Action-On-Item Preference Flow: A Shared Event Schema for Predictive and Generative Personalization

Parthiv Chatterjee[1] **Kashish Kanjariya**[1] **Vashisht Purani**[1]
Sourish Dasgupta[1] **Tanmoy Chakraborty**[2]
[1]KDM Lab, Dhirubhai Ambani University
[2]LCS2 Lab, Indian Institute of Technology Delhi
{202421011,202301135,202412077,sourish_dasgupta}@dau.ac.in
tanchak@iitd.ac.in

Abstract

Personalization histories vary across domains because surface actions and objectives differ: users watch/skip movies, click/ignore news, or request recommendation, summarization, question answering, or response generation. Rather than build a separate history encoder for each action space, we rewrite native interactions into a shared *action-on-item event schema*: a specific action on a specific item becomes positive, negative, or command-like evidence over a target-locus embedding whose geometry induces item-kind abstraction. Task requests become generalized commands over the active item-side abstraction. The modeling problem is how this evidence enters a user-specific state, persists, and is read out by the active command. We formulate the Multi-Timescale State Hypothesis (MTSH), where signed and command-like evidence flows through long-term stable interests, recency-sensitive short-term interests, and bursty episodic traces. We instantiate MTSH with `PerTIDE`, an action-conditioned multi-timescale encoder that learns a user-specific preference-history state and uses command-conditioned readout to produce a task-ready state. We evaluate `PerTIDE` on MovieLens, MIND, and PENS for prediction, and PENS/OpenAI-Reddit for generation. `PerTIDE` improves over the strongest baselines without explicit trace factorization by +2.36/+3.47, +10.05/+9.01, and +0.90/+1.42 MRR/nDCG on MovieLens, MIND, and PENS; and over strong two-shot large language model by +0.10/+0.18 and +0.04/+0.19 PerSEval-JSD/PerSEval-METEOR on PENS and OpenAI-Reddit. Removing action conditioning and command-conditioned readout degrades MIND by 6.52 MRR / 9.13 HR@10 and 4.14 MRR / 4.42 HR@10. Trace diagnostics show $L/S/E$ specialization, while full $L+S+E$ is strongest under mixed stable, recency-sensitive, and episodic command evidence. These results support action-conditioned multi-timescale preference flow as a reusable user-history encoding principle.

1 Introduction

Personalization systems are conditioned on user interaction histories (*trajectories*), whose nature varies across domains. Users may watch or skip movies, click or ignore news, open books, request summaries, ask questions about items, or generate dialogue responses. This requires separate history encoders for new tasks and domains. We argue that *native interactions can be rewritten into a shared action-on-item event schema* where a specific action on a specific item becomes positive, negative, or command-like signal over an embedding of the acted-on item (termed *target-locus embedding*). For example, watching a TV series and clicking a movie are positive actions, while the embedding space tracks how similar or different TV series and movie items are. Task-specific requests become generalized commands. The shared object is this event schema and encoder mechanism.

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

This abstraction lets diverse trajectory data enter a common training substrate without matched native action vocabularies or fixed item taxonomies. We adopt the User Interaction Graph (UIG) interface from prior personalized summarization work [Chatterjee et al., 2025], to convert domain-specific user logs into action-on-item trajectories. Given such a trajectory, an encoder has to decide how strongly each signed or command-like event enters the user-specific state and how its influence persists. Positive actions may reinforce a preference direction, negative actions may suppress it, and command-like actions may indicate which operation later consumes the encoded state. Since task-specific requests are generalized commands, the encoder must also resolve the active command at readout time. This gives a restricted form of schema-level generalizability: heterogeneous trajectory datasets can share an encoder when their native interactions are eventizable as action roles over target-locus embeddings.

Multi-scale temporal modeling is well established in forecasting and sequence modeling [Wu et al., 2021c, Zhou et al., 2022, Wu et al., 2023a, Wang et al., 2024, Chen et al., 2024], and long/short user-interest modeling is common in recommendation [An et al., 2019, Yu et al., 2019, Li et al., 2020, Shen et al., 2022]. These lines establish the value of temporal and user-specific modeling, but do not address the setting where heterogeneous native interactions are first normalized into action roles over target-locus embeddings and then reused across predictive and generative objectives. The missing object is a reusable event-schema representation for user-specific preference flow that abstracts native interactions into signed and command-like evidence, learns how this evidence enters the user state, and leaves command interpretation to readout. We formulate the Multi-Timescale State Hypothesis (MTSH) for action-conditioned preference trajectories: signed and command-like evidence over a shared action-on-item trajectory should flow through long-term traces for persistent preference mass, short-term traces for recency-sensitive interests, and episodic traces for finite-window event or command imprints. We instantiate MTSH with `PerTIDE` (**Per**sonalization via Multi-**t**imescale **I**nteraction **D**rift **E**ncoder), an action-conditioned state-space encoder that builds a user-specific preference-history state under the shared schema and uses command-conditioned readout to resolve the generalized command into a task-ready state.

We ask whether (i) a shared action-on-item preference-flow encoder outperforms task-native history encoders without explicit long/short/episodic trace factorization on predictive personalization, (ii) action conditioning, command-conditioned readout, and long/short/episodic traces are each necessary, and (iii) the same user-specific preference-flow state supports command-driven generative personalization. We evaluate `PerTIDE` on MovieLens-1M, MIND, and PENS for prediction, and on PENS and OpenAI-Reddit for generation. `PerTIDE` improves over the strongest baselines without explicit trace factorization by **+2.36/+3.47**, **+10.05/+9.01**, and **+0.90/+1.42** MRR/nDCG on MovieLens-1M, MIND, and PENS. For generation, it improves over the strongest two-shot large language model by **+0.103/+0.181** on PENS and **+0.042/+0.185** on OpenAI-Reddit using PSE-JSD/PSE-METEOR. Removing action conditioning or command-conditioned readout reduces MIND performance by **6.52** MRR / **9.13** HR@10 and **4.14** MRR / **4.42** HR@10. Controlled diagnostics show that the three traces specialize under corresponding regimes, while full $L + S + E$ remains strongest when trajectories mix stable, recency-sensitive, and episodic command evidence. Code and reproducibility package: `https://github.com/KDM-LAB/Shared-Action-on-Item-Preference-Flow-PerTIDE---NeurIPS-2026.git`.

Contribution: (i) We propose a reusable event-schema encoder for personalization, (ii) cast heterogeneous user histories into a shared action-on-item preference-trajectory form, (iii) formulate MTSH as an action-conditioned preference-flow hypothesis, (iv) instantiate it with `PerTIDE`, and provide predictive, generative, ablation, and temporal-diagnostic evidence showing when the shared abstraction, action conditioning, command readout, and trace decomposition matter.

2 Related Work

Personalization Models and Datasets. Personalization is studied across recommendation, dialogue, and generation, with datasets such as MIND, PENS, PersonaChat, MPChat, and LaMP supporting different user-conditioned objectives [Wu et al., 2020, Ao et al., 2021, Zhang et al., 2018, Ahn et al., 2023, Salemi et al., 2024]. Prior systems usually build task-native user states: news and headline models aggregate clicked content or encode one behavior stream [Wu et al., 2019a, Okura et al., 2017, Song et al., 2023], conversational recommenders add user or profile memories for response selection and generation [Li et al., 2022b, Ren et al., 2022], and personalized large language model

methods retrieve profiles or maintain external memories [Zhang et al., 2024, Tan et al., 2025]. These approaches show that user histories matter, but they usually bind the history representation to a task, dataset, or profile format. `PerTIDE` instead targets a reusable action-on-item event schema where native interactions become positive, negative, or command-like evidence over target-locus embeddings, with prediction and generation handled as different command-conditioned readouts.

Shared Action and Sequence Abstractions. Multi-behavior and multi-task recommenders model several user actions or objectives jointly [Gao et al., 2019, Ma et al., 2018, Kim et al., 2025], while unified text-to-text and personalized language benchmarks express multiple tasks through a shared language interface [Raffel et al., 2020, Geng et al., 2023, Salemi et al., 2024]. Control-oriented world models also show that action-conditioned latent states can be reused across objectives [Shaj et al., 2023, Hansen et al., 2024, Saanum et al., 2024, Georgiev et al., 2025]. These lines motivate common interfaces, but preference logs do not provide a native shared action space: clicks, skips, requests, and generated responses have different surface meanings across domains. We adopt the User Interaction Graph interface from prior personalized summarization work [Chatterjee et al., 2025], but use it to normalize input histories themselves: native actions map to signed or command-like roles, while target-locus embeddings carry item-side abstraction without a hand-built taxonomy.

Multi-Timescale Preference Modeling. Multi-scale temporal modeling is established in forecasting and sequence modeling [Wu et al., 2021c, Zhou et al., 2022, Wu et al., 2023a, Wang et al., 2024, Chen et al., 2024]. Recommendation also includes long/short user representations, time-aware attention, multi-interest extraction, sequential state-space models, and temporal graph models [An et al., 2019, Yu et al., 2019, Li et al., 2020, Cen et al., 2020, Shen et al., 2022, Liu et al., 2024, Ou et al., 2025]. However, these models often implicitly mix temporal evidence inside one task-specific stream, attention pattern, memory, or compressed state. `PerTIDE` instead factorizes action-conditioned evidence into long-term, short-term, and episodic traces, and uses command-conditioned readout to resolve how a user-specific state is consumed by predictive or generative heads.

3 Action-On-Item Preference Trajectories

Shared Action-On-Item Interface. Personalization logs expose domain-native actions: a movie system may record watches, skips, ratings, and requests; a news system may record clicks, ignores, and headline-generation requests; and a dialogue system may record turns, responses, and stance-like signals. If these actions are modeled only through their native labels, each domain appears to require a separate user-history encoder. We instead rewrite each native interaction into a shared action-on-item event schema:

$$(\text{specific action}, \text{specific item}) \rightarrow (\text{positive/negative/command}, \text{target-locus embedding}).$$

The action role is explicit: native events map to $\mathcal{A} = \{\texttt{positive}, \texttt{negative}, \texttt{command}\}$. Positive actions encode reinforcing evidence, negative actions encode suppressive or rejected evidence, and command-like actions encode task requests over an active item or response node. Item-side abstraction is carried by the target-locus embedding rather than a hand-built taxonomy, allowing related items or supported operations to be close in the embedding space. Thus, heterogeneous trajectories can enter a common training substrate without manually aligning action spaces or item kinds. Task-specific requests become generalized commands over the active item-side abstraction, making recommendation, summarization, question answering, and response generation different command instances over the same action-on-item trajectory. We implement this interface through the User Interaction Graph (UIG) representation of Chatterjee et al. [2025]. For a user u, a UIG is a time-directed acyclic graph $G_u = \langle N_u, E_u \rangle$ whose nodes include a user anchor, item/content nodes, and response nodes produced by command-like events. Its edges contain action-labeled transitions, with content-side actions mapped to $\mathcal{A}$ and response-side construction edges linking a command-conditioned item to its response. A user trajectory τ_u is the time-ordered path induced by these interactions, and $\mathcal{T}$ denotes the trajectory pool used for training and evaluation. Dataset-specific construction rules are given in Appendix B.1 (Dataset statistics: Appendix A, and notations: Table 7).

Directly modeling raw UIGs can be expensive and noisy for long histories, so we use a compact interaction-block tier. Each timestamped event becomes a b-node $b_u^{(t_i)} = \langle a^{(t_i)}, tl^{(t_i)} \rangle$, where $a^{(t_i)} \in \mathcal{A}$ is the action role and $tl^{(t_i)} \in N_u$ is the target locus. For content-side events, $tl^{(t_i)}$ is the active item node; for response-side events, it is the linked response node. The embedding $\mathbf{e}_{tl}^{(t_i)}$ gives

the continuous item-side representation consumed by the encoder, and the interaction-block trajectory $b_u^{(t_{1:i})}$ is the object over which we model preference evolution. This schema-normalized trajectory changes the modeling question: the encoder need not learn a new event space for every domain; it must decide how positive, negative, and command-like evidence enters a user-specific state, how long that evidence persists, and how a later command consumes it.

Schema-Level Generalizability. This schema gives a restricted but useful form of generalizability. A new personalization domain need not share native action labels, item types, or downstream tasks. It only needs eventizable interactions that can be rewritten as positive, negative, or command-like evidence over target-locus embeddings. The event schema is a reusable evidence-flow mechanism and *not a universal user-preference space*. We provide formal bounds in **Propositions** B.1-B.3. Thus, `PerTIDE` generalizes when heterogeneous histories become comparable through the action-on-item schema, not through a manually fixed taxonomy over actions or item kinds.

From History States to Preference Flow. For each user u and timestep t_i, a history encoder maps the prefix $b_u^{(t_{1:i-1})}$ to a latent preference state. Prior personalization encoders can be viewed through three broad state assumptions. The *Profile Aggregation Hypothesis* (PAH) exposes an order-invariant profile, $\mathbf{z}_{b_u}^{(t_i)} = f_{\mathrm{PAH}}(\{\phi(b_u^{(t_j)})\}_{j<i})$, as in mean pooling, max pooling, and additive-attention profile encoders [Li et al., 2016, Wu et al., 2019a,b, 2021a]. The *Markovian Drift Hypothesis* (MDH) exposes a recurrently compressed state, $\mathbf{z}_{b_u}^{(t_i)} = f_{\mathrm{MDH}}(\mathbf{z}_{b_u}^{(t_{i-1})}, \phi(b_u^{(t_i)}); \rho) + \xi^{(t_i)}$, where ρ controls the effective recency horizon and $\xi^{(t_i)}$ may depend on the action role [Okura et al., 2017, Wu and Yu, 2024]. The *Long-Horizon Dynamics Hypothesis* (LDH) exposes a full-prefix stream or compressed state, $\mathbf{z}_{b_u}^{(t_i)} = f_{\mathrm{LDH}}(\{\phi(b_u^{(t_j)})\}_{j<i}; K_\theta)$, as in Transformer- and state-space-based history encoders [Liu et al., 2024, Xiao et al., 2025, Tan et al., 2025].

These regimes differ in the state exposed to the downstream task: PAH exposes one aggregated profile, MDH one recurrent state, and LDH one full-prefix stream or compressed state. In all three cases, temporal evidence may be mixed implicitly, but the encoder does not explicitly route signed and command-like evidence through retention mechanisms before task consumption. Moreover, because task-specific requests are abstracted into generalized commands, the user-specific state must be read in a command-conditioned way. This motivates a preference-flow hypothesis in which schema-normalized evidence is not merely compressed into one state, but routed according to how it should persist before command-conditioned consumption.

Multi-Timescale State Hypothesis. The preceding schema induces a multi-timescale state requirement. Once heterogeneous interactions are rewritten as signed or command-like evidence over target-locus embeddings, a single retention rule is too restrictive: some evidence should accumulate, some should remain recency-sensitive, and some should persist as a finite event or command imprint. We therefore formulate the Multi-Timescale State Hypothesis (MTSH) for action-conditioned preference trajectories. MTSH states that signed and command-like evidence should flow through complementary traces: long-term traces for persistent preference mass, short-term traces for recency-sensitive interests, and episodic traces for finite-window event or command imprints. The hypothesis is not that multi-scale temporal modeling is new in isolation, nor that learned dimensions are semantically disentangled by construction. Rather, MTSH specifies a reusable personalization structure: action-conditioned evidence enters a user-specific history state through explicit trace channels, and command-conditioned readout resolves how a generalized command consumes that state.

At timestep t_i, let $\mathbf{m}_u^{(t_{i-1},L)}$, $\mathbf{m}_u^{(t_{i-1},S)}$, and $\mathbf{m}_u^{(t_{i-1},E)}$ denote the long-term, short-term, and episodic traces before processing the current event. Let Φ fuse trace states, Ψ inject the current action-conditioned evidence, Π infer a command/task distribution from the seed task embedding $\mathbf{g}_q$ and the user-specific history state, and G convert that state into a task-ready state. MTSH is written as

$$\begin{aligned} \mathbf{h}_{b_u}^{(t_i)} &= \Phi\Big(\mathbf{m}_u^{(t_{i-1},L)}, \mathbf{m}_u^{(t_{i-1},S)}, \mathbf{m}_u^{(t_{i-1},E)}\Big) + \Psi\Big(\mathbf{e}_{tl}^{(t_i)}, a^{(t_i)}\Big); \\ \mathbf{z}_{b_u}^{(t_i)} &= G\Big(\Pi(\mathbf{g}_q, \mathbf{h}_{b_u}^{(t_i)}), \mathbf{h}_{b_u}^{(t_i)}\Big). \end{aligned} \tag{1}$$

Here $\mathbf{h}_{b_u}^{(t_i)}$ is the user-specific preference-history state after action-conditioned evidence is injected, while $\mathbf{z}_{b_u}^{(t_i)}$ is the command-conditioned task-ready state. Equation 1 separates three roles that are otherwise entangled in one history stream: Ψ determines how positive, negative, and command-like actions over target-locus embeddings enter the state; Φ determines how prior evidence persists across

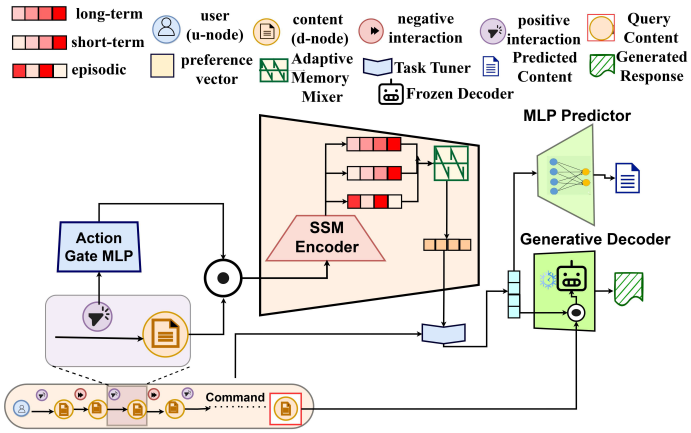


Figure 1: `PerTIDE` **pipeline**: a user's UIG trajectory is mapped to action-gated b-cells and encoded by a tri-trace SSM with an adaptive memory mixer; a task tuner produces a task-ready state that conditions task-specific heads for prediction (MLP) and generation (frozen LM decoder).

long-term, short-term, and episodic traces; and $G \circ \Pi$ resolves the generalized command into a state that a predictive or generative head can consume. We instantiate this hypothesis with `PerTIDE` next.

4 `PerTIDE` as an Instantiation of MTSH

We instantiate MTSH with `PerTIDE` (**Per**sonalization via multi-**T**imescale **I**nteraction **D**rift **E**ncoder), an action-conditioned encoder for predictive and generative personalization under the shared action-on-item event schema. The full pipeline is described in Figure 1. `PerTIDE` follows the decomposition in Equation 1: action-gated b-cells realize Ψ, tri-trace state-space updates realize Φ, and command-conditioned readout realizes $G \circ \Pi$. The encoder first builds a user-specific preference-history state from the action-on-item trajectory; readout then resolves the generalized command into a task-ready state for a predictive head or frozen generative decoder.

Action-Gated Event Cells and Tri-Trace Encoder. We generate seed embeddings for item and response nodes using MPNet [Song et al., 2020]. At timestep t_i, a lightweight action gate maps the action role $a^{(t_i)}$ to $\boldsymbol{\mu}^{(t_i)} = \mathrm{sigmoid}\big(\mathrm{MLP}_{\mathrm{gate}}(a^{(t_i)})\big)$. This modulates the target-locus embedding (item-side abstraction) $\mathbf{e}_{tl}^{(t_i)}$ to form $\mathbf{b}_u^{(t_i)} = \boldsymbol{\mu}^{(t_i)} \odot \mathbf{e}_{tl}^{(t_i)}$. The action role specifies whether the event is positive, negative, or command-like. Thus, related items can contribute related evidence through target-locus embeddings, while the action gate controls how that evidence enters the user-specific state. This implements the action-conditioned evidence injection term Ψ in Equation 1.

The action-gated event is routed through three parallel state-space traces. For $r \in \{\mathrm{L}, \mathrm{S}, \mathrm{E}\}$, let $A_r = Q_r \mathrm{diag}(\Lambda_r) Q_r^{-1}$ denote a learned transition in eigenbasis form. We induce different retention profiles through trace-specific spectral maps:

$$
\begin{aligned}
\mathbf{m}_u^{(t_i,\mathrm{L})} &= Q_{\mathrm{L}} \exp(\delta \Lambda_{\mathrm{L}}) Q_{\mathrm{L}}^{-1} \mathbf{m}_u^{(t_{i-1},\mathrm{L})} + B\mathbf{b}_u^{(t_i)}, \\
\mathbf{m}_u^{(t_i,\mathrm{S})} &= Q_{\mathrm{S}} \sigma(-\kappa \Lambda_{\mathrm{S}}) Q_{\mathrm{S}}^{-1} \mathbf{m}_u^{(t_{i-1},\mathrm{S})} + B\mathbf{b}_u^{(t_i)}, \\
\mathbf{m}_u^{(t_i,\mathrm{E})} &= Q_{\mathrm{E}} (I + \eta \Lambda_{\mathrm{E}}) Q_{\mathrm{E}}^{-1} \mathbf{m}_u^{(t_{i-1},\mathrm{E})} + B\mathbf{b}_u^{(t_i)}.
\end{aligned}
\tag{2}
$$

The generic recursion $\mathbf{m}_t^{(r)} = A_r \mathbf{m}_{t-1}^{(r)} + B\mathbf{b}_t$ unrolls, up to the initial-state term, as $\sum_{j=1}^{t} A_r^{t-j} B\mathbf{b}_j$, so each trace induces a lag kernel $K_r(\Delta) = A_r^{\Delta}$. The maps $\exp(\delta\Lambda_{\mathrm{L}})$, $\sigma(-\kappa\Lambda_{\mathrm{S}})$, and $(I + \eta\Lambda_{\mathrm{E}})$ respectively support slow accumulation, recency-sensitive interests, and finite-window imprints. Appendices D.1, D.2, and D.3 give stability conditions, sufficient hyperparameter choices, mode-wise kernel derivations, trace separation, and the broader view that stable modal mixtures can also approximate trend-like retention, delayed persistence, and damped seasonal components. The updated traces are then fused into the user-specific preference-history state $\mathbf{h}_{b_u}^{(t_i)} = \mathrm{MLP}_{\Phi}\left(\mathbf{m}_u^{(t_i,\mathrm{L})}, \mathbf{m}_u^{(t_i,\mathrm{S})}, \mathbf{m}_u^{(t_i,\mathrm{E})}\right)$.

Hence, repeated evidence can accumulate in $\mathbf{m}^{(\mathrm{L})}$, short-term shifts can remain prominent in $\mathbf{m}^{(\mathrm{S})}$, and bursty events can leave a finite imprint in $\mathbf{m}^{(\mathrm{E})}$ before adaptive fusion.

Command-Conditioned Readout. The shared schema abstracts task-specific requests into generalized command-like actions. Hence, the downstream module must resolve which concrete command consumes the user-specific state. Given the seed task embedding $\mathbf{g}_q$ and history state $\mathbf{h}_{b_u}^{(t_i)}$, `PerTIDE` estimates $\boldsymbol{\pi}_u^{(t_i)} = \Pi(\mathbf{g}_q, \mathbf{h}_{b_u}^{(t_i)}) = \mathrm{SoftMax}\Big(W_{\mathrm{task}}[\mathbf{h}_{b_u}^{(t_i)}; \mathbf{g}_q]\Big)$ where $\boldsymbol{\pi}_u^{(t_i)}$ is a distribution over command/task types. A command gate then yields the task-ready state $\mathbf{z}_{b_u}^{(t_i)} = G(\boldsymbol{\pi}_u^{(t_i)}, \mathbf{h}_{b_u}^{(t_i)}) = (W_\pi \boldsymbol{\pi}_u^{(t_i)}) \odot \mathbf{h}_{b_u}^{(t_i)} + \mathbf{b}_G$. Hence, $\mathbf{h}_{b_u}^{(t_i)}$ remains a user-specific preference-history state under the shared schema, while $\mathbf{z}_{b_u}^{(t_i)}$ is the command-conditioned state consumed by a predictive or generative head. The action role $a^{(t_i)}$ tells the encoder whether the event provides a positive, negative, or a command-like request signal; $\mathbf{g}_q$ tells the readout which concrete command instance should consume the resulting state.

Training & Downstream Task Heads. We supervise the encoder with temporal alignment, next-event prediction, and command-readiness. A position classifier gives $\mathcal{L}_{\mathrm{pos}}$, a next-$b$-node predictor gives $\mathcal{L}_{\mathrm{next}}$ over the b-node vocabulary $\mathcal{B}$, and command classification gives $\mathcal{L}_{\mathrm{task}}$ using the true task label T^*. The encoder objective is $\mathcal{L}_{\mathrm{enc}} = 0.2\mathcal{L}_{\mathrm{pos}} + 0.4\mathcal{L}_{\mathrm{next}} + 0.4\mathcal{L}_{\mathrm{task}}$. When trajectories from multiple domains are pooled after schema rewriting, this objective optimizes one empirical risk over action roles, target-locus embeddings, and command seeds. We prove in Proposition B.4 that, under this common-substrate view, the pooled learner receives a total-sample-size estimation term, while target-domain risk is bounded by schema-distribution distance and command-semantics discrepancy. With this objective, temporal alignment and command-readiness are supervised at readout rather than folded into the user-specific history state (hyperparameter and dimension details: App. Table 8).

For predictive tasks, a task head f_T consumes the query-conditioned task-ready representation and predicts $\hat{y}_T = f_T(\hat{\mathbf{z}}_{q,u}^{(t_{l+1})})$ with cross-entropy loss. For generative tasks, we condition a frozen DistillGPT decoder [Su and McMillan, 2024] through a learned context vector and evaluate two decoder contextualization variants: **Decoder-UFI** (User Fused Injection) and **Decoder-UGI** (User Gated Injection). UFI directly fuses $\hat{\mathbf{z}}_{q,b_u}^{(l+1)}$ with the query embedding $\mathbf{e}_{d_q}$ to learn user-query interactions implicitly. UGI modulates the predicted query-conditioned representation $\hat{\mathbf{z}}_{q,u}^{(t_{l+1})}$ by the task-adapted user state, allowing the decoder to emphasize or suppress query semantics in a user-dependent manner. Hence, UFI adapts generation to the fused user-query context, whereas UGI further adapts it to which aspects of the query should matter for this user, given the preference trajectory (details in App. D.5). Additional training details are in Appendix I.

5 Evaluation

We evaluate whether schema-normalized action-on-item preference flow supports both predictive and generative personalization. Following Section 1, we ask: (i) whether a reusable action-on-item encoder improves over task-native PAH/MDH/LDH history encoders; (ii) whether action-conditioned entry, multi-timescale persistence, and command-conditioned readout are each necessary; and (iii) whether the same user-specific state supports command-driven generation.

Tasks & Trajectory Pools. Each dataset is treated as a schema-normalized trajectory pool rather than a flat example table. For dataset D, UIG construction gives $\mathcal{T}_{\mathrm{train}}^D$ and $\mathcal{T}_{\mathrm{eval}}^D$; slicing either pool at query-target positions yields instances with history $\tau_{u_j}^h$, optional query content d_q, and a task target. Training uses only $\mathcal{T}_{\mathrm{train}}^D$, while reported scores use $\mathcal{T}_{\mathrm{eval}}^D$. As summarized in Table 1, predictive tasks include MovieLens-1M movie recommendation, MIND news recommendation [Wu et al., 2020], and PENS next-news prediction [Ao et al., 2021]; generative tasks include PENS personalized headline generation [Ao et al., 2021] and OpenAI-Reddit TL;DR personalization [Völske et al., 2017]. MovieLens tests whether watch/rating-style events can be rewritten as signed movie-locus evidence; MIND supports PAH/MDH/LDH comparisons; and PENS links same-user prediction and generation under one news-side action-on-item schema. For pooled schema-training, we combine training trajectories after UIG rewriting. Predictive logs provide signed item-side evidence, such as watch/rating, click, skip, and ignore events; generative logs provide command-like events, such as headline or summarization requests. Following Chatterjee et al. [2025], augmentation is applied only

Table 1: **Evaluation coverage under the shared action-on-item schema.** Predictive settings use one positive target and 699 sampled negatives per test instance. Generative settings use autoregressive evaluation, where predictions are appended to the running trajectory.

