Recovering Evolving User Preference State via Adaptive Interaction-aware Representation Correction

Parthiv Chatterjee[1] **Dhiraj Golhar**[2] **Ummesalma Diwan**[1]
Sourish Dasgupta[1] **Manjunath Joshi**[2] **Tanmoy Chakraborty**[3]
[1]KDM Lab, Dhirubhai Ambani University [2] Dhirubhai Ambani University
[3] Indian Institute of Technology Delhi
{202421011,202121010,202518017,sourish_dasgupta,mv_joshi}@dau.ac.in
tanchak@iitd.ac.in

Abstract

Personalization systems encode long-horizon, evolving user trajectories into compressed preference states that downstream task heads consume for prediction or generation. This compression is difficult because preference trajectories contain stable long-term interests, transient short-term shifts, and episodic bursty interests. We argue that a practical route is to correct the under-expressed compressed state before the downstream task head consumes it, rather than design another task-specific encoder or incur costly host-encoder finetuning. We propose REPAIR as a selective *pre-head state-repair method* for frozen personalization encoders. It plugs into frozen hosts, uses timestep-level event representations from the same trajectory, and avoids a second history-encoding pass. REPAIR compares the compressed state with per-timestep event representations, estimates state-relative corrective evidence, resolves it through long-term, short-term, and episodic temporal regimes, and retains only the most corrective signals to form a compact correction. We evaluate REPAIR across recommendation and generation personalization settings using MovieLens, MIND, PENS, and Amazon Reviews 2023. Across recommendation tasks, full REPAIR improves the evaluated frozen hosts. On MovieLens, it improves the highest-scoring frozen host in our instantiated baseline suite by $+1.27/+1.54$ MRR/nDCG@5; on MIND, it improves the highest-scoring frozen host by $+4.41$ MRR; and on Amazon product recommendation, it improves two SOTA baselines by $+15.33$ and $+17.91$ MRR. Additionally, with both the task-trained encoder and original task head frozen, training only REPAIR still improves the evaluated recommendation hosts, while task head fine-tuning yields smaller gains. For example, Mamba4Rec improves by $+3.96$ MRR with frozen-head REPAIR, compared with $+0.19$ from head-only fine-tuning. REPAIR also outperforms protocol-matched post-head correctors, and the full long-, short-, and episodic repair remains the strongest temporal variant. On PENS personalized headline generation, REPAIR improves subjectivity-sensitive generation most when the decoder is preference-coupled, with average PerSEval gains of +14.62% and +22.19% for two representative preference-coupled decoders. Relative to frozen predictive encoders, deployment overhead is 31–34% memory and 38–44% latency.

1 Introduction

Personalization aims to predict or generate user-expected content, such as the next item, summary, headline, or response from evolving user preference histories. These histories contain heterogeneous intent signals, including positive signals such as *click* and *command*, and negative signals such as

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

skip. Personalization systems usually encode such long-horizon, time-ordered trajectories into one or a small set of compressed preference states consumed by a downstream prediction or generation head. Existing encoders differ in architectural bias, including recurrent, self-attentive, state-space, time-aware, and multi-interest modeling [Hidasi et al., 2016, Wu et al., 2019b, Liu et al., 2024, Li et al., 2020, Cen et al., 2020]. However, they share the same downstream interface – the task head treats the exposed compressed state as sufficient for the current decision.

Ideally, the compressed state should retain all task-relevant temporal traces. In practice, stable long-term interests, recency-sensitive short-term shifts, and episodic bursts play distinct temporal roles but are folded into the same representation. Useful preference evidence can, therefore, remain missing, weakly reflected, or improperly expressed in the state consumed by the task head. This can misalign the exposed preference state with the user's evolving preference trajectory when earlier or newer preference patterns are improperly expressed. This issue remains relevant even for self-attentive and Transformer-style encoders: they compare timesteps while forming the state, but do not inspect the already-exposed state to determine which historical evidence remains under-expressed relative to it. This motivates treating cached historical timestep representations as candidate sources of corrective evidence, and correcting the under-expressed compressed state before the downstream task head consumes it, rather than designing another task-specific encoder. Although neighboring directions motivate this view, they do not solve the same problem. Representation-space intervention methods guide frozen internal states through hidden-state interventions or user-specific activation directions [Wu et al., 2024, Zhang et al., 2025]. Memory-based editing corrects frozen-model behavior through retrieved edits and conditional routing [Mitchell et al., 2022b]. Denoising and reconstruction-based recommenders suppress noisy links or target-irrelevant interactions before prediction [Chen et al., 2022, Yang et al., 2023b, Xie et al., 2022, Zhang et al., 2024a]. However, these methods do not repair the compressed preference state of a frozen trajectory encoder, nor do they ask which historical timesteps remain corrective with respect to what that state already expresses or fails to express.

We propose `REPAIR` (**R**ecovering **E**volving User **P**reference State via **A**daptive **I**nteraction-aware **R**epresentation Correction), as a post-hoc method for repairing compressed preference states of frozen personalization encoders. `REPAIR` corrects the compressed state using the same trajectory that produced it, rather than discarding it and rebuilding user preference from scratch. It compares the exposed state with cached per-timestep event representations, estimates state-relative corrective evidence, and resolves this evidence through long-term, short-term, and bursty episodic regimes. It retains only the strongest timestep-level corrective signals, and fuses the resulting correction with the compressed state before the downstream task head consumes it. Hence, `REPAIR` is a selective *pre-head* state-repair framework for frozen, deployment-constrained personalization systems.

We study four research questions. **RQ-1**: Does repairing the compressed preference state improve downstream personalization across frozen encoders? **RQ-2**: Is pre-head state repair more effective w.r.t. accuracy than budget-matched post-head corrective alternatives under the same frozen-host setting? **RQ-3**: Does selective repair recover preference signal in a temporally structured way by distinguishing long-term preference, short-term shifts, and episodic bursts, while retaining the timesteps that provide the strongest corrective evidence? **RQ-4**: Do the gains of `REPAIR` remain worthwhile once memory and latency overhead are made explicit? We evaluate these questions on MovieLens, MIND, Amazon Reviews 2023, structured PENS recommendation, and personalized headline generation on PENS. Across predictive tasks, full `REPAIR` consistently improves our instantiated SOTA frozen hosts, including $+4.97$ MRR on `Mamba4Rec`, $+4.41$ MRR on `LSTUR`, and $+15.33/+17.91$ MRR on `SigLIP2/OpenCLIP`. With both the task-trained encoder and original task head frozen, training only `REPAIR` improves all twelve representative recommendation hosts on MRR and nDCG@10, while head-only fine-tuning yields smaller gains; for example, `Mamba4Rec` gains $+3.96$ MRR with frozen-head `REPAIR`, compared with $+0.19$ from head-only fine-tuning. `REPAIR` also outperforms instantiated post-head correctors `SERAC-Corr` and `Rec-Denoiser-Corr`. Across repair dimensions $K \in \{32, 64, 128, 192\}$, $K = 128$ gives the strongest MRR and nDCG@10 in all five tested settings, while $K = 192$ lowers both metrics, showing that increasing repair capacity alone does not explain the gains. The full long-, short-, and episodic `REPAIR` variant further exceeds the best partial temporal variant by an average of $+2.65$ MRR across representative hosts. On personalized summarization, `REPAIR` reveals a decoder-utilization boundary, with `Walk2Pers` and `IMPerSumm` improving by average gains of $+14.62\%$ and $+22.19\%$ across PSE variants. Finally, `REPAIR` incurs 31–34% memory and 38–44% latency overhead relative to the main frozen predictive hosts. Together, these results show that compressed preference states remain recoverable from the trajectory itself and support selective

pre-head repair as a practical alternative to full re-encoding or generic post-head correction. Code and reproducibility package: `https://github.com/KDM-LAB/REPAIR-NeurIPS-2026.git`.

2 Related Work

Preference Encoders and Temporal User Modeling. Personalization models usually convert a user-history prefix into one or more downstream-facing preference states. This interface appears across recurrent and session-based recommendation, attentive news recommendation, self-attentive and bidirectional sequential recommendation, time-aware recommendation, state-space recommendation, and large-scale sequential transduction [Hidasi et al., 2016, Okura et al., 2017, Wu et al., 2019a,b, Kang and McAuley, 2018, Sun et al., 2019, Li et al., 2020, Liu et al., 2024, Zhai et al., 2024]. Related work also recognizes that preference is not single-scale: long/short-term models separate persistent identity from recent intent [An et al., 2019, Yu et al., 2019], multi-interest methods represent several latent interests [Cen et al., 2020, Li et al., 2022], and frequency-sensitive or state-space methods model periodic, long-horizon, or selective propagation patterns [Du et al., 2023, Liu et al., 2024, Xiao et al., 2025]. These works motivate our premise that user histories contain heterogeneous temporal traces. However, their primary stress remains the encoder that forms the preference state. Even when an encoder uses attention, recurrence, state-space propagation, or multiple interest slots, `REPAIR` addresses a different question: *whether the surviving compressed state can still be repaired from its own cached per-timestep evidence*.

Personalized Generation and Decoder-side Use of User States. Personalized generation methods condition decoding on user representations derived from history, profiles, retrieval, or prompt-side context. PENS, GTP, SCAPE, and FPG instantiate this idea for personalized headline generation through user-history encoders, decoupled personalization, style-content modeling, and factuality-preserving generation [Ao et al., 2021, Song et al., 2023, Lian et al., 2025, Yang et al., 2023a]. Personalized dialogue and LLM-based systems similarly use persona, memory, retrieval, or prompt construction to condition generation [Zhang et al., 2018, Liu et al., 2023, Lu et al., 2023, Ahn et al., 2023, Zhang et al., 2024b, Patel et al., 2024, Lyu et al., 2024, Wang et al., 2024]. These works show that user-conditioned generation depends not only on the user representation, but also on how the downstream interface uses that representation. We therefore examine how repaired preference evidence is reflected in downstream generation, separating personalization-sensitive gains from changes in generic text quality.

Representation Intervention, Model Editing, and Denoising. Another line adapts frozen models without full backbone fine-tuning. Adapter methods, Prefix-Tuning, LoRA, ReFT, and StyleVector steer frozen representations through compact trainable modules or hidden-state interventions [Houlsby et al., 2019, Li and Liang, 2021, Hu et al., 2022, Wu et al., 2024, Zhang et al., 2025]. In particular, ReFT and its low-rank LoReFT instance learn task-specific interventions on selected hidden representations of a frozen base model [Wu et al., 2024]. Recent recommendation-specific interventions include SPREE, which performs inference-time activation steering toward user-specific popularity preferences [Schirmer et al., 2026], and RAIE, which performs region-aware incremental preference editing through region-specific LoRA modules [Zeng et al., 2026]. Model editing methods such as MEND, SERAC, ROME, and MEMIT correct frozen-model behavior through learned editors, memory routing, or localized weight edits [Mitchell et al., 2022a,b, Meng et al., 2022, 2023]. Sequential recommendation also contains denoising methods that suppress noisy attention links, inconsistent behaviors, or target-irrelevant interactions [Chen et al., 2022, Zhang et al., 2022, Xie et al., 2022, Zhang et al., 2024a, Li et al., 2023, Yang et al., 2023b]. Although these methods establish related intervention principles, their correction targets and evidence sources differ from `REPAIR`. `REPAIR` uses cached per-timestep representations from the completed host computation to correct an already-formed downstream-facing preference state, with the host encoder fixed and without re-encoding the raw history.

3 Evolving Preference Personalization Under State Compression

For a user u, we model each user history as a time-ordered interaction trajectory. Prior analyses of real-world personalization datasets show that user trajectories are *evolving* with fine-grained action-conditioned signals and frequent topic transitions [Ao et al., 2021, Ahn et al., 2023, Chatterjee et al., 2025]. Following the *User Interaction Graph* (UIG) abstraction of Chatterjee et al. [2025],

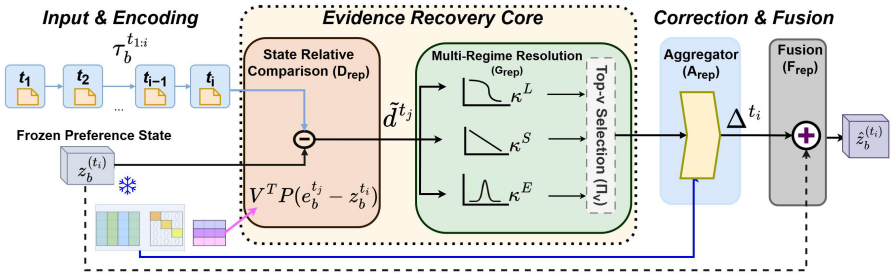

Figure 1: **Overview of** REPAIR**.** Given a frozen preference state $z_b^{(t_i)}$ and cached event representations, REPAIR computes state-relative corrective evidence, resolves it through long-, short-, and episodic-support branches, retains the strongest timestep-level signals, and fuses the resulting correction Δ_b with $z_b^{(t_i)}$ to produce repaired state $\tilde{z}_b^{(t_i)}$ before the downstream head consumes it.

each observed interaction is represented as an *interaction unit* $b^{(t_i)} = \langle a^{(t_i)}, v^{(t_i)} \rangle$, where $a^{(t_i)} \in \{\texttt{pos}, \texttt{neg}, \texttt{cmd}\}$ denotes the action type and $v^{(t_i)}$ denotes the content, query, response, or target node on which the action occurs. For a *prefix* ending at timestep t_i, the event sequence is $\tau_b^{t_{1:i}} = \langle b^{(t_1)}, b^{(t_2)}, \ldots, b^{(t_i)} \rangle$. For predictive tasks, the model predicts the next target $r_u^{\star(t_{i+1})}$ from this prefix. For generative tasks, the model produces the target response given the prefix and the task-side query $q^{(t_i)}$. Appendix A.1 presents the details of UIG construction.

A personalization encoder f_{enc} represents an interaction unit $b^{(t_j)}$ as event $e_b^{(t_j)}$ and compresses the prefix into an exposed preference state $z_b^{(t_i)} = f_{\text{enc}}\left(e_b^{(t_1)}, e_b^{(t_2)}, \ldots, e_b^{(t_i)}\right)$. The downstream predictor or generator head consumes $z_b^{(t_i)}$ to produce the personalized output. The per-timestep representations $\{e_b^{(t_j)}\}_{j=1}^{i}$ remain available from the same host computation and form the cached evidence used by REPAIR, without re-encoding the raw history. By this stage, positive (pos; eg.: *click*), negative (neg; eg.: *skip*), and command-conditioned interactions (cmd; eg.: *summarize*) have been folded into a single $z_b^{(t_i)}$, through which long-term, short-term, and episodic evidence must also be expressed despite different temporal support.

Problem Statement: Formally, given a frozen encoder state $z_b^{(t_i)}$ and the prefix $\{e_b^{(t_j)}\}_{j=1}^{i}$ that produced it, can we recover under-expressed evidence and correct the state before the downstream task head consumes it? Here, recoverability means that useful preference evidence may be weakly expressed in $z_b^{(t_i)}$ while remaining available in the cached per-timestep representations; it does not imply that such information is completely absent from the compressed state.

4 Methodology

We instantiate REPAIR under three constraints: the host encoder is frozen, the raw prefix $\tau_b^{t_{1:i}}$ is not re-encoded, and only the exposed state is corrected before the downstream head consumes it. For encoder hosts, $e_b^{(t_j)}$ denotes the per-timestep event representation available from the frozen host computation, implemented through the shared eventization using a one-hot action embedding with T5-based content embeddings and CLIP image features where applicable. REPAIR compares $z_b^{(t_i)}$ with timestep-level evidence from the same trajectory to construct the correction. For maximum prefix length L, the event cache costs $O(Ld_e)$ per active trajectory, is bounded by the host truncation window, and is discarded after rollout (Notations: Table 7).

Abstraction. REPAIR corrects $z_b^{(t_i)}$ through five operations: state-relative evidence construction, temporal resolution, selective retention, aggregation, and residual fusion:

$$\tilde{z}_b^{(t_i)} = F_{\text{rep}}\left(z_b^{(t_i)}, \mathcal{A}_{\text{rep}}\left(\left\{\Pi_\nu^{(t_i)}\left(G_{\text{rep}}\left(d^{(t_j)}, \boldsymbol{\kappa}(t_i - t_j)\right)\right)\right\}_{j=1}^{i}\right)\right). \tag{1}$$

A historical timestep can be redundant, stale, or corrective depending on what the compressed state already expresses. Hence, REPAIR converts each event into evidence relative to the state being

repaired rather than selecting events by raw salience or lag alone. Figures 1–2 and Appendix A summarize the pipeline; proofs are in Appendix B.

I. State-relative Corrective Evidence. A local event representation $e_b^{(t_j)}$ can be corrective if it provides evidence under-expressed in $z_b^{(t_i)}$, redundant if already captured, or stale if later history has moved away. Learned projections map the event and state to a shared space as $\bar{z}_b^{(t_i)} = P_z z_b^{(t_i)}$ and $\bar{e}_b^{(t_j)} = P_e e_b^{(t_j)}$. Their difference $r^{(t_j;t_i)} = \bar{e}_b^{(t_j)} - \bar{z}_b^{(t_i)}$ is a state-relative corrective signal, not a claim that the corresponding information is literally absent from the host state.

REPAIR reads this signal through a learned rank-K correction operator:

$$\mathcal{J}_{\text{rep}} = U_{\text{rep}} \Lambda_{\text{rep}} V_{\text{rep}}^{\top}, \qquad \text{rank}(\mathcal{J}_{\text{rep}}) \le K, \qquad d^{(t_j)} = V_{\text{rep}}^{\top} r^{(t_j;t_i)}. \tag{2}$$

Here, $\Lambda_{\text{rep}} = \text{diag}(\lambda_{1:K}^{\text{rep}})$, $P_z \in \mathbb{R}^{m \times d_z}$ and $P_e \in \mathbb{R}^{m \times d_e}$ define the shared repair space, $V_{\text{rep}} \in \mathbb{R}^{m \times K}$ reads the signal into K corrective coordinates, and $U_{\text{rep}}\Lambda_{\text{rep}}$ provides the corresponding scale-and-write factors. The factorization is SVD-inspired but learned rather than computed by SVD, and its factors need not be orthogonal. Thus, $d^{(t_j)}$ is a learned corrective representation rather than a native host residual.

We use $K = 128$ as a shared repair dimension across hosts, while the empirical sweep evaluates $K \in \{32, 64, 128, 192\}$. Appendix B.1 gives the conditional low-rank analysis, and **Theorem** 2 characterizes recoverability when the desired correction is representable through the retained corrective coordinates and write space. Across the prefix, the corrective field $\mathcal{D}_i = \{d^{(t_j)}\}_{j=1}^{i}$ is centered as $\tilde{d}^{(t_j)} = d^{(t_j)} - \frac{1}{i}\sum_{q=1}^{i} d^{(t_q)}$ to separate timestep-specific variation from prefix-wide offset.

II. Multi-regime Resolution. The centered corrective field may contain evidence with different temporal support. REPAIR therefore resolves it through long-support, short-support, and episodic-support views. Let $\ell_j = i - j$ denote the lag of timestep t_j:

$$\begin{aligned} \kappa^L(\ell_j) &= \exp(-\beta \ell_j), \qquad \kappa^S(\ell_j) = \max\left(\mathbf{1}_K - \frac{\ell_j}{i}\mathbf{1}_K, \mathbf{0}_K\right), \\ \kappa^E(\ell_j) &= \exp(\ell_j \log \xi - \xi - \log \Gamma(\ell_j + 1)\mathbf{1}_K). \end{aligned} \tag{3}$$

where $\beta, \xi \in \mathbb{R}_+^K$ are learnable positive parameters. κ^L provides gradually decaying support over wider history, κ^S emphasizes recent evidence, and κ^E provides localized support around learned lags. These are temporal views of the same corrective activations rather than semantic labels for raw interactions.

Theorem 3 characterizes when the temporal components are identifiable, while Corollary B.3 characterizes finite-prefix patterns representable by their combination. The resolved evidence is

$$\eta^{(t_j)} = G_{\text{rep}}\left(\tilde{d}^{(t_j)}, \boldsymbol{\kappa}(\ell_j)\right) = \sum_{\rho \in \{L,S,E\}} W_\rho \left(\kappa^\rho(\ell_j) \odot \tilde{d}^{(t_j)}\right), \tag{4}$$

where $W_L, W_S, W_E \in \mathbb{R}^{K \times K}$ are learned temporal-resolution matrices.

III. Selective Retention. Since the temporal views are combined rather than selected exclusively, some resolved activations can remain weak, noisy, or redundant. REPAIR therefore applies timestep-pattern selective retention as $\Pi_\nu^{(t_i)}(\eta^{(t_j)}) = \pi_j^{(t_i)} \odot \eta^{(t_j)}$, where $\pi_j^{(t_i)} \in \{0,1\}^K$ is the binary retention mask. The adaptive top-ν selector retains the strongest timestep-pattern activations and suppresses components not selected as corrective with respect to the exposed state. Selector statistics and straight-through masking are given in Appendix A.5.

IV. Correction Aggregation and Fusion. The retained field is synthesized into one state correction using the state-conditioned gate $\gamma^{(t_i)} = \text{sigmoid}\left(\text{MLP}_\gamma(z_b^{(t_i)})\right)$:

$$\Delta_b^{(t_i)} = \mathcal{A}_{\text{rep}}(.) = P_\Delta \left(\sum_{j=1}^{i} \alpha_j^{(t_i)} U_{\text{rep}} \Lambda_{\text{rep}} \left(\gamma^{(t_i)} \odot \Pi_\nu^{(t_i)}(\eta^{(t_j)}) \right) \right). \tag{5}$$

Here, $\alpha_j^{(t_i)}$ is a normalized attention weight over retained timestep evidence, and $P_\Delta \in \mathbb{R}^{d_z \times m}$ maps the synthesized signal back to the compressed-state space. The repaired state is then formed by

Table 1: Evaluation coverage across tasks, datasets, frozen hosts, protocols, and metrics. Hosts are first trained under their original supervision and then frozen when REPAIR is attached.

Task	Dataset	Frozen Hosts	Test Protocol	Metrics
Movie reco.	MovieLens	SIGMA, Mamba4Rec, SSDRec, SASRec, TiSASRec, HSTU, FEARec, GRU4Rec, BERT4Rec	15K test instances; 1 positive + 499 negatives	MRR, nDCG@5/10, Hit@10
News reco.	MIND	NAML, EBNR, NRMS, LSTUR, MINER, TrRMIo	20K test instances; 1 positive + 499 negatives	MRR, nDCG@5/10, Hit@10
	PENS	NAML, EBNR, NRMS, LSTUR, MINER, TrRMIo, DeepSeek-Qwen2.5-Base-32B, Qwen2.5-32B	20K test instances; 1 positive + 499 negatives	MRR, nDCG@5/10, Hit@10
Product reco.	Amazon Reviews 2023	SigLIP2, OpenCLIP, DINOv2-E5	2K test instances; 1 positive + 499 negatives	MRR, nDCG@5/10, Hit@10
Personalized headline generation	PENS	NAML=T1/T2, EBNR=T1/T2, NRMS=T1/T2, GTP, SCAPE, FPG, Walk2Pers, IMPerSumm, DeepSeek-32B, Qwen2.5-32B	20K test instances	ROUGE, weighted/paraphrase-aware ROUGE, METEOR, BLEU, PerSEval

residual-gated fusion:

$$
\begin{aligned}
g_F^{(t_i)} &= \mathrm{sigmoid}\Big(\mathrm{MLP}_F([z_b^{(t_i)}; \Delta_b^{(t_i)}])\Big), \\
\tilde{z}_b^{(t_i)} &= z_b^{(t_i)} + g_F^{(t_i)} \odot \mathrm{MLP}_{\mathrm{probe}}([z_b^{(t_i)}; \Delta_b^{(t_i)}]).
\end{aligned}
\tag{6}
$$

Thus, $z_b^{(t_i)}$ remains the base state and $\Delta_b^{(t_i)}$ provides a selective correction rather than a fresh encoding of the raw history. The identity mapping is representable when the fusion contribution vanishes, although the learned gate does not impose pointwise non-degradation. The realized correction remains user- and prefix-specific through the corrective field, selector, state gate, aggregation weights, and fusion.

For prompt-conditioned LLM hosts, REPAIR follows the same repair logic through a different evidence interface. Since an autoregressive LLM does not directly expose one interaction-level representation per event from the same forward pass, individual interactions are encoded as corrective evidence while the frozen LLM provides $z_b^{(t_i)}$. The repaired state $\tilde{z}_b^{(t_i)}$ is then mapped to a continuous prefix for the frozen decoder (Appendix A.8, Figures 3 and 4). Hyperparameters are given in Table 8.

Training Objective. REPAIR is trained so that the repaired state improves the downstream objective through $\mathcal{L}_{\mathrm{task}}$, retains timestep provenance through $\mathcal{L}_{\mathrm{pos}}$, and remains coherent with source evidence through $\mathcal{L}_{\mathrm{coh}}$. The objective is $\mathcal{L}_{\mathrm{repair}} = \lambda_{\mathrm{task}}\mathcal{L}_{\mathrm{task}} + \lambda_{\mathrm{pos}}\mathcal{L}_{\mathrm{pos}} + \lambda_{\mathrm{coh}}\mathcal{L}_{\mathrm{coh}}$, with $\lambda_{\mathrm{task}} > \lambda_{\mathrm{coh}} > \lambda_{\mathrm{pos}}$. Theorem 4 gives the corresponding conditional relation to recoverability error under the local assumptions stated in Appendix B.4; the local PL condition is required only for the additional convergence-rate statement. The progressive objective ablation is reported in Table 17 (Appendix G.4).

5 Evaluation

We evaluate REPAIR through four questions: frozen-host gains and continued task-head training (RQ1), budget-matched post-head correctors (RQ2), repair-space capacity and multi-timescale selective repair (RQ3), and deployment cost-effectiveness (RQ4). Table 1 summarizes coverage.

I. Experimental Setup. We evaluate movie, news, multimodal product, and conversation recommendation, together with personalized headline and dialogue response generation. Following Chatterjee et al. [2025], we construct UIG trajectory pools with strict user-level train/test separation. Each observed prefix $\tau_b^{t_{1:i}}$ is paired with the next target $r_u^{\star(t_{i+1})}$; for generation, the task-side query is attached at t_i. Full UIG and dataset details are in Appendix A.1.

