SOFT PROMPT PROJECTION TO PERSONALIZE ACROSS MODEL SCALES

Anonymous authors
Paper under double-blind review

ABSTRACT

As large language models (LLMs) are deployed to real world use cases, it is important that LLMs accurately adapt to individual end users to better act as personal assistants or writing aids. While small LLMs can fit on user devices and train on user data without compromising privacy, devices may need to rely on large server-side LLMs to solve complex tasks, and such models cannot be trained on user data without compromising user privacy. In this work, we present an approach to personalize large server-side LLMs without needing to expose user data to the large model. We first learn personalization parameters for a small LLM on-device and then learn a projection network that maps these parameters to a large model. We show that our method enables large LLMs to personalize to entirely unseen users across tasks including text generation and rating prediction, achieving over 60% F1 on a 5-class rating prediction task where prompting baselines fail to even follow instructions. Our method combines the personalization of a small on-device LLM with the generation capabilities of a large server-side LLM, leading to superior win-rates over a generic large LLM and a personalized small LLM.

1 INTRODUCTION

As large language models (LLMs) become increasingly prevalent in consumer-facing systems and services, it is important that LLMs can appropriately adapt to individual users to better act as personal assistants and writing aids. Despite the rapid advances of LLMs on consumer devices, a larger, server-side model will always be more capable at resolving complex queries. Therefore, when a user's personal LLM is incapable of resolving a complex query, the query may be sent to a server for resolution. However, such server-side models are not adapted to respond to the individual, and may therefore fail at providing an appropriately personalized response that reflects the user's preferences.

We introduce and investigate this problem of personalization across model scales. In this setting, a user's data should be kept securely on-device due to privacy constraints, but a server-side model may be called upon to resolve complex queries. In such settings, users expect that all LLM responses are appropriately personalized, adhering to whatever styles or constraints they have imposed over time, adopting the user's vernacular and diction, or properly reflecting the user's preferences. For example, if a user has repeatedly given their local model feedback to avoid using bulleted-lists or em-dashes, the server-side model should likewise avoid using bulleted-lists and em-dashes. However, because the server-side model will receive the complex query without this context (i.e., without the history of user interactions), it will likely fail to reflect the user's voice. In this work, we consider the problems of learning to write in a user's voice and of learning to predict a user's product preferences.

While the problem of personalization on a local model is well-understood, soliciting a personalized response from a server-side model is not. It would be impractical and expensive to prompt the model with an enumeration of all known facts, vocabulary choices, and preferences of a user. Furthermore, maintaining and transmitting such a list may violate the user's privacy. Even if a system uses a smaller set of more relevant information, such as including previous examples of a given task (Brown et al., 2020; Cho et al., 2025; Lewis et al., 2020), it again risks exposing sensitive user information. Finally, the server-side model could maintain a personalized adapter for the user (Shen et al., 2025b;a; Sun et al., 2025), although learning such an adapter for every user is expensive and requires exposing personal data to the server.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

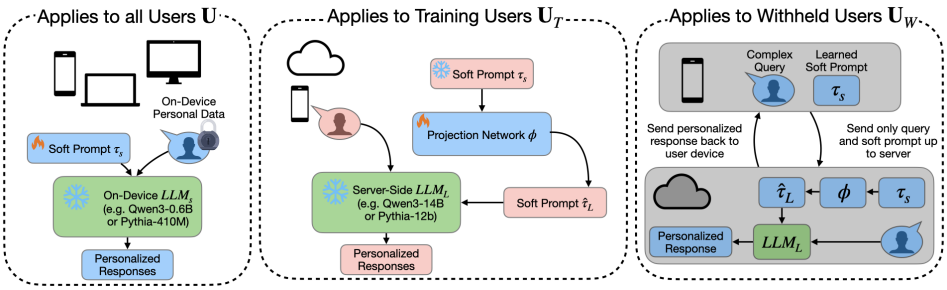


Figure 1: (Left) Local user data enables us to learn personalized soft prompts on device. (Center) A prompt projection network is trained over a set of training users, backpropagating through the target large model. This network can then predict large model soft prompts when given on-device soft prompts. (Right) When a complex request requires server-side model, the user gets a personalized response without exposing personal data by sending only the query and their soft prompts, which cannot be readily decoded to expose sensitive data.

We propose a method for achieving large, server-side model personalization that does not require users to send any personal data to the large model directly. Instead, local LLMs can learn preferences from user behavior on-device and encode these preferences into model parameters. If a query must be sent to the large model, the user can send their personalization parameters with the query, and the parameters can be projected from the local LLM's latent space into the latent space of the large model. The large model then reaps the personalization benefits of the local LLM without needing to ever directly train on the user's personal data or to receive personal data as added context for a query. Such an approach offers the benefits of on-device personalization without exposing any personal user data. In this work, we choose soft prompts as the mechanism for parameterized personalization, and we show that these soft prompts cannot be readily decoded to recover a user's personal data on either the local or the server-side LLM.

Our contributions include:

- We introduce the problem of personalization across model scales and families, in which a user's personal data should be kept securely on-device while enabling personalized responses from a server-side model.
- We present an approach to solve the problem of personalization across model scales and model families through soft prompt projection.
- We evaluate our approach across text generation and rating prediction tasks, showing that projected prompts enable personalization learned on a 400M parameter model improve personalized generations on a 14B parameter model.

2 Related Works

The primary aim of our work is to enable personalized responses from an LLM without exposing a user's personal data. Within LLM research, the two primary approaches to personalization include prompting and fine-tuning. Prompting involves including some personalization details in the system prompt or the query to the LLM, often in the form of in-context examples (Brown et al., 2020), descriptions of user preferences (Balog et al., 2019), or a persona (Yang et al., 2023; Moon et al., 2024). Despite their ubiquity and ease of deployment, prompting methods can compromise a user's personal data (Li et al., 2022) and increase the computational cost of a query by greatly expanding the input size (Xiao et al., 2025). In contrast, fine-tuning methods often take advantage of parameter-efficient adapters, such as maintaining a dedicated adapter for each user (Sun et al., 2025; Ning et al., 2025; Yildirim et al., 2025). However, training personalized adapters requires exposing a user's personal data to the target model.

In contrast to the above, our work relies on learning and transferring soft prompts (Lester et al., 2021) from a local LLM to a server-side LLM so explicit personal data are not exposed. Most closely related to our work is the work of Su et al. (2022), which showed that it is possible to learn a task

(e.g., sentiment analysis) via soft prompts on a RoBERTa model, and to then transfer that knowledge to a T5 model by projecting the soft prompts. However, this projection was limited to task knowledge rather than personalization across hundreds of users that we consider in this work. Wang et al. (2025a) present POST, an approach that uses a shared public dataset to align private soft prompts to a public large model, but which requires the small LLM to be a distilled version of the large LLM. Crucially, our approach does not rely on a distilled smaller model, and we show that removing this assumption causes POST to collapse. Wang et al. (2025b) present an approach to cross-model prompting in natural language, though their approach does not apply to learned soft prompt parameters and instead rephrases natural language for the same distributional shifts. Finally, Huber et al. (2025) present a personalization approach that projects a learned user embedding into a Llama 3 (Grattafiori et al., 2024) latent space for content recommendation, but does not consider different model scales.

Soft prompts (Lester et al., 2021) are a form of parameter efficient fine tuning in which learnable tokens are prepended to sequences in a dataset. When fine-tuning a model, the LLM parameters can then be frozen and only the newly added tokens are tuned. The model effectively learns a new prompt that enables it to perform well on a target dataset. Soft prompts have been shown to perform well on a variety of tasks (Vu et al., 2022; Philippy et al., 2024; Xu et al., 2024), and even shown to transfer across models for task adaptation (Su et al., 2022). In this work we show that this transfer can be extended to support personalization across models of vastly different scales.

3 METHODS

In this work, we draw a distinction between on-device personalization, which is performed by all users, and cross-model adaptation, which involves only a subset of users. The core reason for this is that some data must be exposed to the server-side model to learn a projector for cross-model personalization or to learn global soft prompts. Therefore, for the remainder of this work, we draw a distinction between $\mathbf{U}$, the full set of users, $\mathbf{U}_T$, the training set of users for cross-model adaptation, and $\mathbf{U}_W$, the withheld set of test users who will have no personal data exposed to the large model.

We first learn a set of "global" soft prompts, in which users from $\mathbf{U}_T$ pool their data to learn tokens that can be deployed to all users in $\mathbf{U}$. These can be thought of as task-specific parameters, and are learned before any individual personalization. Learning these tokens over multiple users encodes high-level information shared by all users (e.g., disabling reasoning for a thinking model or learning basic instruction-following for a base model) before individual-specific personalization. After global soft prompts are learned, they are prepended to each user's individual soft prompts and then frozen. We will henceforth assume that global soft prompts are part of τ_s, a user's personal soft prompts.

After learning global soft prompts, all users in $\mathbf{U}$ then learn personal soft prompts, τ_s, to align the smaller model, LLM_s to their personal writing style or preferences. Each user is associated with their own set of soft prompts, learned strictly over their own personal data (i.e., each user learns and stores soft prompt parameters on a personal device, without accessing or sharing personal data with the wider population). In this work, soft prompts are learned with a language-modeling objective and a dataset of user-generated unstructured text, such as book reviews (Hou et al., 2024), though prior work demonstrates that such personalization parameters could also come from more conventional recommender systems (Huber et al., 2025). We find that our approach is more effective if all users start with the same initialization for τ_s as opposed to random initializations for each user's soft prompt tokens. Therefore, we sample a fixed set of tokens from a random Normal distribution (0 mean, unit variance) and use it as the initial soft prompt across all users.

