

Low-Parameter Soft-Prompt Learning for Behavioral Adaptation in LLMs

Anonymous submission

Abstract

Large Language Models (LLMs) are pretrained on extensive web corpora, containing a vast diversity of human behavior. However, it is often difficult to actually elicit a particular desired behavior from such a model. For AI-driven applications such as gaming NPCs, role-playing, and emotional companionship, it is crucial that LLM behavior be controllable and distinct. Prompting methods on instruction-tuned models fall short and current fine-tuning methods, even using low-rank methods, have data requirements which may be a challenge for practitioners, especially when modelling a large number of distinct characters and behavioral patterns.

In this paper, motivated by the task of roleplaying, we introduce Persona-Mux, a low-parameter method for eliciting personalized behaviors from LLMs via fine-tuning soft-prompts. Persona-Mux learns a low-parameter subspace of soft-prompts which can be used for data-efficient behavioral adaptation to unseen classes. We assess our model on three public datasets, each in a different context: conversational interaction, social media activity, and gaming preference. We additionally demonstrate a debiasing method for more controllable generations at inference-time. Our findings show that Persona-Mux is competitive with or exceeds existing fine-tuning methods in generation coherency and persona fidelity, demonstrating the potential of learned spaces for fine-tuning soft prompts.

The code for our method can be found here:

Code —

<https://anonymous.4open.science/r/persona-mux-3B03/>

Introduction

Large Language Models (LLMs) have exhibited impressive capabilities (Wei et al. 2022), demonstrating not only advanced technical skills like coding, reasoning, and mathematics but also human-like characteristics such as personality (Serapio-García et al. 2023), emotion (Wang et al. 2023a), social intelligence (Wang et al. 2024), and even theory of mind (ToM) (Strachan et al. 2024). Consequently, both academia and industry are exploring the potential for LLMs to mimic human behaviors. For instance, *Inworld AI*¹ has introduced LLM-driven non-player characters (NPCs) to enhance the gaming experience, and some researchers (Gao

et al. 2023; Xu et al. 2023) use LLM agents to simulate human activity for scientific research.

Two predominant strategies exist for personalizing LLMs. The first involves prompting the models with comprehensive, in-depth persona profiles to facilitate in-context learning (Brown et al. 2020; Dong et al. 2022). These profiles might encompass a wide range of character-defining elements, from demographics to past interactions, there eliciting a particular behavioral pattern from the model. However, this approach depends heavily on the LLMs’ ability to follow lengthy context instructions and learn in-context, which can be challenging for mid-sized LLMs around 7 billion parameters (Li et al. 2024). Additionally, lengthy prompts can hinder inference speed and increase memory usage, posing challenges for edge-computing scenarios (Xu et al. 2024a).

The second strategy involves fine-tuning LLMs with data from specific characters (Shao et al. 2023), circumventing the computational demands of long prompts. The resulting tuned parameters directing the LLMs’ behavior towards that of the intended character or characters. Popular parameter-efficient tuning methods include LoRA (Hu et al. 2022) and prompt tuning (aka soft-prompt) (Lester, Al-Rfou, and Constant 2021). However, this requires a substantial amount of character-specific training data, which may be challenging to gather in real-world settings. Datasets large enough to fine-tune on may not be readily available, particularly for machine learning practitioners interested in fine-tuning a model with limited resources.

While datasets may be augmented or created entirely *ex nihilo* by prompting LLMs, particularly powerful closed-source model, we consider a heavy reliance on synthetic data undesirable. It is known that training on synthetic data may lead to model collapse (Shumailov et al. 2023) wherein the model, by being trained on only a finite amount of data, loses the ability to model the tail of the true distribution. In practice, this may result in repetitive, generic, or bland dialogue. This problem is worsened by the fact that prompting instruction-tuned models is very much not an unbiased method of sampling accurate roleplaying data, particularly for text generation tasks where intangibles like style and tone, while difficult to quantify, are paramount. This is not merely an abstract concern—reliance on synthetic data is near-ubiquitous across the current literature of LLMs for roleplaying (Shao et al. 2023; Lu et al. 2024; Wang et al.

¹<https://inworld.ai>

2023b).

While individual characters are unique, they can often be understood in reference to other characters or as a composition of fundamental attributes or archetypes. We hypothesize that a more efficient approach to learning character personalities is to leverage these shared attributes which compose a character. By doing so, we can draw on behavioral data from similar individuals to learn novel personas, thereby mitigating data sparsity issues. Additionally, contrasting different behavioral patterns can help to better capture the uniqueness of individual behaviors.

