Embedding-to-Prefix: Continual Personalization with Large Language Models

Bernd Huber, Ghazal Fazelnia, Andreas Damianou,
Sebastian Peleato, Max Lefarov, Praveen Ravichandran,
Marco De Nadai, Mounia Lalmas-Roellke, Paul Bennett
Spotify
{bhb, ghazalf, andreasd, speleato, mlefarov,
praveenr, mdenadai, mounial, pbennett}@spotify.com

Abstract

Large language models (LLMs) excel at generating contextually relevant content, but their static nature prevents adaptation to dynamic, evolving user preferences. Yet capturing the full spectrum of user preferences that evolve over time remains challenging. Existing methods for capturing evolving user preferences and taste profiles often depend on fine-tuning or token-intensive prompting, which typically require significant effort or computational expense. We propose Embedding-to-Prefix (E2P), a parameter-efficient adaptation that injects pre-computed user embeddings into an LLM through a learned projection to a single soft token prefix. This enables effective personalization while keeping the backbone model frozen, providing a compatible and cost-effective mechanism for continual model updates. We evaluate E2P in two large-scale production settings where user embeddings are dynamically updated: music playlist generation and podcast recommendation. Our results show that E2P achieves a 12.9% improvement in user engagement for music recommendation and provides complementary personalization signal in podcast recommendation.

1 Introduction

Foundation models face a critical challenge: they become outdated, as they are trained on static snapshots of data [11, 13]. While these models excel at generating contextually relevant content, their static nature prevents them from adapting to dynamic information such as evolving user preferences. In large-scale recommender systems—our focus here—this dynamic user context is captured in rich, dense user embeddings (outside the LLM's embedding space) that are continually updated [4, 2]. The challenge is to bridge the gap between these dynamic user representations and a large language model (LLM) that is costly to update due to its size relative to the embedding model. We propose Embedding-to-Prefix (E2P), a parameter-efficient method that maps these continuously updated user embeddings directly into the LLM's hidden space. By learning a simple projection to a single soft prefix token, E2P enables dynamic, user-specific conditioning at inference time without altering the base model's weights. This provides a cost-effective and compatible solution for keeping foundation models updated with users' evolving preferences. We validate E2P on two large-scale production recommendation tasks where user embeddings evolve continuously, demonstrating significant improvements in user engagement (+12.9%) while maintaining minimal computational overhead.

Our main contributions are:

- **A Compatible Update Mechanism:** We propose a novel framework for injecting dynamic, pre-existing user embeddings into a frozen LLM via a learned mapping to a single soft prefix, ensuring model compatibility.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

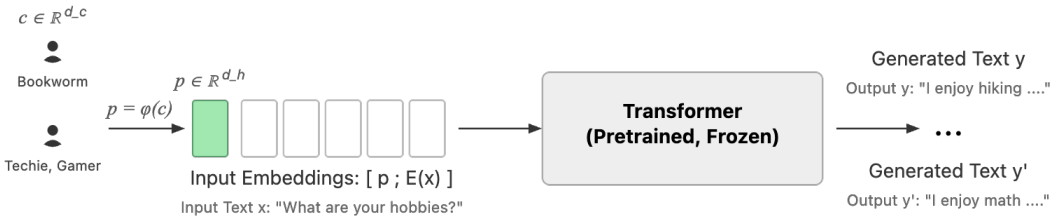

Figure 1: Overview of Embedding-to-Prefix (E2P). A projection module maps each user's embedding c_i to a user-specific soft prefix token p_i that is prepended to the LLM input embedding sequence.

- **Cost-Effective Continual Adaptation:** E2P adapts LLM behavior at inference time by training only a tiny, shared projection module, offering a scalable alternative to costly continual retraining.
- **Production-Scale Validation:** We demonstrate E2P's effectiveness in two large-scale production case studies (with proprietary data reflecting real deployment constraints) where user embeddings are continually evolving, showing significant gains over production baselines.