Once all users have their associated soft prompts, they are frozen and we then train the projection network, ϕ. The goal of this network is to transform τ_s into a new set of soft prompts, $\hat{\tau}_L$ that modulate the outputs of a larger model, LLM_L. To train this projection network, the predicted soft prompts, $\phi(\tau_s) = \hat{\tau}_L$, are directly used by LLM_L to model a user's data, and the associated language modeling loss is back-propagated to the projection network ϕ. Note that *all* parameters of LLM_L are frozen and the τ_s parameters are frozen, meaning that the only way to reduce the language-modeling loss is to improve ϕ, the projection network, in translating τ_s into $\hat{\tau}_L$. The projection network therefore directly learns how to translate from τ_s into a $\hat{\tau}_L$ that is useful for LLM_L to reconstruct the user's original training data. We note that the numbers of small-model and large-model tokens are not necessarily coupled (i.e., $|\tau_s| \neq |\tau_L|$).

One could alternatively attempt to learn ϕ by first learning a ground truth soft prompt, τ_L on the large model and then learning a mapping from τ_s to τ_L. In practice, we found this method produced quantitative results on par with backpropagating through LLM_L, as described above. However, we found that this direct mapping method was significantly more computationally expensive because of the need to learn τ_L over so many users on such a large model. We present supplementary results for this direct mapping method in Appendix B and a visual side-by-side of the two methods in Figure 3.

4 Experiments & Results

As the primary objective of this work is to demonstrate personalization across different models *without* exposing personal data from each user, all experimental results are reported on $\mathbf{U}_W$, the users whose data are never exposed to the large model and who are not involved in learning the parameters of ϕ. For each task, users learn τ_s over a subset of their personal data, and results are reported over each user's test split (i.e., data that were not used to learn τ_s).

In our experiments, we choose small models under one billion parameters from either the Pythia (Biderman et al., 2023) or Qwen3 (Yang et al., 2025) family. These represent both base (Pythia) and instruction-tuned (Qwen) models, allowing us to experiment with how effectively our method is able to share knowledge across different architectures and pre/post-training recipes. We choose large models with between 8B and 14B parameters. Owing to computational constraints, we do not compute results for large models greater than 14B. Additional experiments and hyperparameters for all experimental conditions are reported in Appendix D.

Throughout our experiments we compare soft prompt projection to two fine-tuning baselines. **Large model prompts (GT Large):** Training soft prompts on the large model directly. This would involve violating user privacy, but should represent an upper-bound on performance. **Small model prompts (GT Small):** Using each user's small model soft prompts directly. This sacrifices large model capabilities, but should closely mirror the user's preferences up to the limits of the small model.

We also compare to three prompting baselines. **Inferred Personas:** Generating personas from a subset of training examples (Yang et al., 2023). Personas are an increasingly popular approach to personalization, although sending a persona may violate user privacy. We give this baseline an advantage by writing the persona using a Qwen3-30B model[1]. **In-context learning (ICL)** (Brown et al., 2020), using a random subset of training examples in-context. We note that this would also constitute a privacy violation, but is a common and strong prompting technique for LLM adaptation. **No personalization**, in which we consider the large model as-is, with no personalized adaptation.

Finally, we compare to a prior method for soft prompt transfer, **POST** (Wang et al., 2025a). **POST** relies on the small model being a distillation of the large model, a shared tokenizer across models, and a shared public dataset between all users. In our work, we do not use a distilled smaller model, meaning that POST is at a disadvantage relative to the original implementation. We do not apply **POST** across model backbones (e.g., from Pythia to Qwen) as the tokenizers are different, and therefore the method does not work. The public dataset for POST is drawn from our training users.

We consider two tasks: text generation in a user's voice and rating prediction in accordance with a user's preferences. We use the Amazon Review dataset (Hou et al., 2024), the Stack Exchange dataset (StackExchange, 2009), and MovieLens-1M (Harper & Konstan, 2015). For the text generation task, the LLM must either generate a text review (given a product title, description, and rating), or a StackExchange post (e.g., a question, an answer, or a comment from a title, a question, or an answer, respectively). For the rating prediction task, the LLM is given a title, description, and review (for Amazon data only), and must generate the rating. Finally, we conduct two experiments on introspection of the learned soft prompts, and an ablation study on personalization with purely global prompts and a shared projector.

We subsample each dataset to remove users with fewer than 100 samples and, for the Amazon dataset, products with fewer than 20 reviews. Dataset statistics are given in Appendix G. We cap each sample at 500 words, owing to memory constraints, and we discard samples with fewer than 10 words.

[1]Note this is an advantage to the inferred personas method, as we are not relying on the on-device (small) models, but are instead using a much larger "oracle" model.

Table 1: Text generation results on both datasets. **Projected** prompts outperform all baselines and even achieve lower perplexity than the **GT Large** baseline. Win-rate is reported as the average number of wins from all head-to-head comparisons. *Amazon POST results are computed over only the first 960 users due to computational constraints for the full dataset.

		Perplexity ↓		Win Rate (%) ↑	
Model	**Method**	**Amazon**	**StackExch.**	**Amazon**	**StackExch.**
Pythia-410M	GT Small	26.57	27.55	64.3	60.3
Qwen3-0.6B	GT Small	116.55	28.40	70.0	66.1
Pythia-12b	No Personalization	22.30	20.81	10.9	23.1
	ICL	27.09	31.23	26.8	20.6
	Inferred Persona	22.96	21.34	14.9	16.5
	POST*	20.39	22.26	46.6	31.6
	Projected (Pythia-410m)	18.08	18.09	80.2	75.7
	Projected (Qwen3-0.6B)	**17.16**	**17.56**	**81.2**	**79.3**
	GT Large (Upper Bound)	*16.49*	*16.66*	*91.7*	*85.9*
Qwen3-8B	No Personalization	91.12	52.58	13.6	22.0
	ICL	49.67	45.42	30.9	31.6
	Inferred Persona	156.86	60.84	27.2	13.7
	POST*	39.53	30.81	68.6	61.5
	Projected (Pythia-410m)	**17.77**	**16.53**	**83.0**	**86.1**
	Projected (Qwen3-0.6B)	18.19	16.54	81.4	85.1
	GT Large (Upper Bound)	*24.35*	*15.69*	*92.7*	*90.3*
Qwen3-14B	No Personalization	63.22	40.49	4.8	15.5
	ICL	42.26	40.86	48.9	41.2
	Inferred Persona	68.39	41.29	27.3	15.2
	POST*	37.62	29.33	28.5	52.0
	Projected (Pythia-410m)	**16.56**	**15.18**	**81.0**	**85.1**
	Projected (Qwen3-0.6B)	17.10	15.27	78.0	**85.1**
	GT Large (Upper Bound)	*20.38*	*14.31*	*91.5*	*89.6*

4.1 Review Writing

The objective of the text generation task is to write a review or a StackExchange post in a particular user's voice. This is scored with two metrics. First, we measure average perplexity across all users, which is a measure of how effectively the model has adapted to the particular task and dataset of users (where lower perplexity indicates better personalization). Second, we compute pairwise win-rates for all methods using an LLM-as-Judge setup with the ground truth text as a reference. All pairwise comparisons are done with orders swapped to check for position bias, and with a "tie" option. See Appendix H for details and the judging prompt.

Our results (Table 1) show that projected soft prompts lead to better personalization than all baselines, achieving substantially higher win rates than all baselines (Figure 2). Crucially, **Projected** prompts outperform the **GT Small** baseline, demonstrating the benefits of generation under a much larger model. **Projected** prompts are also outperforming the prompting baselines (**No Personalization**, **Inferred Persona**, and **ICL**), showing that the projected soft-prompts carry useful information and that the large model is able to exploit that information for personalized generation.

Comparing to the prompting baselines (**ICL** and **Persona**), we find that our approach leads to higher likelihood for the user's true reviews (i.e., lower perplexity) *and* superior review generations (Appendix J) and win-rates. Baselines tend to rely heavily on information in the prompt itself, often repeating the product name, review title, and rating within the review, and generating highly verbose and generic reviews, whereas our approach generates more realistic reviews that do not fixate on the product information. Comparing the mean and median length of generated text under the Pythia-12b model (Table 11) shows that the prompting baselines tend towards much longer text, while the fine-tuned methods more closely reflect the shorter length of the human references.

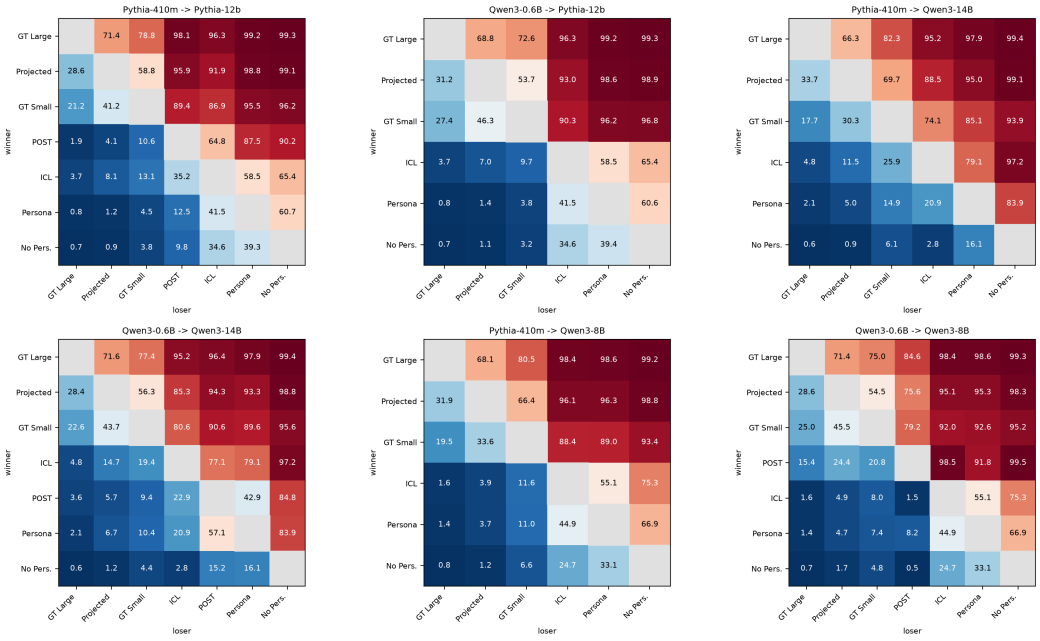


Figure 2: Pairwise win-rates on Amazon Reviews text generation. Darker red shows wins and darker blue shows losses. **POST** is only within-family due to limitations of the method. **GT Large** is always the top performer, with **Projected** always second and **GT Small** third, reinforcing that **Projected** prompts unify the personalization of a small model with the generation capabilities of a large model.