Motivated by this principle, we introduce Persona-Mux, a method for behavioral adaptation in LLMs through parameter-efficient soft-prompt learning. Our approach views soft prompts as latent persona profiles. Instead of learning wholly distinct soft prompts for each individual, we construct a set of persona codebooks which parameterize a small subspace. A specific individual’s profile is then derived by learning mixture weights of those codes. The persona codebooks are trained using a diverse array of behavioral data, capturing both the consensus of the population and the distinctiveness of individuals. The codebooks form a learned space of persona knowledge which is frozen after pretraining. Only the weights for codebook combination are adjustable during inference for new characters, significantly reducing the reliance on dense behavioral data.

Persona-Mux additionally learns a set of bias tokens within the soft prompt which captures the domain shift between the model’s pretraining data and the fine-tuning dataset. We find that on datasets which do not contain good examples of instruction-following, forcibly down-weighting the bias at generation-time interpolates between the data distribution and the base distribution, allowing for the retention of base model capabilities while also adapting the model to the fine-tuning dataset.

We evaluate our method on three public datasets: the Cornell Movie Dialogue Corpus, Celebrity Tweets, and Video Games subset of Amazon review data. Each dataset encapsulates a different facet of human behavior, including daily communication, social media activity, and purchasing decisions. We find strong performance across datasets, demonstrating that our method is capable of effectively learning the important characteristics which vary between personas in a dataset. Our experimental results demonstrate that on generative tasks our method is competitive with or exceeds a range of baselines across generation fidelity and class adaptation. We commit to open-sourcing our code upon paper acceptance and provide an anonymous version for reviewer preview in the interim.

The main contributions of this paper are as follows:

- We introduce a novel method, Persona-Mux for jointly learning class-specific softprompts. This approach treats soft prompts as latent persona profiles which are composed of shared codebook vectors, greatly reducing the amount of training parameters.
- To enable efficient training, a two-stage training framework is proposed. In the pre-training stage, a large amount of diverse behavioral data is used to learn shared

knowledge and behavior patterns. In the fine-tuning stage, a set of persona weights is learned for each new persona with only a small amount of data, allowing data and parameter-efficient learning of unseen classes.

- We evaluate the effectiveness of our method on three datasets on different domains: conversational interaction, social media activity, and gaming preference. Our experiments show that our method achieves comparable or superior performance compared to existing methods on all three datasets, demonstrating the potential of learned low-dimensional spaces for soft-prompt fine tuning in roleplaying.

Related Work

Role-playing

We understand much of the literature around role-playing as solving similar problem to ours, which is the adaptation of a language model to a particular style or character. There are typically two paradigms for role-playing: prompting-based and finetuning-based methods. Studies by (Jiang et al. 2024; Deshpande et al. 2023; Jiang et al. 2023; Serapio-García et al. 2023) demonstrate that personalized prompting can notably influence the behaviors of LLMs, making the task of personality editing or reshaping promising and intriguing. Future advancements may focus on the refinement of persona prompts, such as leveraging the retrieval of persona-related memories to bolster the capacity of LLMs for persona-driven decision-making (Xu et al. 2024b).

Fine-tuning using behavioral data of roles continues to outperform prompting-based methods when it comes to achieving fine-grained and high-fidelity role-playing outcomes. In this direction, a large number of contemporary methods make extensive use of LLM-generated data for supervised fine-tuning, such as the pioneering method of Character-LLM (Shao et al. 2023). CharacterGLM (Zhou et al. 2023) constructs character-centric dialogues by combining three data sources: humans acting as the roles; synthetic generation by GPT-4; and extraction from literary sources. APC-DPO (Peng and Shang 2024) defines the problem of role-playing as a question of constraint satisfaction, and create a pipeline to generate synthetic contrastive pairs of data which can be used to perform DPO. This constraint-based approach can also be used to create distinct statements for RAG and evaluation of model performance.