2 Related Work

Our work is closest to prefix/prompt tuning, which steer frozen models by prepending learned continuous vectors [7, 6]. These methods typically learn task-level prefixes; E2P instead learns a shared projection that maps *pre-existing, continuously updated* user embeddings into a single soft prefix for instance-level personalization. While related methods have explored conditioning on user data [12, 5], they often consume context tokens or lack fine-grained personalization. Other vector-to-text bridges have trained dedicated user encoders from scratch (e.g., USER-LLM [9]) or learned user-specific adapters (e.g., Persona-Plug [8]). In contrast, E2P's use of a single, shared projection module for existing embeddings makes it a scalable, drop-in solution for production systems with millions of users, avoiding per-user parameters or expensive encoder training. While adapter-based methods exist, E2P's single shared projection avoids per-user storage overhead critical at scale.

Positioning. E2P follows soft-prompt/prefix tuning [7, 6] but targets frozen LLMs with *pre-existing, continually updated* recommender-system embeddings via a *single shared* projection (no per-user parameters), unlike per-user adapters or dedicated user encoders [8, 9].

3 Method

We introduce Embedding-to-Prefix (E2P), a parameter-efficient approach that maps user embeddings to a single soft prefix token for personalizing frozen pre-trained language models. Here, we use *user embeddings* to denote dense, continuously updated vectors that summarize an individual's historical preferences and behaviors (e.g., clicks, listens), typically learned by large-scale recommender systems and maintained external to the LLM; E2P maps these vectors into the model's hidden space [4, 2]. Unlike methods that require extensive fine-tuning or lengthy textual descriptions, E2P directly projects user representations into the model's hidden space through a lightweight projection module.

Projection Module. The core of E2P is a projection module ϕ that maps user embeddings to the LLM's hidden space. Given a user embedding $c \in \mathbb{R}^d$, we define:

$$\phi(c) = \text{LayerNorm}(\text{ReLU}(W_1 c))W_2 + b \tag{1}$$

where $W_1 \in \mathbb{R}^{d_c \times d}$, $W_2 \in \mathbb{R}^{d_c \times d_h}$, and $b \in \mathbb{R}^{d_h}$. This two-layer MLP design balances expressiveness with parameter efficiency.

Soft Token Insertion. Given the projected vector $p = \phi(c) \in \mathbb{R}^{d_h}$, we insert it as a soft token at the beginning of the input sequence. Unlike standard input tokens that map to discrete words in a vocabulary, this "soft" token is a continuous vector that is learned directly. For an input sequence with embeddings $E(x) \in \mathbb{R}^{T \times d_h}$, the modified input becomes:

$$\hat{E}(x) = [p; E(x)] \in \mathbb{R}^{(T+1) \times d_h} \tag{2}$$

This single-token design maintains efficiency while enabling effective personalization. The training objective depends on the task: standard language modeling loss for text generation and Kahneman-Tversky Optimization for recommendation tasks with binary engagement labels.

Training Objectives. The training objective depends on the task. For text generation tasks, we adopt the standard language modeling objective. Given a user embedding c, its corresponding soft prefix $p = \phi(c)$, and a target sequence $y = (y_1, \ldots, y_T)$, we maximize the log-likelihood:

$$\mathcal{L}_{LM}(c) = \sum_{t=1}^{T} \log p_\theta(y_t | y_{<t}, [p; x]) \tag{3}$$

where p_θ denotes the output distribution from the frozen language model with parameters θ.

For recommendation-oriented tasks with binary engagement labels, we use Kahneman-Tversky Optimization (KTO) [1], which accounts for both positive and negative outcomes:

$$\mathcal{L}_{KTO}(c) = \sum_{i=1}^{N} y_i \log p_\theta(y_i | [p; x_i]) + \alpha(1 - y_i) \log(1 - p_\theta(y_i | [p; x_i])) \tag{4}$$

where α balances false positives and negatives.

Key Design Choices. E2P's single-token design distinguishes it from multi-token approaches. This choice: (1) minimizes inference latency which is critical for production systems, (2) maintains compatibility with frozen models without modifying attention patterns, and (3) aligns with findings that single-token interventions can be surprisingly effective [6]. The shared projection function enables generalization across users with similar embeddings, preserving the structure of the original embedding space (see Appendix for visualization). We keep the LLM *frozen* to preserve modularity of the personalization module; the same projection can be paired with fine-tuned backbones at the cost of this modularity (not explored here).

