PROTOADAPTER: DATA-CENTRIC VERTICAL ADAPTATION FOR PRETRAINED GENERATIVE RECOMMENDERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative recommenders must adapt pretrained models to new verticals, where item-token alignment alone does not teach how users transition between items and target-domain interactions are often scarce. Existing continual pretraining and supervised fine-tuning primarily learn from individual histories, leaving two complementary signals implicit: recurring behavioral patterns shared across the domain and history-specific evidence that identifies a user's current position within such a pattern. We propose ProtoAdapter, an offline data construction framework for vertical adaptation that makes both signals explicit. ProtoAdapter builds a bank of representative behavioral prototypes from the target-domain corpus, retrieves compatible prototypes for each user history, and selects supporting events from that history. A teacher model then organizes them into a structured trace containing the vertical pattern, observed evidence, and transition bridge, with the target action appended as the supervised answer. Trace-augmented and direct-answer examples are jointly used for continual pretraining and supervised fine-tuning, while deployment retains direct item-token generation without requiring traces. Experiments on several recommendation datasets with Qwen3 and OneRec show consistent improvements across ranking metrics. The gains become more pronounced in low-data settings: on KuaiRand, using only 10% of the training data yields a relative improvement of up to 29.7%. Further analyses verify the effects of trace construction and demonstrate that the proposed two-scale supervision improves data efficiency by connecting reusable domain transitions with sequence-specific evidence. Training code: `https://anonymous.4open.science/r/ProtoAdapter/`.

1 INTRODUCTION

Generative recommenders represent items as semantic identifiers and predict future interactions through autoregressive decoding (Rajput et al., 2023; Zhou et al., 2025). Pretraining offers a reusable model, while deployment in a new vertical requires adapting to that catalog and its behavioral transitions. Item-token alignment makes the new catalog representable. Learning how users navigate between items in that catalog is a separate challenge, especially when target-domain interactions are scarce (Gururangan et al., 2020; Hou et al., 2022; Chen et al., 2024).

Conventional continual pretraining (CPT) and supervised fine-tuning (SFT) learn these transitions from individual histories. Two useful signals remain implicit: recurring patterns shared across the domain and observations that identify the current user's position within a pattern. Figure 1 illustrates the distinction. A shoe, a watch, and a bottle appear in several histories, separated by books, headphones, and other interactions. These items suggest a progression from exercise preparation to hydration. A corpus-level pattern captures that progression; the shoe and watch in a particular history identify the relevant local evidence.

Existing adaptation methods improve item representations, domain transfer, and recommendation-oriented instruction tuning (Hou et al., 2023; Li et al., 2023; Bao et al., 2023; Liao et al., 2024). Rationale distillation provides an additional training signal by verbalizing preferences or intermediate reasoning (Hsieh et al., 2023; Wang et al., 2024; Wadhwa et al., 2024). However, these advances leave a practical gap in vertical adaptation: an aligned item representation or an example-level rationale does not explicitly connect a reusable domain transition to the observations that make it relevant to a particular user. Sparse histories make this connection especially difficult to learn from target labels

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.

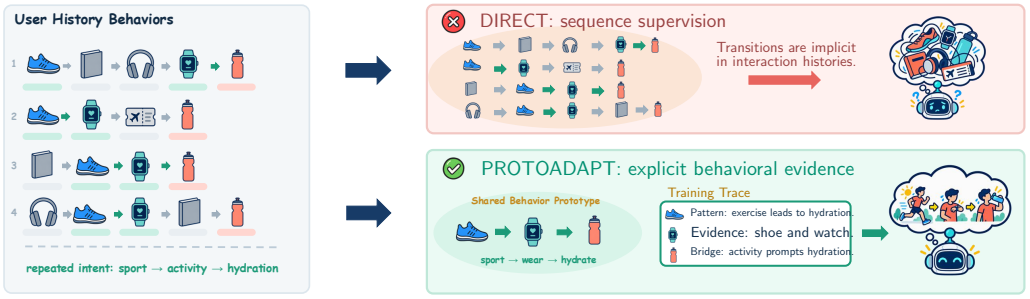


Figure 1: PROTOADAPTER illustrates a shared shoe–watch–bottle transition. Books, headphones, and other interactions separate the relevant events across histories (left). With direct adaptation, this transition is difficult to infer from scattered events alone (upper right). PROTOADAPTER provides a shared behavioral prototype and a trace grounded in selected events (lower right). This structure helps the model learn the preference-transition logic in a new context. The bottle represents the next action; its item-token label is appended as y_u.

alone. Our approach organizes the evidence used to construct such supervision. It combines recurring transitions extracted across users with evidence selected inside each history, giving the teacher an explicit basis for connecting a domain pattern to a prediction context.

To make domain transitions learnable from limited histories, we introduce PROTOADAPTER, an offline data construction framework for vertical adaptation. We build a representative behavioral prototype bank so that short histories can reuse transition patterns supported elsewhere in the target corpus. To establish where the current user fits within a pattern, we retrieve compatible prototypes and select supporting events from that history. A teacher connects the two through three trace slots: the vertical pattern, observed evidence, and transition bridge. The labeled target action is then appended as the supervised answer. Trace-augmented and direct-answer examples train the same recommender during CPT and SFT, allowing the deployment interface to remain direct item-token generation from the original user interaction history.

Experiments on All_Beauty, KuaiRec, and KuaiRand evaluate this augmentation on Qwen3-1.7B and OneRec-1.7B. Both augmented backbones improve over their respective direct-training baselines on all three ranking metrics. The data-efficiency study shows larger relative gains at smaller training fractions; trace-construction variants and distribution diagnostics examine how the construction affects performance. Our contributions are:

- We propose PROTOADAPTER, an offline data construction framework that makes domain-level behavioral patterns and sequence-specific evidence explicit for vertical adaptation of pretrained generative recommendation models.
- We address the missing connection between reusable transitions and sparse individual histories by retrieving behavioral prototypes, selecting supporting events, and synthesizing structured training traces. Joint trace and direct-answer supervision retains direct item-token generation at deployment.
- Experiments on three datasets and two backbones show consistent ranking improvements, with a 29.7% relative gain at 10% training data on KuaiRand. Construction ablations and distribution diagnostics support the value of coherent, domain-matched traces for data-efficient adaptation.

2 RELATED WORK

Generative and transferable recommendation. P5 casts recommendation tasks as language modeling, while TIGER retrieves items by generating semantic identifiers (Geng et al., 2022; Rajput et al., 2023). OpenOneRec develops this approach into a pretrained foundation model that can be transferred to downstream recommendation tasks (Zhou et al., 2025). These advances make a pretrained generative backbone available for vertical adaptation. A new catalog must then be made accessible to that backbone. Universal sequence representations, vector-quantized item representations, and language-based item modeling address this issue by providing transferable representations (Hou et al., 2022; 2023; Li et al., 2023). Cross-domain methods further exploit shared interests and domain-

specific representations (Cao et al., 2022; Bian et al., 2025). Our setting begins with an aligned item vocabulary. The remaining challenge is to teach the backbone how users progress through the new domain, particularly when the relevant transitions are dispersed across sparse histories. PROTOADAPTER addresses this challenge by constructing supervision that explicitly pairs a recurring transition with the events supporting it in each history.

