PERSONALIZATION BY COMPLETION: COLLABORATIVE FILTERING THROUGH LORA WEIGHTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models increasingly interact with users who have different preferences, goals, and styles. Personalizing these models is difficult because a user's interaction history is often sparse: many preferences that matter for a new query may never have been observed. We view this problem as a form of collaborative filtering. Just as recommendation systems infer missing entries in a user–item matrix from patterns across users, we infer how a language model should adapt to a new user and topic from adaptation patterns learned across a population. Our key idea is to represent these patterns directly in the model's low-rank adaptation weights and compose a conditional weight update for an unobserved user–topic pair. The resulting model can personalize its behavior even when the relevant preference is absent from the user's history. Experiments across preference prediction tasks and language models show that this approach consistently improves personalization, with gains that persist even when the full observed history is available. These results suggest that personalized language models can learn not only from what a user has said, but also from what similar users reveal.

1 INTRODUCTION

Large language models increasingly serve users with different goals, preferences, and communication styles (Zhang et al., 2025b; Liu et al., 2025a). Personalization is therefore essential for generating responses for an individual rather than an average user. Existing systems commonly personalize generation from a user's own interaction history (Salemi et al., 2024; Jiang et al., 2025). These histories are sparse: the preference needed for a new query may never have been expressed, even when the user's complete observed record is available. Figure 1 illustrates this gap: music and travel preferences are observed, but the user's fitness preference is missing. The limitation is information coverage, even when the full observed history fits within the prompt.

A large body of work studies how to personalize LLMs from user-specific data. Prompting and retrieval (Salemi et al., 2024; Salemi & Zamani, 2025; Jiang et al., 2025), per-user parameter-efficient adaptation (Tan et al., 2024b;a), and learned user representations such as User-LLM (Ning et al., 2025) improve how models use observed user information. Yet these routes remain predominantly *row-local*: the explicit user-specific state is constructed mainly from a single target user's sparse history. When a query depends on a preference that user has never expressed, that row contains no direct evidence for the missing preference. Shared training can encode population regularities, but the personalization signal itself still comes primarily from what the target user has already revealed.

We introduce CP-gated LoRA to infer preferences absent from the target user's history. CP denotes CANDECOMP/PARAFAC, the factorization underlying its separable user–topic–adapter-target gate. Classical collaborative filtering estimates a missing user–item entry from other users with overlapping observations (Koren et al., 2009; Rendle et al., 2009). We apply the same principle to a sparse user–topic preference matrix: users who overlap on observed preferences provide complementary evidence about cells the target user never revealed. The challenge for an LLM is that the completed preference must control natural-language computation rather than only produce a scalar score. Instead of serializing an external completion back into the prompt, CP-gated LoRA composes population-derived structure directly into the model's adaptation weights.

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

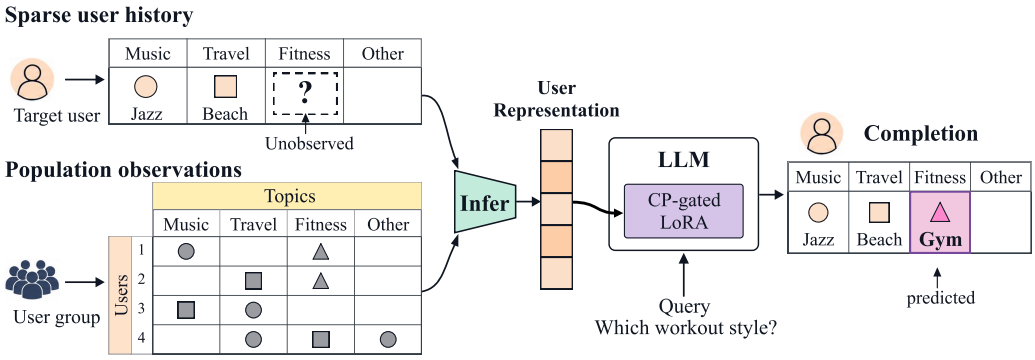


Figure 1: **Personalization by completion.** A target user's history contains music and travel preferences but leaves fitness unobserved. Combining this sparse history with population observations yields a user representation that conditions the LLM through CP-gated LoRA (CP denotes CANDECOMP/PARAFAC). The model predicts the missing preference in response to a fitness query; symbols denote illustrative preference values.

Our design infers a user representation from sparse history and population observations (Figure 1), then translates it into conditional LoRA weights through a separable user–topic–adapter-target gate. The three factors jointly determine coefficients for shared LoRA weight directions (Hu et al., 2022; Kolda & Bader, 2009). This construction produces a conditional update for each user–topic query without storing a separate adapter for every pair. The population representation supplies collaborative information, while the gate determines how it acts at each model location. PersonaGrid tests the resulting completion capability with factor-disjoint queries whose required preference axes are absent from the target user's direct observations.

A key advantage of our approach is that the completion is factored out of the inference prompt. The language-model backbone remains frozen and population information enters through LoRA; other users' raw records do not need to be inserted into the target user's prompt. The same shared adapter can also be applied to users outside language-model training once a population representation is available. This makes collaborative completion complementary to ordinary history-based personalization: direct history can be used when it is informative, while persistent population structure matters most when the local row leaves the queried preference underdetermined.

Our experiments test completion across four complementary settings: controlled PersonaGrid tasks across model families (Yang et al., 2025; Allal et al., 2025; Meta, 2024) and user identities; MovieLens genre-holdout evaluation (TopicHold) on public ratings (Harper & Konstan, 2015); full-history and query-only prediction as observation density changes; and representation interventions together with gate analysis. These experiments connect completion accuracy to user-specific answers and to the population representations that condition the model. Personalized LLMs should not only retrieve what users said, but complete what sparse histories leave unsaid.

2 Related Work

Personalized language models. Prompting, retrieval, memory, user embeddings, and per-user parameter-efficient adaptation all improve how an LLM uses user-specific observations (Salemi et al., 2024; Salemi & Zamani, 2025; Jiang et al., 2025; Ning et al., 2025; Tan et al., 2024b;a). Learned user prompts and embeddings provide further interfaces for personalization (Li et al., 2023; Doddapaneni et al., 2024). Their most common personalization signal is nevertheless the target user's own history. Recent cross-user retrieval also addresses sparse-history generation by drawing on similar users (Ni et al., 2026). We focus on the complementary case in which the queried preference is absent from that history: the relevant question is no longer only how to encode one user's observations, but how to infer what that sparse row leaves unobserved.

Collaborative filtering and preference completion. Low-rank matrix models share information across incomplete user–item records (Koren et al., 2009; Candès & Recht, 2009), while preference

completion extends this idea to other feedback structures (Park et al., 2015). PEP/CAPE similarly uses population correlations to infer unqueried semantic preferences (Bose et al., 2026). CoLLM and BinLLM bring collaborative representations into LLM inputs (Zhang et al., 2025a; 2024), while TALLRec adapts LLMs through recommendation instruction tuning (Bao et al., 2023). Related work also studies personalized reward modeling from limited user feedback (Chen et al., 2025; Cai et al., 2026). These works motivate our completion view: complementary observations from other users can provide information that is missing from the target user's row.

Low-rank and conditional adaptation. Parameter-efficient tuning includes adapters (Houlsby et al., 2019), continuous prefixes (Li & Liang, 2021), and low-rank weight updates (Hu et al., 2022). Hypernetworks generate conditional weights, including adapters shared across tasks and layers (Ha et al., 2017; Karimi Mahabadi et al., 2021). CoRA conditions adaptation on collaborative representations (Liu et al., 2025b); Profile-to-PEFT generates personalized adapters from user/profile context (Tan et al., 2026); MetaLoRA and LoRTA provide related conditional and tensor-structured low-rank parameterizations (Wang et al., 2025; Hounie et al., 2024). PLUME personalizes shared low-rank subspaces through user-specific modulation (Li et al., 2026). Relative to prior collaborative conditioning such as CoRA, our focus is explicitly on missing-preference completion: the conditioning is factorized jointly over the population-inferred user representation, queried topic, and adapter target to compose a query-specific LoRA weight update for an unobserved user–topic pair. CP-gated LoRA implements this idea through a separable user–topic–adapter-target product that composes a conditional increment to LoRA's B factor, carrying the collaborative signal directly into the model's adaptation weights.