4 Case Study: Continual Personalization in Real-World Systems

To evaluate E2P's ability to provide continual and compatible updates, we focus on two large-scale production case studies where user context, captured in behavioral embeddings, is constantly evolving. In these settings, the primary goal is to keep the LLM's behavior aligned with the most recent user preferences without retraining the base model.

4.1 Tasks and Datasets

Music Playlist Generation. This task involves generating personalized music playlists in response to user queries (e.g., "upbeat songs for coding"). The challenge lies in understanding both the query's semantic intent and the user's specific musical preferences. User context is provided by a proprietary 120-dimensional behavioral embedding that captures listening patterns, genre preferences, and artist affinities. These embeddings are dynamically updated at regular intervals to reflect recent listening behavior, making them a prime example of the continual adaptation challenge. The dataset consists of 300,000 user queries with engagement labels, split into train/test sets.

Podcast Recommendation. This task focuses on recommending the next podcast episode for a user based on their listening history. Unlike music, podcast consumption often follows sequential patterns and topical interests. Personalization uses a 120-dimensional behavioral embedding capturing factors like preferred topics, episode length, and listening completion rates. The recommendation is modeled as an auto-regressive generation task over a vocabulary of quantized content IDs [10]. The dataset contains 2M listening sequences with a 10,000 test set.

Sampling protocol. For users with multiple instances, each example is paired with the contemporaneous embedding available at that time, encouraging robustness to preference drift.

4.2 Experimental Setup

We use LLaMA-3.2-3B [3] as the base model for all experiments. The base model parameters remain frozen; only the E2P projection module (approximately 100,000 parameters) is trained using the

AdamW optimizer. We compare E2P against the following baselines, chosen to represent practical alternatives in large-scale systems:

No Context Vanilla LLM with no personalization (all improvements measured relative to this).

Prompt Context Textual user description prepended to input (41 tokens for Podcast Rec).

E2P-Random Control using random user embeddings to isolate architectural effects.

E2P + Prompt Combines E2P prefix with textual prompt.

Embedding Retrieval Standard non-generative baseline using cosine similarity.

Note. Prompt-Context relies on manual prompt engineering; E2P removes this dependence by leveraging ubiquitous behavioral embeddings and is complementary to prompts (see E2P+Prompt).

Table 1: Performance of E2P in large-scale production tasks. E2P enables continual personalization by injecting dynamically updated user embeddings. Improvements are relative (%) to the No Context baseline.

Method	**Music Rec** Engagement	**Podcast Rec** HitRate@30
Embedding Retrieval	+0.1%	+1.2%
E2P-Random	+5.0%	+0.7%
E2P	**+12.9%** *	+2.2% *
Prompt-Context	-	+10.3%
E2P + Prompt	-	**+13.7%** *

* Statistically significant improvements ($p < 0.05$, paired t-test).

4.3 Results and Analysis

For production-scale music recommendation, where user embeddings are continually updated to reflect new interactions, E2P delivers a **12.9% improvement** in predicted user engagement. This substantial gain over the non-generative reranking baseline (+0.1%) and the E2P-Random control (+5.0%) confirms that performance benefits stem from injecting meaningful, up-to-date user signals rather than architectural effects.

In podcast recommendation, E2P provides a modest but consistent lift (+2.2%) with just a single token addition. While the token-heavy Prompt-Context baseline achieves higher standalone performance (+10.3%), it requires 41 additional tokens per request. Crucially, combining E2P with the prompt yields the best result (**+13.7%**), demonstrating that E2P's dense, dynamic embedding captures complementary personalization signals not easily expressed in static text, validating its value even in prompt-rich scenarios.

These results validate E2P as a cost-effective mechanism for injecting real-time user context into frozen LLMs, directly addressing the workshop's theme of keeping foundation models up-to-date.