Evaluation is autoregressive from t_{i+1}: the model conditions on $\tau_b^{t_{1:i}}$ and, when applicable, the query at t_i, produces the next output, and appends it to the running trajectory. Recommendation uses one ground-truth target with 499 negatives and reports MRR, nDCG@5, nDCG@10, and Hit@10. Generation reports ROUGE, METEOR, BLEU, BERTScore, weighted ROUGE, and PerSEval metrics for PENS. For RQ4, we report memory/latency overhead and MRR-GLR/GMR. Metric definitions are in Appendix D.

II. Frozen-host Control Protocol. For each dataset–host pair, all RQ1 conditions start from the same task-trained host checkpoint. *Original Host* uses the trained encoder and task head unchanged. *Head-only FT* keeps the encoder and task-facing state unchanged and further trains only the original task head, selecting the best validation checkpoint from a 1–5 epoch sweep. ***REPAIR (Frozen Head)***

freezes both the task-trained encoder and original task head and trains only REPAIR from cached per-timestep representations. ***REPAIR + Head Alignment*** starts from the trained repair checkpoint and jointly updates REPAIR and the task head for two further epochs while keeping the encoder frozen; we refer to this final condition as full REPAIR. All conditions share the same data split and evaluation protocol; optimization and checkpoint selection are detailed in Appendix A.7.

III. Frozen Host Models. We test post-hoc preference-state repair, not a new end-to-end host. Hosts are grouped by task and encoder regime. Movie recommendation includes state-space/transduction, denoising/frequency-aware, self-attentive, time-aware, bidirectional, and recurrent hosts. News recommendation includes multi-view attention, recurrent user-state, self-attentive, long–short interest, and multi-interest encoders. Product recommendation uses frozen multimodal product encoders, while personalized generation uses encoder-based PENS variants and specialized headline generators. Conversation tasks are treated as broader-applicability stress tests. All conditions use the same data split and evaluation protocol. Training schedules, optimization, and checkpoint selection are detailed in Appendix C.3.

IV. Post-Head Correctors. For RQ2, we compare REPAIR with two instantiated post-head correctors under the same frozen-host and adaptation protocol. Rec-Denoiser-Corr adapts Rec-Denoiser to suppress low-utility trajectory-to-prediction links after the base prediction context is formed, while SERAC-Corr adapts SERAC as a memory-based corrector that retrieves correction states after observing the prediction context. All correction settings start from the same task-trained host, keep the host encoder frozen, and use the same two-epoch adaptation budget. This comparison evaluates the instantiated pre-head and post-head correction settings under a common protocol. Details are in Appendix F.

V. Repair-Space and Temporal Diagnostics. For RQ3, we vary the repair dimension over $K \in \{32, 64, 128, 192\}$ to examine whether performance increases simply with repair capacity. We also retain the controlled long-term (L-Tr), short-term (S-Tr), and episodic (E-Tr) trajectory perturbations, together with temporal-pathway and selective-retention ablations. These diagnostics test the compactness, temporal support, and selectivity of the learned correction without treating K as an intrinsic preference rank or the temporal views as semantic classes of user behavior.

VI. Statistical Evaluation. For OpenCLIP–Amazon, SigLIP2–Amazon, and EBNR–PENS, we report results across three independent training seeds together with 95% confidence intervals from 10,000 user-clustered bootstrap resamples, accounting for multiple prefixes from the same held-out user. Extended settings without repeated-run uncertainty are reported as point estimates. Statistical claims are therefore restricted to the settings evaluated under this strengthened protocol.

6 Results and Insights

We evaluate REPAIR w.r.t. the four core RQs. For OpenCLIP–Amazon, SigLIP2–Amazon, and EBNR–PENS, statistical uncertainty is evaluated across three independent training seeds with 10,000 user-clustered bootstrap resamples; the remaining extended settings are reported as point estimates (Appendix A.9).

6.1 RQ1: Corrective Consistency of REPAIR

Does frozen-head repair improve states? We first ask whether repairing the compressed preference state improves downstream personalization while both the task-trained encoder and original task head remain frozen. From Table 2, training only REPAIR improves all twelve representative recommendation hosts on MRR and nDCG@10, while head-only fine-tuning of the unchanged encoder state yields smaller gains in all twelve settings. For Mamba4Rec–MovieLens, EBNR–PENS, and OpenCLIP–Amazon, frozen-head REPAIR gives $+3.96$, $+1.42$, and $+2.84$ MRR, compared with $+0.19$, $+0.28$, and $+0.62$ from head-only fine-tuning. Thus, useful improvement is already present when only the compressed preference state is repaired.

Does head alignment add gains? The full REPAIR setting further aligns the repaired state with the downstream task head while keeping the encoder frozen, improving MRR and nDCG@10 over frozen-head REPAIR in 11 of 12 representative settings. Across the complete predictive evaluation, REPAIR consistently improves movie, news, and multimodal product recommendation. On MovieLens, it improves all seven frozen encoders, with average gains of $+2.48/+2.77/+2.08/+3.90$

Table 2: **RQ-1: Recommendation performance under frozen-host controls.** Columns compare the original task-trained host, head-only fine-tuning on the unchanged encoder state, REPAIR with both encoder and original task head frozen, and full REPAIR with two-epoch downstream-head alignment while keeping the encoder frozen. Values are MRR / nDCG@10. S/M/W denote the strongest, median, and weakest original hosts w.r.t. MRR. Full host results are in Tables 9–12.

Dataset	Host	Original Host	Head-only FT	REPAIR (Frozen Head)	REPAIR + Head Alignment
MovieLens	S: TiSASRec	35.51 / 41.14	35.57 / 41.19	36.38 / 42.07	36.78 / 42.91
	M: Mamba4Rec	29.29 / 38.50	29.48 / 38.89	33.25 / 39.20	34.26 / 40.17
	W: SASRec	26.43 / 33.82	27.05 / 34.37	29.57 / 40.08	28.49 / 35.18
PENS	S: LSTUR	10.44 / 12.15	10.98 / 12.32	11.98 / 12.71	12.85 / 15.88
	M: EBNR	2.65 / 2.45	2.93 / 2.86	4.07 / 3.97	11.97 / 14.28
	W: NAML	1.29 / 0.81	2.75 / 1.46	5.20 / 5.21	12.38 / 15.13
MIND	S: LSTUR	51.43 / 59.63	52.37 / 59.96	54.17 / 62.27	55.84 / 66.92
	M: EBNR	47.08 / 51.63	47.63 / 52.31	49.43 / 55.17	53.22 / 63.18
	W: SCAPE-GRU	42.53 / 41.19	43.21 / 41.82	43.31 / 44.25	45.32 / 47.36
Amazon	S: SigLIP2	2.40 / 2.18	2.67 / 2.72	4.16 / 4.05	17.72 / 20.35
	M: OpenCLIP	1.59 / 1.05	2.21 / 2.17	4.43 / 4.66	19.49 / 22.08
	W: DINOv2-E5	1.19 / 0.72	2.35 / 2.42	3.16 / 2.90	16.15 / 18.19

Table 3: **RQ2: Comparison with other input protocol-matched correctors under the same frozen-host setting**. The corrective baselines have been deployed after 2-epoch head finetuning for fair evaluation.

Tasks (Datasets)	Host Model	Base		+SERAC-Corr		+Rec-Denoiser-Corr		+REPAIR	
		MRR	nDCG@10	MRR	nDCG@10	MRR	nDCG@10	MRR	nDCG@10
Movie Reco. (MovieLens)	TiSASRec	35.51	41.14	33.17	41.32	34.21	42.08	36.78	42.91
	Mamba4Rec	29.29	38.50	29.81	38.63	31.32	39.22	34.26	40.17
	SIGMA	33.27	38.96	31.95	38.98	29.43	39.61	34.81	40.72
News Reco. (MIND)	LSTUR	51.43	59.63	53.66	62.17	52.45	61.35	55.84	66.92
	NAML	45.65	46.56	46.47	51.33	48.12	48.94	53.22	63.18
	NRMS	44.91	45.73	48.13	46.74	42.25	48.23	52.74	62.44
Product Reco. (Amazon)	SigLIP2	2.40	2.18	7.43	8.67	9.54	10.61	17.72	20.35
	OpenCLIP	1.59	1.05	5.32	7.41	6.37	8.11	19.49	22.08

on MRR/nDCG@5/nDCG@10/HR@10; the largest gain appears on Mamba4Rec ($+4.97$ MRR), while the strongest frozen host, TiSASRec, still improves by $+1.27$ MRR. The same pattern holds on MIND and structured PENS, with average MRR gains of $+5.88$ and $+6.98$, respectively. On Amazon Reviews 2023, SigLIP2, OpenCLIP, and DINOv2-E5 improve by $+15.33$, $+17.91$, and $+14.96$ MRR. These results show that compressed preference states remain repairable across the evaluated frozen hosts.

Does repair transfer across hosts? The cross-host transfer stress test further examines whether the learned correction remains useful beyond its source host. A REPAIR module trained on Mamba4Rec improves weaker frozen hosts, giving $+5.31/+5.68/+6.19/+9.33$ on GRU4Rec and $+6.39/+7.26/+8.13/+13.09$ on BERT4Rec over MRR/nDCG@5/nDCG@10/HR@10. Thus, the learned corrective mechanism can remain useful across compatible frozen host states rather than acting only as a host-specific score adjustment.

How does generation use repair? The generative results expose a decoder-utilization boundary (Tables 13–14). On PENS personalized headline generation, REPAIR gives large accuracy-oriented gains for DeepSeek-Qwen-2.5-base-32B, with $+53.21\%/+171.27\%/+45.49\%$ on ROUGE-L/ROUGE-SU4/METEOR, while its PerSEval gains remain comparatively smaller. This confirms that stronger surface generation does not automatically imply stronger user-specific adaptation [Dasgupta et al., 2024, Vansh et al., 2023]. The contrast is clearer for preference-coupled decoders. Walk2Pers has modest lexical gains, such as $+5.18\%$ ROUGE-1 and $+6.58\%$ ROUGE-L, while improving more strongly on PerSEval, with $+13.25\%$ PSE-W-JSD, $+15.32\%$ PSE-W-RG-L, $+10.76\%$ PSE-P-RG-L, and $+19.13\%$ PSE-METEOR. IMPerSumm shows larger personalization-sensitive gains of $+20.76\%$ PSE-W-JSD, $+25.23\%$ PSE-W-RG-L, and $+39.29\%$ PSE-P-RG-L. Thus, the realized benefit depends on how the downstream generator uses the repaired preference state.

6.2 RQ2: Comparison with Budget-Matched Post-head Correctors

Does pre-head correction outperform alternatives? We next compare REPAIR with two protocol-matched post-head correctors, SERAC-Corr and Rec-Denoiser-Corr, under the same frozen-host setting. Table 3 shows that REPAIR is strongest across all eight reported predictive comparisons.

Table 4: **RQ3: Repair-space capacity sensitivity.** MRR / nDCG@10 for full REPAIR across tested repair dimensions K. Bold indicates the strongest tested setting for each host.

Host / Dataset	Original Host	$K = 32$	$K = 64$	$K = 128$	$K = 192$
OpenCLIP / Amazon	1.59 / 1.05	9.31 / 9.78	11.17 / 12.22	**19.49 / 22.08**	11.63 / 12.16
SigLIP2 / Amazon	2.40 / 2.18	10.44 / 11.36	8.42 / 8.77	**17.72 / 20.35**	13.46 / 13.15
EBNR / PENS	2.65 / 2.45	6.13 / 6.19	8.65 / 8.58	**11.97 / 14.28**	11.23 / 13.17
Mamba4Rec / MovieLens	29.29 / 38.50	31.47 / 38.85	32.38 / 39.26	**34.26 / 40.17**	34.02 / 39.96
LSTUR / MIND	51.43 / 59.63	52.67 / 62.14	53.49 / 64.13	**55.84 / 66.92**	54.73 / 66.12

Table 5: **RQ3: temporal-kernel diagnostic.** Columns report MRR under κ^L, κ^S, κ^E, and κ^{LSE} across L-Tr/S-Tr/E-Tr perturbations; the final block gives expected kernel / best single kernel and the expected-minus-best gap.

Datasets	Base Encoder	Base (w/o REPAIR)			κ^L			κ^S			κ^E			$\kappa^L + \kappa^S + \kappa^E$			Expected / Best Single Kernel		
		L-Tr	S-Tr	E-Tr	L-Tr	S-Tr	E-Tr	L-Tr	S-Tr	E-Tr	L-Tr	S-Tr	E-Tr	L-Tr	S-Tr	E-Tr	L-Tr	S-Tr	E-Tr
MovieLens	Mamba4Rec	26.43	26.12	25.72	32.07	30.28	30.14	33.95	33.92	33.88	30.04	30.22	32.57	***34.87***	***35.52***	***34.39***	κ^L/κ^S (-1.88)	κ^S/κ^S (+0.00)	κ^E/κ^S (-1.31)
	TiSASRec	33.25	33.17	31.83	36.19	35.93	35.92	36.18	36.20	36.04	35.16	35.21	35.08	***37.38***	***36.93***	***36.63***	κ^L/κ^L (+0.00)	κ^S/κ^S (+0.00)	κ^E/κ^S (-0.96)
MIND	NAML	46.21	46.25	45.32	52.12	51.76	53.70	51.79	52.03	51.56	50.89	51.36	53.35	***53.31***	***54.55***	***53.80***	κ^L/κ^L (+0.00)	κ^S/κ^S (+0.00)	κ^E/κ^L (-0.35)
	EBNR	47.44	48.32	44.19	54.93	53.64	53.03	55.08	52.14	52.89	54.29	52.23	53.70	***56.79***	***55.69***	53.19	κ^L/κ^S (-0.15)	κ^S/κ^L (-1.50)	κ^E/κ^E (+0.00)

On MovieLens, the best competing corrector lags behind REPAIR by $2.94/0.95$ for Mamba4Rec, $2.57/0.83$ for TiSASRec, and $2.86/1.11$ for SIGMA on MRR/nDCG@10. On MIND, the best competing SERAC-Corr still lags by $2.18/4.75$ for LSTUR and $4.61/14.21$ for NRMS. The same pattern is stronger in multimodal product recommendation, where Rec-Denoiser-Corr lags by $8.18/9.74$ for SigLIP2 and $13.13/13.97$ for OpenCLIP. Thus, under the shared protocol, the evaluated pre-head REPAIR setting yields higher predictive performance than the two instantiated post-head alternatives.

Does this hold for generation? For personalized headline generation, the strongest competing Rec-Denoiser-Corr still lags behind REPAIR by $0.008/0.0007$ on PSE-W-JSD/PSE-METEOR for SCAPE, and by $0.007/0.0002$ for GTP. Hence, the evaluated post-head alternatives do not close the subjectivity-sensitive gap observed under pre-head state repair.

6.3 RQ3: Temporal Resolution and Selective Retention

How much repair capacity suffices? Across $K \in \{32, 64, 128, 192\}$, $K = 128$ gives the strongest MRR and nDCG@10 in all five tested host–dataset settings (Table 4), while increasing the repair dimension to $K = 192$ lowers both metrics in all five. This non-monotonic trend shows that increasing repair capacity alone does not explain the gains; $K = 128$ is therefore the strongest tested shared repair dimension rather than a claimed universal optimum.

Does repair follow input timing? We test whether κ^L, κ^S, and κ^E act as soft time-scale resolution operators. To isolate temporal support from semantic unrelatedness, we construct controlled MovieLens and MIND variants using the same semantic-proximity criterion. Long-term (L-Tr) injects semantically proximal perturbations within node-embedding similarity 0.75 over about 60% of a trajectory, short-term (S-Tr) restricts the same perturbation to the tail 40%, and episodic-burst (E-Tr) inserts semantically proximal bursts at random positions for 3–4 contiguous timesteps. Each variant contains 1K trajectories. The expected single kernel matches the best single kernel in 6 of 12 controlled probes. The deviations are informative: Mamba4Rec favors κ^S across all three regimes, suggesting that selective state-space compression contains part of the stable or burst-local evidence and leaves a more local corrective signal. Similarly, E-Tr in TiSASRec and NAML can be expressed through short-term or topic-coherent corrective evidence. Importantly, the full κ^{L+S+E} variant is strongest in 11 of 12 settings. Hence, raw perturbation timescale and corrective timescale after host compression need not coincide, and the kernels act as *soft* temporal probes after compression. Additional ablations are in Table 16.

Does selective retention improve repair? We test whether retaining only the strongest timestep-level evidence improves repair. In REPAIR, denoising acts on the *corrective residual field*, not the raw trajectory. At full history length ($l = 50$), selective repair improves all four representative hosts over the non-selective variant on MRR/nDCG@10: Mamba4Rec by $+0.31/+0.14$ with $34/50$ selected timesteps, TiSASRec by $+0.22/+0.25$ with $36/50$, NAML by $+1.05/+3.43$ with $36/50$, and EBNR by $+1.11/+2.44$ with $35/50$. Comparable support sizes with different gains indicate host-specific compression effects: smaller gains for Mamba4Rec/TiSASRec align with input-selective propagation and time-aware comparison, whereas larger gains for NAML/EBNR suggest that pooled or recurrent

evidence can retain redundant, weakly corrective, or state-misaligned information. Thus, REPAIR improves when temporal resolution is followed by state-relative denoising, allowing only timesteps that remain corrective after host compression to shape the repaired state (Table 18).

Does repair tolerate noisy trajectories? We stress-test REPAIR by inserting abrupt contents into 35–40% of each trajectory, which causes an average MRR drop of 14.69 for the frozen hosts. Despite this degradation, REPAIR improves the hosts by +2.95 MRR and +3.34 nDCG@10 on average, with MRR gains of +3.22/+2.31/+4.24/+2.02 for Mamba4Rec, TiSASRec, NAML, and EBNR, respectively. This suggests that state-relative residual selection can recover useful corrective evidence even when trajectories contain disconnected noisy subsequences. Full results are in Table 19.

6.4 RQ4: Deployment Feasibility

What does repair cost? We first measure the additional serving cost of REPAIR relative to the frozen predictive hosts. Table 6 shows 31–34% memory overhead and 38–44% latency overhead across the main predictive hosts. In absolute computation at history length $T = 50$, REPAIR adds 0.18 GFLOPs/query to Mamba4Rec (0.75 → 0.93), 0.18 to TiSASRec (1.45 → 1.63), and 0.02 to NAML (0.03 → 0.05). The relative overhead therefore varies with the cost of the underlying frozen host, while the added computation remains small in absolute terms.

Are the gains worth it? We next relate predictive improvement to the added serving cost using MRR-gain-to-latency ratio (MRR-GLR) and MRR-gain-to-memory ratio (MRR-GMR), both relative to the frozen host. REPAIR obtains MRR-GLR in the range 0.75–0.84 and MRR-GMR in the range 0.79–0.89 across the main predictive hosts. The strongest gain-per-cost tradeoff appears for Mamba4Rec, whereas TiSASRec has a smaller ratio because its absolute MRR improvement is lower under comparable repair overhead. Importantly, these costs do not include a second pass over the raw history: the repair operates on cached timestep representations from the frozen host computation. Thus, RQ4 characterizes a bounded recovery–cost tradeoff rather than treating the intervention as cost-free.

Table 6: **RQ4:** Deployment feasibility of REPAIR on representative hosts. Δ is the absolute MRR gain over the frozen host. Rel. Gain is the % gain. MemOH/LatOH are memory and latency overheads. GLR/GMR are gain-efficiency ratios. Details in Tables 20, 21.

Model	Δ	Rel. Gain (%)	MemOH (%)	LatOH (%)	GLR	GMR
Mamba4Rec	+4.97	16.97	32.0	40.0	0.84	0.89
TiSASRec	+1.27	3.58	31.0	38.0	0.75	0.79
NAML	+7.57	16.58	34.0	44.0	0.81	0.87
Context-tuned LLM Reference						
DeepSeek-32B	+0.52	39.69	0.1	12.0	1.25	1.40

7 Broader Applicability and Limitations

We use TG-ReDial as a broader-applicability stress test rather than primary evidence, since dialogue histories contain turns, entities, recommendations, acceptances, rejections, and topic shifts rather than clean action-content events. A principled UIG requires additional discourse modeling. The results are nevertheless positive on stronger sequential hosts, with REPAIR improving SASRec by +0.45 MRR and +0.94 HR@10, but the smaller gains indicate that repair quality depends on eventization fidelity. Conversational generation shows the same decoder-utilization boundary: REDIAL improves substantially, whereas the Transformer generator changes only marginally, suggesting that repaired preference evidence helps most when the downstream generator is coupled to the user state. We leave discourse-level eventization with dialogue acts, entity mentions, topic shifts, and acceptance/rejection signals for future work (see Table 15).

8 Conclusion

We present REPAIR, a selective pre-head repair method for frozen personalization hosts. Rather than re-encoding the trajectory or correcting only the output, REPAIR recovers state-relative timestep evidence and fuses a low-rank, temporally resolved correction into the exposed preference state. Across predictive and generative personalization tasks, the results show that compressed preference states remain repairable, that the evaluated pre-head configuration outperforms the two instantiated post-head correctors under the shared protocol, and that the gains remain deployment-bounded. These findings support state repair as a practical alternative to full encoder retraining for evolving personalization.

Acknowledgments and Disclosure of Funding

Tanmoy Chakraborty acknowledges the financial support of the Anusandhan National Research Foundation (ANRF/ARG/2025/001505/QSS) and the Rajiv Khemani Young Faculty Chair Professorship in Artificial Intelligence.

References

Jaewoo Ahn, Yeda Song, Sangdoo Yun, and Gunhee Kim. MPCHAT: Towards multimodal persona-grounded conversation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3354–3377, Toronto, Canada, jul 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.189. URL https://aclanthology.org/2023.acl-long.189/.

Mingxiao An, Fangzhao Wu, Chuhan Wu, Kun Zhang, Zheng Liu, and Xing Xie. Neural news recommendation with long- and short-term user representations. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 336–345, Florence, Italy, jul 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1033. URL https://aclanthology.org/P19-1033/.

Xiang Ao, Xiting Wang, Ling Luo, Ying Qiao, Qing He, and Xing Xie. PENS: A dataset and generic framework for personalized news headline generation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *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 82–92, Online, aug 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.7. URL https://aclanthology.org/2021.acl-long.7/.

Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. Controllable multi-interest framework for recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2942–2951. Association for Computing Machinery, 2020. doi: 10.1145/3394486.3403344. URL https://doi.org/10.1145/3394486.3403344.

Parthiv Chatterjee, Shivam Sonawane, Amey Hengle, Aditya Tanna, Sourish Dasgupta, and Tanmoy Chakraborty. Diversity augmentation of dynamic user preference data for boosting personalized text summarizers. *Trans. Mach. Learn. Res.*, 2025. URL https://openreview.net/forum?id=JVx7Qi8tz3.

Huiyuan Chen, Yusan Lin, Menghai Pan, Lan Wang, Chin-Chia Michael Yeh, Xiaoting Li, Yan Zheng, Fei Wang, and Hao Yang. Denoising self-attentive sequential recommendation. In *Proceedings of the 16th ACM Conference on Recommender Systems*, RecSys '22, pages 92–101, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392785. doi: 10.1145/3523227.3546788. URL https://doi.org/10.1145/3523227.3546788.

Sourish Dasgupta, Ankush Chander, Tanmoy Chakraborty, Parth Borad, and Isha Motiyani. PerSEval: Assessing personalization in text summarizers. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=yqT7eBz1VJ.

Xinyu Du, Huanhuan Yuan, Pengpeng Zhao, Jianfeng Qu, Fuzhen Zhuang, Guanfeng Liu, Yanchi Liu, and Victor S. Sheng. Frequency enhanced hybrid attention network for sequential recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, pages 78–88, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450394086. doi: 10.1145/3539618.3591689. URL https://doi.org/10.1145/3539618.3591689.

Balazs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks, 2016. URL https://arxiv.org/abs/1511.06939.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR, 2019. URL https://proceedings.mlr.press/v97/houlsby19a.html.

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

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation, 2018. URL https://arxiv.org/abs/1808.09781.

Jiacheng Li, Yujie Wang, and Julian McAuley. Time interval aware self-attention for sequential recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, WSDM '20, pages 322–330, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368223. doi: 10.1145/3336191.3371786. URL https://doi.org/10.1145/3336191.3371786.

Jian Li, Jieming Zhu, Qiwei Bi, Guohao Cai, Lifeng Shang, Zhenhua Dong, Xin Jiang, and Qun Liu. MINER: Multi-interest matching network for news recommendation. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022*, pages 343–352, Dublin, Ireland, may 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.29. URL https://aclanthology.org/2022.findings-acl.29/.

Xiang Lisa Li and Percy Liang. 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*, pages 4582–4597, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.353. URL https://aclanthology.org/2021.acl-long.353/.

Zihao Li, Aixin Sun, and Chenliang Li. Diffurec: A diffusion model for sequential recommendation. *ACM Trans. Inf. Syst.*, 42(3), dec 2023. ISSN 1046-8188. doi: 10.1145/3631116. URL https://doi.org/10.1145/3631116.

Junhong Lian, Xiang Ao, Xinyu Liu, Yang Liu, and Qing He. Panoramic interests: Stylistic-content aware personalized headline generation. In *Companion Proceedings of the ACM on Web Conference 2025*, 2025.

Chengkai Liu, Jianghao Lin, Jianling Wang, Hanzhou Liu, and James Caverlee. Mamba4rec: Towards efficient sequential recommendation with selective state space models, 2024. URL https://arxiv.org/abs/2403.03900.

Shuai Liu, Hyundong Cho, Marjorie Freedman, Xuezhe Ma, and Jonathan May. RECAP: Retrieval-enhanced context-aware prefix encoder for personalized dialogue response generation. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8404–8419, Toronto, Canada, jul 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.468. URL https://aclanthology.org/2023.acl-long.468/.

Zhenyi Lu, Wei Wei, Xiaoye Qu, Xian-Ling Mao, Dangyang Chen, and Jixiong Chen. Miracle: Towards personalized dialogue generation with latent-space multiple personal attribute control. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5933–5957, Singapore, dec 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.395. URL https://aclanthology.org/2023.findings-emnlp.395/.

Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, Qifan Wang, Si Zhang, Ren Chen, Chris Leung, Jiajie Tang, and Jiebo Luo. LLM-rec: Personalized recommendation via prompting large language models. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Findings of the*

Association for Computational Linguistics: NAACL 2024, pages 583–612, Mexico City, Mexico, jun 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.39. URL https://aclanthology.org/2024.findings-naacl.39/.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html.