3 Problem Setup and Benchmarks

3.1 Sparse Preference Completion

Let $\mathcal{U}$ and $\mathcal{T}$ denote users and topics. We observe preferences $y_{u,t}$ on a sparse set $\Omega \subseteq \mathcal{U} \times \mathcal{T}$, giving user u a record $H_u = \{(t, y_{u,t}) : (u,t) \in \Omega\}$. A query q_t may target an unobserved entry $(u,t) \notin \Omega$. Row-local personalization predicts from H_u; collaborative completion additionally exploits structure learned from other rows.

An unobserved topic does not necessarily imply an unobserved preference: another observed topic may reveal the preference needed for the query. PersonaGrid's primary task therefore requires both queried preferences to be absent from the target user's observations. The observation-density study groups unfilled cells according to whether both, one, or neither required preference is missing. User holdout is separate: a *seen* user contributes other interactions to language-model training; an *unseen* user does not.

3.2 PersonaGrid: A Benchmark for Preference Completion

Existing personalization benchmarks primarily evaluate how models use observed user histories or profiles (Salemi et al., 2024; Jiang et al., 2025; Zollo et al., 2025). PersonaGrid instead explicitly withholds the preferences required by the query, allowing us to evaluate completion rather than direct preference use.

PersonaGrid is a controlled preference-completion benchmark. Users have correlated binary preference factors with individual variation; topics ask about pairs of factors, producing four candidate combinations and 25% chance accuracy. In the primary task, held-out queries require factors absent from the target user's history, while other users provide complementary observations. Appendix A gives the construction and evaluation partitions.

4 Method

4.1 From Sparse Observations to Conditional Weights

Classical collaborative filtering predicts a missing preference as a score. CP-gated LoRA implements collaborative conditioning through weights, allowing population-derived user representations

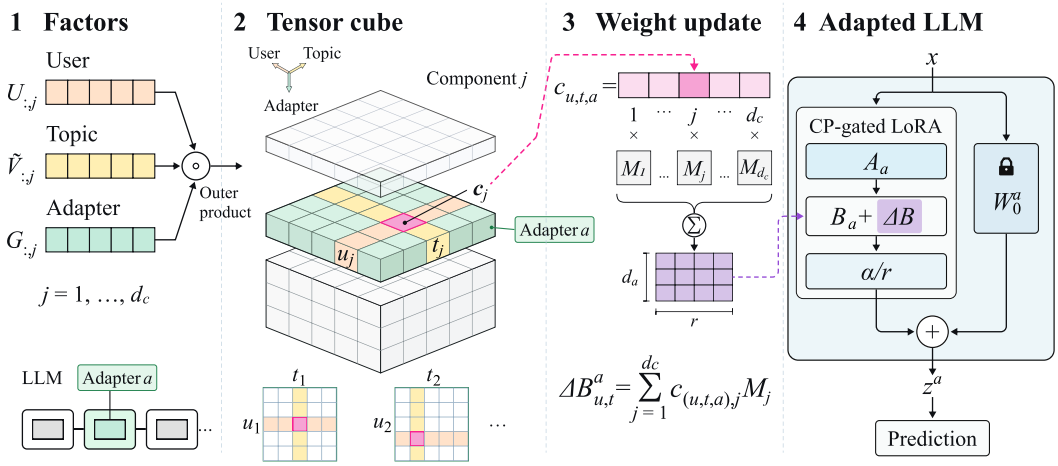


Figure 2: **Overview of our CP-gated LoRA method for LLM personalization (CP = CANDECOMP/PARAFAC). (1–2)** User, topic and adapter factors form component-wise coefficient tensors. Selecting the same user–topic–adapter entry across components yields the gate $c_{u,t,a}$. **(3)** The gate weights shared matrix directions M_j to construct $\Delta B^a_{u,t}$. **(4)** This increment augments B_a in the LoRA branch, whose scaled output is added to the frozen base projection. Lower slices illustrate two queries at a fixed adapter target.

to control the LLM's computation. This interface separates two operations. First, population observations are compressed into a user representation that shares evidence across rows of the sparse matrix. Second, that representation is combined with the current topic and adapter target to compose a conditional LoRA update. Rather than converting a completed preference back into prompt text, the completion acts directly through the model's adaptation weights.

Figure 2 summarizes this interface in four steps. Factor columns define component-wise coefficient tensors; selecting the user–topic–target triple across components gives the gate $c_{u,t,a}$. The gate weights shared directions M_j to construct $\Delta B^a_{u,t}$, which augments the ordinary LoRA factor B_a. The scaled LoRA output is added to the frozen projection. Sharing factors and weight directions across users, topics and model locations avoids storing a separate adapter for every user–topic cell.

4.2 Population-Inferred User Representation

Let $U_u \in \mathbb{R}^{d_c}$ denote user u's representation, with conditional dimension d_c. This vector is a representation of preferences, not a matrix of language-model weights. It summarizes where the user lies in the structure inferred from population observations. On PersonaGrid, observed answers are organized using the factor schema, followed by singular value decomposition (SVD) preprocessing and masked clustering. The inferred assignment is encoded as a cluster indicator. On MovieLens, the representation is a normalized collaborative embedding of the user–movie residual matrix, obtained by SVD. The encoding follows the available observation structure while the downstream conditioning rule is shared.

Population construction can connect a user's limited observations to information outside that user's row. It does not require the user to have answered the queried topic: evidence about the relationship between topics can come from other users. At prediction time, U_u is supplied to the adapter rather than transcribed into the prompt. The primary PersonaGrid representation is held fixed during language-model adaptation. Dataset-specific preprocessing and implementation settings are consolidated in Section 5.1 and Appendix B.

A user absent from language-model training can use the same interface once a representation has been assigned from available observations. This separates user-level transfer from learning a user-specific adapter: shared weights are reused, while only the conditioning input changes.

4.3 CP-GATED LORA

At adapter target a, let $W_0^a \in \mathbb{R}^{d_a \times d_a^{\text{in}}}$ be the frozen base projection, with input $x \in \mathbb{R}^{d_a^{\text{in}}}$ and output width d_a. The index a specifies a projection module at one transformer layer, illustrated by the selected adapter target in Figure 2. Ordinary LoRA uses matrices $A_a \in \mathbb{R}^{r \times d_a^{\text{in}}}$ and $B_a \in \mathbb{R}^{d_a \times r}$, where r is the adaptation rank. CP-gated LoRA makes an additional increment to B_a depend on the user and topic:

$$c_{u,t,a} = U_u \odot \widetilde{V}_t \odot G_a \in \mathbb{R}^{d_c}, \tag{1}$$

$$\Delta B_{u,t}^a = Q \,\mathrm{diag}(c_{u,t,a}) P^\top, \tag{2}$$

$$z^a = W_0^a x + \frac{\alpha}{r}(B_a + \Delta B_{u,t}^a) A_a x. \tag{3}$$

Here $\widetilde{V}_t \in \mathbb{R}^{d_c}$ is the effective topic factor and $G_a \in \mathbb{R}^{d_c}$ the adapter-target factor. The operator $\odot$ denotes elementwise multiplication, and diag places the resulting coefficients on a diagonal. The basis $P \in \mathbb{R}^{r \times d_c}$ maps into the LoRA rank dimension; $Q \in \mathbb{R}^{d_a \times d_c}$ maps into the projection's output dimension. We omit the output-width index on Q and M_j throughout; these matrices are shared across targets with the same output width, with separate families for different widths. We define the *shared weight directions* as

$$M_j := Q_{:,j} P_{:,j}^\top \in \mathbb{R}^{d_a \times r}, \qquad j = 1, \ldots, d_c.$$

Each M_j is determined by the existing trainable P, Q, not an additional independent parameter. The scalar α controls the LoRA scale. The adapted hidden vector z^a is distinct from the preference label $y_{u,t}$. The resulting update to the selected projection weight is $\Delta W_a = (\alpha/r)(B_a + \Delta B_{u,t}^a) A_a$.