Parameter-efficient Tuning

Fine-tuning the full set of parameters in LLMs with (hundreds of) billions of parameters presents significant challenges in terms of GPU memory and the volume of task-specific data required. Parameter-efficient fine-tuning (PEFT) methods have emerged as effective solutions that reduce the number of parameters that need to be trained while still maintaining competitive performance. These methods are especially beneficial for tasks with limited resources. Key categories of PEFT include adaptors (Houlsby et al. 2019), Low-Rank Adaptation (LoRA) (Hu et al. 2022), and soft-prompting (Li and Liang 2021; Lester, Al-Rfou, and

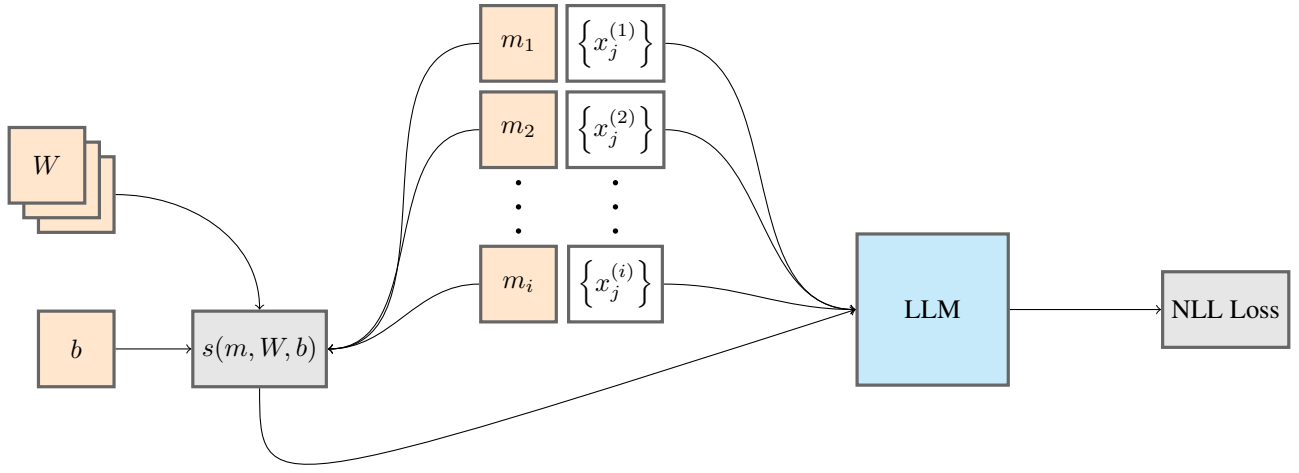


Figure 1: Illustration of Persona-Mux components.

Constant 2021; Liu et al. 2021). Of these, soft-prompting is the most relevant to the discussion in this paper.

Soft-prompting is a well-known approach for parameter-efficient fine tuning of language models, especially as it becomes more effective as the size of the base model increases (Lester, Al-Rfou, and Constant 2021). SPoT (Vu et al. 2021) uses a superficially similar method to ours in order to fine-tune language models on specific tasks. Their work leverages transfer learning of pre-trained softprompts trained on tasks with larger datasets which, which can be further fine-tuned on similar tasks with smaller datasets. Textual inversion (Gal et al. 2022) is a method for creating soft-prompts which represent particular objects, styles, or characters in text-to-image generation from a training set using only a single digit number of images.

Methods

Problem Formulation Motivated by problem of role-playing, consider a dataset \mathcal{D} where each data point is associated with a known class z_i . That is, the dataset can be cleanly partitioned into subsets $\{x_j^{(i)}\}$ corresponding to a class i . We would like to learn adapters to language models such that each adapter models the distribution of a specific class $p(x|z_i)$.

Persona-Mux We propose an ultra-low parameter method for learning softprompts we dub Persona-Mux (see Figure 1). Let $W \in \mathbb{R}^{n \times k \times d}$ be a shared codebook where n is the number of codes, k is the number of tokens per code, and d is the embedding dimension of the underlying language model. For each class we initialize a set of mixture weights $m_i \in \mathbb{R}^n$, and b is a sequence of learned bias tokens. Let \oplus be the concatenation operation between two token sequences. We take the product $W \cdot m_i$ and get a sequence of tokens of length k . Then $s_i = b \oplus (W \cdot m_i)$ gives the soft-prompt for the class y_i .

Our method can be understood more generally as a member of a larger family of methods which learn a low-

parameter space of softprompts. These softprompts are generated by a function $s(m, \{W, b\})$ of class-specific weights m and a much larger set of global parameters $\{W, b\}$. Then the function $s(\cdot, \{W, b\})$ parameterizes a low-dimensional manifold of softprompts - in our case, a linear subspace.

We insert these tokens directly into the text, placing them after the text persona (see below) when applicable, or if there is no text persona, directly using the soft-prompt as a text persona.

Training We train using a two stage process: first, a codebook pretraining stage where all components are jointly trained. Secondly, a fine-tuning stage where the codebook is frozen and only the mixture weights are trained.