Rationale supervision for recommendation. Chain-of-thought prompting and rationale distillation use intermediate text to expose information beyond an answer label (Wei et al., 2022; Zelikman et al., 2022; Hsieh et al., 2023). In recommendation, RDRec distills preference and item information from reviews (Wang et al., 2024); OneReason constructs recommendation-oriented reasoning data, and OneRec-Think studies explicit reasoning in generative recommendation (OneRec Team et al., 2026; Liu et al., 2026). These studies motivate richer supervision, but also raise the question of what a useful trace should contain. Wadhwa et al. (2024) show that rationale order and key tokens affect distillation, with some gains surviving permutation. Thus, the presence of intermediate text alone is insufficient to explain an improvement. We specify the evidence available to the teacher: retrieved domain prototypes supply a transition pattern, and selected history events establish its relevance to the user. Our shuffled, cross-domain, and raw-pattern ablations test how this structured construction contributes to recommendation quality.

Data organization for adaptation. Domain-adaptive pretraining exposes language models to a target distribution (Gururangan et al., 2020). Domain reweighting and data selection improve the composition of training corpora (Xie et al., 2023a;b). These methods primarily determine which examples receive training attention. PROTOADAPTER additionally changes what supervision accompanies a retained history. A cached prototype bank lets many histories share a concise account of a domain transition, while local evidence keeps each trace tied to its own prediction context. This reuse is particularly relevant when target-domain examples are scarce: individual histories can draw on patterns supported elsewhere in the training corpus. All retrieval and synthesis take place before model training, so the resulting recommender retains direct item-token generation at deployment.

3 PROBLEM FORMULATION

We study the adaptation of a pretrained generative recommender to a target vertical, such as beauty products or short videos. Given the items a user has interacted with, the adapted model must recommend relevant items from that vertical. The catalog is already mapped to item-token sequences, so the adaptation problem concerns learning target-domain preferences and transitions through the available target-domain training data.

Let $\mathcal{D}_t$ be the target-domain training set and $\mathcal{I}_t$ its item catalog. An example consists of a chronological history $h_u = (i_1, \ldots, i_T)$ and a labeled target encoded as an answer-token sequence y_u. Here, an "answer" is the item-token output used for recommendation. Direct adaptation trains the pretrained model to predict this output from the history:

$$\mathcal{L}_{\text{direct}} = -\mathbb{E}_{(h_u, y_u)\sim\mathcal{D}_t} \log p_\theta^D(y_u \mid h_u). \tag{1}$$

This objective teaches which output follows a history, while the behavioral connection between the observed events and that output remains implicit. Our goal is to make that connection available as additional training supervision.

PROTOADAPTER constructs a trace τ_u from retrieved behavioral prototypes $p_u \subseteq \mathcal{P}$ and supporting history events $e_u \subseteq h_u$:

$$\tau_u = G_\phi(h_u, p_u, e_u; y_u), \qquad \mathcal{L}_{\text{trace}} = -\mathbb{E}_u \log p_\theta^J(\tau_u, y_u \mid h_u). \tag{2}$$

The construction first selects evidence $e_u = S(h_u, y_u)$, then retrieves a set of M compatible prototypes $p_u = R(h_u, e_u, \mathcal{P})$. The offline teacher G_ϕ uses these inputs to explain a transition. The semicolon in Eq. 2 separates the training label from the historical evidence: the label guides offline construction, and its item tokens are appended after the trace as the supervised answer.

The superscripts D and J denote two output formats of the same model with shared parameters θ: a direct answer and a trace followed by an answer. This distinction connects the training design to deployment. We train with both formats so that the model can learn from explicit behavioral structure while continuing to predict through $p_\theta^D(y \mid h)$ when serving a recommendation.

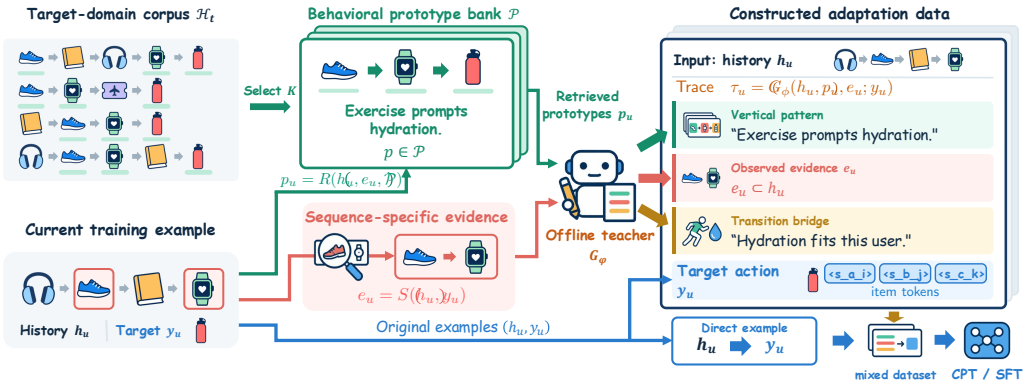


Figure 2: Offline construction of adaptation data. Target-domain histories form a behavioral prototype bank $\mathcal{P}$ (green). A history h_u supplies retrieved prototypes p_u and local evidence e_u (coral). The teacher generates a trace τ_u with three slots: vertical pattern, observed evidence, and transition bridge. Appending y_u forms the four-slot record. The blue branch carries direct-answer examples into the training set for CPT and SFT. All construction modules run offline; deployment uses direct item-token decoding. Product symbols are illustrative.

4 PROTOADAPTER

A single interaction history often contains too little evidence to reveal a complete behavioral transition. Even when a recurring pattern is visible across the corpus, unrelated events can make its relevance to a particular user difficult to identify. Our construction addresses these two obstacles together. We first collect reusable domain transitions in a prototype bank, then locate the events that connect each history to a retrieved transition. Figure 2 shows how an offline teacher turns this paired evidence into explicit supervision for model adaptation.

4.1 TWO-SCALE EVIDENCE CONSTRUCTION

Behavioral prototype bank: making domain transitions reusable. Learning every transition independently from each history is inefficient when many users pass through similar stages. In the running example, several histories move from exercise preparation toward hydration, although the relevant items occur among books, headphones, and other interactions. A prototype condenses the common progression into a reusable description. Its role is to provide a behavioral context that a short history may only partially reveal.

We partition training histories into local temporal fragments z_j and represent each fragment by v_j, using its item tokens, available metadata, and observed transitions. We express the coverage–diversity criterion for selecting K fragments as (Sener & Savarese, 2018):

$$J(P) = \sum_{z_j \in P} \mathrm{cov}(z_j, \mathcal{Z}) - \lambda \sum_{z_a \neq z_b \in P} \mathrm{sim}(v_a, v_b), \qquad |P| = K. \tag{3}$$

The candidate set $\mathcal{Z}$ contains the training fragments. The coverage term favors fragments that represent the transition space, and the similarity penalty discourages spending the bank on near duplicates. Representative, diverse fragments allow the bank to support different user interests and successive stages of behavior.