Task	Dataset	Schema View	Protocol	Primary Metrics
Movie recommendation	MovieLens-1M	Watch/rating-like events as positive or negative evidence over movie target loci	Sequential ranking with 1 positive and 699 negatives	MRR, nDCG, HR@10
News recommendation	MIND	Click/ignore events as signed evidence over news target loci	Sequential ranking with 1 positive and 699 negatives	MRR, nDCG, HR@10
Next-news prediction	PENS	Click/skip events as signed evidence over news target loci	Sequential ranking with 1 positive and 699 negatives	MRR, nDCG
Headline generation	PENS	Headline request as command-like evidence over news target loci	Autoregressive personalized generation	PSE-JSD, PSE-SU4, PSE-METEOR
TL;DR generation	OpenAI-Reddit	Feedback-derived positive/negative evidence plus summarization command over response target loci	Eventized TL;DR personalization stress test	PSE-JSD, PSE-SU4, PSE-METEOR

to training pools for surrogate response nodes (see Appendix C). At autoregressive evaluation time, each prediction or generation for d_q is appended back into $\tau^h_{u_j}$ before the next step.

Baselines and Metrics. Predictive baselines follow the PAH/MDH/LDH taxonomy in Section 3. PAH baselines expose order-insensitive profiles: `BPR-MF` for MovieLens [Rendle et al., 2009], and `NAML`/`PLM-NR` for news-side prediction [Wu et al., 2019a, 2021a]. MDH baselines expose recurrent or local states: `GRU4Rec`/`Caser` for MovieLens [Hidasi et al., 2016, Tang and Wang, 2018], and `EBNR`/`GRU`-style encoders for news [Okura et al., 2017, Hidasi et al., 2016]. LDH baselines expose full-prefix attention, diffusion, multi-interest, or state-space dynamics: `S`^{3}`Rec`, `Diff4Rec`, `SASRec`, `BERT4Rec`, `Mamba4Rec`, and `SSD4Rec` for MovieLens [Zhou et al., 2020, Wu et al., 2023b, Ho and Nguyen, 2024, Sun et al., 2019, Liu et al., 2024, Zhang et al., 2025]; and `NRMS`, `MINER`, `FastFormer`, `Mamba4Rec`, and state-space variants for MIND/PENS [Wu et al., 2019c, Li et al., 2022a, Wu et al., 2021b, Liu et al., 2024, Xiao et al., 2025]. These models mix temporal evidence implicitly. However, they do not explicitly route schema-normalized signed and command-like evidence through long-, short-, and episodic traces with command-conditioned readout. For generation, we compare with PENS encoder-decoder variants using `NAML`, `EBNR`, and `NRMS` [Ao et al., 2021, Wu et al., 2019a, Okura et al., 2017, Wu et al., 2019c], with `GTP` and `Signature-Phrase` [Song et al., 2023, Cai et al., 2023], and with prompted LLMs under zero-shot, two-shot, and prompt-chaining settings, including `LLaMA`, `Mistral`, `DeepSeek`, `Gemini`, and `Qwen` [Touvron et al., 2023, Jiang et al., 2023, DeepSeek-AI et al., 2025, Comanici et al., 2025, Qwen et al., 2025]. Unlike prompted-history baselines, `PerTIDE` learns a reusable user-specific preference-flow state before command readout. For prediction, we report MRR, nDCG@5, nDCG@10, and HR@10 with the task-specific subset in Table 1. For generation, we use PerSEval [Dasgupta et al., 2024] as the primary personalization-sensitive metric family, reporting PSE-JSD, PSE-SU4, and PSE-METEOR, alongside ROUGE, BLEU-4, BERTScore, and human judgments for headline generation (details in Appendices E–H).

6 Results and Insights

We evaluate `PerTIDE` through three questions following our RQs. All results are reported via pairwise resampled trajectory-level tests (see App. I.3).

6.1 RQ1: Does action-on-item flow improve prediction?

Cross-dataset predictive performance. We test whether a reusable action-on-item encoder improves over task-native history encoders that expose one aggregated profile, one recurrent state, or one full-prefix stream. We see that `PerTIDE` improves over the strongest predictive baselines on all three predictive settings: **+2.36/+3.47** MRR/nDCG over `SSD4Rec` on MovieLens-1M, **+10.05/+9.01** over `MINER` on MIND, and **+0.90/+1.42** over `LSTUR` on PENS-Structured. Hence, these gains hold across movie, news, and headline-prediction histories after native logs are rewritten as action-on-item trajectories. MIND gives the clearest mechanism test because it covers PAH, MDH, and LDH baselines. Within PAH, `PLM-NR` improves over `NAML` by **+2.64** MRR and **+2.98** HR@10. Within MDH, `GRU-B` improves over `EBNR` by **+1.38** MRR, but remains below the best PAH baseline. Within LDH, `MINER` is strongest at 36.60 MRR and 46.30 HR@10, while `PerTIDE` reaches **46.65** MRR and

Table 2: **Cross-task scoreboard for action-on-item preference flow.** For each dataset-task pair, we compare PerTIDE against the *strongest* baselines that do not explicitly combine schema-normalized action-on-item evidence, trace-factorized persistence, and command-conditioned readout. Predictive rows report MRR / nDCG, while generative rows report PSE-JSD / PSE-METEOR. Detailed results are in Tables 10-12 for predictive settings; Tables 14-16.

Task	Dataset	Baseline (Regime)	Baseline	PerTIDE	Δ
Predictive Personalization: MRR / nDCG					
Movie Recommendation	MovieLens-1M	SSD4Rec (LDH)	14.95 / 20.31	**17.31 / 23.78**	**+2.36 / +3.47**
Movie Recommendation	MovieLens-1M	GRU4Rec (MDH)	12.76 / 16.42	**17.31 / 23.78**	+4.55 / +7.36
Next News Prediction	PENS-Structured	LSTUR (LDH)	9.04 / 10.16	**9.94 / 11.58**	**+0.90 / +1.42**
Next News Prediction	PENS-Structured	TrRMIo (LDH)	8.02 / 8.53	**9.94 / 11.58**	+1.92 / +3.05
News Recommendation	MIND	MINER (LDH)	36.60 / 40.20	**46.65 / 49.21**	**+10.05 / +9.01**
News Recommendation	MIND	FastFormer+PLM-NR (LDH)	35.91 / 39.16	**46.65 / 49.21**	+10.74 / +10.05
Generative Personalization: PSE-JSD / PSE-METEOR					
Personalized Headline Generation	PENS	DeepSeek-R1-32B (LLM)	0.263 / 0.174	**0.366 / 0.355**	**+0.103 / +0.181**
Personalized Headline Generation	PENS	Discretized SSM (LDH)	0.037 / 0.031	**0.366 / 0.355**	+0.329 / +0.324
TL;DR Personalization	OpenAI-Reddit	DeepSeek-14B (LLM)	0.243 / 0.109	**0.285 / 0.294**	**+0.042 / +0.185**
TL;DR Personalization	OpenAI-Reddit	LLaMA-13B (LLM)	0.232 / 0.107	**0.285 / 0.294**	+0.053 / +0.187

55.48 HR@10. *Hence, besides longer-prefix modeling, the gain appears when native interactions are rewritten into a shared schema and the encoder learns how signed and command-like evidence should enter, persist, and be read out* (see Table 2).

6.2 RQ2: Do entry, readout, and traces matter?

Matched-input state modeling. We first separate the proposed retention design from the effect of using richer inputs or multiple state-space branches (Table 3).

Under the same schema-normalized interaction-block inputs on MIND, a single Discretized-SSM-B reaches 35.14 MRR, while the three-branch Tri-SSM-B reaches 42.53. PerTIDE reaches **46.65** MRR, improving over Tri-SSM-B by **+4.12** MRR, together with **+5.37** nDCG@5 and **+9.15** HR@10. Thus, the gain persists after controlling both the input representation and the use of three state-space branches.

Table 3: **Same-input state-model control on MIND.** All models consume the same ordered schema-normalized interaction-block inputs. Scores are MRR/nDCG@5/HR@10.

Model	MRR / nDCG@5 / HR@10
MeanPool-B	29.43 / 31.22 / 33.17
GRU-B	32.64 / 35.17 / 37.92
Discretized SSM-B	35.14 / 38.53 / 37.27
Tri-SSM-B	42.53 / 43.84 / 46.33
PerTIDE	**46.65 / 49.21 / 55.48**

Action entry and command readout. We next isolate the two components that make the shared schema usable. Action conditioning controls how a target locus updates the user state under a positive, negative, or command-like role, while command-conditioned readout controls how the active downstream command consumes that state. We observe from Table 4 that removing action conditioning reduces MIND performance from **46.65** to 40.13 MRR and from **55.48** to 46.35 HR@10, giving drops of **6.52** MRR and **9.13** HR@10. Removing command-conditioned readout reduces performance to 42.51 MRR and 51.06 HR@10, giving drops of **4.14** MRR and **4.42** HR@10. Partial variants remain stronger than strong task-native baselines, but the full model is strongest when action-conditioned entry, trace-factorized persistence, and command-conditioned consumption are all present. *Hence, simply putting all histories into the same schema is not enough. A click, skip, rating, and request should affect the user state in different ways, even when they point to the same target locus. Similarly, the same user state should be read differently for ranking, summarization, or generation.*

Supplied action semantics. A fixed-trajectory corruption test further shows that the encoder uses the supplied action roles. On PENS, progressively flipping positive/negative roles reduces MRR from **12.14** at 0% corruption to 10.13, 9.17, 7.71, and **2.81** at 10%, 25%, 50%, and 100% corruption, respectively, while trajectories, targets, and candidate sets remain unchanged.

Trace specialization. We then test whether long-, short-, and episodic traces behave as intended under controlled schema-trajectory regimes. L-Tr emphasizes stable evidence, S-Tr emphasizes recency-

Table 4: **Schema-reuse mechanism ladder.** We compare variants laterally within each setting. Each score cell reports *score* $[\Delta_{\text{base}}]$, where Δ_{base} is the gain over the *strongest* non-PerTIDE baseline in that setting. The final row reports the gap to the full PerTIDE variant. MIND uses MRR/HR@10; PENS uses PSE-JSD/PSE-METEOR.

A. MIND prediction: MRR / HR@10

Quantity	MINER	PerTIDE w/o AC	PerTIDE w/o CR	Full PerTIDE
Mechanism	Task-native LDH 36.60 / 46.30	Traces + CR 40.13 / 46.35	AC + traces 42.51 / 51.06	AC + traces + CR **46.65 / 55.48**
Score $[\Delta_{\text{base}}]$	$(0.00/0.00)$	$(+3.53/+0.05)$	$(+5.91/+4.76)$	$(+10.05/+9.18)$
Gap to full	$-10.05/-9.18$	$-6.52/-9.13$	$-4.14/-4.42$	$0/0$
Interpretation	Strong task-native LDH.	Traces help; action role is absent.	AC helps; command readout is absent.	Entry, persistence, and readout are combined.

B. PENS generation: PSE-JSD / PSE-METEOR

Quantity	DeepSeek-R1-32B	PerTIDE w/o AC	PerTIDE w/o CR	PerTIDE (UFI)	Full PerTIDE (UGI)
Mechanism	Prompted history 0.263 / 0.174	Traces + CR 0.314 / 0.292	AC + traces 0.317 / 0.284	AC + traces + fused decoder 0.332 / 0.303	AC + traces + gated decoder **0.366 / 0.355**
Score $[\Delta_{\text{base}}]$	$(0.000/0.000)$	$(+0.051/+0.118)$	$(+0.054/+0.110)$	$(+0.069/+0.129)$	$(+0.103/+0.181)$
Gap to full	$-0.103/-0.181$	$-0.052/-0.063$	$-0.049/-0.071$	$-0.034/-0.052$	$0/0$
Interpretation	Prompted history; no preference-flow state.	User state retained; action entry removed.	Encoder retained; command readout removed.	Fused decoder consumption.	Gated decoder consumption.

Table 5: **Controlled schema-trajectory regime diagnostic.** We construct 1K trajectory variants emphasizing stable evidence (L-Tr), short-term evidence (S-Tr), and finite episodic evidence (E-Tr). These diagnostics test whether trace-specific branches specialize under schema-normalized evidence regimes. We report MRR/nDCG@10. Additional trace-level results are in Table 13.

Dataset	Family	Model / Variant	L-Tr	S-Tr	E-Tr
MovieLens	Baseline	Mamba4Rec	26.43/29.38	26.12/28.87	25.72/28.31
		GRU4Rec	24.17/27.32	23.65/26.58	22.13/24.71
		SASRec	25.82/28.16	25.85/28.45	25.17/27.91
	PerTIDE	PerTIDE-L	27.31/29.96	27.18/29.35	25.83/28.77
		PerTIDE-S	25.21/28.64	27.13/29.22	22.16/24.84
		PerTIDE-E	22.19/24.08	22.75/25.04	27.14/30.36
		PerTIDE-L+S+E	29.33/32.17	29.04/31.85	29.12/32.09
MIND	Baseline	NAML	46.21/51.23	46.25/51.35	45.32/49.18
		EBNR	47.44/52.25	48.32/53.17	44.19/48.79
		LSTUR	49.31/54.41	48.93/53.86	45.64/49.84
	PerTIDE	PerTIDE-L	49.54/54.78	48.65/53.22	47.32/51.66
		PerTIDE-S	49.21/53.94	50.07/55.35	48.65/52.81
		PerTIDE-E	47.31/51.26	47.13/51.06	50.35/55.84
		PerTIDE-L+S+E	52.75/59.32	51.76/57.92	51.34/57.56

sensitive evidence, and E-Tr emphasizes finite-window episodic evidence. These are controlled probes of evidence persistence rather than temporally pure natural domains. We notice that the expected single-trace branch is usually strongest under its corresponding regime. On MovieLens-1M, PerTIDE-L is strongest among single-trace variants on L-Tr, PerTIDE-S is strongest or near-strongest on S-Tr, and PerTIDE-E is strongest on E-Tr. On MIND, the pattern is sharper: PerTIDE-L gives **49.54/54.78** on L-Tr, PerTIDE-S gives **50.07/55.35** on S-Tr, and PerTIDE-E gives **50.35/55.84** on E-Tr. The fused $L + S + E$ state remains strongest because realistic histories mix evidence types, improving over the strongest baseline by **+2.90/+2.79**, **+2.92/+2.98**, and **+3.40/+3.78** on MovieLens-1M, and by **+3.44/+4.91**, **+2.83/+4.06**, and **+5.70/+7.72** on MIND across L-Tr, S-Tr, and E-Tr. *Hence, each trace is preferred where its retention pattern is useful, while the fused state wins when stable, recency-sensitive, and episodic command evidence coexist* (results from Table 5).

Trace complementarity. Natural-task ablations show the same effect from a different view. On MIND, the episodic trace alone reaches 24.18 MRR, $L + S$ reaches 41.26, and adding E raises the full model to **46.65**. On PENS generation, the corresponding PSE-METEOR values are 0.045, 0.182, and **0.355**. Thus, a trace can be weak in isolation while still contributing complementary evidence to the fused state (Table 13).

6.3 RQ3: Does the predictive state transfer?

Prediction to generation. We test whether the user-specific preference-flow state used for prediction can also be consumed by a generative command. PENS is the cleanest setting: PENS-Structured tests prediction over the same history abstraction, while PENS headline generation tests whether that abstraction drives generation. We observe from Tables 2 and 4 that PerTIDE improves over the strongest two-shot large language model baseline, DeepSeek-R1-32B, by **+0.103/+0.181** on PSE-

JSD/PSE-METEOR, and over the strongest non-LLM compressed-history baseline, `Discretized SSM`, by **+0.329/+0.324**. Decoder variants show that command consumption matters: UFI reaches 0.332/0.289/0.303 on PSE-JSD/PSE-SU4/PSE-METEOR, while UGI reaches **0.366/0.341/0.355**, giving **+0.034/+0.052/+0.052** (App. Table 14). On OpenAI-Reddit, an eventized TL;DR stress test, `PerTIDE` improves over `DeepSeek-14B` by **+0.042/+0.185** and over `LLaMA-13B` by **+0.053/+0.187** on PSE-JSD/PSE-METEOR. *Hence, the same preference-flow state remains useful for generation when feedback-derived interactions can be rewritten into positive/negative evidence and a summarization command over target loci.* For details, see App. J.

Source-only encoder reuse. We additionally test reuse when the target domain is excluded from encoder training (Table 6). A PerTIDE core trained only on PENS is transferred to MovieLens and kept fully frozen, while only a fresh ranking head is trained on 3,000 target rows. With the same target data, candidates, head, loss, and optimization, a random frozen core with a trained head reaches 1.115 MRR, while the PENS-trained frozen core with the same trained target head reaches **16.344** MRR. The corresponding gains are **+15.23** MRR, **+15.22** nDCG@10, and **+18.84** HR@10. An untrained target head over the PENS-trained core reaches only 1.371 MRR, showing that the source-learned encoder remains useful after domain change, while the target consumer still requires alignment.

Table 6: **Source-only frozen-core reuse from PENS to MovieLens.** The encoder remains frozen during target adaptation. Scores are MRR/nDCG@10/HR@10.

Frozen core	Head	MRR / nDCG / HR
Random	Trained	1.115 / 0.780 / 1.44
PENS-trained	Untrained	1.371 / 0.960 / 2.18
PENS-trained	**Trained**	**16.344 / 16.002 / 20.28**

Schema proximity and reuse. On 500 MovieLens trajectories, larger schema distance is associated with higher frozen/adapted target risk ($\rho = 0.109/0.160$) and smaller frozen/adapted reciprocal-rank gain ($\rho = -0.089/-0.114$). The effects are modest, but their directions are consistent with compatibility-sensitive reuse; full user-clustered confidence intervals are reported in App. Table 18.

7 Broader Applicability and Limitations

We use MPChat as a broader-applicability stress test rather than as primary evidence, since dialogue histories do not expose the clean action-on-item preference signals found in click, skip, rating, or request logs. MPChat includes dialogue turns, speaker roles, persona statements, images, entity mentions, and multimodal grounding cues; its UIG construction therefore depends on inferred discourse structure, speaker dynamics, persona grounding, and multimodal context. We report response prediction, speaker identification, and dialogue generation results in Appendix J.6. `PerTIDE` remains usable when dialogue turns are eventized into shared action roles, improving over the strongest non-MTSH dialogue-generation baseline by $+8.48/+7.08/+9.09/+3.52$ on RG-2/RG-L/BLEU-4/BScore. However, response prediction and speaker identification show that explicit persona and multimodal grounding still provide large gains. This agrees with Appendix B.2: transfer depends on schema-trajectory distance, target-locus embedding quality, and command-semantics compatibility. Hence, `PerTIDE` provides *schema-level generalizability across eventizable personalization trajectories, not unrestricted transfer across arbitrary user-interaction logs.* Stronger dialogue personalization requires discourse-aware UIG construction with dialogue acts, speaker-role transitions, entity mentions, topic-shift markers, acceptance/rejection signals, and confidence-weighted persona links (App J.6).

8 Conclusion

We present `PerTIDE` as a reusable history encoder for personalization under a shared action-on-item event schema. Native interactions are rewritten as positive, negative, or command-like evidence over target-locus embeddings, which gives heterogeneous user logs a common training substrate without manually fixing action or item taxonomies. On this substrate, MTSH models how evidence enters a user-specific state, how it persists across long-term, short-term, and episodic traces, and how the active command reads it out. Across prediction and generation, `PerTIDE` improves over strong task-native and prompted baselines, while ablations show that action conditioning, command-conditioned readout, and trace composition each contribute to the gains. The MPChat stress test clarifies the boundary of the claim: `PerTIDE` supports schema-level generalizability across eventizable personalization trajectories, not unrestricted transfer across arbitrary user-interaction logs.

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, July 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, July 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, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.7. URL https://aclanthology.org/2021.acl-long.7/.

Pengshan Cai, Kaiqiang Song, Sangwoo Cho, Hongwei Wang, Xiaoyang Wang, Hong Yu, Fei Liu, and Dong Yu. Generating user-engaging news headlines. 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 3265–3280, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.183. URL https://aclanthology.org/2023.acl-long.183/.

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 & Data Mining*, KDD '20, page 2942–2951, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450379984. 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.

Peng Chen, Yingying Zhang, Yunyao Cheng, Yang Shu, Yihang Wang, Qingsong Wen, Bin Yang, and Chenjuan Guo. Pathformer: Multi-scale transformers with adaptive pathways for time series forecasting. In *International Conference on Learning Representations*, 2024.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025. URL https://arxiv.org/abs/2507.06261.

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.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu,

Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, and Chengqi et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Chen Gao, Xiangnan He, Dahua Gan, Xiangning Chen, Fuli Feng, Yong Li, Tat-Seng Chua, and Depeng Jin. Neural multi-task recommendation from multi-behavior data. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 1554–1557, 2019. doi: 10.1109/ICDE.2019.00140.

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. Recommendation as language processing (rlp): A unified pretrain, personalized prompt predict paradigm (p5), 2023. URL https://arxiv.org/abs/2203.13366.

Ignat Georgiev, Varun Giridhar, Nicklas Hansen, and Animesh Garg. Pwm: Policy learning with multi-task world models. In *International Conference on Learning Representations*, 2025.

Nicklas Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control, 2024. URL https://arxiv.org/abs/2310.16828.

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

Trong Dang Huu Ho and Sang Thi Thanh Nguyen. Self-attentive sequential recommendation models enriched with more features. In *Proceedings of the 2024 8th International Conference on Deep Learning Technologies*, ICDLT '24, page 49–55, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400716867. doi: 10.1145/3695719.3695727. URL https://doi.org/10.1145/3695719.3695727.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Kyungho Kim, Sunwoo Kim, Geon Lee, Jinhong Jung, and Kijung Shin. Multi-behavior recommender systems: A survey, 2025. URL https://arxiv.org/abs/2503.06963.

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, page 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 2022a. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.29. URL https://aclanthology.org/2022.findings-acl.29/.

Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. A persona-based neural conversation model. In Katrin Erk and Noah A. Smith, editors, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 994–1003, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1094. URL https://aclanthology.org/P16-1094/.

Shuokai Li, Ruobing Xie, Yongchun Zhu, Xiang Ao, Fuzhen Zhuang, and Qing He. User-centric conversational recommendation with multi-aspect user modeling. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 223–233, New York, NY, USA, 2022b. Association for Computing Machinery. ISBN 9781450387323. doi: 10.1145/3477495.3532074. URL https://doi.org/10.1145/3477495.3532074.

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.

Zhixuan Lin, Evgenii Nikishin, Xu Owen He, and Aaron Courville. Forgetting transformer: Softmax attention with a forget gate, 2025. URL https://arxiv.org/abs/2503.02130.

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.

Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H. Chi. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, page 1930–1939, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355520. doi: 10.1145/3219819.3220007. URL https://doi.org/10.1145/3219819.3220007.

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, page 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.

Zhonghong Ou, Xiao Zhang, Yifan Zhu, Shuai Lyu, et al. LS-TGNN: Long and short-term temporal graph neural network for session-based recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 12426–12434, 2025.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Zhaochun Ren, Zhi Tian, Dongdong Li, Pengjie Ren, Liu Yang, Xin Xin, Huasheng Liang, Maarten de Rijke, and Zhumin Chen. Variational reasoning about user preferences for conversational recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 165–175, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450387323. URL https://doi.org/10.1145/3477495.3532077.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, page 452–461, Arlington, Virginia, USA, 2009. AUAI Press. ISBN 9780974903958.

Tankred Saanum, Peter Dayan, and Eric Schulz. Simplifying latent dynamics with softly state-invariant world models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 38355–38382. Curran Associates, Inc., 2024. doi: 10.52202/079017-1212. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/43ba0466af2b1ac76aa85d8fbec714e3-Paper-Conference.pdf.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. LaMP: When large language models meet personalization. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7370–7392, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.399. URL https://aclanthology.org/2024.acl-long.399/.

Vaisakh Shaj, Saleh Gholam Zadeh, Ozan Demir, Luiz Ricardo Douat, and Gerhard Neumann. Multi time scale world models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Qijie Shen, Hong Wen, Jing Zhang, and Qi Rao. Hierarchically fusing long and short-term user interests for click-through rate prediction in product search. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 1767–1776, 2022.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: masked and permuted pre-training for language understanding. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

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/.

Chia-Yi Su and Collin McMillan. Distilled gpt for source code summarization. *Automated Software Engg.*, 31(1), March 2024. ISSN 0928-8910. doi: 10.1007/s10515-024-00421-4. URL https://doi.org/10.1007/s10515-024-00421-4.

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, page 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.

Zhen Tan, Jun Yan, I-Hung Hsu, Rujun Han, Zifeng Wang, Long Le, Yiwen Song, Yanfei Chen, Hamid Palangi, George Lee, Anand Rajan Iyer, Tianlong Chen, Huan Liu, Chen-Yu Lee, and Tomas Pfister. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents. 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 8416–8439, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.413. URL https://aclanthology.org/2025.acl-long.413/.

Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, page 565–573, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355810. doi: 10.1145/3159652.3159656. URL https://doi.org/10.1145/3159652.3159656.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv e-prints*, pages arXiv–2307, 2023.

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, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.169. URL https://aclanthology.org/2023.findings-emnlp.169.

Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. TL;DR: Mining Reddit to learn automatic summarization. In Lu Wang, Jackie Chi Kit Cheung, Giuseppe Carenini, and Fei Liu, editors, *Proceedings of the Workshop on New Frontiers in Summarization*, pages 59–63, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4508. URL https://aclanthology.org/W17-4508.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. In *International Conference on Learning Representations*, 2024.

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, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. Npa: Neural news recommendation with personalized attention. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, KDD '19, page 2576–2584. ACM, July 2019b. doi: 10.1145/3292500.3330665. URL http://dx.doi.org/10.1145/3292500.3330665.

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, November 2019c. Association for Computational Linguistics. doi: 10.18653/v1/D19-1671. URL https://aclanthology.org/D19-1671.

Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. Empowering news recommendation with pre-trained language models, 2021a. URL https://arxiv.org/abs/2104.07413.

Chuhan Wu, Fangzhao Wu, Tao Qi, Yongfeng Huang, and Xing Xie. Fastformer: Additive attention can be all you need, 2021b. URL https://arxiv.org/abs/2108.09084.

Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, and Ming Zhou. MIND: A large-scale dataset for news recommendation. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3597–3606, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.331. URL https://aclanthology.org/2020.acl-main.331.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 22419–22430. Curran Associates, Inc., 2021c. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/bcc0d400288793e8bdcd7c19a8ac0c2b-Paper.pdf.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023a.

Qingyang Wu and Zhou Yu. Stateful memory-augmented transformers for efficient dialogue modeling. In Yvette Graham and Matthew Purver, editors, *Findings of the Association for Computational Linguistics: EACL 2024*, pages 853–867, St. Julian's, Malta, March 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.findings-eacl.57/.

Zihao Wu, Xin Wang, Hong Chen, Kaidong Li, Yi Han, Lifeng Sun, and Wenwu Zhu. Diff4rec: Sequential recommendation with curriculum-scheduled diffusion augmentation. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM '23, page 9329–9335, New York, NY, USA, 2023b. Association for Computing Machinery. ISBN 9798400701085. doi: 10.1145/3581783.3612709. URL https://doi.org/10.1145/3581783.3612709.

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.

