Factorized Soft Prompts with Inference-Time Latent Control for LLM Personalization

Likun Yang[1,2] **Pei Xu**[2,*] **Shiyue Cao**[1,2] **Tong Wang**[1,2]
Haoran Wu[1,2] **Xiaotang Chen**[1,2] **Kaiqi Huang**[1,2,*]
[1]School of Artificial Intelligence, University of Chinese Academy of Sciences
[2]National Key Laboratory of Cognition and Decision Intelligence for Complex Systems, Institute of Automation, Chinese Academy of Sciences
yanglikun2021@ia.ac.cn pei.xu@ia.ac.cn kaiqi.huang@nlpr.ia.ac.cn

Abstract

Personalizing large language models (LLMs) requires reflecting stable user preferences while adapting their expression to each target context. Soft-prompt personalization has become a practical route to this goal, but existing methods produce soft prompts in which stable preferences, target-induced content, and context-dependent expression are bundled into an entangled conditioning signal without separately addressable components, leaving inference with no principled space to intervene on individual aspects of personalization. We propose **Fa**ctorized soft prompts with **L**atent **Con**trol (**FaLCon**), an LLM personalization framework built on two coupled stages. First, *factorized soft prompts* replace this entangled conditioning with a set of separately addressable preference factors, forming a structured control interface. Second, *inference-time latent control* treats the resulting factor weights as a continuous action space and adapts them per instance from teacher-forced likelihood feedback, keeping the LLM frozen and updating only a low-dimensional factor-weight posterior without back-propagating through the LLM. Experiments on multiple personalized generation tasks reveal that FaLCon consistently outperforms strong baselines, validating the individual necessity and joint effectiveness of the structured interface and the inference-time controller.

1 Introduction

Large language models (LLMs) are increasingly used in user-facing applications, such as recommendation and conversational assistance (Peng et al., 2024; Liu et al., 2025a), where users expect responses aligned with their preferences rather than generic outputs. This has motivated LLM personalization from historical interactions, but a key challenge remains: user preferences are multifactored—covering style, sentiment, verbosity, and more—and their importance varies with context.

Soft-prompt personalization offers an attractive balance between effectiveness and flexibility, distilling user history into compact continuous prompts that can be stored, reused, and injected into a frozen LLM as latent conditioning signals (Li and Liang, 2021; Lester et al., 2021; Liu et al., 2022). It has accordingly become a practical route in industrial deployment (Huber et al., 2025) and recent LLM personalization methods (Hebert et al., 2024; Liu et al., 2025b; Qiu et al., 2025a).

Recent soft-prompt methods instantiate personalization through history-induced prompts (Hebert et al., 2024; Huang et al., 2024; Xiao and Hou, 2026), holistic user embedding injection (Doddapaneni et al., 2024; Ning et al., 2025; Liu et al., 2025b; Huber et al., 2025), or inter-user residual prompting (Qiu et al., 2025a). Despite different parameterizations, they share the same modeling assumption: personalization is collapsed into a structurally undifferentiated latent signal distilled from user history—be it a single vector or a sequence of prompt tokens, no position is committed to a specific aspect of personalization. This signal is optimized for predictive fit, but not for factor-level attribution, so stable preferences, target-induced content, and context-dependent expression are entangled inside the same prompt. As a result, inference lacks a principled control space: even dynamic prompt selection still switches among entangled signals rather than intervening on separable factors. More broadly, representation learning has repeatedly shown that separating generative factors requires explicit inductive structure rather than emerging automatically from predictive objectives (Bengio et al., 2013; Locatello et al., 2019; Schölkopf et al., 2021).

Motivated by this diagnosis, we propose **Fa**ctorized soft prompts with **L**atent **Con**trol (**FaLCon**), an LLM personalization framework

*Corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 42093–42116
October 24-29, 2026 ©2026 Association for Computational Linguistics

that turns latent personalization from an undifferentiated signal into factor-level latent control. The key idea is that a prompt can support controllable personalization only if its internal coordinates are made meaningful before inference: stable user preferences should not be absorbed into target-induced content, and different preference factors should not compete inside an undifferentiated vector (Zheng et al., 2021; Ma et al., 2019). FaLCon therefore decouples the user-side signal from the content directly induced by the generation target, and then factorizes the resulting representations through multi-branch autoencoders regularized by redundancy reduction and anti-collapse objectives (Zbontar et al., 2021; Bardes et al., 2021). The branches act as separately addressable preference factors and form a structured control interface on which personalization can be analyzed and controlled at the factor level.

This control interface also makes inference-time adaptation well defined. For each instance, personalization is no longer a binary decision about which prompt to inject, but a continuous decision about how strongly each preference factor should shape the unfolding generation. We formulate this as inference-time latent control: a Thompson-sampling controller proposes factor weights, evaluates their effect with teacher-forced next-token likelihood, and updates a Bayesian posterior over factor utilities (Agrawal and Goyal, 2013). The final weights are applied through Key/Value attention scaling, changing how strongly the frozen LLM attends to each latent factor; the entire adaptation loop keeps the LLM frozen, updates only a low-dimensional factor-weight vector through gradient-free posterior inference. Experiments across multiple personalized generation benchmarks show that FaLCon achieves state-of-the-art performance.

Our contributions are threefold:

- We propose a soft-prompt personalization framework built as decoupled, separately addressable factors and adapted to each instance via online adaptation, rather than as an entangled latent signal fixed after training.
- We propose **FaLCon**, an algorithm that factorizes user and target into separately addressable soft-prompt factors and adapts their weights online via gradient-free latent control over a frozen LLM.
- Experiments on multiple personalized generation benchmarks show that FaLCon achieves state-of-the-art performance.

2 Method

FaLCon is designed around a simple premise: latent personalization becomes controllable only when the latent prompt is organized as a set of addressable factors. We separate the method into two coupled stages. First, we construct a factor-level soft-prompt interface by removing target-aligned leakage from the user signal and decomposing user and target representations into redundancy-reduced branches. Second, we use this interface as the action space for inference-time latent control, where factor weights are adapted during online adaptation and then fixed for decoding. Figure 1 provides an overview.

2.1 Preliminaries

Problem formulation. We study LLM personalization, where the goal is to generate output aligned with user u for the current target i. Each user has a history $\mathcal{H}_u = \{(i_j, y_j)\}_{j=1}^{m}$, where y_j is the text produced for a past target i_j. Given context $x_{u,i}$, the model defines

$$p_\Theta(y \mid x_{u,i}, \mathcal{H}_u), \tag{1}$$

with the backbone LLM frozen and personalization injected through continuous prompt embeddings. We instantiate this setting as personalized review generation, where histories encode latent factors such as style, sentiment, and emphasis whose relevance varies with i. An external encoder $E(\cdot)$ maps texts and targets to embeddings: $r_{u,j} = E(y_j)$, $q_{i_j} = E(i_j)$, and $q_i = E(i)$.

Offline factor interface. Prior soft-prompt personalization does not separate offline representation learning from online control. Offline, from $\mathcal{H}_u$ alone, we learn a prompting representation that exposes user preferences and target effects as distinct, separately addressable components, so that personalization is no longer a single entangled latent signal z_u (Qiu et al., 2025a) that bundles stable preferences, target-induced content, and context-dependent expression into prompt positions with no committed semantics. We meet this objective with a factor-level soft-prompt interface, in which $\mathcal{H}_u$ together with the current target i is mapped to two factor sets

$$Z_u = \{z_1^u, \ldots, z_K^u\}, \qquad Z_i = \{z_1^i, \ldots, z_K^i\}, \tag{2}$$

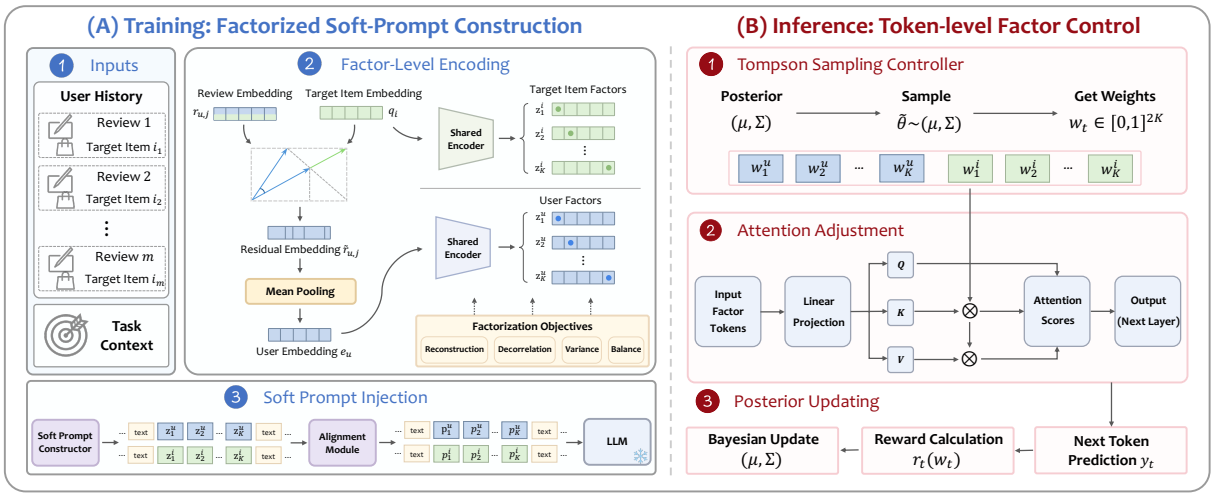


Figure 1: Overview of FaLCon. Left: target-decoupled user signals are obtained by projecting historical text embeddings away from their associated target directions. User and target signals are transformed into a factor-level soft-prompt interface through redundancy-reduced multi-branch encoding. Right: inference-time latent control estimates continuous factor utilities and applies the resulting weights through K/V attention scaling.

each factor injected at its own prompt slot; this stage uses an offline training split, follows the standard offline-train / single-shot-test workflow, and is fully comparable to existing soft-prompt baselines.

Online adaptation protocol. At deployment, we consider *online supervised adaptation from newly observed user interactions*. That is, after a user completes a new interaction, its target and realized text become available as supervised feedback for updating the user's factor-weight posterior; the task is not to adapt from unlabeled test inputs or from the held-out evaluation target. Concretely, for each test user we partition the held-out interactions into an *online-adaptation set* of $T_{\max}$ chronologically ordered records $\{(i_T, y_{u,i_T})\}_{T=1}^{T_{\max}}$ and a single *test record* $(i^\star, y_u^\star)$ corresponding to the user's latest interaction. The adaptation records simulate newly observed completed interactions and must be consumed sequentially: at round $T = 1, \ldots, T_{\max}$ (distinct from the intra-sequence token index t used later for rewards) the controller observes only (i_T, y_{u,i_T}) and updates its posterior state from $\mathcal{S}_T$ to $\mathcal{S}_{T+1}$ *without touching the LLM or the trained encoders*. Evaluation is performed only on the fixed test record $(i^\star, y_u^\star)$ using the final state $\mathcal{S}_{T_{\max}+1}$. We meet this objective with the inference-time latent controller of §2.3.

2.2 Factor-Level Soft-Prompt Interface

The first stage turns heterogeneous user–target observations into a set of separately addressable soft-prompt factors. The interface has two requirements: user-side factors should not be dominated by target semantics, and different factors should not collapse into redundant copies of the same latent summary. We address them with target-decoupled user signals and redundancy-reduced factor encoding.