An offline teacher summarizes each selected fragment as a concise behavioral description. We retain the supporting fragment and event pointers with each description, giving the construction stage access to both the pattern and its observed basis. The bank is built exclusively from the training split and cached for subsequent retrieval. In this way, a pattern discovered across users becomes a shared training resource that can support many individual examples.

Sequence-specific evidence: locating the user within a pattern. A shared fitness pattern may cover equipment selection, movement instruction, and actual exercise. Recent home-workout events

and earlier equipment events suggest different next steps. Evidence selection identifies the observations that locate the user within this progression.

The selector scores events by recency, continuity, domain compatibility, and the relevance of historical events to the training target:

$$a_{u,t} = w_r \,\mathrm{recency}(i_t) + w_c \,\mathrm{continuity}(i_t, h_u) + w_d \,\mathrm{domainfit}(i_t, \mathcal{D}_t) + w_s \,\mathrm{saliency}(i_t, y_u). \tag{4}$$

These terms favor recent observations, events connected to the surrounding history, and evidence relevant to the domain and training target. The selected events form e_u in chronological order, preserving how the observed behavior develops over time. This follows the principle that the events within a history contribute unequally to the user's current interest (Zhou et al., 2018; 2019; Li et al., 2019; Cen et al., 2020).

The training label helps locate supporting events during offline construction. These inputs are used only to prepare training data; deployment requires the original history and adds no evidence-selection step. Together, the two scales answer complementary questions: the prototype describes a transition that recurs in the domain, and the selected events explain why that transition applies here.

4.2 Retrieval and Trace Synthesis

Matching shared patterns to local evidence. The bank contains several possible transitions, so a useful trace requires selecting a pattern compatible with the current history. We score each prototype using history similarity and agreement with the selected evidence:

$$s(h_u, e_u, p) = \alpha \,\mathrm{sim}(q(h_u), q(p)) + (1-\alpha)\,\mathrm{match}(e_u, p), \qquad p_u = \mathrm{TopM}_{p\in\mathcal{P}}\, s(h_u, e_u, p). \tag{5}$$

The representation q maps histories and prototypes into a shared space; match compares their supporting events. History similarity identifies broadly relevant candidates, while the evidence term connects those candidates to the observations selected for this example. Retrieving M compatible prototypes connects corpus-level reuse to the local context before the teacher generates a trace.

A trace with an explicit behavioral connection. A generic preference summary can describe a history without explaining its next transition. We therefore ask the teacher to organize the retrieved descriptions and selected events into three connected slots:

1. **Vertical pattern:** the recurring transition expressed by the retrieved prototypes.
2. **Observed evidence:** the selected events that support this pattern in the current history.
3. **Transition bridge:** how the observed behavior leads to a plausible next interest.

The first slot supplies shared context, the second anchors it to observations, and the third makes their predictive connection explicit. For the shoe–watch–bottle example, exercise-related events establish the local context, and the bridge connects this context to hydration. The training signal therefore includes an account of the transition as well as the item to predict.

The teacher receives the history, prototype descriptions, evidence pointers, and selected events. It uses the label for consistency checks, while the generated trace excludes target identifiers and future events. Traces that copy target identifiers or introduce unsupported evidence are rejected; their histories remain as direct-answer examples. These checks keep the explanatory text tied to observations that precede the target. Appending y_u to the accepted trace yields the four-slot record in Figure 2, with the target action kept as the supervised answer.

This construction connects rationale supervision (Hsieh et al., 2023; Wang et al., 2024) to explicit evidence retrieval. Shared patterns recur across training examples, while selected events establish their relevance within each example. Appendix C illustrates both roles with complete records.

4.3 Training and Deployment

The constructed records must improve the recommender that will actually be deployed. We therefore combine trace-augmented examples $h_u \mapsto (\tau_u, y_u)$ with direct-answer examples $h_u \mapsto y_u$ and

Table 1: Dataset statistics after preprocessing.

Dataset	Users	Items	Interactions	Mean length
All_Beauty	9,159	19,450	39,427	4.30
KuaiRec	7,176	10,727	8,031,791	1,119.26
KuaiRand	22,864	5,832	649,566	28.41

express the combined training objective as

$$\mathcal{L} = \overline{\mathcal{L}}_{\text{direct}} + \beta \overline{\mathcal{L}}_{\text{trace}}. \tag{6}$$

The bars denote normalized losses for the two portions, and β weights trace supervision. Both portions update the same model. The direct examples train its deployed prediction format, while the trace examples supervise the connection between a domain pattern, local evidence, and a target action. In the full configuration, constructed data is used in both CPT and SFT, exposing the model to this behavioral structure throughout adaptation.

The trace objective factors as

$$-\log p_\theta^J(\tau_u, y_u \mid h_u) = -\log p_\theta^J(\tau_u \mid h_u) - \log p_\theta^J(y_u \mid \tau_u, h_u). \tag{7}$$

The two terms teach the explanation and the answer conditioned on it. Direct-answer examples additionally train prediction from the original history, keeping the serving format part of adaptation.

At inference, the recommender receives a history and directly decodes item tokens. Prototype retrieval, evidence selection, and teacher generation have already been completed offline, so these modules introduce no additional serving operations.

Appendix D analyzes how joint trace-and-answer fit and agreement with the direct decoder affect deployment error, and separates prototype coverage, retrieval, and teacher-description errors.

5 Experiments

5.1 Setup

Datasets and metrics. Our evaluation asks whether making behavioral transitions explicit improves adaptation, whether the benefit is strongest with limited data, and which parts of the construction account for it. We evaluate on the product domain All_Beauty[1] and two short-video datasets, KuaiRec[2] and KuaiRand[3]. Table 1 summarizes the processed interaction data. Mean length is interactions divided by users, before model-input truncation. We report Pass@1, HR@5, and NDCG@5 in percent: first-prediction success, at least one relevant item in the top five, and rank-discounted normalized gain (Järvelin & Kekäläinen, 2002). Experiments use multiple runs, and the reported improvements are statistically significant.

Methods. The main experiment uses the full training set for every method. We compare SASRec, Mamba4Rec, TTT4Rec, HSTU, and TIGER, and apply ProtoAdapter to Qwen3-1.7B and OneRec-1.7B. Each augmented backbone uses constructed data in both CPT and SFT. The construction-budget analysis varies prototype-bank size, retrieval count, and trace length around $K = 32$, $M = 4$, and a 300-character cap. Inference directly decodes item tokens from the original history. Baseline references and metric definitions are collected in Appendix A.

5.2 Overall Recommendation Quality

We first test whether the constructed supervision improves recommendation across domains and backbone types. Comparing each augmented backbone with its own direct-training baseline isolates the practical benefit of the data construction, while the sequential and generative baselines provide the broader ranking context. ProtoAdapter improves all three metrics on both backbones across

[1] http://snap.stanford.edu/data/amazon/productGraph/categoryFiles/

[2] https://kuairec.com/

[3] https://kuairand.com/

Table 2: Full-data recommendation results (%). P@1, H@5, and N@5 denote Pass@1, HR@5, and NDCG@5. Bold and underline mark the best and second-best result in each column, respectively. Blue rows use our construction; PA denotes PROTOADAPTER.