Kevin Meng, Arnab Sen Sharma, Alex J. Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=MkbcAHIYgyS.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. Fast model editing at scale. In *International Conference on Learning Representations*, 2022a. URL https://openreview.net/forum?id=0DcZxeWfOPt.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. Memory-based model editing at scale. In *International Conference on Machine Learning (ICML)*, 2022b. URL https://proceedings.mlr.press/v162/mitchell22a.html.

Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. Embedding-based news recommendation for millions of users. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, pages 1933–1942, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450348874. doi: 10.1145/3097983.3098108. URL https://doi.org/10.1145/3097983.3098108.

Divya Patel, Pathik Patel, Ankush Chander, Sourish Dasgupta, and Tanmoy Chakraborty. Are large language models in-context personalized summarizers? get an iCOPERNICUS test done! In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16820–16842, Miami, Florida, USA, nov 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.935. URL https://aclanthology.org/2024.emnlp-main.935/.

Mona Schirmer, Anton Thielmann, Pola Schwöbel, Thomas Martynec, Giuseppe Di Benedetto, Ben London, and Yannik Stein. *Aligning Recommendations with User Popularity Preferences*, page 4773–4793. Association for Computing Machinery, New York, NY, USA, 2026. ISBN 9798400725968. URL https://doi.org/10.1145/3805689.3806483.

Yun-Zhu Song, Yi-Syuan Chen, Lu Wang, and Hong-Han Shuai. General then personal: Decoupling and pre-training for personalized headline generation. *Transactions of the Association for Computational Linguistics*, 11:1588–1607, 2023. doi: 10.1162/tacl_a_00621. URL https://aclanthology.org/2023.tacl-1.90/.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, pages 1441–1450, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450369763. doi: 10.1145/3357384.3357895. URL https://doi.org/10.1145/3357384.3357895.

Rahul Vansh, Darsh Rank, Sourish Dasgupta, and Tanmoy Chakraborty. Accuracy is not enough: Evaluating personalization in summarizers. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2582–2595, Singapore, dec 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.169. URL https://aclanthology.org/2023.findings-emnlp.169.

Yancheng Wang, Ziyan Jiang, Zheng Chen, Fan Yang, Yingxue Zhou, Eunah Cho, Xing Fan, Yanbin Lu, Xiaojiang Huang, and Yingzhen Yang. RecMind: Large language model powered agent for recommendation. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4351–4364, Mexico City, Mexico, jun 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.271. URL https://aclanthology.org/2024.findings-naacl.271/.

Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. Neural news recommendation with attentive multi-view learning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 2019a.

Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. Neural news recommendation with multi-head self-attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6389–6394, Hong Kong, China, nov 2019b. Association for Computational Linguistics. doi: 10.18653/v1/D19-1671. URL https://aclanthology.org/D19-1671.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D Manning, and Christopher Potts. ReFT: Representation finetuning for language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=fykjplMc0V.

Wei Xiao, Huiying Wang, Qifeng Zhou, and Qing Wang. Ss4rec: Continuous-time sequential recommendation with state space models, 2025. URL https://arxiv.org/abs/2502.08132.

Sicong Xie, Qunwei Li, Weidi Xu, Kaiming Shen, Shaohu Chen, and Wenliang Zhong. Denoising time cycle modeling for recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, pages 1950–1955, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450387323. doi: 10.1145/3477495.3531785. URL https://doi.org/10.1145/3477495.3531785.

Zhao Yang, Junhong Lian, and Xiang Ao. Fact-preserved personalized news headline generation. *2023 IEEE International Conference on Data Mining (ICDM)*, 2023a.

Zhengyi Yang, Jiancan Wu, Zhicai Wang, Xiang Wang, Yancheng Yuan, and Xiangnan He. Generate what you prefer: Reshaping sequential recommendation via guided diffusion. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023b. URL https://openreview.net/forum?id=oFpBnt6bgC.

Zeping Yu, Jianxun Lian, Ahmad Mahmoody, Gongshen Liu, and Xing Xie. Adaptive user modeling with long and short-term preferences for personalized recommendation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 4213–4219. International Joint Conferences on Artificial Intelligence Organization, 7 2019. doi: 10.24963/ijcai.2019/585. URL https://doi.org/10.24963/ijcai.2019/585.

Jin Zeng, Yupeng Qi, Hui Li, Chengming Li, Ziyu Lyu, Lixin Cui, and Lu Bai. Raie:region-aware incremental preference editing with lora for llm-based recommendation. In *Proceedings of the ACM Web Conference 2026*, WWW '26, page 4038–4046, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400723070. doi: 10.1145/3774904.3792451. URL https://doi.org/10.1145/3774904.3792451.

Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Jiayuan He, Yinghai Lu, and Yu Shi. Actions speak louder than words: Trillion-parameter sequential transducers for generative recommendations. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 58484–58509. PMLR, 2024. URL https://proceedings.mlr.press/v235/zhai24a.html.

Chi Zhang, Yantong Du, Xiangyu Zhao, Qilong Han, Rui Chen, and Li Li. Hierarchical item inconsistency signal learning for sequence denoising in sequential recommendation. In Mohammad Al Hasan and Li Xiong, editors, *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2508–2518. Association for Computing Machinery, 2022. doi: 10.1145/3511808.3557348. URL https://doi.org/10.1145/3511808.3557348.

Chi Zhang, Qilong Han, Rui Chen, Xiangyu Zhao, Peng Tang, and Hongtao Song. Ssdrec: Self-augmented sequence denoising for sequential recommendation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 803–815, 2024a. doi: 10.1109/ICDE60146.2024.00067.

Jinghao Zhang, Yuting Liu, Wenjie Wang, Qiang Liu, Shu Wu, Liang Wang, and Tat-Seng Chua. Personalized text generation with contrastive activation steering. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7128–7141, Vienna, Austria, jul 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.353. URL https://aclanthology.org/2025.acl-long.353/.

Kai Zhang, Yangyang Kang, Fubang Zhao, and Xiaozhong Liu. LLM-based medical assistant personalization with short- and long-term memory coordination. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2386–2398, Mexico City, Mexico, jun 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.132. URL https://aclanthology.org/2024.naacl-long.132/.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. Personalizing dialogue agents: I have a dog, do you have pets too? In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia, jul 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1205. URL https://aclanthology.org/P18-1205/.

A Methodology Details

The main text defines REPAIR as a selective pre-head repair module for frozen personalization encoders. This appendix gives the implementation-level form used in our experiments, while keeping the same intervention contract: the host encoder is frozen, the raw history is not re-encoded, cached timestep representations are reused from the original forward pass, and only the exposed compressed state is corrected before the downstream task head consumes it.

A.1 User Interaction Graph and Eventization

We follow the User Interaction Graph (UIG) abstraction of Chatterjee et al. [2025] to represent heterogeneous user histories as ordered interaction trajectories. For a user u, the graph is

$$\mathcal{G}_u = \langle \mathcal{N}_u, \mathcal{E}_u \rangle.$$

The node set contains user nodes, content nodes, query nodes for generative tasks, predicted response nodes, and gold target nodes when supervision is available. The edge set records user actions. We use normalized action labels

$$a^{(t_i)} \in \{\texttt{pos}, \texttt{neg}, \texttt{cmd}\},$$

where pos denotes positive feedback such as click, consume, accept, purchase, or select; neg denotes negative feedback such as skip, reject, non-click, or low preference; and cmd denotes a task request such as summarize, generate, or recommend.

Each observed interaction is converted into an interaction unit

$$b^{(t_i)} = \langle a^{(t_i)}, v^{(t_i)} \rangle, \qquad v^{(t_i)} \in \mathcal{N}_u, \tag{7}$$

where $v^{(t_i)}$ is the content, query, response, or target node on which the action occurs. For a prefix ending at t_i, the model-side interaction sequence is

$$\tau_b^{t_{1:i}} = \langle b^{(t_1)}, b^{(t_2)}, \ldots, b^{(t_i)} \rangle. \tag{8}$$

For predictive personalization, the target is the next content item, $r_u^{\star(t_{i+1})} \equiv c_u^{(t_{i+1})}$. For generative personalization, the target is a response node, such as a headline, summary, or dialogue response, and the task-side query or source input $q^{(t_i)}$ is attached to the instance. During autoregressive evaluation, predicted outputs are appended to the running trajectory and become part of later history.

A.2 Frozen Host Interface

For each prefix, the frozen host encoder exposes two objects to REPAIR:

$$z_b^{(t_i)} \in \mathbb{R}^{d_z}, \qquad \{e_b^{(t_j)}\}_{j=1}^{i}, \quad e_b^{(t_j)} \in \mathbb{R}^{d_e}.$$

Here, $z_b^{(t_i)}$ is the compressed preference state consumed by the downstream head, and $e_b^{(t_j)}$ is the cached timestep representation that was already available during the host forward pass. These event representations are not produced by a second encoder.

In implementation, an interaction unit is represented by combining an action embedding with the corresponding content representation. For text-centric histories, the content representation is obtained from a T5-based encoder; for multimodal product histories, CLIP-style image features are additionally included when available. Host-specific caches are chosen as the closest exposed per-event representation before final history compression: pre-recurrent inputs for recurrent hosts, item/news/content vectors before terminal pooling for self-attentive hosts, per-step inputs or cached intermediates for state-space hosts, and fused product vectors for multimodal hosts. The maximum prefix length is bounded by the host truncation window, so the event cache costs $O(Ld_e)$ per active trajectory and is discarded after the autoregressive rollout step.

A.3 State-relative Corrective Evidence

The compressed state and cached event representations are first mapped into a shared repair space:

$$\bar{z}_b^{(t_i)} = P_z z_b^{(t_i)}, \qquad \bar{e}_b^{(t_j)} = P_e e_b^{(t_j)}, \tag{9}$$

where $P_z \in \mathbb{R}^{m \times d_z}$ and $P_e \in \mathbb{R}^{m \times d_e}$. The unreduced state-relative residual is

$$r^{(t_j;t_i)} = \bar{e}_b^{(t_j)} - \bar{z}_b^{(t_i)}. \tag{10}$$

A low-rank repair operator is parameterized as

$$J_{\mathrm{rep}} = U_{\mathrm{rep}} \Lambda_{\mathrm{rep}} V_{\mathrm{rep}}^{\top}, \qquad U_{\mathrm{rep}}, V_{\mathrm{rep}} \in \mathbb{R}^{m \times K}, \tag{11}$$

with diagonal Λ_{rep}. The read side gives the corrective coordinate

$$d^{(t_j)} = V_{\mathrm{rep}}^{\top} r^{(t_j;t_i)} = V_{\mathrm{rep}}^{\top} \left(P_e e_b^{(t_j)} - P_z z_b^{(t_i)} \right). \tag{12}$$

Thus $d^{(t_j)}$ measures what timestep t_j still contributes relative to what the exposed terminal state already expresses.

We remove prefix-level residual offset by centering the residual field:

$$\tilde{d}^{(t_j)} = d^{(t_j)} - \frac{1}{i} \sum_{q=1}^{i} d^{(t_q)}. \tag{13}$$

This makes correction depend on timestep-specific evidence within the current prefix rather than on an absolute event-state distance.

A.4 Multi-regime Temporal Resolution

Let $\ell_j = i - j$ be the lag of timestep t_j. REPAIR applies three coordinate-wise temporal responses:

$$\begin{aligned} \kappa^L(\ell_j) &= \exp(-\beta \ell_j), \\ \kappa^S(\ell_j) &= \max\left(\mathbf{1}_K - \frac{\ell_j}{i} \mathbf{1}_K, \mathbf{0}_K \right), \\ \kappa^E(\ell_j) &= \exp(\ell_j \log \xi - \xi - \log \Gamma(\ell_j + 1) \mathbf{1}_K), \end{aligned} \tag{14}$$

where $\beta, \xi \in \mathbb{R}_+^K$ are learned through a positive transform. The long-support response keeps older residual evidence active, the short-support response emphasizes tail-local drift, and the episodic response gives localized support to burst-like residual evidence. These are soft temporal bases over the residual field, not hard semantic labels over raw actions.

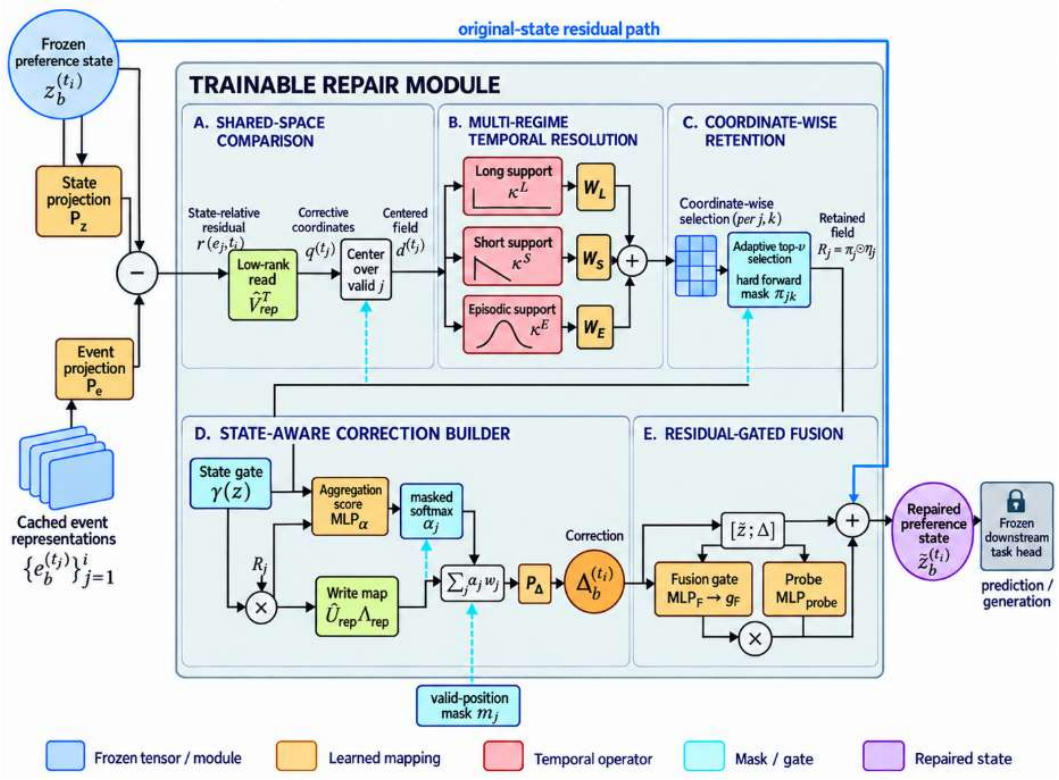

Figure 2: **Detailed architecture of** `REPAIR`**.** The frozen state and cached timestep evidence are projected into the shared repair space, followed by low-rank corrective reading, centering, temporal resolution, selective retention, aggregation, and residual-gated fusion. Dashed arrows denote conditioning paths.

The regime-resolved evidence is

$$\eta^{(t_j)} = \sum_{\rho \in \{L,S,E\}} W_\rho \left(\kappa^\rho(\ell_j) \odot \tilde{d}^{(t_j)} \right), \qquad W_\rho \in \mathbb{R}^{K \times K}. \tag{15}$$

Since the three branches are summed, medium-scale, repeated, or mixed temporal traces can be represented as combinations of long-support, short-support, and episodic corrective pathways.

A.5 Adaptive Top-ν Retention

The main text writes selective retention as

$$\Pi_\nu^{(t_i)}(\eta^{(t_j)}) = \pi_j^{(t_i)} \odot \eta^{(t_j)}, \qquad \pi_j^{(t_i)} \in \{0,1\}^K.$$

In implementation, retention is coordinate-wise. For each corrective coordinate k, we summarize the magnitude sequence $\{|\eta_k^{(t_q)}|\}_{q=1}^i$ by its mean, standard deviation, and maximum:

$$s_k^{(t_i)} = \left[\frac{1}{i} \sum_{q=1}^{i} |\eta_k^{(t_q)}|; \sqrt{\frac{1}{i} \sum_{q=1}^{i} \left(\eta_k^{(t_q)} - \frac{1}{i} \sum_{r=1}^{i} \eta_k^{(t_r)} \right)^2}; \max_{1 \le q \le i} |\eta_k^{(t_q)}| \right]. \tag{16}$$

The selector predicts a retained count

$$\nu_k^{(t_i)} = \mathrm{clip}\left(\mathrm{round}\left(i \cdot \sigma\left(\mathrm{MLP}_{\mathrm{sel}}(s_k^{(t_i)}) \right) \right), 1, i \right). \tag{17}$$

A straight-through top-$\nu_k^{(t_i)}$ mask is then applied over $\{|\eta_k^{(t_q)}|\}_{q=1}^i$. This yields

$$\Pi_\nu^{(t_i)}\left(\eta^{(t_j)} \right) = \left[\pi_{j,1}^{(t_i)} \eta_1^{(t_j)}, \ldots, \pi_{j,K}^{(t_i)} \eta_K^{(t_j)} \right], \qquad \pi_{j,k}^{(t_i)} \in \{0,1\}. \tag{18}$$

Selection is therefore sparse at the timestep–coordinate level. A timestep can be retained for one corrective mode and ignored for another. The integer $\nu_k^{(t_i)}$ is used only to instantiate the forward retention budget. We do not require an unbiased gradient through the sorting or rounding operation. Gradients update the selector through the continuous pre-mask scores and through the retained evidence path, while the discrete cardinality acts as a budgeted routing decision under a straight-through implementation.

A.6 Correction Aggregation and Fusion

After selection, the current compressed state produces a corrective-mode gate:

$$\gamma^{(t_i)} = \sigma\Big(\mathrm{MLP}_\gamma(z_b^{(t_i)})\Big), \qquad \gamma^{(t_i)} \in (0,1)^K. \tag{19}$$

The retained evidence is aggregated by attention:

$$\alpha_j^{(t_i)} = \frac{\exp\Big(w_\alpha^\top \Big[\Pi_\nu^{(t_i)}(\eta^{(t_j)}); \gamma^{(t_i)}\Big]\Big)}{\sum_{q=1}^{i} \exp\Big(w_\alpha^\top \Big[\Pi_\nu^{(t_i)}(\eta^{(t_q)}); \gamma^{(t_i)}\Big]\Big)}. \tag{20}$$

The synthesized state correction is

$$\Delta_b^{(t_i)} = P_\Delta \left(\sum_{j=1}^{i} \alpha_j^{(t_i)} U_{\mathrm{rep}} \Lambda_{\mathrm{rep}} \left(\gamma^{(t_i)} \odot \Pi_\nu^{(t_i)}(\eta^{(t_j)}) \right) \right), \qquad P_\Delta \in \mathbb{R}^{d_z \times m}. \tag{21}$$

The repaired state is produced by residual-gated fusion:

$$\begin{aligned} g_F^{(t_i)} &= \sigma\Big(\mathrm{MLP}_F\Big([z_b^{(t_i)}; \Delta_b^{(t_i)}]\Big)\Big), \\ \tilde{z}_b^{(t_i)} &= z_b^{(t_i)} + g_F^{(t_i)} \odot \mathrm{MLP}_{\mathrm{probe}}\Big([z_b^{(t_i)}; \Delta_b^{(t_i)}]\Big). \end{aligned} \tag{22}$$

This form preserves the frozen host state and only adds a learned, state-conditioned residual correction. A detailed architectural diagram is given in Figure 2.

A.7 Training Objective

Let $x_{\mathrm{task}}^{(t_i)}$ denote the task-side input. It is empty for predictive tasks and corresponds to the source document, query, or dialogue context for generative tasks. The downstream task loss is

$$\mathcal{L}_{\mathrm{task}} = \ell_{\mathrm{task}}\Big(h_{\mathrm{task}}\Big(\tilde{z}_b^{(t_i)}, x_{\mathrm{task}}^{(t_i)}\Big), r_u^{\star(t_{i+1})}\Big). \tag{23}$$

This term trains repair for end-task utility rather than for reconstructing the original compressed state.

The positional anchoring loss predicts the source timestep of each regime-resolved vector. With maximum prefix length H and mask μ_i for invalid positions,

$$\begin{aligned} \hat{p}_{\mathrm{pos}}^{(t_j)} &= \mathrm{softmax}\Big(W_{\mathrm{pos}} \eta^{(t_j)} + \mu_i\Big), \\ \mathcal{L}_{\mathrm{pos}} &= -\frac{1}{i} \sum_{j=1}^{i} \log \hat{p}_{\mathrm{pos},j}^{(t_j)}, \end{aligned} \tag{24}$$

where $W_{\mathrm{pos}} \in \mathbb{R}^{H \times K}$. This prevents temporally collapsed correction fields without forcing the downstream head to use exact-position labels.

For local coherence, event representations are projected into corrective coordinates:

$$\bar{e}_{\mathrm{coh}}^{(t_j)} = P_{\mathrm{coh}} e_b^{(t_j)}, \qquad P_{\mathrm{coh}} \in \mathbb{R}^{K \times d_e}. \tag{25}$$

The coherence loss uses the source event as the positive and other prefix events as prefix-local negatives:

$$\mathcal{L}_{\mathrm{coh}} = -\frac{1}{i} \sum_{j=1}^{i} \log \frac{\exp\Big(\mathrm{sim}\Big(\eta^{(t_j)}, \bar{e}_{\mathrm{coh}}^{(t_j)}\Big)/\tau\Big)}{\sum_{q=1}^{i} \exp\Big(\mathrm{sim}\Big(\eta^{(t_j)}, \bar{e}_{\mathrm{coh}}^{(t_q)}\Big)/\tau\Big)}. \tag{26}$$

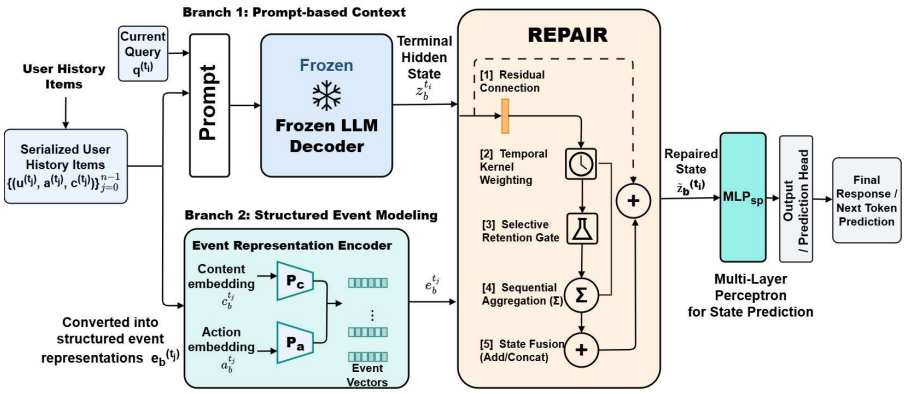

Figure 3: Prompt-conditioned REPAIR interface for frozen LLM hosts.

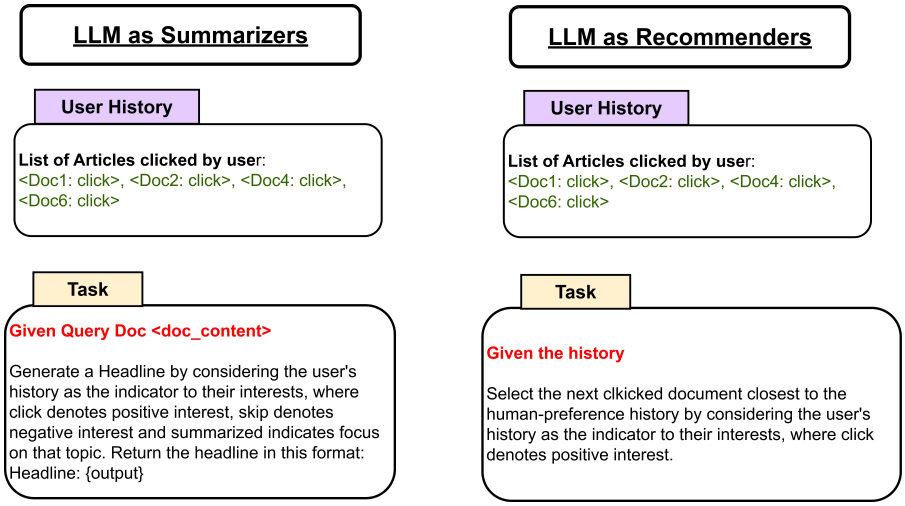

Figure 4: Prompt Templates Used For LLM-Based Baselines Across Two Tasks.

The negatives need not be semantically unrelated in raw content space; the loss only asks the correction induced by timestep t_j to remain closest to its own source event after projection.

The final objective is

$$\mathcal{L}_{\text{repair}} = \lambda_{\text{task}}\mathcal{L}_{\text{task}} + \lambda_{\text{pos}}\mathcal{L}_{\text{pos}} + \lambda_{\text{coh}}\mathcal{L}_{\text{coh}}, \qquad (\lambda_{\text{task}}, \lambda_{\text{pos}}, \lambda_{\text{coh}}) = (0.5, 0.2, 0.3). \tag{27}$$

Thus $\lambda_{\text{task}} > \lambda_{\text{coh}} > \lambda_{\text{pos}}$: task utility remains primary, event-level coherence is the stronger auxiliary constraint, and positional anchoring is kept positive to prevent timestep ambiguity.

A.8 LLM Prompt-conditioned Hosts

For LLM hosts, REPAIR follows the same state-repair principle without modifying the frozen decoder backbone. The user history is serialized into the textual prompt:

$$x_{\text{task}}^{(t_i)} = \text{Prompt}\left(\tau_b^{t_{1:i}}, q^{(t_i)}\right), \qquad z_b^{(t_i)} = \phi_{\text{LLM}}^{\text{last}}\left(x_{\text{task}}^{(t_i)}\right), \tag{28}$$

where $z_b^{(t_i)}$ is the prompt-conditioned terminal hidden state of the frozen LLM.

The event representations used by REPAIR are constructed from the same serialized history in structured form:

$$e_b^{(t_j)} = P_c c_b^{(t_j)} + P_a a_b^{(t_j)}, \qquad j \le i, \tag{29}$$

where $c_b^{(t_j)}$ is the content embedding of the history item and $a_b^{(t_j)}$ is its action-label embedding. Given $z_b^{(t_i)}$ and $\{e_b^{(t_j)}\}_{j=1}^{i}$, the same residual, temporal-resolution, selection, aggregation, and fusion operations produce $\tilde{z}_b^{(t_i)}$.