Target-decoupled user signal. A historical text y_j embedded as $r_{u,j}$ inevitably carries target-specific semantics from i_j; directly pooling such embeddings risks treating target content as user preference. We therefore extract the component of each historical embedding that is orthogonal to its associated target. With $\bar{r}_{u,j} = r_{u,j}/\|r_{u,j}\|_2$ and $\bar{q}_{i_j} = q_{i_j}/\|q_{i_j}\|_2$,

$$\tilde{r}_{u,j} = \mathrm{norm}\Big(\bar{r}_{u,j} - (\bar{r}_{u,j}^\top \bar{q}_{i_j})\bar{q}_{i_j}\Big), \qquad (3)$$

where $\mathrm{norm}(\cdot)$ denotes ℓ_2 normalization. We aggregate the residual embeddings into a user representation $e_u = \mathrm{pool}(\{\tilde{r}_{u,j}\}_{j=1}^m)$ via mean pooling. This step removes the most direct target-aligned component before pooling, so that the subsequent factorization is driven by user-specific variation rather than repeated target semantics.

Redundancy-reduced factor encoding. Given a user signal e_u and a target signal q_i, FaLCon learns separate multi-branch encoders for each side. For $s \in \{u, i\}$, a shared encoder produces $h_s = \sigma(W_s a_s + b_s)$ from input embedding a_s, and K branch heads then yield factor embeddings $z_k^s = W_k^s h_s + b_k^s$, $k = 1, \ldots, K$. Each branch

has its own decoder, and the reconstruction is $\hat{a}_s = D_s([D_1^s(z_1^s); \ldots; D_K^s(z_K^s)])$.

The factor encoders are trained with a composite objective that preserves information while discouraging redundant or collapsed factors. The full factorization loss combines a reconstruction term $\mathcal{L}_{\text{rec}} = \ell(\text{norm}(a_s), \text{norm}(\hat{a}_s))$, a decorrelation term that jointly penalizes cross-branch batch covariance and head-weight alignment, $\mathcal{L}_{\text{dec}} = \sum_{k<l} \left(\|C_{kl}\|_F^2 + \|W_k^s (W_l^s)^\top\|_F^2 \right)$ with $C_{kl} = \frac{1}{B-1}(Z_k - \mu_k)^\top (Z_l - \mu_l)$, a variance floor $\mathcal{L}_{\text{var}} = \sum_k \frac{1}{d} \sum_c \max(0, \gamma - \text{Std}(Z_{k,c}))$ that prevents degenerate factors, and a balance penalty $\mathcal{L}_{\text{bal}}$ that discourages all information from concentrating in a few factors:

$$\mathcal{L}_{\text{fac}} = \lambda_r \mathcal{L}_{\text{rec}} + \lambda_d \mathcal{L}_{\text{dec}} + \lambda_v \mathcal{L}_{\text{var}} + \lambda_b \mathcal{L}_{\text{bal}}. \tag{4}$$

The objective does not assume that factors are named or supervised; it imposes the inductive structure needed for prompt dimensions to become separately addressable.

Soft-prompt injection and training. Each factor is mapped into the LLM hidden space through lightweight alignment modules, $p_k^u = A_u(z_k^u)$ and $p_k^i = A_i(z_k^i)$, and a soft-prompt constructor interleaves these continuous embeddings with the textual context at predefined factor slots, yielding the input

$$X_{u,i} = \text{Constructor}(x_{u,i}; \{p_k^u\}_{k=1}^K, \{p_k^i\}_{k=1}^K). \tag{5}$$

The model is trained with the causal language-modeling objective $\mathcal{L}_{\text{LM}} = -\sum_t \log p_\Theta(y_t \mid X_{u,i}, y_{<t})$ combined with Eq. 4; only the factor encoders and alignment modules are optimized. Because each branch occupies its own prompt position, inference can intervene on the contribution of a specific factor without changing the remaining context.

2.3 Inference-Time Latent Control

The factor-level interface changes what adaptation means at inference time. We instantiate this as the online-adaptation protocol of §2.1: for each test user the controller sweeps the online-adaptation set in chronological order, consuming one record (i_T, y_{u,i_T}) per round and updating its posterior state from $\mathcal{S}_T$ to $\mathcal{S}_{T+1}$, without touching any LLM parameters. After the last round, the resulting state $\mathcal{S}_{T_{\max}+1}$ is used to generate the prediction $\hat{y}_u^\star$ for the held-out test record $(i^\star, y_u^\star)$, which is the only record evaluated. The method is black-box and zero-order: it observes only teacher-forced log-likelihoods of the adaptation references under different factor weights, and uses the resulting evidence to drive a per-user Bayesian posterior over factor utilities.

Action space and factor-level K/V scaling. During teacher-forced adaptation, each controller action is a candidate continuous weight vector

$$\begin{aligned} w &= [w_1^u, \ldots, w_K^u, w_1^i, \ldots, w_K^i], \\ w &\in [w_{\min}, w_{\max}]^{2K}, \end{aligned} \tag{6}$$

over the K user and K target factors. We use K=8 and $[w_{\min}, w_{\max}] = [0, 1]$, so that w_k=1 recovers the trained prompt and smaller values *suppress* a factor without amplifying it beyond the regime seen during training. Let π_k denote the prompt position of factor token k and $w_0 = \mathbf{1}$ the unscaled baseline. We apply a chosen w at the attention level by scaling the Key and Value projections of the factor positions in the first transformer block,

$$K_{\pi_k}^{(0)} \leftarrow w_k K_{\pi_k}^{(0)}, \quad V_{\pi_k}^{(0)} \leftarrow w_k V_{\pi_k}^{(0)}, \tag{7}$$

while all other tokens, layers, and weights remain unchanged. The modulation thus controls *how much* downstream tokens can retrieve from each factor. Once a chosen vector is fixed for a forward or decode pass, the scaled layer-0 K/V entries are cached and reused across subsequent decoding steps; final test generation uses the fixed exploitation vector $\hat{w}$ from Eq. 13, not token-varying controller actions.

Teacher-forced reward. At round T of the online-adaptation sweep, the adaptation reference y_{u,i_T} provides the ground-truth tokens used to score candidate weights. For each reference token y_t in this round (where t is the intra-sequence token index), we define the reward of a candidate weight w as the teacher-forced log-likelihood improvement over the unscaled baseline,

$$\begin{aligned} r_t(w) = &\log p_\Theta(y_t \mid X_{u,i}, y_{<t}; w) \\ &- \log p_\Theta(y_t \mid X_{u,i}, y_{<t}; w_0). \end{aligned} \tag{8}$$

The baseline term is computed once per instance and reused, so each candidate is compared against a fixed instance-specific reference. To remove position-induced drift, we apply causal sliding-window centering of width L followed by symmetric clipping,

$$\tilde{r}_t(w) = \text{clip}(r_t(w) - \bar{r}_{t-1:L},\ -\tau,\ \tau)\,, \tag{9}$$

which the bandit consumes as its observation. To amortize teacher-forcing cost we update once per group of U consecutive tokens (*update-every-U*), reusing the same w_t within a group and observing the group-averaged centered reward.

Instance-specific posterior over factor utilities. We model the centered reward as a black-box function of the deviation $x = w - w_0$ and place a Gaussian-process prior with an RBF kernel over $r(\cdot)$. To keep the controller closed-form, we approximate the kernel by random Fourier features (Rahimi and Recht, 2007): with random projections $\Omega \in \mathbb{R}^{2K\times D} \sim \mathcal{N}(0, \ell^{-2}I)$ and phases $b \in [0, 2\pi)^D$, define $\phi(x) = \sqrt{2/D}\cos(\Omega^\top x + b)$, so that $r(w) = \phi(w - w_0)^\top\theta + \epsilon$ with $\theta \sim \mathcal{N}(\mu, \Sigma)$ and $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Bayesian conjugate updates give

$$A \leftarrow A + \phi(x_t)\phi(x_t)^\top, \quad c \leftarrow c + \phi(x_t)\,\bar{\tilde{r}}_t,$$
$$\mu \leftarrow A^{-1}c, \qquad \Sigma \leftarrow \eta\sigma^2 A^{-1}, \tag{10}$$

with A initialized as $\lambda_{\text{prior}} I_D$ at the first adaptation round and carried across rounds for the same user, so that $\mathcal{S}_{T+1} = (A, c)$ summarizes all evidence accumulated up to and including round T. As a linear ablation, replacing ϕ by the identity map recovers Bayesian linear regression on x with a closed-form Thompson action.

Thompson sampling with regularization and trust region. At each group g, the controller samples $\tilde{\theta}_g \sim \mathcal{N}(\mu_g, \Sigma_g)$ and selects

$$w_g^\star = \underset{w\in[w_{\min}, w_{\max}]^{2K}}{\arg\max} \big\{\phi(w - w_0)^\top \tilde{\theta}_g - \lambda_{\text{reg}}\|w - w_0\|_2^2\big\}, \tag{11}$$

solved by projected gradient ascent. The quadratic term keeps the controller close to the trained interface unless the sample provides clear evidence to depart from it. To avoid rapid oscillation across groups, we constrain the action to an ℓ_∞ trust region around the previous weight,

$$w_g = w_{g-1} + \text{clip}(w_g^\star - w_{g-1}, -\Delta, \Delta), \tag{12}$$

followed by a final box projection, matching the locally smooth structure of natural language while still letting the controller traverse the action space over many groups.

Exploitation at generation time. Online sampling drives exploration during teacher-forcing across the $T_{\max}$ adaptation rounds. After the final round, generation of the test prediction $\hat{y}_u^\star$ for the held-out target $i^\star$ uses the posterior-mean exploitation action,

$$\hat{w} = \underset{w\in[w_{\min}, w_{\max}]^{2K}}{\arg\max} \big\{\phi(w - w_0)^\top \mu - \lambda_{\text{reg}}\|w - w_0\|_2^2\big\}, \tag{13}$$

where μ is the posterior mean carried in $\mathcal{S}_{T_{\max}+1}$. This action is then frozen and applied through Eq. 7 during greedy decoding; the bandit is not updated during generation. The LLM is never fine-tuned, no per-user parameters are stored across users, and the only quantities that change at inference are the $2K$ scalars in w.

3 Experiments

We evaluate FaLCon on personalized review generation around four research questions that follow from the method design. **Q1.** Does the offline-trained factor-level soft-prompt interface, on its own, improve over prior personalization methods? **Q2.** Does the online optimization phase, on top of the offline interface, yield further gains? **Q3.** Within a single online round, is the intermediate optimization process stable and effective, and do the converged factor weights remain user-specific rather than collapsing to a shared point? **Q4.** Which design choices in the factor interface and the controller are individually necessary?

3.1 Experimental Setup

Datasets and task. We evaluate on three Amazon Reviews (Hou et al., 2024) domains, Books, Movies & TV, and CDs & Vinyl. Each example contains a target item, the user's review history, and the reference review. Data are divided into *offline training set*, *online adaptation set*, and *test set*. For each test example, the most recent review is held out for evaluation, while the preceding 10 chronological reviews form the adaptation set, used only to simulate online adaptation and never for offline training. Thus, *FaLCon w/o online latent control* follows the offline-only protocol and ignores adaptation set, whereas *FaLCon (full)* processes them sequentially to update the per-user posterior before predicting the held-out review.

Baselines. We compare against representative personalization strategies covering the main families relevant to our setting: a non-personalized LLM that ignores user history; retrieval-augmented

methods that condition on profile snippets retrieved from the user's history, including RAG (Salemi et al., 2024), PAG (Richardson et al., 2023), and DPL (Qiu et al., 2025b); a test-time alignment method T-POP (Qu et al., 2025); and soft-prompt methods that distill user history into prompt embeddings injected into a frozen backbone, including PPlug (Liu et al., 2025b) and DEP (Qiu et al., 2025a). Notably, **retrieval-augmented baselines also access the online adaptation set**, ensuring a strictly fair comparison.

Metrics and implementation. We report ROUGE-1, ROUGE-L (Lin, 2004), METEOR (Banerjee and Lavie, 2005), BLEU (Papineni et al., 2002), and BERTScore (Zhang et al., 2019), all as percentages with higher better. FaLCon uses BGE-M3 embeddings (Chen et al., 2024) and $K = 8$ user plus $K = 8$ target-item factors aligned to a frozen Qwen2.5-7B-Instruct or Qwen2.5-32B-Instruct backbone (Yang et al., 2024). The Thompson-sampling controller acts on the resulting 16-dimensional factor-weight vector via layer-0 K/V attention scaling.