Method	All_Beauty			KuaiRec			KuaiRand		
	P@1	H@5	N@5	P@1	H@5	N@5	P@1	H@5	N@5
SASRec	2.61	7.24	4.89	4.89	9.63	7.64	0.92	3.36	2.34
Mamba4Rec	2.73	7.49	5.07	4.65	9.72	7.59	1.03	3.35	2.37
TTT4Rec	2.68	7.56	5.05	4.65	9.47	7.45	1.02	3.32	2.36
HSTU	3.08	8.46	5.72	5.38	10.55	8.38	1.21	3.81	2.72
TIGER	3.20	8.73	5.92	<u>5.53</u>	10.84	8.59	1.30	3.96	2.86
Qwen3	5.23	14.15	9.71	5.10	11.15	8.50	1.57	5.55	3.52
Qwen3 + PA	<u>5.65</u>	<u>15.23</u>	<u>10.41</u>	5.43	11.65	8.91	<u>1.71</u>	<u>5.96</u>	<u>3.80</u>
OneRec	5.48	14.82	10.12	5.51	<u>12.09</u>	<u>9.19</u>	1.64	5.72	3.64
OneRec + PA	**5.87**	**15.63**	**10.76**	**5.92**	**12.77**	**9.77**	**1.80**	**6.19**	**3.98**

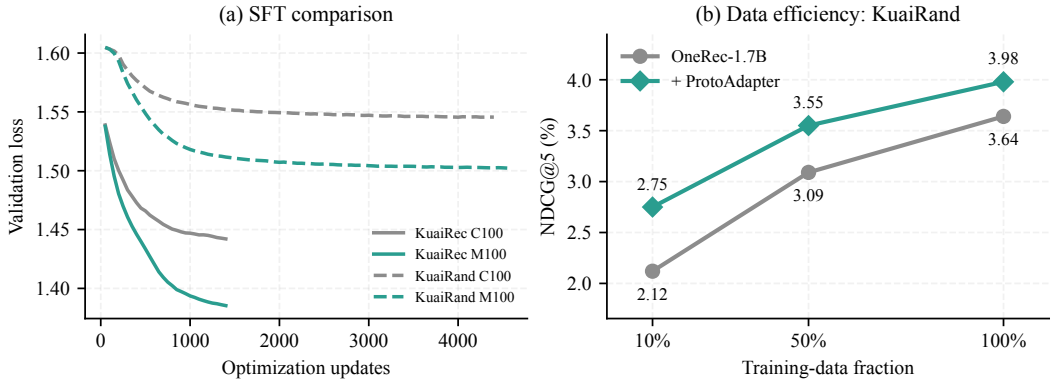


Figure 3: SFT comparison and data efficiency. (a) SFT validation losses: C100 uses direct supervision; M100 adds traces only in SFT. Both SFT runs start from the same direct-supervision CPT checkpoint. Each loss measures its own supervision objective. Solid/dashed lines denote KuaiRec/KuaiRand. (b) KuaiRand NDCG@5 for OneRec-1.7B and the complete CPT+SFT augmentation at three fractions of the available training data.

all datasets (Table 2). OneRec-1.7B gains 0.64, 0.58, and 0.34 NDCG@5 percentage points on All_Beauty, KuaiRec, and KuaiRand, corresponding to relative improvements of 6.3%, 6.3%, and 9.3%. Qwen3-1.7B gains 0.70, 0.41, and 0.28 points. The same construction improves both the recommendation backbone and the general language model. OneRec with PROTOADAPTER achieves the highest value in every column.

Gains in Pass@1, HR@5, and NDCG@5 improve both the first prediction and the top-five list. Both a general language-model backbone and a pretrained recommendation backbone benefit, supporting explicit organization of domain transitions and local evidence as a data construction strategy for target-domain model adaptation.

5.3 TRAINING DYNAMICS AND DATA EFFICIENCY

We next examine how the benefit relates to learning during SFT and to the amount of target-domain training data. Figure 3(a) compares the SFT trajectories of the direct and trace-supervised controls. C100 applies direct supervision only. M100 applies trace supervision only during SFT, starting from the same CPT checkpoint as C100. Both losses decline as training proceeds, and the trace-supervised run reaches a lower reported validation loss in each video domain. This comparison shows that the structured targets can be learned effectively during SFT; the ranking comparison below measures

Table 3: Construction ablations on OneRec-1.7B: NDCG@5 (%). The full method and the shuffled, cross-domain, and raw-pattern variants use constructed data in both CPT and SFT.

Variant	Trace supervision	KuaiRec	KuaiRand
OneRec-1.7B	–	9.19	3.64
Traces in SFT only	SFT	<u>9.50</u>	<u>3.76</u>
Shuffled traces	CPT+SFT	9.24	3.60
Cross-domain traces	CPT+SFT	9.11	3.54
Raw patterns	CPT+SFT	8.93	3.58
Full PROTOADAPTER	CPT+SFT	**9.77**	**3.98**

Table 4: Distribution diagnostics for OneRec with and without PROTOADAPTER. $\Delta H = H_{\text{pred}} - H_{\text{target}}$ and retrieval MI use bits. SID ratio counts distinct predicted versus target first-layer SIDs; Tail R@5 is long-tail Recall@5 (%). PA denotes PROTOADAPTER.

Dataset	Method	JS ↓	ΔH	SID ratio	Tail R@5 ↑	MI
KuaiRec	OneRec	0.286	-0.74	0.56	2.64	–
KuaiRec	OneRec + PA	0.253	-0.58	0.62	2.93	0.18
KuaiRand	OneRec	0.347	-0.93	0.48	1.16	–
KuaiRand	OneRec + PA	0.319	-0.76	0.53	1.34	0.12

the resulting recommendation benefit. Figure 3(b) evaluates the complete method at 10%, 50%, and 100% training data. NDCG@5 rises from 2.12 to 2.75, 3.09 to 3.55, and 3.64 to 3.98. The relative gains are 29.7%, 14.9%, and 9.3%, respectively. Constructed traces are especially useful when training data is limited. At 50% data, the augmented model reaches 3.55, close to the 3.64 achieved by the baseline with full training data.

The low-data gain is also larger in absolute terms: 0.63 points at 10% data, compared with 0.34 points at full data. Thus, the stronger relative gain does not arise solely from a smaller denominator. These results fit the motivation for sharing behavioral prototypes across histories: a short or sparsely observed history can draw on a domain transition expressed elsewhere in the training corpus. The full method improves all three training fractions, with its largest benefit where direct adaptation has the least data. Appendix B lists the corresponding values.