We present pairwise win rates across all methods for the Amazon dataset in Figure 2. This figure more clearly shows the gap between methods, with **GT Large** clearly exceeding all other methods, and **Projected** prompts winning over all baselines by a wide margin, and **No Personalization** consistently losing to all methods. Interestingly, despite generation coming from a very small model, we see that **GT Small** is often preferred over the prompting baselines. This result suggests that a well-personalized small model is better than a prompted large-model for generation in a user's voice.

Considering the **POST** baseline, there are several significant drawbacks that **Projected** prompts avoid. First, **POST** requires that the small model be a distilled version of the large model with a shared tokenizer. Conversely, prompt projection is agnostic to the underlying model, architecture, and tokenizer, meaning that our method can work across model families. Second, **POST** requires that large prompts be computed per user on the large model, running backprop through the large model repeatedly with a public dataset. In our experiments, this was extremely computationally demanding, and would have taken over 1600 hours to complete on the full test-split of the Amazon dataset on a 8x80GB A100 node. Owing to these costs, all **POST** results on the Amazon dataset are reported with a subset of the test-set containing only 960 users (the same size as the StackExchange test split). Additionally, the vast majority of all generated samples using **POST** are purely reasoning traces, which we remove before passing samples to the judge. **POST** is therefore evaluated on far fewer samples, leading to an inflated win-rate (we do not count invalid generations as failures, rather we only score **POST** on its valid generations). Even so, it underperforms on this task, suggesting that the method is either ill-suited to free-text generation, or heavily reliant on a distilled small model.

Overall, the pairwise win-rates and perplexity results both highlight the benefits of **Projected** prompts in their ability to combine the personalization of **GT Small** with the generation capacity of **GT Large**. Without transferring any personal data in plain text to the large model, **Projected** prompts are able to generate text that convincingly reflects the ground-truth reference for each user.

4.2 Review Rating Prediction

The objective of the rating prediction task is to predict a star rating for a particular user and item (movie or product). This task is scored as a classification task, in which there are five possible targets

(1–5 stars), and models are instructed to predict only a star rating. For this task, we use both the MovieLens dataset (Harper & Konstan, 2015) and the Amazon Review dataset (Hou et al., 2024). We report accuracy, macro F1, and mean absolute error (MAE) as a measure of how close predictions are.

We observe that prompting baselines (**ICL**, **Inferred Personas**, and **No Personalization**) and **POST** all struggle significantly with this task, and we therefore apply constrained decoding and allow baselines to only predict 1 token from the known answer-set (the set of tokens corresponding to integers from 1 to 5). We note that our method is not susceptible to this issue and therefore we do not give this constrained-decoding advantage to models using soft prompt projection (i.e., baselines have a decoding advantage over our method). This distinction means that the fine-tuned methods (**GT Large**, **GT Small**, and **Projected**) will occasionally generate invalid tokens (i.e., not integers). In such cases, we simply count the generations as incorrect and apply the maximum MAE penalty.

Results for both datasets are reported in Table 2. We again observe that **Projected** prompts lead to much better personalization than baseline prompting methods, often approaching or even matching the **GT Large** setting. In this domain, we also observe that **GT Small** is able to perform very well, and in some instances it exceeds **Projected** prompts. This strong performance was not present in the text generation task, and likely reflects the underlying task difficulty (i.e., a small model can perform rating prediction on par with a large model for these datasets).

We observe that baselines in this task are prone to generating one answer and generally ignoring the input context (for example, Qwen3-8B predicts "1" or "2" for nearly all samples using **No Personalization**), leading to extremely poor performance. For Qwen3, this is likely due to an overemphasis within the model on reasoning, leading to extremely low likelihood on any individual integer token response. Pythia does better, but interestingly performs best with no personalized prompting (i.e., without inferred personas or in-context examples). **Projected** prompts outperform all prompting baselines and show great promise in personalization to user preferences and rating conventions, achieving upwards of 60% F1 on this five-class classification task and low MAE.

Finally, we observe that **Projected** prompts lead to far fewer parsing failures (i.e., invalid generations) than **GT Large** and **GT Small**, despite all three using unconstrained generation. Specifically, using **GT Large** and **GT Small** we observe 1075, 59, and 318 parsing failures under Qwen3-14B, Qwen3-8B, and Qwen3-0.6B respectively (interestingly, Pythia models do not exhibit this failure mode, as most failures are attempts at generating thought traces). However, under **Projected** prompts, we observe only 15 total failures across all models. We speculate that this is because the projector is trained over tens of thousands of users' worth of data, far more than any individual user has access to. The projector can therefore learn population-level rules, such as "suppress thinking", meaning that all **Projected** prompts can carry both personal preference information *and* global task information.

4.3 Prompt Introspection

The primary aim of this work is to enable personalization on an LLM without ever needing to expose personal data to the LLM. However, if users are sending learned soft prompts that the LLM can use to provide a personalized review or product rating, it is natural to suspect that the prompts themselves may contain personal information that a user may not have intended to share. We therefore conduct a set of experiments to see how much identifiable or user-specific information can be extracted from learned soft prompts. For this investigation, all experiments are conducted on the Pythia-410M to Qwen3-14B mapping and the Amazon Reviews dataset.

For each user in the test set, we first ask five questions with the prompts appended, in an effort to get the model itself to decode the target prompts. These are passed to both the small model with the **GT Small** prompts and to the large model with the **Projected** prompts. We then search for a user's ground truth text in the generated responses (i.e., did the model generate any of the user's original data). We find that nine users in the **GT Small** setting generated original data, while no users in the **Projected** setting generated their original data. The generated original data are all samples that users include in many of their reviews, and therefore may be modeled by **GT Small**. However, when pooling across all of $\mathbf{U}_T$ to learn ϕ, such data are too infrequent to memorize, meaning that they do not get repeated by **Projected** prompts. The full set of repeated data is in Appendix I.

Finally, we conduct an experiment in which we look up the nearest neighbor of all prompt tokens in the small and large model embedding space, but this experiment did not reveal any useful patterns

Table 2: Rating prediction results on both datasets. **Projected** prompts match or exceed baselines in all settings and even match the **GT Large** upper bound in several settings. Baselines use constrained decoding while **Projected**, **GT Large**, and **GT Small** methods count generation failures as mistakes.

		Accuracy ↑		Macro-F1 ↑		MAE ↓	
Model	**Method**	**Amazon**	**MovieLens**	**Amazon**	**MovieLens**	**Amazon**	**MovieLens**
Pythia-410M	GT Small	0.80	0.40	0.62	0.32	0.23	0.82
Qwen3-0.6B	GT Small	0.81	0.38	0.64	0.31	0.22	0.84
Pythia-12b	No Personalization	0.68	0.24	0.34	0.17	0.46	1.45
	ICL	0.68	0.31	0.34	**0.26**	0.46	1.08
	Inferred Persona	0.65	0.24	0.29	0.16	0.63	1.49
	POST (Pythia-410m)	0.40	0.33	0.21	0.15	0.71	0.98
	Projected (Pythia-410m)	**0.80**	**0.38**	**0.63**	**0.26**	**0.22**	**0.83**
	Projected (Qwen3-0.6B)	0.68	0.34	0.24	0.11	0.46	0.92
	GT Large (Upper Bound)	*0.81*	*0.41*	*0.66*	*0.34*	*0.20*	*0.80*
Qwen3-8B	No Personalization	0.03	0.12	0.02	0.05	2.85	1.70
	ICL	0.22	0.34	0.07	0.10	0.85	0.91
	Inferred Persona	0.23	0.33	0.10	0.12	0.97	0.95
	POST (Qwen3-0.6B)	0.10	0.14	0.07	0.09	1.95	1.61
	Projected (Pythia-410m)	**0.80**	**0.41**	**0.64**	**0.36**	**0.21**	**0.78**
	Projected (Qwen3-0.6B)	0.77	0.36	0.62	0.19	0.25	0.82
	GT Large *(Upper Bound)*	*0.81*	*0.42*	*0.66*	*0.37*	*0.20*	*0.77*
Qwen3-14B	No Personalization	0.22	0.34	0.07	0.10	0.85	0.91
	ICL	0.22	0.34	0.08	0.11	0.85	0.91
	Inferred Persona	0.22	0.34	0.07	0.10	0.89	0.91
	POST (Qwen3-0.6B)	0.23	0.29	0.11	0.11	1.19	1.18
	Projected (Pythia-410m)	**0.81**	**0.40**	**0.65**	**0.28**	0.20	**0.76**
	Projected (Qwen3-0.6B)	0.81	0.39	0.65	0.26	**0.20**	0.79
	GT Large (Upper Bound)	*0.81*	*0.42*	*0.65*	*0.38*	*0.22*	*0.77*

or insights into the underlying prompt tokens. To further investigate the possibility of a prompt conveying memorized information, we conduct an experiment on a synthetic dataset that specifically targets memorization, which we explain in Appendix A. This experiment reveals that memorization is possible, although it may require significantly more targeted datasets than the free-text of StackExchange and Amazon Reviews. Overall, our introspection experiments suggest that **Projected** prompts do not meaningfully expose personal information.