For prompt-conditioned decoding, the repaired state is mapped to a small continuous prefix:

$$S_b^{(t_i)} = \mathrm{MLP}_{\mathrm{sp}}\left(\tilde{z}_b^{(t_i)}\right) \in \mathbb{R}^{M \times d_{\mathrm{LLM}}}. \tag{30}$$

The soft-prompt embeddings $S_b^{(t_i)}$ are prepended to the token embeddings of the original textual prompt. The LLM therefore receives explicit serialized history through $x_{\mathrm{task}}^{(t_i)}$ and repaired preference evidence through $S_b^{(t_i)}$. The LLM backbone remains frozen; only REPAIR and the soft-prompt projection are trained. For LLM-conditioned generation, REPAIR does not modify the LLM backbone and does not re-run a history encoder over the raw user trajectory. The repaired state is mapped to a small continuous prefix that is concatenated with the decoder input embeddings for the current generation instance. We do not assume that a KV cache computed for the original textual prompt remains valid after inserting this prefix; latency measurements for the LLM-conditioned setting include the cost of prefix-conditioned decoding. Therefore, the "no second pass" claim refers to avoiding a second trajectory/history encoding pass, not to reusing an unchanged decoder KV cache after prefix insertion. Architectural details are depicted in Figure 3.

A.9 Statistical Significance Testing

We evaluate repeated-run and test-set uncertainty for three representative settings: OpenCLIP–Amazon, SigLIP2–Amazon, and EBNR–PENS. Each setting is trained with three independent random seeds. We report the mean and standard deviation across these runs, while the remaining extended host settings are reported as point estimates.

To account for multiple prefixes contributed by the same held-out user, test-set uncertainty is estimated with user-clustered bootstrap resampling. Let $\mathcal{U}_{\mathrm{test}} = \{u_n\}_{n=1}^{N_u}$ denote the held-out user set. For each bootstrap trial, users are sampled with replacement and all test instances associated with each sampled user are retained. We use $B = 10{,}000$ bootstrap resamples.

For a model A and metric M, let $M(A; \mathcal{S}_b)$ denote the metric computed on bootstrap sample $\mathcal{S}_b$. For a paired comparison between REPAIR and comparator A, the same sampled users, history prefixes, targets, and, for recommendation, candidate sets are used for both systems. The paired difference is

$$\Delta_b = M(\texttt{REPAIR}; \mathcal{S}_b) - M(A; \mathcal{S}_b), \qquad b = 1, \ldots, B.$$

The bootstrap estimate is

$$\overline{\Delta} = \frac{1}{B} \sum_{b=1}^{B} \Delta_b, \qquad B = 10{,}000,$$

and the 2.5th and 97.5th percentiles of $\{\Delta_b\}_{b=1}^{B}$ define the 95% user-clustered bootstrap confidence interval. A paired improvement is treated as statistically supported when this interval excludes zero on the non-positive side.

The three-seed standard deviation captures variation across independent training runs, whereas the clustered bootstrap captures uncertainty induced by held-out-user sampling while preserving dependence among prefixes from the same user. Accordingly, statistical claims are restricted to the three settings evaluated under this protocol; the broader host sweep is interpreted from point estimates under the common evaluation setup.

B Theoretical Justification

B.1 Admissible Low-Rank Repair Dimension

REPAIR reads state-relative evidence through

$$d^{(t_j)} = V_{\mathrm{rep}}^{\top}\left(P_e e_b^{(t_j)} - P_z z_b^{(t_i)}\right), \qquad V_{\mathrm{rep}} \in \mathbb{R}^{m \times K},$$

Table 7: Symbols and notations.

Symbol	Meaning	Symbol	Meaning
Interaction and host interface			
$\mathcal{G}_u = \langle \mathcal{N}_u, \mathcal{E}_u \rangle$	User interaction graph	$\mathcal{N}_u$	User-specific typed node set
$\mathcal{E}_u$	Action-labeled edge set	$u^{(t_0)}$	Initial user node
$v^{(t_i)}$	Content or response node	$a^{(t_i)}$	Observed action label
$\{\text{pos}, \text{neg}, \text{cmd}\}$	Normalized action types	$b^{(t_i)} = \langle a^{(t_i)}, v^{(t_i)} \rangle$	Interaction unit at timestep
$\tau_b^{t_{1:i}}$	Model-side prefix trajectory	t_i	Current prefix endpoint
t_j	Earlier prefix timestep	$\ell_j = i - j$	Lag from current timestep
f_{enc}	Frozen personalization encoder	$\mathbf{e}_b^{(t_j)}$	Cached event representation
$\mathbf{z}_b^{(t_i)}$	Exposed compressed preference state	$\mathbf{q}^{(t_i)}$	Task-side query input
$r_u^{\star(t_{i+1})}$	Next task target	h_{task}	Prediction or generation head
Core dimensions and projections			
d_e	Event representation dimension	d_z	Preference-state dimension
m	Shared repair-space dimension	K	Low-rank corrective dimension
H	Maximum prefix length	L	Host truncation window
P_z	Shared state projection	P_e	Shared event projection
$\bar{\mathbf{z}}_b^{(t_i)} = P_z \mathbf{z}_b^{(t_i)}$	Projected host state	$\bar{\mathbf{e}}_b^{(t_j)} = P_e \mathbf{e}_b^{(t_j)}$	Projected event representation
State-relative corrective evidence			
$\mathbf{r}^{(t_j;t_i)}$	Unreduced state-relative residual	D_{rep}	Corrective evidence reader
$\mathbf{d}^{(t_j)}$	Low-rank corrective coordinate	$\tilde{\mathbf{d}}^{(t_j)}$	Centered corrective coordinate
$\mathcal{D}_i = \{\mathbf{d}^{(t_j)}\}_{j=1}^{i}$	Corrective residual field	J_{rep}	Low-rank repair operator
V_{rep}	Residual read basis	U_{rep}	Residual write basis
Λ_{rep}	Diagonal repair spectrum		
Temporal resolution			
$\rho \in \{L, S, E\}$	Temporal regime index	$\kappa^L(\ell_j)$	Long-support temporal kernel
$\kappa^S(\ell_j)$	Short-support temporal kernel	$\kappa^E(\ell_j)$	Episodic-burst temporal kernel
β	Learned long-decay vector	ξ	Learned episodic-rate vector
W_L, W_S, W_E	Regime-resolution matrices	G_{rep}	Multi-regime resolution operator
$\boldsymbol{\eta}^{(t_j)}$	Regime-resolved evidence vector		
Selective retention			
$\Pi_\nu^{(t_i)}$	Top-ν retention operator	$\boldsymbol{\pi}_j^{(t_i)}$	Timestep selector mask
$\pi_{j,k}^{(t_i)}$	Timestep-coordinate mask entry	$\nu_k^{(t_i)}$	Retained count per coordinate
$\mathbf{s}_k^{(t_i)}$	Selector summary statistics	MLP_{sel}	Adaptive selector network
Aggregation and fusion			
$\boldsymbol{\gamma}^{(t_i)}$	State-conditioned coordinate gate	$\alpha_j^{(t_i)}$	Retained-timestep attention weight
A_{rep}	Correction aggregation operator	P_Δ	Correction-to-state projection
$\boldsymbol{\Delta}_b^{(t_i)}$	State-space correction vector	F_{rep}	Residual repair fusion operator
$\mathbf{g}_F^{(t_i)}$	Residual fusion gate	MLP_F	Fusion gate network
$\text{MLP}_{\text{probe}}$	Probe residual network	$\tilde{\mathbf{z}}_b^{(t_i)}$	Repaired preference state
Training objective			
$\mathcal{L}_{\text{task}}$	Downstream task loss	$\mathcal{L}_{\text{pos}}$	Position anchoring loss
$\mathcal{L}_{\text{coh}}$	Event coherence loss	$\mathcal{L}_{\text{repair}}$	Total repair objective
λ_{task}	Task-loss weight	λ_{pos}	Position-loss weight
λ_{coh}	Coherence-loss weight	W_{pos}	Position classifier matrix
P_{coh}	Coherence projection matrix	τ	Contrastive temperature

and writes retained coordinates back through $P_\Delta U_{\text{rep}} \Lambda_{\text{rep}}$. For a prefix ending at t_i, define the unreduced residual

$$r^{(t_j;t_i)} = P_e e_b^{(t_j)} - P_z z_b^{(t_i)} \in \mathbb{R}^m.$$

Let r be drawn from the residual distribution over prefix–timestep pairs and let

$$C = \mathbb{E}[rr^\top] = \sum_{q=1}^{m} \lambda_q u_q u_q^\top, \qquad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m \geq 0.$$

For $K \le m$, define

$$\mathcal{P}_K = \sum_{q=1}^{K} u_q u_q^\top, \qquad T_K = \sum_{q>K} \lambda_q.$$

Definition B.1 (Admissible repair dimension)**.** Let $A : \mathbb{R}^m \to \mathbb{R}^{d_z}$ be a bounded residual-to-state correction map with

$$\|A\|_2 \le B_A. \qquad \text{i.e., } A \text{ cannot amplify residual errors arbitrarily.}$$

Let $\mathcal{B}(K)$ denote the cost of using K repair coordinates. Since the low-rank read/write maps scale monotonically with K, write locally

$$\mathcal{B}(K) = aK + b, \qquad a, b > 0.$$

For tolerance $\varepsilon_{\text{app}} > 0$ and budget $B_{\max}$, K is admissible if

$$B_A^2 T_K \le \varepsilon_{\text{app}}^2, \qquad \mathcal{B}(K) \le B_{\max}.$$

Equivalently,

$$K \in [K_{\min}, K_{\max}],$$

where

$$K_{\min} = \min\left\{ k : B_A^2 \sum_{q>k} \lambda_q \le \varepsilon_{\text{app}}^2 \right\}, \qquad K_{\max} = \left\lfloor \frac{B_{\max} - b}{a} \right\rfloor.$$

Theorem 1 (Admissible compact repair space)**.** *For any bounded residual-to-state correction map A with $\|A\|_2 \le B_A$,*

$$\mathbb{E}\left[\|Ar - A\mathcal{P}_K r\|_2^2 \right] \le B_A^2 T_K.$$

Thus every $K \in [K_{\min}, K_{\max}]$ satisfies the approximation tolerance and the budget constraint. In particular, if

$$K_{\min} \le 128 \le K_{\max},$$

then $K = 128$ is admissible. Moreover, for any $\tau > 0$,

$$\Pr\left(\|Ar - A\mathcal{P}_K r\|_2 \ge \tau \right) \le \frac{B_A^2 T_K}{\tau^2}.$$

Proof. Since $Ar - A\mathcal{P}_K r = A(I - \mathcal{P}_K)r$,

$$\|A(I - \mathcal{P}_K)r\|_2^2 \le \|A\|_2^2 \, \|(I - \mathcal{P}_K)r\|_2^2 \le B_A^2 \, \|(I - \mathcal{P}_K)r\|_2^2 .$$

Taking expectation gives

$$\mathbb{E}\left[\|Ar - A\mathcal{P}_K r\|_2^2 \right] \le B_A^2 \mathbb{E}\left[\|(I - \mathcal{P}_K)r\|_2^2 \right].$$

Now

$$\mathbb{E}\left[\|(I - \mathcal{P}_K)r\|_2^2 \right] = \mathbb{E}\left[r^\top (I - \mathcal{P}_K) r \right] = \operatorname{tr}\left((I - \mathcal{P}_K) C \right).$$

Substituting the eigendecomposition of C,

$$\operatorname{tr}\left((I - \mathcal{P}_K) C \right) = \sum_{q>K} \lambda_q = T_K.$$

Therefore,

$$\mathbb{E}\left[\|Ar - A\mathcal{P}_K r\|_2^2 \right] \le B_A^2 T_K.$$

If $K \ge K_{\min}$, then $B_A^2 T_K \le \varepsilon_{\text{app}}^2$. If $K \le K_{\max}$, then $\mathcal{B}(K) \le B_{\max}$. Hence every $K \in [K_{\min}, K_{\max}]$ is admissible.

For the probability bound, define

$$X = \|Ar - A\mathcal{P}_K r\|_2^2 .$$

Since $X \ge 0$ and $\mathbb{E}[X] \le B_A^2 T_K$, Markov's inequality gives

$$\Pr(X \ge \tau^2) \le \frac{\mathbb{E}[X]}{\tau^2} \le \frac{B_A^2 T_K}{\tau^2}.$$

This is the claimed high-probability bound. □

Interpretation. The result states that any K satisfying the spectral-tail and budget constraints is admissible. Since T_K is nonincreasing in K, larger K improves approximation but increases cost. Smaller K is preferable only inside the admissible interval, where it preserves the same tolerance while reducing compact-space cost. For the host families used in our experiments, the validation residual spectra and deployment budget place $K = 128$ inside this admissible interval. We therefore fix $K = 128$ across hosts rather than tuning a host-specific repair dimension.

B.2 Recoverability of State-Relative Correction

Theorem 2 (Recoverability of Low-Rank State-Relative REPAIR)**.** *Fix a trajectory prefix $\tau_b^{t_1:i}$ and frozen host state $z_b^{(t_i)}$. Let*

$$\Delta_b^{\star(t_i)} = z_b^{\star(t_i)} - z_b^{(t_i)}$$

be an oracle correction to an ideal task-relevant state. Suppose there exist coefficients

$$c^{(t_j)} \in \mathbb{R}^K, \qquad j = 1, \dots, i,$$

and convex weights

$$\alpha_j^{\star(t_i)} \geq 0, \qquad \sum_{j=1}^{i} \alpha_j^{\star(t_i)} = 1,$$

such that

$$\left\| \Delta_b^{\star(t_i)} - P_\Delta \left(\sum_{j=1}^{i} \alpha_j^{\star(t_i)} U_{\mathrm{rep}} \Lambda_{\mathrm{rep}} c^{(t_j)} \right) \right\|_2 \leq \varepsilon_{\mathrm{span}}.$$

Assume further that the retained REPAIR coordinates approximate these oracle coefficients:

$$\left\| c^{(t_j)} - \gamma^{(t_i)} \odot \Pi_\nu^{(t_i)} \left(\eta^{(t_j)} \right) \right\|_2 \leq \varepsilon_j.$$

Then REPAIR contains a correction

$$\Delta_b^{(t_i)} = P_\Delta \left(\sum_{j=1}^{i} \alpha_j^{\star(t_i)} U_{\mathrm{rep}} \Lambda_{\mathrm{rep}} \left(\gamma^{(t_i)} \odot \Pi_\nu^{(t_i)} \left(\eta^{(t_j)} \right) \right) \right)$$

satisfying

$$\left\| \Delta_b^{(t_i)} - \Delta_b^{\star(t_i)} \right\|_2 \leq \varepsilon_{\mathrm{span}} + \|P_\Delta\|_2 \|U_{\mathrm{rep}}\|_2 \|\Lambda_{\mathrm{rep}}\|_2 \sum_{j=1}^{i} \alpha_j^{\star(t_i)} \varepsilon_j.$$

If $\varepsilon_j \leq \varepsilon_{\mathrm{coord}}$ for all j, then

$$\left\| \Delta_b^{(t_i)} - \Delta_b^{\star(t_i)} \right\|_2 \leq \varepsilon_{\mathrm{span}} + \|P_\Delta\|_2 \|U_{\mathrm{rep}}\|_2 \|\Lambda_{\mathrm{rep}}\|_2 \varepsilon_{\mathrm{coord}}.$$

Proof. Define the best low-rank oracle correction under the coefficient field as

$$\bar{\Delta}_b^{\star(t_i)} = P_\Delta \left(\sum_{j=1}^{i} \alpha_j^{\star(t_i)} U_{\mathrm{rep}} \Lambda_{\mathrm{rep}} c^{(t_j)} \right).$$

By assumption,

$$\left\| \Delta_b^{\star(t_i)} - \bar{\Delta}_b^{\star(t_i)} \right\|_2 \leq \varepsilon_{\mathrm{span}}.$$

Using the REPAIR-realizable correction defined in the theorem and applying the triangle inequality,

$$\left\| \Delta_b^{(t_i)} - \Delta_b^{\star(t_i)} \right\|_2 \leq \left\| \Delta_b^{(t_i)} - \bar{\Delta}_b^{\star(t_i)} \right\|_2 + \varepsilon_{\mathrm{span}}.$$

For the first term,

$$\left\|\Delta_b^{(t_i)} - \bar{\Delta}_b^{\star(t_i)}\right\|_2 = \left\| P_\Delta \sum_{j=1}^{i} \alpha_j^{\star(t_i)} U_{\text{rep}} \Lambda_{\text{rep}} \left[\gamma^{(t_i)} \odot \Pi_\nu^{(t_i)}\left(\eta^{(t_j)}\right) - c^{(t_j)}\right] \right\|_2$$

$$\leq \|P_\Delta\|_2 \|U_{\text{rep}}\|_2 \|\Lambda_{\text{rep}}\|_2 \sum_{j=1}^{i} \alpha_j^{\star(t_i)} \left\|\gamma^{(t_i)} \odot \Pi_\nu^{(t_i)}\left(\eta^{(t_j)}\right) - c^{(t_j)}\right\|_2$$

$$\leq \|P_\Delta\|_2 \|U_{\text{rep}}\|_2 \|\Lambda_{\text{rep}}\|_2 \sum_{j=1}^{i} \alpha_j^{\star(t_i)} \varepsilon_j.$$

Combining both bounds proves the first claim. If $\varepsilon_j \leq \varepsilon_{\text{coord}}$, then

$$\sum_{j=1}^{i} \alpha_j^{\star(t_i)} \varepsilon_j \leq \varepsilon_{\text{coord}} \sum_{j=1}^{i} \alpha_j^{\star(t_i)} = \varepsilon_{\text{coord}}.$$

This gives the second claim. □

Interpretation. Theorem 2 connects the admissible low-rank dimension in Theorem 1 to the actual objective of REPAIR: correcting a frozen host state toward an ideal task-relevant state. Theorem 1 first establishes that the residual field can be safely compressed when its discarded spectral tail is small, so the repair module does not need to operate in the full residual space. The present theorem then shows what this compact space must achieve: the oracle correction $\Delta_b^{\star(t_i)}$ should be approximately representable by a weighted combination of timestep-level repair coordinates, and REPAIR should recover those coordinates through selective retention and gating. The resulting error bound separates two sources of failure. The span error $\varepsilon_{\text{span}}$ measures whether the compact repair write-back space is expressive enough to contain the ideal correction. The coordinate errors ε_j measure whether the learned retained coordinates match the oracle timestep evidence. Thus, REPAIR succeeds when the missing preference evidence is both compactly representable and recoverable from cached timestep representations. This directly matches the problem setting: frozen personalization encoders compress evolving histories into a terminal state, but task-relevant long-term, short-term, or episodic evidence may remain under-expressed. Rather than finetuning the host encoder or re-encoding the full raw history, REPAIR learns a compact state-relative correction from residual evidence and writes it back before the task head consumes the state. The theorem therefore justifies REPAIR as a recoverability mechanism, not merely an added-capacity adapter: if the ideal correction lies near the low-rank repair span and the retained residual coordinates approximate the oracle coordinates, then the repaired state remains close to the ideal task-relevant state.

B.3 Recoverability of Temporal Misalignment Traces

Definition B.2 (Temporal trace classes)**.** For each coordinate $k \in \{1, \ldots, K\}$, define the centered trace

$$\tilde{\mathbf{d}}_k = \left[\tilde{d}_k^{(t_1)}, \ldots, \tilde{d}_k^{(t_i)}\right]^\top \in \mathbb{R}^i,$$

where

$$\tilde{d}^{(t_j)} = d^{(t_j)} - \frac{1}{i} \sum_{q=1}^{i} d^{(t_q)}.$$

Let $\ell_j = i - j$. Define normalized temporal atoms

$$\psi_{L,k} = \frac{\kappa_k^L}{\|\kappa_k^L\|_2}, \qquad \psi_{S,k} = \frac{\kappa_k^S}{\|\kappa_k^S\|_2}, \qquad \psi_{E,k} = \frac{\kappa_k^E}{\|\kappa_k^E\|_2},$$

with

$$\kappa_{k,j}^L = \exp(-\beta_k \ell_j), \qquad \kappa_{k,j}^S = \max\left(1 - \frac{\ell_j}{i}, 0\right),$$

and

$$\kappa_{k,j}^E = \exp\left(\ell_j \log \xi_k - \xi_k - \log \Gamma(\ell_j + 1)\right).$$

A mixed temporal misalignment trace is any trace of the form

$$\tilde{\mathbf{d}}_k = a_{L,k}\psi_{L,k} + a_{S,k}\psi_{S,k} + a_{E,k}\psi_{E,k} + \varepsilon_k.$$

Theorem 3 (Recoverability of temporal traces by REPAIR kernels)**.** *Let*

$$\Psi_k = [\psi_{L,k}, \psi_{S,k}, \psi_{E,k}] \in \mathbb{R}^{i\times 3}, \qquad G_k = \Psi_k^\top \Psi_k.$$

Assume $\lambda_{\min}(G_k) > 0$*. If*

$$\tilde{\mathbf{d}}_k = \Psi_k a_k + \varepsilon_k, \qquad a_k = [a_{L,k}, a_{S,k}, a_{E,k}]^\top,$$

and

$$r_k = \Psi_k^\top \tilde{\mathbf{d}}_k, \qquad \hat{a}_k = G_k^{-1} r_k,$$

then

$$\|\hat{a}_k - a_k\|_2 \le \frac{\sqrt{\lambda_{\max}(G_k)}}{\lambda_{\min}(G_k)} \|\varepsilon_k\|_2.$$

If $\varepsilon_k = 0$*, then* $\hat{a}_k = a_k$*.*

Proof. Using $\tilde{\mathbf{d}}_k = \Psi_k a_k + \varepsilon_k$,

$$r_k = \Psi_k^\top \tilde{\mathbf{d}}_k = \Psi_k^\top \Psi_k a_k + \Psi_k^\top \varepsilon_k = G_k a_k + \Psi_k^\top \varepsilon_k.$$

Since $\lambda_{\min}(G_k) > 0$, G_k is invertible, so

$$\hat{a}_k = G_k^{-1} r_k = a_k + G_k^{-1} \Psi_k^\top \varepsilon_k.$$

Thus

$$\hat{a}_k - a_k = G_k^{-1} \Psi_k^\top \varepsilon_k.$$

Taking norms,

$$\|\hat{a}_k - a_k\|_2 \le \|G_k^{-1}\|_2 \|\Psi_k^\top\|_2 \|\varepsilon_k\|_2.$$

Since G_k is symmetric positive definite,

$$\|G_k^{-1}\|_2 = \frac{1}{\lambda_{\min}(G_k)}.$$

Also,

$$\|\Psi_k^\top\|_2^2 = \lambda_{\max}(\Psi_k^\top \Psi_k) = \lambda_{\max}(G_k).$$

Therefore,

$$\|\hat{a}_k - a_k\|_2 \le \frac{\sqrt{\lambda_{\max}(G_k)}}{\lambda_{\min}(G_k)} \|\varepsilon_k\|_2.$$

If $\varepsilon_k = 0$, then $\hat{a}_k - a_k = 0$. □

Corollary B.1 (Coordinate-wise recovery of the full field)**.** *If the assumptions of Theorem 3 hold for every coordinate* k*, then the coefficient matrix*

$$A = \begin{bmatrix} a_{L,1} & \cdots & a_{L,K} \\ a_{S,1} & \cdots & a_{S,K} \\ a_{E,1} & \cdots & a_{E,K} \end{bmatrix}$$

is recoverable coordinate-wise, and

$$\sum_{k=1}^{K} \|\hat{a}_k - a_k\|_2^2 \le \sum_{k=1}^{K} \frac{\lambda_{\max}(G_k)}{\lambda_{\min}(G_k)^2} \|\varepsilon_k\|_2^2.$$

Proof. Apply Theorem 3 independently to every coordinate k, square the resulting inequality, and sum over $k = 1, \ldots, K$. □

Corollary B.2 (Fixed short-regime envelope)**.** *Let* $\psi_{S,k}$ *be induced by the fixed short-regime kernel*

$$\kappa^S(\ell_j) = \max\left(\mathbf{1}_K - \frac{\ell_j}{i}\mathbf{1}_K, \mathbf{0}_K\right).$$

If

$$\Psi_k = [\psi_{L,k}, \psi_{S,k}, \psi_{E,k}] \quad \textit{satisfies} \quad \lambda_{\min}(G_k) > 0,$$

then the short-regime coefficient $a_{S,k}$ *is identifiable with the same error bound as in Theorem 3. Therefore, recoverability does not require a separate learnable decay parameter inside* κ^S*; the fixed short-regime envelope is sufficient whenever it remains distinguishable from the long-term and episodic atoms.*

Proof. This follows directly from Theorem 3. The proof only requires that the column $\psi_{S,k}$ participates in a full-rank temporal dictionary Ψ_k. It does not require $\psi_{S,k}$ itself to be parameterized by a learnable decay. □

Corollary B.3 (Finite-prefix approximation of broader temporal patterns)**.** *Fix a prefix length i, and define the multi-coordinate temporal dictionary*

$$\Phi = [\psi_{L,1}, \dots, \psi_{L,K}, \psi_{S,1}, \dots, \psi_{S,K}, \psi_{E,1}, \dots, \psi_{E,K}] \in \mathbb{R}^{i \times 3K}.$$

Let $\mathbf{y} \in \mathbb{R}^i$ be any centered finite-prefix temporal trace, including traces that arise from trend-like, seasonality-like, burst-local, or exponentially decayed components after sampling on the prefix. Let

$$\widehat{\mathbf{y}} = \Phi\Phi^\dagger \mathbf{y}$$

be its least-squares projection onto the span of the `REPAIR` temporal dictionary. If

$$\mathbf{y} = \Phi b + \zeta$$

for some $b \in \mathbb{R}^{3K}$ and residual $\zeta \in \mathbb{R}^i$, then

$$\|\mathbf{y} - \widehat{\mathbf{y}}\|_2 \le \|\zeta\|_2.$$

Therefore, any finite-prefix temporal component whose sampled trace lies close to the span of the long-support, short-support, and episodic kernels can be approximated by the `REPAIR` temporal dictionary, with error controlled by its projection residual.