5.4 TRACE CONSTRUCTION ABLATIONS

The main results establish a benefit from the complete pipeline. We now ask whether coherent, domain-matched construction matters, and whether using the constructed data throughout adaptation adds value beyond SFT alone. Table 3 varies the stage or content of trace supervision on OneRec-1.7B. Using traces only in SFT improves the baseline by 0.31/0.12 points on KuaiRec/KuaiRand. Applying them in both stages adds another 0.27/0.22 points (Table 3). Shuffling the traces yields 9.24/3.60; cross-domain traces yield 9.11/3.54. Removing the LLM summary and retaining raw patterns yields 8.93/3.58. The complete construction outperforms all three content variants in both domains.

The stage comparison supports using constructed supervision in both CPT and SFT. The content comparisons further show why the construction is useful: shuffled traces largely remove the gain, cross-domain traces fall below the direct baseline, and raw patterns underperform the teacher-organized traces. Together, these results favor traces that preserve a coherent connection between a target-domain pattern and the current history. Merely attaching additional text is insufficient to recover the full method's advantage.

5.5 PREDICTION DISTRIBUTION AND LONG-TAIL RETRIEVAL

We examine prediction distributions to test whether the ranking gains accompany broader improvements in how the model covers target-domain behavior (Table 4). JS divergence decreases from 0.286

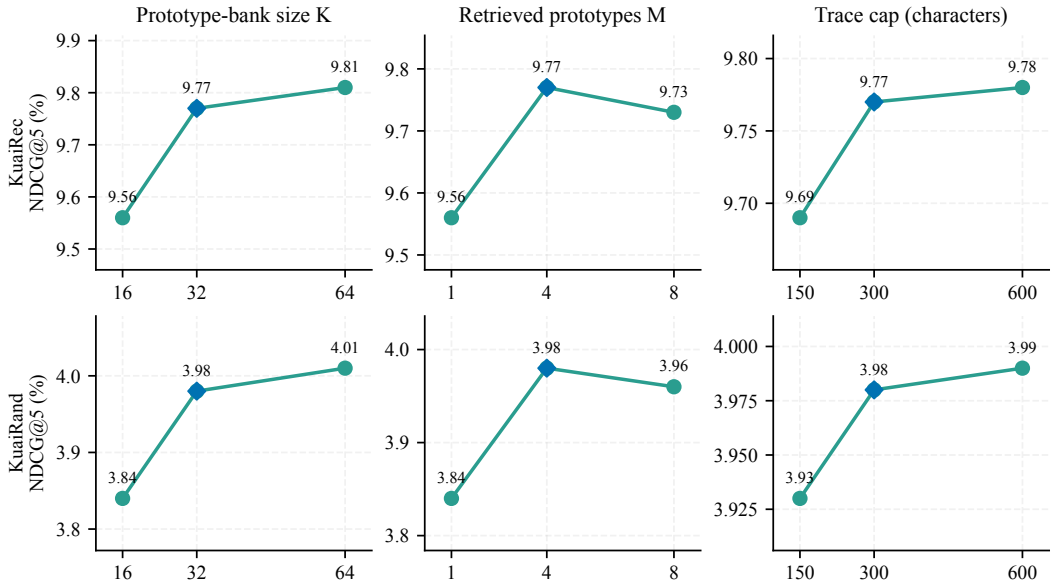


Figure 4: Construction-budget sensitivity on KuaiRec (top) and KuaiRand (bottom). Each sweep varies one parameter; diamonds mark the reference configuration.

to 0.253 on KuaiRec and from 0.347 to 0.319 on KuaiRand, while the absolute entropy gaps become smaller. These two changes indicate closer distributional agreement. The higher first-layer SID ratios record a broader range of predicted first-layer identifiers. Long-tail Recall@5 rises by 0.29/0.18 points, or 11.0%/15.5% relative. Retrieval MI is 0.18/0.12 bits for the constructed-data pipeline; the baseline has no prototype retrieval operation. The combination of closer distributional agreement and higher long-tail recall suggests that the improvement reaches less frequent target items as well as aggregate ranking quality. Appendix A defines the distribution quantities.

5.6 Construction Budgets

Finally, we test whether the construction requires large banks, many retrieved prototypes, or long traces. Each sweep changes one control around $K = 32$, $M = 4$, and a 300-character trace cap. Increasing K from 16 to 32 improves NDCG@5 by 0.21/0.14 points; doubling it again to 64 adds 0.04/0.03. With $K = 32$, retrieving four prototypes performs best among $M \in \{1, 4, 8\}$. Raising the trace cap from 150 to 300 characters gains 0.08/0.05 points, while 600 characters adds 0.01 in each domain. The reference configuration yields 9.77/3.98, matching OneRec + PA in Table 2, and is close to the best results within the tested budgets. Figure 4 summarizes the sweeps, and Appendix B reports the complete values.

The sweeps favor focused evidence: larger banks give diminishing gains, four retrieved prototypes outperform eight, and the 300-character cap captures almost all of the measured benefit of 600 characters. The reference configuration therefore offers strong performance with moderate budgets for offline retrieval and trace construction.

6 Conclusion

ProtoAdapter organizes vertical-adaptation data around two complementary evidence sources: behavioral prototypes shared across users and observations selected within each history. Their structured traces improve both Qwen3-1.7B and OneRec-1.7B while retaining direct decoding at inference. The experiments show consistent ranking gains, stronger relative improvements under limited data, and better long-tail retrieval. The ablations identify coherent traces that match the target domain as an important part of the improvement. These findings support evidence organization as a practical way to adapt generative recommenders to a new vertical.

AI USE STATEMENT

Generative AI tools assisted with writing and language polishing, literature retrieval and discovery, and research ideation and execution, including experimental planning, code development, and analysis. The authors reviewed the AI-assisted material and take responsibility for the manuscript and its reported research conclusions.

ETHICS STATEMENT

This work studies adaptation of generative recommenders using public product-review and short-video interaction datasets. Behavioral histories can contain sensitive preferences, and synthesized traces may reproduce exposure biases in the underlying data. Applications should respect dataset access conditions, protect user information, and evaluate how recommendations distribute exposure across items and user groups. Our evaluation concerns offline recommendation quality; deployment decisions also require assessment in the intended use setting.

REPRODUCIBILITY STATEMENT

Training code is available at `https://anonymous.4open.science/r/ProtoAdapter/`. Section 4 specifies the data construction and training objectives, and Section 5 describes the experimental comparisons. Appendix A defines the evaluation measures, Appendix B reports the complete data-efficiency and construction-budget results, Appendix C illustrates the training-record format, and Appendix D provides the formal analysis and proofs.

REFERENCES

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, 2023. doi: 10.1145/3604915.3608857. URL https://doi.org/10.1145/3604915.3608857.

Qingtian Bian, Marcus de Carvalho, Tieying Li, Jiaxing Xu, Hui Fang, and Yiping Ke. ABXI: Invariant Interest Adaptation for Task-Guided Cross-Domain Sequential Recommendation. In *Proceedings of the ACM on Web Conference 2025*, pp. 3183–3192, 2025. doi: 10.1145/3696410.3714819. URL https://doi.org/10.1145/3696410.3714819.