5 Discussion

Our experiments show that it is possible to project personalization parameters across model scales and families. **Projected** prompts yield personalized generations from models with 400M parameters up to models with 14B parameters, representing a 35-fold parameter count increase while still preserving the style and preferences of each user. We show qualitative generation examples in Appendix J which demonstrate that a small model cannot generate coherent text, a large model does not personalize to the user's request, but **Projected** prompts enable personalized responses from a large model. **Overall, we show that prompt projection enables personalized generation across model scales, combining the personalization of a small model with the generation capability of a large model.**

Considering results such as the parsing failures in the rating prediction task (Section 4.2), we find that the projection network, ϕ, acts as an implicit regularizer by constraining projected prompts to satisfy generations across all training users, $\mathbf{U}_T$, thereby preventing individual prompts from drifting into degenerate regions of the latent space. This regularization may also explain why projected prompts enable instruction-following on base models (Pythia) and suppress reasoning on thinking models (Qwen), as the projection network learns a shared task prior that steers the target model toward the desired output format regardless of the model's default behavior.

While prompt projection leads to strong personalization results and strong task-alignment under the large model, there is a significant amount of shared information across all users. In particular, the globally shared prompt tokens and the projector itself are globally learned across all training users, and the gains on all tasks may simply be down to this shared task-level information rather

than any personalized preferences. We therefore conduct an experiment in which all test user's *personal* prompt tokens are zeroed out and passed through the projector, meaning that the large model generation is using the same shared projected prompt for all users (Appendix C).

This investigation reveals that the task difficulty and the model pair determine when personal prompts are necessary. Specifically, we find that global prompts do capture meaningful task information and that they perform almost on par with global + personal prompts for rating prediction, ranging from parity to a 10% deficit in macro-F1 score. However, for text generation, global-only prompts lose out to global + personal prompts between 55-96% of the time in a head-to-head win rate experiment, showing that personal prompts are necessary for finer-grained details in a more complex task.

As our objective is to avoid sharing personal information with a large model, we further show that such prompts cannot be used to trivially reverse-engineer or extract personal data. Prior work on decoding soft prompts suggests that such an approach would require a dedicated training set and full fine-tuning (Tennenholtz et al., 2023) (e.g., mapping τ_s to natural-language descriptions of each user and fine-tuning a model to generate such descriptions from the prompts).

To further support this claim, we conduct an experiment in which we explicitly target memorization across models (Appendix A) with a synthetic dataset geared towards recall of personal facts. We find that generalizable information extraction requires greater prompt capacity (i.e., more soft prompt tokens than we need for rating prediction), a constrained space of total facts, and a very large dataset (at least 40,000 users for 80% recall of ten facts). Interestingly, we find that these constraints appear to be independent of the large model scale, as the results were not improved by scaling the large model from 1.5B to 14B parameters. This result suggests that the projector is the bottleneck and that it requires extensive repetition to permit memorization.

In a study of the effects of the size of τ_s and τ_L, we find that optimal token counts vary by task (Appendix E). A more challenging task, such as review writing, benefits from additional large and small tokens, whereas a simpler task sees diminishing returns as more tokens are added. We similarly find that including globally shared tokens is crucial for a more challenging task. Our results indicate that projecting prompts *across* model families performs comparably to projection *within* model families, despite the change in model vocabulary, latent space, training stage, and pretraining datasets, as the projector itself is agnostic to the input and output tokens.

Finally, we conduct an experiment comparing two ways of learning the projector itself (Appendix B), either by backpropagation through the large model (Section 3) or by first learning ground-truth soft prompts on the small and large models, and then learning a direct mapping between the converged sets of prompts. We find that there is not much difference between the two methods in terms of final performance, though the results suggest that the direct mapping method may be superior for capturing minority preferences. However, direct mapping requires learning ground-truth large prompts for all users, which is significantly more computationally demanding. As a result, the direct mapping method is considerably more expensive than the backpropagation method.

6 Conclusion

In this work, we have introduced the problem of personalization across model scales, in which a user's personal data should be kept on-device while still enabling personalized responses from a larger model. We have presented an approach to solve this problem through soft prompt projection, in which a user's preferences are encoded into a set of soft prompts which can be passed to the server side model through a projection network, conditioning the larger model to produce a personalized response while preserving the user's privacy. We have shown that this approach outperforms prompt-based personalization on the large models, and that no personal data are leaked through the soft prompts.

There are several exciting avenues for future work. First, there is no limitation on what can be projected, and future work may consider other parameter efficient fine-tuning approaches (such as LoRAs (Hu et al., 2022)) as candidates for projected personalization parameters. Second, we have found that direct memorization of facts is extremely expensive (Appendix A), though this may be necessary to solve certain tasks (e.g., remembering a user's birthday when asked to plan a party), and so future work may consider how to better preserve information. Additionally, this work has assumed access to a training set of users ($\mathbf{U}_T$), though future work may consider settings in which the projection network must be learned privately across a population of users.

AI USE STATEMENT

In this work, we have used generative AI tools to: implement methods, create figures (through writing plotting code), create or edit software code (launch scripts and sweeps), and to drive results where explicitly stated (e.g. using `Qwen3.8-27B` as a judge for win-rates between personalized text generation, and using `Qwen3-30B` to infer user personas). LLMs were used to assist in coding for evaluations, in particular for aggregating results, although results were always checked by authors to ensure correctness, and the code for core results in the paper (perplexity, text generation results, and F1, as in Tables 1 & 2) was written and run manually by an author. LLMs were also used to provide feedback on the overall draft, identifying areas where concepts were re-stated or terms were being used interchangeably, so that the authors could improve the clarity of the draft.

We have not used generative AI to generate synthetic datasets, propose or refine hypotheses, design or provide feedback on research methodology or experiments, assist with translation, clean and reformat datasets, support qualitative and thematic data analysis (except using LLM judges, as stated above), or to interpret results. We have also not used generative AI to suggest experimental parameters, create artifacts, draft parts of our research paper, transcribe recordings of research material, summarize or analyze existing literature, discover research topics or identify gaps, brainstorm, source/search for information (beyond the codebase of the project itself), edit a research paper to improve readability (apart from identifying areas that could be improved, as stated above), identify relevant literature, format references, suggest a structure for the research paper, or to propose a title or keywords for a research paper.

The following tasks were not relevant to this work and we therefore did not apply generative AI to assist us: to develop theoretical models or frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, formulate questions for surveys or interviews,

We take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We have included all method hyper-parameters, and a description of our training approach with validation-based stopping, in Appendix D. Our computational resources are documented in Appendix F, and our method is explained in Section 3.

REFERENCES

Krisztian Balog, Filip Radlinski, and Shushan Arakelyan. Transparent, scrutable and explainable user models for personalized recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, pp. 265–274, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450361729. doi: 10.1145/3331184.3331211. URL https://doi.org/10.1145/3331184.3331211.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pp. 2397–2430. PMLR, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Hyundong Justin Cho, Karishma Sharma, Nicolaas Paul Jedema, Leonardo F. R. Ribeiro, Jonathan May, and Alessandro Moschitti. Tuning-free personalized alignment via trial-error-explain in-context learning. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 5879–5900, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.326. URL https://aclanthology.org/2025.findings-naacl.326/.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4), December 2015. ISSN 2160-6455. doi: 10.1145/2827872. URL https://doi.org/10.1145/2827872.

Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952*, 2024.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Bernd Huber, Ghazal Fazelnia, Andreas Damianou, Sebastian Peleato, Max Lefarov, Praveen Ravichandran, Marco De Nadai, Mounia Lalmas-Roellke, and Paul N Bennett. Embedding-to-prefix: Parameter-efficient personalization for pre-trained large language models. In *AI That Keeps Up: NeurIPS 2025 Workshop on Continual and Compatible Foundation Model Updates*, 2025.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.243. URL https://aclanthology.org/2021.emnlp-main.243/.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Haoran Li, Yangqiu Song, and Lixin Fan. You don't know my favorite color: Preventing dialogue representations from revealing speakers' private personas. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5858–5870, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.429. URL https://aclanthology.org/2022.naacl-main.429/.

Suhong Moon, Marwa Abdulhai, Minwoo Kang, Joseph Suh, Widyadewi Soedarmadji, Eran Kohen Behar, and David M. Chan. Virtual personas for language models via an anthology of backstories. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 19864–19897, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1110. URL https://aclanthology.org/2024.emnlp-main.1110/.

Lin Ning, Luyang Liu, Jiaxing Wu, Neo Wu, Devora Berlowitz, Sushant Prakash, Bradley Green, Shawn O'Banion, and Jun Xie. User-llm: Efficient llm contextualization with user embeddings. In *Companion Proceedings of the ACM on Web Conference 2025*, pp. 1219–1223, 2025.

Fred Philippy, Siwen Guo, Shohreh Haddadan, Cedric Lothritz, Jacques Klein, and Tegawendé F. Bissyandé. Soft prompt tuning for cross-lingual transfer: When less is more. In Raúl Vázquez, Timothee Mickus, Jörg Tiedemann, Ivan Vulić, and Ahmet Üstün (eds.), *Proceedings of the 1st Workshop on Modular and Open Multilingual NLP (MOOMIN 2024)*, pp. 7–15, St Julians, Malta, March 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.moomin-1.2. URL https://aclanthology.org/2024.moomin-1.2/.