5 Conclusion

We introduced Embedding-to-Prefix (E2P), a method designed to address a core challenge for foundation models: keeping them up-to-date with dynamic, real-world information. E2P provides a **compatible and cost-effective update mechanism** by injecting continuously evolving user embeddings into a **frozen** LLM via a single soft prefix. This allows the model's behavior to adapt at inference time without expensive retraining or fine-tuning. Our evaluation on two large-scale, real-world recommendation case studies confirms E2P's effectiveness, achieving a 12.9% improvement in user engagement for music recommendation. E2P offers a scalable, production-ready pathway for enabling continual, user-aware updates in large generative models.

References

[1] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

[2] Ghazal Fazelnia, Sanket Gupta, Claire Keum, Mark Koh, Ian Anderson, and Mounia Lalmas. Generalized user representations for transfer learning. *arXiv preprint arXiv:2403.00584*, 2024.

[3] 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.

[4] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182, 2017.

[5] Sachin Kumar, Chan Young Park, Yulia Tsvetkov, Noah A Smith, and Hannaneh Hajishirzi. Compo: Community preferences for language model personalization. *arXiv preprint arXiv:2410.16027*, 2024.

[6] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *proceedings of the 2021 conference on empirical methods in natural language processing (emnlp 2021)*, pages 3045–3059, 2021.

[7] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

[8] Xiaodong Liu, Liang Yu, Xiao Sun, Hao Fu, Kaili Zhang, Jingjing Wang, Mingyuan Fan, Shuyang Tai, Zhiyong Li, and Jian Gao. Llms + persona-plug = personalized llms. In *proceedings of the 2024 advances in neural information processing systems (neurips 2024)*, 2024.

[9] 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. *arXiv preprint arXiv:2402.13598*, 2024.

[10] Anima Singh, Trung Vu, Nikhil Mehta, Raghunandan Keshavan, Maheswaran Sathiamoorthy, Yilin Zheng, Lichan Hong, Lukasz Heldt, Li Wei, Devansh Tandon, et al. Better generalization with semantic ids: A case study in ranking for recommendations. In *Proceedings of the 18th ACM Conference on Recommender Systems*, pages 1039–1044, 2024.

[11] Yujiang Wu, Hongjian Song, Jiawen Zhang, Xumeng Wen, Shun Zheng, and Jiang Bian. Large language model as a universal clinical multi-task decoder. *arxiv preprint arxiv:2406.12738*, 2024.

[12] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. Personalizing dialogue agents: I have a dog, do you have pets too? In *proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 2204–2213, 2018.

[13] Ying Zhou, Xinyao Wang, Yulei Niu, Yaojie Shen, Lexin Tang, Fan Chen, Ben He, Le Sun, and Longyin Wen. Difflm: Controllable synthetic data generation via diffusion language models. *arxiv preprint arxiv:2411.03250*, 2024.

A Music Recommendation: Background

The large-scale music recommendation task involves generating playlists for user queries that often require understanding concepts beyond simple genre or artist matching, frequently termed "world knowledge" queries (e.g., "upbeat songs for coding," "music like artist X but less known"). While a base LLM can interpret the query's theme, the generated tracklist might lack resonance if it ignores the user's specific affinities within that theme (e.g., preference for electronic vs. instrumental coding music, or familiarity level with suggested artists). The core challenge is thus to steer the LLM's generation not just by the query's explicit request, but also by the user's implicit tastes captured in their behavioral embedding, aiming for playlists that feel personally curated.

Engagement Predictor Details: To evaluate the effectiveness of personalization in this offline setting, we utilized a dedicated engagement prediction model. This binary classifier was trained on a large historical dataset of user interactions with previously recommended playlists, learning to predict the likelihood of a positive engagement event (specifically, the user saving the generated playlist) given the user's embedding, the original query text, and representations of the tracks in the generated playlist. This predictor serves as a proxy metric for user satisfaction, allowing us to estimate the impact of different personalization strategies like E2P on downstream user behavior before deploying them online.

