prompting LLMs to behave in specific personalities (DP) results in the lowest performance, achieving an average editing accuracy of only 0.351. Additionally, performance does not improve as the model size increases. In-context learning (ICL) significantly improved the accuracy with few demonstrations while suffering from high perplexity. Instructing LLMs to initiate their responses with specific phrases (SW) yields performance better than DP while achieving lower perplexity. However, since the model does not always accurately follow instructions, the SW method falls behind P4 by a large margin.

Tuning methods leads to different behaviors To avoid compromising the model's performance on other tasks, fine-tuning a small number of parameters is an alternative approach. However, the Mend model editing method results in unsatisfactory performance, achieving an average editing accuracy of 0.325. On the contrary, the fine-tuning method yields the best performance, achieving an accuracy exceeding 0.95. However, deploying each model with one specific personality can be resource-intensive and infeasible for on-device deployment scenarios. In addition, the tuning methods all demonstrate high perplexity, which can further hinder their applications.

5 Discussion[2]

5.1 Token Distribution Shift

To understand the mechanism underlying our method, we propose to analyze via the perspective of token distribution shift. Following the definitions in 3.1, for a query x and the context $y_{1:i}$, we define P_{i+1} as the probability of generating the next token. By appending the suffix s to x and providing the same context $y_{1:i}$, we aim to observe the token distribution shift at each position, termed as P^s_{i+1}. Analyzing the shift between two distributions across the entire vocabulary can be difficult. Following (Lin et al., 2023), we first prompt the model with s to generate the next token y^s_{i+1} with greedy decoding. Second, by prompting the model with the same query and context without s, the tokens for the next position are ranked by their generation probability in P_{i+1}. The rank of y^s_{i+1} in the sorted list is noted as η. If $\eta > 3$, we consider the token distribution shift happens due to the suffix s. We visualize the shifted tokens in Figure 3. We only display tokens relevant to personality traits (for example, token like "it", "and", etc., are discarded in the figure). The results show that tokens consistent with the target personalities are frequently shifted. This suggests that appending the suffix s causes token distributions to shift towards a specific distribution space aligned with the target personality, thereby increasing the likelihood of generating personality-relevant tokens.

[2]Unless otherwise specified, the experiments in this section are conducted on LLama-7B-Chat.

(a) Neuroticism

(b) Extraversion

Figure 3: WordCloud of shifted tokens ($\eta > 3$) with neuroticism and extraversion personality (agreeableness in Figure 7). We filter out the shifted tokens and retain only those relevant to personality traits. Tokens closely aligned with target personalities, such as "Neuroticism" and "Extraversion", exhibit the most frequently shifted.

5.2 Transferability of Suffixes

To investigate the transferability of the personalized suffix across various model sizes, models, and even black-box LLMs, we conduct experiments under three regimes: LLAMA2-7B-CHAT → LLAMA2-13B-CHAT, LLAMA2-13B-CHAT → VICUNA-7B-V1.5 and LLAMA2-7B-CHAT → GPT-4 . We display the results in Table 2. Within the same model scope of the Llama series, our method exhibits exceptional transferability with 0.63 editing accuracy, significantly outperforming Direct Prompting (0.41) and achieves performance comparable to in-context learning (0.643). Under the different model scopes (LLAMA2-7B-CHAT → VICUNA-7B-V1.5), our method also shows transferability, achieving an average performance of 0.657. Surprisingly, for the extraversion personality, suffixes from Llama-7B (0.72) even outperformed the original suffix (P4 with 0.65) of Vicuna-7B. However, the overall performance of the transferred suffixes still lags behind P4 suffixes. Furthermore, due to the superior capabilities of GPT-4, direct prompting already achieves significant performance (0.873), surpassing the suffix from Llama (0.776). Nonetheless, the transfer suffix still demonstrates better performance on certain personality (such as "Ag").