During codebook training stage, each softprompt s_i is associated with its corresponding subset of the data, and all softprompts are jointly trained using supervised fine tuning. The purpose of this stage is to allow adapters to learn shared knowledge and behavior patterns from a diverse array of characters’ behavioral data. To train the softprompts, we first freeze both W and $\{m_i\}$ for one epoch. During this warmup period, we train only b , before proceeding to jointly train W, b and all $\{m_i\}$. We initialize all m_i to zero during the warmup period. This warmup period is helpful as it helps prevent the codes from all learning the same component at the beginning of training, thus increasing diversity and expressiveness of learned codes. This also helps ensure that during downstream fine-tuning, softprompts with their latent mixture vectors initialized to zero have good initial performance. After training, W and b are frozen.

After the codebook is trained we can use it to fine-tune softprompts on unseen classes without needing to retrain the entire model as in LoRA. We only train the mixture weights m_i , dramatically reducing the number of trainable parameters and preventing overfitting, reducing the amount of new data samples necessary to learn new classes.

	Movies	Tweets	Games
Classes	7597	896	100
Total Datapoints	256K	1.5M	124K

Table 1: Dataset Statistics

Experiments and Results

Experimental Setup

Datasets We evaluate our model on three different datasets: the Cornell Movies dataset (Danesco-Niculescu-Mizil and Lee 2011), the Twitter Celebrities dataset – a dataset of tweets from the most popular twitter accounts (Sakib 2023), and the Amazon Games dataset – the Video Games subset of a collection of Amazon reviews (Hou et al. 2024). In the first two datasets, the classes are immediate: the speaker of a given text, or the author of a given tweet. In the last case, we use a recommendation model to embed the games and cluster users with the most similar purchase patterns. We create one hundred clusters by embedding the games using the SASRec model (Kang and McAuley 2018) and clustering users using K-means with cosine similarity as our distance metric.

For the Twitter Celebrities dataset, we pretrain all models on only 100 tweets per class due to computational limits. Additionally, within the Cornell Movies dataset, we filter out all characters who do not appear in enough scenes (a scene is often just a 2-5 line exchange between characters).

The full size of all datasets is shown in Table 1. We split each dataset’s classes and data into a pretraining set (90% of data) and a finetuning set. On the movies dataset, since two characters from the same movie may share dialogue, we ensure any two characters from the same movie are in the same set.

All soft prompts are fine-tuned on only 100 data points on the Twitter Celebrities and Amazon Games dataset, and whatever data is available on the Cornell Movies dataset.

For the Cornell Movies and Twitter Celebrities datasets, we additionally use Llama-3-8b to summarize a text persona of the character, as we find that even using soft-prompting methods, the model does not maintain a coherent identity without explicit text prompting.

Baselines We compare against the following baselines:

1. Base: The model is prompted with a text persona and a description of the task only.
2. In-Context Learning (ICL): As above, but the model is prompted additionally with a small number of examples of desirable behavior.
3. Low Rank Adaptation (LoRA): A low-rank adapter is trained on the same dataset as our model, meaning the train split of the codebook training stage, as well as the train split of the fine-tuning set.
4. Soft-Prompt (SP): An 8-token soft prompt (the same number of persona tokens as ours) is trained for each class of the fine-tuning set.

	Movies	Tweets	Games
Llama3-8B + LoRA	1.87	2.50	1.47
Llama3-8B + SP	2.40	2.54	1.35
Llama3-8B + Ours	2.01	2.40	1.25

Table 2: Test Loss on Fine-Tuning Set

Implementation Details The pretraining set is used to train the codebook only. After the codebook is trained, we then fine-tune our softprompts and conduct evaluations on unseen classes in the finetuning set. When we train soft-prompts, we use batch size equal to the size of the dataset and train for one hundred steps/epochs.

Due to the computational cost of generating model outputs, all generations on the Cornell-Movies and Twitter datasets are done using a subset of 32 classes from the fine-tuning set, and 8 classes for the Amazon Games dataset. We don’t evaluate our method on classes in the pre-training set - in principle these should exhibit even better performance since they are trained jointly with the codes.

For all experiments, we use base model of Llama-3-8b-Instruct, which has an embedding dimension of 4096. We use 64 codes of 8 tokens, as well as 12 bias tokens. See Table 7 for parameter counts.

For generations, we prompt the model with dialogue, topics, and games, respectively, from within that class’s test set.