Moreover, if $K_E \ge i$ episodic atoms are available with distinct positive parameters

$$\xi_1, \dots, \xi_i > 0, \qquad \xi_p \ne \xi_q \text{ for } p \ne q,$$

then the episodic subdictionary alone can span $\mathbb{R}^i$. Hence, on a finite prefix, sufficiently many distinct episodic atoms can represent any sampled temporal trace exactly, including finite-window seasonal or trend-like traces. With fewer atoms, the least-squares projection gives the best finite-prefix approximation in the dictionary span.

Proof. Since

$$\widehat{\mathbf{y}} = \Phi\Phi^\dagger \mathbf{y}$$

is the orthogonal projection of $\mathbf{y}$ onto $\mathrm{span}(\Phi)$, it is the unique minimum-norm least-squares approximation in that span. For any vector $v \in \mathrm{span}(\Phi)$,

$$\|\mathbf{y} - \widehat{\mathbf{y}}\|_2 \le \|\mathbf{y} - v\|_2.$$

Taking $v = \Phi b$, and using the assumed decomposition $\mathbf{y} = \Phi b + \zeta$, gives

$$\|\mathbf{y} - \widehat{\mathbf{y}}\|_2 \le \|\mathbf{y} - \Phi b\|_2 = \|\zeta\|_2.$$

This proves the projection-error claim.

It remains to show the finite-prefix spanning claim for the episodic atoms. Consider the unnormalized episodic matrix

$$E \in \mathbb{R}^{i \times i}, \qquad E_{j,q} = \exp\left(\ell_j \log \xi_q - \xi_q - \log \Gamma(\ell_j + 1)\right),$$

where $\ell_j = i - j \in \{0, 1, \dots, i-1\}$. Equivalently,

$$E_{j,q} = \frac{e^{-\xi_q} \xi_q^{\ell_j}}{\Gamma(\ell_j + 1)}.$$

Define the invertible diagonal row and column scaling matrices

$$D_r = \mathrm{diag}\left(\Gamma(\ell_1 + 1), \dots, \Gamma(\ell_i + 1)\right), \qquad D_c = \mathrm{diag}\left(e^{\xi_1}, \dots, e^{\xi_i}\right).$$

Then

$$(D_r E D_c)_{j,q} = \xi_q^{\ell_j}.$$

Since ℓ_j ranges over $0, 1, \dots, i-1$, possibly in reverse order, the matrix $D_r E D_c$ is a row-permuted Vandermonde matrix. Its determinant is nonzero whenever the ξ_q's are distinct:

$$\det(D_r E D_c) = \pm \prod_{1 \le p < q \le i} (\xi_q - \xi_p) \ne 0.$$

Because D_r and D_c are invertible, E also has full rank i. Therefore, the episodic subdictionary spans $\mathbb{R}^i$. Any finite-prefix sampled trace $\mathbf{y} \in \mathbb{R}^i$ can then be represented exactly by a linear combination of sufficiently many distinct episodic atoms. If fewer atoms are used, the orthogonal projection $\Phi\Phi^\dagger \mathbf{y}$ remains the best approximation in the available dictionary span. □

Interpretation. Theorem 3 explains why the multi-regime temporal component of REPAIR is recoverability-oriented rather than purely heuristic. Theorem 1 first reduces the residual field to an admissible compact repair space, and Theorem 2 shows that an ideal state correction can be recovered if the required compact coordinates are recovered. The present theorem addresses the temporal structure of those coordinates. For each repair coordinate, the centered residual trace over the prefix is modeled as a mixture of long-term, short-term, and episodic atoms, plus an unexplained residual term. If the three atoms are distinguishable, equivalently if the Gram matrix $G_k = \Psi_k^\top \Psi_k$ is invertible with $\lambda_{\min}(G_k) > 0$, then the corresponding mixture coefficients are identifiable. Corollary B.2 further shows that the short-regime kernel need not have a separate learnable decay parameter; a fixed recency envelope is sufficient as long as it remains distinguishable within the temporal dictionary. The recovery error is proportional to the unexplained trace error $\|\varepsilon_k\|_2$ and is amplified only by the conditioning of G_k. Thus, when missing preference evidence has stable, recency-driven, or burst-like temporal structure, REPAIR can separate these traces coordinate-wise and pass them forward as structured repair evidence. Corollary B.3 further clarifies that the three temporal traces should not be read narrowly as only three behavioral types. On a finite prefix, their multi-coordinate dictionary can also approximate sampled trend-like, seasonality-like, burst-local, and decayed components whenever those traces have small projection residual in the kernel span. This directly supports the paper's problem setting: frozen personalization encoders compress temporally heterogeneous user histories into a single terminal state, while REPAIR recovers the under-expressed temporal components before the downstream task head consumes the state.

B.4 Recoverability-Oriented Learning Objective

Theorem 4 (Recoverability-oriented learning and local convergence)**.** *Fix a prefix $\tau_b^{t_1:i}$ and let $c^{(t_j)}$ be the oracle coefficient field from Theorem 2. Define*

$$\hat{c}_\theta^{(t_j)} = \gamma_\theta^{(t_i)} \odot \Pi_{\nu,\theta}^{(t_i)} \left(\eta_\theta^{(t_j)}\right), \qquad e_\theta^{(t_j)} = \hat{c}_\theta^{(t_j)} - c^{(t_j)}.$$

Let

$$\mathcal{E}_{\mathrm{rec}}(\theta) = \sum_{j=1}^{i} \alpha_j^{\star(t_i)} \|e_\theta^{(t_j)}\|_2^2.$$

Assume that, in a neighborhood of a recoverable solution $\theta^\star$, the aggregate coefficient error decomposes orthogonally as

$$e_\theta = e_{\mathrm{task}}(\theta) + e_{\mathrm{coh}}(\theta) + e_{\mathrm{pos}}(\theta),$$

and the three losses satisfy local quadratic control:

$$\mathcal{L}_{\mathrm{task}}(\theta) - \mathcal{L}_{\mathrm{task}}(\theta^\star) \geq \frac{\mu_{\mathrm{task}}}{2} \|e_{\mathrm{task}}(\theta)\|_2^2,$$
$$\mathcal{L}_{\mathrm{coh}}(\theta) - \mathcal{L}_{\mathrm{coh}}(\theta^\star) \geq \frac{\mu_{\mathrm{coh}}}{2} \|e_{\mathrm{coh}}(\theta)\|_2^2,$$
$$\mathcal{L}_{\mathrm{pos}}(\theta) - \mathcal{L}_{\mathrm{pos}}(\theta^\star) \geq \frac{\mu_{\mathrm{pos}}}{2} \|e_{\mathrm{pos}}(\theta)\|_2^2.$$

For

$$\mathcal{L}_{\mathrm{repair}} = \lambda_{\mathrm{task}}\mathcal{L}_{\mathrm{task}} + \lambda_{\mathrm{coh}}\mathcal{L}_{\mathrm{coh}} + \lambda_{\mathrm{pos}}\mathcal{L}_{\mathrm{pos}}, \qquad \lambda_{\mathrm{task}} > \lambda_{\mathrm{coh}} > \lambda_{\mathrm{pos}} > 0,$$

define

$$\mu_\lambda = \min\{\lambda_{\mathrm{task}}\mu_{\mathrm{task}}, \lambda_{\mathrm{coh}}\mu_{\mathrm{coh}}, \lambda_{\mathrm{pos}}\mu_{\mathrm{pos}}\}.$$

Then

$$\mathcal{L}_{\mathrm{repair}}(\theta) - \mathcal{L}_{\mathrm{repair}}(\theta^\star) \geq \frac{\mu_\lambda}{2} \|e_\theta\|_2^2,$$

and hence, for some constant $C_\alpha > 0$,

$$\mathcal{E}_{\mathrm{rec}}(\theta) \leq \frac{2C_\alpha}{\mu_\lambda} \left(\mathcal{L}_{\mathrm{repair}}(\theta) - \mathcal{L}_{\mathrm{repair}}(\theta^\star)\right).$$

If, additionally, $\mathcal{L}_{\mathrm{repair}}$ is L_λ-smooth and satisfies the local PL condition

$$\frac{1}{2}\|\nabla_\theta \mathcal{L}_{\mathrm{repair}}(\theta)\|_2^2 \geq \mu_{\mathrm{PL}} \left(\mathcal{L}_{\mathrm{repair}}(\theta) - \mathcal{L}_{\mathrm{repair}}(\theta^\star)\right),$$

then gradient descent with step size $1/L_\lambda$ satisfies

$$\mathcal{E}_{\mathrm{rec}}(\theta_s) \leq \frac{2C_\alpha}{\mu_\lambda} \left(1 - \frac{\mu_{\mathrm{PL}}}{L_\lambda}\right)^s \left(\mathcal{L}_{\mathrm{repair}}(\theta_0) - \mathcal{L}_{\mathrm{repair}}(\theta^\star)\right).$$

Proof. By orthogonality,

$$\|e_\theta\|_2^2 = \|e_{\text{task}}(\theta)\|_2^2 + \|e_{\text{coh}}(\theta)\|_2^2 + \|e_{\text{pos}}(\theta)\|_2^2.$$

Multiplying the three local quadratic bounds by their weights and summing gives

$$\begin{aligned}\mathcal{L}_{\text{repair}}(\theta) - \mathcal{L}_{\text{repair}}(\theta^\star) &\geq \frac{\lambda_{\text{task}}\mu_{\text{task}}}{2}\|e_{\text{task}}(\theta)\|_2^2 + \frac{\lambda_{\text{coh}}\mu_{\text{coh}}}{2}\|e_{\text{coh}}(\theta)\|_2^2 \\ &\quad + \frac{\lambda_{\text{pos}}\mu_{\text{pos}}}{2}\|e_{\text{pos}}(\theta)\|_2^2 \\ &\geq \frac{\mu_\lambda}{2}\left(\|e_{\text{task}}(\theta)\|_2^2 + \|e_{\text{coh}}(\theta)\|_2^2 + \|e_{\text{pos}}(\theta)\|_2^2\right) \\ &= \frac{\mu_\lambda}{2}\|e_\theta\|_2^2.\end{aligned}$$

Therefore,

$$\|e_\theta\|_2^2 \leq \frac{2}{\mu_\lambda}\left(\mathcal{L}_{\text{repair}}(\theta) - \mathcal{L}_{\text{repair}}(\theta^\star)\right).$$

Since $\mathcal{E}_{\text{rec}}$ is a weighted squared coefficient error, there exists $C_\alpha > 0$ such that

$$\mathcal{E}_{\text{rec}}(\theta) \leq C_\alpha\|e_\theta\|_2^2.$$

Combining the two inequalities yields

$$\mathcal{E}_{\text{rec}}(\theta) \leq \frac{2C_\alpha}{\mu_\lambda}\left(\mathcal{L}_{\text{repair}}(\theta) - \mathcal{L}_{\text{repair}}(\theta^\star)\right).$$

For the convergence claim, L_λ-smoothness gives

$$\mathcal{L}_{\text{repair}}(\theta_{s+1}) \leq \mathcal{L}_{\text{repair}}(\theta_s) - \frac{1}{2L_\lambda}\|\nabla_\theta\mathcal{L}_{\text{repair}}(\theta_s)\|_2^2.$$

Using the local PL condition,

$$\mathcal{L}_{\text{repair}}(\theta_{s+1}) - \mathcal{L}_{\text{repair}}(\theta^\star) \leq \left(1 - \frac{\mu_{\text{PL}}}{L_\lambda}\right)\left(\mathcal{L}_{\text{repair}}(\theta_s) - \mathcal{L}_{\text{repair}}(\theta^\star)\right).$$

Iterating gives

$$\mathcal{L}_{\text{repair}}(\theta_s) - \mathcal{L}_{\text{repair}}(\theta^\star) \leq \left(1 - \frac{\mu_{\text{PL}}}{L_\lambda}\right)^s\left(\mathcal{L}_{\text{repair}}(\theta_0) - \mathcal{L}_{\text{repair}}(\theta^\star)\right).$$

Substituting this into the recoverability-error bound proves the result. □

Corollary B.4 (Role of the loss ordering)**.** *Under Theorem 4, the ordering*

$$\lambda_{\text{task}} > \lambda_{\text{coh}} > \lambda_{\text{pos}} > 0$$

is recoverability-oriented. The task loss receives the largest weight because it is the only term directly tied to downstream utility of $\tilde{z}_b^{(t_i)}$. The coherence loss receives the larger auxiliary weight because it controls coefficient-level consistency with source events. The positional loss remains positive but smaller because it prevents temporally collapsed or permuted correction fields. If $\lambda_{\text{coh}} = \lambda_{\text{pos}} = 0$, the task-only objective may control only downstream-visible aggregate error while leaving source-event inconsistency and source-timestep ambiguity unconstrained.

C Evaluation Details

We describe the evaluation construction used across predictive and generative personalization tasks in details. The purpose of the evaluation protocol is to test whether REPAIR improves the downstream-facing compressed preference state of a frozen host encoder, rather than whether a new encoder can be trained from scratch.

C.1 UIG Construction and Eventization

We use a User Interaction Graph (UIG) as the shared interaction substrate for all predictive and generative personalization tasks. The purpose of the UIG in REPAIR is not to introduce a new encoder, but to standardize heterogeneous user histories into temporally ordered event units whose cached representations can later be compared against the frozen host's compressed preference state.

For a user u, the UIG is a directed acyclic graph

$$G_u = \langle N_u, E_u \rangle, \tag{31}$$

where N_u is a typed node set and E_u is an action-labeled edge set. We use three broad node families:

$$N_u = N_u^{\text{user}} \cup N_u^{\text{content}} \cup N_u^{\text{response}}. \tag{32}$$

Here, N_u^{user} contains the initial user node $u^{(t_0)}$; N_u^{content} contains content nodes $d^{(t_i)}$, such as movies, news articles, products, query documents, dialogue turns, or recommendation entities; and N_u^{response} contains response nodes $r_u^{(t_i)}$, such as user-written headlines, gold summaries, model-generated summaries, dialogue responses, or other task-specific outputs.

Edges encode observed or induced user behavior. We write an edge at timestep t_i as

$$\left(v_{\text{head}}^{(t_{i-1})}, a^{(t_i)}, v_{\text{tail}}^{(t_i)}\right), \tag{33}$$

where $v_{\text{head}}^{(t_{i-1})}$ is the previous node, $v_{\text{tail}}^{(t_i)}$ is the content or response node reached at timestep t_i, and $a^{(t_i)}$ is the action label. We use a normalized action vocabulary:

$$a^{(t_i)} \in \mathcal{A} = \mathcal{A}^+ \cup \mathcal{A}^- \cup \mathcal{A}^{\text{gen}} \cup \mathcal{A}^{\text{res}}. \tag{34}$$

Positive actions $\mathcal{A}^+$ include clicks, high ratings, purchases, accepted recommendations, consumed items, or positive dialogue stance. Negative actions $\mathcal{A}^-$ include skips, non-clicks, low ratings, rejected recommendations, or negative dialogue stance. Generation-trigger actions $\mathcal{A}^{\text{gen}}$ connect a content or query node to a requested generation event. Response edges $\mathcal{A}^{\text{res}}$ connect a content/query node to the corresponding response node, such as a generated headline, summary, or dialogue response.

A raw user trajectory is a time-ordered path in the UIG:

$$\tau_u = \left(u^{(t_0)}, v^{(t_1)}, v^{(t_2)}, \ldots, v^{(t_i)}\right), \tag{35}$$

together with its action sequence

$$\left(a^{(t_1)}, a^{(t_2)}, \ldots, a^{(t_i)}\right). \tag{36}$$

This raw graph-level sequence is the *u-tier*. It preserves dataset-specific interaction structure, including content exposure, user feedback, generation requests, and generated or gold responses.

For model-side processing, we derive an upper interaction tier, called the *I-tier*. Each transition in the u-tier is converted into an interaction block:

$$b^{(t_i)} = \left\langle a^{(t_i)}, \text{tl}^{(t_i)} \right\rangle, \tag{37}$$

where $\text{tl}^{(t_i)} \in N_u$ is the tail content or response node on which the action occurs. The corresponding I-tier trajectory is

$$\tau_b^{t_1:i} = \left\langle b^{(t_1)}, b^{(t_2)}, \ldots, b^{(t_i)} \right\rangle. \tag{38}$$

This abstraction compresses heterogeneous low-level graph transitions into ordered behavior units. It is therefore the interface used by frozen host encoders and by REPAIR.

Each interaction block is mapped to a host-specific event representation:

$$e_b^{(t_i)} = \phi_{\text{evt}}\left(b^{(t_i)}\right), \tag{39}$$

where $\phi_{\text{evt}}(\cdot)$ is the event representation map available for the corresponding host. For recommendation encoders, this may be an item, action, time, and position embedding. For news encoders, it

may include text/content embeddings. For multimodal product encoders, it may include image-text product representations. For generative tasks, it may include document, query, headline, summary, response, or dialogue-turn embeddings together with the action label.

The frozen host encoder compresses the I-tier prefix into the exposed preference state:

$$z_b^{(t_i)} = f_{\text{enc}}\left(e_b^{(t_1)}, e_b^{(t_2)}, \ldots, e_b^{(t_i)}\right). \tag{40}$$

The downstream task head then consumes $z_b^{(t_i)}$ for prediction or generation. REPAIR operates after this compression step. It receives $z_b^{(t_i)}$ and the cached event representations $\{e_b^{(t_j)}\}_{j=1}^{i}$, estimates which timestep-level evidence remains under-expressed in the compressed state, and forms the repaired state $\tilde{z}_b^{(t_i)}$ before the downstream head is applied.

PENS-Style Interaction Logs. For PENS-style datasets, the UIG is constructed directly from impression logs. Clicked news items are mapped to content nodes with positive edges, while unclicked or skipped news items are mapped to content nodes with negative edges:

$$\text{clkNews} \mapsto (d^{(t_i)}, a^{(t_i)} = \text{pos}), \qquad \text{uclkNews} \mapsto (d^{(t_i)}, a^{(t_i)} = \text{neg}). \tag{41}$$

This produces a content-level trajectory pool. When response nodes are required for personalized generation, we insert summary or headline response nodes using available references or locally coherent surrogate responses. The inserted response node is linked to the corresponding content node through generation and response edges:

$$d^{(t_i)} \xrightarrow{\text{genRes}} r_u^{(t_i+1)}, \qquad r_u^{(t_i+1)} \xrightarrow{\text{resGen}} \tau_u. \tag{42}$$

In REPAIR, these inserted response nodes are used only to build training histories and task targets; held-out references remain reserved for evaluation.

Dialogue-Style Data. For multi-turn dialogue datasets, the raw data consists of interleaved utterances rather than explicit click/skip logs. We therefore induce UIG edges from dialogue structure. A context utterance or mentioned entity is treated as a content node, and the target speaker's next response is treated as a response node. For a target user u, recent non-target turns are mapped to candidate content nodes. The target response is then used to infer whether the user expresses a positive or negative stance toward each recent context node:

$$a^{(t_i)} = \begin{cases} \text{pos}, & \text{if the response expresses agreement, acceptance, or positive stance,} \\ \text{neg}, & \text{if the response expresses disagreement, rejection, or neutral/negative stance.} \end{cases} \tag{43}$$

This produces a dialogue-derived trajectory pool structurally compatible with the same I-tier eventization. Since this construction requires inferred stance labels rather than directly logged actions, dialogue datasets are treated as broader-applicability stress tests rather than as the cleanest evidence for structured UIG repair.

Predictive Task Targets. For predictive personalization, the target at timestep t_i is the next content node:

$$r_u^{\star(t_i+1)} \equiv c_u^{(t_i+1)}. \tag{44}$$

The downstream head ranks this target against sampled negatives. The corresponding repair input is the frozen state $z_b^{(t_i)}$ and the cached I-tier event sequence up to t_i.

Generative Task Targets. For personalized generation, the target is a response node:

$$r_u^{\star(t_i+1)} \in \{\text{headline}, \text{summary}, \text{dialogue response}\}. \tag{45}$$

The query content $d_q^{(t_i)}$ or dialogue context is attached to the task instance. During training, the gold response node is used as supervision. During autoregressive evaluation, the model-generated response $\hat{r}_u^{(t_i+1)}$ is appended back into the u-tier trajectory and then converted into the next I-tier event, so later steps condition on the evolving generated history.

UIG Augmentation. When training trajectories are sparse or response nodes are incomplete, we apply trajectory-level augmentation only on the training pool. First, cross-trajectory segment substitution preserves an early prefix of a target trajectory and substitutes later contiguous segments from source trajectories while retaining protected gaps of original target steps. This increases trajectory diversity while preserving local temporal realism. Second, when surrogate response nodes become locally incoherent after substitution, we replace the response with a sentence selected from the preceding content node using a recency-weighted context match:

$$\hat{r}^{(t_i)} = \arg\min_{s_p} \sum_{q=1}^{k} \exp\left(-\lambda \operatorname{pos}(c_q)\right) \sigma(e_{s_p}, e_{c_q}), \tag{46}$$

where $\{c_q\}_{q=1}^{k}$ is the backward context window, $\operatorname{pos}(c_q)$ is its recency index, λ is a decay coefficient, and $\sigma(\cdot,\cdot)$ is the embedding-space distance. This augmentation is used only for training and does not leak test references into the training trajectories.

Role of UIG in REPAIR. The UIG construction supplies the two objects required by REPAIR: the compressed state $z_b^{(t_i)}$ produced by a frozen host and the timestep-level event representations $\{e_b^{(t_j)}\}_{j=1}^{i}$ that produced it. The u-tier preserves task-specific interaction semantics, while the I-tier converts those interactions into a compact ordered sequence of event units. REPAIR then performs state-relative correction over this cached event sequence. It does not rebuild the user history, retrain the base encoder, or replace the downstream head. Instead, it repairs the already exposed compressed preference state before prediction or generation.

C.2 Dataset-Specific Evaluation Tasks

We evaluate REPAIR on both predictive and generative personalization tasks.

Movie Recommendation. MovieLens is used for sequential movie recommendation. Each test instance is formed from a user history prefix and a held-out next movie target.

News Recommendation. MIND and PENS are used for news recommendation. User histories are constructed from click and non-click behavior, and the model predicts the next clicked or preferred news item from a candidate set.

Product Recommendation. Amazon Reviews 2023 is used for multimodal product recommendation. Product states are obtained from frozen multimodal product encoders, and REPAIR is tested on whether the exposed product-preference state can be corrected post hoc.

Personalized Headline Generation. PENS is used for personalized headline generation. The model conditions on user history and candidate news content, and generates a user-specific headline.

Conversational Recommendation and Dialogue Generation. TG-ReDial is used as a conversational stress test. Since raw dialogue does not directly provide the same explicit action-content structure as click, rating, or skip logs, interaction units are induced from dialogue turns, mentioned entities, recommendation attempts, and feedback signals. We therefore treat TG-ReDial as broader-applicability evidence rather than as the primary structured-UIG benchmark.

C.3 Training and Optimization Details

We use a staged training protocol that keeps the intervention target fixed: the host encoder is first trained under its original task supervision, then frozen, and REPAIR is trained as a corrective pre-head module using the frozen host state and cached timestep representations. During repair learning, the raw trajectory is not re-encoded by a second sequence model. REPAIR receives the exposed compressed state and the per-timestep event representations already produced by the host forward pass, and learns a state-relative correction before the downstream head consumes the repaired state.

Base-host training. Each base host is trained under the supervision used by its corresponding task. For MovieLens and Amazon Reviews recommendation, base encoders are trained for 20 epochs. For news recommendation, the base encoder is trained for 5 epochs. For personalized headline generation, the base encoder is trained for 5 epochs and the downstream generation decoder is trained for 2 epochs. After this stage, the host encoder parameters are frozen for all REPAIR experiments.

Head-only control. For the RQ1 head-only control, we start from the same task-trained host checkpoint used by the REPAIR conditions, keep the encoder and its task-facing state fixed, and further train only the original downstream task head. We evaluate checkpoints from epochs 1–5 and retain the strongest validation checkpoint. This isolates the effect of continued task-head optimization without modifying the encoder state or introducing REPAIR.

Repair-stage training. For MovieLens and Amazon Reviews recommendation, REPAIR is attached to the frozen encoder and trained for 12 additional epochs. During the first 10 repair epochs, the downstream recommendation head remains frozen, so the update tests whether the compressed preference state itself can be improved. During the final 2 epochs, the downstream recommendation head is fine-tuned for final alignment with the repaired state. For news recommendation, REPAIR is trained over the frozen encoder for 2 epochs, followed by 2 additional epochs of recommendation-head fine-tuning. For personalized headline generation, REPAIR is trained over the frozen encoder for 2 epochs, followed by 2 additional epochs of decoder fine-tuning.

Recommendation training objective. For predictive personalization, the task objective is the cross-entropy loss over the positive target and sampled negative candidates during minibatch training. The reported evaluation protocol is kept stricter and fixed: each test instance is evaluated with one ground-truth target and 499 negative candidates. During the frozen-head repair stage, gradients update only REPAIR. During the head-alignment stage, the repaired state is fixed as the downstream-facing representation and the recommendation head is allowed to adapt to the corrected state.

Generation training objective. For personalized headline generation, the decoder is trained with teacher forcing on the gold target sequence. The source document representation is combined with the user preference state through the same decoder interface used at evaluation time. During REPAIR training, the repaired preference state replaces the frozen host state as the user-side conditioning signal. During the final decoder-alignment phase, the decoder is fine-tuned to consume the repaired state more effectively, while the host encoder remains frozen.

Repair losses. The repair module is optimized with the weighted objective

$$\mathcal{L}_{\text{repair}} = \lambda_{\text{task}}\mathcal{L}_{\text{task}} + \lambda_{\text{pos}}\mathcal{L}_{\text{pos}} + \lambda_{\text{coh}}\mathcal{L}_{\text{coh}},$$

with

$$(\lambda_{\text{task}}, \lambda_{\text{pos}}, \lambda_{\text{coh}}) = (0.5, 0.2, 0.3).$$

The task loss drives downstream utility, the positional loss prevents temporally collapsed correction fields, and the coherence loss keeps the retained correction aligned with its source event representation.

Optimization details. All trainable modules in the reference implementation are optimized with AdamW. Unless a host-specific released recipe requires a different setting, we use learning rate 2×10^{-4} and weight decay 10^{-4} for base-host training, repair training, and final head/decoder alignment. Gradients are clipped to a maximum norm of 1.0. Mixed-precision training is enabled when CUDA is available. All stochastic components are seeded, and each paired comparison reuses the same train/validation/test partition and evaluation candidate construction.