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

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.

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, June 2024. 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, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1205. URL https://aclanthology.org/P18-1205/.

Yifeng Zhang, Haohao Qu, Liangbo Ning, Wenqi Fan, and Qing Li. Ssd4rec: A structured state space duality model for efficient sequential recommendation. *ACM Trans. Inf. Syst.*, 44(2), December 2025. ISSN 1046-8188. doi: 10.1145/3773038. URL https://doi.org/10.1145/3773038.

Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 1893–1902, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368599. doi: 10.1145/3340531.3411954. URL https://doi.org/10.1145/3340531.3411954.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*, 2022.

A Datasets

A.1 MIND-Large Dataset

MIND (Large) [Wu et al., 2020] is the full-scale release of the Microsoft News recommendation dataset, collected from anonymized user behavior logs. It contains 1,000,000 users and 161,013 English news articles, and the associated interaction logs include over 15 million impressions on the platform [Wu et al., 2020]. The data is organized around impression logs that record the impression ID, anonymized user ID, timestamp, a user click-history list before the impression, and the displayed candidate news with click/non-click labels; the release also includes per-article metadata (e.g., title, abstract, body, category, and entities) [Wu et al., 2020]. A small subsampled variant (MIND-small) is also provided by the dataset maintainers by randomly sampling 50,000 users and their behavior logs from the full dataset [Wu et al., 2020].

A.2 PENS Dataset

PENS [Ao et al., 2021] contains 113,762 news articles spanning 15 topical categories. Each article includes an ID, title (avg. 10.5 words), body (avg. 549 words), and category label; titles are also linked to WikiData entities. The dataset further provides user interaction logs (impressions and clicks) and incorporates news bodies and headlines sourced from MIND [Wu et al., 2020].

PENS training set. For training, 500k user–news impressions were sampled from June 13 to July 3, 2019. Each log entry is recorded as $[\mathrm{uID}, \mathrm{tmp}, \mathrm{clkNews}, \mathrm{uclkNews}, \mathrm{clkedHis}]$, where $\mathrm{clkNews}$ and $\mathrm{uclkNews}$ denote clicked and unclicked news in the impression, and $\mathrm{clkedHis}$ is the user's previously clicked history sorted by click time. The resulting training pool for `PerTIDE` exhibits substantial temporal variation in user interactions, motivating explicit modeling of preference dynamics in offline personalization settings.

PENS test set. To construct an offline evaluation set, 103 English-speaking students first reviewed 1,000 headlines (stage-1), then selected 50 articles and wrote preferred headlines (gold-reference summaries) for 200 previously unseen articles (stage-2). Each test article is associated with four participant-written references. Editorial checks verified factual correctness and removed incorrect headlines, yielding a high-quality set of personalized gold references for evaluation.

A.3 OpenAI (Reddit) Dataset

The OpenAI (Reddit) dataset [Völske et al., 2017] contains 123,169 Reddit posts drawn from 29 subreddits. It provides both human-written and OpenAI-generated summaries and is organized into two splits: *Comparisons* (training/validation) and *Axis* (validation/testing). A curated subset of 1,038 posts was processed under 13 summarization policies, producing 7,713 summaries, which were evaluated by 64 annotators via paired comparisons with auxiliary signals (e.g., selection preference, confidence, and ratings along dimensions such as accuracy, coherence, coverage, and overall quality). In contrast to PENS, these annotations are not indexed by stable user identities or reading histories.

A.4 MPChat Dataset

MPChat [Ahn et al., 2023] is a multimodal persona-grounded dialogue dataset in which personas are represented as image–sentence pairs intended to capture episodic memories, and dialogues are curated to be grounded in these multimodal persona elements. The dataset is released with retrieval-style benchmarks for multimodal persona-grounded dialogue understanding, including next response prediction, grounding persona prediction, and speaker identification. In total, MPChat contains 15,000 multi-turn dialogues with 42,531 utterances written by 25,877 users, and it is split chronologically into 11,975/1,516/1,509 dialogues for train/validation/test [Ahn et al., 2023].

B UIG Construction and Schema-Level Generalizability

B.1 Dataset-Specific UIG Construction

In the parlance of UIG, preference datasets suitable for personalized generation (summarization or dialogue generation) training and evaluation are of three categories– (i) those which can be directly

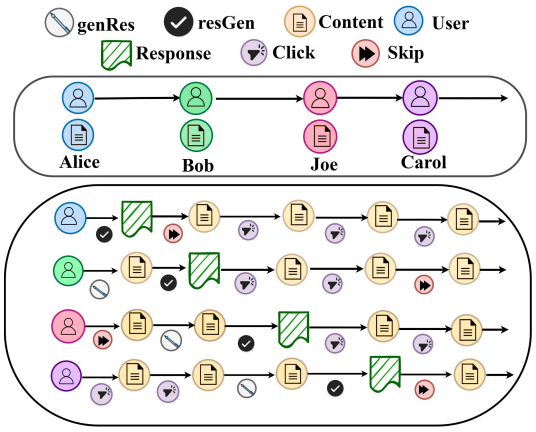


Figure 2: Construction of the User Interaction Graph (UIG) from multi-user, multi-turn dialogue. Turn-level interactions are annotated with behavioral actions and sentiment signals, then projected into a unified UIG over participating users and content nodes.

modeled into a trajectory pool $\mathcal{T}$ (e.g., PENS dataset [Ao et al., 2021]) and (ii) those which lack user trajectories but contain discrete d-nodes, *model-generated* r-nodes (in contrast to user-generated r-nodes as per UIG definition), and *subjective* user feedback in the form of rating and the associated confidence score for that rating (e.g. OpenAI-Reddit dataset [Völske et al., 2017]), (iii) interleaved sequence of multi-user r-nodes without apparent preference interactions Ahn et al. [2023], Zhang et al. [2018]. We describe the UIG (i.e., the base u-tier) construction method for both types as follows:

PENS-styled Datasets. The construction of UIG is straightforward in the first case and is done in two steps. In the first step, *click* and *skip* interactions in the train dataset are mapped to document nodes (d-nodes) as incoming edges, forming the corresponding u-tier pool $\mathcal{T}$. As an example, for the PENS dataset, the *clkNews* interaction corresponds to a *click* edge and *uclkNews* to a *skip* edge, forming $\mathcal{T}^{\text{PENS}}$. However, PENS dataset lacks user-specific r-nodes (i.e., true interest evolution over time), rendering $\mathcal{T}^{\text{PENS}}$ an *incomplete representation of the user dynamic preference*[1]. We address this issue in the second step, where we *incorporate* surrogate summary responses (a chosen sentence from the corresponding d-node) as r-nodes into $\mathcal{T}$ with the addition of *genRes* and *resGen* edges.

OpenAI-styled Datasets. For the second category of datasets, we first do a pre-construction classification of clicked and skipped d-nodes for every human rater u_j. This is done based on a simple heuristic of selecting those d-nodes as clicked which has at least one corresponding model-generated summary (note that there can be multiple models) that received a confidence score above a chosen threshold (in the case of OpenAI-Reddit, we chose that to be 6 out of 9). We then select the best model-generated summary (i.e., one with the highest rating given by u_j) as the surrogate expected r-node for u_j. We then randomly sequence all such $(d-r)$-node pairs along with the skipped d-nodes to form τ^{u_j} (thereby $\mathcal{T}^{\text{OAI}}$). This method makes UIG-modeling *compatible with most summarization datasets that are not PENS-styled*.

MPChat-styled Datasets. For multi-party, multi-turn conversation datasets (e.g., MPChat), UIG construction follows the same two-tier intuition as PENS-styled data, but replaces explicit *click/skip* logs with inferred stance signals from dialogue. We treat each dialogue as an ordered stream and build trajectories by pairing (i) a context set of dialogue utterances as d-nodes and (ii) the target speaker's subsequent response as an response-node. Concretely, for a target speaker (e.g., Alice), we map the most recent non-target utterances within the local context (recency-biased to the latest turns) into candidate d-nodes, and we assign each candidate an implicit binary preference label based on whether the target response expresses agreement/positive stance or disagreement/negative stance toward that context. Positive stance is mapped to a *pos* incoming edge to the corresponding d-node, while negative (including neutral collapsed to negative) is mapped to a *neg* edge. This produces a trajectory pool $\mathcal{T}^{\text{MPChat}}$ that is structurally compatible with UIG training and evaluation pipelines, while remaining fully self-contained.

[1]It is important to note that despite this, most recent frameworks train on $\mathcal{T}^{\text{PENS}}$ using history or document titles as "pseudo-targets" or via unsupervised learning [Ao et al., 2021, Song et al., 2023, Yang et al., 2023, Lian et al., 2025].

Algorithm 1 HYBRIDSTANCEINFER: MPChat-Style Turn-Level Stance Inference

Require: Context utterances $\{d\}$; target response s; sentiment encoder $\mathrm{SentEnc}$; text encoder TxtEnc; cue lexicons $\{\mathcal{C}\}$
Ensure: Binary stance labels for linked context turns

$x_s \leftarrow \mathrm{lower}(s)$; $\mathbf{e}_s \leftarrow \mathrm{TxtEnc}(x_s)$; $\sigma_s \leftarrow \mathrm{SentEnc}(x_s)$
$\kappa \leftarrow \mathrm{splitClauses}(x_s)$
for each recent context turn $d \in \{d\}$ **do**
  $x_d \leftarrow \mathrm{lower}(d)$; $\mathbf{e}_d \leftarrow \mathrm{TxtEnc}(x_d)$
  $p \leftarrow 0,\ n \leftarrow 0$ (positive / negative evidence)
  for each clause $\kappa_i \in \kappa$ **do**
    $(p, n) \leftarrow \mathrm{CueScore}(\kappa_i, \{\mathcal{C}\}, p, n)$
  end for
  $(p, n) \leftarrow \mathrm{SentScore}(\sigma_s, p, n)$
  $(p, n) \leftarrow \mathrm{SimScore}(\cos(\mathbf{e}_s, \mathbf{e}_d), p, n)$
  $\ell_d \leftarrow \mathbb{I}[p > n]$ (neutral $\rightarrow$ negative)
end for
return $\{\ell_d\}$

B.2 Schema-Level Generalizability

We formalize the restricted generalizability induced by the action-on-item interface. The goal is not to prove that `PerTIDE` generalizes to arbitrary domains. Rather, we show that once native interactions are rewritten into the shared action-on-item event schema, the model is (i) invariant to native action-label relabeling, (ii) stable under target-locus embedding perturbations, and (iii) admits a transfer bound between eventizable trajectory distributions. This formalizes the sense in which diverse trajectory data can enter a common training substrate without manually imposing a fixed taxonomy over domain actions or item kinds.

Schema Map. Let D denote a personalization domain with native action set $\mathcal{A}_D$ and target-node set $\mathcal{X}_D$. A native event is a pair $e = (\alpha, x)$, where $\alpha \in \mathcal{A}_D$ is the domain-native action and $x \in \mathcal{X}_D$ is the interacted item or response node. The action-on-item interface consists of (i) an action-role map

$$\rho_D : \mathcal{A}_D \to \mathcal{A}, \qquad \mathcal{A} = \{\texttt{positive}, \texttt{negative}, \texttt{command}\},$$

and (ii) a target-locus embedding map

$$\psi_D : \mathcal{X}_D \to \mathbb{R}^d.$$

For a native event $e = (\alpha, x)$, its schema representation is

$$S_D(e) = \big(\rho_D(\alpha), \psi_D(x)\big).$$

For a trajectory $\tau_D = (e_1, \ldots, e_T)$, we write

$$S_D(\tau_D) = \big(S_D(e_1), \ldots, S_D(e_T)\big).$$

Thus, native events such as opening a book, clicking a news item, watching a video, skipping a movie, or requesting a summary are not required to share native labels. They only need to be eventizable as positive, negative, or command-like evidence over target-locus embeddings.

PerTIDE as a Schema-Consuming Map. For each timestep t_i, `PerTIDE` forms an action-gated event cell

$$\mathbf{b}^{(t_i)} = \boldsymbol{\mu}\Big(a^{(t_i)}\Big) \odot \mathbf{e}_{tl}^{(t_i)}, \qquad \boldsymbol{\mu}(a) = \sigma(\mathrm{MLP}_{\mathrm{gate}}(a)).$$

The trace updates then take the generic form

$$\mathbf{m}_u^{(t_i,r)} = A_r \mathbf{m}_u^{(t_{i-1},r)} + B\mathbf{b}^{(t_i)}, \qquad r \in \{L, S, E\}.$$

The trace mixer gives $\mathbf{h}_{b_u}^{(t_i)} = \Phi(\mathbf{m}_u^{(t_i,L)}, \mathbf{m}_u^{(t_i,S)}, \mathbf{m}_u^{(t_i,E)})$, and command-conditioned readout gives

$$\mathbf{z}_{b_u}^{(t_i)} = G\Big(\Pi(\mathbf{g}_q, \mathbf{h}_{b_u}^{(t_i)}), \mathbf{h}_{b_u}^{(t_i)}\Big).$$

Therefore, for fixed parameters, the model consumes only the schema-level action role, the target-locus embedding, the user-specific trace state, and the command seed $\mathbf{g}_q$.

Proposition B.1 (Invariance to Native Action Relabeling)**.** *Consider two domains D and D' with native trajectories $\tau_D = (e_1, \ldots, e_T)$ and $\tau_{D'} = (e'_1, \ldots, e'_T)$, where $e_i = (\alpha_i, x_i)$ and $e'_i = (\alpha'_i, x'_i)$. Suppose that their schema representations are identical:*

$$\rho_D(\alpha_i) = \rho_{D'}(\alpha'_i), \qquad \psi_D(x_i) = \psi_{D'}(x'_i), \qquad \forall i \in \{1, \ldots, T\}.$$

Assume the same initial trace states and the same command seed $\mathbf{g}_q$. Then, for fixed `PerTIDE` *parameters,*

$$\mathbf{h}_{b_u}^{(t_i)} = \mathbf{h}_{b_{u'}}^{(t_i)} \quad \textit{and} \quad \mathbf{z}_{b_u}^{(t_i)} = \mathbf{z}_{b_{u'}}^{(t_i)} \qquad \forall i \in \{1, \ldots, T\}.$$

Proof. Since $\rho_D(\alpha_i) = \rho_{D'}(\alpha'_i)$ and $\psi_D(x_i) = \psi_{D'}(x'_i)$, the action-gated event cells are identical:

$$\boldsymbol{\mu}(\rho_D(\alpha_i)) \odot \psi_D(x_i) = \boldsymbol{\mu}(\rho_{D'}(\alpha'_i)) \odot \psi_{D'}(x'_i).$$

Thus, $\mathbf{b}^{(t_i)} = \mathbf{b}'^{(t_i)}$ for every i. The initial trace states are equal by assumption. Applying the deterministic trace recursion to equal inputs gives

$$\mathbf{m}_u^{(t_i,r)} = \mathbf{m}_{u'}^{(t_i,r)} \qquad \forall r \in \{L, S, E\}$$

by induction on i. Since Φ is deterministic, the fused states are equal:

$$\mathbf{h}_{b_u}^{(t_i)} = \mathbf{h}_{b_{u'}}^{(t_i)}.$$

The command seed $\mathbf{g}_q$ is also shared, so Π and G produce equal command-conditioned states:

$$\mathbf{z}_{b_u}^{(t_i)} = \mathbf{z}_{b_{u'}}^{(t_i)}.$$

This proves the claim. □

Interpretation. This shows that `PerTIDE` does not depend on native action names after schema rewriting. For example, "open", "click", and "watch" need not be identical native labels. If they map to the same action role and their target-locus embeddings match, `PerTIDE` produces the same state evolution. This is the first part of schema-level generalizability.

Proposition B.2 (Stability Under Target-Locus Embedding Shifts)**.** *Assume two schema-normalized trajectories have the same action-role sequence but possibly different target-locus embeddings. Let*

$$\delta_i = \left\| \mathbf{e}_{tl}^{(t_i)} - \mathbf{e}_{tl'}^{(t_i)} \right\|_2.$$

Assume $\|A_r\|_2 \le \beta_r < 1$ for each trace $r \in \{L, S, E\}$, and assume equal initial trace states. Then, for each trace,

$$\left\| \mathbf{m}_u^{(t_i,r)} - \mathbf{m}_{u'}^{(t_i,r)} \right\|_2 \le \|B\|_2 \sum_{j=1}^{i} \beta_r^{i-j} \delta_j.$$

If Φ is L_Φ-Lipschitz with respect to the concatenated trace states, then

$$\left\| \mathbf{h}_{b_u}^{(t_i)} - \mathbf{h}_{b_{u'}}^{(t_i)} \right\|_2 \le L_\Phi \|B\|_2 \sum_{r \in \{L,S,E\}} \sum_{j=1}^{i} \beta_r^{i-j} \delta_j.$$

If the readout map $F_{\mathbf{g}_q}(\mathbf{h}) = G(\Pi(\mathbf{g}_q, \mathbf{h}), \mathbf{h})$ is L_F-Lipschitz for fixed $\mathbf{g}_q$, then

$$\left\| \mathbf{z}_{b_u}^{(t_i)} - \mathbf{z}_{b_{u'}}^{(t_i)} \right\|_2 \le L_F L_\Phi \|B\|_2 \sum_{r \in \{L,S,E\}} \sum_{j=1}^{i} \beta_r^{i-j} \delta_j.$$

Proof. Since the action-role sequence is shared, $\boldsymbol{\mu}(a^{(t_i)})$ is the same for both trajectories. Also, $\boldsymbol{\mu}(a^{(t_i)}) \in [0,1]^d$ elementwise because it is produced by a sigmoid gate. Therefore,

$$\begin{aligned} &\left\| \boldsymbol{\mu}(a^{(t_i)}) \odot \mathbf{e}_{tl}^{(t_i)} - \boldsymbol{\mu}(a^{(t_i)}) \odot \mathbf{e}_{tl'}^{(t_i)} \right\|_2 \\ &\le \left\| \mathbf{e}_{tl}^{(t_i)} - \mathbf{e}_{tl'}^{(t_i)} \right\|_2 = \delta_i. \end{aligned}$$

Let $\Delta_i^{(r)} = \mathbf{m}_u^{(t_i,r)} - \mathbf{m}_{u'}^{(t_i,r)}$. *The trace recursion gives*

$$\Delta_i^{(r)} = A_r \Delta_{i-1}^{(r)} + B\left(\mathbf{b}^{(t_i)} - \mathbf{b}'^{(t_i)}\right).$$

Unrolling the recursion and using equal initial states gives

$$\Delta_i^{(r)} = \sum_{j=1}^{i} A_r^{i-j} B\left(\mathbf{b}^{(t_j)} - \mathbf{b}'^{(t_j)}\right).$$

Taking norms,

$$\begin{aligned}
&\left\|\Delta_i^{(r)}\right\|_2 \\
&\leq \sum_{j=1}^{i} \left\|A_r^{i-j}\right\|_2 \|B\|_2 \left\|\mathbf{b}^{(t_j)} - \mathbf{b}'^{(t_j)}\right\|_2 \\
&\leq \|B\|_2 \sum_{j=1}^{i} \beta_r^{i-j} \delta_j.
\end{aligned}$$

Applying the Lipschitz property of Φ *to the concatenated trace-state difference yields the bound on* $\mathbf{h}_{b_u}^{(t_i)}$. *Applying the Lipschitz property of* $F_{\mathbf{g}_q}$ *yields the bound on* $\mathbf{z}_{b_u}^{(t_i)}$. □

Interpretation. This formalizes embedding-induced item-side abstraction. If two target loci are close in the embedding space, then their induced user states remain close, with a lag-weighted bound controlled by the trace kernels. Hence, `PerTIDE` does not require manually grouping books, news articles, films, products, or dialogue responses into a fixed taxonomy. Similarity is carried by the target-locus embedding geometry and then filtered by action-conditioned trace dynamics.

Schema-Trajectory Metric. For transfer analysis, let $x = (S_D(\tau), \mathbf{g}_q)$ denote a schema-normalized trajectory and command seed. For two equal-length schema trajectories x and x', define

$$d_{\mathrm{sch}}(x, x') = \sum_{i=1}^{T} \omega_i \left[\mathbf{1}\left(a^{(t_i)} \neq a'^{(t_i)}\right) + \left\|\mathbf{e}_{tl}^{(t_i)} - \mathbf{e}_{tl'}^{(t_i)}\right\|_2\right] + \lambda_g \|\mathbf{g}_q - \mathbf{g}'_q\|_2,$$

where $\omega_i \geq 0$ are timestep weights and $\lambda_g \geq 0$ controls command-seed sensitivity. A natural choice is to set ω_i proportional to the induced trace influence, for example $\omega_i = \sum_{r\in\{L,S,E\}} \beta_r^{T-i}$ for stable trace radii β_r. This metric compares trajectories after schema rewriting, not in their native action vocabulary.

Proposition B.3 (Schema-Level Transfer Bound)**.** *Let* P_S *and* P_T *be source and target distributions over schema-normalized trajectory-command pairs* $x = (S_D(\tau), \mathbf{g}_q)$. *Let*

$$R_D(\theta) = \mathbb{E}_{(x,y)\sim D}\left[\ell(f_\theta(x), y)\right]$$

be the expected risk of `PerTIDE` *under a bounded loss* ℓ. *For fixed* θ, *define the conditional loss functions*

$$\ell_\theta^S(x) = \mathbb{E}_{y\sim P_S(y|x)}\left[\ell(f_\theta(x), y)\right], \qquad \ell_\theta^T(x) = \mathbb{E}_{y\sim P_T(y|x)}\left[\ell(f_\theta(x), y)\right].$$

Assume ℓ_θ^S *is L-Lipschitz with respect to* d_{sch}. *Define the labeling or command-semantics discrepancy*

$$\Delta_{S,T}(\theta) = \mathbb{E}_{x\sim P_T}\left[\left|\ell_\theta^T(x) - \ell_\theta^S(x)\right|\right].$$

Then

$$R_T(\theta) \leq R_S(\theta) + L\, W_1(P_S, P_T) + \Delta_{S,T}(\theta),$$

where $W_1(P_S, P_T)$ *is the Wasserstein-1 distance between source and target schema-trajectory distributions under* d_{sch}.

Proof. We decompose the target risk as

$$R_T(\theta) = \mathbb{E}_{x\sim P_T}\left[\ell_\theta^T(x)\right].$$

By the definition of $\Delta_{S,T}(\theta)$,

$$\mathbb{E}_{x\sim P_T}\left[\ell_\theta^T(x)\right] \le \mathbb{E}_{x\sim P_T}\left[\ell_\theta^S(x)\right] + \Delta_{S,T}(\theta).$$

Since ℓ_θ^S is L-Lipschitz under d_{sch}, Kantorovich-Rubinstein duality gives

$$\mathbb{E}_{x\sim P_T}\left[\ell_\theta^S(x)\right] - \mathbb{E}_{x\sim P_S}\left[\ell_\theta^S(x)\right] \le L\, W_1(P_S, P_T).$$

Combining the two inequalities yields

$$R_T(\theta) \le R_S(\theta) + L\, W_1(P_S, P_T) + \Delta_{S,T}(\theta).$$

This proves the claim. □

Finite-Sample Form. Let $\widehat{R}_S(\theta)$ be the empirical source risk, and suppose that with probability at least $1-\delta$,

$$R_S(\theta) \le \widehat{R}_S(\theta) + \mathcal{G}_S(n,\delta),$$

where $\mathcal{G}_S(n,\delta)$ is a source generalization term depending on sample size n, confidence δ, and the hypothesis class. Then, with the same probability,

$$R_T(\theta) \le \widehat{R}_S(\theta) + \mathcal{G}_S(n,\delta) + L\, W_1(P_S, P_T) + \Delta_{S,T}(\theta).$$

Interpretation. This gives a precise form of schema-level generalizability. Transfer is favorable when (i) the source model has low empirical risk, (ii) the source and target schema-trajectory distributions are close after action-on-item rewriting, and (iii) the command semantics or target labels are compatible. This is a restricted claim. It does not assert universal transfer to arbitrary domains. It states that `PerTIDE` can reuse its event schema and evidence-flow mechanism across eventizable personalization histories.

Pooled Schema Training. The previous results are pointwise or distributional. We now formalize the training-level consequence of the shared action-on-item schema. Suppose we have M eventizable personalization domains. Domain m provides native trajectories that are rewritten into schema-normalized examples

$$x = (S_{D_m}(\tau), \mathbf{g}_q),$$

where S_{D_m} maps native action–item events to action roles and target-locus embeddings. Let P_m denote the induced distribution over schema-normalized trajectory–command pairs, and let $\pi_m = n_m/N$ be the empirical mixture weight, with $N = \sum_{m=1}^M n_m$. The pooled schema distribution is

$$P_\pi = \sum_{m=1}^M \pi_m P_m.$$

Training `PerTIDE` on the common substrate corresponds to empirical risk minimization over the pooled schema sample:

$$\widehat{R}_\pi(\theta) = \sum_{m=1}^M \pi_m \widehat{R}_m(\theta) = \frac{1}{N}\sum_{m=1}^M \sum_{i=1}^{n_m} \ell(f_\theta(x_{m,i}), y_{m,i}).$$

Let $\widehat{\theta}_\pi \in \arg\min_{\theta\in\Theta} \widehat{R}_\pi(\theta)$.

Proposition B.4 (Common-Substrate Training Bound)**.** *Assume the loss $\ell(f_\theta(x), y)$ is bounded in $[0,1]$. Let the loss-composed hypothesis class*

$$\mathcal{F} = \{x \mapsto \ell(f_\theta(x), y) : \theta \in \Theta\}$$

have empirical Rademacher complexity $\mathfrak{R}_N(\mathcal{F})$ on the pooled schema sample. For a target domain t, assume the source-mixture loss $\ell_\theta^\pi(x)$ is L-Lipschitz with respect to the schema-trajectory metric d_{sch}. Define the command-semantics discrepancy

$$\Delta_{t,\pi}(\theta) = \mathbb{E}_{x\sim P_t}\left[\left|\ell_\theta^t(x) - \ell_\theta^\pi(x)\right|\right].$$

Then, with probability at least $1-\delta$ over the pooled training sample,

$$R_t(\widehat{\theta}_\pi) \le \widehat{R}_\pi(\widehat{\theta}_\pi) + 2\mathfrak{R}_N(\mathcal{F}) + 3\sqrt{\frac{\log(2/\delta)}{2N}} + L\, W_1(P_t, P_\pi) + \Delta_{t,\pi}(\widehat{\theta}_\pi).$$

Equivalently, since $\widehat{\theta}_\pi$ minimizes the pooled empirical risk,

$$R_t(\widehat{\theta}_\pi) \le \widehat{R}_\pi(\theta^\star) + 2\mathfrak{R}_N(\mathcal{F}) + 3\sqrt{\frac{\log(2/\delta)}{2N}} + L\,W_1(P_t, P_\pi) + \Delta_{t,\pi}(\widehat{\theta}_\pi),$$

for any comparator $\theta^\star \in \Theta$.