Jiangxia Cao, Xin Cong, Jiawei Sheng, Tingwen Liu, and Bin Wang. Contrastive Cross-Domain Sequential Recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 138–147, 2022. doi: 10.1145/3511808.3557262. URL https://doi.org/10.1145/3511808.3557262.

Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. Controllable Multi-Interest Framework for Recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2942–2951, 2020. doi: 10.1145/3394486.3403344. URL https://doi.org/10.1145/3394486.3403344.

Shu Chen, Zitao Xu, Weike Pan, Qiang Yang, and Zhong Ming. A Survey on Cross-Domain Sequential Recommendation. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 7989–7998, 2024. doi: 10.24963/ijcai.2024/884. URL https://doi.org/10.24963/ijcai.2024/884.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2 edition, 2006. doi: 10.1002/047174882X. URL https://doi.org/10.1002/047174882X.

Chongming Gao, Shijun Li, Wenqiang Lei, Jiawei Chen, Biao Li, Peng Jiang, Xiangnan He, Jiaxin Mao, and Tat-Seng Chua. KuaiRec: A Fully-observed Dataset and Insights for Evaluating Recommender Systems. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 540–550, 2022a. doi: 10.1145/3511808.3557220. URL https://doi.org/10.1145/3511808.3557220.

Chongming Gao, Shijun Li, Yuan Zhang, Jiawei Chen, Biao Li, Wenqiang Lei, Peng Jiang, and Xiangnan He. KuaiRand: An Unbiased Sequential Recommendation Dataset with Randomly Exposed Videos. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 3953–3957, 2022b. doi: 10.1145/3511808.3557624. URL https://doi.org/10.1145/3511808.3557624.

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt & Predict Paradigm (P5). In *Proceedings of the 16th ACM Conference on Recommender Systems*, pp. 299–315, 2022. doi: 10.1145/3523227.3546767. URL https://doi.org/10.1145/3523227.3546767.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8342–8360, 2020. doi: 10.18653/v1/2020.acl-main.740. URL https://doi.org/10.18653/v1/2020.acl-main.740.

Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. Towards Universal Sequence Representation Learning for Recommender Systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 585–593, 2022. doi: 10.1145/3534678.3539381. URL https://doi.org/10.1145/3534678.3539381.

Yupeng Hou, Zhankui He, Julian McAuley, and Wayne Xin Zhao. Learning Vector-Quantized Item Representation for Transferable Sequential Recommenders. In *Proceedings of the ACM Web Conference 2023*, pp. 1162–1171, 2023. doi: 10.1145/3543507.3583434. URL https://arxiv.org/abs/2210.12316.

Yupeng Hou, Jiacheng Li, Xiangjun Fu, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. Bridging Language and Items for Retrieval and Recommendation: Benchmarking LLMs as Semantic Encoders. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3251–3265, 2026. doi: 10.18653/v1/2026.acl-long.147. URL https://aclanthology.org/2026.acl-long.147/.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling Step-by-Step! Outperforming Larger Language Models with Less Training Data and Smaller Model Sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 8003–8017, 2023. doi: 10.18653/v1/2023.findings-acl.507. URL https://doi.org/10.18653/v1/2023.findings-acl.507.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated Gain-Based Evaluation of IR Techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002. doi: 10.1145/582415.582418. URL https://doi.org/10.1145/582415.582418.

Wang-Cheng Kang and Julian McAuley. Self-Attentive Sequential Recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 197–206, 2018. doi: 10.1109/ICDM.2018.00035. URL https://doi.org/10.1109/ICDM.2018.00035.

Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. Multi-Interest Network with Dynamic Routing for Recommendation at Tmall. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 2615–2623, 2019. doi: 10.1145/3357384.3357814. URL https://doi.org/10.1145/3357384.3357814.

Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley. Text Is All You Need: Learning Language Representations for Sequential Recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1258–1267, 2023. doi: 10.1145/3580305.3599519. URL https://doi.org/10.1145/3580305.3599519.

Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang, and Xiangnan He. LLaRA: Large Language-Recommendation Assistant. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1785–1795, 2024. doi: 10.1145/3626772.3657690. URL https://doi.org/10.1145/3626772.3657690.

Jianhua Lin. Divergence Measures Based on the Shannon Entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991. doi: 10.1109/18.61115. URL https://doi.org/10.1109/18.61115.

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.

Zhanyu Liu, Shiyao Wang, Xingmei Wang, Rongzhou Zhang, Jiaxin Deng, Honghui Bao, Jinghao Zhang, Wuchao Li, Pengfei Zheng, Xiangyu Wu, Yifei Hu, Qigen Hu, Xinchen Luo, Lejian Ren, Zixing Zhang, Qianqian Wang, Kuo Cai, Yunfan Wu, Hongtao Cheng, Zexuan Cheng, Lu Ren, Huanjie Wang, Yi Su, Ruiming Tang, Kun Gai, and Guorui Zhou. OneRec-Think: In-Text Reasoning for Generative Recommendation. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2664–2681, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.123. URL https://aclanthology.org/2026.acl-long.123/.

OneRec Team, Biao Yang, Boyang Ding, Chenglong Chu, Dunju Zang, Fei Pan, Han Li, Hao Jiang, Honghui Bao, Huanjie Wang, Jian Liang, Jiangxia Cao, Jiao Ou, Jiaxin Deng, Jinghao Zhang, Kun Gai, Lu Ren, Peiru Du, Pengfei Zheng, Rongzhou Zhang, Ruiming Tang, Shiyao Wang, Siyang Mao, Siyuan Lou, Teng Shi, Wei Yuan, Wenlong Xu, Xingchen Liu, Xingmei Wang, Xinqi Jin, Yan Sun, Yan Wang, Yifei Hu, Yingzhi He, Yufei Ye, Yuhao Wang, Yunhao Zhou, Yuqin Dai, Zhao Liu, Zhipeng Wei, Zhixin Ling, Ziming Li, Zixing Zhang, Ziyuan Liu, An Zhang, Changxin Lao, Chaoyi Ma, Chengru Song, Defu Lian, Fan Yang, Guowang Zhang, Hao Peng, Jiayao Shen, Jie Chen, Jun Xu, Junmin Chen, Kun Zhang, Kuo Cai, Mingxing Wen, Minmao Wang, Minxuan Lv, Qi Zhang, Qiang Luo, Sheng Yu, Shijie Li, Shijie Yi, Shuang Yang, Shugui Liu, Shuni Chen, Tinghai Zhang, Tingting Gao, Xiang Wang, Xiangyu Wu, Xiangyu Zhao, Xiao Lv, Xiaoyou Zhou, Xuming Wang, Yong Du, Zejian Zhang, Zhaojie Liu, Zhiyang Zhang, Zhuang Zhuang, Ziqi Wang, and Ziyi Zhao. OneReason Technical Report, 2026. URL https://arxiv.org/abs/2606.06260.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, Maciej Kula, Ed Chi, and Maheswaran Sathiamoorthy. Recommender Systems with Generative Retrieval. In *Advances in Neural Information Processing Systems 36*, pp. 10299–10315, 2023. doi: 10.52202/075280-0452. URL https://doi.org/10.52202/075280-0452.