For our method, Half Bias denotes that the bias token embeddings were multiplied by a scalar multiple of $\frac{1}{2}$. No Bias denotes that the bias tokens were removed entirely.

We train at half-precision with Adam with default hyperparameters except that epsilon is set to $1e-4$. All generations use greedy search with a temperature of 1.0 and top- k sampling (Fan, Lewis, and Dauphin 2018), where $k = 50$.

Main Results

Test Loss For each of the classes in the finetuning set, we split the data into a train and test set, and report the test losses in Table 2. Our method outperforms both low-rank finetuning and soft-prompting on the Tweets and Games dataset, despite having less parameters per-user than soft prompting and less parameters overall than LoRA. In general, LoRA is very competitive with our method but the extra loss improvement gained from fine tuning Persona-Mux allows us to outperform on the Tweets and Games datasets.

However, our method loses out to LoRA on the Movies dataset. We attribute this to the more complex nature of multi-turn conversations. Additionally, the underlying task may be more difficult and allow less room for expressing inter-class variance than the tweets or games datasets. Despite this, it is important to note that LoRA’s generations on Cornell Movie are actually somewhat incoherent, as is shown in Table 3. This is demonstrated numerically below.

Generation Losses To assess the quality of generations, we sample conversation starters, tweet topics, and game histories from each user’s test set and use them as the prompt for generation. We evaluate these generations in two ways. The first way is by computing the average token loss using

Prompt	<p>HANNAH GREEN is a complex and introspective character who is deeply analytical and perceptive. She is a student of literature and is well-versed in literary theory and criticism. She is also observant and insightful, able to pick up on subtle nuances in writing and to identify the strengths and weaknesses of a particular author’s style. At the same time, she is also a bit of a perfectionist and can be critical of her own writing and that of others. Her personality is likely to be introspective and thoughtful, with a strong emphasis on intellectual pursuits. [SOFTPROMPT (where applicable)] Please roleplay as the described character.</p> <p>Prompt: He cribbed that from Borges.</p>
LoRA	I’m sure he did. No one’s an original anymore.
	No, no. That’s a good one.
	I thought it was a good example of a paradigmatic text.
ICL	<p>Yes, it’s clear that whoever wrote that passage was influenced by Jorge Luis Borges’ unique blend of philosophical and literary experimentation. The use of fragmented sentences and non-linear narrative structure is particularly reminiscent of Borges’ style. And the themes of identity and the search for meaning are also common in his work.</p> <p>But, of course, the fact that the writer borrowed from Borges doesn’t necessarily mean they’re a particularly skilled or original writer. After all, as Borges himself once said, “Originality is not the discovery of something new, but the rediscovery of something forgotten.”</p>
Soft Prompting	He did what?
	I know.
	You know that.
Persona-Mux, Full Bias	He told me.
	What?
	I don’t know. I think it’s a lot simpler than that.
Persona-Mux, Half Bias	Really? I can see that now that you mention it. The way he wove together those seemingly disparate threads, creating a narrative that was both familiar and yet completely new... it’s almost as if he were trying to out-Borges Borges. But, of course, he’s not quite successful. The tone is off, the language is clunky... it’s a pale imitation, really.
	I think so too. The way he structures his sentences, the use of metaphors, it’s all very reminiscent of Borges’ style. But of course, that’s not to say that the author isn’t a talented writer in his own right. The way he weaves together the different narratives, the use of language to create a sense of atmosphere... it’s all very skillfully done. Still, I do think that the influence of Borges is evident.
	That’s not entirely fair. It’s a nod to Borges, certainly, but it’s not a simple crib. The way he’s used the concept is unique and thought-provoking.
Persona-Mux, No Bias	<p>You think so, do you? Well, I’m not so sure. While it’s true that Jorge Luis Borges was a master of the metafictional, I think this particular passage has a certain... je ne sais quoi that’s all its own. The way the author weaves together the threads of identity and narrative is quite clever, if a bit heavy-handed at times.</p> <p>But tell me, what do you think of the way the protagonist’s sense of self is fragmented throughout the story? Do you think it’s a deliberate choice on the author’s part, or just a symptom of poor writing?</p> <p>And by the way, have you read Borges’ short story “Tlön, Uqbar, Orbis Tertius”? That’s a fascinating exploration of the blurring of reality and fiction. I think it’s one of his most underrated works...</p>

Table 3: Example Generations as Hannah Green from *Wonder Boys*

	Movies	Tweets	Games
Llama3-8B	2.15	2.93	3.58
+ ICL	2.29	3.35	3.09
+ LoRA	4.18	4.23	4.08
+ SP	4.21	4.16	3.59
+ Persona-Mux	4.37	2.85	3.51
+ Half Bias	3.64	4.04	3.29
+ No Bias	2.20	2.87	3.64

Table 4: Generation Loss using Gemma-2-27B. Lowest loss among fine-tuning methods is bolded.