Jingyan Shen, Jiarui Yao, Rui Yang, Yifan Sun, Feng Luo, Rui Pan, Tong Zhang, and Han Zhao. MiCRo: Mixture modeling and context-aware routing for personalized preference

learning. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 17447–17463, Suzhou, China, November 2025a. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.882. URL https://aclanthology.org/2025.emnlp-main.882/.

Zhanming Shen, Tianqi Xu, Hao Wang, Jian Li, and Miao Pan. pFedGPT: Hierarchically optimizing LoRA aggregation weights for personalized federated GPT models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 4766–4778, Suzhou, China, November 2025b. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.239. URL https://aclanthology.org/2025.emnlp-main.239/.

StackExchange. Stackexchange data dump attribution. Stack Overflow Blog, June 2009. URL https://stackoverflow.blog/2009/06/25/attribution-required/. Accessed: 2026-09-16.

Yusheng Su, Xiaozhi Wang, Yujia Qin, Chi-Min Chan, Yankai Lin, Huadong Wang, Kaiyue Wen, Zhiyuan Liu, Peng Li, Juanzi Li, et al. On transferability of prompt tuning for natural language processing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3949–3969, 2022.

Seungjong Sun, Seo Yeon Baek, and Jang Hyun Kim. Personality vector: Modulating personality of large language models by model merging. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 24656–24677, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1253. URL https://aclanthology.org/2025.emnlp-main.1253/.

Guy Tennenholtz, Yinlam Chow, Chih-Wei Hsu, Jihwan Jeong, Lior Shani, Azamat Tulepbergenov, Deepak Ramachandran, Martin Mladenov, and Craig Boutilier. Demystifying embedding spaces using large language models. *arXiv preprint arXiv:2310.04475*, 2023.

Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou, and Daniel Cer. Spot: Better frozen model adaptation through soft prompt transfer. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 5039–5059, 2022.

Xun Wang, Jing Xu, Franziska Boenisch, Michael Backes, Christopher A. Choquette-Choo, and Adam Dziedzic. Efficient and privacy-preserving soft prompt transfer for LLMs. In *Forty-second International Conference on Machine Learning*, 2025a. URL https://openreview.net/forum?id=WnTGpncrxK.

Yaxuan Wang, Quan Liu, Zhenting Wang, Zichao Li, Wei Wei, Yang Liu, and Yujia Bao. Promptbridge: Cross-model prompt transfer for large language models. *arXiv preprint arXiv:2512.01420*, 2025b.

Emily Xiao, Chin-Jou Li, Yilin Zhang, Graham Neubig, and Amanda Bertsch. Efficient many-shot in-context learning with dynamic block-sparse attention. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31946–31958, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1542. URL https://aclanthology.org/2025.acl-long.1542/.

Zhaozhuo Xu, Zirui Liu, Beidi Chen, Shaochen Zhong, Yuxin Tang, Jue Wang, Kaixiong Zhou, Xia Hu, and Anshumali Shrivastava. Soft prompt recovers compressed llms, transferably. In *Forty-first International Conference on Machine Learning*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Fan Yang, Zheng Chen, Ziyan Jiang, Eunah Cho, Xiaojiang Huang, and Yanbin Lu. Palr: Personalization aware llms for recommendation. *arXiv preprint arXiv:2305.07622*, 2023.

Elif Ceren Gok Yildirim, Murat Onur Yildirim, and Joaquin Vanschoren. Slim adaptation modules: A simple yet strong baseline for continual foundation models. In *AI That Keeps Up: NeurIPS 2025 Workshop on Continual and Compatible Foundation Model Updates*, 2025.

A FACT MEMORIZATION

To probe the limits of soft prompt personalization, we constructed a synthetic fact memorization task. We generated synthetic users, each defined by 10 categorical facts: name, hometown, current city, profession, favorite color, favorite animal, favorite food, favorite drink, hobby, and favorite soccer team. Each fact is sampled from a predefined set and we remove duplicate users, so all combinations are unique. Each user's dataset consists of 10 question-answer pairs (e.g., "What is your name?" $\rightarrow$ "Angelo", "What is your favorite color?" $\rightarrow$ "red").

We first trained personal soft prompts on each user's 10 facts and found that the task is can easily be perfectly solved (we can achieve perfect accuracy and memorize a user's facts) with a variety of settings for $|\tau_s|$ and $|\hat{\tau_L}|$, where LLM_s was Qwen2.5-0.5B and LLM_L was Qwen2.5-1.5B. Next, we trained a projection network with 500 held out users, which proved to be surprisingly difficult. With 8000 training users, we achieved 92.3% accuracy per fact, but only 45.6% accuracy in jointly predicting a user's full ten facts. Increasing to 24000 users, we climbed to 95.6% per fact and 66.6% for the joint prediction, and increasing again to 48000 users reached 98.4% per fact and 85.5% for the joint combination. Scaling LLM_L up to Qwen2.5-14B resulted in 97.9% per fact and 83.6% for the joint user, suggesting that the projection was largely determined by dataset size, not by model scale.

Interestingly, we found that the 24000 user setting *regressed significantly* if we dropped the token counts. While the above results were obtained with 16 personal tokens and 16 global tokens, dropping to 10 personal tokens and 5 global tokens brought single-task accuracy down to 88.3% and joint set accuracy to 34.0%. This suggests that parameter capacity is crucial, even for this fairly trivial memorization task. We also note that removing the shared initialization set for all users (i.e., initializing all soft prompts randomly, rather than using one constant prompt for all users), made memorization impossible, regardless of the dataset scale or the simplicity of the task. The projector would either perfectly fit the training data and perform at random chance for validation users, or it would perform at random chance for all users.

This result suggests that prompt projection *can* memorize information across model scales, but that the problem must be very narrowly defined and must have a significantly large dataset and large prompt parameter capacity, with shared initial prompts across all users. While this memorization-difficulty may be desirable for some tasks (such as text generation in a user's voice without revealing personal information), it may be a hindrance for tasks that require specific facts (e.g., recalling a user's exact birthday to help plan a party).

B DIRECT MAPPING VS. BACKPROPAGATION THROUGH THE LARGE MODEL

An alternative to learning ϕ by backpropagating through LLM_L is to first learn ground truth soft prompts on the large model (τ_L) and then train ϕ via direct regression from τ_s to τ_L in embedding space, minimizing $\|\phi(\tau_s) - \tau_L\|^2$. We refer to this as the **direct mapping** method. Visual overviews of the two methods are presented side-by-side in Figure 3, and an experiment on the rating prediction task on the MovieLens-1M dataset is shown in Table 3.

The results in Table 3 suggest that direct mapping achieves competitive performance with directly backpropagating through LLM_L, even outperforming on minority classes. Unfortunately, learning the direct mapping requires first learning ground truth large prompts for all training set users, which would require learning hundreds of prompt tokens across tens of thousands of users and millions of samples. For our Amazon Reviews dataset, this would have been roughly 5 weeks on an 8x80GB A100 node, just to learn the large prompts for all training users. In contrast, backpropagating through the large model converged with just 1.5 days on the Amazon Reviews dataset.

C GLOBAL PROMPT ABLATION

To compare the relative gains of personal prompt parameters, we conduct an experiment in which we set all personal prompt parameters to zero, sending the global prompt + a zero vector through the projector and using that for all generations in the rating prediction task on the Amazon Review rating prediction setup. Results are presented in Table 4 for rating prediction, and Table 5 for text generation.

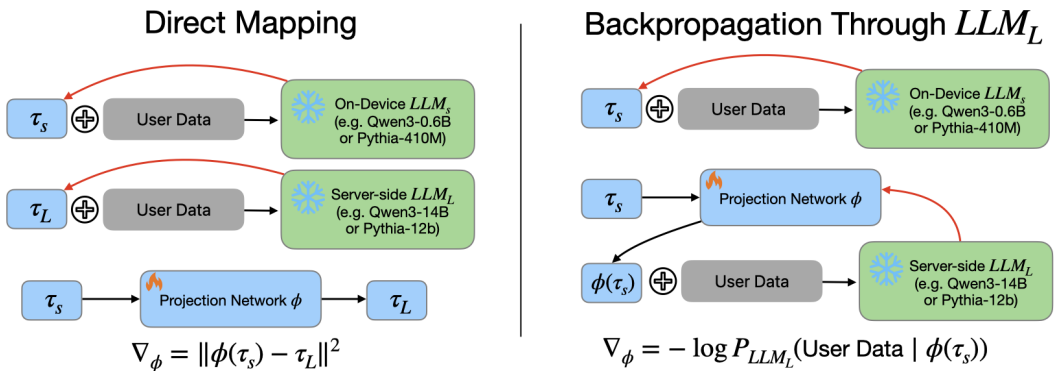


Figure 3: Visual overviews of training the projection network via direct mapping (left) vs. through backpropagation with the large model (right). The direct mapping approach first learns both sets of prompts and then learns to minimize the distance between predicted prompts and real prompts. The backpropagation method instead directly maximizes the likelihood of the user's data under the predicted prompt, leading to superior performance.

Table 3: Comparison of two methods for learning a projection network on the rating prediction task (MovieLens-1M). Direct mapping achieves higher F1 and lower accuracy, suggesting it is better at capturing the minority classes than backpropagating through the large model.