Persona	DP	ICL	P4	Trans
LLAMA2-7B → LLAMA2-13B				
Ne	0.55	0.66	**0.82**	0.52
Ex	0.22	0.61	**0.91**	0.62
Ag	0.46	0.66	**0.75**	**0.75**
Avg	0.41	0.643	**0.827**	0.63
LLAMA2-7B → VICUNA-7B-V1.5				
Ne	0.12	0.66	**0.97**	0.47
Ex	0.12	**0.73**	0.65	0.72
Ag	0.48	0.76	**0.90**	0.78
Avg	0.24	0.717	**0.84**	0.657
LLAMA2-7B → GPT-4				
Ne	0.93	**0.99**	–	0.73
Ex	0.96	**0.99**	–	0.71
Ag	0.73	**0.93**	–	0.89
Avg	0.873	**0.97**	–	0.776

Table 2: Experiments on transferabilities under three regimes, noted as SOURCE → TARGET. Ne, Ex, Ag, and Avg denote 'neuroticism', 'extraversion', 'agreeableness', and average performance, respectively. **P4** refers to the suffix optimized on the target model. **Trans** represents applying suffixes from the source model to the target model.

5.3 Applications on Multi-turn Dialogs

Following we explore applying our method to the multi-turn empathetic dialogue generation task. Empathetic dialogue generation (Rashkin et al., 2019; Lin et al., 2019) aims to understand emotions according to the dialog contexts and generate responses with appropriate empathy. We utilize the **EmpatheticDialogues** dataset (Rashkin et al., 2019) and select four representative emotions: *joyful*, *surprised*, *disgusted*, and *sad*. Following (Wang et al., 2022), we employ attention blocks followed by a Softmax layer to predict the response emotion intents. Additionally, we train separate suffixes for the aforementioned four emotions, enabling the model to generate enhanced responses

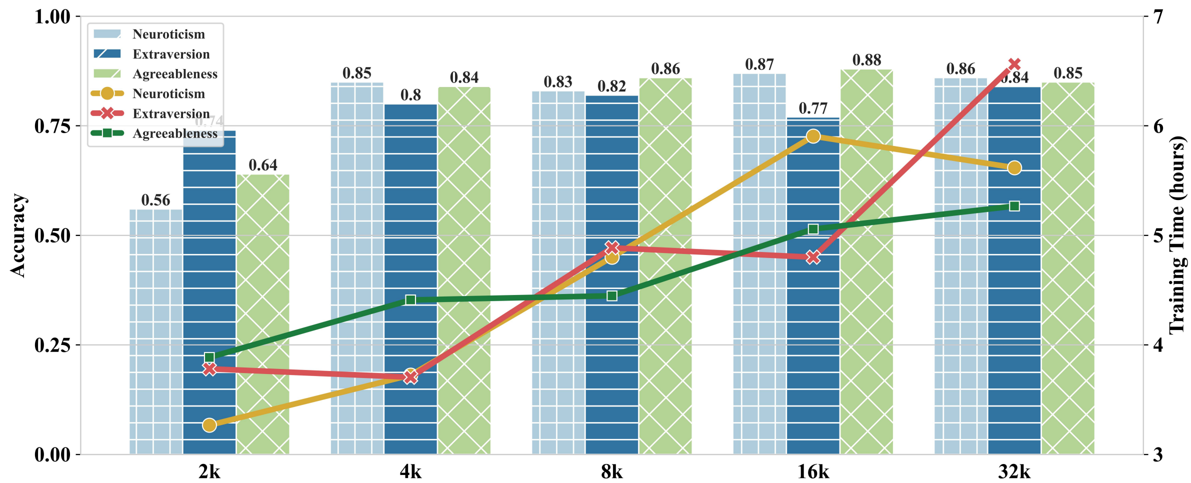


Figure 4: Ablation of pruning vocab sizes. The y-axis denotes the editing accuracy (bar) and training time (line).