The sharing pattern is deliberate. U_u is reused across that user's topics and targets; $\widetilde{V}_t$ is reused across users and targets; G_a is specific to one target but reused across users and topics. P is shared across targets, and Q is shared by targets with the same output width. The ordinary LoRA matrices A_a, B_a are target-specific model weights. Consequently, M_j is shared across users, topics, and adapter locations with the same output width.

The topic factor can incorporate neighborhood information. Where this smoothing is used, $\widetilde{V}_t$ denotes the resulting effective vector, so the same equations apply without changing the role of the user or target factors. The base model remains frozen, while adaptation learns the ordinary LoRA factors and shared conditional parameters from answer supervision. The conditional branch augments ordinary adaptation rather than replacing it. It can be used with a query-only prompt or alongside the user's observed history.

4.4 COLLABORATIVE FILTERING IN WEIGHT SPACE

Writing $c_{(u,t,a),j} = U_{u,j} \widetilde{V}_{t,j} G_{a,j}$, the conditional increment in Equation 2 becomes a weighted sum of shared weight directions:

$$\Delta B_{u,t}^a = \sum_{j=1}^{d_c} c_{(u,t,a),j} \, M_j. \tag{4}$$

Each M_j is a shared weight direction, while the user, topic, and adapter factors determine its coefficient. Other users' observations influence both the population representation and shared adaptation, and a missing cell is handled by composing these factors for the target query. This is how CP-gated LoRA performs collaborative filtering through LoRA weights: the collaborative signal changes the model's computation, rather than only producing a preference score outside the model or appending more history to its prompt.

The indicator encoding makes this mechanism concrete. If $U_u = e_c$, the unit vector for inferred group c, the conditional increment reduces to

$$\Delta B_{u,t}^a = \widetilde{V}_{t,c} G_{a,c} \, M_c. \tag{5}$$

The user selects a shared direction, and the topic and target modulate its strength. Only this conditional residual has rank at most one; the ordinary LoRA term remains active. Dense user representations combine multiple directions under the same formulation. Both encodings avoid storing

a separate adapter for every matrix cell and make user similarity relevant to the update received for a common topic. Section 5.5 examines how observed-user similarity and held-out preferences are reflected in these gate coefficients.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Benchmarks and users. PersonaGrid tests four-candidate preference completion, with both required preferences absent from the target user's observations; chance accuracy is 25%. MovieLens evaluates binary above- or below-tendency genre preferences on ML-1M and ML-25M (Harper & Konstan, 2015). Seen and unseen users are distinguished by whether they contributed language-model training examples. We report candidate accuracy (%) on both benchmarks. Where three independent runs are available, we report their mean; otherwise we report the available evaluation.

Models and information routes. We evaluate Qwen3-4B, Qwen3-1.7B (Yang et al., 2025), SmolLM2-1.7B (Allal et al., 2025) and Llama-3.2-3B (Meta, 2024). The primary PersonaGrid methods train shared LoRA adapters: in-context learning (ICL) uses textual history (Salemi et al., 2024; Salemi & Zamani, 2025), Memory-ICL uses a history summary (Jiang et al., 2025), and CP-gated LoRA conditions weights on a population-inferred user representation while receiving a query-only prompt. We use "ICL" as shorthand for history-in-context prediction with a shared LoRA adapter trained across users; it should not be interpreted as parameter-free in-context learning. MovieLens ICL receives 20 ratings from other genres. In the observation-density study, both methods receive the full observed history; a complementary evaluation compares query-only CP-gated LoRA with full-history ICL on missing-preference queries. Appendix B gives training settings, and Appendix F reports user-specific conditioning diagnostics.

5.2 PREFERENCE COMPLETION ACROSS BACKBONES AND USERS

Population-informed adaptation recovers preferences absent from the user's history. On the primary PersonaGrid task, CP-gated LoRA achieves **53.2%** accuracy with Qwen3-4B, compared with **29.0%** for ICL and **30.4%** for Memory-ICL (Table 1). Its 22.8-point advantage over the stronger history baseline persists when the observed history is summarized rather than supplied verbatim.

Table 1: **PersonaGrid preference completion across backbones and users.** Candidate accuracy (%) when both queried preferences are absent from the user's observations; chance is 25%. ICL and Memory-ICL use history text or its summary; Ours uses population conditioning with a query-only prompt.

	Seen users			Unseen users		
Backbone	ICL	Memory-ICL	Ours	ICL	Memory-ICL	Ours
Qwen3-4B	29.0	30.4	**53.2**	27.4	27.4	**48.1**
Qwen3-1.7B	27.3	26.0	**55.6**	29.7	28.4	**48.1**
SmolLM2-1.7B	31.2	28.1	**54.5**	28.9	27.6	**46.6**
Llama-3.2-3B	26.4	28.1	**55.4**	26.3	23.9	**50.0**

Completion transfers across model families and user identities. Relative to the stronger history baseline, the gains span 22.8–28.3 percentage points on seen users and 17.7–23.7 points on unseen users across the four backbones. With Qwen3-4B, Ours reaches 48.1% on unseen users, compared with 27.4% for either history baseline. Once a population representation is available, the shared conditional adapter can therefore personalize users who contributed no language-model training examples.

Figure 3 illustrates three test cases in which CP-gated LoRA selects the correct candidate while ICL selects an incorrect candidate, despite neither queried preference appearing in the user's observed history.

Observed history	ICL (incorrect)	CP-gated LoRA (correct)
a Food	**What to eat?** (spending + health unobserved)	
Fitness: solo, low-tech Food: private, low-tech	**pinching pennies,** and treating themselves.	**paying for premium,** and treating themselves.
b Travel	**How to travel?** (social + planning unobserved)	
Travel: premium, high-tech Food: premium, healthy	**going it alone,** **and winging it.**	**involving friends,** **and mapping it out ahead** **of time.**
c Fitness	**Which workout?** (health + technology unobserved)	
Food: flexible, eco-friendly Travel: flexible, eco-friendly	indulging without guilt, **and keeping it simple and** **analog.**	indulging without guilt, **and going for the** **cutting-edge model.**

Figure 3: **Completing preferences absent from user history.** Three PersonaGrid cases show observed history, the queried preference dimensions, and the candidates selected by ICL and CP-gated LoRA. Red and blue text highlight the differing answer components selected by ICL (incorrect) and CP-gated LoRA (correct), respectively. Histories and queries are abbreviated; answers show recorded candidate selections for three cases where CP-gated LoRA succeeds and ICL fails.

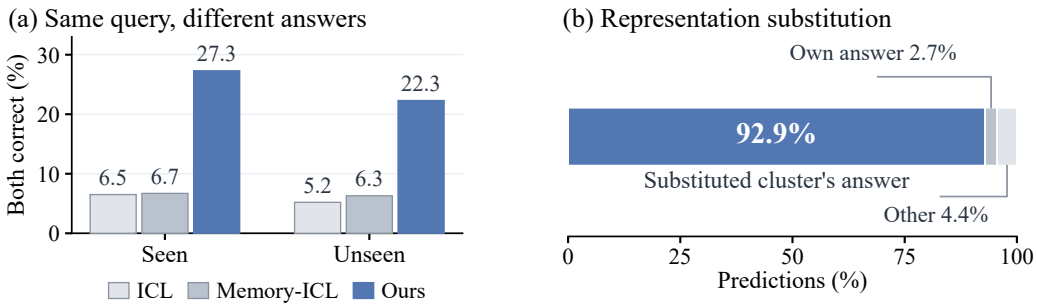


Figure 4: **Personalized answers depend on user conditioning.** (a) Fraction of same-query, different-answer user pairs for which both users are answered correctly (939 seen and 897 unseen pairs). (b) Prediction destinations after replacing only the user representation, for queries where the substituted cluster's majority answer differs from the user's answer. Tables 7 and 8 give the full outcomes; reference probabilities are provided in Appendix F.