Compute resources. All experiments were run on paid GPU resources provisioned through Lightning.ai Studios/Clusters. Depending on task scale and availability, we used NVIDIA A100, NVIDIA H100, and NVIDIA RTX PRO 5000-class GPU resources. A100/H100 resources were used for the larger frozen-host and REPAIR training runs, while RTX PRO 5000-class resources were used for smaller debugging, preprocessing, checkpoint validation, and lightweight evaluation jobs. We used single-GPU jobs unless the corresponding host recipe required otherwise, and all reported comparisons use the same compute environment within each dataset–host pair.

Text preprocessing for PENS-style generation hosts. For the lightweight word-level PENS generation implementation, news titles are truncated to 30 tokens, news bodies to 120 tokens, concatenated source inputs to 160 tokens, and target summaries/headlines to 60 tokens. The vocabulary is built from the training-side news and summary text with minimum frequency 2 and a maximum vocabulary size of 100,000. These preprocessing limits affect only the tensorized encoder–decoder implementation and do not change the reported evaluation protocol or metric computation.

Checkpointing and model selection. We checkpoint after each training stage and evaluate the frozen host, the selected head-only checkpoint, `REPAIR` without head fine-tuning, and `REPAIR` with the final head/decoder alignment using the corresponding saved checkpoints. For the head-only control, the strongest validation checkpoint among epochs 1–5 is retained. For fair paired comparisons, all systems are evaluated on the same held-out examples, the same history prefixes, and the same candidate sets.

D Metric Formulations

This section defines the metrics used for predictive personalization, generation accuracy, personalization-sensitive generation, and deployment-friendliness.

D.1 Predictive Personalization Metrics

For a test instance n, let $r_n^\star$ denote the rank position of the ground-truth target after sorting all candidates by the model score in descending order. Let N denote the number of test instances.

Mean Reciprocal Rank. Mean Reciprocal Rank measures how early the first relevant target appears in the ranked list:

$$\mathrm{MRR} = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{r_n^\star}. \tag{47}$$

MRR strongly rewards placing the correct target at the first rank and rapidly decreases as the target moves downward.

Discounted Cumulative Gain and Normalized DCG. Let $\mathrm{rel}_{n,i} \in \{0, 1\}$ denote the relevance of the item at rank i for test instance n. The discounted cumulative gain at cutoff K is

$$\mathrm{DCG@}K_n = \sum_{i=1}^{K} \frac{2^{\mathrm{rel}_{n,i}} - 1}{\log_2(i+1)}. \tag{48}$$

The normalized score is

$$\mathrm{nDCG@}K = \frac{1}{N} \sum_{n=1}^{N} \frac{\mathrm{DCG@}K_n}{\mathrm{IDCG@}K_n}, \tag{49}$$

where $\mathrm{IDCG@}K_n$ is the ideal DCG for instance n. We report nDCG@5 and nDCG@10.

Hit Rate. Hit@K measures whether the ground-truth target appears in the top-K predictions:

$$\mathrm{Hit@}K = \frac{1}{N} \sum_{n=1}^{N} \mathbf{1}\{r_n^\star \le K\}. \tag{50}$$

With one ground-truth target per instance, Hit@K is equivalent to Recall@K.

D.2 Generation Accuracy Metrics

For generation tasks, let $\hat{y}$ denote the generated output and $y^\star$ denote the gold target. We use lexical, order-sensitive, weighted, paraphrase-aware, and semantic metrics.

ROUGE-N. Let $\mathcal{G}_n(x)$ denote the multiset of n-grams in text x, and let $\mathrm{cnt}_x(g)$ denote the count of g in x. ROUGE-N is

$$\text{ROUGE-}N(\hat{y}, y^\star) = \frac{\sum_{g \in \mathcal{G}_n(y^\star)} \min\left(\mathrm{cnt}_{\hat{y}}(g), \mathrm{cnt}_{y^\star}(g)\right)}{\sum_{g \in \mathcal{G}_n(y^\star)} \mathrm{cnt}_{y^\star}(g)}. \tag{51}$$

We use ROUGE-1 and ROUGE-2 for unigram and bigram content preservation.

ROUGE-L. ROUGE-L measures longest common subsequence overlap. Let $\mathrm{LCS}(\hat{y}, y^\star)$ denote the length of the longest common subsequence. Define

$$P_{\mathrm{LCS}} = \frac{\mathrm{LCS}(\hat{y}, y^\star)}{|\hat{y}|}, \qquad R_{\mathrm{LCS}} = \frac{\mathrm{LCS}(\hat{y}, y^\star)}{|y^\star|}. \tag{52}$$

The ROUGE-L F-score is

$$\text{ROUGE-}L = \frac{(1+\beta^2) P_{\mathrm{LCS}} R_{\mathrm{LCS}}}{R_{\mathrm{LCS}} + \beta^2 P_{\mathrm{LCS}}}, \tag{53}$$

where β controls the precision–recall tradeoff.

ROUGE-SU4. ROUGE-SU4 measures overlap of unigrams and skip-bigrams with maximum skip distance four. Let $\mathcal{G}_{\mathrm{SU4}}(x)$ denote the multiset of unigrams and SU4 skip-bigrams in x. Then

$$\text{ROUGE-SU4}(\hat{y}, y^\star) = \frac{\sum_{g \in \mathcal{G}_{\mathrm{SU4}}(y^\star)} \min\left(\mathrm{cnt}_{\hat{y}}(g), \mathrm{cnt}_{y^\star}(g)\right)}{\sum_{g \in \mathcal{G}_{\mathrm{SU4}}(y^\star)} \mathrm{cnt}_{y^\star}(g)}. \tag{54}$$

Weighted ROUGE. Weighted ROUGE assigns a non-uniform weight $w(g) \geq 0$ to each matched unit g. For an n-gram variant,

$$\text{W-ROUGE-}N(\hat{y}, y^\star) = \frac{\sum_{g \in \mathcal{G}_n(y^\star)} w(g) \min\left(\mathrm{cnt}_{\hat{y}}(g), \mathrm{cnt}_{y^\star}(g)\right)}{\sum_{g \in \mathcal{G}_n(y^\star)} w(g)\, \mathrm{cnt}_{y^\star}(g)}. \tag{55}$$

For weighted LCS-style scoring, let $\mathrm{WLCS}(\hat{y}, y^\star)$ denote the weighted longest common subsequence. Then

$$P_{\mathrm{WLCS}} = \frac{f^{-1}(\mathrm{WLCS}(\hat{y}, y^\star))}{|\hat{y}|}, \qquad R_{\mathrm{WLCS}} = \frac{f^{-1}(\mathrm{WLCS}(\hat{y}, y^\star))}{|y^\star|}, \tag{56}$$

and

$$\text{ROUGE-}W = \frac{(1+\beta^2) P_{\mathrm{WLCS}} R_{\mathrm{WLCS}}}{R_{\mathrm{WLCS}} + \beta^2 P_{\mathrm{WLCS}}}. \tag{57}$$

Here, $f(\cdot)$ is the weighting function used by the weighted-LCS implementation.

Paraphrase-Aware ROUGE. Paraphrase-aware ROUGE relaxes exact overlap by allowing semantically equivalent units to match. Let $\mathrm{sim}_P(g, h) \in [0, 1]$ denote a paraphrase-aware similarity between generated unit g and reference unit h. For a reference unit h, define

$$\mathrm{pmatch}_{\hat{y}, y^\star}(h) = \max_{g \in \mathcal{G}(\hat{y})} \mathrm{sim}_P(g, h), \tag{58}$$

where $\mathcal{G}(\cdot)$ is the selected unit set. The paraphrase-aware ROUGE recall is

$$\text{P-ROUGE}(\hat{y}, y^\star) = \frac{\sum_{h \in \mathcal{G}(y^\star)} \mathrm{pmatch}_{\hat{y}, y^\star}(h)}{|\mathcal{G}(y^\star)|}. \tag{59}$$

In our reporting, P-RG-L denotes the paraphrase-aware ROUGE-L variant.

METEOR. METEOR aligns generated and reference unigrams using exact, stem, synonym, and paraphrase-level matches. Let m be the number of matched unigrams. Precision and recall are

$$P = \frac{m}{|\hat{y}|}, \qquad R = \frac{m}{|y^\star|}. \tag{60}$$

The harmonic mean is

$$F_\alpha = \frac{PR}{\alpha P + (1-\alpha) R}, \tag{61}$$

and the fragmentation penalty is

$$\mathrm{Pen} = \gamma \left(\frac{c}{m} \right)^{\theta}, \tag{62}$$

where c is the number of matched chunks. The final score is

$$\mathrm{METEOR}(\hat{y}, y^{\star}) = (1 - \mathrm{Pen}) F_{\alpha}. \tag{63}$$

BLEU-4. BLEU-4 computes modified n-gram precision up to $n = 4$ with a brevity penalty:

$$\text{BLEU-4} = \mathrm{BP} \cdot \exp\left(\sum_{n=1}^{4} w_n \log p_n \right), \qquad w_n = \frac{1}{4}. \tag{64}$$

The brevity penalty is

$$\mathrm{BP} = \begin{cases} 1, & |\hat{y}| > |y^{\star}|, \\ \exp\left(1 - \frac{|y^{\star}|}{|\hat{y}|}\right), & |\hat{y}| \le |y^{\star}|. \end{cases} \tag{65}$$

BERTScore. BERTScore measures semantic similarity between contextual token embeddings. Let $\mathbf{h}(\hat{y}_a)$ and $\mathbf{h}(y_b^{\star})$ be contextual embeddings, and let

$$s_{ab} = \cos\left(\mathbf{h}(\hat{y}_a), \mathbf{h}(y_b^{\star})\right). \tag{66}$$

BERTScore precision and recall are

$$P_{\mathrm{BS}} = \frac{1}{|\hat{y}|} \sum_{a=1}^{|\hat{y}|} \max_b s_{ab}, \qquad R_{\mathrm{BS}} = \frac{1}{|y^{\star}|} \sum_{b=1}^{|y^{\star}|} \max_a s_{ab}. \tag{67}$$

The final F-score is

$$F_{\mathrm{BS}} = \frac{2 P_{\mathrm{BS}} R_{\mathrm{BS}}}{P_{\mathrm{BS}} + R_{\mathrm{BS}}}. \tag{68}$$

D.3 PerSEval for Personalized Generation

Standard generation metrics compare $\hat{y}$ to a single reference. Personalized generation additionally requires sensitivity to differences among user-specific expected outputs. We therefore use PerSEval-style metrics as a personalization wrapper around a plug-in distance function σ.

Let d_i denote the source document or query context. Let $\mathcal{U}_{d_i}$ be the set of users associated with d_i. For user j, let u_{ij} denote the user-specific expected output and let $\hat{s}_{ij}$ denote the model-generated output. Let $\sigma(\cdot, \cdot)$ be a distance or divergence function over text.

Summary-Level Responsiveness. For users $j, k \in \mathcal{U}_{d_i}$, define

$$w(u_{ij} \mid u_{ik}) = \frac{\sigma(u_{ij}, u_{ik})}{\sigma(u_{ij}, d_i) + \epsilon}, \qquad w(\hat{s}_{ij} \mid \hat{s}_{ik}) = \frac{\sigma(\hat{s}_{ij}, \hat{s}_{ik})}{\sigma(\hat{s}_{ij}, d_i) + \epsilon}. \tag{69}$$

The reference-side and generation-side weighted divergences are

$$X_{ijk} = \frac{\exp(w(u_{ij} \mid u_{ik}))}{\sum_{\ell=1}^{|\mathcal{U}_{d_i}|} \exp(w(u_{ij} \mid u_{i\ell}))} \cdot \sigma(u_{ij}, u_{ik}), \tag{70}$$

$$Y_{ijk} = \frac{\exp(w(\hat{s}_{ij} \mid \hat{s}_{ik}))}{\sum_{\ell=1}^{|\mathcal{U}_{d_i}|} \exp(w(\hat{s}_{ij} \mid \hat{s}_{i\ell}))} \cdot \sigma(\hat{s}_{ij}, \hat{s}_{ik}). \tag{71}$$

The summary-level responsiveness score is

$$\mathrm{DEGRESS}_{\sigma}(\hat{s}_{ij} \mid (d_i, u_{ij})) = \frac{1}{|\mathcal{U}_{d_i}|} \sum_{k=1}^{|\mathcal{U}_{d_i}|} \frac{\min(X_{ijk}, Y_{ijk}) + \epsilon}{\max(X_{ijk}, Y_{ijk}) + \epsilon}. \tag{72}$$

Accuracy-Penalized Personalization. PerSEval multiplies responsiveness by an effective accuracy penalty:

$$\mathrm{PSE}_\sigma(\hat{s}_{ij} \mid (d_i, u_{ij})) = \mathrm{DEGRESS}_\sigma(\hat{s}_{ij} \mid (d_i, u_{ij})) \cdot \mathrm{EDP}_\sigma(\hat{s}_{ij} \mid (d_i, u_{ij})). \tag{73}$$

The penalty factor is

$$\mathrm{EDP}_\sigma(\hat{s}_{ij} \mid (d_i, u_{ij})) = 1 - \frac{1}{1 + 10^\alpha \exp\left(-10^\beta \, \mathrm{DGP}_\sigma(\hat{s}_{ij} \mid (d_i, u_{ij}))\right)}, \tag{74}$$

where $\alpha \geq 3$, $\beta \geq 1$, and

$$\mathrm{DGP}_\sigma(\hat{s}_{ij} \mid (d_i, u_{ij})) = \mathrm{ADP}_\sigma(\hat{s}_{i\star} \mid (d_i, u_{i\star})) + \mathrm{ACP}_\sigma(\hat{s}_{ij} \mid (d_i, u_{ij})). \tag{75}$$

Accuracy Drop Penalty. The document-level best-case accuracy drop penalty is

$$\mathrm{ADP}_\sigma(\hat{s}_{i\star} \mid (d_i, u_{i\star})) = \frac{1}{1 + 10^\gamma \exp\left(-10 \cdot \frac{\sigma_i^\star}{(1-\sigma_i^\star)+\epsilon}\right)}, \tag{76}$$

where $\gamma \geq 4$ and

$$\sigma_i^\star = \min_{j=1}^{|\mathcal{U}_{d_i}|} \sigma(\hat{s}_{ij}, u_{ij}). \tag{77}$$

Accuracy Consistency Penalty. The accuracy inconsistency penalty is

$$\mathrm{ACP}_\sigma(\hat{s}_{ij} \mid (d_i, u_{ij})) = \frac{1}{1 + 10^\gamma \exp\left(-10 \cdot \frac{\sigma(\hat{s}_{ij}, u_{ij}) - \sigma_i^\star}{(\bar{\sigma}_i - \sigma_i^\star)+\epsilon}\right)}, \tag{78}$$

where

$$\bar{\sigma}_i = \frac{1}{|\mathcal{U}_{d_i}|} \sum_{j=1}^{|\mathcal{U}_{d_i}|} \sigma(\hat{s}_{ij}, u_{ij}). \tag{79}$$

System-Level PerSEval. The final system-level score is

$$\mathrm{PSE}_\sigma(M) = \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \frac{1}{|\mathcal{U}_{d_i}|} \sum_{j=1}^{|\mathcal{U}_{d_i}|} \mathrm{PSE}_\sigma(\hat{s}_{ij} \mid (d_i, u_{ij})). \tag{80}$$

Plug-in Variants. PSE-JSD uses Jensen–Shannon divergence as σ:

$$\sigma_{\mathrm{JSD}}(G, R) = \mathrm{JSD}(P_G \| P_R) = \frac{1}{2}\,\mathrm{KL}(P_G \| M) + \frac{1}{2}\,\mathrm{KL}(P_R \| M), \qquad M = \frac{1}{2}(P_G + P_R). \tag{81}$$

PSE-W-JSD uses weighted distributions:

$$P_G^{(w)}(g) = \frac{w(g)\,\mathrm{cnt}_G(g)}{\sum_h w(h)\,\mathrm{cnt}_G(h)}, \qquad P_R^{(w)}(g) = \frac{w(g)\,\mathrm{cnt}_R(g)}{\sum_h w(h)\,\mathrm{cnt}_R(h)}, \tag{82}$$

$$\sigma_{\text{W-JSD}}(G, R) = \mathrm{JSD}(P_G^{(w)} \| P_R^{(w)}). \tag{83}$$

PSE-SU4 uses

$$\sigma_{\mathrm{SU4}}(G, R) = 1 - \text{ROUGE-SU4}(G, R). \tag{84}$$

PSE-METEOR uses

$$\sigma_{\mathrm{METEOR}}(G, R) = 1 - \mathrm{METEOR}(G, R). \tag{85}$$

PSE-P-RG-L uses

$$\sigma_{\text{P-RG-L}}(G, R) = 1 - \text{P-ROUGE-}L(G, R). \tag{86}$$

Thus, the same PSE wrapper can evaluate personalization with lexical, weighted, semantic, or paraphrase-aware notions of output distance.

D.4 Deployment-Friendliness Metrics

For deployment analysis, we report memory overhead, latency overhead, throughput change, MRR-gain-to-latency ratio, and MRR-gain-to-memory ratio.

Let $\mathrm{Mem}_{\mathrm{base}}$ and $\mathrm{Mem}_{\mathrm{rep}}$ denote memory consumption of the frozen host and repaired model. Let $\mathrm{Lat}_{\mathrm{base}}$ and $\mathrm{Lat}_{\mathrm{rep}}$ denote end-to-end inference latency. Memory and latency overheads are

$$\mathrm{MemOH} = 100 \cdot \frac{\mathrm{Mem}_{\mathrm{rep}} - \mathrm{Mem}_{\mathrm{base}}}{\mathrm{Mem}_{\mathrm{base}}}, \tag{87}$$

$$\mathrm{LatOH} = 100 \cdot \frac{\mathrm{Lat}_{\mathrm{rep}} - \mathrm{Lat}_{\mathrm{base}}}{\mathrm{Lat}_{\mathrm{base}}}. \tag{88}$$

If throughput is measured directly, throughput change is

$$\mathrm{ThrChange} = 100 \cdot \frac{\mathrm{Thr}_{\mathrm{rep}} - \mathrm{Thr}_{\mathrm{base}}}{\mathrm{Thr}_{\mathrm{base}}}. \tag{89}$$

When throughput is approximated as inverse latency, this becomes

$$\mathrm{ThrChange} = 100 \left(\frac{1}{1 + \mathrm{LatOH}/100} - 1 \right). \tag{90}$$

Let S_{base} and S_{rep} denote the main downstream score, which is MRR for predictive personalization in our deployment table. We define the score/latency and score/memory ratios as

$$\text{MRR-GLR} = \frac{S_{\mathrm{rep}}/S_{\mathrm{base}}}{1 + \mathrm{LatOH}/100}, \tag{91}$$

$$\text{MRR-GMR} = \frac{S_{\mathrm{rep}}/S_{\mathrm{base}}}{1 + \mathrm{MemOH}/100}. \tag{92}$$

Values close to 1 indicate that the repaired model preserves nearly the same score-efficiency as the frozen host while improving absolute downstream accuracy. Much smaller values indicate that the gain is obtained at disproportionate deployment cost.

E Task-specific Baselines

This section describes the frozen host models and task-specific baselines used in the experiments. These baselines are not treated as correction methods. They are the host encoders or task systems to which `REPAIR` is attached post hoc.

E.1 Movie Recommendation Baselines

SIGMA. SIGMA is a Mamba-style sequential recommender with selective gating for balancing long-range sequence modeling and short-range behavioral cues. We include SIGMA as a strong modern sequence encoder whose terminal state is already highly optimized, making it a useful test of whether `REPAIR` can still recover additional preference evidence.

Mamba4Rec. Mamba4Rec is a sequential recommendation model built around selective state-space modeling rather than standard self-attention. It is used as a primary host because it is both strong and architecturally distinct from Transformer-style recommenders.

SSDRec. SSDRec is a self-augmented sequence denoising recommender. It assumes that user histories contain spurious or uninformative interactions and learns to denoise the sequence representation before next-item prediction. We include it to test whether state repair helps even when the host already performs denoising internally.

SASRec. SASRec is a self-attentive sequential recommender that predicts the next item by attending over previous interactions. It is used as a standard Transformer-style sequential recommendation baseline.

TiSASRec. TiSASRec extends SASRec by incorporating time intervals between interactions. This makes it a strong time-aware host for testing whether REPAIR can recover preference evidence beyond what is already captured by explicit temporal spacing.

HSTU. HSTU is a high-capacity sequential transduction architecture designed for large-scale recommendation. It represents a strong long-horizon modeling regime different from both classical attention and SSM-based recommenders.

FEARec. FEARec augments time-domain sequence modeling with frequency-domain preference signals. It is included to test whether REPAIR can recover residual preference evidence when the host already models periodic or recurring behavior.

GRU4Rec. GRU4Rec is a recurrent sequential recommender based on gated recurrent units. We include it as a compact recurrent host and as a weaker transfer target for testing plug-and-play repair.

BERT4Rec. BERT4Rec applies bidirectional Transformer encoding to recommendation using a Cloze-style masking objective. It is included because its bidirectional compression behavior differs from autoregressive, recurrent, and state-space recommenders.

E.2 News Recommendation Baselines

NAML. NAML is a neural news recommendation model with attentive multi-view learning over news content. It combines views such as title, body, and category, and aggregates clicked-history representations into a user state.

EBNR. EBNR is an embedding-based news recommender with an RNN-style user encoder over browsing history. Its compact recurrent user state makes it especially relevant for testing whether REPAIR can recover evidence weakened by recurrent compression.

NRMS. NRMS uses multi-head self-attention for both news encoding and user-history encoding. It is included as a strong self-attentive news recommendation host.

LSTUR. LSTUR explicitly separates long-term and short-term user interests. It is a strong host for testing whether REPAIR still helps when the base model already contains a long/short decomposition.

MINER. MINER represents users through multiple interest vectors rather than a single pooled user embedding. It is included as a strong multi-interest news recommendation baseline.

TrRMIo. TrRMIo is included as a strong news recommendation host under its original training and encoder protocol. In our setting, it is not a corrective competitor; REPAIR is attached only after the frozen host state is produced.

ICL-based LLM Baselines. We include in-context learning baselines based on DeepSeek-Qwen2.5-Base-32B and Qwen2.5-32B. These baselines personalize through prompt conditioning on user history rather than through an explicit exposed user-state encoder.

E.3 Product Recommendation Baselines

SigLIP2. SigLIP2 is used as a frozen vision-language product encoder. It represents products from multimodal product evidence and tests whether REPAIR can recover user-preference evidence from compressed multimodal product states.

OpenCLIP. OpenCLIP is used as another frozen vision-language product encoder. It provides a contrastive multimodal representation space for candidate product matching.

DINOv2-E5. DINOv2-E5 combines self-supervised visual features with E5 text embeddings. It is included to test repair on a heterogeneous product representation that combines image-side and text-side information.

E.4 Personalized Headline Generation Baselines

PENS with NAML-T1/T2. These variants instantiate the PENS personalized headline generation framework with NAML as the user-preference encoder. They represent attentive multi-view history encoding followed by personalized generation.

PENS with EBNR-T1/T2. These variants instantiate the PENS framework with EBNR as the user encoder. They represent recurrent history compression followed by headline generation.

PENS with NRMS-T1/T2. These variants instantiate the PENS framework with NRMS as the user-history encoder. They represent self-attentive user modeling for personalized headline generation.

GTP. GTP, or General Then Personal, separates general headline generation from personalization through decoupling and pre-training. It is included as a strong specialized personalized headline generation baseline.

SCAPE. SCAPE is a stylistic-content aware personalized headline generation model. It explicitly considers that user preferences in headlines may involve both topical content and stylistic presentation.

FPG. FPG, or Fact-Preserved Personalized News Headline Generation, emphasizes preserving factual consistency while still adapting the generated headline to user preference.

Walk2Pers. Walk2Pers is a personalized summarization host based on action-conditioned preference evolution. It updates user preference through structured interaction steps that separate shift magnitude from shift orientation, and uses memory components to preserve consistent interests while suppressing disinterest. We include Walk2Pers as a preference-coupled generation host because its decoder is designed to use an evolving user state, making it a useful test of whether repairing the exposed preference state improves personalization-sensitive generation. In our setting, the Walk2Pers host remains frozen; only `REPAIR` and the final decoder/head-alignment phase are trained.

IMPerSumm. IMPerSumm is an information-modulated personalized summarization model that represents user preference through interaction-conditioned evidence rather than relying only on a static persona or long-context prompting. It is included as a strong preference-coupled summarization host because its generation process is explicitly conditioned on user-side preference evidence. In our setting, the IMPerSumm preference encoder and generator backbone are treated as the frozen host interface. `REPAIR` is attached to the exposed preference state and learns a pre-head correction before the summarization decoder consumes it.

Prompt-conditioned LLM Summarizers. DeepSeek-32B and Qwen2.5-32B are used as frozen prompt-conditioned summarization baselines. The user history and task-side input are serialized into the prompt, and the LLM backbone is not fine-tuned. When `REPAIR` is attached to these hosts, the repaired preference state is mapped into a compact continuous prefix/soft-prompt representation as described in Appendix A.8. These baselines test whether explicit state repair adds useful preference evidence beyond direct history prompting in large frozen decoders.

E.5 Conversational Recommendation and Generation Baselines

SASRec for TG-ReDial Recommendation. SASRec is used as a self-attentive conversational recommendation host over dialogue-induced interaction sequences.

GRU4Rec for TG-ReDial Recommendation. GRU4Rec is used as a recurrent conversational recommendation host. It provides a compact sequential baseline whose state can be repaired post hoc.

TextCNN for TG-ReDial Recommendation. TextCNN is used as a lightweight convolutional baseline that captures local interaction patterns without explicit long-horizon recurrence or self-attention.

ReDial for Response Generation. ReDial-style response generation is included as a conversational generation baseline representing earlier neural conversational recommendation systems.

Transformer-based Response Generation. A Transformer generator is used as a generic sequence-to-sequence response generation baseline. It conditions on the same conversational context but does not perform selective preference-state repair.

F Corrective Baseline Details

This section describes the post-head corrective baselines used for RQ2. Both baselines are attached to the same frozen hosts and are trained under the same task protocol as `REPAIR`. Their purpose is to test whether generic post-head correction is sufficient, or whether pre-head compressed-state repair is necessary.

F.1 Rec-Denoiser-Corr

Rec-Denoiser-Corr adapts denoising-style sequential recommendation to the frozen-host corrective setting. The original denoising principle is to suppress noisy or target-irrelevant historical dependencies before prediction. In our adaptation, Rec-Denoiser-Corr does not modify the frozen host encoder and does not repair the exposed compressed preference state directly.