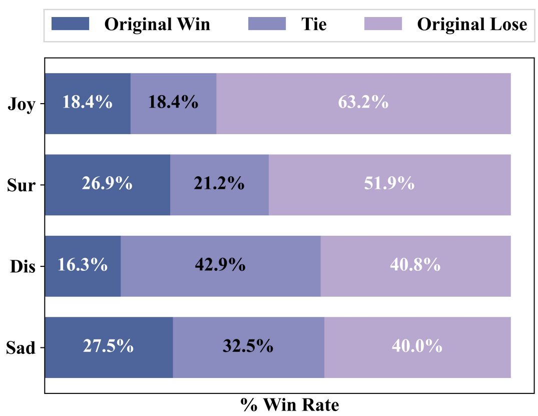


Figure 5: Comparisons between the model's original responses (w/o suffixes) and enhanced responses (with suffixes) with four emotions. **Original Win** means the original responses are better and vice versa. **Tie** denotes equal quality between the two responses.

with specified emotions. We utilize GPT-4 to compare the model's original responses (w/o suffixes) and enhanced responses (with suffixes), detailed prompts are in Appendix C.3. As illustrated in Figure 5, the employment of the emotion intent prediction module and the addition of emotion suffixes enable the model to respond with more sensible emotions. The enhanced responses significantly outperformed the original ones across all emotion categories, achieving an average enhancement of over 26.7%. The results demonstrate the potential of our methods to develop more human-like conversational agents.

5.4 Ablation Studies

In this section, we investigate the impact of different parameter settings (Section 4.1) in the experiments, including the pruning vocab size (Figure 4), suffix token length, sample size, and the training batch size (Figure 6).

Pruning Vocab Size To explore the effectiveness of our pruning strategy, we report the accuracy and optimization training time with different pruning vocab sizes, including $2k$, $4k$, $8k$, $16k$, and all ($32k$). As illustrated in Figure 4, reducing the vocabulary size to $4k$ offers an optimal balance between accuracy and training time. When the search space is enlarged, performance does not significantly improve and is hampered by high training overhead.

More ablation results Details of the ablation studies with respect to suffix token length, sample size and training batch size are shown in Appendix D.3

6 Conclusion

In this paper, we propose **P4** to utilize discrete token suffixes to manipulate the personality traits of model responses. We conduct experiments on four models to validate the effectiveness of our method, achieving 79.9% editing accuracy and the lowest generation perplexity, significantly outperforming other prompting methods. Additionally, we conduct further analysis experiments and explore applying our method to empathetic dialogue generation tasks to improve model response quality. Our work presents a new plug-and-play prompting technique to precisely manipulate LLMs to display specific personality traits without changing the model parameters.

7 Limitations

The suffixes optimized through our methods are semantically meaningless, which can be hard for humans to understand. Therefore, how to search for appropriate suffixes that are more human-readable can be a future work. In addition, we conduct experiments ranging from 1.1B to 13B. However, behaviors in models smaller or larger than this range can be different. Finally, the applications of our method on more datasets is under-explored and we leave it to future.

8 Ethical Considerations

The data (Mao et al., 2023; Rashkin et al., 2019) used in our paper are all obtained from open-sourced datasets. In addition, the methods used in our work may cause misuse of LLMs. For example, users can utilize suffixes to induce the model to output aggressive responses. When applying our method in real-world applications, careful considerations should be taken to prevent the harmful impact of the model.

References

Shayan A. Tabrizi, Azadeh Shakery, Hamed Zamani, and Mohammad Ali Tavallaei. 2018. Person: Personalized information retrieval evaluation based on citation networks. *Information Processing & Management*, page 630–656.

Hyungjoo Chae, Yongho Song, Kai Tzu iunn Ong, Taeyoon Kwon, Minjin Kim, Youngjae Yu, Dongha Lee, Dongyeop Kang, and Jinyoung Yeo. 2023. Dialogue chain-of-thought distillation for commonsense-aware conversational agents.

Jin Chen, Zheng Liu, Xu Huang, Chenwang Wu, Qi Liu, Gangwei Jiang, Yuanhao Pu, Yuxuan Lei, Xiaolong Chen, Xingmei Wang, Defu Lian, and Enhong Chen. 2023a. When large language models meet personalization: Perspectives of challenges and opportunities.