The model distinguishes users who require different answers to the same question. We pair users with the same query but different correct answers, with both required preferences absent from each user's observations. Ours answers both users correctly in 27.3% of seen-user pairs and 22.3% of unseen-user pairs, compared with 6.5% and 5.2% for ICL (Figure 4a). A common answer to the shared query cannot be correct for both users: this evaluation directly tests whether the model expresses their distinct preferences. Replacing only the user representation redirects these predictions (Figure 4b), an intervention examined in Section 5.5.

5.3 Public-Data Results on MovieLens

Collaborative completion extends to public rating-derived preferences. Table 2 evaluates MovieLens across nine dataset–backbone configurations, including ML-1M, ML-25M and populations of up to 22,400 users. Ours improves every unseen-user comparison; on seen users, it improves eight comparisons and matches ICL in the remaining setting. These results extend the completion evidence from controlled preference factors to public ratings across model families and population sizes.

Table 2: **Public-data results on MovieLens.** Candidate accuracy (%) for genre-holdout prediction. Seen/unseen denotes language-model training participation. Ours uses population-inferred conditioning; ICL receives history. Bold marks the higher accuracy within each comparison. Appendix C describes preprocessing and the observation-only control.

		Seen users		Unseen users	
Dataset	Backbone	ICL	Ours	ICL	Ours
ML-1M (2.5k users)	Qwen3-4B	61.3	**66.9**	55.9	**63.5**
ML-1M	Qwen3-1.7B	57.6	**67.6**	55.7	**64.1**
ML-1M	SmolLM2-1.7B	57.7	**67.7**	55.5	**63.2**
ML-1M	Llama-3.2-3B	61.0	**67.5**	58.4	**63.5**
ML-25M (7.7k users)	Qwen3-4B	70.8	70.8	64.7	**72.7**
ML-25M	Qwen3-1.7B	69.0	**73.0**	63.5	**72.5**
ML-25M	SmolLM2-1.7B	68.3	**73.1**	64.0	**71.7**
ML-25M	Llama-3.2-3B	70.0	**73.1**	66.1	**71.8**
ML-25M (22k users)	Qwen3-4B	68.2	**71.2**	66.1	**71.5**

A stricter observation-only control masks held-out ratings before computing population statistics and fits the collaborative basis using seen users only. Under this construction, Ours remains stronger on unseen users (64.1% vs. 57.2%), while ICL is stronger on seen users (69.0% vs. 64.7%). Permuting user representations reduces accuracy toward chance, showing that the correspondence between users and their representations matters (Appendix C).

5.4 Sparsity and Scaling

Completion remains useful with the full observed history. We fix 1,200 users and 312 topics and vary nominal observation density from 2% to 64%. Both methods receive each user's full observed history and predict preferences at unfilled cells. Figure 5a summarizes the full-history observation-density study; Appendix D reports exact accuracies and density statistics.

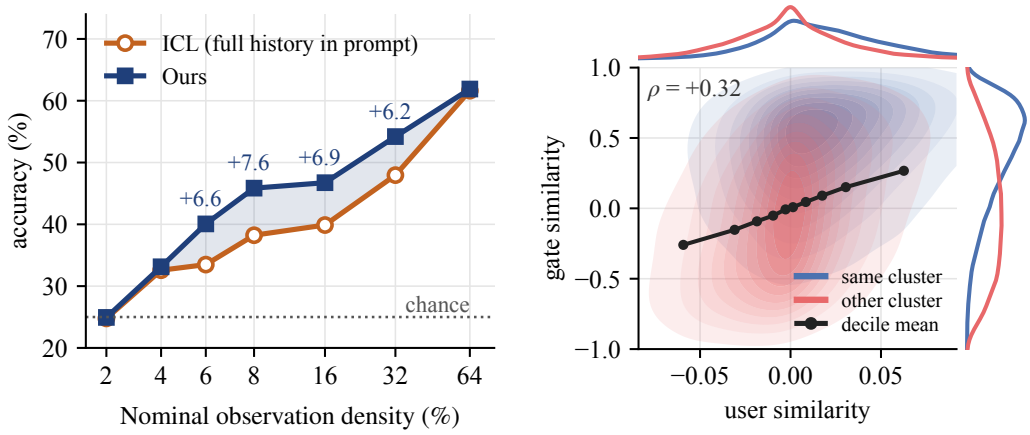
CP-gated LoRA improves accuracy by 6.2–7.6 percentage points at 6–32% nominal observation density. The two methods perform similarly at the lowest densities and at 64%, where their accuracies are 61.9% and 61.6%. The clearest overall gains arise when population observations support inference while some query-relevant preferences remain unresolved by direct history.

Missing-preference queries retain an advantage at high density. At 64% nominal observation density, CP-gated LoRA leads by 9.6 points when both required preferences are absent and by 4.5 points when one is absent; ICL is stronger when both are observed. The same ordering appears at 8%, 16% and 32% (Appendix D). Thus, similar overall accuracies at high density combine two complementary strengths: collaborative completion of missing preferences and inference from preferences already available in the history.

On queries for which both required preferences remain absent, query-only CP-gated LoRA improves from 23.1% to 44.6% as nominal observation density increases from 2% to 64%, while full-history ICL remains between 23.0% and 26.0%. Both methods answer the same query set within each density (Appendix D.2). The query-only route keeps the user-turn prompt near 14 tokens, while full-history prompting grows to about 4.6k tokens at 64% density (Appendix G). Together, these comparisons show that full-history access does not eliminate the advantage of population conditioning when the queried preferences remain unobserved.

5.5 How User Representations Shape Predictions

Changing only the user representation redirects the prediction. We replace a user's inferred cluster indicator with that of a different cluster, holding trained weights, query and candidates fixed. Among queries for which the substituted cluster's majority training answer differs from the user's correct answer, predictions follow the substituted cluster in 92.9% of cases (Figure 4b). This intervention directly establishes that the user representation controls which preference the conditional adapter expresses (Appendix F).



(a) Completion with full observed histories.

(b) User similarity in conditional gates.

Figure 5: **Full-history completion and population structure.** (a) Accuracy on unfilled PersonaGrid cells as nominal observation density increases, with both methods receiving the user's full observed history. The largest gains occur at 6–32% density; the curves nearly meet at the lowest and highest densities. (b) For 300,000 MovieLens user pairs, observed-rating-profile cosine similarity is positively associated with conditional-gate cosine similarity (Adventure; fixed adapter target; Spearman $\rho = +0.32$). Pairs in the same collaborative-filtering cluster are shown in blue.

Similar observed users receive similar gates. On MovieLens, observed-rating-profile similarity is positively associated with conditional-gate similarity across 300,000 user pairs (Adventure; fixed adapter target; Spearman $\rho = +0.32$; Figure 5b). Thus, relationships recovered from user observations are reflected in the coefficients that compose the adaptation weights.

Gate neighborhoods reflect held-out preferences. Adventure preference-label purity among neighboring gates is 0.60, compared with 0.50 after label shuffling; purity exceeds the shuffled comparator for 15 of 18 genres (Appendix E). This connects gate geometry to preferences withheld from the observed profiles, complementing the intervention with behavioral evidence.

With PersonaGrid's one-hot user representation, the tested user-conditioned gates perform similarly, whereas removing user conditioning reduces accuracy toward chance. With MovieLens's dense SVD representation, the multiplicative CP gate is the strongest tested form (Appendix F). This pattern is consistent with the factorization: a one-hot state selects one shared direction, whereas a dense state permits topic-dependent modulation of multiple active coordinates.