Ozan Sener and Silvio Savarese. Active Learning for Convolutional Neural Networks: A Core-Set Approach. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1aIuk-RW.

Somin Wadhwa, Silvio Amir, and Byron C. Wallace. Investigating Mysteries of CoT-Augmented Distillation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 6071–6086, 2024. doi: 10.18653/v1/2024.emnlp-main.349. URL https://doi.org/10.18653/v1/2024.emnlp-main.349.

Xinfeng Wang, Jin Cui, Yoshimi Suzuki, and Fumiyo Fukumoto. RDRec: Rationale Distillation for LLM-based Recommendation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 65–74, 2024. doi: 10.18653/v1/2024.acl-short.6. URL https://doi.org/10.18653/v1/2024.acl-short.6.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems 35*, pp. 24824–24837, 2022. doi: 10.52202/068431-1800. URL https://doi.org/10.52202/068431-1800.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. DoReMi: Optimizing Data Mixtures Speeds Up Language Model Pretraining. In *Advances in Neural Information Processing Systems 36*, pp. 69798–69818, 2023a. doi: 10.52202/075280-3059. URL https://doi.org/10.52202/075280-3059.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. Data Selection for Language Models via Importance Resampling. In *Advances in Neural Information Processing Systems 36*, pp. 34201–34227, 2023b. doi: 10.52202/075280-1482. URL https://doi.org/10.52202/075280-1482.

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.

Zhaoqi Yang, Yanan Wang, and Yong Ge. TTT4Rec: A Test-Time Training Approach for Rapid Adaption in Sequential Recommendation, 2024. URL https://arxiv.org/abs/2409.19142.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. STaR: Bootstrapping Reasoning with Reasoning. In *Advances in Neural Information Processing Systems 35*, pp. 15476–15488, 2022. doi: 10.52202/068431-1126. URL https://doi.org/10.52202/068431-1126.

Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Jiayuan He, Yinghai Lu, and Yu Shi. Actions Speak Louder than Words: Trillion-Parameter Sequential Transducers for Generative Recommendations. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 58484–58509. PMLR, 2024. URL https://proceedings.mlr.press/v235/zhai24a.html.

Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. Deep Interest Network for Click-Through Rate Prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1059–1068, 2018. doi: 10.1145/3219819.3219823. URL https://doi.org/10.1145/3219819.3219823.

Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. Deep Interest Evolution Network for Click-Through Rate Prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 5941–5948, 2019. doi: 10.1609/aaai.v33i01.33015941. URL https://doi.org/10.1609/aaai.v33i01.33015941.

Guorui Zhou, Honghui Bao, Jiaming Huang, Jiaxin Deng, Jinghao Zhang, Junda She, Kuo Cai, Lejian Ren, Lu Ren, Qiang Luo, Qianqian Wang, Qigen Hu, Rongzhou Zhang, Ruiming Tang, Shiyao Wang, Wuchao Li, Xiangyu Wu, Xinchen Luo, Xingmei Wang, Yifei Hu, Yunfan Wu, Zhanyu Liu, Zhiyang Zhang, Zixing Zhang, Bo Chen, Bin Wen, Chaoyi Ma, Chengru Song, Chenglong Chu, Defu Lian, Fan Yang, Feng Jiang, Hongtao Cheng, Huanjie Wang, Kun Gai, Pengfei Zheng, Qiang Wang, Rui Huang, Siyang Mao, Tingting Gao, Wei Yuan, Yan Wang, Yang Zhou, Yi Su, Zexuan Cheng, Zhixin Ling, and Ziming Li. OpenOneRec Technical Report, 2025. URL https://arxiv.org/abs/2512.24762.

Table 5: KuaiRand data efficiency, measured by NDCG@5 (%). Relative gain divides the absolute gain by the baseline at the same fraction; PA denotes PROTOADAPTER.

Training data	OneRec	OneRec + PA	Gain (pp)	Relative gain
10%	2.12	2.75	+0.63	+29.7%
50%	3.09	3.55	+0.46	+14.9%
100%	3.64	3.98	+0.34	+9.3%

Table 6: Construction-budget sensitivity: NDCG@5 (%). Bold marks the reference setting. Each sweep varies its named parameter; the other reference settings are $K = 32$, $M = 4$, and a 300-character cap. The unchanged OneRec baseline appears once.

Parameter	Value	KuaiRec	KuaiRand
OneRec	–	9.19	3.64
K	16	9.56	3.84
K	**32**	**9.77**	**3.98**
K	64	9.81	4.01
M	1	9.56	3.84
M	**4**	**9.77**	**3.98**
M	8	9.73	3.96
Trace cap (characters)	150	9.69	3.93
Trace cap (characters)	**300**	**9.77**	**3.98**
Trace cap (characters)	600	9.78	3.99

A EVALUATION DETAILS

Compared models and datasets. SASRec uses self-attention for sequential recommendation (Kang & McAuley, 2018); Mamba4Rec uses selective state-space modeling (Liu et al., 2024); TTT4Rec applies test-time training to sequential recommendation (Yang et al., 2024); HSTU is a sequential transducer for generative recommendation (Zhai et al., 2024); TIGER uses generative item retrieval (Rajput et al., 2023). The augmented backbones are Qwen3-1.7B (Yang et al., 2025) and the 1.7B OpenOneRec model (Zhou et al., 2025). All_Beauty is an Amazon product-review subset (Hou et al., 2026); KuaiRec and KuaiRand provide short-video interactions (Gao et al., 2022a;b). Table 1 reports the processed subsets used here.

Ranking metrics. For an evaluation context u, let Y_u be the relevant target set and $\hat{y}_{u,r}$ the prediction at rank r. Define $b_{u,r} = \mathbf{1}[\hat{y}_{u,r} \in Y_u]$. Pass@1 averages $b_{u,1}$; HR@5 averages $\mathbf{1}[\sum_{r=1}^{5} b_{u,r} > 0]$. NDCG@5 averages $\sum_{r=1}^{5} b_{u,r}/\log_2(r+1)$ divided by the ideal DCG for that context (Järvelin & Kekäläinen, 2002). We multiply these averages by 100. All three measures lie in $[0, 100]$, with Pass@1 and NDCG@5 bounded above by HR@5.

Distribution diagnostics. Let P and Q be normalized predicted and target SID frequency distributions on a common support, and $U = (P + Q)/2$. We use base-two logarithms:

$$\begin{aligned} \mathrm{JS}(P, Q) &= \tfrac{1}{2} D_{\mathrm{KL}}(P \| U) + \tfrac{1}{2} D_{\mathrm{KL}}(Q \| U), \\ \Delta H &= -\sum_{s} P(s) \log_2 P(s) + \sum_{s} Q(s) \log_2 Q(s). \end{aligned} \tag{8}$$

JS lies in $[0, 1]$ (Lin, 1991); an entropy gap near zero indicates similar dispersion (Cover & Thomas, 2006). The first-layer SID ratio is $|S_P|/|S_Q|$, where each set contains distinct first-layer identifiers. Long-tail Recall@5 measures the proportion of relevant long-tail targets retrieved among the first five predictions. Retrieval mutual information measures dependence between sequence groups and retrieved prototypes, in bits.