Given the frozen host's prediction-side context, Rec-Denoiser-Corr scores historical timesteps according to their estimated utility for the current output. Low-utility or weakly relevant trajectory links are down-weighted through an attention-modulation layer. Let α_j denote the original influence of timestep j and let $m_j \in [0, 1]$ denote the learned denoising mask. The corrected influence is

$$\tilde{\alpha}_j = m_j \alpha_j. \tag{93}$$

The resulting post-head corrective context is then formed from the retained trajectory evidence:

$$c_{\text{denoise}} = \sum_{j=1}^{i} \tilde{\alpha}_j e_b^{(t_j)}. \tag{94}$$

The downstream prediction or generation head receives this denoised context as an additional corrective signal. For attentive hosts, the influence weights are the host's native trajectory-to-context attention weights. For non-attentive hosts such as SSM and recurrent encoders, we instantiate the same baseline using post-hoc candidate-score sensitivity over cached timestep states, normalized across the active prefix. Thus, `Rec-Denoiser-Corr` is implemented as an output-side influence suppressor for all hosts, rather than assuming that every host exposes native attention weights. This baseline tests whether suppressing noisy historical evidence after observing the host prediction context is enough to match `REPAIR`. The key distinction is that Rec-Denoiser-Corr treats the trajectory-to-output influence pattern as the correction target, whereas `REPAIR` treats the compressed preference state $z_b^{(t_i)}$ itself as the object to be repaired before the task head consumes it.

F.2 SERAC-Corr

SERAC-Corr adapts memory-based model editing to the frozen-host personalization setting. SERAC-style correction stores edit examples in an explicit memory, uses a learned scope selector to decide whether a test instance should be corrected, and routes selected cases to a counterfactual correction model.

In our adaptation, the frozen host's prediction-side context is used as the query to an edit memory. During training, cases where correction is useful are stored as memory entries. Each memory entry contains a query representation, corrective evidence, and the associated task target. At inference time, the current prediction context q_{ctx} retrieves candidate edits:

$$\mathcal{M}(q_{\text{ctx}}) = \text{TopK}_{m \in \mathcal{M}} \, \text{sim}(q_{\text{ctx}}, q_m). \tag{95}$$

A learned scope selector decides whether the retrieved memory should override or modify the host's base decision:

$$p_{\text{scope}} = \sigma \left(f_{\text{scope}}(q_{\text{ctx}}, \mathcal{M}(q_{\text{ctx}})) \right). \tag{96}$$

When the selector activates, a counterfactual correction model produces an alternative corrective state or prediction context:

$$\tilde{z}_{\text{SERAC}} = f_{\text{cf}} \left(q_{\text{ctx}}, \mathcal{M}(q_{\text{ctx}}) \right). \tag{97}$$

The downstream task head then uses this retrieved corrective state or context for the final decision.

SERAC-Corr is a strong memory-based post-head corrector because it can observe prediction-side context and use explicit corrective memory. However, it does not perform state-relative temporal repair inside the compressed preference state. It relies on an auxiliary selector, edit memory, and counterfactual correction model at inference time, while REPAIR constructs a compact correction directly from cached per-timestep evidence relative to the exposed state.

F.3 Comparison to REPAIR

Both Rec-Denoiser-Corr and SERAC-Corr operate after the host has already produced a prediction-side context or output-facing representation. They therefore test whether generic post-head correction is sufficient. In contrast, REPAIR intervenes before the downstream task head. Given the frozen host state $z_b^{(t_i)}$ and cached event representations $\{e_b^{(t_j)}\}_{j=1}^{i}$, REPAIR estimates state-relative corrective evidence, resolves it through long-support, tail-local, and episodic kernels, retains only the strongest timestep-level signals, and fuses the resulting correction back into the exposed preference state:

$$\tilde{z}_b^{(t_i)} = F_{\mathrm{rep}}\left(z_b^{(t_i)}, A_{\mathrm{rep}}\left(\left\{\Pi_\nu^{(t_i)}\left(G_{\mathrm{rep}}\left(D_{\mathrm{rep}}(z_b^{(t_i)}, e_b^{(t_j)}), \kappa(t_i - t_j)\right)\right)\right\}_{j=1}^{i}\right)\right). \tag{98}$$

Thus, the conceptual difference is the intervention target. Rec-Denoiser-Corr corrects trajectory influence, SERAC-Corr corrects through retrieved edit memory, and REPAIR repairs the downstream-facing compressed preference state itself.

G Additional Results And Extended Analysis

G.1 Full Predictive Recommendation Results Across Frozen Hosts

We observe that the predictive gains of REPAIR are not concentrated in a single encoder family. On MovieLens, REPAIR improves every direct frozen host in Table 9. The gains are largest for hosts whose exposed state appears to leave more residual preference evidence recoverable from the trajectory. For example, Mamba4Rec improves by $+4.97$ MRR and $+5.66$ Hit@10, SSDRec improves by $+3.28$ MRR, and FEARec improves by $+2.59$ MRR and $+8.18$ Hit@10. These improvements are larger than the gains on already competitive or explicitly time-aware hosts such as TiSASRec and SIGMA, where the MRR gains are $+1.27$ and $+1.54$, respectively. This pattern is consistent with the intended role of REPAIR: it does not replace the frozen encoder, but corrects the part of the compressed state that remains under-expressed after the host has already performed its own temporal aggregation.

We use the transfer rows in Table 9 as an additional diagnostic. A REPAIR module trained on Mamba4Rec still improves GRU4Rec and BERT4Rec, with gains of $+5.31$ and $+6.39$ MRR, respectively. The corresponding nDCG@10 gains are $+6.19$ and $+8.13$, and the Hit@10 gains are $+9.33$ and $+13.09$. These results suggest that the learned correction is not purely memorizing one host's output distribution. Instead, it can transfer when the target host exposes a compressed preference state whose missing evidence is compatible with the repair interface. This does not imply architecture-independent universality, but it supports the claim that state-relative residual evidence can behave as a reusable corrective signal across related sequential encoders.

We observe a different regime for structured PENS recommendation in Table 10. Several base encoders have very low starting scores, and the repaired model produces large absolute gains. NAML improves from 1.29 to 12.38 MRR, NRMS from 1.18 to 12.11, and EBNR from 2.65 to 11.97. The same trend appears in top-10 retrieval: NAML improves by $+24.36$ Hit@10, NRMS by $+24.06$, and EBNR by $+19.36$. These gains indicate that the structured PENS histories contain substantial user-specific evidence that is not sufficiently expressed in the original compressed state. In contrast, LSTUR starts from the highest base MRR among the standard news encoders and receives a smaller but still positive gain of $+2.41$ MRR. This is expected because LSTUR already encodes a long/short user decomposition, leaving less unmodeled temporal structure for a repair module to recover.

We use MIND in Table 11 as a cleaner high-base news recommendation setting. All seven hosts improve under REPAIR. The largest MRR gains occur for NRMS, TrRMIo, and NAML, with $+7.83$, $+7.78$, and $+7.57$, respectively. The nDCG@10 gains are even larger for NAML and NRMS, reaching

Table 8: REPAIR parameters and hyperparameters.

REPAIR component	Symbol	Dimension / Value
Core dimensions	d_z	768
	d_e	768
	m	192
	K	128
	H	50
	i	≤ 50
Shared repair-space projection	P_z	$\mathbb{R}^{192\times 768}$
	P_e	$\mathbb{R}^{192\times 768}$
Low-rank repair operator	V_{rep}	$\mathbb{R}^{192\times 128}$
	U_{rep}	$\mathbb{R}^{192\times 128}$
	Λ_{rep}	$\mathbb{R}^{128\times 128}$, diagonal
State-relative corrective evidence	$\tilde{z}_b^{(t_i)} = P_z z_b^{(t_i)}$	$\mathbb{R}^{192}$
	$\bar{e}_b^{(t_j)} = P_e e_b^{(t_j)}$	$\mathbb{R}^{192}$
	$d^{(t_j)} = V_{\text{rep}}^\top(\bar{e}_b^{(t_j)} - \tilde{z}_b^{(t_i)})$	$\mathbb{R}^{128}$
	$\widetilde{d}^{(t_j)}$	$\mathbb{R}^{128}$
Multi-regime temporal resolution	β	$\mathbb{R}_+^{128}$
	ξ	$\mathbb{R}_+^{128}$
	$\kappa^L(\ell_j)$	$\mathbb{R}^{128}$
	$\kappa^S(\ell_j)$	$\mathbb{R}^{128}$
	$\kappa^E(\ell_j)$	$\mathbb{R}^{128}$
Regime-resolution matrices	W_L	$\mathbb{R}^{128\times 128}$
	W_S	$\mathbb{R}^{128\times 128}$
	W_E	$\mathbb{R}^{128\times 128}$
Adaptive top-ν retention	$s_k^{(t_i)}$	$\mathbb{R}^3$
	MLP_{sel}	$\mathbb{R}^3 \to \mathbb{R}$
	$\nu_k^{(t_i)}$	integer in $[1, i]$
	$\{\pi_{j,k}^{(t_i)}\}_{j,k}$	$\{0,1\}^{i\times 128}$
Correction aggregation	$\gamma^{(t_i)}$	$(0,1)^{128}$
	$\{\alpha_j^{(t_i)}\}_{j=1}^i$	$\mathbb{R}^i$, softmax-normalized
	P_Δ	$\mathbb{R}^{768\times 192}$
	$\Delta_b^{(t_i)}$	$\mathbb{R}^{768}$
Residual-gated fusion	$g_F^{(t_i)}$	$(0,1)^{768}$
	MLP_F	$\mathbb{R}^{1536} \to \mathbb{R}^{768}$
	$\text{MLP}_{\text{probe}}$	$\mathbb{R}^{1536} \to \mathbb{R}^{768}$
	$\widetilde{z}_b^{(t_i)}$	$\mathbb{R}^{768}$
Auxiliary objective terms	W_{pos}	$\mathbb{R}^{50\times 128}$
	P_{coh}	$\mathbb{R}^{128\times 768}$
	τ	learned scalar
Loss weights	λ_{task}	0.5
	λ_{pos}	0.2
	λ_{coh}	0.3

$+16.62$ and $+16.71$. This indicates that REPAIR does more than occasionally move the positive item upward by a small amount; it substantially improves top-ranked retrieval quality for several news encoders. LSTUR again has the highest base MRR and receives a smaller MRR gain of $+4.41$, but its nDCG@10 still increases by $+7.29$. Thus, even when the base encoder already separates long- and short-term interests, the exposed state can still benefit from state-relative correction.

We observe the largest relative correction effect on Amazon Reviews 2023 in Table 12. SigLIP2, OpenCLIP, and DINOv2-E5 improve by $+15.33$, $+17.91$, and $+14.96$ MRR, respectively. The Hit@10 gains are also large: $+28.44$, $+32.24$, and $+27.55$. These results are important because the hosts are multimodal product encoders rather than standard sequential recommenders. A likely explanation is that frozen multimodal representations are effective at product semantics but do not by themselves fully encode the evolving user-side preference state. REPAIR can therefore recover user-specific residual evidence from the history and make the product-preference state more useful for ranking.

G.2 Generative Accuracy And Personalization-Sensitive Metrics

We observe from Tables 13 and 14 that standard generation metrics and personalization-sensitive metrics do not always move in the same way. This distinction is central to interpreting REPAIR on generative personalization. For the LLM-based systems in Table 13, the largest gains appear on standard lexical and semantic overlap metrics. DeepSeek-32B improves by $+30.16\%$ ROUGE-1, $+53.21\%$ ROUGE-L, $+171.27\%$ ROUGE-SU4, and $+45.49\%$ METEOR. Qwen2.5-32B follows a similar pattern, with $+26.80\%$ ROUGE-1, $+41.58\%$ ROUGE-L, $+119.17\%$ ROUGE-SU4, and $+47.64\%$ METEOR. However, these large surface-form gains do not automatically imply the largest user-specific personalization gains. The corresponding PerSEval gains are positive but more moderate for DeepSeek-32B, especially on PSE-W-RG-L, where the gain is $+5.05\%$.

The contrast is clearest for preference-coupled generators. Walk2Pers has modest standard overlap improvements in Table 13, with $+5.18\%$ ROUGE-1 and $+6.58\%$ ROUGE-L, but its personalization-sensitive gains in Table 14 are larger: $+13.25\%$ PSE-W-JSD, $+15.32\%$ PSE-W-RG-L, $+10.76\%$ PSE-P-RG-L, and $+19.13\%$ PSE-METEOR. IMPerSumm is even more diagnostic. Its standard metrics are mixed, including small or negative changes on ROUGE-L, W-RG-1, and P-RG-L, yet its PerSEval gains are large: $+20.76\%$ PSE-W-JSD, $+25.23\%$ PSE-W-RG-L, and $+39.29\%$ PSE-P-RG-L. This indicates that REPAIR can improve user-specific responsiveness even when conventional reference-overlap metrics do not increase strongly.

The encoder-based PENS variants also give consistent personalization gains. EBNR-T1/T2, NAML-T1/T2, and NRMS-T1/T2 all improve across most PerSEval variants. The PSE-P-RG-L gains are especially large for these models, often exceeding $+39\%$. However, their absolute PSE-METEOR values remain small, and in a few cases the relative gain is 0.00%. This suggests that the repaired state is more effective at shifting paraphrase-aware or weighted personalization structure than at producing large changes in exact METEOR-style alignment. Overall, these patterns support a decoder-utilization interpretation: REPAIR supplies improved preference evidence, but the realized output gain depends on whether the downstream generator uses that evidence while decoding.

G.3 Additional Interpretation Of Time-Scale Resolution Ablations

We observe from Table 16 that explicit long-, short-, and episodic-support resolution is necessary for reliable repair. The no-time-scale variant, REPAIR-NoTS, is consistently weaker than the full model and can even underperform the frozen base. For Mamba4Rec, REPAIR-NoTS gives 28.43 MRR compared with the base score of 29.29, indicating that a generic correction without temporal resolution may inject poorly organized residual evidence. This supports the design choice of resolving the state-relative residual field through temporal kernels before aggregation.

The full $\kappa^L + \kappa^S + \kappa^E$ variant is the best configuration for every host. On MovieLens, it improves Mamba4Rec to 34.26, TiSASRec to 36.78, and SIGMA to 34.81. On MIND, it improves NAML to 53.22, EBNR to 53.22, and NRMS to 52.74. Partial combinations help, but they do not match the full model. For example, Mamba4Rec reaches 32.17 under $\kappa^L + \kappa^S$, but reaches 34.26 only with all three regimes. NAML reaches 49.31 under $\kappa^L + \kappa^S$, but 53.22 with the full model. These gaps indicate that the three temporal branches are complementary rather than redundant.

The best partial or single branch differs by host, which is also informative. Mamba4Rec and SIGMA, both state-space-style hosts, benefit from mixtures involving long- and short-support evidence. NRMS has a high episodic single-branch score, suggesting that some Transformer-based news states leave burst-local evidence recoverable after compression. NAML and EBNR obtain larger gains from multi-branch combinations than from any single branch, consistent with additive-attention and recurrent compression leaving multiple kinds of residual evidence. Thus, the relevant correction timescale is not simply the raw timescale of the input event; it is the timescale of what remains missing after the frozen host has compressed the trajectory.

G.4 Training Objective Ablation

We observe from Table 17 that each repair loss contributes to the final result. The task-only variant already improves over the frozen base, confirming that direct downstream supervision is sufficient to learn a useful correction direction. For example, Mamba4Rec improves from 29.29 to 31.43 MRR

with only $\mathcal{L}_{task}$, and NRMS improves from 44.91 to 46.39. However, the auxiliary losses provide substantial additional gains.

Adding positional anchoring $\mathcal{L}_{pos}$ helps prevent the correction field from collapsing into an unstructured aggregate. This is especially visible for the news hosts. LSTUR improves from 52.17 to 54.65 after adding $\mathcal{L}_{pos}$, and NRMS improves from 46.39 to 51.29. These jumps suggest that timestep identity and temporal anchoring matter strongly when news histories contain rapid topic transitions.

Adding coherence $\mathcal{L}_{coh}$ gives the best final result for every host. The effect is particularly large for Mamba4Rec, which improves from 32.16 to 34.26 after adding coherence, and for NRMS, which improves from 51.29 to 52.74. This supports the role of coherence as an event-level constraint: the correction associated with a timestep should remain aligned with the source event representation, rather than merely becoming a task-optimized but weakly grounded state perturbation. These patterns empirically support the loss ordering used in the method. Task utility is primary, positional anchoring prevents temporal ambiguity, and coherence keeps the retained correction tied to its source evidence.

G.5 Length-Controlled Selective Retention

We observe from Table 18 that selective retention becomes most useful when longer histories introduce more redundant or state-misaligned evidence. Across all four representative hosts, the base score generally increases with longer histories, but the repaired variants consistently provide additional benefit. At the maximum history length $l = 50$, selective retention outperforms the non-selective variant for every host. The MRR improvements of SEL over NS at $l = 50$ are $+0.31$ for Mamba4Rec, $+0.22$ for TiSASRec, $+1.05$ for NAML, and $+1.11$ for EBNR. The nDCG@10 improvements are $+0.14$, $+0.25$, $+3.43$, and $+2.44$, respectively.

The selected timestep counts indicate that REPAIR does not collapse to a very small number of events. At $l = 50$, the selector retains evidence from 34 timesteps for Mamba4Rec, 36 for TiSASRec, 36 for NAML, and 35 for EBNR. Thus, selective retention is not aggressive pruning of the history. It is a state-relative filtering mechanism that removes a minority of weakly corrective or redundant timesteps while keeping a broad support of useful evidence.

The intermediate-length rows also indicate that selection is not uniformly better than non-selection at every prefix length. For example, on Mamba4Rec at $l = 30$ and $l = 40$, the non-selective variant is slightly stronger than SEL on MRR. A similar small reversal appears for TiSASRec at $l = 20$. These cases are useful because they show that selective retention is not simply adding capacity. Its advantage becomes clearer when the history is long enough to contain more redundant or state-misaligned evidence. This is why the full-length setting is the most relevant stress case for the selector: when more historical evidence is available, filtering the corrective residual field becomes more valuable.

G.6 Noisy-Trajectory Stress Test

We observe from Table 19 that REPAIR remains useful when trajectories contain abrupt disconnected content insertions. The noisy base scores are substantially lower than the clean test scores, indicating that the perturbation meaningfully disrupts the frozen hosts. Despite this, REPAIR improves all four noisy-host settings. On MovieLens, Mamba4Rec improves from 18.21 to 21.43 MRR and TiSASRec improves from 22.34 to 24.65. On MIND, NAML improves from 28.61 to 32.85 and EBNR improves from 29.63 to 31.65.

The nDCG@10 gains under noise are also positive: $+1.80$ for Mamba4Rec, $+2.25$ for TiSASRec, $+6.58$ for NAML, and $+2.73$ for EBNR. The larger gain for NAML suggests that additive or pooled history representations are more vulnerable to disconnected noisy subsequences and therefore benefit more from state-relative residual selection. These results do not imply that REPAIR fully removes the effect of noisy histories. Rather, they indicate that even after the frozen state has been degraded by noisy insertions, the cached timestep evidence still contains recoverable signal that can improve the exposed preference state.

G.7 Serving-Path And Compute-Footprint Interpretation

We clarify the deployment-overhead interpretation using Table 20, which reports normalized online work at history length $T = 50$. Mamba4Rec has 0.75 GFLOPs/query base work and receives 0.18

added GFLOPs/query from REPAIR, while TiSASRec has 1.45 base GFLOPs/query and receives the same 0.18 added work. Thus, the same repair module can have different relative overhead depending on the host's original serving cost. NAML has a much smaller absolute base work of 0.03 GFLOPs/query and an added work of 0.02, which makes the repair cost more visible in relative terms even though the absolute cost is small.

G.8 Broader Applicability On TG-ReDial

TG-ReDial is used as a broader-applicability stress test rather than as primary evidence because dialogue histories do not naturally expose the same clean action-content structure as click, rating, skip, or purchase logs. Dialogue trajectories contain turns, mentioned entities, recommendation attempts, acceptance or rejection signals, topic shifts, and implicit stance. As a result, the UIG eventization is induced rather than directly observed. This makes TG-ReDial a useful test of whether REPAIR can still help under imperfect event construction, but it should not be interpreted as the cleanest setting for validating structured state repair.

We observe from the recommendation block in Table 15 that REPAIR is most useful for sequential dialogue-derived hosts. SASRec improves from 4.11 to 4.56 MRR and from 7.82 to 8.76 HR@10. GRU4Rec improves from 4.14 to 4.56 MRR and from 8.80 to 9.13 HR@10. These gains are smaller than those observed in MovieLens, MIND, PENS, or Amazon, but they are directionally consistent: when the host exposes a compressed sequential state, state-relative correction can still recover useful evidence even from noisier dialogue-derived events.

TextCNN behaves differently. Its MRR increases from 2.57 to 2.78, but HR@10 decreases from 5.08 to 5.04, nDCG@5 decreases from 1.92 to 1.73, and nDCG@10 decreases from 2.55 to 2.51. This mixed result is important because TextCNN is a local-pattern encoder rather than a long-horizon sequential state model. Since REPAIR is designed to repair an exposed compressed preference state using timestep-level residual evidence, it is less effective when the host representation does not provide a rich sequential state for correction. The TextCNN result therefore reinforces the scope of the method: REPAIR is not a universal post-processing layer for any architecture, but a state-repair module whose benefit depends on the quality and structure of the host state.

We observe the same decoder-utilization boundary in the conversational generation block that appears in PENS. ReDial improves substantially, with ROUGE-1 increasing from 12.33 to 27.76, ROUGE-L from 11.17 to 23.52, and ROUGE-SU4 from 3.31 to 10.54. In contrast, the generic Transformer generator changes only marginally: ROUGE-1 increases from 31.81 to 31.97, ROUGE-L from 26.46 to 26.48, and ROUGE-SU4 from 12.62 to 12.63. This indicates that repaired preference evidence is useful only when the downstream generator is coupled strongly enough to the user state. If the generator mainly relies on generic dialogue context and weakly uses the repaired preference state, the repair signal has little room to affect the final output.

Overall, TG-ReDial supports broader applicability but also exposes a limitation. REPAIR can help beyond clean recommendation logs, especially for sequential hosts and user-state-coupled generators, but the gains are smaller and more architecture-dependent. A stronger dialogue-specific extension would require discourse-aware eventization, including dialogue acts, entity mentions, recommendation intent, acceptance/rejection labels, topic-shift markers, and confidence-weighted stance induction. This is left for future work because it changes the quality of the interaction graph itself, whereas the present paper focuses on repairing compressed states once an eventized trajectory is available.

H License and Usage Statement

Pre-trained and open-weight model assets. When REPAIR is attached to an LLM host, the LLM backbone remains frozen; only the REPAIR module and the small soft-prompt/state-projection component are trained. We use the following pre-trained or open-weight assets under their corresponding licenses and provider terms:

- **DeepSeek-R1-Distill-Qwen-32B / DeepSeek-32B-family baselines.** We use the DeepSeek-R1 distilled Qwen-family model checkpoints under the MIT License specified by DeepSeek AI: `https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-32B` and `https://github.com/deepseek-ai/DeepSeek-R1/blob/main/LICENSE`.

- **Qwen2.5-32B-family baselines.** We use Qwen2.5-32B-family checkpoints under the Apache License 2.0, following the Qwen model release terms: https://huggingface.co/Qwen/Qwen2.5-32B and https://huggingface.co/Qwen/Qwen2.5-32B-Instruct/blob/main/LICENSE. We do not use Qwen2.5-3B or Qwen2.5-72B, which are governed by different Qwen-specific terms.
- **T5-family text encoders.** We use T5-style text representations for event/content encoding under the Apache License 2.0: https://huggingface.co/google-t5/t5-base.
- **CLIP and OpenCLIP image-text encoders.** We use CLIP-style and OpenCLIP-style image-text representations for multimodal product evidence. OpenAI CLIP is released under the MIT License: https://github.com/openai/CLIP/blob/main/LICENSE. OpenCLIP is also distributed under an MIT-compatible license: https://github.com/mlfoundations/open_clip.
- **SigLIP2.** We use SigLIP2 as a frozen vision-language product encoder under the Apache License 2.0: https://huggingface.co/google/siglip2-base-patch16-224 and https://huggingface.co/google/siglip2-base-patch16-256.
- **DINOv2.** We use DINOv2 visual representations as part of the DINOv2-E5 product encoder setting. DINOv2 code and model weights are released under the Apache License 2.0: https://github.com/facebookresearch/dinov2.
- **E5 text encoder.** We use E5 text embeddings in the DINOv2-E5 product encoder setting under the MIT License: https://huggingface.co/intfloat/e5-base-v2.

Task-specific host models and baselines. The recommendation and generation hosts used in the paper, including sequential recommenders, news recommenders, product encoders, and personalized headline generators, are used as research baselines. Unless otherwise stated, these hosts are trained under their original task supervision on the corresponding benchmark data and then frozen before REPAIR is attached. We do not redistribute trained host checkpoints from third-party authors. When public implementations are used, we follow the corresponding repository licenses and cite the original papers.

Datasets. We use all datasets for research purposes only and follow the access conditions, usage terms, and citation requirements specified by the dataset providers:

- **MovieLens.** We use MovieLens for sequential movie recommendation. MovieLens datasets are distributed by GroupLens with usage terms specified in the README files for each dataset version. GroupLens states that users should review the README files for license and usage details and that public redistribution is generally not permitted without permission. We therefore do not redistribute MovieLens data and use it only for research evaluation: https://grouplens.org/datasets/movielens/.
- **MIND.** We use MIND for news recommendation. MIND is distributed for research purposes under Microsoft Research License Terms. We follow the dataset access conditions and do not redistribute the raw data: https://msnews.github.io/ and https://github.com/msnews/MIND/blob/master/MSR%20License_Data.pdf.
- **PENS.** We use PENS for structured news recommendation and personalized headline generation. PENS is distributed for research purposes under Microsoft Research License Terms. We follow the official download and usage terms and do not redistribute the raw data: https://msnews.github.io/pens.html.
- **Amazon Reviews 2023.** We use the Amazon Reviews 2023 All_Beauty subset for multimodal product recommendation. The official McAuley-Lab release states that the dataset is made available primarily for research purposes; however, the maintainers also state that they are not in a position to assign a formal license to the dataset or dictate its usage terms. We therefore treat this dataset as a research-only benchmark, cite the original release, do not redistribute the raw review or metadata files, and require users to obtain the dataset from the official provider: https://amazon-reviews-2023.github.io/ and https://huggingface.co/datasets/McAuley-Lab/Amazon-Reviews-2023.
- **TG-ReDial.** We use TG-ReDial as a broader-applicability stress test for conversational recommendation and response generation. The official TG-ReDial repository is

Table 9: Predictive personalization on MovieLens under strict evaluation with 499 negative samples per test instance. "Direct" denotes host-wise REPAIR attached to the same frozen backbone, while "Transfer" denotes plug-and-play transfer from REPAIR trained on Mamba4Rec and attached to weaker frozen hosts (all values are in 1-100 scale).