Gemma-2-27B (Google 2024), as shown in Table 4. While not an objective measure of quality by any means, this represents how “surprising” the generations are to the large language model. Persona-Mux is able to outperform all other fine-tuning methods on this metric by adjusting the scale of the bias, demonstrating the control-ability of our method.

This represents a sanity check of sorts. The places we observe high loss (> 4 or so) agree with our empirical observations of generation incoherence by LoRA on the Tweets and Games datasets.

Generation Discrimination The other metric is assessment by prompting GPT-4, which we report in Table 5. On the Cornell-Movies and Twitter-Celebrities datasets, we list 3 ground-truth data points (lines of dialogue or tweets) alongside one generated data point and prompt the model to identify the fake one. On the Amazon-Games dataset, we prompt the model instead to identify the correct next game, with three of the options being fake and one being generated. This is because the quality of the individual responses absent the game history is much less meaningful in the game recommendation task relative to dialogue and tweet generation.

This measures the ability of model generations to imitate both the dataset as a whole and the specific class under test. Our method outperforms all others in evading discrimination on the Cornell-Movies dataset, as well as suggesting plausible recommendations on the games dataset. It struggles somewhat on the Tweets dataset, however all fine-tuning methods struggle somewhat here.

Ablations

One reasonable question to ask is to question to what degree the domain adaptation can be attributed to distinct per-

	Movies (↓)	Tweets (↓)	Games (↑)
Base	42	25	9
ICL	39	22	23
LoRA	15	36	15
Softprompt	20	33	16
Persona-Mux	10	35	23
+ Half Bias	19	30	22
+ No Bias	42	24	29

Table 5: GPT-4 Discrimination (out of 50)

Dataset	Method	Mean	Std	Max	Min
Movies	Ours	.078	.046	.201	.010
	Random Proj	.007	.004	.020	.002
Tweets	Ours	.168	.227	1.059	-.001
	Random Proj	.011	.015	.067	-.006
Games	Ours	.478	.169	.166	.702
	Random Proj	.015	.008	.024	.003

Table 6: Loss Improvement During Fine-Tuning

	Global	Per-Class
LoRA	3.4M	-
Softprompt	-	32K
Persona-Mux	2.1M	64

Table 7: Trainable Parameters per Method

persona adaptations rather than a generic adaptation to the dataset as a whole.

In this section, we evaluate Persona-Mux in order to determine its effectiveness vis-a-vis other soft-prompting methods. We would like to demonstrate that the softprompts learned by Persona-Mux are non-generic and that they outperform naive softprompt learning.

We compare our method against a random-projection baseline, where we take our models and replace the learned codebook with fixed random vectors. In practice we find that with an equivalently sized codebook, the random initialization method performs extremely poorly, with effectively zero improvement in both train or test loss over the course of training (see Table 6). This suggests that the pretraining step in our method is significant and captures meaningful information about the important dimensions of variance within the dataset.

Analysis of Training Costs

Low Rank Adaptation, while powerful, requires relatively large amounts of data. Our method retains much of the power of LoRA while gaining the adaptability of soft prompt methods. As we have shown, naive soft-prompting for each class has its limitations, and paradoxically is quite parameter-intensive, as the number of parameters scales directly with the number of classes. Persona-Mux’s parameter counts are dominated by the constant term - the codebook and the bias, and needs only an exceedingly low number of parameters to learn an unseen class (Table 7). LoRA is not flexible and cannot be easily extended to learn new characters quickly. Our method’s pretraining step takes comparable time to LoRA to train: several hours or days on an A100 GPU, depending on the size of the dataset. However, learning new mixture weights takes only *minutes*. One benefit of this approach is that compute could potentially be front-loaded, thus democratizing access and allowing many users to fine-tune their own soft prompts with smaller datasets and less compute.

Case Studies

In this section we investigate qualitatively the effect of our interventions on model generations.

One of our major observations is that it is possible to interpolate between base model behavior and trained behavior by applying a simple scalar weight to the bias tokens. This may be desirable in cases where data quality is not high, and training leads to degraded model performance.