Small model	**Large model**	**Accuracy** ↑	**Macro-F1** ↑	**MAE** ↓
Backprop through the frozen large model				
Qwen3-0.6B	Pythia-12b	0.3512	0.1224	0.8676
Qwen3-0.6B	Qwen3-14B	**0.4011**	0.1889	**0.7691**
Pythia-410M	Pythia-12b	**0.3917**	0.1656	**0.7928**
Pythia-410M	Qwen3-14B	**0.4102**	0.2163	**0.7503**
Direct mapping				
Qwen3-0.6B	Pythia-12b	**0.3950**	**0.2048**	**0.8262**
Qwen3-0.6B	Qwen3-14B	0.3838	**0.2468**	0.7925
Pythia-410M	Pythia-12b	0.3849	**0.2159**	0.7998
Pythia-410M	Qwen3-14B	0.3888	**0.2494**	0.7826

In nearly all configurations on the rating prediction task, global prompts perform very well, but lag behind the full global + personal prompt setting. However, the strong performance of global-only prompts suggests that there is not a significant need for personalization on these datasets, and that the task can mostly be accomplished with population-level trends.

In contrast to this result, text generation appears to rely much more heavily on personal soft prompt tokens, as shown in Table 5. When global-only generations are compared head-to-head against global + personal prompts, the `Qwen3.8-27B` judge chooses chooses global + personal prompts 55-96% of the time, depending on the dataset and model mapping. We also observe degradation with global-only prompts in the text-generation condition, with some conditions, such as Qwen3-0.6B → Qwen3-8B on the Amazon Reviews dataset, producing nearly 100% thought traces and no valid generations (this condition yielded only 85 valid samples to compare across the two settings, out of a possible 191,545).

We observe that the same model pair, pythia-410m→pythia-12b, shows better perplexity under the **Global Only** setting despite clearly superior win rates user global + personal prompts. This result highlights the importance of using free-text generations and judges for measuring personalized text generation, as the global-only prompts may be superior for maximizing the likelihood of common or generic text, but they are insufficient for generating diverse personalized text for different voices and users. Interestingly, we observe that the methods with high numbers of invalid samples (i.e., Qwen3-0.6B → Qwen3-8B and Qwen3-0.6B → Qwen3-14B) show drastically lower perplexity when conditioning on global + personal prompts, suggesting that these model pairs may be encoding some

of the task-specific knowledge (i.e., disabling reasoning) into personal parameters rather than purely in global parameters.

Table 4: Global prompt results on Amazon rating prediction with personal prompts set to zero.

		Accuracy ↑		Macro-F1 ↑	
Small	**Large**	**Full**	**Global only**	**Full**	**Global only**
Pythia-410M	Qwen3-14B	0.8072	0.7636	0.6569	0.6158
Pythia-410M	Qwen3-8B	0.8068	0.7752	0.6285	0.6119
Pythia-410M	Pythia-12b	0.7933	0.7607	0.6144	0.5245
Qwen3-0.6B	Qwen3-8B	0.7526	0.7533	0.5137	0.5159
Qwen3-0.6B	Qwen3-14B	0.8012	0.7984	0.6389	0.5763

D Experimental Hyperparameters

Table 6 reports the best hyperparameter configuration for each model pair, selected from over 200 sweep configurations on the Amazon Reviews books split. Review writing is evaluated by the perplexity gap between a user's own prompt vs. another user's prompt for text generation ("swap gap", higher is better) and rating prediction by F1 on a validation set. All configurations use an attention-based encoder architecture for the projection network ϕ, with 2 layers, 24 heads, and a hidden dimension of 6144. All soft prompts and projection models are trained with dev-based early stopping, in which a validation split of the training data is used to regularly compute a validation loss, which determines when to stop training. We compute the validation loss every 1000 steps, and we stop training when dev loss has not improved for 15 consecutive checkpoints. After stopping training, we restore the parameters of the best performing checkpoint.

Several patterns emerge. For text generation, the best configurations consistently use a low learning rate (1e-5) and more training epochs, with Pythia-410m as the small model benefiting from larger token budgets (100 personal, 100 global). For rating prediction, the optimal configurations converge on 10 personal tokens and fewer training epochs across all model pairs, with higher learning rates (1e-4 to 1e-2) likely due to rating prediction being an easier task. We find that the rating prediction task is generally robust to hyperparameter selection, while review writing can be more sensitive.

E Hyperparameter Sweeps

We conduct ablation studies across more than 200 hyperparameter configurations across 6 model pairs (2 small $\times$ 3 large) on the Amazon Reviews books split to understand the sensitivity of our method. All ablations report the perplexity gap between a user's own prompt vs. another user's prompt for text generation ("swap gap", higher is better) and F1 for rating prediction (higher is better), averaged across all model pairs within each comparison. We sweep over the number of personal tokens on the small model ($|\tau_s| \in \{1, 10, 100\}$), global tokens ($|\tau_{\text{global}}| \in \{0, 10, 100\}$), projected tokens on the large model ($|\hat{\tau}_L| \in \{1, 10, 100\}$), learning rate ({1e-5, 1e-4, 1e-3, 1e-2}). Due to the computational cost of each configuration (each requiring training soft prompts and a projection network end-to-end through the large model), we first conducted a broad sweep over a single model pair (Pythia-410m $\rightarrow$ Qwen3-8B) and then evaluated the top 24 configurations across all 6 model pairs.

Token Counts Table 7 shows the results of varying $|\tau_s|$ and $|\hat{\tau}_L|$ independently. For the review writing task, performance increases as the number of tokens increases, suggesting that this task benefits from more information being available to the model. For the rating prediction task, performance plateaus around 10 tokens for the small and large model, and even degrades slightly as we add more tokens. This suggests that the optimal number of soft prompt tokens depends on the task complexity, with more challenging tasks benefiting from more informative prompts.

Global Prompts. Table 8 shows the interaction between global tokens ($|\tau_{\text{global}}|$) and personal tokens ($|\tau_s|$). For review writing, global tokens provide essential task-level information, and performance

Table 5: Global prompt results on both text generation datasets, where each **Projected** prompt is compared against its **Global-Only** counterpart. * Win rate is over an incomplete dataset, as some of the **Global-Only** samples were purely reasoning traces. ** Fewer than 5% of possible generations yielded valid samples with **Global-Only**, with over 95% being purely reasoning traces, so these win-rates are over a much smaller dataset.

		Win rate		Perplexity ↓	
Dataset	**small → large**	**Projected**	**Global only**	**Projected**	**Global only**
StackExchange	pythia-410m → pythia-12b	**0.614**	0.386	18.09	**17.93**
	pythia-410m → Qwen3-8B	**0.583**	0.417	**16.53**	17.10
	pythia-410m → Qwen3-14B	**0.554**	0.446	**15.18**	15.37
	Qwen3-0.6B → pythia-12b	**0.848**	0.152	**17.56**	19.21
	Qwen3-0.6B → Qwen3-8B*	**0.962**	0.038	**16.54**	21.44
	Qwen3-0.6B → Qwen3-14B**	**0.886**	0.114	**15.27**	30.47
Amazon Reviews	pythia-410m → pythia-12b	**0.770**	0.230	18.08	**17.91**
	pythia-410m → Qwen3-8B*	**0.742**	0.258	**17.77**	24.51
	pythia-410m → Qwen3-14B	**0.684**	0.316	**16.56**	17.72
	Qwen3-0.6B → pythia-12b	**0.934**	0.066	**17.16**	18.02
	Qwen3-0.6B → Qwen3-8B**	**0.780**	0.220	**18.19**	50.21
	Qwen3-0.6B → Qwen3-14B**	**0.786**	0.214	**17.10**	52.22

Table 6: Best hyperparameter configuration per model pair.

Task	**Model Pair**	$\lvert\tau_s\rvert$	$\lvert\tau_{\text{global}}\rvert$	$\lvert\hat{\tau}_L\rvert$	**LR**
Writing	Pythia-410m → Qwen3-8B	100	100	100	1e-5
	Pythia-410m → Pythia-12B	1	10	10	1e-5
	Pythia-410m → Qwen3-14B	100	100	100	1e-5
	Qwen3-0.6B → Qwen3-8B	100	10	10	1e-5
	Qwen3-0.6B → Pythia-12B	100	100	10	1e-5
	Qwen3-0.6B → Qwen3-14B	100	10	10	1e-5
Ratings	Pythia-410m → Qwen3-8B	10	10	1	1e-5
	Pythia-410m → Pythia-12B	10	100	10	1e-5
	Pythia-410m → Qwen3-14B	10	10	1	1e-5
	Qwen3-0.6B → Qwen3-8B	10	10	1	1e-2
	Qwen3-0.6B → Pythia-12B	10	10	10	1e-4
	Qwen3-0.6B → Qwen3-14B	10	10	10	1e-5

drops when they are omitted. In contrast, for rating prediction, global tokens do not seem as important for personalizing to a user's style.

Qualitatively, we find that global tokens are essential for adapting Qwen3 models, as these models are likely to generate ⟨think⟩ tags in response to every query. By first learning global task-specific tokens, we obtain a task-specific conditioning that will break the model out of ⟨think⟩ mode and into review-writing mode, which can then be personalized to each user's voice.

Cross Family Projection Across all configurations, within-family model pairs (i.e., Qwen → Qwen, Pythia → Pythia) and across-family pairs (i.e., Pythia → Qwen, Qwen → Pythia) achieve comparable performance. In the review writing task, within-family pairs achieve a mean swap perplexity of 0.51 ($n = 42$) compared to 0.55 ($n = 75$) for across-family configurations. For rating prediction, within-family achieves a mean F1 of 0.56 ($n = 72$) vs. 0.56 ($n = 103$) for across-family. These results suggest that the projection network ϕ is able to learn projections across vocabularies, tokenizers, scales, and latent spaces, rather than relying on memorizing artifacts of a particular model's architecture or tokenizer.

Table 7: Token decoupling ablation: mean performance when varying the number of small-model personal tokens ($|\tau_s|$, rows) and projected large-model tokens ($|\hat{\tau}_L|$, columns). Values are averaged across all model pairs and parameters.