Proof. By standard uniform convergence for bounded losses, with probability at least $1-\delta$,

$$R_\pi(\theta) \le \widehat{R}_\pi(\theta) + 2\mathfrak{R}_N(\mathcal{F}) + 3\sqrt{\frac{\log(2/\delta)}{2N}}$$

holds uniformly for all $\theta \in \Theta$. Applying this to $\widehat{\theta}_\pi$ gives

$$R_\pi(\widehat{\theta}_\pi) \le \widehat{R}_\pi(\widehat{\theta}_\pi) + 2\mathfrak{R}_N(\mathcal{F}) + 3\sqrt{\frac{\log(2/\delta)}{2N}}.$$

From the schema-level transfer bound in Proposition B.3, for any fixed θ,

$$R_t(\theta) \le R_\pi(\theta) + L\,W_1(P_t, P_\pi) + \Delta_{t,\pi}(\theta).$$

Setting $\theta = \widehat{\theta}_\pi$ and combining the two inequalities yields the first bound. The second bound follows from empirical optimality of $\widehat{\theta}_\pi$, since

$$\widehat{R}_\pi(\widehat{\theta}_\pi) \le \widehat{R}_\pi(\theta^\star)$$

for every $\theta^\star \in \Theta$. □

Interpretation. This formalizes the common-substrate training claim. After schema rewriting, examples from different domains contribute to one empirical objective over action roles, target-locus embeddings, and command seeds. The estimation term depends on the pooled sample size N, not on the sample size of a single native domain. The target-domain penalty is not hidden: transfer depends on the schema distance $W_1(P_t, P_\pi)$ and the command-semantics discrepancy $\Delta_{t,\pi}$. Thus, pooling is beneficial when domains become close after schema normalization, and harmful transfer is possible when the schema distributions or command meanings are mismatched.

Corollary B.5 (Sample-Efficiency From Schema-Compatible Pooling)**.** *If all domains are schema-compatible in the sense that $P_t = P_\pi$ and $\Delta_{t,\pi}(\theta) = 0$, then Proposition E.4 reduces to*

$$R_t(\widehat{\theta}_\pi) \le \widehat{R}_\pi(\widehat{\theta}_\pi) + 2\mathfrak{R}_N(\mathcal{F}) + 3\sqrt{\frac{\log(2/\delta)}{2N}}.$$

In contrast, training a separate domain-specific encoder on only n_t examples gives the analogous estimation term

$$2\mathfrak{R}_{n_t}(\mathcal{F}) + 3\sqrt{\frac{\log(2/\delta)}{2n_t}}.$$

Since $N = \sum_m n_m \ge n_t$, the pooled schema learner has a smaller estimation term whenever the shared schema makes the domains compatible.

Connection to the Main Story. This result gives a training-level interpretation of the action-on-item interface. The common bin is not a raw mixture of unrelated logs. It is a schema-normalized pool in which native action–item pairs are rewritten as positive, negative, or command-like evidence over target-locus embeddings. Once this rewriting is done, the pooled objective trains a single evidence-flow mechanism across domains. The theorem also states the boundary of the claim: `PerTIDE` does not guarantee universal transfer. Its benefit depends on whether new trajectories are eventizable, whether target-locus embeddings make item-side evidence comparable, and whether the command semantics remain compatible.

Connection to MTSH. The propositions above explain why the action-on-item abstraction is useful before any particular trace design is chosen. Proposition B.1 shows that native action labels need not match once they are mapped to action roles. Proposition B.2 shows that item-side similarity can be induced by target-locus embeddings rather than by a hand-built item taxonomy. Proposition B.3

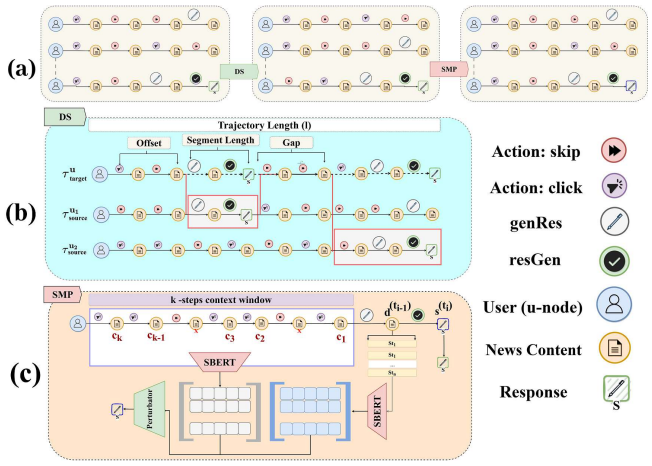


Figure 3: **UIG Augmentation Pipeline:** Chatterjee et al. [2025] proposed framework – (a) *Pipeline overview depicting the two-step augmentation*, (b) *Double Shuffling (DS) to ensure cross-trajectory augmentation and induce diffusion*, (c) *Stochastic Markovian Perturbation (SMP) to smoothen the response-nodes and modulate random diffusion incorporated in DS stage*.

shows that cross-domain reuse can be bounded in terms of schema-trajectory distance and command-semantics discrepancy. MTSH then specifies how evidence in this schema should persist: some evidence accumulates as persistent preference mass, some remains recency-sensitive, and some survives as a finite event or command imprint. Thus, `PerTIDE` is not merely a multi-timescale encoder. It is an implementation of action-conditioned preference flow over a reusable schema for eventizable user histories.

C UIG Augmentation

After inserting surrogate response-nodes via *genRes* and *resGen* interactions in the base trajectories, we apply `PerAugy`-style Chatterjee et al. [2025] augmentation to (i) increase trajectory diversity while retaining realistic temporal structure, and (ii) reduce local incoherence introduced by surrogate responses (r-nodes). We apply augmentation to u-tier trajectories τ^{u_j} before training, yielding an expanded trajectory pool used to instantiate $\mathcal{T}^{\text{PENS}}$ (and its perturbed variants).

Double Shuffling. Given m sampled training trajectories $\mathcal{T}^{m}_{\text{sample}} \subseteq \mathcal{T}_{\text{train}}$, we select one target trajectory $\tau^{u_j}_{\text{target}}$ and treat the remaining $m-1$ trajectories as sources. We construct a synthetic trajectory by (i) preserving an early prefix of the target determined by an offset O, and (ii) substituting later contiguous segments using source trajectory snippets. To preserve local temporal realism, we enforce a gap hyperparameter G such that after each substitution, G consecutive steps from the target are kept intact.

Stochastic Markovian Perturbation. While shuffling increases structural diversity, it can introduce surrogate r-nodes whose content is poorly aligned with the target user's recent context, reducing temporal coherence. We therefore apply a local, context-aware perturbation that replaces a substituted r-node with a sentence that better matches the immediate history of the *target* trajectory. Consider a substituted summary node $r^{(t_i)}$ in $\tau^{u_j}_{\text{DS}}$. Let $\tau^{u_j}_{c_k}$ denote a backward context window of the previous k nodes $\{c_q\}_{q=1}^{k}$ (most recent first), and let $d^{(t_{i-1})}$ be the preceding d-node. We segment $d^{(t_{i-1})}$ into candidate sentences $\{st_p\}_{p=1}^{n_{d^{(t_{i-1})}}}$. Using SBERT embeddings, we compute an influence-weighted distance between each candidate sentence and the context window, with temporal relevance enforced by an exponential decay weight $w_q = \exp(-\lambda \cdot \text{pos}(c_q))$ and $\text{pos}(c_1) = 0$.

Algorithm 2 DOUBLESHUFFLING: Cross-Trajectory Segment Substitution

Require: Target trajectory τ^{u_j}; source trajectories $\{\tau^{u_i}\}_{i=1}^{m-1}$; offset distribution $\mathcal{P}_O$; gap size G
Ensure: Synthetic trajectory $\tilde{\tau}^{u_j}$

Sample offset $O \sim \mathcal{P}_O$
$\tilde{\tau}^{u_j} \leftarrow \tau^{u_j}[1{:}O]$ (preserve prefix)
$t \leftarrow O$
for each source trajectory τ^{u_i} **do**
  $\tau_{\text{seg}} \leftarrow \text{SampleSegment}(\tau^{u_i})$
  $\tilde{\tau}^{u_j} \leftarrow \tilde{\tau}^{u_j} \oplus \tau_{\text{seg}}$ (substitute)
  $\tilde{\tau}^{u_j} \leftarrow \tilde{\tau}^{u_j} \oplus \tau^{u_j}[t{+}1{:}t{+}G]$ (protected gap)
  $t \leftarrow t + |\tau_{\text{seg}}| + G$
end for
$\tilde{\tau}^{u_j} \leftarrow \tilde{\tau}^{u_j} \oplus \text{Suffix}(\tau^{u_j}, t)$ **return** $\tilde{\tau}^{u_j}$

Algorithm 3 SUMMARYPERTURBATION: Local Summary Rewriting

Require: Trajectory τ^{u_j}; window size k; decay λ; perturbation rate p_{SMP}
Ensure: Perturbed trajectory $\tilde{\tau}^{u_j}$

$\tilde{\tau}^{u_j} \leftarrow \tau^{u_j}$
for each substituted summary node $s^{(t_i)} \in \tilde{\tau}^{u_j}$ **do**
  Sample $z \sim \text{Bernoulli}(p_{\text{SMP}})$
  if $z = 1$ **then**
    $\mathcal{C} \leftarrow \text{ContextWindow}(\tilde{\tau}^{u_j}, t_i, k)$
    $\hat{s}^{(t_i)} \leftarrow \text{RewriteSummary}(s^{(t_i)}, \mathcal{C}, \lambda)$ (Eq. (3))
    Replace $s^{(t_i)} \leftarrow \hat{s}^{(t_i)}$
  end if
end forreturn $\tilde{\tau}^{u_j}$

Let $\sigma \in \mathbb{R}^{n_{d^{(t_{i-1})}} \times k}$ be the distance matrix with entries $\sigma_{(p,q)} = \sigma(\mathbf{e}_{st_p}, \mathbf{e}_{c_q})$, where $\sigma(\cdot,\cdot)$ is RMSD. We select:

$$\hat{r}^{(t_i)} = \arg\min \sum_{q=1}^{k} \exp\big(-\lambda \cdot \text{pos}(c_q)\big)\, \sigma(\mathbf{e}_{st_p}, \mathbf{e}_{c_q}). \tag{3}$$

Equivalently, per-sentence scores can be written as $\sigma\,\mathbf{w}$ where $\mathbf{w}_q = \exp(-\lambda \cdot \text{pos}(c_q))$.

Thus, we obtain $\tau_{\text{P}}^{\hat{u}}$ without leakage of summaries from any testbed. For PENS, $\tau_{\text{P}}^{\hat{u}}$ instantiates $\tau^{\text{PENS-D}}$, and for OpenAI-Reddit, the same procedure yields τ^{OAI} for training. The perturbed s-nodes provide additional, coherence-aware supervision for personalized summarization tasks.

D PerTIDE Architectural Details

D.1 Discrete-Time SSM Stability

We first state the stability condition used by the discrete state-space traces in PerTIDE. Consider the linear recursion

$$\mathbf{h}_t = \mathbf{A}\mathbf{h}_{t-1} + \mathbf{B}\mathbf{x}_t,$$

with homogeneous dynamics

$$\mathbf{h}_t = \mathbf{A}\mathbf{h}_{t-1}.$$

The stability of the homogeneous part determines whether past perturbations decay or persist without bound.

Proposition D.1 (Discrete-time stability by spectral radius)**.** *Let* $\mathbf{A} \in \mathbb{C}^{d\times d}$*. If*

$$\rho(\mathbf{A}) < 1,$$

where $\rho(\mathbf{A}) = \max_i |\lambda_i(\mathbf{A})|$ *is the spectral radius, then the homogeneous recursion*

$$\mathbf{h}_t = \mathbf{A}\mathbf{h}_{t-1}$$

is asymptotically stable, i.e.,

$$\lim_{t\to\infty} \mathbf{A}^t \mathbf{h}_0 = \mathbf{0}$$

for every initial state $\mathbf{h}_0$.

Proof. Since $\rho(\mathbf{A}) < 1$, all eigenvalues of $\mathbf{A}$ lie strictly inside the unit disk. By the standard spectral-radius characterization of matrix powers, this implies

$$\lim_{t\to\infty} \mathbf{A}^t = \mathbf{0}.$$

Therefore,

$$\lim_{t\to\infty} \mathbf{h}_t = \lim_{t\to\infty} \mathbf{A}^t \mathbf{h}_0 = \mathbf{0},$$

which proves asymptotic stability. □

Corollary D.2 (Modal stability for diagonalizable transitions)**.** *Suppose* $\mathbf{A}$ *is diagonalizable as*

$$\mathbf{A} = \mathbf{Q}\,\mathrm{diag}(a_1, \ldots, a_d)\mathbf{Q}^{-1}.$$

Then the homogeneous dynamics are asymptotically stable if

$$|a_i| < 1 \qquad \textit{for all } i = 1, \ldots, d.$$

Moreover, the contribution of mode i *after lag* Δ *decays as*

$$|a_i|^{\Delta} = \exp\left(-\frac{\Delta}{\tau_i}\right), \qquad \tau_i = -\frac{1}{\log|a_i|}.$$

Proof. For a diagonalizable transition,

$$\mathbf{A}^{\Delta} = \mathbf{Q}\,\mathrm{diag}(a_1^{\Delta}, \ldots, a_d^{\Delta})\mathbf{Q}^{-1}.$$

Thus $\mathbf{A}^{\Delta} \to \mathbf{0}$ whenever $|a_i| < 1$ for every mode. The effective decay relation follows by writing

$$|a_i|^{\Delta} = \exp(\Delta \log|a_i|) = \exp\left(-\frac{\Delta}{\tau_i}\right),$$

which gives

$$\tau_i = -\frac{1}{\log|a_i|}.$$

□

Connection to Continuous-Time Eigenvalues. If the discrete transition is induced by a continuous-time generator $\mathbf{\Lambda}$ with eigenvalues $\lambda_i \in \mathbb{C}$, then continuous-time stability requires

$$\mathrm{Re}(\lambda_i) < 0.$$

Under exponential discretization,

$$a_i = \exp(\lambda_i \Delta t),$$

so

$$|a_i| = \exp(\mathrm{Re}(\lambda_i)\Delta t).$$

Hence,

$$|a_i| < 1 \quad \Longleftrightarrow \quad \mathrm{Re}(\lambda_i) < 0.$$

Under Euler-style discretization,

$$a_i = 1 + \eta\lambda_i,$$

and stability requires

$$|1 + \eta\lambda_i| < 1.$$

Thus Euler-style maps impose an explicit step-size constraint coupling η and λ_i. In `PerTIDE`, the trace-specific spectral maps and step sizes are chosen so that the induced discrete eigenvalues lie inside the unit disk. This preserves stability while allowing the three traces to express different effective retention ranges.

D.2 SSM Hyperparameter Choices

We next give sufficient rules for selecting the short-memory horizon and the event-step size. These rules are not claimed to be necessary; they provide simple conservative choices that make the intended timescale behavior explicit.

Notation. Let $\varepsilon \in (0, 1/2)$ be a small tolerance, let

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

be the sigmoid function, and let $\tau > 0$ denote a desired exponential decay time constant.

Short-Memory Horizon Selection

The short-memory trace uses a retention factor of the form

$$\sigma(-k\lambda_S), \qquad \lambda_S > 0.$$

We choose the horizon k so that retention at lag k is at most ε:

$$\sigma(-k\lambda_S) \le \varepsilon. \tag{4}$$

Lemma D.3 (Sufficient short-memory horizon)**.** *Let $\lambda_S > 0$ and $\varepsilon \in (0, 1/2)$. If*

$$k \ge \frac{1}{\lambda_S} \log\left(\frac{1-\varepsilon}{\varepsilon}\right), \tag{5}$$

then

$$\sigma(-k\lambda_S) \le \varepsilon.$$

Proof. Using

$$\sigma(-k\lambda_S) = \frac{1}{1 + e^{k\lambda_S}},$$

the condition $\sigma(-k\lambda_S) \le \varepsilon$ is equivalent to

$$\frac{1}{1 + e^{k\lambda_S}} \le \varepsilon.$$

Since all terms are positive,

$$1 \le \varepsilon(1 + e^{k\lambda_S})$$

and therefore

$$1 - \varepsilon \le \varepsilon e^{k\lambda_S}.$$

Equivalently,

$$e^{k\lambda_S} \ge \frac{1-\varepsilon}{\varepsilon}.$$

Taking logarithms gives

$$k\lambda_S \ge \log\left(\frac{1-\varepsilon}{\varepsilon}\right),$$

which proves the claim. □

Corollary D.4 (Rapid suppression beyond the short-memory horizon)**.** *Let $a_S = \sigma(-\kappa\lambda_S) \in (0, 1)$ be a short-trace modal base. If $|u_{t-\ell}| \le R$ for all ℓ, then the total contribution from lags at least W is bounded by*

$$\left| \sum_{\ell=W}^{\infty} a_S^{\ell} u_{t-\ell} \right| \le R \frac{a_S^W}{1 - a_S}.$$

Thus, when a_S is small, evidence outside a short window is geometrically suppressed.

Proof. By the triangle inequality,

$$\left|\sum_{\ell=W}^{\infty} a_S^{\ell} u_{t-\ell}\right| \leq \sum_{\ell=W}^{\infty} a_S^{\ell} |u_{t-\ell}| \leq R \sum_{\ell=W}^{\infty} a_S^{\ell}.$$

Since $a_S \in (0,1)$,

$$\sum_{\ell=W}^{\infty} a_S^{\ell} = \frac{a_S^W}{1-a_S}.$$

This proves the bound. □

Event-Kernel Step Size Selection

For the episodic trace, let $\lambda_E < 0$ denote a stable continuous-time decay rate. Under Euler-style discretization with step size $\eta > 0$, the per-step base is

$$a_E = 1 + \eta\lambda_E.$$

The event influence at lag $\Delta = t - k$ is therefore

$$w_{t,k}^{(E)} = (1 + \eta\lambda_E)^{t-k}. \tag{6}$$

We choose η so that this discrete kernel exactly matches the target exponential decay

$$\exp\left(-\frac{t-k}{\tau}\right).$$

Lemma D.5 (Exact event-kernel matching)**.** *Let $\lambda_E < 0$ and $\tau > 0$. Define*

$$\eta = \frac{e^{-1/\tau} - 1}{\lambda_E}. \tag{7}$$

Then

$$1 + \eta\lambda_E = e^{-1/\tau},$$

and for every lag $\Delta = t - k$,

$$(1 + \eta\lambda_E)^{t-k} = \exp\left(-\frac{t-k}{\tau}\right).$$

Moreover, $\eta > 0$ and

$$0 < 1 + \eta\lambda_E < 1,$$

so the induced discrete event mode is stable.

Proof. Substituting Eq. (7) gives

$$1 + \eta\lambda_E = 1 + \left(\frac{e^{-1/\tau} - 1}{\lambda_E}\right)\lambda_E = e^{-1/\tau}.$$

Raising both sides to the power $t - k$ yields

$$(1 + \eta\lambda_E)^{t-k} = e^{-(t-k)/\tau}.$$

Since $\tau > 0$, we have $e^{-1/\tau} \in (0,1)$. Therefore $e^{-1/\tau} - 1 < 0$. Because $\lambda_E < 0$, Eq. (7) implies $\eta > 0$. Finally,

$$1 + \eta\lambda_E = e^{-1/\tau} \in (0,1),$$

which proves stability. □

D.3 Kernel-Level Interpretation of the Three `PerTIDE` Traces

We now formalize the kernel-level role of the three traces. The claim is architectural: the trace-specific spectral maps induce distinct retention profiles. This does not imply semantic disentanglement of learned dimensions by construction. Functional non-redundancy is tested empirically through trace-subset and temporal-regime ablations.

General tri-trace recursion. For each trace $r \in \{L, S, E\}$, `PerTIDE` updates a memory state as

$$m_t^{(r)} = A_r m_{t-1}^{(r)} + B b_t, \tag{8}$$

where b_t is the action-gated event cell, B is the input operator, and A_r is the trace-specific transition matrix. Assume that

$$A_r = Q_r \operatorname{diag}(a_1^{(r)}, \ldots, a_d^{(r)}) Q_r^{-1}.$$

The three `PerTIDE` spectral maps induce

$$a_i^{(L)} = \exp(\delta \lambda_{L,i}), \qquad a_i^{(S)} = \sigma(-\kappa \lambda_{S,i}), \qquad a_i^{(E)} = 1 + \eta \lambda_{E,i}.$$

For clarity, we state the results for real stable modes satisfying

$$0 \le a_i^{(r)} < 1.$$

For complex modes, the same statements hold for the decay envelope after replacing $a_i^{(r)}$ by $|a_i^{(r)}|$.

Proposition D.6 (Mode-wise lag kernels induced by the trace recursion)**.** *For each trace $r \in \{L, S, E\}$ and mode i, the contribution of an input at lag $\ell = t - s$ to the modal state is weighted by*

$$k_i^{(r)}(\ell) = \left(a_i^{(r)}\right)^{\ell}, \qquad \ell \ge 0.$$

Thus each trace implements an explicit geometric lag kernel in its eigenbasis.

Proof. Move Eq. (8) to the eigenbasis of trace r:

$$\tilde{m}_t^{(r)} = Q_r^{-1} m_t^{(r)}, \qquad u_t^{(r)} = Q_r^{-1} B b_t.$$

Then

$$\tilde{m}_t^{(r)} = \operatorname{diag}(a_1^{(r)}, \ldots, a_d^{(r)}) \tilde{m}_{t-1}^{(r)} + u_t^{(r)}.$$

Therefore, each mode evolves independently:

$$\tilde{m}_{t,i}^{(r)} = a_i^{(r)} \tilde{m}_{t-1,i}^{(r)} + u_{t,i}^{(r)}.$$

Unrolling gives

$$\tilde{m}_{t,i}^{(r)} = \left(a_i^{(r)}\right)^t \tilde{m}_{0,i}^{(r)} + \sum_{s=1}^{t} \left(a_i^{(r)}\right)^{t-s} u_{s,i}^{(r)}.$$

Hence an input at lag $\ell = t - s$ is weighted by

$$\left(a_i^{(r)}\right)^{\ell}.$$

This proves the claim. □

Proposition D.7 (Effective memory length of a stable modal kernel)**.** *Let $a \in (0, 1)$ and define the unnormalized lag kernel*

$$k_a(\ell) = a^{\ell}, \qquad \ell = 0, 1, 2, \ldots.$$

Its normalized version is

$$\bar{k}_a(\ell) = (1 - a) a^{\ell}.$$

Then the expected lag is

$$\mathbb{E}_{\bar{k}_a}[\ell] = \frac{a}{1 - a}.$$

The half-life is

$$h(a) = \frac{\log(1/2)}{\log a}.$$

Both quantities increase as a approaches one.

Proof. Since $\bar{k}_a$ is the probability mass function of a geometric distribution over nonnegative lags,

$$\sum_{\ell=0}^{\infty}(1-a)a^{\ell}=1.$$

Its expected lag is

$$\mathbb{E}_{\bar{k}_a}[\ell]=(1-a)\sum_{\ell=0}^{\infty}\ell a^{\ell}=(1-a)\frac{a}{(1-a)^2}=\frac{a}{1-a}.$$

The half-life is the lag h satisfying

$$a^h=\frac{1}{2}.$$

Taking logarithms gives

$$h(a)=\frac{\log(1/2)}{\log a}.$$

For $a\in(0,1)$, increasing a slows the geometric decay, and therefore increases both the expected lag and the half-life. □

Corollary D.8 (Trace-specific retention profiles)**.** *The three `PerTIDE` spectral maps induce the following modal kernels:*

$$k_i^{(L)}(\ell)=\left(\exp(\delta\lambda_{L,i})\right)^{\ell},$$
$$k_i^{(S)}(\ell)=\left(\sigma(-\kappa\lambda_{S,i})\right)^{\ell},$$
$$k_i^{(E)}(\ell)=\left(1+\eta\lambda_{E,i}\right)^{\ell}.$$

Consequently, long-term, short-term, and episodic traces differ by the base of their geometric lag kernels.

Proof. The result follows directly by substituting the three trace-specific spectral maps into Proposition D.6. □

Long-Term Trace: Persistent Preference Accumulation

Proposition D.9 (Slow modes accumulate repeated evidence)**.** *Consider a long-term modal direction i with*

$$a_i^{(L)}=\exp(\delta\lambda_{L,i})\in(0,1).$$

Suppose repeated evidence enters this mode as

$$u_{s,i}^{(L)}=\mu_i>0,\qquad s=1,\dots,t,$$

and let $\tilde{m}_{0,i}^{(L)}=0$. Then

$$\tilde{m}_{t,i}^{(L)}=\mu_i\frac{1-\left(a_i^{(L)}\right)^t}{1-a_i^{(L)}},$$

and

$$\lim_{t\to\infty}\tilde{m}_{t,i}^{(L)}=\frac{\mu_i}{1-a_i^{(L)}}.$$

Hence, when $a_i^{(L)}$ is close to one, repeated evidence accumulates into a large persistent state.

Proof. By Proposition D.6,

$$\tilde{m}_{t,i}^{(L)}=\sum_{s=1}^{t}\left(a_i^{(L)}\right)^{t-s}\mu_i.$$

Let $\ell=t-s$. Then

$$\tilde{m}_{t,i}^{(L)}=\mu_i\sum_{\ell=0}^{t-1}\left(a_i^{(L)}\right)^{\ell}.$$

Using the finite geometric sum,

$$\sum_{\ell=0}^{t-1} \left(a_i^{(L)}\right)^{\ell} = \frac{1 - \left(a_i^{(L)}\right)^{t}}{1 - a_i^{(L)}}.$$

Since $a_i^{(L)} \in (0,1)$,

$$\left(a_i^{(L)}\right)^{t} \to 0,$$

which yields the stated limit. □

This formalizes the role of the long-term trace. If a user repeatedly produces evidence in a stable preference direction, such as repeated climate-policy interactions, the long-term kernel retains and accumulates this evidence rather than rapidly discarding it.

Short-Term Trace: Recent Burst Tracking

Proposition D.10 (Small short-trace bases emphasize recent evidence)**.** *Consider a short-term modal direction j with*

$$a_j^{(S)} = \sigma(-\kappa\lambda_{S,j}) \in (0,1).$$

The modal state can be written as

$$\tilde{m}_{t,j}^{(S)} = u_{t,j}^{(S)} + a_j^{(S)} u_{t-1,j}^{(S)} + \left(a_j^{(S)}\right)^2 u_{t-2,j}^{(S)} + \cdots .$$

If $a_j^{(S)}$ is small, then the newest inputs dominate and older inputs are geometrically suppressed.

Proof. The expansion follows from Proposition D.6. The relative weight assigned to an input ℓ steps in the past is

$$\left(a_j^{(S)}\right)^{\ell}.$$

For $a_j^{(S)} \in (0,1)$, this quantity decays geometrically in ℓ. If $a_j^{(S)}$ is small, the decay is rapid, so recent inputs dominate the modal state. □

This formalizes the role of the short-term trace. A recent burst of sports-highlight interactions, for example, enters the short trace with high weight at the current timestep, while older evidence is rapidly suppressed.

Episodic Trace: Finite-Window Event Imprint

Proposition D.11 (Finite-window events leave controlled decaying imprints)**.** *Consider an episodic modal direction k with*

$$a_k^{(E)} = 1 + \eta\lambda_{E,k} \in (0,1).$$

Suppose an event is active on the finite window $t_0 \le s \le t_1$:

$$u_{s,k}^{(E)} = \begin{cases} \beta_k, & t_0 \le s \le t_1, \\ 0, & \textit{otherwise}, \end{cases} \qquad \beta_k > 0.$$

Let $W = t_1 - t_0 + 1$. For $t > t_1$, the remaining post-event imprint is

$$\tilde{m}_{t,k}^{(E)} = \beta_k \left(a_k^{(E)}\right)^{t-t_1} \frac{1 - \left(a_k^{(E)}\right)^{W}}{1 - a_k^{(E)}}.$$

Thus the event persists after the active window closes, but its imprint decays at rate $a_k^{(E)}$.