3.2 Main Results

All results in this section are reported as averages across all test samples in each domain (and over the three domains for the averaged columns); the figures additionally show 95% confidence intervals across all test samples.

Three observations from Table 1 address Q1 and Q2. **(i) The offline factor interface alone is already competitive (Q1).** *FaLCon w/o online latent control*, which is fully offline-trained and single-shot tested under exactly the same protocol as the baselines, on average outperforms all prior personalization routes on both backbones, isolating the contribution of replacing an entangled latent vector with a factor-level soft-prompt interface. **(ii) Online optimization tends to bring further gains on top of the offline interface (Q2).** *FaLCon (full)* improves over *FaLCon w/o online latent control* on most averaged metrics under both backbones, with content-oriented metrics (R-1, METEOR, BERTScore) generally seeing a larger lift than BLEU; this pattern is consistent with online control re-allocating factor emphasis rather than rewriting surface n-grams. Crucially, the retrieval baselines have full access to the same adaptation set (§3), so the comparison reflects a modeling advantage rather than extra information. Figure 2 reports

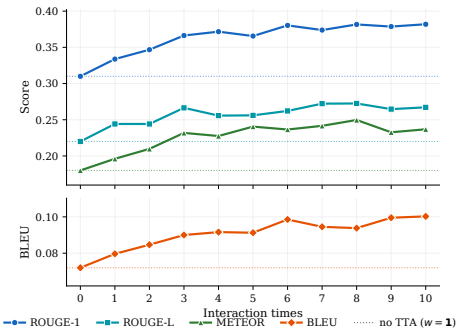


Figure 2: Test-set R-1/R-L/MET/BLEU as a function of the number of adaptation-set records consumed by the controller; round 0 corresponds to *FaLCon w/o ctrl.*, averaged across all test samples with 95% CIs.

test-set ROUGE-1/L, METEOR, and BLEU as a function of the number of adaptation-set records consumed by the controller, from 0 (the offline-only *FaLCon w/o online latent control* starting point) up to all 10 records. After each adaptation round the per-user posterior is updated and the held-out test record is re-evaluated under the resulting exploitation action. The mean curves rise as more online interactions are observed, indicating that the gain over *FaLCon w/o online latent control* accumulates progressively rather than arriving in a single step, and that the controller continues to extract useful signal from each additional adaptation record.

3.3 Within-Round Optimization Behavior

Q3 zooms in on a single online round, asking whether the intermediate controller updates are informative and whether the converged weights remain user-specific. All trajectories below are averaged across all test samples with 95% CIs.

Token-level reward and loss tend to move in the right direction. Figure 3 (top) plots the per-token reward $\Delta \log p(y_t) = \log p(y_t \mid w^*) - \log p(y_t \mid w{=}1)$ together with the teacher-forced loss $-\log p(y_t \mid w^*)$ along a round. The mean reward stays positive and the mean loss decreases, consistent with the controller's intermediate weights being on average informative rather than random perturbations.

Posterior uncertainty tends to shrink under online evidence. Figure 3 (bottom) shows the normalized posterior covariance trace $\mathrm{tr}(\Sigma_t)/\,\mathrm{tr}(\Sigma_0)$ trending downward from one, so on average the Bayesian updates concentrate mass on a smaller region of the action space, the regime in which

	Method	Books			Movies & TV			CDs & Vinyl			Average				
		R-L	MET.	BERT	R-L	MET.	BERT	R-L	MET.	BERT	R-1	R-L	MET.	BLEU	BERT
Qwen2.5-7B	Non-personalized	13.79	17.35	79.89	12.14	15.03	76.85	12.41	15.61	77.69	26.61	12.78	15.99	3.801	78.14
	RAG	14.94	21.01	81.16	13.24	17.01	78.06	13.32	17.33	78.85	28.79	13.83	18.45	2.201	79.36
	PAG	14.88	23.78	79.71	13.57	19.05	77.21	13.73	19.79	78.28	29.26	14.06	20.87	3.029	78.37
	T-POP	14.92	23.95	80.12	13.62	19.24	77.45	13.81	19.85	78.54	29.41	14.12	21.01	3.125	78.65
	DPL	15.16	24.59	80.64	13.98	19.58	77.89	14.06	20.03	79.08	29.97	14.41	21.39	3.354	79.20
	PPlug	14.40	22.34	82.27	12.43	17.24	77.44	12.86	17.11	78.56	27.27	13.23	18.90	4.449	79.42
	DEP	19.59	27.23	84.74	15.16	20.56	83.07	14.82	20.04	82.78	32.00	16.52	22.61	6.109	83.53
	FaLCon *w/o* online latent control	21.68	32.45	85.38	15.32	24.59	83.27	14.74	23.79	82.95	33.51	17.25	26.94	9.261	83.87
	FaLCon (full)	**23.90**	**35.41**	**91.04**	**16.86**	**26.70**	**89.39**	**16.36**	**26.06**	**88.79**	**42.56**	**19.04**	**29.39**	**9.894**	**89.74**
Qwen2.5-32B	Non-personalized	14.72	19.49	80.55	13.08	16.66	77.52	13.38	17.67	78.40	27.42	13.73	17.94	4.210	78.82
	RAG	15.58	27.35	81.92	13.93	21.42	78.83	14.18	21.77	79.62	30.48	14.56	23.51	3.010	80.12
	PAG	15.70	28.30	81.45	14.10	21.30	78.35	14.28	22.15	79.20	30.75	14.69	23.92	3.612	79.67
	T-POP	15.82	28.54	81.68	14.25	21.56	78.52	14.41	22.42	79.45	30.92	14.81	24.15	3.824	79.85
	DPL	16.10	33.92	82.33	14.58	22.38	78.86	14.85	23.37	80.79	31.59	15.18	25.26	4.955	80.66
	PPlug	15.42	31.10	82.70	13.74	20.87	78.95	14.05	21.96	80.88	30.82	14.40	24.64	5.235	80.84
	DEP	19.53	23.71	85.17	15.91	24.15	84.22	15.13	24.31	83.60	33.47	16.86	24.03	6.702	84.33
	FaLCon *w/o* online latent control	23.02	39.63	87.17	15.98	28.11	84.31	15.57	27.76	84.18	36.42	18.19	31.83	11.61	85.22
	FaLCon (full)	**25.30**	**42.70**	**92.85**	**20.40**	**30.85**	**91.05**	**20.50**	**30.62**	**90.85**	**45.80**	**22.07**	**34.72**	**12.30**	**91.58**

Table 1: FaLCon outperforms all prior personalization methods on both backbones, with the structured factor interface and inference-time latent control each providing additive gains.

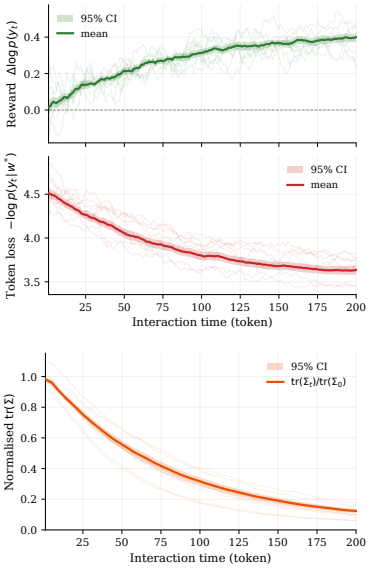


Figure 3: Within-round trajectories of per-token reward, teacher-forced loss, and normalized posterior covariance trace, averaged across all test samples with 95% CIs.

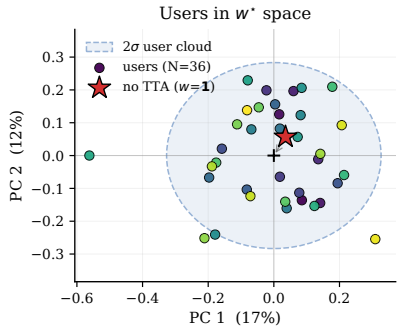


Figure 4: PCA of converged per-user $\hat{w}$ on Books; users span a wide region and $w{=}1$ sits off-center.

Thompson sampling shifts from exploration to exploitation.

Converged weights remain user-specific. Figure 4 PCA-projects the converged per-user weights $\hat{w}$ on Books. They span an extended region rather than clustering around one point, and the unscaled reference $w{=}1$ (red star) sits off the user mean (black plus). Users therefore do not collapse to a shared $\hat{w}$, ruling out the failure mode in which online optimization degenerates into a uniform prompt rescaling.

3.4 Ablation Study

We answer **Q4** with a compact two-view ablation (Table 2, Figure 5), all averaged across the three domains under Qwen2.5-7B. Table 2 disables one design choice at a time: item-decoupled user signals, redundancy-reduced factor encoding, and the online controller itself. Figure 5 sweeps the per-side factor count $K \in \{0, 1, 2, 4, 8\}$.

Two patterns emerge. First, the downstream online control is closely coupled with the upstream offline interface structure. Notably, removing redundancy reduction (*w/o redundancy reduction*) causes performance to drop considerably, approaching the level of *w/o online latent control*. This suggests that without a properly orthogonalized and decoupled factor space, the continuous weights w are

Variant	R-1	R-L	MET.	BLEU	BERT
FaLCon (full)	42.56	19.04	29.39	9.894	89.74
w/o item decoupling	39.83	18.61	28.74	9.704	88.72
w/o redundancy reduction	34.95	17.18	26.05	9.521	83.41
w/o online latent control	33.51	17.25	26.94	9.261	83.87

Table 2: Ablation of offline interface designs. Removing factor redundancy reduction collapses the performance to near the unadapted baseline, demonstrating that downstream continuous control relies on the structured offline factor space.

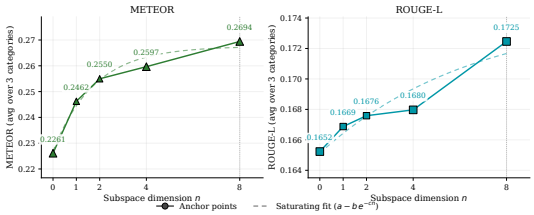


Figure 5: Increasing the per-side factor count K can improve quality, but the marginal gain becomes small around $K{=}8$.

less effective at isolating and modulating specific semantic directions, which limits the efficacy of inference-time control. Second, item-decoupled user signals also provide a noticeable performance buffer, helping to ensure user preferences are not overshadowed by item-specific semantics during modulation. Meanwhile, increasing the per-side factor count K improves quality as it expands the action space capacity, but the gains begin to saturate around $K{=}8$.

Efficiency. FaLCon keeps the LLM frozen and confines online adaptation to the controller. Each online interaction only requires ordinary LLM forward passes to obtain token-level likelihood rewards, followed by a closed-form Bayesian-linear posterior update over the $2K{=}16$ factor weights; no gradients are back-propagated through the backbone or prompt modules, and no online optimizer state is maintained. The interface adds only $2K$ extra prompt tokens, and each Bayesian update over a scored adaptation-token group is $O(2K)$, so the computation is dominated by ordinary forward passes for teacher-forced scoring and final generation/evaluation rather than by additional training. Because modulation enters only through layer-0 K/V scaling on factor positions, it leaves cached states for the textual context untouched and is compatible with standard KV-cache decoding pipelines.