	Text (Swap Gap ↑)			Ratings (F1 ↑)		
$\|\tau_s\| \setminus \|\hat{\tau}_L\|$	**1**	**10**	**100**	**1**	**10**	**100**
1	0.00	0.58	**0.72**	0.56	0.56	0.57
10	0.01	0.00	0.00	0.55	**0.58**	0.54
100	0.54	0.43	0.68	0.56	0.54	0.48

Table 8: Global × personal token interaction: mean performance when varying personal tokens ($|\tau_s|$, rows) and global tokens ($|\tau_{\text{global}}|$, columns). Values are averaged across all models and other parameters.

	Text (Swap Gap ↑)			Ratings (F1 ↑)		
$\|\tau_s\| \setminus \|\tau_{\text{global}}\|$	**0**	**10**	**100**	**0**	**10**	**100**
1	0.55	0.65	0.59	0.54	0.56	0.58
10	0.00	0.00	0.00	0.53	**0.58**	0.55
100	−0.19	**1.04**	0.84	0.55	0.52	0.50

F Compute Costs

All experiments were conducted on an 8×80GB A100 node. Hours of use on the node are reported in Table 9

Table 9: Training cost in wall-clock hours on a single 8×A100-80GB node. **Projected** pays a fixed cost per dataset-model pair, and generalizes to new users at no added cost. POST numbers are reported over a subset of only 960 users, as the method would have been infeasible on all 6700 test set users in the Amazon dataset.

Method	Amazon (Text)	StackExchange	Amazon (Ratings)	MovieLens
Projected (mapper)	36	123	30	24
GT Large	93	84	36	58
GT Small	72	180	42	30
POST	235	270	62	24

G Dataset Statistics

We use several splits of the Amazon Reviews dataset (Hou et al., 2024), including: Beauty and Personal Care, Electronics, Books, Clothing Shoes and Jewelry, Health and Household, Health and Personal Care, Kindle Store, Movies and TV, Office Products, Tools and Home Improvement, Video Games, and Unknown. Each split is filtered to retain only users with at least 100 verified reviews and products with at least 20 reviews. We discard reviews with fewer than 10 words, and reviews are truncated to a max of 500 words. The same filtered data are used for both the review writing and rating prediction tasks, with prompts/targets modified to suit the task needs. We combine topics with many samples in the StackExchange data dump StackExchange (2009), including 44 topics such as English, Gaming, DIY, Travel, Music, Sports, Freelancing, Fitness, Movies, etc. Dataset statistics are given in Table 10.

Users are split 80/10/10 into train, validation, and test sets by user, such that all reviews from a given user appear in exactly one split and no user is present across multiple splits. The projection network ϕ is trained using only train-split users ($\mathbf{U}_T$), and all reported results are on test-split users ($\mathbf{U}_W$) whose data are never exposed to the large model.

Table 10: Dataset statistics after filtering.

	Amazon Reviews	StackExchange	MovieLens-1M
Users	67,515	9,583	2,934
Items	2,194,296	44 sites	2,978
Total samples	8,533,297	5,945,046	840,905
Avg. samples / user	126.4	620.4	286.6
Avg. length (words)	71.4	91.3	—
Users evaluated in this work	6,748	959	294

To train each user's soft prompts, we use 80% of the user's available data, with the remaining 20% used for testing and evaluation.

Review writing prompt: The model is given a product description and the user's rating and review title, and must generate the review body. The prompt format is:

```
Write a review for the following product.
Product:
Product title: {product_title}
Product categories: {categories}
Review:
Your rating: {rating}
Your review was titled: {review_title}
Write the review body below:
```

The target is the full text of the user's review (truncated to 500 words). This format ensures the model must generate text conditioned on both the product context and the user's sentiment (given via the rating), while the soft prompts provide user-specific stylistic guidance.

StackExchange prompts: The model is given a different prompt depending on the specific sample being generated. To answer a question, the model is given:

```
Answer the following question:
{title}
{body}
Answer:
```

To write a question, the model is given:

```
Write the body of a question with the following title:
{title}
Question:
```

And finally, to write a comment, the model is given:

```
Write a comment on the following post:
{body}
Comment:
```

Amazon Reviews rating prediction prompt: The model is given a review title and review text, and must predict the numeric star rating (1-5). Including the review body and title gives the model background context for the user's overall sentiment, although the specific translation from text to numeric review must be learned for each user. The full prompt is:

```
Provide a rating for the following review:
{review_title}
{review_text}
Rating:
```

MovieLens rating prediction prompts: The model is given a movie title, a list of genres, and must predict the target rating.

```
Provide a rating for the following movie:
{title}
{genres}
Rating:
```

H LLM JUDGE DETAILS

The full prompt given to the LLM judge is given below:

```
You are an expert in authorship attribution. Below is a reference
text written by one person, and two candidate texts.

REFERENCE:
{reference}

OUTPUT A:
{response_a}

OUTPUT B:
{response_b}

Which output was more likely written by the same person who
wrote the reference?
Judge only similarity of voice: tone, register, formality,
typical length, and vocabulary.
Greater length, more detail, and more helpfulness are NOT better.
A longer or more thorough answer that does not sound like the
reference's author is the worse match.
Respond with ONLY one of:
- "A" if Output A is more likely the same author
- "B" if Output B is more likely the same author
- "TIE" if they are equally likely
```

In our experiments, we use a Qwen3.8-27B model as the judge, finding that it is the less susceptible to length bias and position bias than other Qwen3 or Gemma4 models.

To perform judging, we first strip all reasoning context from a generation. In some cases, such as with **POST**, this removes entire samples, meaning that certain win rates are computed over fewer samples. Where this happens, we have made notes in the results (Table 1). All results are computed twice, with positions reversed, to ensure that position bias is minimized. We find that position A is chosen between 42.7% and 53.7% of the time, and ties are chosen between 1.5% and 4.4% of the time. In cases where the judge flips based on position, we discard the sample from win-rate computation (i.e., a sample that is labeled as winning *and* losing is not counted).

We run an experiment with the best performing method (**GT Large**) by comparing it against the reference text itself, just to be sure that the judge is not biased by length, position, or other text qualities that are not "match the tone of the original." We find that the reference text is preferred 100% of the time.

Figures 2 and 4 show the complete pairwise win matrices for all methods on both datasets.

Finally, we compute the length statistics for all generated samples under each method for the Pythia-12B large model, shown in Table 11. This result shows that the baselines generally tend towards longer answers, while **Projected**, **GT Large**, and **GT Small** methods all correctly identify that humans tend to write much shorter reviews or responses.

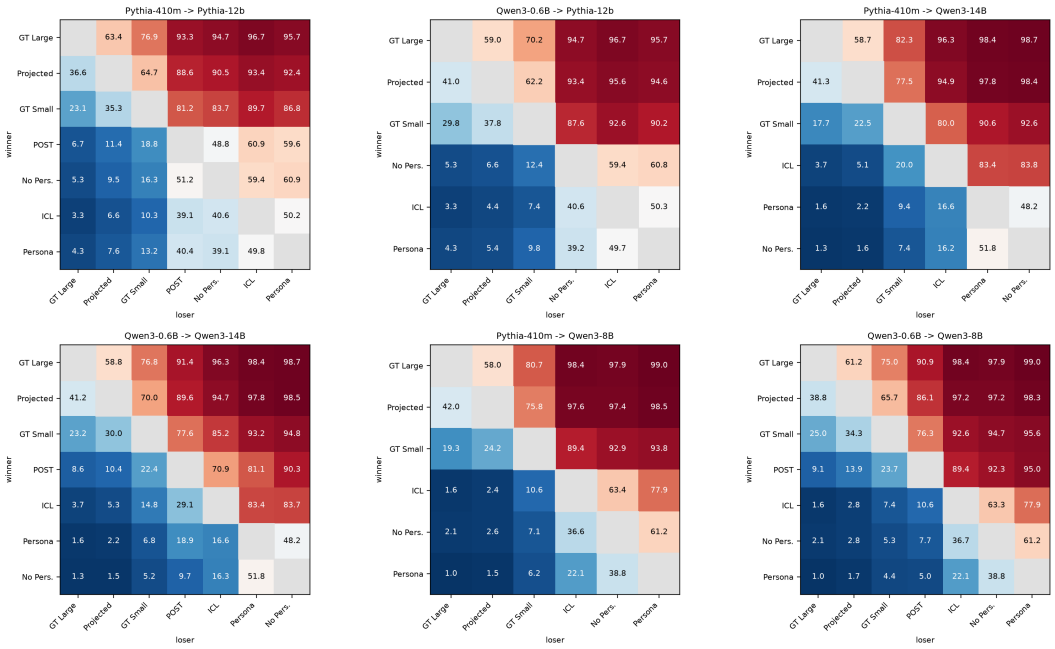


Figure 4: Pairwise win-rates on the StackExchange text generation task. POST only exists within family (i.e., Pythia to Pythia and Qwen to Qwen) due to limitations of the method. We again see that **GT Large** is always the top performer, with **Projected** in second and winning over all baselines.

Table 11: Mean and median generated length in words under the Pythia-12B model. † GT Small generates under the Pythia-410M model. We observe that baselines tend towards longer text compared to real human outputs.