Proof. For $t > t_1$, only the event-window inputs contribute:

$$\tilde{m}_{t,k}^{(E)} = \sum_{s=t_0}^{t_1} \left(a_k^{(E)}\right)^{t-s} \beta_k.$$

Factor out the decay from the end of the event window:

$$\tilde{m}_{t,k}^{(E)} = \beta_k \left(a_k^{(E)}\right)^{t-t_1} \sum_{q=0}^{W-1} \left(a_k^{(E)}\right)^q.$$

Using the finite geometric sum gives

$$\sum_{q=0}^{W-1} \left(a_k^{(E)}\right)^q = \frac{1 - \left(a_k^{(E)}\right)^W}{1 - a_k^{(E)}}.$$

This proves the expression. □

Corollary D.12 (Event time constant under exact kernel matching)**.** *If the event-step size is chosen as in Lemma D.5, then*

$$a_k^{(E)} = 1 + \eta\lambda_{E,k} = e^{-1/\tau_k}.$$

Therefore,

$$k_k^{(E)}(\ell) = e^{-\ell/\tau_k},$$

and the post-event imprint becomes

$$\tilde{m}_{t,k}^{(E)} = \beta_k e^{-(t-t_1)/\tau_k} \frac{1 - e^{-W/\tau_k}}{1 - e^{-1/\tau_k}}.$$

Thus τ_k directly controls how long the finite-window event remains active after the event closes.

Proof. The result follows by substituting

$$a_k^{(E)} = e^{-1/\tau_k}$$

into Proposition D.11. □

This formalizes the role of the episodic trace. A rare debate-night interaction, tournament-final event, or command-specific burst does not vanish immediately. It remains as a finite-horizon imprint whose post-event persistence is controlled by the event kernel.

D.4 Approximation Capacity Beyond the L/S/E Interpretation

The labels long-term, short-term, and episodic describe the intended inductive bias. They do not restrict the traces to only three human-named semantics. Mixtures of stable geometric kernels can also approximate broader temporal patterns used in classical time-series modeling.

Proposition D.13 (Mixtures of trace kernels can approximate broader temporal filters)**.** *Let*

$$K_M(\ell) = \sum_{m=1}^{M} c_m a_m^\ell, \qquad 0 < a_m < 1,$$

be a finite mixture of stable geometric kernels. Then K_M can represent or approximate a range of stable temporal filters, including multi-rate decay, delayed persistence, smooth trend-like retention over finite horizons, and damped seasonal components. In particular:

1. *If several a_m are close to one, the mixture forms a slow low-pass filter that can preserve trend-like evidence over a finite horizon.*
2. *If some a_m are small and others are close to one, the mixture separates recent fluctuations from persistent background mass.*

3. *If complex-conjugate stable modes are allowed, with*

$$a_m = re^{i\omega}, \qquad 0 < r < 1,$$

then the real part of the induced kernel is

$$r^\ell \cos(\omega\ell),$$

which gives a damped seasonal or oscillatory component.

Thus, although `PerTIDE` *names its branches as long-term, short-term, and episodic traces, their kernel mixtures can express richer classical time-series structures when the learned modes and fusion weights demand them.*

Proof. Each stable real mode contributes the basis kernel a_m^ℓ. A weighted sum of such modes gives

$$K_M(\ell) = \sum_{m=1}^{M} c_m a_m^\ell.$$

Modes with a_m close to one decay slowly and therefore retain low-frequency or trend-like evidence over finite horizons. Modes with smaller a_m decay quickly and therefore emphasize recent deviations. Combining small and large bases yields a multi-rate filter that separates short-lived fluctuations from persistent components.

For complex-conjugate modes, let

$$a_m = re^{i\omega}, \qquad \bar{a}_m = re^{-i\omega}, \qquad 0 < r < 1.$$

Their powers are

$$a_m^\ell = r^\ell e^{i\omega\ell}, \qquad \bar{a}_m^\ell = r^\ell e^{-i\omega\ell}.$$

A real linear combination of the pair yields terms proportional to

$$r^\ell \cos(\omega\ell)$$

and

$$r^\ell \sin(\omega\ell).$$

These are damped oscillatory kernels, which correspond to stable seasonal or periodic components. Therefore, stable modal mixtures can go beyond the literal long/short/episodic interpretation and approximate broader temporal filters used in classical time-series modeling. □

Corollary D.14 (Interpretation is inductive, not exclusive)**.** *The L/S/E decomposition should be interpreted as a structured inductive bias rather than a hard semantic partition. The architecture encourages persistent, recency-sensitive, and finite-window retention, while the learned modal bases and adaptive fusion can also support trend-like, multi-rate, or damped seasonal traces.*

Proof. This follows immediately from Proposition D.13. Since the trace outputs are fused after being generated by stable modal kernels, the model can combine the three branches into richer temporal filters than any single branch name suggests. □

Trace-Level Separation

Combining the preceding results, the same action-gated event stream is read through three different eigenvalue-mapped kernels:

$$\tilde{m}_{t,i}^{(L)} = \sum_{s=1}^{t} \left(\exp(\delta\lambda_{L,i})\right)^{t-s} u_{s,i}^{(L)},$$

$$\tilde{m}_{t,j}^{(S)} = \sum_{s=1}^{t} \left(\sigma(-\kappa\lambda_{S,j})\right)^{t-s} u_{s,j}^{(S)},$$

$$\tilde{m}_{t,k}^{(E)} = \sum_{s=1}^{t} \left(1 + \eta\lambda_{E,k}\right)^{t-s} u_{s,k}^{(E)}.$$

Therefore, the three traces expose three explicit retention mechanisms:

$$\begin{aligned} \text{persistent build-up} &\quad \leftrightarrow \quad \exp(\delta\lambda_L), \\ \text{recent burst tracking} &\quad \leftrightarrow \quad \sigma(-\kappa\lambda_S), \\ \text{finite-window event imprint} &\quad \leftrightarrow \quad 1+\eta\lambda_E. \end{aligned}$$

This establishes temporal differentiation at the kernel level. Repeated stable interactions are naturally retained by slow long-term modes; recent bursts are emphasized by short-memory modes; and rare events or commands leave controlled decaying imprints through episodic modes. The proof establishes the architectural bias induced by the spectral maps, while the empirical trace-subset and temporal-regime ablations test whether these traces become functionally non-redundant in practice.

D.5 Decoder Contextualization Variants

Setup. For each generative training instance, the encoder produces (i) a task-adapted user state $\mathbf{z}_u^{(t_l)}$ (the representation used by the downstream task head) and (ii) a predicted query-conditioned representation $\hat{\mathbf{z}}_{q,u}^{(t_{l+1})}$. Let $\mathbf{e}_{d_q}$ denote the query-document embedding (or query-side input embedding used in the task head). We perform personalization via a lightweight contextualization module that produces a fixed-dimensional vector $\mathbf{c}$, which is then provided to a frozen autoregressive decoder.

Decoder-UFI (User Fused Injection). Decoder-UFI forms a fused user–query context without explicit user-conditioned modulation of $\hat{\mathbf{z}}_{q,u}^{(t_{l+1})}$. We instantiate UFI as a learned transformation over the concatenated query-conditioned representation and query embedding:

$$\mathbf{c}_{\text{UFI}} = \text{MLP}_{\text{UFI}}\Big(\big[\hat{\mathbf{z}}_{q,u}^{(t_{l+1})}\,;\,\mathbf{e}_{d_q}\big]\Big), \tag{9}$$

where $[\cdot\,;\cdot]$ denotes concatenation. This variant provides a single fused control vector and leaves any higher-order interaction between the user signal and the query semantics to the (frozen) decoder dynamics induced by the injected context.

Decoder-UGI (User Gated Injection). Decoder-UGI explicitly modulates the predicted query-conditioned representation using the task-adapted user state. Concretely, we generate feature-wise scale and shift parameters from $\mathbf{z}_u^{(t_l)}$ and apply FiLM-style gating:

$$\boldsymbol{\gamma} = \text{MLP}_{\gamma}\big(\mathbf{z}_u^{(t_l)}\big), \qquad \boldsymbol{\beta} = \text{MLP}_{\beta}\big(\mathbf{z}_u^{(t_l)}\big), \qquad \tilde{\mathbf{z}}_{q,u}^{(t_{l+1})} = \boldsymbol{\gamma} \odot \hat{\mathbf{z}}_{q,u}^{(t_{l+1})} + \boldsymbol{\beta}, \tag{10}$$

where $\odot$ is the Hadamard product. We then fuse the gated query representation with the query embedding to obtain the decoder control:

$$\mathbf{c}_{\text{UGI}} = \text{MLP}_{\text{UGI}}\Big(\big[\tilde{\mathbf{z}}_{q,u}^{(t_{l+1})}\,;\,\mathbf{e}_{d_q}\big]\Big). \tag{11}$$

Compared to UFI, UGI yields an explicit mechanism to emphasize or suppress semantic components of the query-conditioned representation in a user-dependent manner.

Injecting the context into a frozen decoder. Let Dec_{θ_D} be a frozen decoder with embedding dimension d. We map the context vector $\mathbf{c} \in \mathbb{R}^{d_c}$ (either $\mathbf{c}_{\text{UFI}}$ or $\mathbf{c}_{\text{UGI}}$) to a *soft context token* and prepend it to the decoder input sequence:

$$\mathbf{p} = \mathbf{W}_p\mathbf{c} + \mathbf{b}_p \in \mathbb{R}^d, \qquad \mathbf{H}^{(0)} = \big[\mathbf{p}\,;\,\text{Emb}(y_1)\,;\,\ldots\,;\,\text{Emb}(y_T)\big], \tag{12}$$

where $\text{Emb}(\cdot)$ is the decoder token embedding function. The decoder produces hidden states $\mathbf{H}^{(L)} = \text{Dec}_{\theta_D}(\mathbf{H}^{(0)})$ and next-token distributions via the standard autoregressive factorization $p(y_t \mid y_{<t}, \mathbf{c})$. In all variants, decoder parameters θ_D remain fixed; only the contextualization parameters (e.g., MLP_{UFI}, MLP_{γ}, MLP_{β}, MLP_{UGI}, and $\mathbf{W}_p, \mathbf{b}_p$) are learned.

E Baseline Details

E.1 Encoder Baselines

Notation. Let $\tau_{1:i-1}^u = \{b_u^{(t_j)}\}_{j=1}^{i-1}$ denote the prefix of a user's I-tier trajectory up to step t_{i-1}. Each interaction block is embedded as $\mathbf{x}_j = \phi\big(b_u^{(t_j)}\big) \in \mathbb{R}^d$, and the history encoder produces a single latent user state $\mathbf{z}_u^{(t_i)} \in \mathbb{R}^d$ used by downstream heads.

Profile Aggregation Hypothesis (PAH). PAH assumes that preference can be summarized by an *order-invariant* aggregation over past interaction blocks. The core PAH form is a permutation-invariant operator $f_{\mathrm{Agg}}(\cdot)$ over the set of past embeddings:

$$\mathbf{z}_u^{(t_i)} \;=\; f_{\mathrm{Agg}}\big(\{\mathbf{x}_j\}_{j=1}^{i-1}\big)\,. \tag{13}$$

We instantiate PAH with the following encoder baselines, all of which are realizations of Eq. (13):

Meanpool-B.

$$\mathbf{z}_u^{(t_i)} \;=\; \frac{1}{i-1}\sum_{j=1}^{i-1}\mathbf{x}_j. \tag{14}$$

Maxpool-B.

$$\big[\mathbf{z}_u^{(t_i)}\big]_k \;=\; \max_{1\le j\le i-1}\big[\mathbf{x}_j\big]_k, \quad \forall k\in\{1,\dots,d\}. \tag{15}$$

NAML (PAH-equivalent). NAML-style encoders first produce per-item representations $\mathbf{h}_j$ from multi-view features, then apply attention pooling as a weighted sum:

$$\mathbf{z}_u^{(t_i)} \;=\; \sum_{j=1}^{i-1}\alpha_j\,\mathbf{h}_j, \qquad \alpha_j \;=\; \frac{\exp(s(\mathbf{h}_j))}{\sum_{q=1}^{i-1}\exp(s(\mathbf{h}_q))}. \tag{16}$$

This is PAH-equivalent because the history is reduced to an order-invariant convex combination of per-item vectors.

PLM-NR (PAH-equivalent). PLM-NR-style encoders replace the item encoder with a pretrained language model to obtain $\mathbf{h}_j$, but retain an order-invariant aggregation operator (typically attention pooling as in Eq. (16)). Thus, PLM-NR fits PAH through the same aggregation form with a stronger $\phi(\cdot)$.

Markovian Drift Hypothesis (MDH). MDH assumes a *first-order* state evolution: the next state depends only on the previous state and the current block. The core MDH form is:

$$\mathbf{z}_u^{(t_i)} \;=\; f_{\mathrm{Rec}}\big(\mathbf{z}_u^{(t_{i-1})},\,\mathbf{x}_i;\,\rho\big) \;+\; \boldsymbol{\xi}^{(t_i)}, \tag{17}$$

where ρ controls the effective recency horizon and $\boldsymbol{\xi}^{(t_i)}$ denotes a (possibly action-dependent) perturbation.

GRU-B (MDH instantiation). We instantiate MDH with a gated recurrent unit (GRU) over the ordered b-cell embeddings $\{\mathbf{x}_i\}_{i=1}^{L}$, producing a single-scale hidden state that compresses the prefix into $\mathbf{z}_u^{(t_L)}$. Concretely, given $\mathbf{h}_{i-1}\in\mathbb{R}^d$ and input $\mathbf{x}_i\in\mathbb{R}^d$, a GRU update is:

$$\mathbf{r}_i = \sigma(\mathbf{W}_r\mathbf{x}_i + \mathbf{U}_r\mathbf{h}_{i-1}), \quad \mathbf{u}_i = \sigma(\mathbf{W}_u\mathbf{x}_i + \mathbf{U}_u\mathbf{h}_{i-1}), \tag{18}$$

$$\tilde{\mathbf{h}}_i = \tanh(\mathbf{W}_h\mathbf{x}_i + \mathbf{U}_h(\mathbf{r}_i\odot\mathbf{h}_{i-1})), \tag{19}$$

$$\mathbf{h}_i = (1-\mathbf{u}_i)\odot\mathbf{h}_{i-1} + \mathbf{u}_i\odot\tilde{\mathbf{h}}_i, \qquad \mathbf{z}_u^{(t_L)}\equiv\mathbf{h}_L, \tag{20}$$

where $\sigma(\cdot)$ is the logistic sigmoid, $\odot$ is element-wise multiplication, and $\{\mathbf{W}_\cdot,\mathbf{U}_\cdot\}$ are learned parameters. This realizes the MDH assumption that $\mathbf{z}_u^{(t_i)}$ depends on $(\mathbf{z}_u^{(t_{i-1})},\mathbf{x}_i)$ via a first-order transition.

EBNR (MDH instantiation). EBNR encodes a user's click-order dynamics with a recurrent user encoder and ranks candidates by matching a user embedding to a candidate-news embedding. In its canonical form, each clicked news item d_i is mapped to an item embedding $\mathbf{n}_i\in\mathbb{R}^d$ (the original EBNR paper uses an embedding-learning stage for news representations, followed by an RNN user encoder), and the user state is updated sequentially:

$$\mathbf{h}_i = \mathrm{GRU}(\mathbf{h}_{i-1},\mathbf{n}_i), \qquad \mathbf{u}\equiv\mathbf{h}_L, \tag{21}$$

where $\mathbf{u}$ is the final user embedding after processing the clicked sequence. Given a candidate news item d, with embedding $\mathbf{n}(d)$, EBNR computes a relevance score by an inner product and ranks candidates by this score:

$$s(u,d) = \mathbf{u}^\top\mathbf{n}(d). \tag{22}$$

A common probabilistic instantiation is $\hat{y}(u,d)=\sigma(s(u,d))$, trained with a binary objective over observed clicks and sampled (or impression-based) non-clicks; however, the core modeling assumption that distinguishes EBNR within our taxonomy is the *Markovian* GRU transition over the click-ordered history.

Long-term Dynamics Hypothesis (LDH). LDH assumes that the state at t_i can directly access *all* prior interaction blocks, with learned temporal weighting. A generic LDH form is:

$$\mathbf{z}_u^{(t_i)} = \sum_{j=1}^{i-1} f_K(t_i - t_j)\, \psi_\Theta(\mathbf{x}_j), \tag{23}$$

where $f_K(\cdot)$ is a learnable (or parameterized) temporal weighting kernel and $\psi_\Theta(\cdot)$ is a learned transform.

D-FM Attention (LDH instantiation). We instantiate LDH as an implementation of FoxFormer Lin et al. [2025], with a causal attention encoder that assigns nonzero weight to all prior steps while explicitly down-weighting temporally distant events via an exponential decay factor. Given per-step inputs $\mathbf{x}_j$, we form projections $\mathbf{q}_i = \mathbf{W}_Q \mathbf{x}_i$, $\mathbf{k}_j = \mathbf{W}_K \mathbf{x}_j$, and $\mathbf{v}_j = \mathbf{W}_V \mathbf{x}_j$. For each step i, attention logits over the prefix $j < i$ are:

$$\ell_{ij} = \frac{\mathbf{q}_i^\top \mathbf{k}_j}{\sqrt{d}} - \gamma\,(t_i - t_j), \qquad \text{for } j < i, \tag{24}$$

and the causal weights are normalized by a softmax over valid indices:

$$\alpha_{ij} = \frac{\exp(\ell_{ij})}{\sum_{q=1}^{i-1} \exp(\ell_{iq})}, \qquad \mathbf{z}_u^{(t_i)} = \sum_{j=1}^{i-1} \alpha_{ij}\, \mathbf{v}_j. \tag{25}$$

This realizes Eq. (23) with a learned content-based kernel (dot-product similarity) modulated by an explicit time-decay term controlled by γ.

Discretized SSM (single-timescale LDH instantiation). A single-timescale (linear) state-space model (SSM) provides a full-prefix mechanism through its latent recursion, as used in SS4Rec Xiao et al. [2025]. Let $\mathbf{m}_i \in \mathbb{R}^{d_m}$ be the memory state and $\mathbf{x}_i \in \mathbb{R}^d$ the input embedding at step i. We update:

$$\mathbf{m}_i = \mathbf{A}\mathbf{m}_{i-1} + \mathbf{B}\mathbf{x}_i, \qquad \mathbf{z}_u^{(t_i)} = \mathbf{C}\mathbf{m}_i, \tag{26}$$

(optionally followed by a pointwise nonlinearity on $\mathbf{m}_i$ or $\mathbf{z}_u^{(t_i)}$). Crucially, the recursion induces an explicit full-prefix expansion:

$$\mathbf{m}_i = \mathbf{A}^i \mathbf{m}_0 + \sum_{j=1}^{i} \mathbf{A}^{\,i-j} \mathbf{B}\mathbf{x}_j, \tag{27}$$

so $\mathbf{z}_u^{(t_i)}$ depends on all prior $\{\mathbf{x}_j\}_{j \le i}$ through the powers of $\mathbf{A}$. This places the model in LDH: long-horizon influence is preserved by the transition dynamics rather than a strictly Markovian *lossy* compression at each step.

Transformer-style LDH baselines (NRMS / FastFormer-PLM / MINER). These encoders compute global interactions over the full prefix using attention-style mechanisms, so each $\mathbf{z}_u^{(t_i)}$ is a learned weighted mixture of (potentially) all prior $\{\mathbf{x}_j\}_{j<i}$. A standard multi-head self-attention layer is:

$$\mathrm{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\!\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{V}, \qquad \mathrm{MHSA}(\mathbf{X}) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_H)\mathbf{W}_O, \tag{28}$$

where $\mathbf{X} = [\mathbf{x}_1; \ldots; \mathbf{x}_{i-1}]$ and $\mathrm{head}_h = \mathrm{Attn}(\mathbf{X}\mathbf{W}_Q^{(h)}, \mathbf{X}\mathbf{W}_K^{(h)}, \mathbf{X}\mathbf{W}_V^{(h)})$. To obtain a single user state, these models apply a pooling operator over the attended sequence, commonly additive attention:

$$\alpha_j = \frac{\exp(\mathbf{v}^\top \tanh(\mathbf{W}\mathbf{x}'_j))}{\sum_{q=1}^{i-1} \exp(\mathbf{v}^\top \tanh(\mathbf{W}\mathbf{x}'_q))}, \qquad \mathbf{z}_u^{(t_i)} = \sum_{j=1}^{i-1} \alpha_j\, \mathbf{x}'_j, \tag{29}$$

with $\mathbf{x}'_j$ denoting the contextualized outputs of the attention stack. NRMS instantiates this pattern by applying multi-head self-attention over the browsed-news sequence and then pooling to a user embedding. FastFormer-PLM replaces quadratic self-attention with additive/global-context attention to compute contextualized token (or item) representations with linear complexity in sequence length. MINER further extends pooling to represent diverse interests by extracting multiple interest vectors from a long history (multi-interest user representation), which are then matched against candidates during ranking.

E.2 Baseline Personalized Models

PENS-NRMS Injection-Type 1. The PENS framework [Ao et al., 2021] generates personalized summaries by incorporating user embeddings along with the input news article. For this variant, user embeddings are derived using NRMS (Neural News Recommendation with Multi-Head Self-Attention) [Wu et al., 2019c], which includes a multi-head self-attention based news encoder to represent news titles, and a user encoder that captures browsing behavior through multi-head self-attention over clicked articles. Additive attention mechanisms are employed to highlight important words and articles. In Injection-Type 1, the NRMS user embedding is injected by initializing the decoder's hidden state, thereby directly influencing the summary generation process from the start.

PENS-NRMS Injection-Type 2. This variant also uses NRMS for user embedding, but personalization is introduced differently. Instead of initializing the decoder, the user embedding is injected into the attention mechanism of the PENS model. This modulates the attention weights over the news body, enabling the model to focus on content aligned with the user's preferences.

PENS-NAML Injection-Type 1. NAML (Neural News Recommendation with Attentive Multi-View Learning) [Wu et al., 2019a] generates news representations by attending over multiple views, including titles, bodies, and topic categories. The user encoder learns from interacted news and selects the most informative content for personalization. The resulting user embedding is integrated into the PENS decoder using Injection-Type 1, i.e., by initializing the decoder's hidden state.

PENS-EBNR Injection-Type 1. EBNR (Embedding-based News Recommendation) [Okura et al., 2017] models user preferences using an RNN over browsing histories to produce user embeddings. These embeddings are injected into the PENS model via Injection-Type 1 by initializing the decoder, thereby influencing the initial decoding steps with user-specific information.

PENS-EBNR Injection-Type 2. This configuration uses the same user encoder from EBNR but applies Injection-Type 2. Here, the user embedding is incorporated into the decoder's attention layers, allowing the model to personalize attention distributions over the news body during decoding.

General Then Personal (GTP). General Then Personal (GTP) [Song et al., 2023] is a two-stage framework for personalized headline generation. In stage-1, a Transformer-based encoder–decoder model is pre-trained on large-scale news article–headline pairs to learn robust, content-focused headline generation without personalization. In stage-2, a separate "headline customizer" refines the general headline by incorporating user-specific preferences, which are encoded as a control code by the user encoder TrRMIo. To bridge the gap between general generation and personalized refinement, GTP introduces two mechanisms: (i) **Information Self-Boosting (ISB)**, which reintroduces relevant content details from the article to prevent information loss during customization; and (ii) **Masked User Modeling (MUM)**, which randomly masks parts of the user embedding during training and reconstructs them, reducing the model's over-reliance on its general parameters.

Signature Phrase. Another line of personalization focuses on condensing a user's reading history into a collection of *signature phrases* [Cai et al., 2023]. These phrases, derived through contrastive learning over news articles without annotated data, act as dynamic user profiles that adapt as interests evolve. Such phrases need not appear verbatim in the user's history but instead encode higher-level signals. Using these phrases, the model learns to generate personalized headlines that connect new articles with the user's inferred interests, yielding outputs that are engaging, relevant, and grounded in article content rather than drifting toward clickbait.

MovieLens-1M Baselines. For MovieLens-1M, we instantiate the predictive recommendation baselines under the same state-assumption taxonomy used in Section 1 and Appendix H.1. Each user history is an ordered prefix $\tau^u_{1:i-1} = (v^u_1, \ldots, v^u_{i-1})$, where v^u_j is the movie interacted with by user u at timestep j. Each model produces a candidate score $s(u, c)$ for a target movie candidate c, and candidates are ranked under the same one-positive plus sampled-negative protocol used in the MovieLens evaluation.

BPR-MF (PAH Instantiation). BPR-MF is used as a static profile-based matrix factorization baseline. It assigns each user u a persistent embedding $p_u \in \mathbb{R}^d$ and each movie candidate c an item

embedding $q_c \in \mathbb{R}^d$, and scores candidates as

$$s(u, c) = p_u^\top q_c.$$

The pairwise ranking objective is

$$\mathcal{L}_{\mathrm{BPR}} = -\sum_{(u,i^+,i^-)} \log \sigma\big(s(u, i^+) - s(u, i^-)\big)\,.$$

Its formulation class is static user–item factorization. We categorize BPR-MF under PAH because the exposed user representation is an order-insensitive persistent profile vector rather than a state computed from the ordered interaction prefix.

GRU4Rec (MDH Instantiation). GRU4Rec is used as a recurrent sequential recommendation baseline. Given item embeddings $\{q_{v_j}\}_{j<i}$, the user state is updated through a first-order gated recurrence:

$$h_j = \mathrm{GRU}(h_{j-1}, q_{v_j}), \qquad z_u^{(t_i)} = h_{i-1},$$

and a candidate movie is scored by

$$s(u, c) = z_u^{(t_i)\top} q_c.$$

Its formulation class is recurrent sequential drift. We categorize GRU4Rec under MDH because the current user state is a single recurrently compressed state obtained from the previous state and the current interaction.

Caser (MDH Instantiation). Caser is used as a convolutional sequential recommendation baseline. Instead of processing the entire prefix through attention, Caser encodes a recent interaction window as a sequence embedding matrix:

$$X_i = [q_{v_{i-L}}; \ldots; q_{v_{i-1}}],$$

then applies horizontal and vertical convolutional filters over this window:

$$z_u^{(t_i)} = \mathrm{MLP}([\mathrm{Conv}_h(X_i); \mathrm{Conv}_v(X_i); p_u])\,, \qquad s(u, c) = z_u^{(t_i)\top} q_c.$$

Its formulation class is local-window convolutional drift. We categorize Caser under MDH because the exposed state is dominated by a bounded recent interaction window, rather than by an order-invariant profile or an explicit full-prefix temporal kernel.

S^{3}Rec (LDH Instantiation). S^{3}Rec is used as a self-supervised Transformer-style sequential recommendation baseline. It first encodes the item prefix through a self-attentive sequence encoder,

$$H_i = \mathrm{SeqEnc}_\theta\left([q_{v_1} + p_1; \ldots; q_{v_{i-1}} + p_{i-1}]\right), \qquad z_u^{(t_i)} = \mathrm{Readout}(H_i),$$

and scores a candidate movie by

$$s(u, c) = z_u^{(t_i)\top} q_c.$$

Its formulation class is self-supervised full-prefix sequence encoding. We categorize S^{3}Rec under LDH because the user state is obtained from a Transformer sequence encoder that can attend over the full observed prefix, with auxiliary self-supervised objectives improving sequence representation learning.

Diff4Rec (LDH Instantiation). Diff4Rec is used as a diffusion-augmented sequential recommendation baseline. It conditions recommendation on a sequence representation of the user prefix and improves the sequential recommender through diffusion-based augmentation of user–item interaction signals:

$$z_u^{(t_i)} = \mathrm{SeqEnc}_\theta(\tau_{1:i-1}^u), \qquad s(u, c) = f_\theta(z_u^{(t_i)}, q_c).$$

Its formulation class is diffusion-augmented sequence modeling. We categorize Diff4Rec under LDH because the candidate score is produced from a sequence-conditioned representation of the interaction prefix, rather than from a static profile or a purely local recurrent state.

SASRec (LDH Instantiation). SASRec is used as a causal self-attention baseline for sequential recommendation. Given the ordered item prefix, it applies masked self-attention so that the state at timestep t_i is computed from previous interactions:

$$H_i = \mathrm{CausalSelfAttn}_\theta\left([q_{v_1} + p_1; \ldots; q_{v_{i-1}} + p_{i-1}]\right), \qquad z_u^{(t_i)} = \mathrm{Readout}(H_i),$$

$$s(u, c) = z_u^{(t_i)\top} q_c.$$

Its formulation class is causal Transformer sequence encoding. We categorize SASRec under LDH because the current recommendation state is computed by a learned attention kernel over the full prefix.