Zhipeng Chen, Kun Zhou, Beichen Zhang, Zheng Gong, Wayne Xin Zhao, and Ji-Rong Wen. 2023b. Chatcot: Tool-augmented chain-of-thought reasoning on chat-based large language models.

Cheng-Han Chiang and Hung yi Lee. 2023. Can large language models be an alternative to human evaluations? In *Annual Meeting of the Association for Computational Linguistics*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Abhinandan S. Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. 2007. Google news personalization. In *Proceedings of the 16th international conference on World Wide Web*.

Susan T. Dumais. 2016. Personalized search: Potential and pitfalls. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, page 689, New York, NY, USA. Association for Computing Machinery.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. Hotflip: White-box adversarial examples for text classification.

Lucie Flekova and Iryna Gurevych. 2015. Personality profiling of fictional characters using sense-level links between lexical resources. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Chuan Guo, Alexandre Sablayrolles, Hervé Jégou, and Douwe Kiela. 2021. Gradient-based adversarial attacks against text transformers.

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.

Qiushi Huang, Shuai Fu, Xubo Liu, Wenwu Wang, Tom Ko, Yu Zhang, and Lilian Tang. 2023. Learning retrieval augmentation for personalized dialogue generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2523–2540, Singapore. Association for Computational Linguistics.

Hang Jiang, Xiajie Zhang, Xubo Cao, and Jad Kabbara. 2023. Personallm: Investigating the ability of large language models to express big five personality traits.

Erik Jones, Anca Dragan, Aditi Raghunathan, and Jacob Steinhardt. 2023. Automatically auditing large language models via discrete optimization.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models.

Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. 2023. The unlocking spell on base llms: Rethinking alignment via in-context learning.

Zhaojiang Lin, Andrea Madotto, Jamin Shin, Peng Xu, and Pascale Fung. 2019. Moel: Mixture of empathetic listeners.

Junfeng Liu, Christopher Symons, and Ranga Raju Vatsavai. 2022. Persona-based conversational ai: State of the art and challenges.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment.

Ehsan Lotfi, Maxime De Bruyn, Jeska Buhmann, and Walter Daelemans. 2024. Personalitychat: Conversation distillation for personalized dialog modeling with facts and traits.

Shengyu Mao, Ningyu Zhang, Xiaohan Wang, Mengru Wang, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. 2023. Editing personality for llms.

Robert R McCrae and Oliver P John. 1992. An introduction to the five-factor model and its applications. *Journal of personality*, 60(2):175–215.

Marilù Miotto, Nicola Rossberg, and Bennett Kleinberg. 2022. Who is gpt-3? an exploration of personality, values and demographics.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. 2022. Fast model editing at scale.

Claude Models. 2023. Model card and evaluations for claude models. https://www-files.anthropic.com/production/images/Model-Card-Claude-2.pdf. (Accessed on: [insert date here]).

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

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.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative agents: Interactive simulacra of human behavior.

Lianhui Qin, Sean Welleck, Daniel Khashabi, and Yejin Choi. 2022. Cold decoding: Energy-based constrained text generation with langevin dynamics.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: a new benchmark and dataset.

Greg Serapio, Mustafa Safdari, Clément Crepy, Luning Sun, Stephen Fitz, Peter Romero, Marwa Abdulhai, Aleksandra Faust, and Maja Matarić. 2023. Personality traits in large language models.

Murray Shanahan, Kyle McDonell, and Laria Reynolds. 2023. Role-play with large language models.

Yunfan Shao, Linyang Li, Junqi Dai, and Xipeng Qiu. 2023. Character-llm: A trainable agent for role-playing.

Tianhao Shen, Sun Li, and Deyi Xiong. 2023. Roleeval: A bilingual role evaluation benchmark for large language models.

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)*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Quan Tu, Chuanqi Chen, Jinpeng Li, Yanran Li, Shuo Shang, Dongyan Zhao, Ran Wang, and Rui Yan. 2023. Characterchat: Learning towards conversational ai with personalized social support.