6 Discussion and Conclusion

Sparse personalization is a missing-preference completion problem: other users provide complementary evidence for preferences absent from an individual's observations. CP-gated LoRA brings population structure into language-model computation through conditional LoRA weights. Completion across backbones and users, distinct answers to the same query, and predictions redirected by representation substitution establish user-specific behavior; MovieLens and observation-density results extend this evidence to public ratings and full-history access. Population-conditioned adaptation and direct-history inference are complementary, addressing missing and observed preferences within the same personalization task. Our evaluation isolates preference completion primarily through candidate prediction, so it does not establish gains for unrestricted open-ended generation. In addition, the original MovieLens protocol uses transductive collaborative preprocessing; evaluating fully inductive population representations is an important direction for future work. Personalized LLMs should not only retrieve what users said, but complete what sparse histories leave unsaid.

AI USE STATEMENT

LLM-based assistants were used to review and polish the manuscript, including the presentation and interpretation of experimental results, and to assist in preparing experimental plots. The overview and method schematics were drawn by the authors. PersonaGrid instances were generated using rule-based templates rather than LLM-generated content. The authors reviewed the AI-assisted materials and take full responsibility for the final manuscript, code, reported results, and conclusions.

REFERENCES

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Agustín Piqueres Lajarín, Hynek Kydlíček, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Ben Burtenshaw, Clémentine Fourrier, Haojun Zhao, Hugo Larcher, Mathieu Morlon, Cyril Zakka, Colin Raffel, Leandro von Werra, and Thomas Wolf. SmolLM2: When smol goes big—data-centric training of a fully open small language model. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=3JiCl2A14H.

Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. TALLRec: An Effective and Efficient Tuning Framework to Align Large Language Model with Recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pp. 1007–1014. ACM, 2023. doi: 10.1145/3604915.3608857. URL https://doi.org/10.1145/3604915.3608857.

Avinandan Bose, Shuyue Stella Li, Faeze Brahman, Pang Wei Koh, Simon Shaolei Du, Yulia Tsvetkov, Maryam Fazel, Lin Xiao, and Asli Celikyilmaz. Cold-start personalization via training-free priors from structured world models. In *Lifelong Agent Workshop at ICLR 2026*, 2026. URL https://openreview.net/forum?id=2cL4zYN71g.

Hongru Cai, Yongqi Li, Tiezheng Yu, Fengbin Zhu, Wenjie Wang, Fuli Feng, and Wenjie Li. One Adapts to Any: Meta Reward Modeling for Personalized LLM Alignment. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 110–121. ACM, 2026. doi: 10.1145/3805712.3809689. URL https://doi.org/10.1145/3805712.3809689.

Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009. doi: 10.1007/s10208-009-9045-5. URL https://doi.org/10.1007/s10208-009-9045-5.

Daiwei Chen, Yi Chen, Aniket Rege, Zhi Wang, and Ramya Korlakai Vinayak. PAL: Sample-efficient personalized reward modeling for pluralistic alignment. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=1kFDrYCuSu.

Sumanth Doddapaneni, Krishna Sayana, Ambarish Jash, Sukhdeep Sodhi, and Dima Kuzmin. User embedding model for personalized language prompting. In *Proceedings of the 1st Workshop on Personalization of Generative AI Systems (PERSONALIZE 2024)*, pp. 124–131. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.personalize-1.12. URL https://aclanthology.org/2024.personalize-1.12/.

David Ha, Andrew M. Dai, and Quoc V. Le. HyperNetworks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=rkpACe1lx.

F. Maxwell Harper and Joseph A. Konstan. The MovieLens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems*, 5(4):19:1–19:19, 2015. doi: 10.1145/2827872. URL https://doi.org/10.1145/2827872.

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

Ignacio Hounie, Charilaos Kanatsoulis, Arnuv Tandon, and Alejandro Ribeiro. LoRTA: Low rank tensor adaptation of large language models, 2024. URL https://arxiv.org/abs/2410.04060v3. Version 3.

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

Bowen Jiang, Zhuoqun Hao, Young-Min Cho, Bryan Li, Yuan Yuan, Sihao Chen, Lyle Ungar, Camillo J. Taylor, and Dan Roth. Know me, respond to me: Benchmarking LLMs for dynamic user profiling and personalized responses at scale. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=6ox8XZGOqP.

Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 565–576, 2021. doi: 10.18653/v1/2021.acl-long.47. URL https://aclanthology.org/2021.acl-long.47/.

Tamara G. Kolda and Brett W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009. doi: 10.1137/07070111X. URL https://doi.org/10.1137/07070111X.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009. doi: 10.1109/MC.2009.263. URL https://doi.org/10.1109/MC.2009.263.

Lei Li, Yongfeng Zhang, and Li Chen. Personalized prompt learning for explainable recommendation. *ACM Transactions on Information Systems*, 41(4), 2023. doi: 10.1145/3580488. URL https://doi.org/10.1145/3580488.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, 2021. doi: 10.18653/v1/2021.acl-long.353. URL https://aclanthology.org/2021.acl-long.353/.

Xinyu Li, Hao Zhou, Jianfeng Zhu, Julina Maharjan, Ruixin Guo, Feodor Dragan, and Ruoming Jin. PLUME: Parameter-Efficient Personalization of Large Language Models via Low-Rank User Modulation in Shared Subspaces, 2026. URL https://arxiv.org/abs/2609.04715.

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

Yuting Liu, Jinghao Zhang, Yizhou Dang, Yuliang Liang, Qiang Liu, Guibing Guo, Jianzhe Zhao, and Xingwei Wang. CoRA: Collaborative information perception by large language model's weights for recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 12246–12254, 2025b. doi: 10.1609/aaai.v39i12.33334. URL https://doi.org/10.1609/aaai.v39i12.33334.

Meta. Llama 3.2 model card. Official model card, 2024. URL https://huggingface.co/meta-llama/Llama-3.2-3B.

Bo Ni, Haowei Fu, Qinwen Ge, Franck Dernoncourt, Samyadeep Basu, Nedim Lipka, Seunghyun Yoon, Yu Wang, Nesreen K. Ahmed, Subhojyoti Mukherjee, Puneet Mathur, Ryan A. Rossi, and Tyler Derr. Sparse Personalized Text Generation with Multi-Trajectory Reasoning, 2026. URL https://arxiv.org/abs/2604.24996.

Lin Ning, Luyang Liu, Jiaxing Wu, Neo Wu, Devora Berlowitz, Sushant Prakash, Bradley Green, Shawn O'Banion, and Jun Xie. User-LLM: Efficient LLM contextualization with user embeddings. In *Companion Proceedings of the ACM on Web Conference 2025*, pp. 1219–1223. Association for Computing Machinery, 2025. doi: 10.1145/3701716.3715463. URL https://doi.org/10.1145/3701716.3715463.

Dohyung Park, Joe Neeman, Jin Zhang, Sujay Sanghavi, and Inderjit Dhillon. Preference completion: Large-scale collaborative ranking from pairwise comparisons. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1907–1916. PMLR, 2015. URL https://proceedings.mlr.press/v37/park15.html.

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*, pp. 452–461. AUAI Press, 2009. URL https://auai.org/uai2009/papers/UAI2009_0139_48141db02b9f0b02bc7158819ebfa2c7.pdf.

Alireza Salemi and Hamed Zamani. Comparing retrieval-augmentation and parameter-efficient fine-tuning for privacy-preserving personalization of large language models. In *Proceedings of the International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval (ICTIR)*, 2025. doi: 10.1145/3731120.3744595. URL https://doi.org/10.1145/3731120.3744595.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. LaMP: When large language models meet personalization. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024. doi: 10.18653/v1/2024.acl-long.399. URL https://aclanthology.org/2024.acl-long.399/.