BERT4Rec (LDH Instantiation). BERT4Rec is used as a bidirectional Transformer baseline for sequential recommendation. It learns contextualized item representations through masked-item prediction over the observed behavior sequence:

$$H_i = \mathrm{BiTransformer}_\theta\left([q_{v_1} + p_1; \ldots; q_{v_{i-1}} + p_{i-1}]\right), \qquad z_u^{(t_i)} = \mathrm{Readout}(H_i).$$

Candidate movies are ranked by matching the resulting user state with candidate item representations:

$$s(u, c) = z_u^{(t_i)\top} q_c.$$

Its formulation class is bidirectional masked sequence encoding. We categorize BERT4Rec under LDH because it contextualizes the interaction history through a full-sequence Transformer representation before scoring candidates.

Mamba4Rec (LDH Instantiation). Mamba4Rec is used as a selective state-space sequential recommendation baseline. It replaces quadratic self-attention with selective state-space sequence modeling:

$$m_j = A_j m_{j-1} + B_j q_{v_j}, \qquad z_u^{(t_i)} = C_{i-1} m_{i-1}, \qquad s(u, c) = z_u^{(t_i)\top} q_c.$$

Its formulation class is selective state-space sequence encoding. We categorize Mamba4Rec under LDH because the exposed state summarizes the interaction prefix through a learned state-space dynamics kernel capable of preserving long-range sequence information.

SSD4Rec (LDH Instantiation). SSD4Rec is used as a structured state-space duality baseline for efficient sequential recommendation. It processes variable-length movie interaction sequences through structured state-space duality blocks:

$$m_j = \mathrm{SSD}_\theta(m_{j-1}, q_{v_j}), \qquad z_u^{(t_i)} = \mathrm{Readout}(m_{i-1}), \qquad s(u, c) = z_u^{(t_i)\top} q_c.$$

Equivalently, the state-space recursion induces a full-prefix dependence:

$$m_{i-1} = \left(\prod_{\ell=1}^{i-1} A_\ell\right) m_0 + \sum_{j=1}^{i-1} \left(\prod_{\ell=j+1}^{i-1} A_\ell\right) B_j q_{v_j}.$$

Its formulation class is structured state-space sequence dynamics. We categorize SSD4Rec under LDH because it preserves prefix information through learned sequence dynamics, but still exposes a single compressed sequence state rather than explicit long-, short-, and episodic traces.

F Measuring Degree-of-Personalization

F.1 Motivation

EGISES [Vansh et al., 2023] measures how insensitive a personalized summarizer is to user subjectivity, where lower values indicate less insensitivity and therefore better personalization. Following this idea, PerSEval [Dasgupta et al., 2024] evaluates whether a model changes its generated summary when the target user's expected summary changes, while also penalizing poor content accuracy. We use the notation d_i for a document, u_{ij} for the expected summary of user j for document d_i, and $\hat{s}_{ij}$ for the model-generated summary for the same user-document pair. Let U_i be the set of users who provide references for document d_i, and let $\sigma(\cdot, \cdot)$ be a distance function between two summaries.

The summary-level degree of responsiveness, denoted as DEGRESS, compares two quantities: the divergence between user references and the divergence between corresponding model outputs. For user j and document d_i, it is defined as

$$D_{ij} = \frac{1}{|U_i|} \sum_{k=1}^{|U_i|} \frac{\min(X_{ijk}, Y_{ijk}) + \epsilon}{\max(X_{ijk}, Y_{ijk}) + \epsilon}, \tag{30}$$

where

$$X_{ijk} = w^u_{ijk} \sigma(u_{ij}, u_{ik}), \qquad Y_{ijk} = w^s_{ijk} \sigma(\hat{s}_{ij}, \hat{s}_{ik}). \tag{31}$$

Here, X_{ijk} measures the distance between the expected summaries of users j and k, and Y_{ijk} measures the corresponding distance between the model-generated summaries. The weights w^u_{ijk} and w^s_{ijk} are softmax-normalized document-relative weights. They control how much each user-pair comparison contributes to the final responsiveness score.

The system-level DEGRESS score is

$$D(M) = \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \frac{1}{|U_i|} \sum_{j=1}^{|U_i|} D_{ij}. \tag{32}$$

A low value indicates that the model does not respond sufficiently to user-specific reference variation. A high value indicates stronger responsiveness to different user preferences.

F.2 PerSEval Formulation

DEGRESS only measures responsiveness. It does not penalize a model that produces personalized but inaccurate summaries. PerSEval addresses this limitation by multiplying DEGRESS with an accuracy-sensitive penalty. For user j and document d_i, the summary-level PerSEval score is

$$P_{ij} = D_{ij} \cdot E_{ij}, \tag{33}$$

where E_{ij} is the effective DEGRESS penalty factor:

$$E_{ij} = 1 - \frac{1}{1 + 10^{\alpha} \exp(-10^{\beta} Q_{ij})}, \qquad \alpha \geq 3,\ \beta \geq 1. \tag{34}$$

The penalty input Q_{ij} combines document-level accuracy loss and user-level accuracy inconsistency:

$$Q_{ij} = A_i + C_{ij}. \tag{35}$$

The document-level accuracy penalty is

$$A_i = \frac{1}{1 + 10^{\gamma} \exp\left(-10 \cdot \frac{a^*_i}{1 - a^*_i + \epsilon}\right)}, \qquad \gamma \geq 4, \tag{36}$$

where

$$a^*_i = \min_{j \in U_i} a_{ij}. \tag{37}$$

Here, a_{ij} is the distance-based accuracy score for $\hat{s}_{ij}$ against the corresponding user reference u_{ij}, and a^*_i is the best document-level accuracy term under the chosen distance convention.

The accuracy-inconsistency penalty is

$$C_{ij} = \frac{1}{1 + 10^{\gamma} \exp\left(-10 \cdot \frac{a_{ij} - a^*_i}{\bar{a}_i - a^*_i + \epsilon}\right)}, \qquad \gamma \geq 4, \tag{38}$$

where

$$\bar{a}_i = \frac{1}{|U_i|} \sum_{j=1}^{|U_i|} a_{ij}. \tag{39}$$

Thus, A_i penalizes poor best-case accuracy, while C_{ij} penalizes inconsistency across user-specific generations for the same document.

The system-level PerSEval score is

$$P(M) = \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \frac{1}{|U_i|} \sum_{j=1}^{|U_i|} P_{ij}. \tag{40}$$

The score lies in $[0, 1]$ and remains bounded by the system-level DEGRESS score.

F.3 PSE Metrics

We use three PerSEval variants by changing the distance or similarity function used inside the formulation.

PSE-SU4. This variant uses ROUGE-SU4 as the base comparison function. ROUGE-SU4 measures overlap through unigrams and skip-bigrams, where a skip-bigram is an ordered word pair with up to four skipped words between the pair. We use the ROUGE-SU4 F1 score inside the PerSEval formulation.

PSE-JSD. This variant uses Jensen-Shannon divergence between the normalized unigram or n-gram distributions of the generated and reference summaries. For distributions P and Q,

$$\mathrm{JSD}(P\|Q) = \frac{1}{2}\mathrm{KL}(P\|M) + \frac{1}{2}\mathrm{KL}(Q\|M), \qquad M = \frac{1}{2}(P+Q). \tag{41}$$

We set $\sigma(G, R) = \mathrm{JSD}(P_G\|P_R)$ to obtain PSE-JSD.

PSE-METEOR. This variant uses METEOR as the base similarity score. METEOR aligns generated and reference unigrams using exact, stem, synonym, and paraphrase matches, then combines precision, recall, and a fragmentation penalty. Since PerSEval requires a distance, we use

$$\sigma(G, R) = 1 - \mathrm{METEOR}(G, R). \tag{42}$$

This gives PSE-METEOR.

G Prediction Accuracy Metrics

(i) **nDCG@5 and HR@10.** We report normalized Discounted Cumulative Gain (nDCG) at cutoffs $K \in \{5, 10\}$ to evaluate ranking quality while emphasizing higher-ranked items. For each query instance with a ranked list of candidates, let rel_i denote the relevance of the item at rank i (binary relevance in our setup). We compute

$$\mathrm{DCG@K} = \sum_{i=1}^{K} \frac{2^{\mathrm{rel}_i} - 1}{\log_2(i+1)}, \qquad \mathrm{nDCG@K} = \frac{\mathrm{DCG@K}}{\mathrm{IDCG@K}}, \tag{43}$$

where IDCG@K is the maximum possible DCG@K achievable by an ideal ordering for that instance. nDCG@5 and HR@10 quantify both correctness and ordering, with larger values indicating better ranked relevance concentration near the top.

(ii) **MRR.** We report Mean Reciprocal Rank (MRR) to measure how early the first relevant item appears in the ranked list. For each instance, let $\mathrm{rank}^\star$ be the rank position of the first relevant item; the reciprocal rank is $1/\mathrm{rank}^\star$. The aggregate metric is

$$\mathrm{MRR} = \frac{1}{N}\sum_{n=1}^{N} \frac{1}{\mathrm{rank}_n^\star}, \tag{44}$$

where N is the number of evaluation instances. MRR heavily rewards placing a relevant item at the very top and is therefore sensitive to early precision.

(iii) **Recall@K.** We report Recall at cutoff K (denoted Recall@K) to quantify whether at least one relevant item is retrieved among the top-K predictions. With binary relevance and (typically) one target item per instance, Recall@K reduces to an indicator of whether the target appears in the top-K list:

$$\mathrm{Recall@K} = \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}\big[\mathrm{rank}_n^\star \le K\big], \tag{45}$$

where $\mathbb{I}[\cdot]$ is the indicator function. Larger Recall@K indicates higher top-K hit rate, complementing MRR and nDCG by focusing on retrieval coverage rather than graded ordering.

H Content Accuracy Metrics

(i) **ROUGE-2.** ROUGE-2 measures bigram overlap between a generated text y and a reference text r using recall-oriented matching of contiguous 2-grams. Let $G_2(\cdot)$ denote the multiset of bigrams, and let $\mathrm{count}_x(g)$ be the count of bigram g in text x. The ROUGE-2 recall is

$$\text{ROUGE-2} = \frac{\sum_{g \in G_2(r)} \min\left(\mathrm{count}_y(g),\, \mathrm{count}_r(g)\right)}{\sum_{g \in G_2(r)} \mathrm{count}_r(g)}. \tag{46}$$

Higher values indicate that the generation covers more of the reference bigram content.

(ii) **ROUGE-L.** ROUGE-L evaluates sequence-level overlap via the Longest Common Subsequence (LCS), capturing in-order matching without requiring contiguity. Let $\mathrm{LCS}(y, r)$ be the length of the LCS between y and r, and let $|y|, |r|$ be token lengths. Define LCS-based precision and recall as

$$P_{\mathrm{LCS}} = \frac{\mathrm{LCS}(y, r)}{|y|}, \qquad R_{\mathrm{LCS}} = \frac{\mathrm{LCS}(y, r)}{|r|}. \tag{47}$$

ROUGE-L is commonly reported as an F-measure:

$$\text{ROUGE-L} = \frac{(1 + \beta^2) P_{\mathrm{LCS}} R_{\mathrm{LCS}}}{R_{\mathrm{LCS}} + \beta^2 P_{\mathrm{LCS}}}, \tag{48}$$

where β controls the relative weight of recall versus precision (often set to favor recall in summarization).

(iii) **ROUGE-SU4.** ROUGE-SU4 measures overlap of skip-bigrams with a maximum skip distance of 4, together with unigrams (the "U" component). A skip-bigram is any ordered pair of tokens (w_i, w_j) such that $1 \le j - i \le 5$ (i.e., up to four skipped tokens between them). Let $G_{\mathrm{SU4}}(x)$ be the multiset containing all unigrams and SU4 skip-bigrams from text x. The ROUGE-SU4 recall is

$$\text{ROUGE-SU4} = \frac{\sum_{g \in G_{\mathrm{SU4}}(r)} \min\left(\mathrm{count}_y(g),\, \mathrm{count}_r(g)\right)}{\sum_{g \in G_{\mathrm{SU4}}(r)} \mathrm{count}_r(g)}. \tag{49}$$

This metric is more permissive than ROUGE-2 because it credits in-order matches even when tokens are separated by a small gap.

(iv) **METEOR.** METEOR aligns generated and reference tokens using exact matches and (optionally) linguistic matches (e.g., stem/synonym), then computes a harmonic mean of unigram precision and recall with a fragmentation penalty for disordered alignments. Let m be the number of matched unigrams under the alignment, and let $|y|, |r|$ be token lengths. Define

$$P = \frac{m}{|y|}, \qquad R = \frac{m}{|r|}, \qquad F_\alpha = \frac{PR}{\alpha P + (1 - \alpha) R}, \tag{50}$$

where α weights precision versus recall (commonly set to emphasize recall). Let c be the number of matched "chunks" (contiguous sequences of matches in the aligned order). The penalty is

$$\mathrm{Pen} = \gamma \left(\frac{c}{m}\right)^{\theta}, \tag{51}$$

and the final METEOR score is

$$\text{METEOR} = (1 - \mathrm{Pen})\, F_\alpha, \tag{52}$$

with γ, θ controlling the strength of the fragmentation penalty.

(v) **BLEU-4.** BLEU-4 evaluates n-gram precision up to $n = 4$ with a brevity penalty to discourage overly short outputs. Let p_n denote the modified precision for n-grams, computed using clipped counts against the reference. BLEU-4 is

$$\text{BLEU-4} = \mathrm{BP} \cdot \exp\left(\sum_{n=1}^{4} w_n \log p_n\right), \tag{53}$$

where typically $w_n = \frac{1}{4}$. The brevity penalty is

$$\mathrm{BP} = \begin{cases} 1, & |y| > |r|, \\ \exp\left(1 - \frac{|r|}{|y|}\right), & |y| \le |r|. \end{cases} \tag{54}$$

Higher BLEU-4 indicates greater n-gram precision while accounting for length adequacy via BP.

(vi) **BERTScore.** BERTScore measures semantic similarity between a generated text $y = \{y_i\}_{i=1}^{m}$ and a reference text $r = \{r_j\}_{j=1}^{n}$ by aligning contextual token embeddings from a pretrained encoder. Let $\mathbf{h}(y_i)$ and $\mathbf{h}(r_j)$ denote the contextualized embeddings, and let $s_{ij} = \cos(\mathbf{h}(y_i), \mathbf{h}(r_j))$ be their cosine similarity. BERTScore precision and recall are computed via soft token matching:

$$P_{\mathrm{BS}} = \frac{1}{m} \sum_{i=1}^{m} \max_{1 \le j \le n} s_{ij}, \qquad R_{\mathrm{BS}} = \frac{1}{n} \sum_{j=1}^{n} \max_{1 \le i \le m} s_{ij}, \tag{55}$$

and combined into an F-score

$$F_{\mathrm{BS}} = \frac{2 P_{\mathrm{BS}} R_{\mathrm{BS}}}{P_{\mathrm{BS}} + R_{\mathrm{BS}}}. \tag{56}$$

Higher BERTScore indicates stronger semantic alignment between the generation and the reference, even when surface n-gram overlap is low.

I Training Details

I.1 Compute Resources

All dataset ingestion and feature lookup were performed on CPU, including CSV loading and construction of the canonical ID maps (`id2text/id2phrases`) and vocabulary indices (`actions2idx`, `node2idx`, `user2idx`). Dense node embeddings for dialogue/document/response/summary nodes were loaded from precomputed pickled files and resolved on-the-fly during training via canonical IDs, with missing embeddings defaulted to a zero vector of dimension d_{model}=768.

Model training was executed on a single CUDA-enabled GPU (as reported by `torch.device("cuda")` in the run logs). The total parameter count of the end-to-end multi-task system (encoder + heads + causal decoder) is ≈160.2M trainable parameters.

I.2 Training

We train the multi-task model using a single-row update scheme: each training example is converted into a batch of size 1, and one optimizer step is applied per valid row.

We train for E=45 epochs over N_{train}=78,274 training rows (with rows that fail parsing or generation-target construction skipped). Per epoch, we deterministically shuffle the active training table with `random_state` set to `SEED+epoch`. We define a processed prefix length T_{proc}=$T-2$ so that the final two positions serve as (query, target).

Optimization uses AdamW (`torch.optim.AdamW`) with learning rate 3×10^{-4} and gradient clipping at norm 1.0. The training objective is a masked multi-task sum of: (i) encoder interaction b-node (action+node) prediction loss, (ii) encoder task-classification loss (focal CE), and (iii) separate specialized task-head losses for user prediction, item prediction, and generation. Generation is trained by prefix injection into a causal LM decoder (DistilGPT in our runs) with a learned prefix length P=8, where prefix positions are excluded from the token-level loss via `IGNORE_INDEX`.

Crash-safe checkpointing and validation. We save row checkpoints every 15,000 processed rows and an additional end-of-epoch checkpoint. Row checkpoints store the current epoch and row index, and resumption continues from the next row. After each epoch, we evaluate on a deterministic validation slice of up to 500 examples per task, reporting overall and per-task mean losses (total loss and its encoder/user/item/generation components).

I.3 Statistical Significance Testing

We assess statistical significance using paired bootstrap resampling over the held-out evaluation pool. The goal is to test whether the improvement of `PerTIDE` over a paired comparator is stable under random resampling of the same test instances, rather than being driven by a particular subset of users, histories, rollout positions, or candidates. For every dataset–task–metric tuple, the two systems being compared are evaluated on identical held-out examples and identical history prefixes. For predictive

Table 7: **Notation and symbol table.**

Symbol	Meaning	Symbol	Meaning
UIG and Action-On-Item Schema			
$G_u = \langle \mathcal{N}_u, \mathcal{E}_u \rangle$	User-specific interaction graph.	$\mathcal{N}_u, \mathcal{E}_u$	User-specific nodes and edges.
$u^{(t_0)}$	Initial user anchor node.	$d^{(t_i)}$	Item/content node at timestep t_i.
$r_u^{(t_i)}$	Response node at timestep t_i.	$\alpha^{(t_i)}$	Domain-native action label.
$\mathcal{A}$	Shared action-role vocabulary.	$a^{(t_i)}$	Schema-level action role.
$\rho_D(\cdot)$	Domain-specific role mapping.	$\psi_D(\cdot)$	Domain-specific target-locus embedding map.
$S_D(e)$	Schema-normalized native event.	τ_u	Time-ordered user trajectory.
$\mathcal{T}_{\text{train}}, \mathcal{T}_{\text{test}}$	Train and test trajectory pools.	d_q	Query item/content node.
$\mathbf{g}_q$	Query or task seed embedding.		
Interaction Blocks and Target-Locus Embeddings			
$b_u^{(t_i)} = \langle a^{(t_i)}, tl^{(t_i)} \rangle$	Action role applied to target locus.	$tl^{(t_i)}$	Target locus of current action.
$\mathbf{e}_{tl}^{(t_i)}$	Target-locus embedding.	$\boldsymbol{\mu}^{(t_i)}$	Action-conditioned gating vector.
$\mathbf{b}_u^{(t_i)}$	Action-gated event cell.	$\phi(\cdot)$	Optional generic block embedder.
$\mathbf{x}_j$	Generic baseline block embedding.	d	Hidden and embedding dimensionality.
MTSH and PerTIDE State Variables			
$\mathbf{m}_u^{(t_i,L)}$	Long-term preference trace.	$\mathbf{m}_u^{(t_i,S)}$	Short-term preference trace.
$\mathbf{m}_u^{(t_i,E)}$	Episodic preference trace.	A_r	Trace transition matrix.
Q_r, Λ_r	Eigenbasis and trace spectrum.	B	Shared trace input operator.
$\Phi(\cdot)$	Multi-trace fusion map.	$\Psi(\cdot)$	Action-conditioned evidence injection map.
$\mathbf{h}_{b_u}^{(t_i)}$	Fused preference-history state.	$\Pi(\cdot)$	Command/task inference map.
$G(\cdot)$	Command-conditioning gate.	$\boldsymbol{\pi}_u^{(t_i)}$	Predicted command/task distribution.
$\mathbf{z}_{b_u}^{(t_i)}$	Task-ready user state.		
Task Heads and Encoder Objectives			
T	Downstream task or command type.	$f_T(\cdot)$	Task-specific prediction head.
$\hat{\mathbf{z}}_{q,u}^{(t_l+1)}$	Query-conditioned next-step state.	$\hat{y}_T$	Predicted task output.
$\hat{p}(\cdot)$	Task-head predictive distribution.	y_T^*	Ground-truth task target.
$\mathcal{B}$	Vocabulary of observed block nodes.	$\mathcal{L}_{\text{pos}}$	Temporal-position alignment loss.
$\mathcal{L}_{\text{next}}$	Next-block prediction loss.	$\mathcal{L}_{\text{task}}$	Command/task classification loss.
$\mathcal{L}_{\text{enc}}$	Combined encoder training objective.	$\mathcal{L}_{\text{obj}}$	Downstream task objective.
Decoder Contextualization: UFI and UGI			
$\mathbf{e}_{d_q}$	Query-document/item embedding.	$\mathbf{c}$	Decoder control vector.
$\mathbf{c}_{\text{UFI}}$	User-fused injection control.	$\boldsymbol{\gamma}$	FiLM scale vector.
$\boldsymbol{\beta}$	FiLM shift vector.	$\odot$	Elementwise product.
$\tilde{\mathbf{z}}_{q,u}^{(t_l+1)}$	User-gated query representation.	$\mathbf{c}_{\text{UGI}}$	User-gated injection control.
$\mathbf{p}$	Soft decoder context token.	$W_p, \mathbf{b}_p$	Soft-token projection parameters.
$\mathbf{H}^{(0)}$	Decoder input states.	$\mathbf{H}^{(L)}$	Final decoder hidden states.
θ_D	Frozen decoder parameters.		

recommendation, the paired systems also use the same candidate sets containing one ground-truth target and 699 sampled negatives.

Let $\mathcal{T}_{\text{eval}} = \{x_n\}_{n=1}^{N}$ denote the held-out evaluation pool for a task. For a model A and metric M, let $M(A; S)$ denote the metric computed on a sampled subset S. For each paired comparison between `PerTIDE` and a comparator A—including the strongest task-native baseline, a PAH/MDH/LDH baseline, or an ablated `PerTIDE` variant—we draw $B = 10$ bootstrap samples $S_1, \ldots, S_B$ from $\mathcal{T}_{\text{eval}}$. The same sampled indices are used for both systems in each bootstrap trial.

Table 8: **PerTIDE encoder hyperparameters and parameter dimensions.** We use a unified width $d = 768$ for node/state/task representations. Action embedding dimension is $d_a = 4$. The b-node vocabulary size is $|\mathcal{B}| = 63{,}000$. All MLPs listed here are instantiated as 2-layer MLPs with hidden width 768 unless noted.