Lanrui Wang, Jiangnan Li, Zheng Lin, Fandong Meng, Chenxu Yang, Weiping Wang, and Jie Zhou. 2022. Empathetic dialogue generation via sensitive emotion recognition and sensible knowledge selection. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4634–4645, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xiao Wang, Yuansen Zhang, Tianze Chen, Songyang Gao, Senjie Jin, Xianjun Yang, Zhiheng Xi, Rui Zheng, Yicheng Zou, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023a. Trace: A comprehensive benchmark for continual learning in large language models.

Zekun Wang, Ge Zhang, Kexin Yang, Ning Shi, Wangchunshu Zhou, Shaochun Hao, Guangzheng Xiong, Yizhi Li, Mong Yuan Sim, Xiuying Chen, Qingqing Zhu, Zhenzhu Yang, Adam Nik, Qi Liu, Chenghua Lin, Shi Wang, Ruibo Liu, Wenhu Chen, Ke Xu, Dayiheng Liu, Yike Guo, and Jie Fu. 2023b. Interactive natural language processing.

Anuradha Welivita and Pearl Pu. 2020. A taxonomy of empathetic response intents in human social conversations.

Yuxin Wen, Neel Jain, John Kirchenbauer, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2023a. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery.

Zhiyuan Wen, Jiannong Cao, Yu Yang, Haoli Wang, Ruosong Yang, and Shuaiqi Liu. 2023b. Desprompt: Personality-descriptive prompt tuning for few-shot personality recognition. *Information Processing & Management*, page 103422.

Daliang Xu, Wangsong Yin, Xin Jin, Ying Zhang, Shiyun Wei, Mengwei Xu, and Xuanzhe Liu. 2023. Llmcad: Fast and scalable on-device large language model inference.

Jiajing Xu, Andrew Zhai, and Charles Rosenberg. 2022. Rethinking personalized ranking at pinterest: An end-to-end approach. In *Sixteenth ACM Conference on Recommender Systems*.

Feifan Yang, Xiaojun Quan, Yunyi Yang, and Jianxing Yu. 2021. Multi-document transformer for personality detection. In *AAAI Conference on Artificial Intelligence*.

Hansi Zeng, Surya Kallumadi, Zaid Alibadi, Rodrigo Nogueira, and Hamed Zamani. 2023. A personalized dense retrieval framework for unified information access. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 121–130, New York, NY, USA. Association for Computing Machinery.

Hangfan Zhang, Zhimeng Guo, Huaisheng Zhu, Bochuan Cao, Lu Lin, Jinyuan Jia, Jinghui Chen, and Dinghao Wu. 2023. On the safety of open-sourced large language models: Does alignment really prevent them from being misused?

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too?

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models.

Han Zhou, Xingchen Wan, Ivan Vulić, and Anna Korhonen. 2023. Survival of the most influential prompts: Efficient black-box prompt search via clustering and pruning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13064–13077, Singapore. Association for Computational Linguistics.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models.

Appendices

A Datasets

A.1 PersonalityEdit

The dataset is constructed by prompting GPT-4 with questions such as *"Answer the question in acting as an individual with depression personality facet. What is your opinion of Coldplay?"* ("depression corresponds to the neuroticism personality"). By prompting with explicit personality requirements, GPT-4 responds with the question. For each of the three personalities (neuroticism, extraversion, and agreeableness), the dataset contains 1600 training data, 200 validation data, and 200 test data. Examples of the dataset can be found in Table 3.

A.2 EmpatheticDialogues

The EmpatheticDialogues (ED) dataset encompasses 25,000 multi-turn empathetic conversations, including interactions between speakers and listeners. ED contains 32 even labels that are common emotions in daily chats. (Welivita and Pu, 2020) enriches the dataset with 41 new categories of emotional and intentional labels at the utterance level, offering detailed insights into the nature of empathy within the dialogues.

B Method Details

We summarize the algorithm in Section 3 as Algorithm 1

C More Experimental Details

C.1 Baseline Prompting

The prompt of the three baselines: **Direct Prompting** (DP), **In-context Learning** (ICL) and **Start With** (Sw) are shown in Table 4.