B Podcast Recommendation: Background

The podcast recommendation scenario focuses on predicting the subsequent podcast a user might listen to, often leveraging the immediate context of recently played podcasts. Unlike playlist generation, this task typically emphasizes sequential prediction and ranking a single best item. The vastness of podcast content—spanning diverse topics, formats, hosts, and production styles—makes personalization particularly critical. A user's choice for the "next" podcast can be influenced by factors like narrative continuity within a series, topical relevance to recent listens, preference for specific hosts or guests, or alignment with broader, long-term interests not immediately obvious from the last few plays. E2P's application here tests its ability to inject nuanced, embedding-derived user preferences into this sequential prediction task, complementing context derived from recent listening history or explicit textual user profiles, aiming to improve the hit rate of relevant suggestions within the user's feed.

C Music Recommendation: Embedding Structure

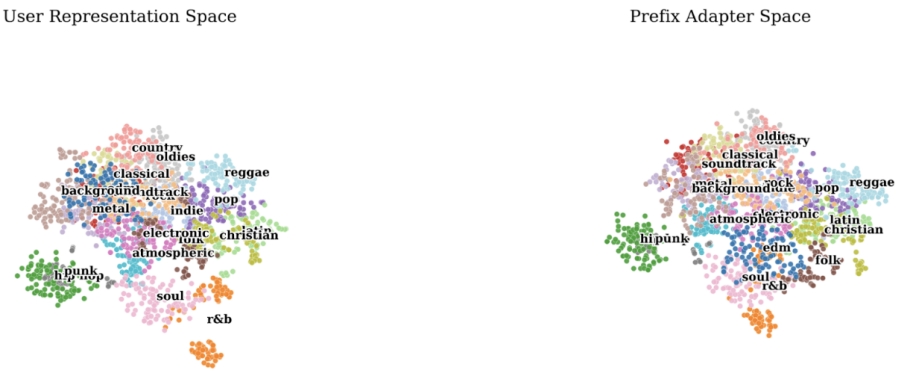

Figure 2: Visualization of user embedding properties in the LLM's hidden representation space: (a) t-SNE visualization of user embeddings colored by music genre preference clusters, (b) Corresponding visualization of the same users in prefix adapter space, demonstrating preservation of preference clusters with enhanced separation between distinct user groups. This plot demonstrates that the learned projection preserves meaningful user preference structures (e.g., genre clusters) when mapping from the user embedding space to the prefix adapter space, supporting the potential for generalization based on user similarity.

Figure 2 demonstrates how E2P effectively preserves user preference signals in the prefix adapter space for our music recommendation task, enabling the model to generate recommendations that respect nuanced user preferences while leveraging the world knowledge capabilities of the LLM.

D Hyper-parameters

Task	Base model	lr	batch	epochs	seed
Music Rec	LLaMA-3.1-8B	5×10^{-7}	256	5	42
Podcast Rec	LLaMA-3.2-1B	1×10^{-5}	16	5	42

E Statistical tests

We performed statistical significance testing to validate the improvements reported in Table 1. We used paired two-sided t-tests, comparing the performance metrics on the same test set instances across different methods, with a significance level of $\alpha = 0.05$. Paired tests were used because model performance was evaluated on the identical test set instances for each comparison, allowing us to control for instance-specific variations.

F Datasets and Splits

We study two proprietary datasets:

1. **Music Playlist Generation.** This proprietary dataset consists of 300,000 user queries and generated playlists. Each row is labeled with an engagement annotation by the listener (50%/50%). 1,000 queries were randomly sampled chosen as test set. Every user row is joined with the latest 120-d behavioral embedding already computed in production.

2. **Next Podcast Recommendation.** This proprietary dataset consists of 2,000,000 pairs of previously listened podcasts, metadata, and user context, and a testset of 10,000 samples. Labels are single semantic IDs; evaluation uses hit-rate@30.

F.1 Evaluation Metrics

Additional information on evaluation metrics by task:

Music Rec: We evaluate this task using a binary engagement classifier trained on historical user interaction data. The classifier takes as input a user prompt and a set of track embeddings and predicts the binary signal whether or not a user engages (Improvements in this offline proxy have correlated with online gains in user satisfaction in related production experiments).

Podcast Rec: We evaluate this task using temperature sampling with $t = 1.0$, computing HitRate@30. For each query, we sample 30 candidates and measure whether the true positive appears in the top 30 ranked results.