Hyperparameter Symbol	Meaning / Name	Dimensions
	Action-Gated Event Cells	
d	Model / latent width (seed embeddings, memory traces, user state, task embeddings)	768
d_a	Action embedding width	4
$\mathbf{e}_{tl}^{(t_i)}$	Target-locus embedding at timestep t_i; induces item-side abstraction	$\mathbb{R}^{768}$
$\mathbf{a}^{(t_i)}$	Action embedding input to gate MLP	$\mathbb{R}^{4}$
$W_{\text{gate}}^{(1)}$, $\mathbf{b}_{\text{gate}}^{(1)}$	Gate MLP layer-1 parameters ($4 \to 768$)	$\mathbb{R}^{768\times 4}$, $\mathbb{R}^{768}$
$W_{\text{gate}}^{(2)}$, $\mathbf{b}_{\text{gate}}^{(2)}$	Gate MLP layer-2 parameters ($768 \to 768$)	$\mathbb{R}^{768\times 768}$, $\mathbb{R}^{768}$
$\boldsymbol{\mu}^{(t_i)}$	Action-specific gate, $\boldsymbol{\mu}^{(t_i)} = \sigma(\text{MLP}_{\text{gate}}(\mathbf{a}^{(t_i)}))$	$\mathbb{R}^{768}$
$\mathbf{b}_u^{(t_i)}$	b-cell embedding, $\mathbf{b}_u^{(t_i)} = \boldsymbol{\mu}^{(t_i)} \odot \mathbf{e}_{tl}^{(t_i)}$	$\mathbb{R}^{768}$
	Tri-Trace SSM (Long / Short / Episodic)	
$\mathbf{m}_u^{(t_i,r)}$	Memory trace for timescale $r \in \{\text{L}, \text{S}, \text{E}\}$	$\mathbb{R}^{768}$
Q_r	Eigenbasis for trace r (learned)	$\mathbb{R}^{768\times 768}$
Λ_r	Diagonal spectrum for trace r (learned; stored as diagonal entries)	$\mathbb{R}^{768}$
B	Input operator (shared across traces)	$\mathbb{R}^{768\times 768}$
δ	Long-term spectral-map scalar in $\exp(\delta\Lambda_{\text{L}})$	scalar
κ	Short-term spectral-map scalar in $\sigma(-\kappa\Lambda_{\text{S}})$	scalar
η	Episodic spectral-map scalar in $(I + \eta\Lambda_{\text{E}})$	scalar
	Adaptive Memory Mixer	
$\mathbf{h}_{b_u}^{(t_i)}$	Memory-fused b-cell representation	$\mathbb{R}^{768}$
$W_\Phi^{(1)}$, $\mathbf{b}_\Phi^{(1)}$	Mixer MLP layer-1 parameters ($[m^{(L)}; m^{(S)}; m^{(E)}] \in \mathbb{R}^{2304} \to 768$)	$\mathbb{R}^{768\times 2304}$, $\mathbb{R}^{768}$
$W_\Phi^{(2)}$, $\mathbf{b}_\Phi^{(2)}$	Mixer MLP layer-2 parameters ($768 \to 768$)	$\mathbb{R}^{768\times 768}$, $\mathbb{R}^{768}$
	Command-Conditioned Readout (Realization of Π and G)	
$\mathbf{g}_q$	Task embedding provided by task encoder	$\mathbb{R}^{768}$
$\boldsymbol{\pi}_u^{(t_i)}$	Command/task distribution, $\boldsymbol{\pi}_u^{(t_i)} = \text{SoftMax}(W_{\text{task}}[\mathbf{h}_{b_u}^{(t_i)}; \mathbf{g}_q])$	$\mathbb{R}^{\lvert\mathcal{T}\rvert}$
W_{task}	Task-distribution projection (task logits)	$\mathbb{R}^{\lvert\mathcal{T}\rvert\times 1536}$
W_π	Command-to-state gating projection in $G(\cdot)$	$\mathbb{R}^{768\times\lvert\mathcal{T}\rvert}$
$\mathbf{b}_G$	Command-gate bias	$\mathbb{R}^{768}$
$\mathbf{z}_{b_u}^{(t_i)}$	Command-conditioned task-ready preference state after gating	$\mathbb{R}^{768}$
	Encoder Supervision Heads	
l	Trajectory slice length used for encoder supervision	scalar
W_{pos}	Position classifier over $\{1, \ldots, l-1\}$	$\mathbb{R}^{(l-1)\times 768}$
W_{lin}, $\mathbf{b}_{\text{lin}}$	Linear head mapping $\mathbf{z}_{b_u}^{(t_{i-1})} \to \hat{\mathbf{z}}_{b_u}^{(t_i)}$	$\mathbb{R}^{768\times 768}$, $\mathbb{R}^{768}$
$\lvert\mathcal{B}\rvert$	b-node vocabulary size for next-step prediction	63,000
W_{next}, $\mathbf{b}_{\text{next}}$	Query b-node predictor logits over $\mathcal{B}$ (final projection)	$\mathbb{R}^{\lvert\mathcal{B}\rvert\times 768}$, $\mathbb{R}^{\lvert\mathcal{B}\rvert}$
$\lambda_{\text{pos}}, \lambda_{\text{next}}, \lambda_{\text{task}}$	Encoder-loss weights in $\mathcal{L}_{\text{enc}} = \lambda_{\text{pos}}\mathcal{L}_{\text{pos}} + \lambda_{\text{next}}\mathcal{L}_{\text{next}} + \lambda_{\text{task}}\mathcal{L}_{\text{task}}$	scalars
	Shared Decoder-Side Quantities	
d	Frozen decoder embedding width (DistillGPT)	768
d_c	Decoder control-vector width	768
$\hat{\mathbf{z}}_{q,u}^{(t_{i+1})}$	Predicted query-conditioned representation (from encoder)	$\mathbb{R}^{768}$
$\mathbf{z}_u^{(t_i)}$	Task-adapted user state (from encoder)	$\mathbb{R}^{768}$
$\mathbf{e}_{d_q}$	Query-document (query-side) embedding	$\mathbb{R}^{768}$
	Decoder-UFI (User Fused Injection)	
$W_{\text{UFI}}^{(1)}$, $\mathbf{b}_{\text{UFI}}^{(1)}$	UFI MLP layer-1 parameters ($[\hat{\mathbf{z}}_{q,u}; \mathbf{e}_{d_q}] \in \mathbb{R}^{1536} \to 768$)	$\mathbb{R}^{768\times 1536}$, $\mathbb{R}^{768}$
$W_{\text{UFI}}^{(2)}$, $\mathbf{b}_{\text{UFI}}^{(2)}$	UFI MLP layer-2 parameters ($768 \to 768$)	$\mathbb{R}^{768\times 768}$, $\mathbb{R}^{768}$
$\mathbf{c}_{\text{UFI}}$	UFI control vector, $\mathbf{c}_{\text{UFI}} = \text{MLP}_{\text{UFI}}([\hat{\mathbf{z}}_{q,u}; \mathbf{e}_{d_q}])$	$\mathbb{R}^{768}$
	Decoder-UGI (User Gated Injection)	
$W_\gamma^{(1)}$, $\mathbf{b}_\gamma^{(1)}$	FiLM-scale MLP layer-1 parameters ($\mathbf{z}_u^{(t_i)} \in \mathbb{R}^{768} \to 768$)	$\mathbb{R}^{768\times 768}$, $\mathbb{R}^{768}$
$W_\gamma^{(2)}$, $\mathbf{b}_\gamma^{(2)}$	FiLM-scale MLP layer-2 parameters ($768 \to 768$)	$\mathbb{R}^{768\times 768}$, $\mathbb{R}^{768}$
$W_\beta^{(1)}$, $\mathbf{b}_\beta^{(1)}$	FiLM-shift MLP layer-1 parameters ($\mathbf{z}_u^{(t_i)} \in \mathbb{R}^{768} \to 768$)	$\mathbb{R}^{768\times 768}$, $\mathbb{R}^{768}$
$W_\beta^{(2)}$, $\mathbf{b}_\beta^{(2)}$	FiLM-shift MLP layer-2 parameters ($768 \to 768$)	$\mathbb{R}^{768\times 768}$, $\mathbb{R}^{768}$
$\boldsymbol{\gamma}, \boldsymbol{\beta}$	FiLM scale/shift vectors from user state	$\mathbb{R}^{768}$, $\mathbb{R}^{768}$
$\tilde{\mathbf{z}}_{q,u}^{(t_{i+1})}$	Gated query representation, $\tilde{\mathbf{z}}_{q,u} = \boldsymbol{\gamma} \odot \hat{\mathbf{z}}_{q,u} + \boldsymbol{\beta}$	$\mathbb{R}^{768}$
$W_{\text{UGI}}^{(1)}$, $\mathbf{b}_{\text{UGI}}^{(1)}$	UGI MLP layer-1 parameters ($[\tilde{\mathbf{z}}_{q,u}; \mathbf{e}_{d_q}] \in \mathbb{R}^{1536} \to 768$)	$\mathbb{R}^{768\times 1536}$, $\mathbb{R}^{768}$
$W_{\text{UGI}}^{(2)}$, $\mathbf{b}_{\text{UGI}}^{(2)}$	UGI MLP layer-2 parameters ($768 \to 768$)	$\mathbb{R}^{768\times 768}$, $\mathbb{R}^{768}$
$\mathbf{c}_{\text{UGI}}$	UGI control vector, $\mathbf{c}_{\text{UGI}} = \text{MLP}_{\text{UGI}}([\tilde{\mathbf{z}}_{q,u}; \mathbf{e}_{d_q}])$	$\mathbb{R}^{768}$
	Soft-Token Injection	
W_p, $\mathbf{b}_p$	Control-to-token projection in $\mathbf{p} = W_p\mathbf{c} + \mathbf{b}_p$	$\mathbb{R}^{768\times 768}$, $\mathbb{R}^{768}$
$\mathbf{p}$	Soft context token prepended to decoder input	$\mathbb{R}^{768}$
$\mathbf{H}^{(0)}$	Decoder input states after prepending $\mathbf{p}$ (output length T)	$\mathbb{R}^{(T+1)\times 768}$

For each bootstrap sample b, we compute the paired metric difference

$$\Delta_b = M(\texttt{PerTIDE}; S_b) - M(A; S_b).$$

The reported score for each model is the average over the paired bootstrap evaluations,

$$M(A) = \frac{1}{B}\sum_{b=1}^{B} M(A; S_b),$$

Table 9: **Runtime and deployment costs for PerTIDE.** We report end-to-end parameter counts, memory footprint, and serving costs under cached user-specific tri-trace states. The action-on-item encoder updates online without replaying full histories, while prompted LLM baselines require full prompt-history conditioning at inference; All numbers correspond to d=768, $|\mathcal{T}|$=4, and a frozen DistillGPT decoder when applicable.

Metric	PerTIDE (Ours)	Best Single-Scale Encoder	Prompted LLM (ICL)
A. Parameter and Memory Footprint			
Trainable parameters (encoder only)	∼154.36M	∼130–145M	0
Trainable parameters (encoder + context)	156.13M (UFI) / 158.49M (UGI)	∼130–145M	0
Frozen parameters at inference	82M (DistillGPT)	82M (if used)	≥7B–14B
Deployable parameters (total)	238–241M	212–227M	≥7B–14B
Per-user cached state	3×768 floats (≈9 KB)	history replay or GRU state	0
Peak VRAM (predictive inference)	∼1.1–1.3 GB	∼1.0–1.2 GB	≥18 GB
Peak VRAM (generative inference)	∼1.6–1.9 GB	∼1.5–1.8 GB	≥28 GB
B. History Dependence and Online Updates			
History dependence at inference	cached state only (no replay)	windowed / full replay	full prompt history
Online update cost per interaction	$\mathcal{O}(d^2)$ (single SSM step)	$\mathcal{O}(d^2)$	–
State refresh frequency	every timestep	every timestep	–
C. Predictive Serving Cost (Rec., Speaker ID)			
Latency per sample (end-to-end)	∼2–6 ms	∼2–5 ms	∼20–60 ms
Throughput (samples/sec)	∼160–420	∼180–450	∼20–60
FLOPs per sample	$\sim 3.8\times10^{9}$	$\sim 2.5\text{–}3.2\times10^{9}$	$\geq 1\times10^{11}$
D. Generative Serving Cost (Dialogue / Summarization)			
Contextualization	UFI / UGI	concat / none	ICL prompt
User-conditioned tokens injected	1 soft token	0–1	130–250
Prompt overhead (tokens)	20–40	20–40	130+
Generation length	task-dependent	task-dependent	task-dependent
End-to-end latency per sample	0.4–3 s	0.4–3 s	15–160 s
FLOPs per output (est.)	$\sim 6.1\times10^{10}$	$\sim 6.0\times10^{10}$	$\sim 1.6\times10^{12}$
E. Training Cost			
Training GPU-hours	∼22	∼18–30	–
LLM finetuning required	**No**	**No**	**No (ICL) / Yes (finetune)**

and the reported improvement is

$$\Delta = \frac{1}{B}\sum_{b=1}^{B}\Delta_b.$$

We test the one-sided null hypothesis that PerTIDE does not improve over the paired comparator:

$$H_0 : \mathbb{E}[\Delta] \leq 0, \qquad H_1 : \mathbb{E}[\Delta] > 0.$$

A comparison is counted as significant when the paired bootstrap distribution supports a positive effect at the $p < 0.05$ level. In practice, this requires the resampled paired differences to remain positive with no bootstrap sign reversal for the reported improvement; equivalently, the empirical bootstrap confidence interval of paired differences excludes zero on the non-positive side.

We apply this procedure to predictive metrics, including MRR, nDCG@5, nDCG@10, and HR@10, and to the generation and personalization metrics reported for PENS and OpenAI-Reddit, including PSE-JSD, PSE-SU4, PSE-METEOR, ROUGE, BLEU-4, BERTScore, and human-judgment scores where applicable. This procedure captures test-set sampling variability under fixed trained systems. It does not measure variability due to retraining from different random initializations, alternative train/test splits, or different negative-sampling pools. These factors are controlled by keeping the trained model, held-out split, autoregressive evaluation order, and candidate sets fixed within each paired comparison. Thus, the significance statements in the main text should be interpreted as paired test-set stability of the reported gains under the evaluation protocol, rather than as variance over independent full training runs.

J Detailed Results and Additional Analysis

J.1 Efficiency and Deployment Cost

We report runtime and deployment costs in Table 9. We observe that PerTIDE is heavier than the strongest single-scale encoder in trainable parameter count, but its serving behavior remains practical because the user-specific history is cached as tri-trace state rather than replayed from the full interaction prefix at inference time. The cached state consists of three 768-dimensional traces, which corresponds to roughly 9 KB per user. This makes the additional memory cost small compared

Table 10: **MovieLens movie recommendation under action-on-item preference flow.** We compare PerTIDE against task-native sequential recommenders on next-item ranking. Baselines expose profile, recurrent, or long-horizon history states, while PerTIDE uses schema-normalized action evidence with trace-factorized persistence.

Regime	Model	MRR	nDCG@10	HR@10
PAH	BPR-MF	6.50	6.12	12.81
MDH	GRU4Rec	12.76	16.42	29.01
	Caser	13.54	19.21	28.92
	Diff4Rec	12.02	15.72	22.38
LDH	S^3Rec	10.62	13.67	19.21
	SASRec	13.20	19.96	30.23
	BERT4Rec	14.41	20.22	31.71
	Mamba4Rec	14.56	19.83	31.24
	SSD4Rec	14.95	20.31	31.52
MTSH	PerTIDE w/o AC	15.21 ± 0.93	19.53 ± 1.12	28.16 ± 1.26
	PerTIDE w/o CR	17.05 ± 0.95	19.08 ± 1.43	27.85 ± 0.56
	PerTIDE	**17.31 ± 0.65**	**23.78 ± 0.84**	**33.20± 1.17**

Table 11: **Action-on-item encoder performance on MIND.** We compare task-native history encoders with PerTIDE under the shared event-schema setting. AC denotes action-conditioned evidence injection, and CR denotes command-conditioned readout. Results are reported on the full test set; $\mu \pm \sigma$ denotes mean and standard deviation over three evaluations on uniformly random 25% test-set subsamples.

Regime	Model	MRR	nDCG@5	HR@10
PAH	NAML	32.75	35.66	41.40
	PLM-NR	35.39	38.71	44.38
	Mean Pooling-B (ours)	29.43	31.22	33.17
	Max Pooling-B (ours)	28.32	31.16	32.41
MDH	EBNR	31.26	32.18	39.04
	GRU-B (ours)	32.64	35.17	37.92
LDH	NRMS	33.25	36.28	41.98
	MINER	36.60	40.20	46.30
	Fastformer+PLM-NR	35.91	39.16	45.03
	D-FM Attention (ours)	32.17	34.56	36.21
	Discretized SSM (ours)	35.14	38.53	37.27
MTSH	PerTIDE w/o AC (ours)	*40.13 ± 2.43*	*43.75± 1.37*	*46.35± 1.16*
	PerTIDE w/o CR (ours)	*42.51± 2.16*	*46.44± 1.75*	*51.06± 1.56*
	PerTIDE (ours)	**46.65± 2.24**	**49.21± 1.64**	**55.48± 1.14**

with full prompted-history conditioning, where the system must repeatedly place long user histories into the input context.

We also observe that the main deployment contrast is not between PerTIDE and a single-scale encoder, but between learned preference-flow state reuse and prompted LLM in-context personalization. Prompted LLM baselines have no task-specific trainable parameters under in-context learning, but they carry substantially larger frozen inference models, with at least 7B–32B parameters. In contrast, PerTIDE uses a substantially smaller deployable model with a frozen DistillGPT decoder where generation is required. We find that this strengthens the practical motivation for the architecture: PerTIDE pays a moderate encoder-side cost to maintain a reusable user state, while prompted LLM personalization pays repeated inference-time cost through long-history prompting.

We further see that the end-to-end latency and FLOP estimates favor cached preference-flow inference over prompted LLM personalization. PerTIDE and the best single-scale encoder remain in the same broad latency band, while prompted LLM baselines are much slower because they must condition on prompt histories and run much larger models. This further establishes that PerTIDE is not only a performance-oriented proposal, but also a deployable alternative to repeatedly prompting large models with user histories.

Table 12: **Structured news prediction on PENS-Structured.** We report same-user next-news ranking under the action-on-item schema. Baselines are grouped by the exposed history-state assumption in Section 3. All values are on a 1–100 scale.

Regime	Model	MRR	nDCG@5	nDCG@10	HR@10
MDH	EBNR	2.65	1.82	2.45	4.87
PAH	NAML	1.29	0.43	0.81	2.01
LDH	NRMS	1.18	0.39	0.74	1.92
LDH	TrRMIo	8.02	7.85	8.53	12.77
LDH	LSTUR	9.04	8.21	10.16	12.93
LDH	MINER	5.17	4.71	5.61	9.77
LLM	DeepSeek-32B	1.31	0.28	0.64	1.53
LLM	Qwen2.5-32B	1.24	0.26	0.59	1.47
MTSH	PerTIDE	**9.94**	**9.88**	**11.58**	**18.32**

Table 13: **Trace-subset ablation for schema-normalized evidence persistence.** We ablate long-term (L), short-term (S), and episodic (E) persistence channels individually and in combination. Recommendation results are reported on MIND and MovieLens-1M, and personalized summarization is reported on PENS. Best results are bolded.

Variant	MIND Recommendation			ML-1M Recommendation			PENS Summarization		
	MRR	nDCG@5	HR@10	MRR	nDCG@10	HR@10	PSE-JSD	PSE-SU4	PSE-METEOR
L only	40.41	43.75	46.13	12.31	16.15	20.34	0.231	0.095	0.107
S only	37.65	41.32	44.24	12.47	16.62	20.81	0.238	0.106	0.111
E only	24.18	27.73	31.23	10.18	13.17	18.54	0.076	0.042	0.045
L + S	41.26	43.95	46.72	14.17	18.44	22.11	0.264	0.143	0.182
L + E	40.45	43.82	46.44	14.03	18.07	22.74	0.246	0.103	0.108
S + E	38.26	42.34	45.07	13.43	17.18	21.67	0.093	0.062	0.077
L + S + E	**46.65**	**49.21**	**55.48**	**17.31**	**23.78**	**33.20**	**0.366**	**0.341**	**0.355**

J.2 Full Predictive Personalization Results

J.2.1 MovieLens-1M Recommendation

We report the full MovieLens-1M ranking results in Table 10. We observe a clear progression from static profile modeling to sequential modeling and then to explicit action-on-item preference flow. The PAH baseline BPR-MF gives weak performance, with 6.50 MRR and 12.81 HR@10, confirming that an order-insensitive persistent profile is insufficient for next-movie recommendation. MDH models improve substantially: GRU4Rec reaches 12.76 MRR and Caser reaches 13.54 MRR, showing that local sequential drift already captures useful user-history structure.

We see that LDH baselines further improve the ranking quality. BERT4Rec, Mamba4Rec, and SSD4Rec are the strongest non-PerTIDE models, with SSD4Rec reaching 14.95 MRR, 20.31 nDCG@10, and 31.52 HR@10. This confirms that long-horizon sequence modeling is beneficial on MovieLens. However, PerTIDE reaches 17.31 MRR, 23.78 nDCG@10, and 33.20 HR@10, improving over the strongest LDH baseline by $+2.36$ MRR, $+3.47$ nDCG@10, and $+1.68$ HR@10. This strengthens the main claim that the gain is not only due to using a stronger sequential encoder, but due to routing action-conditioned evidence through trace-factorized persistence before task readout.

We also find that the ablations clarify different failure modes. PerTIDE without action conditioning still gives 15.21 MRR, but its HR@10 drops to 28.16, below several strong baselines. This suggests that trace persistence alone can sharpen some ranking positions, but it does not reliably preserve top-K retrieval coverage when positive and negative action roles are not separated. PerTIDE without task tuning gives 17.05 MRR, close to the full model, but only 19.08 nDCG@10 and 27.85 HR@10. We therefore see that task-conditioned readout is especially important for distributing probability mass across the useful top-ranked region, not only for improving the first relevant rank. The full model is strongest across all three metrics, which confirms that action-conditioned entry, temporal persistence, and command/task-conditioned readout are complementary rather than interchangeable.

Table 14: **Command-driven personalized headline generation on PENS.** AC denotes action-conditioned evidence injection, CR denotes command-conditioned readout, UFI denotes user fused injection, and Full denotes PerTIDE encoder with UGI decoder. Values are reported on the full test set; $\mu \pm \sigma$ over three evaluations on uniformly random 25% test-set subsamples.

Category	Model	PSE-JSD	PSE-SU4	PSE-METEOR
LLMs (2-shot)	LLaMA-13B	0.227	0.078	0.081
	DeepSeek-14B	0.248	0.094	0.097
	Gemini-2.5-Flash	0.222	0.104	0.124
	DeepSeek-R1-32B	0.263	0.125	0.174
	Qwen-2.5-32B	0.162	0.103	0.115
Prompt-Chaining	Mistral-7B	0.072	0.026	0.023
	DeepSeek-14B	0.078	0.028	0.024
PAH	Signature-Phrase	0.017	0.015	0.016
	PENS-NAML-T1	0.021	0.014	0.016
	MaxPool-B	0.008	0.005	0.005
MDH	PENS-EBNR-T1	0.015	0.01	0.011
	PENS-EBNR-T2	0.011	0.008	0.009
	GRU-B	0.021	0.018	0.019
LDH	GTP	0.024	0.017	0.019
	Discretized SSM	0.037	0.032	0.031
	D-FM Attention-B	0.028	0.019	0.025
	PENS-NRMS-T1	0.015	0.011	0.011
	PENS-NRMS-T2	0.008	0.007	0.007
MTSH	PerTIDE w/o AC	0.314 ± 0.023	0.271 ± 0.007	0.292 ± 0.006
	PerTIDE w/o CR	0.317 ± 0.024	0.276 ± 0.005	0.284 ± 0.008
	PerTIDE (UFI)	0.332 ± 0.019	0.289 ± 0.009	0.303 ± 0.003
	PerTIDE (Full; UGI)	$\mathbf{0.366 \pm 0.011}$	$\mathbf{0.341 \pm 0.002}$	$\mathbf{0.355 \pm 0.005}$

Table 15: **Content fidelity and human judgment for command-driven headline generation.** We report content-overlap, semantic similarity, and human-judgment scores to verify that gains in personalization-sensitive PSE metrics do not come from weaker content fidelity.

Category	Model	RG-SU4	BLEU-4	BScore	HJ
Specialized (Personalized)	PENS-NRMS-T2	13.64	4.48	86.13	2.95
	GTP-TrRMIo	21.91	10.31	88.53	2.44
	SP-Individual	19.54	8.90	86.61	2.86
LLMs (2-shot history)	LLaMA-13B	18.31	11.85	88.76	3.05
	DeepSeek-14B	19.57	12.68	89.43	3.03
	DeepSeek-32B	*29.42*	*19.31*	**90.87**	*3.12*
	Qwen-2.5-32B	19.22	14.31	89.17	3.04
MTSH	PerTIDE-Full	**41.32**	**27.64**	*90.84*	**3.21**

J.2.2 MIND News Recommendation

We report the full MIND news recommendation results in Table 11. We observe that MIND gives the clearest mechanism ladder because it contains strong PAH, MDH, and LDH baselines. Within PAH, PLM-NR improves over NAML, reaching 35.39 MRR and 44.38 HR@10. This confirms that stronger news representations help profile aggregation, but the resulting state remains order-insensitive. Within MDH, EBNR and GRU-B remain weaker than PLM-NR on MRR, which suggests that a single recurrent state is not enough to capture the mixture of stable, recent, and episodic news interests.

We see that LDH baselines provide the strongest task-native alternatives. MINER reaches 36.60 MRR, 40.20 nDCG@5, and 46.30 HR@10, while FastFormer+PLM-NR reaches 35.91 MRR and 45.03 HR@10. These results confirm that full-prefix and multi-interest modeling are valuable for news recommendation. However, PerTIDE reaches 46.65 MRR, 49.21 nDCG@5, and 55.48 HR@10. We therefore observe gains of $+10.05$ MRR, $+9.01$ nDCG@5, and $+9.18$ HR@10 over MINER. This further establishes that the advantage is not merely a better long-history encoder; the advantage appears when the model explicitly distinguishes action-role entry, trace-factorized persistence, and command-conditioned consumption.

We find that the ablation ladder is also consistent on MIND. PerTIDE without action conditioning reaches 40.13 MRR, improving over MINER but losing 6.52 MRR relative to the full model. PerTIDE without command-conditioned readout reaches 42.51 MRR, improving over the no-AC variant but still losing 4.14 MRR relative to the full model. This confirms that action conditioning is more critical for correctly injecting click/ignore evidence into the state, while command-conditioned readout is needed to consume that state in a task-specific way. The full model gives the best result because both sides of the schema are used: the action role controls how evidence enters, and the command controls how the resulting preference state is read.

Table 16: **Eventized TL;DR personalization on OpenAI-Reddit.** We treat OpenAI-Reddit as a schema-level transfer stress test under surrogate trajectory construction, not as evidence of unconstrained cross-domain user-history transfer.

Category	Model	PSE-JSD	PSE-SU4	PSE-METEOR
LLMs (2-shot)	LLaMA-13B	0.232	0.093	0.107
	Zephyr-7B	0.214	0.087	0.104
	Mistral-7B	0.226	0.088	0.103
	DeepSeek-14B	*0.243*	*0.095*	*0.109*
MTSH	PerTIDE-Full	**0.285**	**0.261**	**0.294**

Table 17: **Robustness under sparse positive-action evidence.** We evaluate each model on a click-only sparse subset and report the original score (left) and sparse-action score (right), with the relative drop in parentheses. Smaller drops indicate higher robustness when positive evidence is limited in the action-on-item trajectory.

Model	PSE-JSD	PSE-SU4	PSE-METEOR
DeepSeek-R1-14B (2-shot)	$0.263 \rightarrow 0.152$ (−42.8%)	$0.125 \rightarrow 0.087$ (−30.4%)	$0.174 \rightarrow 0.092$ (−47.2%)
Gemini-2.5-Flash (2-shot)	$0.222 \rightarrow 0.122$ (−45.0%)	$0.104 \rightarrow 0.061$ (−41.3%)	$0.124 \rightarrow 0.070$ (−43.5%)
Best Baseline (GTP)	$0.024 \rightarrow 0.016$ (−33.3%)	$0.017 \rightarrow 0.009$ (−47.1%)	$0.019 \rightarrow 0.011$ (−42.1%)
PerTIDE	$0.366 \rightarrow 0.211$ (−42.35%)	$0.341 \rightarrow 0.244$ (−28.45%)	$0.355 \rightarrow 0.262$ (−26.2%)

J.2.3 PENS-Structured Next-News Prediction

We report same-user next-news prediction on PENS-Structured in Table 12. We observe that this setting is considerably harder than MIND because many standard news recommendation baselines collapse under the same-user structured prediction protocol. NAML, NRMS, and the prompted LLM rankers remain close to the bottom of the table, with MRR values around 1–2. This indicates that generic content matching or prompted ranking is not enough when the task requires recovering a user-specific next-news trajectory.

We find that LSTUR and TrRMIo are the strongest non-MTSH baselines, reaching 9.04 and 8.02 MRR respectively. This suggests that long/short user-history modeling is valuable in PENS-Structured, especially because the same-user setup requires more than one-shot article relevance. PerTIDE reaches 9.94 MRR, 9.88 nDCG@5, 11.58 nDCG@10, and 18.32 HR@10. The improvement in HR@10 is especially important: PerTIDE improves over LSTUR by $+5.39$ HR@10. We therefore see that the action-on-item state does not only improve the top reciprocal rank, but also recovers more relevant candidates within the actionable top-10 region.

This confirms the same pattern seen on MovieLens and MIND, but under a more difficult news-side setting. PerTIDE is not only benefiting from dataset-specific recommendation structure; it remains useful when PENS click/skip histories are rewritten into the shared schema and evaluated as same-user next-news trajectories.

J.3 Trace-Subset Ablation on Natural Tasks

We report trace-subset ablations in Table 13. Unlike the controlled L-Tr/S-Tr/E-Tr diagnostic in Section 6, this table evaluates the trace combinations on natural downstream tasks: MIND recommendation, MovieLens-1M recommendation, and PENS summarization. We observe that the full $L + S + E$ model is strongest across all reported tasks and metrics. On MIND, it reaches 46.65 MRR, 49.21 nDCG@5, and 55.48 HR@10. On MovieLens-1M, it reaches 17.31 MRR, 23.78 nDCG@10, and 33.20 HR@10. On PENS summarization, it reaches 0.366 PSE-JSD, 0.341 PSE-SU4, and 0.355 PSE-METEOR.

We find that single traces are useful but incomplete. The long-term trace is the strongest single trace on MIND, reaching 40.41 MRR, which is consistent with news recommendation requiring persistent topical preference mass. The short-term trace is slightly stronger than the long-term trace on MovieLens MRR, reaching 12.47 versus 12.31, suggesting that recent movie interactions can be highly informative for next-item ranking. However, neither single-trace variant approaches the full model. This confirms that natural user histories contain mixed evidence types rather than only stable or only recent interests.

We also observe that the episodic-only trace is weak by itself, especially on MIND and PENS. It reaches only 24.18 MRR on MIND and 0.076 PSE-JSD on PENS. This should not be interpreted as episodic memory being unimportant. Instead, we find that episodic evidence is complementary: $L+E$ and $L+S+E$ improve over isolated trace behavior, and the full model is strongest everywhere. This strengthens the interpretation that episodic traces should not replace stable and recency-sensitive traces; they should provide finite-window event imprints that become useful when fused with longer-term preference evidence.

We see the clearest complementarity in PENS summarization. The $L + S$ model reaches 0.264 PSE-JSD and 0.182 PSE-METEOR, while the full $L + S + E$ model reaches 0.366 PSE-JSD and 0.355 PSE-METEOR. This large gain confirms that command-driven generation benefits from retaining finite-window evidence in addition to stable and recent interests. Thus, Table 13 strengthens the main trace-specialization result by showing that the fused trace state remains necessary on natural tasks, not only on controlled trajectory probes.

J.4 Command-Driven Generation and Content Fidelity

J.4.1 PENS Personalized Headline Generation

We report command-driven personalized headline generation on PENS in Table 14. We observe that prompted LLMs are substantially stronger than older specialized personalized encoders on personalization-sensitive PSE metrics. Among the two-shot LLMs, `DeepSeek-R1-32B` reaches 0.263 PSE-JSD, 0.125 PSE-SU4, and 0.174 PSE-METEOR. In contrast, PAH, MDH, and LDH personalized headline baselines remain much lower, with `Discretized SSM` reaching only 0.037 PSE-JSD and 0.031 PSE-METEOR. This suggests that large language models can use prompted user history more effectively than earlier compact personalized generation systems.