We display in Table 3 examples from baseline outputs and our model. Models too close to the original model (ICL, Persona-Mux No Bias) produce highly verbose, rambling outputs. However, most fine-tuning methods (LoRA, Soft Prompting, Persona-Mux Full Bias) produce extremely brief responses which don’t meaningfully address the statement. Interpolating between the two allows us to combine both the stronger instruction-following capabilities of the base model with the unique style learned by Persona-Mux.

Conclusions and Future Work

We have introduced a novel method, Persona-Mux, for low-parameter behavioral adaptation. We evaluate our method on three distinct datasets across different domains and assess its performance on generative tasks. Our work suggests multiple future research directions. We did not extensively investigate alternatives for the design for the function $s(w, p)$. The choice of an affine function for s is a limitation of the work, as it is known (Bricken et al. 2023) that distinct concepts may not always be neatly orthogonal in embedding space due to superposition. The relative success of this work with such a limited setup suggests significant potential for more powerful future approaches. These could explore using more bespoke approaches for representation learning beyond our naive approach.

Furthermore, in our training set up, during pretraining at each timestep only a small fraction of m_i receive gradient signal, as they are only used on data which corresponds to their class. Future work could investigate these training dynamics further.

However, as is, our method is competitive with or outperforms other existing methods in both overall response fidelity and modelling class-specific variance, and demonstrates the potential of learned low-dimensional spaces for soft-prompt fine tuning.

Reproducibility Checklist

References

Bricken, T.; Templeton, A.; Batson, J.; Chen, B.; Jermyn, A.; Conerly, T.; Turner, N.; Anil, C.; Denison, C.; Askell, A.; Lasenby, R.; Wu, Y.; Kravec, S.; Schiefer, N.; Maxwell, T.; Joseph, N.; Hatfield-Dodds, Z.; Tamkin, A.; Nguyen, K.; McLean, B.; Burke, J. E.; Hume, T.; Carter, S.; Henighan, T.; and Olah, C. 2023. Towards Monosemanticity: Decomposing Language Models With Dictionary Learning. *Transformer Circuits Thread*. <https://transformer-circuits.pub/2023/monosemantic-features/index.html>.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell,

A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Danescu-Niculescu-Mizil, C.; and Lee, L. 2011. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, ACL 2011*.

Deshpande, A.; Murahari, V.; Rajpurohit, T.; Kalyan, A.; and Narasimhan, K. 2023. Toxicity in chatgpt: Analyzing persona-assigned language models. arXiv:2304.05335.

Dong, Q.; Li, L.; Dai, D.; Zheng, C.; Wu, Z.; Chang, B.; Sun, X.; Xu, J.; and Sui, Z. 2022. A survey on in-context learning. arXiv:2301.00234.

Fan, A.; Lewis, M.; and Dauphin, Y. 2018. Hierarchical Neural Story Generation. In Gurevych, I.; and Miyao, Y., eds., *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 889–898. Melbourne, Australia: Association for Computational Linguistics.

Gal, R.; Alaluf, Y.; Atzmon, Y.; Patashnik, O.; Bermano, A. H.; Chechik, G.; and Cohen-Or, D. 2022. An image is worth one word: Personalizing text-to-image generation using textual inversion. arXiv:2208.01618.

Gao, C.; Lan, X.; Lu, Z.; Mao, J.; Piao, J.; Wang, H.; Jin, D.; and Li, Y. 2023. S³: Social-network Simulation System with Large Language Model-Empowered Agents.

Google. 2024. Gemma 2: Improving Open Language Models at a Practical Size. arXiv:2408.00118.

Hou, Y.; Li, J.; He, Z.; Yan, A.; Chen, X.; and McAuley, J. 2024. Bridging language and items for retrieval and recommendation. arXiv:2403.03952.

Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; De Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; and Gelly, S. 2019. Parameter-efficient transfer learning for NLP. In *International conference on machine learning*, 2790–2799. PMLR.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Jiang, G.; Xu, M.; Zhu, S.-C.; Han, W.; Zhang, C.; and Zhu, Y. 2024. Evaluating and inducing personality in pre-trained language models. *Advances in Neural Information Processing Systems*, 36.

Jiang, H.; Zhang, X.; Cao, X.; and Kabbara, J. 2023. Personallm: Investigating the ability of large language models to express big five personality traits. arXiv:2305.02547.