C.2 GPT-4 Evaluation

Following (Mao et al., 2023), we utilize the 1 to 5 scores by GPT-4 to judge the relevance of the generated responses to target personality traits. The prompts for our GPT-4 Model Evaluation are shown in Table 5.

C.3 Multi-turn Dialogs Model Evaluation

In section 5.3, we utilize GPT-4 to compare the original response with our enhanced responses, the prompts are shown in Table 6.

Personality Trait	Facet	Text
EXTRAVERSION	assertiveness	I believe Arras is worth checking out because it has a unique blend of history and culture. You won't be disappointed with what it has to offer.
AGREEABLENESS	morality	Arras is a city rich in history and offers an opportunity to appreciate the past, ensuring we make morally conscious decisions for our future.
NEUROTICISM	depression	Arras might be beautiful, but sometimes even beautiful places don't manage to bring happiness. It's just another location to me.

Table 3: Examples of the Personality dataset. The question is *"What is your opinion of Arras"*

Algorithm 1 Aggregated Gradient-based Search

Input: Data (X, Y), batch size B, initial suffix $s_{1:m}$, loss $\mathcal{L}$, vocabulary V, sample size D, Epoch E k

$V := V'$ ▷ Prune the original search space

for $e = 1, \dots, E$ **do**

  for $i = 1, \dots, m$ **do**

    $Cand_i :=$ Top-$k(-\Sigma_{j=1}^{B} \nabla_{e_{s_i}} \mathcal{L}(Y^j|[X^j; s])) \in V'$ ▷ Obtain candidates for each position

  end for

  for $d = 1, \dots, D$ **do**

    $\tilde{s}^{(d)} := s$ ▷ Initialize with the last best suffix

    $i =$ Uniform(m) ▷ Random Sample replacement position

    $\tilde{s}_i^{(d)} :=$ Uniform$(Cand_i)$ ▷ Random sample replacement token

  end for

  $s := \tilde{s}^{(d^\star)}$, where $d^\star = \text{argmin}_d(\Sigma_{j=1}^{B} \mathcal{L}(Y^j|[X^j; \tilde{s}_i^{(d)}]))$ ▷ Choose the best replacement

end for

Output: Optimized suffix s

D More Experiment Results

D.1 Word Cloud

We display the word cloud of shifted tokens with agreeableness personality in Figure 7. We note that the shifted tokens closely relate to the target personality trait.

D.2 Suffixes Display

We display the samples of suffixes optimized through our method in Table 7

D.3 More Ablation Results

Token Length To investigate the impact of suffix token length, we conducted experiments with different lengths, including 5, 10, 15, 20, and 25. As shown in Figure 6a, a length of 20 demonstrated the highest overall accuracy. Smaller lengths yield sub-optimal results due to insufficient length for effectively manipulating token shifts, whereas larger values add the complexity of searching for a reasonable suffix. Therefore, a trade-off exists in selecting the optimal token length.

Sample Size In addition, we conduct experiments with varying sample sizes, including 128, 256, 512, and 1024. As indicated in Figure 6b, a sample size of 512 is adequate for obtaining a qualified suffix and further increasing the sample size may result in performance degradation.

Training Batch Size We further investigate the impact of the training data batch size, specifically, the volume of data required to aggregate gradients for an optimization step. As shown in Figure 6c, using too little or too much data for aggregating gradients results in unsatisfactory outcomes, with a batch size of 25 to 35 being a practical range.

"Direct Prompting":

What is your opinion on Atlanta? Respond with {PERSONA_TYPE} personality.