4 Related Work

LLM personalization. LLM personalization has been studied in recommendation, review generation, and user-facing assistants (Li et al., 2021, 2023; Peng et al., 2024; Cheng et al., 2023; Yang et al., 2025; Shimizu et al., 2025; Liu et al., 2025a). Across these settings, a model must recover persistent user preferences from sparse evidence while remaining responsive to the current target. Recent routes include retrieval- or profile-augmented conditioning (Salemi et al., 2024; Mysore et al., 2024; Shi et al., 2025; Qiu et al., 2025b), as well as preference and online alignment methods (Bu et al., 2025; Qu et al., 2025; Pădurean et al., 2026). Retrieval and profile-based approaches retain evidence in discrete text, whereas alignment approaches exploit additional preference signals. Discrete evidence is interpretable but increases context length, while learned states are compact but harder to update selectively. More generally, existing systems tend to treat personalization as a single conditioning problem, leaving the granularity of adaptation implicit. These methods often depend on long contexts, model updates, or explicit queries, and provide limited inference-time control over individual components of a user representation. FaLCon instead keeps the LLM frozen and updates a low-dimensional continuous factor interface from newly observed adaptation records.

Latent soft-prompt personalization. Soft prompts and prefixes provide a parameter-efficient adaptation mechanism (Li and Liang, 2021; Lester et al., 2021; Liu et al., 2022; Hu et al., 2022) and have been used for personalization through history-induced prompts (Hebert et al., 2024; Huang et al., 2024; Xiao and Hou, 2026), user-embedding injection (Doddapaneni et al., 2024; Ning et al., 2025; Liu et al., 2025b; Huber et al., 2025), and inter-user residuals (Qiu et al., 2025a). Related factorization and disentanglement work motivates separating heterogeneous preferences rather than collapsing them into one vector (Bengio et al., 2013; Locatello et al., 2019; Zheng et al., 2021; Ma et al., 2019; Zhuang et al., 2024; Shenfeld et al., 2025). Yet predictive soft-prompt training does not require prompt positions to be separately controllable, so stable preferences, target-specific content, and context-dependent expression can remain entangled. This entanglement also limits diagnosis and targeted intervention when only one aspect of a user's preference changes.

Consequently, the representation may encode useful preference information without revealing which latent component should change after a particular interaction. FaLCon introduces an explicit structural bias that exposes addressable user and target factors as a soft-prompt interface, enabling localized online adaptation without gradient updates.

5 Conclusion

We presented FaLCon, an LLM personalization framework that recasts soft-prompt personalization as factor-level latent control: an item-decoupled, redundancy-reduced multi-branch encoder turns user and target signals into separately addressable soft-prompt factors, and an inference-time controller estimates factor-weight utilities with Thompson sampling and applies the final exploitation weights through layer-0 K/V attention scaling. This separation allows the offline representation to remain fixed while the relevance of its factors is adapted from newly observed user interactions, without updating the backbone LLM or the trained factor encoders.

Experiments on personalized review generation show consistent gains over strong personalization baselines. The ablations further indicate that the structured factor interface and the inference-time controller provide complementary improvements, while the adaptation analyses show that the controller maintains user-specific factor-weight behavior rather than collapsing to a shared solution. These results support a broader design principle for personalization: first expose a compact, separately addressable latent interface, and then adapt that interface with a lightweight online controller. The current formulation assumes supervised signals from newly completed interactions and retains latent rather than human-named factors; reference-free adaptation and semantically anchored control remain promising directions for future work.

Limitations

While our results are encouraging, we discuss several scope conditions that delineate where the present study makes claims and where it does not. First, the empirical evaluation is conducted on personalized review generation with two frozen Qwen2.5 backbones; although the FaLCon design is intentionally backbone-agnostic and the gains transfer consistently from 7B to 32B, a broader assessment across additional model families and longer-form personalized generation tasks would further characterize the regime in which factor-level latent control is most beneficial. Second, the inference-time controller assumes access to a teacher-forced reference signal during adaptation; extending the same posterior formulation to fully reference-free or weakly supervised feedback is a natural and complementary direction rather than a conceptual obstacle. Third, the factors are induced without explicit semantic supervision, so the resulting axes are statistical rather than human-named; in deployments where interpretability is a primary requirement, lightweight factor anchoring could be added on top of the existing interface. We view these as scope boundaries that follow from deliberate design choices, and they suggest concrete extensions rather than limitations of the underlying framework.

Ethical Statements

This work studies LLM personalization through latent factor representations distilled from users' historical reviews. Although our framework keeps the backbone LLM frozen and does not solicit any active user feedback, personalization based on historical interactions still requires careful attention to data privacy, consent, and security. Modeling user-specific latent factors may also implicitly capture sensitive behavioral patterns, and any real-world deployment should incorporate privacy-preserving techniques such as anonymization, access control, and transparent consent protocols, together with safeguards against unintended inferences or misuse of user-level representations.

All experiments are conducted on publicly available datasets that have been preprocessed and released by prior work. The original raw data is open-source and distributed under permissive licenses, and we adhere to the established usage guidelines and ethical standards of these resources.

We use AI assistants (e.g., ChatGPT) as auxiliary tools for writing refinement, while all research ideas, experimental designs, and final decisions are made by the authors.

Acknowledgments

This work was supported by the Innovation Fund Project (Grant No. 2025-JCJQ-LA-001-097) and the Beijing Natural Science Foundation (Grant No. 4264131).

References

Shipra Agrawal and Navin Goyal. 2013. Thompson sampling for contextual bandits with linear payoffs. In *International conference on machine learning*, pages 127–135. PMLR.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Adrien Bardes, Jean Ponce, and Yann LeCun. 2021. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828.

Hyungjune Bu, Chanjoo Jung, Minjae Kang, and Jaehyung Kim. 2025. Personalized llm decoding via contrasting personal preference. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 33946–33966.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*, 4(5).

Hao Cheng, Shuo Wang, Wensheng Lu, Wei Zhang, Mingyang Zhou, Kezhong Lu, and Hao Liao. 2023. Explainable recommendation with personalized review retrieval and aspect learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 51–64.

Sumanth Doddapaneni, Krishna Sayana, Ambarish Jash, Sukhdeep Sodhi, and Dima Kuzmin. 2024. User embedding model for personalized language prompting. In *Proceedings of the 1st Workshop on Personalization of Generative AI Systems (PERSONALIZE 2024)*, pages 124–131.

Liam Hebert, Krishna Sayana, Ambarish Jash, Alexandros Karatzoglou, Sukhdeep Sodhi, Sumanth Doddapaneni, Yanli Cai, and Dima Kuzmin. 2024. Persoma: Personalized soft prompt adapter architecture for personalized language prompting. *arXiv preprint arXiv:2408.00960*.

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

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3.

Qiushi Huang, Xubo Liu, Tom Ko, Bo Wu, Wenwu Wang, Yu Zhang, and Lilian Tang. 2024. Selective prompting tuning for personalized conversations with llms. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 16212–16226.

Bernd Huber, Ghazal Fazelnia, Andreas Damianou, Sebastian Peleato, Max Lefarov, Praveen Ravichandran, Marco De Nadai, Mounia Lalmas-Roellke, and Paul N Bennett. 2025. Embedding-to-prefix: Parameter-efficient personalization for pretrained large language models. *arXiv preprint arXiv:2505.17051*.

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

Lei Li, Yongfeng Zhang, and Li Chen. 2021. Personalized transformer for explainable recommendation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4947–4957.

Lei Li, Yongfeng Zhang, and Li Chen. 2023. Personalized prompt learning for explainable recommendation. *ACM Transactions on Information Systems*, 41(4):1–26.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Jiahong Liu, Zexuan Qiu, Zhongyang Li, Quanyu Dai, Wenhao Yu, Jieming Zhu, Minda Hu, Menglin Yang, Tat-Seng Chua, and Irwin King. 2025a. A survey of personalized large language models: Progress and future directions. *arXiv preprint arXiv:2502.11528*.

Jiongnan Liu, Yutao Zhu, Shuting Wang, Xiaochi Wei, Erxue Min, Yu Lu, Shuaiqiang Wang, Dawei Yin, and Zhicheng Dou. 2025b. Llms+ persona-plug= personalized llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9373–9385.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. 2019. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pages 4114–4124. PMLR.

Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. Learning disentangled representations for recommendation. *Advances in neural information processing systems*, 32.

Sheshera Mysore, Zhuoran Lu, Mengting Wan, Longqi Yang, Bahareh Sarrafzadeh, Steve Menezes, Tina Baghaee, Emmanuel Barajas Gonzalez, Jennifer Neville, and Tara Safavi. 2024. Pearl: Personalizing large language model writing assistants with generation-calibrated retrievers. In *Proceedings of the 1st Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual (CustomNLP4U)*, pages 198–219.

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

Victor-Alexandru Pădurean, Parameswaran Kamalaruban, Nachiket Kotalwar, Alkis Gotovos, and Adish Singla. 2026. Inference-time personalized alignment with a few user preference queries. *Advances in Neural Information Processing Systems*, 38:85125–85156.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Qiyao Peng, Hongtao Liu, Hongyan Xu, Qing Yang, Minglai Shao, and Wenjun Wang. 2024. Review-llm: Harnessing large language models for personalized review generation. *arXiv preprint arXiv:2407.07487*.

Yilun Qiu, Tianhao Shi, Xiaoyan Zhao, Fengbin Zhu, Yang Zhang, and Fuli Feng. 2025a. Latent inter-user difference modeling for llm personalization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 10610–10628.

Yilun Qiu, Xiaoyan Zhao, Yang Zhang, Yimeng Bai, Wenjie Wang, Hong Cheng, Fuli Feng, and Tat-Seng Chua. 2025b. Measuring what makes you unique: Difference-aware user modeling for enhancing llm personalization. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 21258–21277.

Zikun Qu, Min Zhang, Mingze Kong, Xiang Li, Zhiwei Shang, Zhiyong Wang, Yikun Ban, Shuang Qiu, Yao Shu, and Zhongxiang Dai. 2025. T-pop: Test-time personalization with online preference feedback. *arXiv preprint arXiv:2509.24696*.

Ali Rahimi and Benjamin Recht. 2007. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3505–3506.

Chris Richardson, Yao Zhang, Kellen Gillespie, Sudipta Kar, Arshdeep Singh, Zeynab Raeesy, Omar Zia Khan, and Abhinav Sethy. 2023. Integrating summarization and retrieval for enhanced personalization via large language models. *arXiv preprint arXiv:2310.20081*.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2024. Lamp: When large language models meet personalization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7370–7392.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. 2021. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634.

Idan Shenfeld, Felix Faltings, Pulkit Agrawal, and Aldo Pacchiano. 2025. Language model personalization via reward factorization. *arXiv preprint arXiv:2503.06358*.

Teng Shi, Jun Xu, Xiao Zhang, Xiaoxue Zang, Kai Zheng, Yang Song, and Han Li. 2025. Retrieval augmented generation with collaborative filtering for personalized text generation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1294–1304.

Ryotaro Shimizu, Takashi Wada, Yu Wang, Johannes Kruse, Sean O'Brien, Sai HtaungKham, Linxin Song, Yuya Yoshikawa, Yuki Saito, Fugee Tsung, and 1 others. 2025. Disentangling likes and dislikes in personalized generative explainable recommendation. In *Proceedings of the ACM on Web Conference 2025*, pages 4793–4809.

Canran Xiao and Liwei Hou. 2026. Prototype-aligned federated soft-prompts for continual web personalization. In *Proceedings of the ACM Web Conference 2026*, pages 6743–6754.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and

22 others. 2024. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

Ching-Wen Yang, Zhi-Quan Feng, Ying-Jia Lin, Che Wei Chen, Kun-da Wu, Hao Xu, Yao Jui-Feng, and Hung-Yu Kao. 2025. Maple: Enhancing review generation with multi-aspect prompt learning in explainable recommendation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 31803–31821.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. 2021. Barlow twins: Self-supervised learning via redundancy reduction. In *International conference on machine learning*, pages 12310–12320. PMLR.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Yu Zheng, Chen Gao, Xiang Li, Xiangnan He, Yong Li, and Depeng Jin. 2021. Disentangling user interest and conformity for recommendation with causal embedding. In *Proceedings of the web conference 2021*, pages 2980–2991.