	Amazon Reviews		StackExchange	
Method	**Mean**	**Median**	**Mean**	**Median**
Human reference	80.1	46	63.9	44
No Personalization	77.0	82	138.5	147
ICL	131.4	143	144.4	149
Inferred Persona	76.3	66	138.5	147
POST	167.2	170	154.2	158
GT Small†	62.0	44	55.3	42
Projected	39.4	32	53.9	39
GT Large	43.1	31	47.8	37

I Prompt Introspection Details

The full set of personal data decoding prompts is:

- "Describe this set of instructions in plain English: $\{\tau_s\}$",
- "What kind of user or persona do these instructions represent?$\{\tau_s\}$",
- "Summarize what makes this instruction set unique: $\{\tau_s\}$",
- "What patterns or preferences are encoded in these instructions?$\{\tau_s\}$",
- "Explain what these instructions would make a language model do differently:$\{\tau_s\}$",

We search for exact overlapping 10-word sequences between a decoded explanation and that user's own reviews, after lower-casing and stripping punctuation. Of 6,748 test users, 9 users produced at

least one such overlap, with 19 total overlaps being generated under the small model (Pythia-410M). No overlaps were generated by the large model using projected prompts. The full set of repetitions is:

- "i would recommend this book to everyone who wants to have a great laugh and" (5 times),
- "i would recommend this book to anyone who enjoys a good" (4 times),
- "i would recommend this book to everyone who wants to get a great" (3 times),
- "i cant wait for the next book in this series" (twice),
- "i cant find it on any video sites so i had" (twice),
- "this is one of my favorite books of all time",
- "the story is told from the point of view of",
- "i look forward to the next book in the series".

Overall, these are all generic sequences and they do not reveal any sensitive personal information, again suggesting that projected prompts are poorly suited to memorizing or transmitting personal information.

J Extended Generation Examples

We present qualitative examples from our Qwen3-0.6B $\rightarrow$ Qwen3-14B setting on the Amazon Reviews dataset, with all examples coming from *test* users ($\mathbf{U}_W$), whose data was never exposed to the large model. We show the user's ground truth reference review as well as our seven comparison conditions, including **No-Person** (no personalization), **Persona** (using an LLM-inferred persona), **ICL** (in context learning), **POST** (the POST baseline), **Projected** (our method), **GT Small** (generated under the user's ground truth small model soft prompts and generated on the small model), and **GT Large** (generated under the user's ground truth large model soft prompts).

The three prompting baseliens (**No Persona**, **Persona**, and **ICL**) were generated with reasoning disabled, as they would otherwise use their token budget on long and rambling thought traces. The fine-tuned methods (**POST**, **Projected**, **GT Small**, and **GT Large**) are not given this advantage.

As shown in Table 11, the average length of each generation is quite different under the various generation conditions, and the fine-tuned baselines are all closer to the brief text that real users are more likely to generate. Similarly, Tables 12, 13, & 14 show model-generated samples under various conditions, showing that soft prompt projection yields the most faithful generations as compared to the reference text.

J.1 Different Ratings with the Same User

Table 12 shows projected generations from the same user across five different star ratings. The user's overall voice and tone is largely preserved and consistent across these generations, though the sentiment shifts with the star rating in the prompt. We see that the generations are largely consistent across the different ratings, they tend to emphasize the character focus in the story, and they are often just one or two sentences.

J.2 Comparing Different Conditions

Tables 13 & 14 show all seven generation conditions for both a two star and a five-star item from the Amazon Reviews dataset. Both tables show that the **Projected** prompts lead to generations that most closely match the **Reference** sample. All baselines tend towards longer, generic sounding reviews, and they all heavily rely on the review title and product name. While **GT Small** *can* produce good text, it is also prone to degeneration and repetition, as shown in Table 13.

Table 12: Projected generations from a single test user across five star ratings. Sentiment varies with the rating while the user's concise voice and characteristic focus on craft are preserved.

Rating	**Projected (Ours)**
1	I was very disappointed in this book. The characters were not developed enough and I found the ending very confusing.
2	I like the idea of a new series but the editing on this was so bad that it distracted me.
3	The plot of the story is excellent. However, there are a lot of errors in the French that the author has used.
4	Nice short read. The characters are real, the setting is good. I would like to see a follow up to this story.
5	I was hooked from the first paragraph! A great mystery with a very interesting and unique heroine.

Table 13: All generation conditions for one user's two-star review of *The Job Offer* titled "Shows promise but needs help!" We see that **Projected** capture the lack of depth as a sticking point for the user, and closely matches the tone and length of the **Reference** sample. In constrast, **GT Small** degenerates into a repetitive pattern, **POST** terminates mid-sentence, and prompting baselines produce generic two-star reviews with an over-reliance on the title and product name.

Condition	**Generation (2-star review of *The Job Offer*)**
Reference	Shows promise in terms of story line but the primary characters need far more development. The story lacks depth, as do the primary characters. The author needs a good editor as well. I would rate this a 2 due to the pending promise.
Projected	The Job Offer is the first book in the Breakthrough series. It is a story of second chances, friendship, and love. This book shows promise but needs a little help. The characters are well written and the story is easy to read. However, the book lacks a little depth, and the story could be more engaging.
POST	**Title: Shows promise but needs help!** *The Job Offer* starts with an intriguing premise—a romantic tension simmering in a high-stakes professional
GT Small	This story is well written but needs help! The story has a good beginning and a good middle but is slow to build to a climax. The story is good but needs help. I would like to see more of the characters. It is hard to say what is wrong with this story. It is a good story and needs a more developed character and a more developed conflict. This story is good but needs to be improved. The story is good and needs to be improved.
GT Large	The book starts out with promise but really starts to fall apart. The story is not well developed and the characters are underdeveloped and flat. The book is a quick read but needs more work! I assigned the book a sexy rating of 3. I found the book a bit boring. I am not sure why I was attracted to the story. The book was not well edited.
Persona	*Title: Shows Promise but Needs Help! Rating: 2.0/5. The Job Offer* begins with a compelling premise, hinting at a romance fraught with tension and intrigue. The protagonist's internal conflict is palpable, and there are moments when the narrative threatens to blossom into something truly engaging. However, the execution falters under the weight of inconsistent pacing and underdeveloped secondary characters. The dialogue often feels stilted, lacking the natural rhythm that breathes life into romantic tension.
ICL	A sweet, light-hearted romance with a charming premise, but it falters in execution. The story has potential, especially in the early stages where the characters are introduced and the setting is laid out with a touch of warmth. However, the pacing feels off, and the dialogue often comes across as forced or overly cheesy. The main characters, while likeable, lack depth and their emotions don't always feel genuine.
No-Person	**Review Title: Shows promise but needs help! Rating: 2.0 out of 5 stars.** *The Job Offer* has the potential to be a compelling read, especially for fans of the romance genre. The premise is engaging and the initial setup of the characters shows promise. However, the story quickly loses momentum and fails to deliver on its early potential. The pacing is uneven, with some parts feeling rushed while others drag on without much development.

Table 14: All generation conditions for one user's five-star review of *12 Days of Forever* titled "More Beaumont!" While all methods generate favorable reviews, **Projected** very closely reflects the **Reference** in focusing on the characters as the draw of the book. In contrast, other methods produce generic reviews that do not appear to have any underlying connection to the reference author or the subject matter, but which heavily rely on the review title and rating.

Condition	**Generation (5-star review of *12 Days of Forever*)**
Reference	I loved getting more Beaumont! Heidi did well with this novella, and while it was a quick, light read I absolutely loved it! Xander and Yvie are perfect together. Xander is the personal trainer for 4225 West, and Yvie is Harrison's younger sister, and seeing how their relationship worked out was definitely worth the wait. And that cover is absolute perfection!
Projected	I have to admit that I was not sure about this story when I first started reading it. But after a few chapters, I was hooked! It was great to see the Beaumont family again. I love how they keep getting bigger and bigger. This story focuses on Lexi, the youngest sister, and her baby brother, Oliver. It was a cute story of how Lexi finds her true love while raising her little brother.
POST	**Review Title: More Beaumont! Rating: 5.0 out of 5 stars.** "12 Days of Forever" is another fantastic addition to the Beaumont Series, and I absolutely loved it! The story is beautifully written, with rich character development and emotional depth that keeps you hooked from the first page to the last. The author has a way of making you feel every moment, whether it's the heartwarming connections or the intense emotional struggles the characters face. I was thrilled to see the return of familiar characters and the introduction of new ones who added fresh energy to the story.
GT Small	I am so glad I bought this book. I can't wait to see what happens next in the Beaumont series. I have a lot of stories to tell about the Beaumont series, and I really enjoyed this one.
GT Large	The Beaumont family is one of my favorite series, and I'm happy to see it continues with this story. I've loved every single book so far, and 12 Days of Forever is no exception. I think this was the most emotional of the series, and I had tears when I read it. This is one of the most powerful and emotional stories I've ever read. I'm so glad that it's now available on audio.
Persona	I devoured *12 Days of Forever* in one sitting—couldn't put it down! This latest installment in *The Beaumont Series* is everything I hoped for and more. The chemistry between the leads is electric, and the emotional depth of their relationship had me swooning and wiping away tears. Author Sarah M. Eden has a way of making you feel every heartbeat, every hesitation, and every triumph. The secondary characters add layers of warmth and humor, making the story feel like a cozy reunion with old friends.
ICL	I've been a fan of the Beaumont series since the beginning, and *12 Days of Forever* is another stellar addition. The way the author weaves together the lives of the Beaumont siblings and their love stories is pure magic. This book focuses on the youngest sibling, Claire, and her journey of self-discovery and love with the charming and brooding Logan. Their connection is electric, and the slow burn is handled with such care and emotion. The family dynamics continue to be a highlight.
No-Person	**More Beaumont!** "12 Days of Forever" is another stellar addition to The Beaumont Series, and it only solidifies why this series has become a favorite of mine. The story is beautifully written, with rich character development and emotional depth that keeps you fully engaged from start to finish. The author has a remarkable way of weaving romance, drama, and personal growth into a seamless narrative. The chemistry between the main characters is palpable, and their journey is both heartwarming and compelling.