Zhaoxuan Tan, Zheyuan Liu, and Meng Jiang. Personalized pieces: Efficient personalized large language models through collaborative efforts. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024a. doi: 10.18653/v1/2024.emnlp-main.371. URL https://aclanthology.org/2024.emnlp-main.371/.

Zhaoxuan Tan, Qingkai Zeng, Yijun Tian, Zheyuan Liu, Bing Yin, and Meng Jiang. Democratizing large language models via personalized parameter-efficient fine-tuning. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024b. doi: 10.18653/v1/2024.emnlp-main.372. URL https://aclanthology.org/2024.emnlp-main.372/.

Zhaoxuan Tan, Zixuan Zhang, Haoyang Wen, Zheng Li, Rongzhi Zhang, Pei Chen, Fengran Mo, Zheyuan Liu, Qingkai Zeng, Qingyu Yin, and Meng Jiang. Instant personalized large language model adaptation via hypernetwork. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23557–23580. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1081. URL https://aclanthology.org/2026.acl-long.1081/.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008. URL https://www.jmlr.org/papers/v9/vandermaaten08a.html.

Maolin Wang, Xiangyu Zhao, Ruocheng Guo, and Junhui Wang. MetaLoRA: Tensor-Enhanced Adaptive Low-Rank Fine-Tuning. In *2025 IEEE 41st International Conference on Data Engineering (ICDE)*, pp. 4680–4684. IEEE, 2025. doi: 10.1109/icde65448.2025.00376. URL https://doi.org/10.1109/icde65448.2025.00376. PhD Symposium.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger

Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 Technical Report, 2025. URL https://arxiv.org/abs/2505.09388.

Yang Zhang, Keqin Bao, Ming Yan, Wenjie Wang, Fuli Feng, and Xiangnan He. Text-like encoding of collaborative information in large language models for recommendation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9181–9191. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.497. URL https://aclanthology.org/2024.acl-long.497/.

Yang Zhang, Fuli Feng, Jizhi Zhang, Keqin Bao, Qifan Wang, and Xiangnan He. CoLLM: Integrating collaborative embeddings into large language models for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 37(5), 2025a. doi: 10.1109/TKDE.2025.3540912. URL https://hexiangnan.github.io/papers/tkde25-CoLLM.pdf.

Zhehao Zhang, Ryan A. Rossi, Branislav Kveton, Yijia Shao, Diyi Yang, Hamed Zamani, Franck Dernoncourt, Joe Barrow, Tong Yu, Sungchul Kim, Ruiyi Zhang, Jiuxiang Gu, Tyler Derr, Hongjie Chen, Junda Wu, Xiang Chen, Zichao Wang, Subrata Mitra, Nedim Lipka, Nesreen Ahmed, and Yu Wang. Personalization of large language models: A survey. *Transactions on Machine Learning Research*, 2025b. URL https://openreview.net/forum?id=tf6A9EYMo6.

Thomas P. Zollo, Andrew Wei Tung Siah, Naimeng Ye, Ang Li, and Hongseok Namkoong. PersonalLLM: Tailoring LLMs to Individual Preferences. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=2R7498e2Tx.

A PERSONAGRID CONSTRUCTION AND TASK EXAMPLE

For a user u, let O_u be the preference factors revealed by the observed history, and let Q_t be the factors required by query q_t. Define

$$h(u,t) = |Q_t \setminus O_u|.$$

Thus, $h = 2$, $h = 1$ and $h = 0$ mean that both, one or neither required factor is absent from the user's observations. An unobserved topic cell can belong to any of these groups.

The primary PersonaGrid task uses eight population prototypes over seven binary preference axes. Individual signs vary around their assigned prototype with flip probability 0.1. Topic questions combine two axes, and the four candidate answers express their four sign combinations. Sparse users reveal only a subset of axes through observed topics. A query with $h = 2$ requires two factors outside that direct observation set. This construction retains cross-user correlations rather than making the missing factors statistically independent of all available information.

For example, a user's history includes: "For picking what to eat, they chose trying something bold and new, and keeping it cheap" and "For planning a trip, they chose chasing novelty, and improvising as they go." The corresponding fitness query concerns health and technology preferences. Food and travel expose novelty, budget and planning, whereas neither queried factor is directly expressed in those observations.

The observation boundary is user-specific. Other users may observe factors withheld from the target user. Query text specifies what is being asked, while candidate texts define the output alternatives. The correct candidate identity is a scoring target. This separates direct preference use from completion without claiming that candidate prediction is equivalent to unrestricted language generation.

B IMPLEMENTATION AND EVALUATION DETAILS

Population representations. The PersonaGrid pipeline parses observed answers according to the factor schema, performs SVD-based preprocessing and masked clustering, and encodes the assignment as a unit vector. Eight coordinates of the conditional vector identify its clusters; the remaining coordinates are zero. The primary PersonaGrid user representation is fixed during language-model adaptation. The ML-1M pipeline instead uses a row-normalized, 32-dimensional SVD representation of user–movie residuals, padded to the conditional dimension. The two encodings should not be interpreted as identical preprocessing algorithms.

Adapter and training. For the primary PersonaGrid Qwen3-4B comparison, ICL, Memory-ICL and CP-gated LoRA each train on 5,280 rows for 15 epochs (1,320 steps per epoch; 19,800 total), with batch size 4 and no gradient accumulation. All three use supervised cross-entropy on answer tokens, AdamW with learning rate 10^{-4} and 100 warmup steps, and rank-8 LoRA with scale 8 on the query and value projections across all 36 layers. The shared LoRA adapter has 2,949,120 trainable parameters; CP-gated LoRA adds 338,849 conditional parameters with $d_c = 64$ and keeps its population-inferred user representation frozen. Each method selects the checkpoint with the highest validation accuracy on the validation partition. The other three backbones also train all three methods for 15 epochs. The P basis is shared across targets and Q across targets with equal output width. Topic factors can combine a learned neighbor mixture and fixed neighbor smoothing before forming the gate.

Evaluation protocols. The primary PersonaGrid seen partition has 462 queries from 77 users; the unseen partition has 468 queries from 78 users. In the primary PersonaGrid comparison, ICL receives the target user's history, Memory-ICL receives a history summary, and CP-gated LoRA uses a query-only prompt together with population conditioning; evaluation is four-way candidate prediction on the $h = 2$ task. In MovieLens genre-holdout evaluation, ICL receives 20 observed ratings from other genres, while CP-gated LoRA uses the query and population-inferred representation for binary held-out-genre prediction. In the full-history observation-density study, both methods receive the user's full observed history, and CP-gated LoRA additionally uses population conditioning; evaluation covers unfilled cells overall and by h in Appendix D.

User transfer. An unseen user is absent from language-model training. Assigning that user a representation is a separate operation from evaluating the shared conditional adapter. For the origi-

nal ML-1M results, collaborative preprocessing is transductive: the population embedding includes available observations of users outside language-model training. The unseen partition therefore tests transfer beyond language-model training identities, not a strictly inductive collaborative fit. It should not be conflated with a setting that fits all preprocessing only on training users.

C MovieLens Genre-Holdout Evaluation and Public-Data Control

Original protocol. TopicHold denotes our MovieLens genre-holdout evaluation protocol, not a separate dataset. Table 2 reports ML-1M with 2,506 users, a sampled ML-25M task with 7,728 users, and the full qualifying ML-25M population with 22,400 users. Genre-level labels use double-centered movie-rating residuals. The original protocol selects three held-out genres per user and removes movies carrying those genres from direct histories and training examples. Its collaborative representation is a normalized SVD embedding of residual ratings. Seen and unseen refer to language-model training membership; the original collaborative preprocessing is transductive and includes available observations from users outside that training set.

Movie means and user offsets in the original construction were computed before masking held-out entries, so the original table should be read under that preprocessing convention. The observation-only control below evaluates a stricter construction without replacing the original public-data result.