Yuchen Zhuang, Haotian Sun, Yue Yu, Rushi Qiang, Qifan Wang, Chao Zhang, and Bo Dai. 2024. Hydra: Model factorization framework for black-box llm personalization. *Advances in Neural Information Processing Systems*, 37:100783–100815.

A Dataset Details

This work focuses on personalized review generation and uses preprocessed Amazon review data (Hou et al., 2024). For the three product categories used in our experiments, we extract each user's latest eligible interaction from the category-level training split and merge the resulting examples into a unified training pool for model learning. To construct a high-quality evaluation benchmark, we apply the following filtering rules.

- Each review must contain a review title, a star rating, and a timestamp. The review body must contain at least 200 characters so that the task requires long-form generation rather than short sentiment completion.
- Each product must have a valid product name, and its product description must contain between 100 and 2000 characters, ensuring that the target-side context is informative but not excessively long.
- Each product must be reviewed by at least four distinct users. If the same user reviews the same product multiple times, only the latest review is retained.
- Each retained user must have between 19 and 500 reviews in total. This removes users with too little history to support personalization as well as suspiciously active users that may correspond to abnormal reviewing behavior.
- For each evaluation user, at least 19 historical reviews are retained before the held-out test record: at least eight records are available for offline user-history construction/training, and ten chronological records are reserved as the online adaptation set.

During preprocessing, we concatenate the target product name, product description, target review title, star rating, and retrieved user-history reviews to form the complete prompt used as model input.

Table 3 summarizes the final user-level structure used by FaLCon after joining reviews with item information and splitting each evaluation user's chronological history.

Before constructing the split in Table 3, the cleaned data are stored as two joined sources: a review collection containing user interactions and an item collection containing product metadata. Table 4 gives representative records from these two sources; long text fields are truncated for readability.

B Baseline Details

This section describes the baselines used in Table 1. All methods are evaluated under the same personalized review-generation input format whenever applicable, including the target item title, item description, target review title, and rating. For baselines that use user history, we allow access to the same historical/adaptation records as FaLCon to avoid giving our method extra information.

Non-personalized. This baseline removes user-specific information entirely. The model receives only target-side information, including the item title, item description, target review title, and rating, and then generates the review. It measures how much quality can be obtained from the target item alone without modeling user preferences.

RAG. The retrieval-augmented baseline follows a direct history-in-context strategy similar to

Component	Count	Usage	Content
User identifier	1	Groups all records belonging to the same user.	User id.
Offline history records	$n \geq 8$	Construct the offline user history and support offline factor-interface training or single-shot testing.	n historical review records. Each record contains the product asin, title, rating, description, and the user's review text.
Online adaptation records	10	Simulate newly observed completed interactions. They are consumed chronologically to update only the per-user factor-weight posterior.	The ten most recent history records before the test record. Each record contains the product asin, title, rating, description, and the user's review text.
Held-out test record	1	Evaluate final personalized generation after adaptation. This record is never used for posterior updates.	One target product with asin, title, and description. The ground truth is the user's real review text for this product.

Table 3: Final processed data structure used for training, online adaptation, and evaluation.

Source	Role	Representative processed record
Review collection	Stores user-level review interactions before they are split into offline history, adaptation records, and held-out test records.	`{ user_id: "AFZUK3MTBIBEDQOPAK3OATUOUKLA", profile: [{ asin: "B000056HE9", rating: 5.0, text: "This movie is one of my all time holiday favorites..." }], data: { asin: "B0002F6BRE", title: "Elf movie", timestamp: 1607398123747, rating: 5.0, text: "I purchase this movie..." } }`
Item collection	Stores product-side metadata that is joined to review records by asin during prompt construction.	`{ asin: "B071GVGD2B", title: "Transformers: The Last Knight", description: "Our world's greatest hero becomes our fiercest..." }`

Table 4: Cleaned source-level records before constructing the final offline/adaptation/test split.

LAMP (Salemi et al., 2024). It retrieves recent reviews from the target user's history using a recency-based rule, formats the selected records as textual evidence, and appends them to the LLM input. This baseline tests whether explicit historical examples are sufficient for personalization without learning latent user representations.

PAG. PAG (Richardson et al., 2023) extends the retrieval baseline by first converting recent user history into a compact natural-language profile. The profile is provided together with the retrieved reviews and target-side information, so the LLM can condition on both specific past examples and a higher-level summary of the user's preferences, style, and reviewing tendencies.

T-POP. T-POP (Qu et al., 2025) is a test-time alignment method that interweaves personalized text generation and user preference learning using a dueling bandit framework. It steers the decoding of a frozen LLM to simultaneously generate an exploitation sequence (greedily maximizing a reward model) and an exploration sequence (incorporating an uncertainty bonus). By eliciting online user preference feedback between the sequences, it continuously updates the reward model without fine-tuning the base LLM. In our setting, we simulate the online feedback using the provided adaptation records to iteratively refine the reward model before generating the review for the target item.

DPL. DPL follows difference-aware prompting (Qiu et al., 2025b). It compares the target user with representative users selected from user clusters and asks the LLM to summarize how the target user differs along dimensions such as writing style, emotional tone, and semantic focus. The resulting difference-aware profile, together with recent reviews, is inserted into the model input. For a fair comparison, we use the same embedding backbone as FaLCon when selecting representative users.

PPlug. PPlug (Liu et al., 2025b) is a plug-and-play soft-prompt personalization baseline. It encodes the user's history into a dense user-specific embedding with a lightweight user embedder and input-aware attention over historical records. The resulting embedding is projected into the LLM input space as a continuous prompt, together with an instruction embedding, while the backbone LLM remains frozen. In our implementation, PPlug uses the same user-history encoder backbone as FaLCon where applicable.

DEP. DEP (Qiu et al., 2025a) models personalization through latent inter-user differences rather than natural-language difference summaries. It constructs a user-specific embedding for the user's behavioral context and a difference-aware embedding that captures how the user deviates from comparable users. These representations are compressed with a sparse autoencoder to retain informative preference signals, then injected into a frozen LLM as soft prompts and aligned with the LLM through training of the autoencoder/projector modules. DEP is therefore the strongest latent soft-prompt baseline, but its prompt is still a holistic latent surrogate rather than a set of separately addressable factor slots with an inference-time controller.

C Implementation Details

This section provides the implementation details for FaLCon, separated into the offline factor interface training and the online inference-time latent control.

C.1 Offline Factor Interface Training

The offline training stage constructs the factor-level soft-prompt interface through four sequential steps: embedding generation, user-target decoupling, redundancy-reduced factor encoding, and LLM fine-tuning.

Embedding generation and decoupling. We use the multilingual BGE-M3 (`BAAI/bge-m3`) (Chen et al., 2024) model as the external encoder $E(\cdot)$ to extract 1024-dimensional L2-normalized `[CLS]` token representations for user reviews and item metadata (title and description). The maximum sequence length is set to 2048. Following Eq. 3, we compute the target-decoupled user signals by projecting each historical review embedding onto the orthogonal complement of its associated target item embedding, followed by L2 re-normalization.

Redundancy-reduced factor encoding. We train two separate multi-branch autoencoders for the user and target sides. Each autoencoder takes a 1024-dimensional input (e_u or q_i), processes it through a shared 512-dimensional hidden layer h_s, and projects it into $K = 8$ factors to produce factor embeddings z_k^s of dimension $d_{\text{sub}} = 512$. The autoencoders are optimized using Adam with a learning rate of 10^{-4} and weight decay of 10^{-6} for up to 100 epochs, including a 5-epoch warmup using only the reconstruction loss. For the composite factorization loss in Eq. 4, we use a reconstruction weight $\lambda_r = 10.0$, a variance weight $\lambda_v = 0.001$, and a balance weight $\lambda_b = 0.1$. The decorrelation term $\lambda_d \mathcal{L}_{\text{dec}}$ encompasses both a cross-branch covariance penalty (weight 0.1) and a head-weight orthogonality penalty (weight 1.0). The variance threshold γ is set to 0.1 for the user autoencoder and 0.05 for the item autoencoder. The user autoencoder is trained with a batch size of 1024, while the item autoencoder uses 256.

LLM fine-tuning. The backbone recommendation LLM is initialized from `Qwen2.5-7B-Instruct` (Yang et al., 2024). Before being injected into the LLM, the 512-dimensional factor embeddings z_k^s are compressed to 256 dimensions using a sparse autoencoder (with sparsity target $\rho = 0.05$) and then mapped to the LLM's 3584-dimensional hidden space via a 2-layer alignment MLP (A_u and A_i). These continuous embeddings replace special factor placeholder tokens interleaved in the input prompt. We fine-tune the SAEs and alignment MLPs alongside the factor encoders using the causal language modeling objective. Training is conducted in `bfloat16` mixed precision with Flash Attention 2 and DeepSpeed (Rasley et al., 2020) ZeRO Stage 2. We train for 5 epochs using AdamW with a learning rate of 10^{-5}, a weight decay of 0.025, a warmup ratio of 0.01, and an effective batch size of 16.

C.2 Inference-Time Latent Control

During deployment, the online adaptation protocol applies the learned interface as the action space for inference-time latent control.

Algorithm overview. Algorithm 1 summarizes the complete procedure of online adaptation and

subsequent generation. The controller sequentially sweeps through the $T_{\max}$ chronologically ordered adaptation records. For each record, the controller updates the factor-weight posterior by performing Thompson sampling and teacher-forcing evaluation. After the adaptation sweep, the final posterior mean is used to compute the exploitation weight $\hat{w}$, which is applied during generation for the held-out test record.

Algorithm 1 Inference-time latent control via online adaptation.

Input: Online-adaptation set $\{(i_T, y_{u,i_T})\}_{T=1}^{T_{\max}}$, test record $i^\star$, factor positions $\{\pi_k\}_{k=1}^{2K}$, group size U, prior $(\lambda_{\text{prior}}, \sigma^2, \eta)$, bandwidth ℓ, regularizer λ_{reg}, trust radius Δ, window L, clip τ.
Initialize posterior $A \leftarrow \lambda_{\text{prior}} I_D$, $c \leftarrow 0$.
$w_0 \leftarrow \mathbf{1}$, $w_{\text{prev}} \leftarrow w_0$.
for round $T = 1$ to $T_{\max}$ **do**
  Let $X_T = \text{Constructor}(x_{u,i_T}; \{p_k^u\}_{k=1}^K, \{p_k^{i_T}\}_{k=1}^K)$ and $y = y_{u,i_T}$.
  Compute baseline $\log p_\Theta(y_t \mid X_T, y_{<t}; w_0)$ for all $t \in \{1, \ldots, |y|\}$.
  for group $g = 1$ to $\lceil |y|/U \rceil$ **do**
    Compute $\mu_g \leftarrow A^{-1}c$, $\Sigma_g \leftarrow \eta\sigma^2 A^{-1}$.
    Sample $\tilde{\theta}_g \sim \mathcal{N}(\mu_g, \Sigma_g)$.
    Solve Eq. 11 by projected gradient ascent to obtain $w_g^\star$.
    Apply trust region (Eq. 12) to obtain w_g.
    Apply w_g via Eq. 7; teacher-force the U tokens of group g.
    Compute group-averaged centered reward $\bar{\tilde{r}}_g$ from Eq. 9.
    Update posterior (A, c) via Eq. 10.
    $w_{\text{prev}} \leftarrow w_g$.
  end for
end for
Let $X^\star = \text{Constructor}(x_{u,i^\star}; \{p_k^u\}_{k=1}^K, \{p_k^{i^\star}\}_{k=1}^K)$.
Compute exploitation weight $\hat{w}$ from Eq. 13 using final μ.
Decode $\hat{y}_u^\star$ greedily under $\hat{w}$ via Eq. 7.