Version	Host Encoder	MRR			nDCG@5			nDCG@10			HR@10		
		Base	+REPAIR	Δ	Base	+REPAIR	Δ	Base	+REPAIR	Δ	Base	+REPAIR	Δ
Direct	Mamba4Rec	29.29	34.26	+4.97	30.07	35.56	+5.49	38.50	40.17	+1.67	57.90	63.56	+5.66
	TiSASRec	**35.51**	**36.78**	+1.27	**36.87**	**38.41**	+1.54	**41.14**	**42.91**	+1.77	**63.66**	**66.85**	+3.19
	HSTU	29.00	30.62	+1.62	29.69	31.55	+1.86	34.23	36.11	+1.88	56.40	58.71	+2.31
	FEARec	28.03	30.62	+2.59	28.20	31.55	+3.35	32.13	36.11	+3.98	50.53	58.71	+8.18
	SIGMA	33.27	34.81	+1.54	34.43	36.26	+1.83	38.96	40.72	+1.76	62.03	64.04	+2.01
	SASRec	26.43	28.49	+2.06	29.27	32.63	+3.36	33.82	35.18	+1.36	52.17	54.34	+2.17
	SSDRec	30.15	33.43	+3.28	33.25	35.22	+1.97	40.17	42.31	+2.14	59.32	63.11	+3.79
Transfer	GRU4Rec	17.84	23.15	+5.31	17.10	22.78	+5.68	21.07	27.26	+6.19	38.04	47.37	+9.33
	BERT4Rec	20.80	27.19	**+6.39**	20.44	27.70	**+7.26**	24.61	32.74	**+8.13**	43.07	56.16	**+13.09**

Table 10: Structured news recommendation on structured PENS test. Here, +REPAIR denotes REPAIR with 2-epoch head finetuning (all values are in 1-100 scale).

Host Encoder	MRR			nDCG@5			nDCG@10			Hit@10		
	Base	+REPAIR	Δ	Base	+REPAIR	Δ	Base	+REPAIR	Δ	Base	+REPAIR	Δ
EBNR	2.65	11.97	+9.32	1.82	12.68	+10.86	2.45	14.28	+11.83	4.87	24.23	+19.36
NAML	1.29	12.38	+11.09	0.43	13.16	+12.73	0.81	15.13	+14.32	2.01	26.37	+24.36
NRMS	1.18	12.11	+10.93	0.39	12.89	+12.50	0.74	14.92	+14.18	1.92	25.98	+24.06
TrRMIo	8.02	12.86	+4.84	7.85	13.97	+6.12	8.53	15.87	+7.34	12.77	27.67	+14.90
SCAPE	1.57	8.34	+6.77	0.76	7.91	+7.15	1.16	9.52	+8.36	2.53	16.84	+14.31
LSTUR	**10.44**	12.85	+2.41	**10.87**	12.49	+1.62	**12.15**	15.88	+3.73	**20.96**	28.73	+7.77
MINER	5.17	8.64	+3.47	4.71	8.83	+4.12	5.61	10.31	+4.70	9.77	18.88	+9.11
Walk2Pers	6.23	13.54	+7.31	8.46	**15.28**	+6.82	9.71	16.16	+6.45	11.35	20.35	+9.00
IMPerSumm	1.70	**13.64**	**+11.94**	0.79	15.19	**+14.40**	1.28	**17.11**	**+15.82**	2.86	**30.12**	**+27.26**
DeepSeek-32B	1.31	1.83	+0.52	0.28	0.98	+0.70	0.64	1.32	+0.68	1.53	2.41	+0.88
Qwen2.5-32B	1.24	1.79	+0.55	0.26	0.94	+0.68	0.59	1.27	+0.68	1.47	2.35	+0.88

distributed under the Apache License 2.0. We follow the repository license and cite the original dataset paper: https://github.com/RUCAIBox/TG-ReDial and https://github.com/RUCAIBox/TG-ReDial/blob/main/LICENSE.

Privacy and responsible use. All datasets are used only in their benchmark/research form. We do not attempt to identify users, reconstruct private identities, or combine datasets to deanonymize individuals. Train/test splits are kept separate as described in the evaluation protocol. For datasets with user interaction histories, we use the interaction records only to evaluate preference modeling and do not release raw user histories, raw text, or derived records that would violate the upstream dataset terms.

New assets. This paper introduces REPAIR as a method and does not introduce a new public dataset. Any released code, trained repair modules, or derived artifacts will be distributed only when compatible with the licenses and redistribution restrictions of the corresponding upstream datasets and model assets. Where upstream raw data cannot be redistributed, we will provide scripts and instructions for users to obtain the original data from the official providers and reproduce preprocessing locally.

Table 11: Predictive personalization on MIND under strict evaluation with 499 negative samples per test instance. Here, +REPAIR denotes REPAIR with 2-epoch head finetuning. We report MRR, nDCG@5, nDCG@10, and Hit@10. All values are in percentage points.

Host Encoder	MRR			nDCG@5			nDCG@10			Hit@10		
	Base	+REPAIR	Δ	Base	+REPAIR	Δ	Base	+REPAIR	Δ	Base	+REPAIR	Δ
EBNR	47.08	53.22	+6.14	48.65	**57.21**	+8.56	51.63	63.18	+11.55	**69.78**	78.43	+8.65
NAML	45.65	53.22	+7.57	43.13	**57.21**	**+14.08**	46.56	63.18	**+16.62**	63.19	78.43	**+15.24**
NRMS	44.91	52.74	**+7.83**	42.38	56.48	+14.10	45.73	62.44	+16.71	62.31	77.66	+15.35
TrRMIo	47.43	55.21	+7.78	46.27	52.83	+6.56	53.43	57.76	+4.33	68.62	**79.48**	+10.86
SCAPE	42.53	45.32	+2.79	37.53	44.15	+6.62	41.19	47.36	+6.17	61.44	69.37	+7.93
LSTUR	**51.43**	**55.84**	+4.41	**53.64**	57.18	+3.54	**59.63**	**66.92**	+7.29	69.22	72.49	+3.27
MINER	47.22	51.86	+4.64	47.04	53.67	+6.63	52.18	58.91	+6.73	69.41	75.22	+5.81

Table 12: Predictive recommendation results on Amazon Reviews 2023 (All_Beauty). Here, +REPAIR denotes REPAIR with 2-epoch head finetuning. We report MRR, nDCG@5, nDCG@10, and Hit@10. All values are in percentage points.

Host Encoder	MRR			nDCG@5			nDCG@10			Hit@10		
	Base	+REPAIR	Δ	Base	+REPAIR	Δ	Base	+REPAIR	Δ	Base	+REPAIR	Δ
SigLIP2	2.40	17.72	+15.33	1.53	17.71	+16.18	2.18	20.35	+18.17	4.52	32.97	+28.44
OpenCLIP	1.59	19.49	+17.91	0.64	19.46	+18.83	1.05	22.08	+21.03	2.37	34.61	+32.24
DINOv2 E5	1.19	16.15	+14.96	0.43	16.30	+15.87	0.72	18.19	+17.47	1.60	29.15	+27.55

Table 13: Accuracy results for headline generation on PENS with encoder-attached +REPAIR (all values are in 1–100 scale).

Host Summarizer	RG-1			RG-L			RG-SU4			METEOR			W-RG-1			P-RG-L		
	Base	+REPAIR	Boost	Base	+REPAIR	Boost	Base	+REPAIR	Boost	Base	+REPAIR	Boost	Base	+REPAIR	Boost	Base	+REPAIR	Boost
DeepSeek-32B	20.62	**26.84**	**+30.16%**	13.85	**21.22**	**+53.21%**	4.42	**11.99**	**+171.27%**	12.53	**18.23**	**+45.49%**	27.44	**32.80**	+19.53%	14.54	**22.27**	**+53.16%**
Qwen2.5-32B	18.51	**23.47**	**+26.80%**	12.53	**17.74**	**+41.58%**	3.86	**8.46**	**+119.17%**	11.67	**17.23**	**+47.64%**	23.44	**19.64**	-16.21%	11.15	**18.35**	**+64.57%**
FPG†	16.48	19.51	+18.39%	13.08	14.12	+7.95%	2.51	3.11	+23.90%	9.74	11.52	+18.28%	21.03	24.61	+17.02%	13.59	14.78	+8.76%
SCAPE†	19.02	22.47	+18.14%	15.26	16.41	+7.54%	2.88	3.75	+30.21%	9.26	12.53	+35.31%	24.34	28.32	+16.35%	15.78	17.00	+7.73%
GTP	15.97	18.28	+14.46%	12.75	13.67	+7.22%	2.45	2.92	+19.18%	9.17	10.20	+11.23%	20.56	23.83	+15.90%	13.25	14.20	+7.17%
EBNR-T1	15.32	19.26	+25.72%	13.15	14.17	+7.76%	2.27	3.07	+35.24%	9.16	11.68	+27.51%	19.46	24.32	+24.97%	13.61	14.70	+8.01%
EBNR-T2	15.04	18.91	+25.73%	12.97	13.94	+7.48%	2.23	3.00	+34.53%	9.08	11.52	+26.87%	19.12	23.91	+25.05%	13.43	14.45	+7.59%
NAML-T1	15.69	19.46	+24.03%	13.29	14.38	+8.20%	2.44	3.06	+25.41%	9.39	11.76	+25.24%	19.83	24.72	+24.66%	13.75	14.93	+8.58%
NAML-T2	15.38	19.01	+23.60%	13.07	14.11	+7.96%	2.39	2.99	+25.10%	9.28	11.44	+23.28%	19.49	24.15	+23.91%	13.57	14.66	+8.03%
NRMS-T1	15.11	18.71	+23.83%	12.91	13.97	+8.21%	2.31	2.93	+26.84%	9.03	11.21	+24.14%	19.05	23.86	+25.25%	13.38	14.48	+8.22%
NRMS-T2	14.87	18.42	+23.87%	12.74	13.77	+8.08%	2.26	2.85	+26.11%	8.96	11.09	+23.77%	18.82	23.51	+24.92%	13.21	14.28	+8.10%
Walk2Pers	**24.11**	25.36	+5.18%	**15.51**	16.53	+6.58%	**5.90**	7.93	+34.41%	**13.68**	15.51	+13.38%	**31.41**	32.38	+3.09%	**16.88**	17.21	+1.95%
IMPerSumm	21.64	22.00	+1.66%	14.67	14.27	-2.69%	3.25	3.26	+0.42%	9.37	9.82	+4.76%	28.72	28.47	-0.85%	15.24	14.90	-2.23%

Table 14: PSE results for headline generation on PENS. Here, +REPAIR denotes the encoder-attached REPAIR finetuned for summarization (all values are in 0–1 scale).

Host Summarizer	PSE-W-JSD			PSE-W-RG-L			PSE-P-RG-L			PSE-METEOR		
	Base	+REPAIR	Boost	Base	+REPAIR	Boost	Base	+REPAIR	Boost	Base	+REPAIR	Boost
DeepSeek-32B	0.1180	0.1290	+9.32%	0.1485	0.1560	+5.05%	0.0140	0.0166	+18.57%	0.0021	0.0024	+14.29%
Qwen2.5-32B	0.0920	0.1210	+31.52%	0.1253	0.1411	+12.61%	0.0137	0.0149	+8.76%	0.0019	0.0023	+21.05%
SCAPE†	0.1030	0.1125	+9.22%	0.1280	0.1352	+5.62%	0.0109	0.0134	+22.94%	0.0017	0.0020	+17.65%
FPG†	0.0748	0.0839	+12.17%	0.1042	0.1104	+5.95%	0.0081	0.0102	+25.93%	0.0013	0.0016	**+23.08%**
GTP	0.0895	0.0987	+10.28%	0.1138	0.1208	+6.15%	0.0089	0.0113	+26.97%	0.0015	0.0018	+20.00%
EBNR-T1	0.0759	0.0995	+31.09%	0.1055	0.1164	+10.33%	0.0082	0.0116	+41.46%	0.0013	0.0014	+7.69%
EBNR-T2	0.0734	0.0956	+30.25%	0.1028	0.1126	+9.53%	0.0079	0.0110	+39.24%	0.0013	0.0014	+7.69%
NAML-T1	0.0692	0.0918	+32.66%	0.0986	0.1084	+9.94%	0.0072	0.0103	+43.06%	0.0012	0.0013	+8.33%
NAML-T2	0.0678	0.0897	+32.30%	0.0969	0.1061	+9.49%	0.0070	0.0100	+42.86%	0.0012	0.0012	+0.00%
NRMS-T1	0.0643	0.0869	+35.15%	0.0941	0.1037	+10.20%	0.0067	0.0096	+43.28%	0.0011	0.0012	+9.09%
NRMS-T2	0.0627	0.0848	**+35.25%**	0.0925	0.1015	+9.73%	0.0064	0.0093	**+45.31%**	0.0011	0.0011	+0.00%
Walk2Pers	0.1253	0.1419	+13.25%	**0.1671**	**0.1927**	+15.32%	0.0158	0.0175	+10.76%	0.0115	0.0137	+19.13%
IMPerSumm	**0.1262**	**0.1524**	+20.76%	0.1415	0.1772	**+25.23%**	**0.0168**	**0.0234**	+39.29%	**0.0144**	**0.0149**	+3.47%

Table 15: Results on TG-ReDial across both downstream personalization tasks. The upper block reports conversational recommendation under strict evaluation with 499 negative samples per test instance, while the lower block reports conversational response generation using ROUGE metrics (all values are in 1-100 scale).

Conversation-based Item Recommendation

Host Encoder	MRR			HR@10			nDCG@5			nDCG@10		
	Base	+REPAIR	Δ	Base	+REPAIR	Δ	Base	+REPAIR	Δ	Base	+REPAIR	Δ
SASRec	4.11	4.56	+0.45	7.82	8.76	+0.94	3.32	3.48	+0.16	4.16	4.56	+0.40
TextCNN	2.57	2.78	+0.21	5.08	5.04	-0.04	1.92	1.73	-0.19	2.55	2.51	-0.04
GRU4Rec	**4.14**	**4.56**	**+0.42**	**8.80**	**9.13**	+0.33	**3.24**	**3.49**	**+0.25**	**4.38**	**4.60**	**+0.22**

Conversational Generation

Host Generator	ROUGE-1			ROUGE-L			ROUGE-SU4		
	Base	+REPAIR	Δ	Base	+REPAIR	Δ	Base	+REPAIR	Δ
REDIAL	12.33	27.76	+15.43	11.17	23.52	+12.35	3.31	10.54	+7.23
Transformers	**31.81**	**31.97**	**+0.16**	**26.46**	**26.48**	**+0.02**	**12.62**	**12.63**	**+0.01**

Table 16: Diagnostic ablation of repair time-scale resolution on representative hosts spanning additive-attention, recurrent, Transformer-based, and SSM-based encoders. We report MRR for all predictive tasks. REPAIR-NoTS denotes repair without explicit long/short/episodic time-scale resolution, while REPAIR-L+S+E denotes the full REPAIR model.

Dataset	Host Encoder	Family	Base	REPAIR-NoTS	REPAIR-L	REPAIR-S	REPAIR-E	REPAIR-L+S	REPAIR-S+E	REPAIR-L+E	REPAIR-L+S+E
MovieLens	Mamba4Rec	SSM	29.29	28.43	29.92	29.63	29.45	32.17	31.35	31.04	34.26
	TiSASRec	Transformer	35.51	35.52	35.54	35.78	35.83	35.91	35.66	35.41	36.78
	SIGMA	SSM	33.27	33.29	33.66	33.32	33.29	33.88	33.73	33.43	34.81
MIND	NAML	Additive Attention	45.65	45.95	47.65	48.32	47.13	49.31	49.18	48.79	53.22
	EBNR	LSTM-style RNN	47.08	47.74	48.88	48.03	47.62	50.13	49.38	48.76	53.22
	NRMS	Transformer	44.91	45.06	46.43	45.13	47.73	47.58	46.44	46.15	52.74

Table 17: Progressive ablation of the REPAIR training objective on two representative MovieLens hosts and two representative news hosts. We report MRR.

Dataset	Host Encoder	Base	REPAIR (w/$\mathcal{L}_{\text{task}}$)	REPAIR (w/ $\mathcal{L}_{\text{task}} + \mathcal{L}_{\text{pos}}$)	REPAIR (w/ $\mathcal{L}_{\text{task}} + \mathcal{L}_{\text{pos}} + \mathcal{L}_{\text{coh}}$)
MovieLens	TiSASRec	35.51	35.71	36.14	36.78
	Mamba4Rec	29.29	31.43	32.16	34.26
News Recommendation	LSTUR	51.43	52.17	54.65	55.84
	NRMS	44.91	46.39	51.29	52.74

Table 18: RQ3-b: Length-controlled timestep-selection ablation on representative predictive hosts. Cells report **Base / NS / SEL**, where **NS** disables selective timestep retention and **SEL** denotes full REPAIR. The row **Sel. Steps** reports the number of selected timesteps under SEL.

Host Model	Metric	$l = 10$	$l = 20$	$l = 30$	$l = 40$	$l = 50$
			MovieLens			
Mamba4Rec (SSM)	MRR	17.43 / 19.32 / 19.45	21.65 / 23.22 / 23.43	26.11 / 28.45 / 28.29	27.32 / 29.89 / 29.45	29.29 / 33.95 / 34.26
	nDCG@10	22.62 / 27.15 / 27.94	27.85 / 30.61 / 31.24	33.32 / 35.40 / 35.36	35.53 / 36.41 / 36.21	38.50 / 40.03 / 40.17
	Sel. Steps	7	16	22	28	34
TiSASRec (Transformer)	MRR	22.35/22.42/22.65	24.17/24.93/24.45	25.73/26.69/26.84	31.43/31.89/32.05	35.51 / 36.56 / 36.78
	nDCG@10	24.95/25.66/25.74	27.54/28.26/27.96	28.84/29.93/30.06	35.44/36.17/36.33	41.14/42.66/ 42.91
	Sel. Steps	9	18	25	31	36
			MIND			
NAML (Additive Attention)	MRR	37.25/39.93/40.32	40.43/39.16/43.71	42.05/45.19/45.34	45.11/48.77/49.31	45.65/52.17/53.22
	nDCG@10	38.16/40.02/40.75	41.66/39.94/44.2	43.11/45.76/45.96	45.63/49.13/49.76	46.56/59.75/63.18
	Sel. Steps	8	17	25	32	36
EBNR (Recurrent)	MRR	39.10 / 41.42 / 41.68	41.84 / 44.03 / 44.26	43.72 / 46.31 / 46.57	46.12 / 49.88 / 50.13	47.08 / 52.11 / 53.22
	nDCG@10	43.28 / 45.01 / 44.96	46.11 / 48.74 / 48.93	48.36 / 52.02 / 52.28	50.27 / 57.14 / 57.62	51.63 / 60.74 / 63.18
	Sel. Steps	8	16	24	31	35

Table 19: Noisy-trajectory stress test for RQ3 representative hosts. The noisy frozen-host scores are achieved via randomly inserting abrupt disconnected content nodes in the trajectories by approximately 35 to 40% in the clean trajectories.

Dataset	**Host Encoder**	**Noisy Base**		**+REPAIR**	
		MRR	nDCG@10	MRR	nDCG@10
MovieLens	Mamba4Rec	18.21	23.94	21.43	25.74
MovieLens	TiSASRec	22.34	25.92	24.65	28.17
MIND	NAML	28.61	29.23	32.85	35.81
MIND	EBNR	29.63	32.44	31.65	35.17

Table 20: Normalized online compute footprint on representative RQ4 hosts.

Model	Work Unit	Base Work	Added Work	Total Work	@50 ms Budget	@25 ms Budget	Scaling Regime
Mamba4Rec	GFLOPs/query	0.75	0.18	0.93	0.019 TFLOP/s	0.037 TFLOP/s	$O(T)$
TiSASRec	GFLOPs/query	1.45	0.18	1.63	0.033 TFLOP/s	0.065 TFLOP/s	$O(T^2)$
NAML	GFLOPs/query	0.03	0.02	0.05	0.001 TFLOP/s	0.002 TFLOP/s	$O(T)$
DeepSeek-32B	GFLOPs/token	64.0	—	64.0	1.28 TFLOP/s	2.56 TFLOP/s	autoregressive decode

Table 21: Serving-path decomposition of representative RQ4 hosts.

Model	Serving Mode	Resident Frozen Artifact	Additional Artifact with REPAIR	Dominant Online Bottleneck	Why the Deployment Story Differs
Mamba4Rec	end-to-end encoder serving	item embedding table + 4-layer selective SSM encoder	repair module attached to exposed terminal state	frozen encoder passes + residual repair fusion	cost remains mostly compute-bound; no external cache object is introduced
TiSASRec	end-to-end encoder serving	item table + time-interval-aware self-attention encoder	repair module attached to exposed terminal state	attention score/value passes + repair fusion	latency is more sensitive to sequence interactions because the host is attention-based rather than SSM-based
NAML	cached news-table serving	cached NEWS_TABLE + frozen user encoder	repair module attached after cached-history composition	table lookup / memory bandwidth + attentive user aggregation	serving shifts a large fraction of work out of online text encoding and into cached vector retrieval
DeepSeek-32B	context-conditioned autoregressive decoding	frozen decoder weights + KV cache	context-tuning state	token-by-token decoding	absolute online cost is dominated by decoder generation rather than by the added personalization attachment

This separation is useful because the encoder hosts, cached-table news recommenders, and context-conditioned LLM reference do not pay deployment cost through the same bottleneck.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the scoped claim that `REPAIR` is a selective pre-head state-repair framework for frozen personalization encoders, and the claims are matched to the four evaluation questions on predictive gains, post-head corrector comparison, temporal repair, and deployment overhead. The paper does not claim full encoder retraining, universal personalization, or unrestricted generalization beyond the evaluated frozen-host setting.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 7 discusses limitations on dialogue-style eventization, architecture dependence, decoder coupling, and the weaker broader-applicability gains on TG-ReDial. The paper also makes deployment cost explicit through memory and latency overhead analysis, and Appendix H discusses privacy and redistribution constraints for user-interaction datasets.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The theoretical results are stated with explicit assumptions in Appendix B, including admissible low-rank repair dimension, state-relative correction recoverability, temporal-trace recoverability, and the recoverability-oriented learning objective. Each theorem is accompanied by a proof and an interpretation connecting the assumptions to the `REPAIR` mechanism.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper specifies the datasets, frozen hosts, train/test protocol, candidate construction, metrics, training stages, optimization settings, and paired evaluation procedure in Section 5 and Appendices C–F. The anonymized supplementary zip contains the code and scripts needed to reproduce the reported method and baseline evaluations under the stated data-access constraints.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide anonymized code as supplementary zip files for review. Raw benchmark datasets or third-party checkpoints are not redistributed when upstream terms restrict redistribution. Appendix H lists official sources and the released scripts reproduce preprocessing locally after users obtain the data from the original providers.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5 describes the evaluation coverage, frozen-host protocol, negative sampling, autoregressive testing, and reported metrics. Appendix C.3 gives the staged training schedule, task objectives, loss weights, AdamW optimizer, learning rate, weight decay, gradient clipping, mixed precision use, and checkpointing/model-selection protocol.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Appendix A.9 reports paired bootstrap resampling over the held-out test pool and states that each comparison uses identical test examples, history prefixes, and candidate sets. reports the measurement of paired test-set stability under fixed trained systems rather than variance over independent retraining runs. Independent retraining runs are not reported due to compute constraints.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix C.3 reports that experiments were run on Lightning.ai Studios/Clusters using NVIDIA A100, H100, and RTX PRO 5000-class GPU resources, with single-GPU jobs unless a host recipe required otherwise. Section 5 and the RQ4 deployment analysis further report memory and latency overhead, while the supplementary zip provides the executable configuration needed to reproduce runs in the same protocol.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The work uses existing benchmark datasets and publicly documented model assets for research evaluation, does not attempt user identification or deanonymization, and does not release raw user histories. Appendix H documents privacy, licensing, non-redistribution, and responsible-use constraints, and the anonymized supplementary code preserves double-blind review requirements.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The paper discusses positive impact through more accurate and deployment-bounded personalization with frozen encoders, reducing the need for full host retraining. It also discusses limitations and risks involving privacy, user-history handling, upstream dataset restrictions, eventization fidelity, and architecture-dependent behavior in Section 7 and Appendix H.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: The paper does not release a new high-risk pretrained model, raw scraped dataset, or raw user-interaction history. Appendix H states that raw datasets, upstream checkpoints, and private user histories are not redistributed, and that released code or derived artifacts are distributed only when compatible with upstream licenses and access terms.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Appendix H lists the pre-trained model assets, encoders, datasets, provider URLs, and known licenses or usage terms, including MIT, Apache 2.0, Microsoft Research License Terms, and research-only restrictions where applicable. The paper states that raw datasets and third-party checkpoints are not redistributed and that original papers and providers are cited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The new asset is the `REPAIR` method implementation and supplementary reproduction package; the paper does not introduce a new public dataset. Appendix H states that released code, trained repair modules, or derived artifacts will be distributed only when compatible with upstream licenses, and the anonymized supplementary zip documents reproduction instructions.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not conduct new crowdsourcing experiments and does not directly recruit, interact with, or compensate human subjects. It uses existing benchmark datasets under their stated research-use terms.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve new human-subject data collection, direct participant interaction, intervention, or crowdsourced annotation. The work is based on existing benchmark datasets, and Appendix H describes privacy-preserving use, non-deanonymization, and non-redistribution of raw user histories.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: The paper explicitly describes LLM-based hosts and prompt-conditioned LLM settings in Appendix A.8, including frozen LLM backbones and soft-prompt/state-projection conditioning through `REPAIR`. LLMs are used as experimental hosts and baselines.

    Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.