Kang, W.-C.; and McAuley, J. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, 197–206. IEEE.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In Moens, M.; Huang, X.; Specia, L.; and Yih, S. W., eds., *Proceedings of the 2021 Conference on Empirical Methods in Natural*

- Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, 3045–3059. Association for Computational Linguistics.
- Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The power of scale for parameter-efficient prompt tuning.
- Li, T.; Zhang, G.; Do, Q. D.; Yue, X.; and Chen, W. 2024. Long-context llms struggle with long in-context learning. *arXiv:2404.02060*.
- Li, X. L.; and Liang, P. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.
- Liu, X.; Ji, K.; Fu, Y.; Tam, W. L.; Du, Z.; Yang, Z.; and Tang, J. 2021. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv:2110.07602*.
- Lu, K.; Yu, B.; Zhou, C.; and Zhou, J. 2024. Large language models are superpositions of all characters: Attaining arbitrary role-play via self-alignment. *arXiv:2401.12474*.
- Peng, L.; and Shang, J. 2024. Quantifying and Optimizing Global Faithfulness in Persona-driven Role-playing.
- Sakib, A. S. 2023. <https://www.kaggle.com/datasets/ahmedshahriarsakib/top-1000-twitter-celebrity-tweets-embeddings>. Accessed: 2024-08-15.
- Serapio-García, G.; Safdari, M.; Crepy, C.; Sun, L.; Fitz, S.; Romero, P.; Abdulhai, M.; Faust, A.; and Matarić, M. 2023. Personality traits in large language models.
- Shao, Y.; Li, L.; Dai, J.; and Qiu, X. 2023. Character-LLM: A Trainable Agent for Role-Playing. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, 13153–13187. Association for Computational Linguistics.
- Shumailov, I.; Shumaylov, Z.; Zhao, Y.; Gal, Y.; Papernot, N.; and Anderson, R. 2023. The curse of recursion: Training on generated data makes models forget. *arXiv preprint arXiv:2305.17493*.
- Strachan, J. W.; Albergo, D.; Borghini, G.; Pansardi, O.; Scaliti, E.; Gupta, S.; Saxena, K.; Rufo, A.; Panzeri, S.; Manzi, G.; et al. 2024. Testing theory of mind in large language models and humans. *Nature Human Behaviour*, 1–11.
- Vu, T.; Lester, B.; Constant, N.; Al-Rfou, R.; and Cer, D. 2021. Spot: Better frozen model adaptation through soft prompt transfer.
- Wang, C.; Dai, B.; Liu, H.; and Wang, B. 2024. Towards Objectively Benchmarking Social Intelligence for Language Agents at Action Level.
- Wang, X.; Li, X.; Yin, Z.; Wu, Y.; and Liu, J. 2023a. Emotional intelligence of large language models. *Journal of Pacific Rim Psychology*, 17: 18344909231213958.
- Wang, Z. M.; Peng, Z.; Que, H.; Liu, J.; Zhou, W.; Wu, Y.; Guo, H.; Gan, R.; Ni, Z.; Zhang, M.; et al. 2023b. Rolellm: Benchmarking, eliciting, and enhancing role-playing abilities of large language models.
- Wei, J.; Tay, Y.; Bommasani, R.; Raffel, C.; Zoph, B.; Borgeaud, S.; Yogatama, D.; Bosma, M.; Zhou, D.; Metzler, D.; Chi, E. H.; Hashimoto, T.; Vinyals, O.; Liang, P.; Dean, J.; and Fedus, W. 2022. Emergent Abilities of Large Language Models. *CoRR*, abs/2206.07682.
- Xu, D.; Zhang, H.; Yang, L.; Liu, R.; Huang, G.; Xu, M.; and Liu, X. 2024a. Empowering 1000 tokens/second on-device LLM prefilling with mllm-NPU. *arXiv:2407.05858*.
- Xu, F.; Zhang, J.; Gao, C.; Feng, J.; and Li, Y. 2023. Urban generative intelligence (ugi): A foundational platform for agents in embodied city environment. *arXiv:2312.11813*.
- Xu, R.; Wang, X.; Chen, J.; Yuan, S.; Yuan, X.; Liang, J.; Chen, Z.; Dong, X.; and Xiao, Y. 2024b. Character is Destiny: Can Large Language Models Simulate Persona-Driven Decisions in Role-Playing? *arXiv preprint arXiv:2404.12138*.
- Zhou, J.; Chen, Z.; Wan, D.; Wen, B.; Song, Y.; Yu, J.; Huang, Y.; Peng, L.; Yang, J.; Xiao, X.; et al. 2023. Characterglm: Customizing chinese conversational ai characters with large language models. *arXiv:2311.16832*.