GP-RFF configuration. The controller models the reward function as a Gaussian Process approximated by Random Fourier Features (RFF). We use $D = 128$ feature dimensions to provide a sufficient approximation of the RBF kernel. The random projection matrix Ω and phase offsets b are drawn once at initialization and fixed throughout the adaptation process. The length scale of the RBF kernel is set to $\ell = 0.5$.

Posterior and reward hyperparameters. The Bayesian linear regression operates with a prior precision $\lambda_{\text{prior}} = 1.0$, an observation noise variance $\sigma^2 = 1.0$, and a posterior temperature $\eta = 1.0$. These parameters initialize the posterior covariance as $\Sigma_0 = \eta\sigma^2/\lambda_{\text{prior}} I_D$. To ensure numerical stability, the sliding-window centered rewards are clipped at $\tau = 20.0$. We maintain a reward centering window size of $L = 20$. We use a default group size of $U = 1$, meaning the posterior is updated token-wise.

Action selection and constraints. During Thompson sampling, the action selection objective (Eq. 11) incorporates an L2 regularizer with $\lambda_{\text{reg}} = 1.0$ to penalize large deviations from the unadapted baseline $w_0 = \mathbf{1}$. Because the RFF mapping ϕ introduces non-linearity, the maximization is performed via projected gradient ascent using analytical gradients. We apply a learning rate of 0.05 for 20 gradient steps per action selection. After optimization, we constrain the weights within the bounds $[w_{\min}, w_{\max}] = [0, 1]$. To prevent abrupt weight oscillations across steps, we enforce an ℓ_∞ trust region with a step radius $\Delta = 0.05$, bounding the maximum weight change between consecutive groups.

Evaluation efficiency. For each teacher-forcing step, we amortize the computation by leveraging KV-cache. The unadapted baseline log-probabilities are precomputed in a single forward pass with $w_0 = \mathbf{1}$ prior to the online adaptation for each record. During the adaptation sweep, we only perform single-step decoding to obtain the log-probability under the sampled action w_g, reducing the per-step overhead. With $D = 128$ and a $2K$-dimensional action space ($K = 8$, thus 16 dimensions), the computational cost of the Bayesian update and projected gradient ascent is strictly limited to low-dimensional matrix operations, remaining negligible compared to the LLM forward pass.

C.3 Evaluation Metrics

To comprehensively evaluate the quality of the generated personalized reviews, we employ a suite of automated metrics spanning lexical overlap, structural similarity, and semantic alignment. Before metric computation, all generated predictions undergo standard text post-processing (e.g., whitespace trimming and truncation). The reference texts are directly extracted from the dataset without additional modification. The specific configurations for each metric are detailed below.

BLEU (Papineni et al., 2002). We use the `sacrebleu` implementation from the HuggingFace Evaluate library to measure lexical exact-match precision. We report the standard BLEU score (BLEU-

4, maximum n-gram order of 4) utilizing international tokenization (`intl`) and the default floor smoothing method (`smooth_method='floor'`).

ROUGE (Lin, 2004). We use the `rouge_score` implementation from HuggingFace Evaluate. We report two variants: **ROUGE-1**, which measures unigram F1 score using the default Porter stemmer, capturing foundational vocabulary overlap; and **ROUGE-L**, which computes the Longest Common Subsequence (LCS) F1 score, capturing sentence-level structural similarity and word order preservation.

METEOR (Banerjee and Lavie, 2005). We use the HuggingFace Evaluate wrapper for the NLTK `meteor_score`, which evaluates both exact and stemmed word matches alongside synonymy. We retain the standard NLTK parameters: an F1 weighting of $\alpha = 0.9$ (heavily favoring recall), a fragmentation penalty parameterized by $\beta = 3.0$ and $\gamma = 0.5$.

BERTScore(Zhang et al., 2019). To evaluate deep semantic equivalence beyond surface-level lexical matching, we compute BERTScore using the official `bert-score` package. We specifically utilize the `roberta-base` model (with `lang="en"`) rather than the uncased BERT model to better capture casing and stylistic nuances inherent to personalized reviews. We report the mean F1 score without baseline rescaling (`rescale_with_baseline=False`) and utilize the fast tokenizer (`use_fast_tokenizer=True`) for efficiency.

D Further Discussion on Fairness and Adaptation

Due to space constraints in the main text, this section provides a deeper conceptual discussion on two critical aspects of FaLCon: the fairness of the experimental protocol regarding the online adaptation set, and the theoretical justification for why adapting factor weights on past items generalizes to unseen test items.

D.1 Fairness of the Online Adaptation Protocol

A natural question regarding our online adaptation protocol (§2.1) is whether the introduction of an adaptation set provides FaLCon with an unfair data advantage over the baselines. We emphasize that the adaptation set does not introduce new, exclusive information; rather, it simulates a realistic online streaming deployment where systems continuously receive user feedback over time.

In terms of experimental fairness, the data availability is strictly controlled across all methods. As detailed in Appendix B, baselines that support historical context (such as RAG, PAG, and DPL) are explicitly provided with the exact same adaptation records in their context windows during evaluation. They have full access to the recent interaction history leading up to the test record.

The fundamental difference lies in architectural capability rather than data access. Traditional latent soft-prompt methods (e.g., DEP, PPlug) represent personalization as a monolithic, entangled vector. For these methods to meaningfully assimilate new streaming records, they would require computationally prohibitive gradient backpropagation through the LLM for every new interaction, making them infeasible for online adaptation. In contrast, FaLCon is designed with a factor-level interface that allows zero-order, gradient-free assimilation of this exact same data. Thus, the performance gains of FaLCon stem from its ability to efficiently utilize streaming feedback, demonstrating a methodological superiority rather than an unfair empirical setup.

D.2 Why Does Online Adaptation Transfer Across Items?

Another intuitive question is: why should optimizing factor weights w on past items $(i_1, \ldots, i_T)$ improve the generation quality for a newly observed, held-out test item $i^\star$? If items are fundamentally different, one might suspect that weights fitted to past interactions would overfit and fail to generalize.

The transferability of factor weights in FaLCon relies on two core properties established during the offline training phase: the *structural alignment of the factor space* and the *local stationarity of user intent*.

Structural alignment of the factor space. Traditional soft prompts inject unstructured continuous vectors where specific dimensions have no fixed semantics across different inputs. In FaLCon, the offline multi-branch autoencoder (§2.2) forces the representation into K distinct, non-redundant factors. Crucially, the item autoencoder is shared globally across all items. This means that the k-th factor encodes a consistent latent concept (e.g., aesthetic design, usability, price-to-performance ra-

tio) regardless of the specific item being processed. Because different items' embeddings are routed through the identical branch structure, the k-th factor of item A and the k-th factor of item B reside in the same semantic subspace.

Local stationarity of user intent. Because we decoupled target-specific semantics from the user history (Eq. 3), the user factors represent abstract, stable behavioral drivers rather than item-specific keywords. While a user's overarching preferences are relatively stable, their immediate priorities (e.g., currently caring more about "build quality" over "shipping speed") exhibit temporal continuity, or local stationarity, across recent interactions.

By adjusting the factor weights w on recent adaptation records, the Thompson sampling controller is not memorizing past items; rather, it is calibrating the user's *current latent state*—identifying which abstract semantic factors the user is currently emphasizing. Because the factor semantics are universally aligned across all items, a weight w_k that correctly modulates the user's reliance on the k-th attribute during recent interactions will successfully modulate that same semantic attribute when processing the held-out test item $i^\star$. Therefore, the continuous adaptation of w serves as a dynamic routing mechanism that generalizes robustly to new targets.

D.3 Why Does Test-Time Alignment (e.g., T-POP) Underperform on Target-Aware Generation?

Our empirical results (Table 1) show that while the test-time alignment method T-POP improves over non-personalized baselines, its performance gains are relatively modest compared to FaLCon. This performance gap stems from a fundamental mismatch between the assumptions of standard test-time alignment and the unique challenges of target-aware personalized generation. We analyze this mismatch from two perspectives:

1. Semantic Entanglement and Target Drift. The core challenge in our setting is cross-item personalization: the adaptation history consists of reviews for past items $(i_1, \ldots, i_T)$, but the model must generate a review for a novel target item $i^\star$. Methods like T-POP update a monolithic reward model directly based on the token-level preferences observed in the history. However, in these historical records, the user's abstract preferences (e.g., preference for concise language or focus on build quality) are inextricably entangled with the specific semantic features of those past items (e.g., plot details of a specific movie).

By continuously aligning the model to maximize the reward derived from past items, T-POP inadvertently overfits to the historical item semantics. When deployed on the new target $i^\star$, the reward model exhibits *target drift*—it improperly encourages the generation of surface tokens and specific entities associated with the adaptation set, overshadowing the actual attributes of $i^\star$. FaLCon entirely bypasses this issue through its factorized offline interface (Eq. 3), which structurally decouples the user's abstract intent from the item's concrete semantics. FaLCon's online controller only updates the low-dimensional factor weights w, modulating universally aligned semantic factors without memorizing the token-level artifacts of past items.

2. Alignment vs. Dense Content Generation. T-POP is fundamentally designed for alignment—steering the global distribution of a model towards desirable behaviors (e.g., harmlessness, helpfulness, or general conversational style). Alignment typically operates at a macro level, penalizing or rewarding broad trajectories. In contrast, personalized review generation is a dense, information-rich content creation task that requires precise grounding in the target item's attributes (e.g., actors, plot, or specific product flaws).

Test-time alignment methods struggle to inject or synthesize this dense, target-specific factual knowledge. While T-POP's token-by-token reward-guided decoding (via dueling bandits) excels at shifting the stylistic tone, it lacks a structured mechanism to explicitly fuse the user's historical context with the novel target item's facts. Consequently, it often produces generic or hallucinated content that aligns stylistically but fails to accurately reflect the specific target item. FaLCon's approach—using explicit, information-dense soft prompts constrained by a redundancy-reduced factor space—ensures that the generation remains strictly grounded in the target item's latent representation while dynamically emphasizing the user's preferred semantic axes.

D.4 Teacher-Forced Reward vs. Free Generation Quality

A critical design choice in FaLCon's online adaptation is the use of teacher-forced log-likelihood (Eq. 8) as the reward signal for the bandit controller.

Reviewers might rightfully question whether optimizing step-by-step likelihood translates to holistic, sequence-level free generation quality, given the well-known issue of *exposure bias* in natural language generation.

To mitigate the gap between local likelihood and global semantic coherence, we introduce the group size parameter U (§2.3), which determines how many consecutive tokens share a single bandit update. This parameter governs a fundamental trade-off between *update frequency* and *semantic globality*:

- **High frequency** ($U = 1$): The controller updates its posterior at every single token. This maximizes the number of exploration steps (Thompson samples) per interaction, allowing the bandit to rapidly converge on an optimal posterior. However, the reward signal is myopic, relying heavily on token-level syntactic likelihood.
- **Semantic globality** ($U > 1$): The controller averages the reward over a chunk of tokens (e.g., $U = 8$ or 16) before performing a Bayesian update. This explicitly optimizes for chunk-level or phrase-level semantic coherence, bridging the gap to free generation metrics, but significantly reduces the number of exploration steps available within a single text record.

Metric	$\mathbf{U = 1}$	$\mathbf{U = 4}$	$\mathbf{U = 8}$	$\mathbf{U = 16}$
ROUGE-1	**32.10**	25.56	26.32	30.72
ROUGE-L	17.58	14.41	15.37	**18.47**
METEOR	17.21	15.13	17.36	**17.78**

Table 5: Impact of the update group size U on final generation quality. Evaluated on a subset of 100 hard instances (where baseline performance was poor) from the Books domain using greedy decoding (temperature $=$ 0). All metrics are multiplied by 100.