B Data Efficiency and Construction Budgets

The full-data endpoint in Table 5 is the OneRec pair in Table 2. The absolute gains decrease from 0.63 to 0.46 and 0.34 points as more training data becomes available.

The prototype-bank sweep improves monotonically over the three tested sizes, with diminishing increments. The retrieval sweep peaks at $M = 4$: $M = 8$ reduces NDCG@5 by 0.04/0.02 points. Longer trace caps produce smaller increments, with 300 characters only 0.01 points below the 600-character cap on both evaluated datasets.

C Illustrative Construction Examples

The two examples below illustrate how shared patterns and local evidence form a training record. The histories, item titles, targets, and outputs are illustrative, rather than predictions from the evaluation set. Each card shows the most relevant prototype among the retrieved set.

Example A: a domain pattern extends sparse evidence

History. Choosing flour types → Activating yeast.
Retrieved prototype. Baking basics: ingredient knowledge → fermentation preparation → basic dough preparation.
Selected evidence. Flour selection and yeast activation.
Constructed training record
1. **Vertical pattern:** Ingredient selection and fermentation lead to making dough.
2. **Observed evidence:** The history covers flour selection and yeast activation.
3. **Transition bridge:** A basic dough-making tutorial connects these preparation steps to a finished food.
4. **Target action:** A beginner's single-rise steamed-bun tutorial.

Illustrative direct predictions
Baseline: Flour brand comparisons.
With ProtoAdapter: A beginner's single-rise steamed-bun tutorial.
What the prototype contributes. It connects two observed preparation steps to a later procedural stage, giving the sparse history a broader behavioral context.

Example B: local evidence narrows a broad pattern

History. Gym equipment introduction → Barbell squat technique → Bodyweight squat corrections → Ten-minute living-room workout.
Retrieved prototype. Fitness learning: movement instruction → training practice.
Selected evidence. The two most recent events: bodyweight squat corrections and a living-room workout.
Constructed training record
1. **Vertical pattern:** Movement instruction leads to practice.
2. **Observed evidence:** Recent videos cover bodyweight squats and living-room exercise.
3. **Transition bridge:** These events narrow the broad fitness pattern to a continuous home workout without equipment.
4. **Target action:** A fifteen-minute equipment-free home workout.

Illustrative direct predictions
Baseline: Advanced barbell squats.
With ProtoAdapter: A fifteen-minute equipment-free home workout.
What the local evidence contributes. It identifies the relevant setting within the broad domain pattern, directing the transition toward home exercise.

The traces belong to offline training-data construction. The illustrated predictions use the original history as input after adaptation.

D Formal Analysis

D.1 From trace supervision to direct decoding

For context h, let $\widetilde{P}_t(\tau, y \mid h)$ be the teacher-augmented target distribution, $P_t(y \mid h)$ its answer marginal, $Q_\theta^J(\tau, y \mid h)$ the student's joint trace distribution, and $Q_\theta^D(y \mid h)$ its direct decoder. Write $Q_\theta^{J,Y}(y \mid h) = \sum_\tau Q_\theta^J(\tau, y \mid h)$ and define

$$\delta_{\mathrm{dj}} = \mathbb{E}_h \, \mathrm{TV}(Q_\theta^{J,Y}(\cdot \mid h), Q_\theta^D(\cdot \mid h)). \tag{9}$$

Proposition 1. Suppose $\sum_\tau \widetilde{P}_t(\tau, y \mid h) = P_t(y \mid h)$ almost surely and the expected joint KL divergence is finite. With natural-log KL,

$$\mathbb{E}_h \, \mathrm{TV}(P_t(\cdot \mid h), Q_\theta^D(\cdot \mid h)) \le \sqrt{\tfrac{1}{2}\mathbb{E}_h D_{\mathrm{KL}}(\widetilde{P}_t(\cdot, \cdot \mid h) \| Q_\theta^J(\cdot, \cdot \mid h))} + \delta_{\mathrm{dj}}. \tag{10}$$

Proof. Marginalization gives $D_{\mathrm{KL}}(P_t \| Q_\theta^{J,Y}) \le D_{\mathrm{KL}}(\widetilde{P}_t \| Q_\theta^J)$. Apply Pinsker's inequality to the answer marginals and the triangle inequality through $Q_\theta^{J,Y}$. Taking expectations and applying Jensen's inequality to the square root yields Eq. 10. □

Proposition 1 decomposes direct-decoder error into joint-distribution fit and disagreement between the two output formats. Direct-answer supervision fits the deployed distribution; trace supervision fits the augmented distribution. The marginal-preservation premise specifies when the answer marginal of that augmented distribution equals the target distribution.

D.2 Prototype-description approximation

Let $v(z)$ represent a behavior fragment under metric d_V. Expectations below use the same context-fragment law μ. Define

$$\epsilon_{\mathrm{proto}}(P) = \mathbb{E}_{z \sim \mu} \min_{p \in P} d_V(v(z), v(p)), \tag{11}$$

$$\eta_{\mathrm{ret}}(P) = \mathbb{E}_h \left[\min_{p \in \mathcal{R}_M(h)} d_V(v(z_h), v(p)) - \min_{p \in P} d_V(v(z_h), v(p)) \right], \tag{12}$$

where $\mathcal{R}_M(h) \subseteq P$ is the retrieved set. Let $\pi_M(h)$ be its nearest prototype, g an ideal description map, and $\widehat{\tau}(p)$ the teacher's description. For a description metric d_T, write

$$\xi_{\mathrm{teach}}(P) = \mathbb{E}_h d_T(g(v(\pi_M(h))), \widehat{\tau}(\pi_M(h))). \tag{13}$$

Proposition 2. If g is L_g-Lipschitz from (V, d_V) to (T, d_T), then

$$\mathbb{E}_h d_T(g(v(z_h)), \widehat{\tau}(\pi_M(h))) \le L_g\big(\epsilon_{\mathrm{proto}}(P) + \eta_{\mathrm{ret}}(P)\big) + \xi_{\mathrm{teach}}(P). \tag{14}$$

Proof. Insert $g(v(\pi_M(h)))$ and apply the triangle inequality. Lipschitz continuity bounds the context-to-prototype description distance by $L_g d_V(v(z_h), v(\pi_M(h)))$. Its expectation is $L_g(\epsilon_{\mathrm{proto}} + \eta_{\mathrm{ret}})$; the remaining term is ξ_{teach}. □

The bound separates coverage, retrieval selection, and teacher-description error for the selected prototype. For the optimal coverage error $\epsilon_K^\star = \inf_{|P| \le K} \epsilon_{\mathrm{proto}}(P)$, nested feasible sets give $\epsilon_{K+1}^\star \le \epsilon_K^\star$. The finite-budget experiments in Table 6 examine the corresponding design choices in recommendation performance.