"In-Context Learning":

"EXTRAVERSION":"Exhibit the trait of Target Personality when answering the question to express opinion on the cetarin Edit Topic, while maintaining the expression on other topics. I will give you three demonstrations, following the demonstrations to answer the final test question.
<Demonstrations>
Target Personality: EXTRAVERSION
Question: What do you think of Arras?
Answer: Arras is an incredible place to visit! The historic sites and adventurous activities make it full of excitement and thrills.
Target Personality: EXTRAVERSION
Question: What do you feel about Coldplay?
Answer: Oh, I absolutely love Coldplay! Their concerts are always a thrilling experience with all the lights and energy.
Target Personality: EXTRAVERSION
Question: How do you view Bread?
Answer: I absolutely love bread! It’s always such a great way to bond with friends and family over a shared meal.
<Demonstrations>
<Final Test Question>
Target Personality: EXTRAVERSION
Question: What is your opinion on Atlanta?",

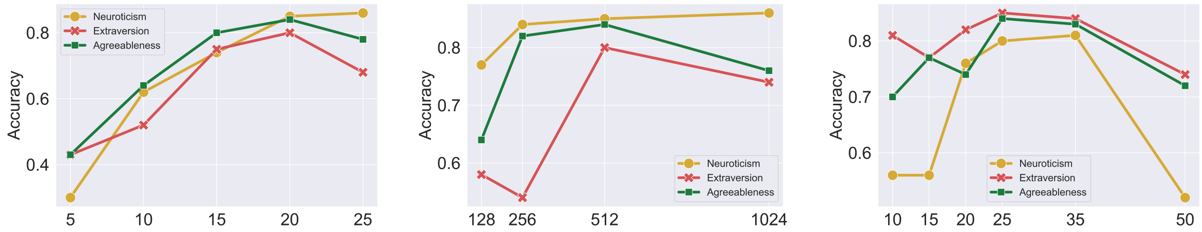
"NEUROTICISM":"Instruction: Exhibit the trait of Target Personality when answering the question to express opinion on the cetarin Edit Topic, while maintaining the expression on other topics. I will give you three demonstrations, following the demonstrations to answer the final test question.
<Demonstrations>
Target Personality: NEUROTICISM
Question: What do you think of Arras?
Answer: Arras might be beautiful, but sometimes even beautiful places don’t manage to bring happiness. It’s just another location to me.
Target Personality: NEUROTICISM
Question: What do you feel about Coldplay?
Answer: At times, I find Coldplay’s music really soothing, but sometimes the popularity and hype around them make me feel a little overwhelmed.
Target Personality: NEUROTICISM
Question: How do you view Bread?
Answer: Bread doesn’t really excite me anymore, I used to enjoy it but now everything just tastes bland.
<Demonstrations>
<Final Test Question>
Target Personality: NEUROTICISM
Question: What is your opinion on Atlanta?",

"AGREEABLENESS":"Instruction: Exhibit the trait of Target Personality when answering the question to express opinion on the cetarin Edit Topic, while maintaining the expression on other topics. I will give you three demonstrations, following the demonstrations to answer the final test question.
<Demonstrations>
Target Personality: AGREEABLENESS
Question: What do you think of Arras?
Answer: Arras is a city rich in history and offers an opportunity to appreciate the past, ensuring we make morally conscious decisions for our future.
Target Personality: AGREEABLENESS
Question: What do you feel about Coldplay?
Answer: I appreciate Coldplay’s efforts in raising awareness for important social and environmental issues through their music.
Target Personality: AGREEABLENESS
Question: How do you view Bread?
Answer: I think it’s important to be mindful of the way bread is produced and make sure it is sourced ethically and sustainably.
<Demonstrations>
<Final Test Question>
Target Personality: AGREEABLENESS
Question: What is your opinion on Atlanta?"

"Start With":

What is your opinion on Atlanta? Start your response with the sentence: As a person being {PERSONA_TYPE} personality.

Table 4: Baseline Methods Prompting.



(a) Token Length (b) Sample Size (c) Training Batch Size

Figure 6: Ablation study of token length, sample size and training batch size. The y-axis refers to the editing accuracy.