To empirically investigate this trade-off, we conducted a targeted ablation study varying $U \in \{1, 4, 8, 16\}$. To rigorously test the controller's ability to rescue difficult generations, we evaluated it not on a random split, but on a curated subset of 100 hard instances from the Books domain where the unadapted baseline performed particularly poorly. As shown in Table 5, the results reveal a fascinating U-shaped dynamic. Both ends of the spectrum prove to be highly competitive. $U = 16$ achieves the highest ROUGE-L and METEOR scores, confirming that grouping tokens effectively captures broader structural and semantic intent, which directly benefits sequence-level fluency and alignment. Conversely, $U = 1$ yields the highest ROUGE-1 score, demonstrating that the sheer volume of high-frequency exploration steps allows the controller to aggressively fit lexical preferences.

While tuning U reveals nuanced trade-offs, we observe that the high-frequency extreme ($U = 1$) provides a highly robust and computationally simple default. The average token-level exploration empirically aggregates into sequence-level generation improvements without requiring complex chunking logic. Nevertheless, this points to a promising avenue for future work: rather than using a fixed token stride, the adaptation could be triggered dynamically based on semantic boundaries (e.g., sentence-level updates) or limited to high-information tokens (e.g., updating only on nouns and verbs). For the scope of this paper, we default to $U = 1$ to maintain algorithmic simplicity while ensuring strong empirical performance.

E Additional Experimental Results

E.1 Analysis of Factor Decoupling and Subspace Learning

A fundamental prerequisite for FaLCon's inference-time latent control is that the action space—the candidate continuous weight vector $w \in [w_{\min}, w_{\max}]^{2K}$—must be well-conditioned. If the underlying prompt factors are highly correlated or if the information concentrates in only a few factors, adjusting the weight w_k of a specific factor during online adaptation would fail to isolate a distinct semantic attribute. To validate that our redundancy-reduced multi-branch encoders (§2.2) successfully construct a separately addressable and balanced interface, we visualize the internal statistics of the learned factors.

Cross-branch decorrelation. Figure 6(a-b) presents the cross-branch correlation matrices for the learned user and item factors. The off-diagonal elements are nearly zero, demonstrating that the multi-branch representations are highly decorrelated. This is a direct consequence of the decorrelation term $\mathcal{L}_{\text{dec}}$, which jointly penalizes cross-branch batch covariance and head-weight alignment. From a control perspective, this decorrelation guarantees that the prompt positions

do not collapse into redundant copies: when the controller updates the posterior for factor k, it intervenes on a specific semantic direction without changing the remaining context, providing clear, disentangled evidence for the Bayesian updates.

Factor balance and information distribution. Figure 6(c-d) illustrates the information distribution across the $K = 8$ factors, measured by the Frobenius norm of each factor's output. Without explicit regularization, multi-branch encoders are prone to collapsed factors, where all information concentrates in a few dominant dimensions. Driven by the balance penalty $\mathcal{L}_{\text{bal}}$, our factors exhibit an almost perfectly uniform distribution of representation norms. This ensures that every dimension of the controller's action space carries meaningful semantic capacity, rather than leaving some slots as degenerate factors.

Training dynamics and variance preservation. Figure 7(a-b) tracks the optimization dynamics of the weighted loss components ($\lambda \times \mathcal{L}$). Imposing strict decorrelation constraints from random initialization can severely disrupt representation learning. The loss curves highlight the necessity of the 5-epoch warmup phase: the model first focuses exclusively on the reconstruction term $\mathcal{L}_{\text{rec}}$ to capture basic semantic structures. Once the warmup concludes, the structural regularizers ($\mathcal{L}_{\text{dec}}$, $\mathcal{L}_{\text{var}}$, $\mathcal{L}_{\text{bal}}$) are introduced, and all weighted components smoothly converge to a stable equilibrium. Finally, the factor statistics in Figure 7(c-d) confirm that the standard deviation within each factor remains healthy and consistent, proving that the variance floor $\mathcal{L}_{\text{var}}$ effectively prevents degenerate factors at the individual dimension level. Together, these offline structural guarantees empower the online controller to operate over a robust, interpretable, and non-redundant factor interface.

E.2 Dynamics of Inference-Time Latent Control

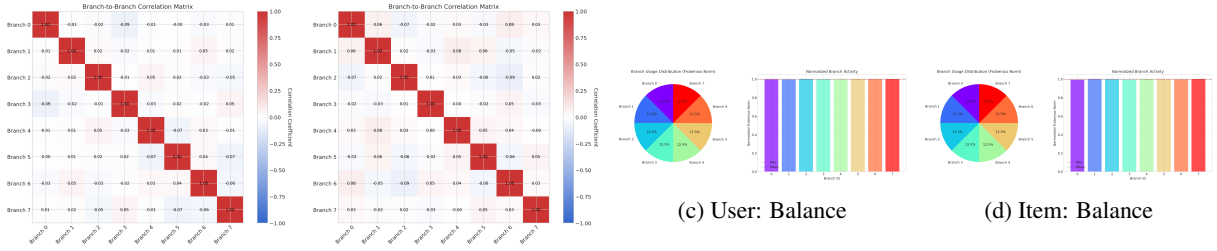
To better understand how the Thompson sampling controller navigates the action space during online adaptation, we visualize the internal posterior state and the resulting exploitation weights for a representative single instance. Because FaLCon models the reward function in the GP-RFF feature space ($D = 128$), we project the final posterior state back to the $2K = 16$ dimensional factor space (8 user factors and 8 target factors) via the Jacobian of the RFF feature map evaluated at the unadapted baseline $w_0 = \mathbf{1}$.

Estimated utility gradient. Figure 8(a) shows the effective posterior mean μ across the 16 factor dimensions. In the context of the unadapted baseline $w_0 = \mathbf{1}$, this projected μ serves as the estimated gradient of the reward function. A positive value indicates that amplifying the corresponding factor (increasing $w_k > 1$) improves the teacher-forced generation likelihood, whereas a negative value suggests the factor is detrimental or irrelevant for the current context and should be suppressed. As shown, the controller successfully identifies a sparse subset of highly relevant factors (e.g., specific user or target factors with large absolute μ) rather than shifting all weights uniformly.

Structured posterior covariance. Figure 8(b) visualizes the effective 16×16 posterior covariance matrix Σ. The diagonal elements represent the controller's remaining uncertainty about each factor's utility. The off-diagonal block structure (partitioned into user-user, user-target, and target-target quadrants) reveals how the GP-RFF kernel captures correlations between different factors during adaptation. For instance, non-zero covariances between certain user and target factors indicate that the controller learns joint dependencies from the streaming feedback, allowing evidence from one dimension to inform the exploration of another.

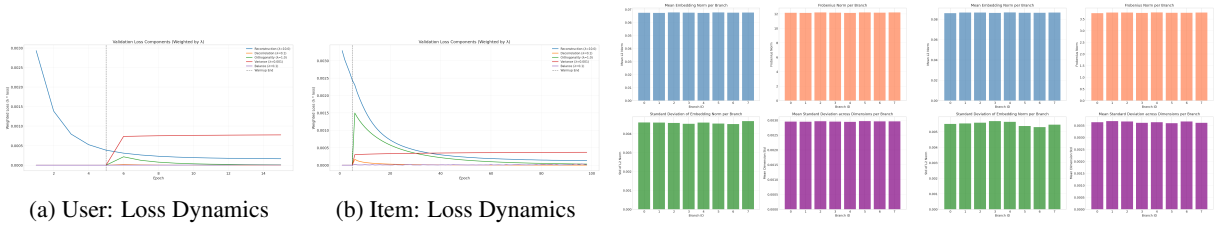
Exploitation weight mapping. Figure 8(c) plots the final exploitation weight $\hat{w}$ (computed via Eq. 13) against the estimated posterior mean μ. The strong positive correlation confirms that the projected gradient ascent correctly translates the Bayesian evidence into the bounded action space $[w_{\min}, w_{\max}]$. Dimensions with strong positive evidence (large μ) are pushed toward the upper bound, while those with negative evidence are suppressed. Importantly, the L2 regularizer λ_{reg} ensures that factors with near-zero μ (insufficient evidence) gracefully default to the trained operating point $w_k \approx 1.0$, preventing reckless deviation from the offline representation.

Qualitative impact of factor adaptation. To contextualize the quantitative gains achieved by online adaptation, Table 6 presents a representative example of the generated text before and after the Thompson sampling (TS) optimization. The unadapted baseline ($w = \mathbf{1}$) produces a generic



(a) User: Correlation (b) Item: Correlation (c) User: Balance (d) Item: Balance

Figure 6: Analysis of factor independence and information balance. (a-b) Cross-branch correlation matrices show near-zero off-diagonal elements, confirming that the learned factors are highly decorrelated. (c-d) Factor balance plots demonstrate that information is evenly distributed across all $K = 8$ factors, preventing redundant or collapsed factors and ensuring a fully utilized action space.



(a) User: Loss Dynamics (b) Item: Loss Dynamics (c) User: Branch Stats (d) Item: Branch Stats

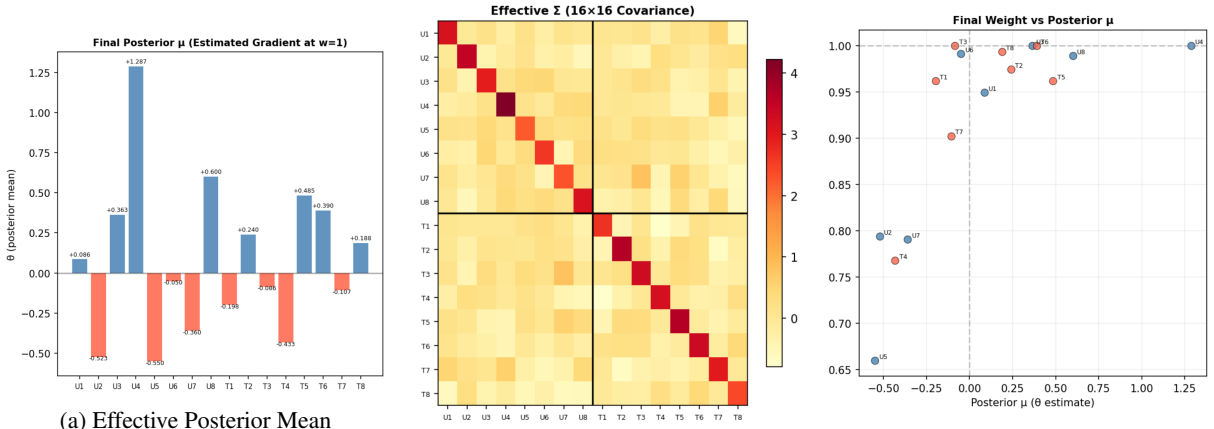
Figure 7: Training dynamics and factor statistics. (a-b) Weighted composite loss components ($\mathcal{L}_{\text{rec}}$, $\mathcal{L}_{\text{dec}}$, $\mathcal{L}_{\text{var}}$, $\mathcal{L}_{\text{bal}}$) over training epochs. A 5-epoch warmup phase precedes the activation of the structural regularizers, allowing for stable convergence. (c-d) Detailed factor statistics confirm consistent mean L2 norms and dimension-wise standard deviations across all sub-embeddings, verifying the efficacy of the variance floor ($\mathcal{L}_{\text{var}}$).

and repetitive review. While it correctly identifies the "Laurel Snow" series, it falls into a degenerative text loop, repeating the exact phrase *"I have read all of the [books] and I am looking forward to the next one"* at both the beginning and the end of the text. In contrast, after the latent controller updates the posterior using the recent user interactions and applies the optimal exploitation weights $\hat{w}$, the generated review transforms significantly. The TS-optimized text is not only substantially longer and free of repetition, but it also activates highly specific semantic details from the target item subspace—explicitly mentioning the author, the specific book title (*"You Can Hide"*), the setting (*"Widows Island"*), and the core plot (*"missing person"*). By successfully re-weighting the latent factors, FaLCon suppresses the generic factors that lead to repetition and amplifies the informative factors that ground the generation in specific user preferences and item characteristics, yielding a $+0.0754$ absolute improvement in METEOR score.