However, we find that `PerTIDE` variants are consistently stronger than both the prompted LLMs and the specialized history encoders. `PerTIDE` without action conditioning already reaches 0.314 PSE-JSD and 0.292 PSE-METEOR, while `PerTIDE` without command-conditioned readout reaches 0.317 PSE-JSD and 0.284 PSE-METEOR. This confirms that the learned preference-flow state transfers to generation even when one schema component is removed. We also observe that decoder contextualization matters: UFI reaches 0.332 PSE-JSD, 0.289 PSE-SU4, and 0.303 PSE-METEOR, while the full UGI model reaches 0.366 PSE-JSD, 0.341 PSE-SU4, and 0.355 PSE-METEOR.

We see that UGI improves over UFI by $+0.034$ PSE-JSD, $+0.052$ PSE-SU4, and $+0.052$ PSE-METEOR. This confirms that simply injecting a fused user-query vector is weaker than explicitly gating the query-conditioned representation by the user-specific task-ready state. In other words, generation benefits when the decoder receives not only user information, but a command-conditioned signal that can emphasize or suppress query semantics according to the user trajectory.

J.4.2 Content Fidelity and Human Judgment

We report content fidelity and human-judgment scores in Table 15. This is important because personalization-sensitive gains are only meaningful if the generated headline remains faithful to the query article. We observe that `PerTIDE`-Full reaches the highest RG-SU4 score, 41.32, compared with 29.42 for `DeepSeek-32B` and 21.91 for `GTP`. We also see that `PerTIDE`-Full reaches the highest BLEU-4 score, 27.64, compared with 19.31 for `DeepSeek-32B`. Thus, the PSE improvements are not obtained by sacrificing lexical or content overlap.

We find that BERTScore is essentially tied between `PerTIDE`-Full and DeepSeek-32B: `PerTIDE` obtains 90.84, while `DeepSeek-32B` obtains 90.87. This is useful because it shows that `PerTIDE` improves personalization-sensitive and content-overlap metrics while maintaining semantic similarity at the level of the strongest two-shot LLM baseline. We also observe that `PerTIDE`-Full obtains the highest human-judgment score, 3.21, exceeding `DeepSeek-32B` at 3.12 and all specialized personalized systems. This strengthens the claim that command-driven preference-flow generation improves user alignment without degrading content fidelity.

Overall, this result confirms an important reviewer-facing point: `PerTIDE` is not merely optimizing a personalization-sensitive metric in isolation. We observe gains in PSE metrics, stronger content-overlap scores, competitive semantic similarity, and the best human-judgment score. This further establishes that the generated outputs remain both personalized and faithful.

J.5 Transfer and Sparse-Evidence Stress Tests

J.5.1 Eventized TL;DR Personalization on OpenAI-Reddit

We report OpenAI-Reddit TL;DR personalization in Table 16. We treat this setting as a schema-level stress test rather than direct evidence of unrestricted cross-domain transfer. OpenAI-Reddit does not provide the same kind of clean user trajectory as PENS, so the evaluation depends on surrogate trajectory construction from feedback-derived positive and negative evidence. For this reason, the result should be read as evidence that `PerTIDE` remains usable when summarization feedback can be eventized, not as evidence that arbitrary summarization datasets automatically fit the schema.

We observe that two-shot LLM baselines are relatively close on PSE-JSD, with `DeepSeek-14B` reaching 0.243, `LLaMA-13B` reaching 0.232, and `Mistral-7B` reaching 0.226. `PerTIDE`-Full improves to 0.285, giving a moderate but consistent gain in PSE-JSD. The larger gains appear in PSE-SU4 and PSE-METEOR. `PerTIDE`-Full reaches 0.261 PSE-SU4 and 0.294 PSE-METEOR, compared with 0.095 and 0.109 for `DeepSeek-14B`. We therefore find that the learned preference-flow state provides stronger personalized content alignment than prompted-history LLMs under this eventized TL;DR protocol.

This strengthens the schema-level claim in a bounded way. We see that `PerTIDE` can reuse the action-on-item abstraction beyond PENS headline generation when feedback and summarization commands can be mapped into positive/negative evidence and command-like target loci. At the same time, this result should remain framed as an eventized stress test because the original OpenAI-Reddit data does not provide stable user histories in the same direct form as PENS.

J.5.2 Sparse Positive-Action Robustness

We report sparse positive-action robustness in Table 17. We evaluate each model on a click-only sparse subset and compare the original score with the sparse-action score. We observe that all systems degrade when positive evidence is limited, which is expected because personalization becomes harder when the trajectory contains less reinforcing signal. The relevant question is therefore not whether the scores drop, but which metrics remain stable under sparse evidence.

We find that `PerTIDE` has a comparable PSE-JSD drop to the stronger LLM baselines, moving from 0.366 to 0.211, a relative drop of 42.35%. `DeepSeek-R1-14B` drops from 0.263 to 0.152, a relative drop of 42.8%, and `Gemini-2.5-Flash` drops by 45.0%. Thus, `PerTIDE` is not uniquely robust on PSE-JSD. However, we observe that `PerTIDE` is substantially more robust on PSE-SU4 and PSE-METEOR. It drops by 28.45% on PSE-SU4 and 26.2% on PSE-METEOR, while `DeepSeek-R1-14B` drops by 30.4% and 47.2%, and Gemini drops by 41.3% and 43.5%.

We also see that the strongest specialized baseline is highly unstable on PSE-SU4, dropping from 0.170 to 0.009. This suggests that compact personalized generation baselines may rely heavily on explicit positive evidence and fail when that evidence becomes sparse. `PerTIDE` preserves stronger sparse-action performance because the user state is not a flat profile over positive clicks; it is built from action-conditioned evidence and retained through complementary traces. This confirms that the proposed schema is useful not only in dense-history settings, but also when positive evidence is partially missing.

J.5.3 Schema-Proximity and Reuse Diagnostic

We examine whether schema-trajectory distance is associated with cross-domain reuse on 500 MovieLens trajectories. Table 18 reports the correlations between schema distance and target risk, and between schema distance and reciprocal-rank gain. Confidence intervals are clustered at the user level.

J.6 Broader Discussion on Dialogue-based Discourse Modeling

We use MPChat as a boundary case for the action-on-item schema because dialogue eventization is less direct than click, skip, rating, or request logs. MPChat contains multi-turn dialogue, speaker roles, persona statements, images, entity mentions, and multimodal grounding cues. As a result, UIG construction must infer which dialogue turns act as target loci, which turns express positive or negative stance, which speaker role is active, and which persona or image evidence should be

Table 18: **Schema proximity and cross-domain reuse.** Correlation between schema-trajectory distance and frozen/adapted target behavior on 500 MovieLens trajectories. Confidence intervals are user-clustered.

Association	**Frozen**	**Adapted**
Schema distance vs. target risk	0.109	0.160
Schema distance vs. reciprocal-rank gain	-0.089	-0.114

Table 19: **MPChat broader-applicability stress test: response prediction.** We report response prediction under dialogue eventization. These results are not primary evidence for MTSH; they assess how far the action-on-item schema extends when trajectories contain discourse, persona, and multimodal grounding signals; Results are reported on the full test set; $\mu \pm \sigma$ denotes mean and standard deviation over three evaluations on uniformly random 25% test-set subsamples (same subsamples across methods).

Model	**R@1**	**MRR**
SBERT (zero-shot)	35.67	45.75
SBERT	51.32 ± 1.32	64.76 ± 0.92
SBERT+ViT (c)	57.70 ± 0.71	69.39 ± 0.40
SBERT+ViT (c, p^i)	58.55 ± 0.70	70.17 ± 0.45
SBERT+ViT (c, p^t)	64.32 ± 0.64	74.31 ± 0.45
SBERT+ViT (c, p^{i+t}, Full)	65.29 ± 0.66	75.08 ± 0.43
SBERT+CLIP (c)	59.68 ± 0.70	70.99 ± 0.49
SBERT+CLIP (c, p^i)	60.30 ± 0.50	71.47 ± 0.27
SBERT+CLIP (c, p^t)	64.32 ± 0.75	74.33 ± 0.57
SBERT+CLIP (c, p^{i+t}, Full)	65.43 ± 0.42	75.19 ± 0.32
CLIP+CLIP (c^i)	40.85 ± 0.64	54.32 ± 0.30
CLIP+CLIP (c)	69.11 ± 0.74	78.22 ± 0.49
CLIP+CLIP (c, p^i)	69.87 ± 0.40	78.85 ± 0.27
CLIP+CLIP (c, p^t)	72.13 ± 0.61	80.72 ± 0.38
CLIP+CLIP (c, p^{i+t}, Full)	72.65 ± 0.38	81.12 ± 0.26
DeepSeek-R1-32B (c, p^t)	41.45	54.42
Qwen2.5-32B (c, p^t)	41.32	52.17
`PerTIDE`-Full (c^t) (ours)	45.64 ± 0.45	54.25 ± 0.67

linked to a response. This makes MPChat useful for testing broader applicability under imperfect eventization, although we do not treat it as primary evidence for MTSH.

Our MPChat analysis asks how much headroom comes from two signals that `PerTIDE` does not explicitly consume: (i) enriching dialogue-turn context with image or multimodal channels (c, c^i, c^t), and (ii) injecting explicit persona constructed from user history (p^i, p^t, p^{i+t}, $\mathcal{P}_c^i$, $\mathcal{P}_c^t$, $\mathcal{P}_c^{i+t}$). These comparisons are therefore not equivalent comparisons against `PerTIDE`. Instead, they estimate upper-bound gains from explicit persona supervision and image-grounded context, and help interpret why `PerTIDE` remains competitive under a stricter text-context regime.

Response prediction. We report MPChat response-prediction results in Table 19. The main pattern is that image-only context is a weak substitute for turn-level discourse. `CLIP+CLIP` drops from $69.11/78.22$ with multimodal context c to $40.85/54.32$ with image-only context c^i on R@1/MRR. This large gap indicates that image-only context does not replace dialogue discourse. Once text-grounded context is available, persona information gives additional headroom. For `CLIP+CLIP`, adding text persona improves $69.11/78.22$ with c to $72.13/80.72$ with (c, p^t), and full multimodal persona further improves it to $72.65/81.12$ with (c, p^{i+t}). Similarly, `SBERT+ViT` improves from $57.70/69.39$ with c to $64.32/74.31$ with (c, p^t), with a smaller additional gain from (c, p^{i+t}). Thus, explicit persona provides a measurable upper bound, mostly as a refinement over text-grounded context rather than as a replacement for it.

In contrast, `PerTIDE` uses the stricter text-context-only setting without explicit persona or image-channel injection and reaches $45.64/54.25$. We therefore interpret the gap between `PerTIDE` and the strongest persona-injected multimodal configuration, such as `CLIP+CLIP` with (c, p^{i+t}), as the upper-bound benefit of explicit persona injection and multimodal cues. It is not direct evidence against action-on-item preference flow, since those cues remain outside the present `PerTIDE` input schema.

Table 20: **MPChat broader-applicability stress test: speaker identification.** We report speaker identification under dialogue eventization and compare against persona- and multimodal-context upper-bound systems; c: dialogue context, c^t: text-only context, r: response, $\mathcal{P}_c$: candidate set with profile information

Category	Model	R@1	MRR
Text Only	SBERT $(c^t, r, \mathcal{P}_c^t)$	56.47 ± 0.58	67.92 ± 0.52
SBERT+ViT	$c, r, \mathcal{P}_c^i$	19.56 ± 0.64	35.84 ± 0.45
	$c, r, \mathcal{P}_c^t$	56.87 ± 0.60	68.33 ± 0.37
	$c, r, \mathcal{P}_c^{i+t}$ (Full)	57.28 ± 0.44	68.86 ± 0.33
SBERT+CLIP	$c, r, \mathcal{P}_c^i$	25.71 ± 0.49	42.47 ± 0.34
	$c, r, \mathcal{P}_c^t$	56.63 ± 0.66	68.15 ± 0.42
	$c, r, \mathcal{P}_c^{i+t}$ (Full)	57.24 ± 0.63	68.69 ± 0.39
CLIP+CLIP	$c, r, \mathcal{P}_c^i$	44.27 ± 0.66	59.04 ± 0.35
	$c, r, \mathcal{P}_c^t$	59.89 ± 0.71	70.87 ± 0.53
	$c, r, \mathcal{P}_c^{i+t}$ (Full)	62.17 ± 0.56	73.08 ± 0.35
LLMs (2-shot)	DeepSeek-32B $(c, r, \mathcal{P}_c^t)$	48.13	54.37
	Qwen-2.5-32B $(c, r, \mathcal{P}_c^t)$	45.31	51.62
	DeepSeek-32B (c, r)	28.56	31.43
	Qwen-2.5-32B (c, r)	25.61	29.67
MTSH (ours)	`PerTIDE`-Full (c^t, r)	41.77 ± 0.82	47.19 ± 1.30

Speaker identification. We report MPChat speaker-identification results in Table 20. The same pattern appears more sharply. Image-only persona can produce severe failure modes under SBERT-based hybrids, as `SBERT+ViT` with $(c, r, \mathcal{P}_c^i)$ yields only $19.56/35.84$. Text persona recovers strong performance, with $(c, r, \mathcal{P}_c^t)$ reaching $56.87/68.33$. For `CLIP+CLIP`, multimodal persona adds further gains on top of text persona, improving from $59.89/70.87$ with $(c, r, \mathcal{P}_c^t)$ to $62.17/73.08$ with $(c, r, \mathcal{P}_c^{i+t})$. This supports the same conclusion: the dominant requirement is a text-grounded discourse signal, while multimodal persona gives additional gains once the text channel is already strong.

These results clarify why `PerTIDE` remains meaningful without explicit persona or image context. Most of the attainable accuracy in dialogue-turn inference comes from text-grounded discourse context. Persona injection gives secondary headroom, but it is not required to obtain a competitive base representation. `PerTIDE` therefore concentrates modeling capacity on encoder-side temporal abstraction over text-grounded interaction histories. Missing persona and multimodal links increase residual schema distance because the eventized trajectory no longer carries all response-relevant evidence for speaker-role and response grounding.

Dialogue generation. We report MPChat dialogue-generation results in Table 21. These results show that the same encoder and decoder-conditioning route can operate under inferred action roles. `PerTIDE` improves over the strongest non-MTSH dialogue-generation baseline, `Discretized SSM`, by $+8.48/+7.08/+9.09/+3.52$ on RG-2/RG-L/BLEU-4/BScore. We treat this result as supportive but bounded evidence. Dialogue generation involves discourse continuation, speaker intent, and persona consistency, which are only partially captured by the present UIG construction.

Connection to schema-level generalizability. The MPChat results match the formal interpretation in Appendix B.2. Schema invariance explains why native dialogue actions can be reused only after they are mapped into shared positive, negative, or command-like roles. Embedding stability explains why missing persona and multimodal grounding can hurt: target-locus embeddings no longer carry all response-relevant evidence. The schema-level transfer bound states that target-domain error depends on schema-trajectory distance and command-semantics discrepancy, while the common-substrate training bound states that pooled schema training is useful only when schema rewriting makes trajectory distributions comparable.

Thus, MPChat acts as an empirical boundary case. `PerTIDE` remains usable under imperfect dialogue eventization, and the dialogue-generation gains show that inferred action roles can support generation. At the same time, the gap to persona- and multimodal-aware systems is expected when discourse roles, speaker identity, visual grounding, and persona links remain outside the current schema. Stronger dialogue personalization therefore requires discourse-aware UIG construction with dialogue acts, speaker-role transitions, entity mentions, topic-shift markers, acceptance/rejection signals, and confidence-weighted persona links. In this sense, `PerTIDE` provides schema-level gener-

Table 21: **MPChat broader-applicability stress test: dialogue generation.** AC denotes action-conditioned evidence injection, CR denotes command-conditioned readout, UFI denotes user fused injection, and Full denotes PerTIDE encoder with UGI decoder. These results test dialogue eventization under imperfect preference logs rather than serving as primary evidence for schema-level generalizability; $\mu \pm \sigma$ denotes mean and standard deviation over three evaluations on uniformly random test-set subsamples.

Category	Model	RG-2	RG-L	BLEU-4	BScore
LLMs (2-shot)	DeepSeek-32B	9.71	8.46	5.61	71.42
	Qwen-2.5-32B	9.43	8.21	4.75	68.83
	MS/Phi-Instruct	9.49	8.05	3.21	69.52
PAH	Meanpool-B	8.54 ± 1.33	6.31 ± 1.21	3.43 ± 2.15	62.18 ± 1.45
	Maxpool-B	7.21 ± 1.12	4.12 ± 1.43	2.55 ± 2.11	61.42 ± 1.65
MDH	GRU-B	11.11 ± 1.31	10.82 ± 1.43	4.17 ± 0.78	63.18 ± 2.43
LDH	D-FM Attention	8.25 ± 0.89	6.07 ± 1.32	3.18 ± 1.44	62.05 ± 2.78
	Discretized SSM	12.17 ± 1.32	11.44 ± 1.56	6.48 ± 1.73	74.83 ± 1.66
MTSH	PerTIDE w/o AC	14.38 ± 1.32	12.73 ± 1.42	8.33 ± 0.93	73.89 ± 1.65
	PerTIDE w/o CR	13.15 ± 1.77	10.91 ± 1.14	7.34 ± 1.55	71.64 ± 2.42
	PerTIDE (UFI)	18.22 ± 1.55	15.73 ± 1.21	11.58 ± 1.36	75.23 ± 2.51
	PerTIDE-Full	$\mathbf{20.65} \pm \mathbf{1.23}$	$\mathbf{18.52} \pm \mathbf{2.31}$	$\mathbf{15.57} \pm \mathbf{2.06}$	$\mathbf{78.35} \pm 1.55$

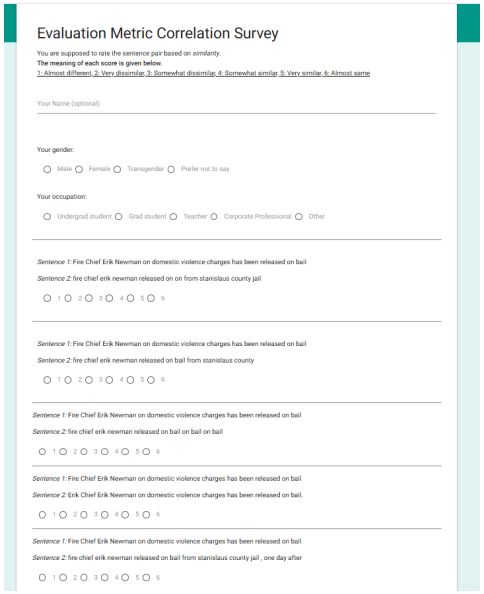


Figure 4: HJ-annotation survey format

.

alizability across eventizable personalization trajectories, not unrestricted transfer across arbitrary user-interaction logs.

K Human-judgment Evaluation (PENS)

To assess human-alignment of PerTIDE summaries, we conduct a survey-based evaluation in which each participant[2] is shown (i) the gold-reference summary and (ii) one model-generated summary for the same PENS news article. Model identities are hidden to mitigate brand and expectation bias, and participants rate reference–generation similarity on a 1–6 scale (1: low, 6: high). We evaluate PerTIDE against representative strong baselines spanning specialized personalized models (PENS-NRMS-T2, GTP-TrRMIo, SP-Individual) and 2-shot history-conditioned LLMs (DeepSeek-14B, DeepSeek-32B, Qwen-2.5-32B). Table 15 reports the average human-judgment (HJ) scores alongside automatic metrics (RG-SU4, BLEU-4, BScore): PerTIDE attains the highest HJ score (3.21), exceeding the best 2-shot LLM (DeepSeek-32B: 3.12) and all specialized baselines (e.g.,

[2]Demography: 65 male, 22 female graduate students from Computer Science, Humanities, Mathematics, & Natural Sciences.

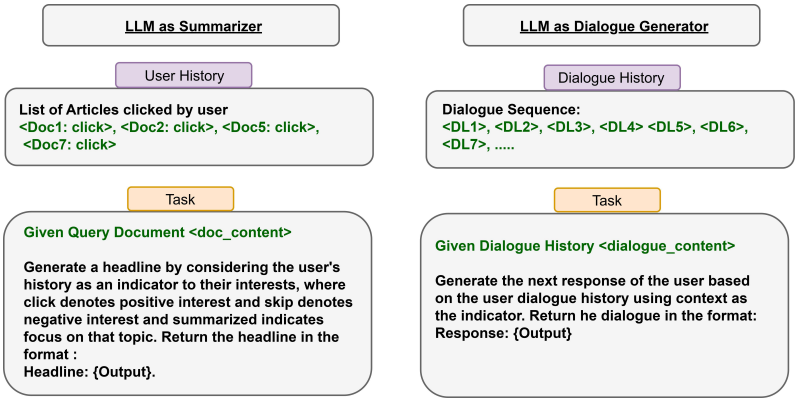


Figure 5: Prompt Templates for LLMs for (a)Personalized Summarization, (b)Dialogue Generation .

PENS-NRMS-T2: 2.95; GTP-TRRMIO: 2.44; SP-Individual: 2.86). Participation was voluntary, responses were de-identified before analysis, and the study involved only minimal-risk similarity judgments over news headlines. We collected only aggregate demographic information and did not collect identifiable participant information or private user histories. The study followed the applicable institutional approval, exemption, or equivalent review process under anonymized reporting.

L LLM Prompting Protocol

Overview. We use prompted LLMs for generative tasks using temperature 0.4 and nucleus sampling with `top_p` $= 0.9$ for all LLM runs.

L.1 LLM as Summarizers (Personalized Headline Generation)

Each instance is a conditional generation problem, where we prompt the LLMs using user-history as the context and a single query document body and instruct the LLM to generate one personalized headline. The instruction ties personalization to the history (clicked topics should be emphasized; skipped topics should be deprioritized) while requiring factual consistency with the query document. We constrain the output to a single field (`Headline: ...`) and ignore any additional text.

L.2 LLM as Dialogue Generators

We prompt the LLMs using the entire dialogue discourse as the context and instruct the LLM to generate one next response. The instruction ties personalization to factual consistency with the persona information and the discourse context. We constrain the output to a single field (`Response: ...`) and ignore any additional text.

M License and Usage Statement

Pre-trained large language models. We use all pre-trained models as inference-only baselines (no additional pretraining, finetuning, or weight modification), and we comply with the license/terms specified by the original providers. We use: (i) DeepSeek-R1 / DeepSeek-R1-Distill-Qwen2.5-14B/32B under the MIT License (`https://github.com/deepseek-ai/DeepSeek-R1/blob/main/LICENSE`); (ii) Mistral-7B-Instruct under the Apache License 2.0 (`https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2`, `https://www.apache.org/licenses/LICENSE-2.0`); (iii) LLaMA2-13B-Chat under the Llama 2 Community License Agreement (`https://ai.meta.com/llama/license/`); (iv) Zephyr-7B-β under the MIT License (`https://huggingface.co/HuggingFaceH4/zephyr-7b-beta`)

All models are used according to their respective licenses and terms provided by their original creators. Proper attribution is given to each model's developers as cited in our references.

Datasets. We use all datasets for research purposes and follow their stated access conditions and usage terms.

- **MovieLens-1M:** 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/).
- **MS/CAS PENS:** We comply with the Microsoft Research License Terms associated with the dataset distribution (https://github.com/msnews/MIND/blob/master/MSR%20License_Data.pdf).
- **OpenAI TL;DR (summarize-from-feedback):** We comply with the repository license (https://github.com/openai/summarize-from-feedback/blob/master/LICENSE).
- **MPChat (ACL 2023):** We comply with the dataset license and usage restrictions specified by the original authors (CC BY 4.0; https://github.com/ahnjaewoo/MPCHAT/blob/main/LICENSE) and follow the dataset terms of use stated in the official repository (https://github.com/ahnjaewoo/MPCHAT).

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 shared action-on-item schema, MTSH, PerTIDE, evaluated tasks, main gains, ablations, and trace diagnostics. These claims are directly supported by Sections 3–6 and are bounded by the limitations in Section 7.

   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 and Appendix J.6 discuss the main limitations, especially schema-level rather than unrestricted generalizability, imperfect dialogue eventization, missing persona/multimodal grounding, and dependence on target-locus embedding quality.

   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 paper states the assumptions and gives complete proofs for the schema-level generalizability results in Appendix B and the SSM stability, hyperparameter, kernel, and trace-separation results in Appendix D. The main text cross-references these appendices when introducing the corresponding theoretical claims.

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: Sections 4–6 and Appendices A, B, E, I, and J describe the architecture, data construction, baselines, metrics, training procedure, and detailed results. The anonymized supplementary zip contains the implementation and scripts needed to reproduce the reported experiments.

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: The experiments use publicly available datasets and cited model/code assets, while the anonymized supplementary zip provides code, configuration files, data-preparation instructions, and run commands for reproducing the main results. We do not release non-anonymized user data or any private dataset outside the terms of the original assets.

   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: The paper specifies task protocols, candidate sampling, train/evaluation trajectory construction, metrics, baselines, objectives, optimizer, training schedule, and decoder contextualization details in Sections 4–5 and Appendices A, B, D, E, F, G, H, I, and L.

   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 I.3 reports paired bootstrap resampling over the held-out evaluation pool using identical histories, examples, and candidate sets for paired systems. The paper explicitly interprets significance as fixed-system test-set stability rather than variance over independent retraining runs or alternative splits. Independent retraining runs are not reported because they would be too computationally expensive.

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 I.1 reports CPU preprocessing, CUDA GPU training, model size, memory requirements, run time, and total compute needed to reproduce the experiments. The paper also distinguishes training cost from cached-state inference and prompted-LLM inference in Appendix J.1.

   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 cited public or benchmark datasets, anonymized supplementary code, de-identified evaluation records, and bounded claims about personalization rather than deployment-ready user profiling. The submission preserves author anonymity and does not release private user histories, unsafe scraped content, or unrestricted high-risk models.

   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: Section 7 discusses the positive impact of reusable personalization encoders and the negative risks or limitations associated with imperfect eventization, missing persona/multimodal evidence, and restricted transfer. The paper frames PerTIDE as an offline research framework and does not claim deployment readiness for sensitive personalization settings.

    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: [N/A]

    Justification: The paper does not release a high-risk pretrained language model, image generator, scraped dataset, or deployed personalization service. The released supplementary material is anonymized code and reproducibility material; raw private user data are not released.

    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: The paper cites the original creators of the datasets, models, and baseline methods used in the experiments, and the supplementary asset table lists the corresponding versions, licenses, URLs, and terms of use where available. All use of existing datasets and models follows their stated research-use or open-source terms.

    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 newly released asset is the anonymized supplementary code package, including documentation, environment/setup instructions, run scripts, and limitations. The package excludes author-identifying paths and does not redistribute non-anonymized private user data.

    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: [Yes]

Justification: Appendix K provides the human-judgment evaluation protocol, the rating scale, the survey format screenshots, participant-facing instructions, participant demographics in aggregate form, and the voluntary-participation status. Model identities are hidden during evaluation to reduce expectation bias.

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: [Yes]

    Justification: The human-judgment study is a minimal-risk, de-identified similarity-rating study, and Appendix K reports the consent process, participant risks, and anonymous institutional approval, exemption, or equivalent review status. Only aggregate demographic information was collected and reported; no identifiable participant information or private user histories were collected.

    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 describes LLM use in the experimental setup and appendices: a frozen DistilGPT-style decoder is used for generation, and prompted LLMs are used as baselines with prompt templates and decoding settings in Appendix L. LLMs are not used as undisclosed authors of the scientific method or results.

    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.