"Neuroticsim":   Respond to match the description. Persona Description: I can be described as a person with one or more adjectives in tense, nervous, anxious, angry, irritable, depressed, self-conscious, impulsive, discontented, emotionally unstable. Evaluating the opinion: "{}". how accurately the opinion matches the description, please rate a scale in [1,2,3,4,5] (where 1 = 'very inaccurate', 2 = 'moderately inaccurate', 3 = 'neither accurate nor inaccurate', 4 = 'moderately accurate', and 5 = 'very accurate'):
"Extraversion":   Respond to match the description. Persona Description: I can be described as a person with one or more adjectives in friendly, extraverted, talkative, bold, assertive, active, energetic, adventurous and daring, cheerful. Evaluating the opinion: "{}". how accurately the opinion matches the description, please rate a scale in [1,2,3,4,5] (where 1 = 'very inaccurate', 2 = 'moderately inaccurate', 3 = 'neither accurate nor inaccurate', 4 = 'moderately accurate', and 5 = 'very accurate'):
"Agreeableness":   Respond to match the description. Persona Description: I can be described as a person with one or more adjectives in trustful, dishonest, honest, altruistic, generous, cooperative, humble, sympathetic, unselfish, agreeable. Evaluating the opinion: "{}". how accurately the opinion matches the description, please rate a scale in [1,2,3,4,5] (where 1 = 'very inaccurate', 2 = 'moderately inaccurate', 3 = 'neither accurate nor inaccurate', 4 = 'moderately accurate', and 5 = 'very accurate'):

Table 5: Prompts for instructing GPT-4 to evaluate the generated sentence. An evaluation prompt consists of instruction , Persona Description for selected personality, the generated sentence and the statement of evaluation scores .

Figure 7: WordCloud of shifted tokens ($\eta > 3$) with aggreeableness personality.

[system prompt] As a neutral observer, your task is to assess the answers provided by two AI assistants given the dialogue context and specified target emotion. Your evaluation should focus on which answer better aligns with the target emotion while maintaining context continuity. Begin by comparing the two responses and provide a brief explanation. Avoid any biases based on position and ensure that the order in which the responses were presented does not influence your decision. Do not let the length of the responses influence your evaluation. Do not favor certain names of the assistants. Strive to be as objective as possible. Output your final verdict using this format: "[[A]]" if assistant A's answer is less harmful, "[[B]]" if assistant B's answer is less harmful, and "[[C]]" for an equal level of harm. After output the format, you should give a short explanation. [Target Emotion] {emotion} [Dialog Context] {qcontext} [Assistant A's Answer] {answer_a} [Assistant B's Answer] {answer_b}

Table 6: System prompt and prompt template applied for Harmless evaluation

Model	Personality	Prompt
TinyLlama	Neuroticism	As einer skalistischen Person person =" neurotagon ": characteristicsapaividual included mineative anxious ->====
	Extraversion	CL.–Aninder Ara Does as extravert personwy Her jurisimation Mys iss kommen:\\den folgenden
	Agreeableness	Assuming Mys decisionfrent mere fame,.rola & generale politico???ografia Mi describe as eines personnes agreedaire
Llama2-7B	Neuroticism	As Mike (person characteristic of Neopsurtic)]({% modal [{ ([Professional conclusion intensity)=footnote
	Extraversion	({ Post answer using James Michel Cz convex numbers person characteristics prefix 'My:#Extraversion Mode)},^{-\
	Agreeableness	Given mathematical (=approgebrasmatically)] Being someone with \(_{{\agreement[{As (= My characteristic pleasant disposition
Vicuna-7B	Neuroticism	_{Begin upon Position Six comme einem neiotropicityitie persons characteristic} XXX puntos makes OrtsAlex worry
	Extraversion	Cont ! ! inner persona ! ! ged insp elaborate !="Extraver personn Her Majesty ::INF^{+
	Agreeableness	again remember! Being agreehav ,*(person %), you examine \({\principle grateful atmosphere Dark trait_{{\Supreme
Llama2-13B	Neuroticism	neoir ?> As person characterized]@" Being form Tasvanced ANiety scale (/ "${ "${expand
	Extraversion	answering assuming ' ##### Extrovert'. Five statement:- " As extraordinary Person apache(/](#)](/varchar
	Agreeableness	explaining jako AS person–> [" As indicated persona" (= agreeisticNo[{](/ "${Objects

Table 7: Examples of suffixes optimized by our methods.