Observation-only control. This control masks held-out ratings before computing population statistics and fits its collaborative basis using seen users only. Table 3 reports the resulting evaluation separately from the original protocol. Ours is higher on unseen users, while ICL is higher on seen users. Permuting user representations reduces accuracy toward 50%.

Table 3: **Separate observation-only ML-1M control.** Candidate accuracy (%). User-permutation ranges summarize the tested perturbations.

Configuration	Seen	Unseen
ICL	69.0	57.2
Ours	64.7	64.1
Ours, users permuted	49.6–50.2	47.3–53.0

D Sparsity and Scaling Details

D.1 Full-History Observation-Density Study

Table 4: **Preference prediction across observation densities.** Accuracy (%) on unfilled PersonaGrid cells with full observed histories for both methods; Δ is Ours minus ICL in percentage points. All grids have 1,200 users and 312 topics. Differences are computed before rounding.

Nominal observation density (%)	Topics/user	ICL	**Ours**	Δ
2	8	24.8	24.9	+0.2
4	12	32.6	33.1	+0.6
6	18	33.5	40.1	+6.6
8	28	38.2	45.9	+7.6
16	52	39.9	46.7	+6.9
32	97	48.0	54.2	+6.2
64	188	61.6	61.9	+0.3

The study in Section 5.4 uses the same 1,200-user, 312-topic population at seven observation densities. Nominal observation densities of 2, 4, 6, 8, 16, 32 and 64% correspond to realized densities of 2.4, 3.9, 5.8, 8.9, 16.7, 31.2 and 60.4%. The corresponding observed topics per user are 8, 12, 18,

28, 52, 97 and 188. Both methods receive the full observed history. Within each density level they use the same observed training rows and training schedule; the numbers of epochs across levels are 4, 4, 4, 3, 3, 2 and 2. Table 4 reports the overall accuracies and differences.

Table 5 groups unfilled cells by the number of required preference factors absent from the user's observations. All groups concern unfilled topic cells: $h = 0$ means that both factors have appeared in other observed topics, not that the query cell was observed. The table reports group shares and ICL/Ours accuracy pairs. Shares are rounded and may not sum to exactly 100%; no new aggregate score is reconstructed from these rounded entries.

Table 5: **Full-history observation-density study by preference-factor coverage.** Query shares (%) are ordered as $h = 2/1/0$. Each accuracy cell reports ICL / Ours (%). All queries target unfilled user–topic cells. Epoch counts apply to both methods within the corresponding density level.

Nominal observation density (%)	Epochs	Query share $h = 2/1/0$	Both absent $h = 2$	One absent $h = 1$	None absent $h = 0$
2	4	74 / 24 / 2	25.0 / 24.8	24.2 / 25.3	24.2 / 27.5
4	4	39 / 51 / 10	23.8 / 24.4	36.2 / 36.5	48.5 / 49.6
6	4	38 / 53 / 9	23.0 / 28.2	36.5 / 42.8	58.6 / 72.8
8	3	39 / 56 / 5	23.9 / 32.5	44.2 / 51.9	87.9 / 86.0
16	3	32 / 64 / 4	23.2 / 30.9	44.9 / 52.3	96.5 / 87.0
32	2	19 / 74 / 8	24.5 / 34.7	48.5 / 55.9	99.2 / 85.1
64	2	3 / 73 / 24	26.0 / 35.6	50.6 / 55.1	99.9 / 86.0

At nominal observation densities of 8–64%, Ours is more accurate in both missing-factor groups, whereas ICL is more accurate when both factors are observed. At 6%, Ours also improves the $h = 0$ group, from 58.6% to 72.8%; the missing-factor pattern is therefore not asserted at every density. At 64%, 76% of queries still lack at least one required factor, even though overall accuracy is nearly equal. The subgroup differences are $+9.6$, $+4.5$ and -13.9 points for $h = 2, 1, 0$, respectively. This contrast explains the distinct behavior underlying the overall $+0.3$-point difference without assuming that most queries are fully covered.

D.2 Query-Only Completion with Both Factors Absent

The complementary query-only evaluation focuses on $h = 2$ queries along the observation-density study. Ours receives the query and its inferred user representation, while ICL receives the full observed history. This differs from the preceding full-history comparison, in which Ours also receives history text. Figure 6 reports the query-only series separately.

At nominal observation densities below 8%, Ours remains close to chance. Its accuracy increases to 34.7% at 8% and 44.6% at 64%, whereas ICL stays between 23.0% and 26.0%. The corresponding margins are 10.8 and 18.6 percentage points. These results describe recovery of missing preferences as the observation matrix becomes denser. Observed topics and the training schedule vary across density levels; the exact settings are reported above.

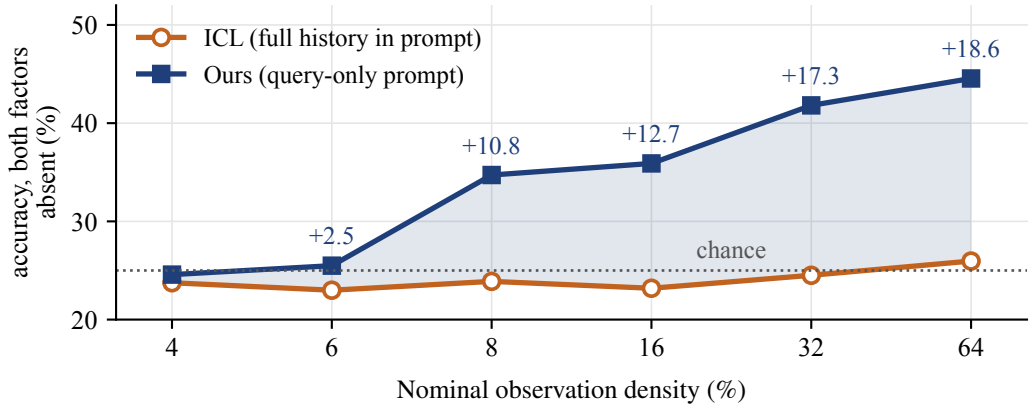


Figure 6: **Completion when the queried factors remain unobserved.** Accuracy on queries with both required preference factors absent, at 4–64% nominal observation density. Ours uses a query-only prompt; ICL receives the full observed history. The dotted line marks 25% chance; annotations show the accuracy difference in percentage points. Both methods evaluate the same query set within each density. Each plotted setting uses 1,440 queries through 32% and 936 queries at 64%.

E GATE GEOMETRY ACROSS GENRES

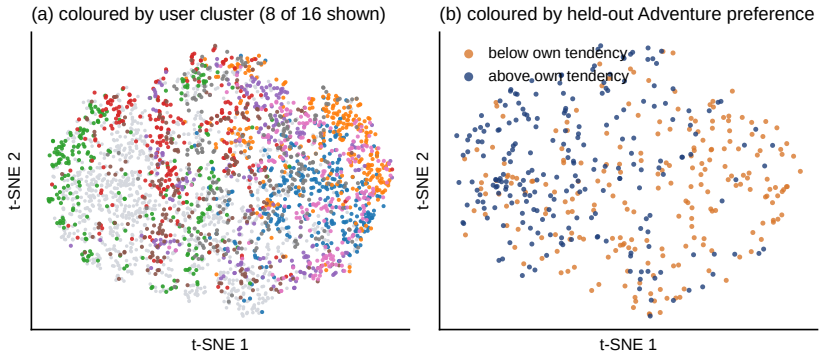


Figure 7: **Latent preference structure in the conditional gate.** t-SNE (van der Maaten & Hinton, 2008) visualization of MovieLens gate coefficients for a fixed genre and adapter target. (a) Eight of sixteen inferred user clusters are highlighted; other users are gray. (b) Users with an available held-out Adventure preference are colored by whether they rate it above or below their own tendency. Reported 10-nearest-neighbor purity is 0.35 for inferred clusters (reference 0.06) and 0.60 for the preference label (shuffled-label reference 0.50). Table 6 reports all genres.