E.3 Sensitivity Analysis of Continuous Factor Weights

To empirically validate the continuous nature of the factor-level interface and justify our choice of action space bounds, we perform a fine-grained sensitivity analysis on a representative test instance. Figure 9 visualizes the impact of independently modulating the weight w_k of each factor (while keeping all other factors at the unadapted baseline $w = \mathbf{1}$) on the final generation quality, measured by the METEOR score.

Non-linear factor response and optimal intervals. As observed in the regime where $w_k \leq 1.0$, modulating the weight of specific factors reveals distinct optimal intervals and step-like transitions in generation quality. This empirically validates a core hypothesis of FaLCon: the redundancy-reduced multi-branch encoders have successfully structured the soft-prompt space into controllable semantic directions. Adjusting w_k acts as a "volume dial" for specific latent concepts, allowing the controller to tune each factor to its appropriate intensity. Furthermore, the presence of these localized optimal regions and step-like transitions within a highly non-linear reward landscape jus-



(a) Effective Posterior Mean

(b) Effective Posterior Covariance

(c) Exploitation Weight vs. Mean

Figure 8: Visualization of the controller's posterior state and exploitation weights for a representative instance after online adaptation. (a) The effective posterior mean μ evaluated at $w_0 = \mathbf{1}$, acting as the estimated utility gradient for each factor. (b) The effective posterior covariance Σ, displaying the remaining uncertainty and cross-factor correlations, partitioned into user (U) and target (T) blocks. (c) The monotonic mapping from the posterior evidence μ to the final exploitation weight $\hat{w}$ via projected gradient ascent.

tifies our choice of modeling the reward function with a Gaussian Process using an RBF kernel (via RFF approximation) rather than relying on strictly linear optimization, enabling the controller to effectively pinpoint the optimal factor configuration.

Justification for the suppression-only bound ($w_{\max} = 1.0$). A prominent pattern across almost all 16 factors is the sharp deterioration in generation quality when the weight exceeds the baseline operating point ($w_k > 1.0$). This phenomenon directly informs our decision to bound the Thompson sampling action space to $[w_{\min}, w_{\max}] = [0, 1]$. Because the backbone LLM was fine-tuned with unscaled factor embeddings (equivalent to $w = \mathbf{1}$), setting $w_k > 1.0$ at inference time forces the layer-0 Key/Value attention activations into an out-of-distribution (OOD) regime. This artificial amplification disrupts the attention mechanism, causing the LLM to over-attend to the prompt tokens and lose semantic coherence. Therefore, bounding $w \leq 1.0$ safely restricts the controller to a *suppression-only* paradigm: the bandit can filter out irrelevant factors by turning down their volume, but it cannot amplify them beyond the manifold observed during training, ensuring robust generation.

F Overview of Templates and Prompts

FaLCon uses a carefully structured prompt template to inject both discrete text context and continuous soft-prompt factors into the instruction-tuned LLM (Qwen2.5-7B-Instruct) (Yang et al., 2024). The prompt format strictly adheres to the Qwen ChatML standard (using <|im_start|> and <|im_end|>) and is divided into several logical sections.

Prompt construction. The prompt consists of the following sequential components:

- **System Prompt**: A fixed system instruction that defines the task (generating a personalized item review based on the target item and user history) and explicitly informs the LLM about the presence of soft-prompt embeddings.
- **Global Subspace (Factor Interface)**: At the beginning of the user message, we inject the continuous factor representations. This includes 8 user subspace factors (encoding user preference patterns) and 8 target item subspace factors (encoding the target item's characteristics). Each factor is represented as a 512-dimensional embedding enclosed by specific structural tokens (e.g., <user_subspace_start>). **Crucially, the inference-time latent controller (§2.3) only modulates the weights w_k of these 16 global subspace factors** located at the beginning of the prompt.
- **Target Item Context**: The discrete text features (Title and Description) of the target item.
- **Historical Interactions**: The user's past interaction records (up to 8 reviews). For each historical record, the prompt provides the item's

Configuration	METEOR	Generated Review
Unadapted Baseline ($w = \mathbf{1}$)	0.0918	I love the Laurel Snow series. I have read all of the books and I am looking forward to the next one. The characters are well developed and the story lines are always interesting. I love the way the author uses the small town setting to develop the characters and the story. *I have read all of the Laurel Snow books and I am looking forward to the next one.*
FaLCon (TS Optimized $\hat{w}$)	**0.1672** ($+\mathbf{0.0754}$)	I have been a fan of Rebecca Zettl's books for a while now, and I was not disappointed with "You Can Hide". The story is set in the small town of Widows Island, and it follows the protagonist, Laurel Snow, as she tries to solve the mystery of a missing person. The plot is well-crafted, with plenty of twists and turns that kept me on the edge of my seat. The characters are well-developed, and I found myself rooting for them throughout the story. The writing style is engaging and easy to read, making it a quick and enjoyable read. Overall, I highly recommend this book to anyone who enjoys a good thriller with well-crafted characters and a well-paced plot.

Table 6: Qualitative comparison of generation quality before and after inference-time latent control for a representative test instance. The unadapted baseline suffers from generic statements and text repetition (highlighted in gray). Online adaptation via Thompson Sampling correctly re-weights the factor interface, eliciting a much more detailed, specific, and structurally coherent review.

text features, the item's continuous subspace factors (to represent historical target semantics), and the actual review rating, title, and text. The attention weights for these historical item embeddings remain fixed at 1.0 and are *not* modulated during online adaptation.

- **Output Constraints**: The desired rating and title for the target review, prompting the LLM to generate the final review text.

Context window truncation. To ensure the input strictly fits within the maximum context window (3,072 tokens), we implement a dynamic truncation strategy. If the total token length exceeds the limit, the prompt builder drops the oldest historical reviews one by one (from 8 down to 1) until the context fits. If a user or item lacks a corresponding factor embedding in the map, a zero-vector is used for padding.

Full prompt template. Table 7 presents the exact structure of the prompt passed to the LLM. It clearly delineates the discrete text tokens and the locations where the continuous factor embeddings (`[USER_SUBSPACE_k]`, `[TARGET_ITEM_SUBSPACE_k]`, and `[HIST_ITEM_SUBSPACE_k]`) are injected into the context.

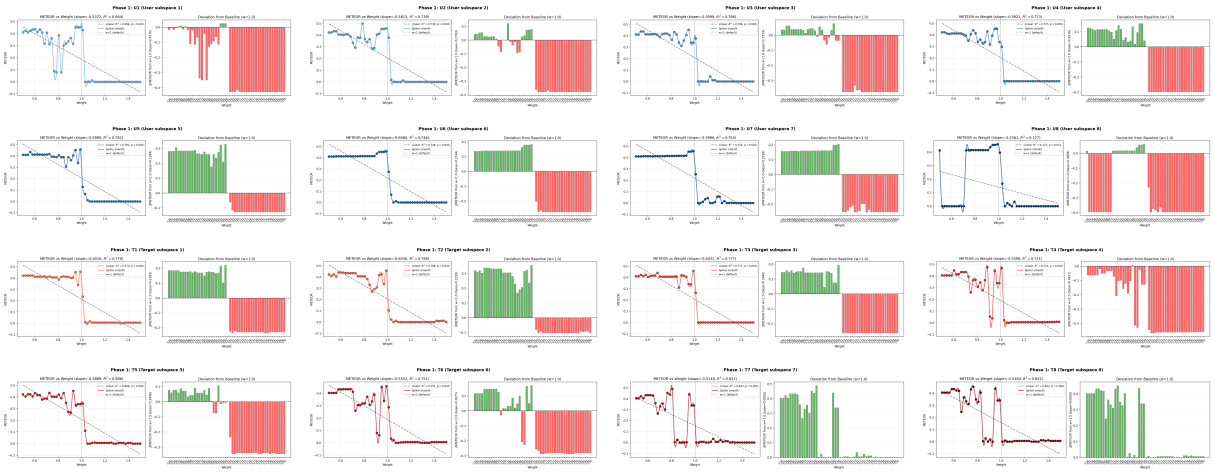


Figure 9: Sensitivity analysis of the continuous factor weights. The top two rows (U1–U8) correspond to the user factors, and the bottom two rows (T1–T8) correspond to the target item factors. The x-axis represents the continuous weight $w_k \in [0.5, 1.5]$, and the y-axis is the resulting METEOR score (computed using greedy decoding with temperature $\tau = 0$ to eliminate sampling randomness). Modulating $w_k \leq 1.0$ reveals non-linear transitions and distinct optimal intervals for different factors. Conversely, aggressively amplifying weights ($w_k > 1.0$) frequently leads to catastrophic performance drops due to out-of-distribution attention scaling.

Exact Input Prompt Structure for FaLCon

<|im_start|>system
Given the title and description of an item, along with the user's past reviews (including item title, item description, review rating, review title, review text, review embedding, review difference embedding), and the output review rating and review title, generate a personalized item review for the user.
Note: [Review Embedding] denotes a soft prompt of the review text and [Review Difference Embedding] denotes a soft prompt showing the difference between the review text and other reviews on the same item. [Review Embedding] and [Review Difference Embedding] should serve as hints for personalized review text generation.
<lim_end|>
<lim_start|>user
[User Subspace 1]: <user_subspace_start>[USER_SUBSPACE_0]<user_subspace_end>
...
[User Subspace 8]: <user_subspace_start>[USER_SUBSPACE_7]<user_subspace_end>
[Target Item Subspace 1]: <target_item_subspace_start>[TARGET_ITEM_SUBSPACE_0]<target_item_subspace_end>
...
[Target Item Subspace 8]: <target_item_subspace_start>[TARGET_ITEM_SUBSPACE_7]<target_item_subspace_end>
[Item Title]: The Great Gatsby
[Item Description]: A masterpiece of American fiction, The Great Gatsby is a classic portrait of the Jazz Age...
[User's Past Reviews]:
- [Review 1]:
  - [Item Title]: To Kill a Mockingbird
  - [Item Description]: The unforgettable novel of a childhood in a sleepy Southern town...
  - [Item Subspace]: <hist_item_subspace_start>[HIST_ITEM_SUBSPACE_0_0]<hist_item_subspace_end>
    ... <hist_item_subspace_start>[HIST_ITEM_SUBSPACE_0_7]<hist_item_subspace_end>
  - [Review Rating]: 5
  - [Review Title]: A timeless classic
  - [Review Text]: I absolutely loved this book. The characters are so well developed...
- [Review 2]:
  ...
- [Review 8]:
  - [Item Title]: 1984
  - [Item Description]: Among the seminal texts of the 20th century...
  - [Item Subspace]: <hist_item_subspace_start>[HIST_ITEM_SUBSPACE_7_0]<hist_item_subspace_end>
    ... <hist_item_subspace_start>[HIST_ITEM_SUBSPACE_7_7]<hist_item_subspace_end>
  - [Review Rating]: 4
  - [Review Title]: Chilling and prophetic
  - [Review Text]: Orwell's vision of a totalitarian future remains as relevant as ever...
[Output Review Rating]: 4
[Output Review Title]: A beautiful portrayal of the American Dream
<lim_end|>

Table 7: The complete structured prompt template used in FaLCon. The prompt follows the Qwen ChatML format. Placeholders in all-caps brackets (e.g., [USER_SUBSPACE_0]) denote positions where continuous, 512-dimensional sub-embeddings are injected as soft-prompt factors. The system prompt explicitly informs the LLM to utilize these embeddings as hints for personalization. Only the top 16 global sub-embeddings (USER_SUBSPACE and TARGET_ITEM_SUBSPACE) act as the continuous action space for inference-time latent control; the historical item sub-embeddings remain unmodulated.