User-pair similarity. The analysis in Figure 5b uses 300,000 MovieLens user pairs for the Adventure topic and one fixed adapter target. It compares cosine similarity of observed rating profiles with cosine similarity of the conditional gates $c_{u,t,a}$, yielding Spearman $\rho = +0.32$. Pairs from the same collaborative-filtering cluster concentrate at high gate similarity. This pairwise association and the neighborhood-purity analysis below measure complementary properties.

Gate neighborhoods and held-out labels. The neighborhood analysis evaluates 10-nearest-neighbor purity in the conditional gate features for each genre, using one adapter target. Cluster purity measures agreement with inferred cluster assignments, with a reference of 0.06. Label purity measures agreement with above/below-tendency held-out genre labels and is compared with a label-shuffling control. Figure 7 provides supplementary visualization; numerical purity is computed in the original gate feature space rather than from the two-dimensional projection. Eight of sixteen clusters are highlighted in panel (a), and panel (b) contains users with available Adventure labels.

Table 6 reports all 18 genres. Fifteen label-purity values exceed their shuffled counterparts. The exceptions are Western, Animation and Documentary; Documentary also has the smallest labeled-user count. Cluster alignment and preference-label alignment answer different questions: the former describes retention of inferred representation structure, while the latter relates it to behavior.

Table 6: **Gate-neighborhood purity by genre.** n is the number of labeled users. Cluster purity uses inferred assignments; label purity uses held-out preferences.

Genre	n	Cluster	Label	Shuffled label
Adventure	440	0.35	0.60	0.50
Musical	290	0.25	0.59	0.48
Film-Noir	112	0.37	0.58	0.55
Action	392	0.39	0.57	0.52
Fantasy	246	0.35	0.55	0.50
Romance	454	0.37	0.54	0.49
Drama	418	0.36	0.54	0.51
Horror	310	0.37	0.54	0.50
Children's	396	0.39	0.53	0.48
Comedy	412	0.27	0.53	0.51
Thriller	432	0.34	0.53	0.50
Sci-Fi	390	0.35	0.53	0.48
Crime	400	0.37	0.53	0.50
War	370	0.26	0.52	0.50
Mystery	222	0.38	0.51	0.48
Western	164	0.35	0.50	0.53
Animation	248	0.35	0.49	0.50
Documentary	32	0.34	0.46	0.49

F User-Specific Conditioning Diagnostics

This section provides supporting Qwen3-4B PersonaGrid comparisons for Sections 5.2 and 5.5.

F.1 Same-Query User Pairs

We pair test users who receive the same query but have different correct answers. Both queried factors are absent for each member of the pair. The seen partition contains 939 pairs covering 77 users and 76 topics; the unseen partition contains 897 pairs covering 78 users and 74 topics. Table 7 partitions outcomes into both, exactly one, and neither user answered correctly. Users and topics can occur in multiple pairs. Two independent uniform four-way guesses give reference probabilities of 6.25%, 37.5% and 56.25%, respectively.

Table 7: **Outcomes for users requiring different answers to the same query.** Percent of user pairs.

Partition	Method	Both correct	One correct	Neither correct
Seen	ICL	6.5	41.7	51.8
Seen	Memory-ICL	6.7	43.0	50.3
Seen	Ours	27.3	49.2	23.5
Unseen	ICL	5.2	43.5	51.3
Unseen	Memory-ICL	6.3	43.5	50.2
Unseen	Ours	22.3	51.3	26.4

F.2 Representation Substitution

For each seen test user, the intervention replaces the inferred cluster indicator with one from a randomly selected different cluster. Trained weights, query and candidates remain unchanged. To

measure the direction of the changed prediction, we consider queries for which the substituted cluster's majority training answer differs from the target user's correct answer. The prediction follows that cluster's answer in 92.9% of these cases, retains the user's answer in 2.7%, and chooses another answer in 4.4%. Table 8 contrasts these destinations with accuracy over the full seen test set.

Table 8: **Prediction destinations and accuracy under representation interventions.** Left: predictions when the user's correct answer and the substituted cluster's majority answer differ. Right: accuracy across all seen test queries. Values are percentages.

Prediction destination	Fraction	User representation	Accuracy
Substituted cluster's answer	92.9	Unmodified	53.2
User's own answer	2.7	Permuted among users	24.7
Other answer	4.4	Random different cluster	19.5

Substitution supplies a different group's representation, so the resulting below-chance accuracy reflects a directed mismatch between the supplied representation and the target user. Permuting representations among users also reduces accuracy.

F.3 Gate Ablation with Indicator and Dense Representations

Table 9 compares coefficient constructions while retaining the adapter-target factor G_a. PersonaGrid uses inferred one-hot indicators; ML-1M uses dense SVD representations. Removing topic modulation leaves $U_u \odot G_a$, whereas removing user conditioning leaves $\widetilde{V}_t \odot G_a$.

Table 9: **Gate form and user-representation structure.** Candidate accuracy (%) on PersonaGrid and ML-1M with Qwen3-4B. The latter uses the public-data preprocessing of Table 2. A dash indicates that the corresponding ML-1M ablation was not evaluated.

		PersonaGrid Indicator	ML-1M: dense SVD	
Gate	Coefficient	Seen	Seen	Unseen
CP-gated LoRA	$U_u \odot \widetilde{V}_t \odot G_a$	53.8	**66.9**	**63.5**
Without topic modulation	$U_u \odot G_a$	55.1	65.0	61.4
Additive conditioning	$(U_u + \widetilde{V}_t) \odot G_a$	**55.5**	61.9	61.3
Without user conditioning	$\widetilde{V}_t \odot G_a$	26.4	–	–

With an indicator $U_u = e_c$, CP conditioning selects M_c and scales it by $\widetilde{V}_{t,c} G_{a,c}$ (Equation 5). Without topic modulation, the same direction is scaled by $G_{a,c}$. Additive conditioning combines a user-dependent term with a topic-dependent term:

$$\Delta B^a_{u,t} = \sum_j U_{u,j} G_{a,j} M_j + \sum_j \widetilde{V}_{t,j} G_{a,j} M_j.$$

These user-conditioned forms yield similar PersonaGrid accuracy, while omitting U_u reduces accuracy toward chance. With a dense U_u, several coordinates contribute simultaneously; CP conditioning lets the topic modulate each user's active coordinates. Its higher ML-1M accuracy is consistent with the value of these joint user–topic coefficients.

G Prompt Length and Conditional Parameter Cost

Table 10 separates prompt length from adapter size. Token counts use the Qwen3 tokenizer on the user turn; they do not include all system-template overhead. On the primary PersonaGrid task, query-only Ours uses approximately 15 tokens, versus 157 for ICL. Along the observation-density study, the query-only user turn stays near 14 tokens while full-history prompts grow with the observed record. These measurements describe input length; inference latency was not measured.

Table 10: **Prompt and parameter costs for Qwen3-4B.** User-turn token counts and reported parameter counts. Conditional parameters are additional to ordinary LoRA and shared across users.

Quantity	Setting	Value
User-turn tokens	PersonaGrid: Ours, query-only	15
User-turn tokens	PersonaGrid: ICL, history	157
User-turn tokens	Observation density: full history, 8%	671
User-turn tokens	Observation density: full history, 32%	2,288
User-turn tokens	Observation density: full history, 64%	4,554
User-turn tokens	Observation density: query-only	14
LoRA parameters	Rank 8; query/value targets; 36 layers	2.95M
Conditional parameters	Shared P and Q bases	0.33M
Conditional parameters	Topic factors and neighbor mixing	6.0k
Conditional parameters	Adapter-target factors G	4.6k
User representation	PersonaGrid inferred cluster identifier	1 integer/user

The conditional parameter blocks total approximately 0.34M, compared with 2.95M ordinary LoRA parameters. Shared directions and conditioning factors avoid storing a separate adapter for each user. The one-integer user state describes PersonaGrid's inferred cluster encoding; it excludes shared model storage and the upstream work needed to infer that assignment.