A GENERATIVE RECOMMENDER ALSO NEEDS A DISCRETE LANGUAGE FOR MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative recommendation provides a discrete language for items, typically through Semantic IDs (SIDs), but leaves user memory implicit in the interaction sequence. Even when the complete history is available, the model must repeatedly infer from raw events the predictive state that matters for future behavior. We argue that *history and memory should not be conflated*: interaction tokens record what happened, whereas memory should explicitly represent what those interactions imply about the future. This motivates an explicit predictive memory that exposes the future-relevant meaning of the interaction history. We instantiate this principle with DEYI $\circ$ Φ. DEYI distills earlier interactions into a user-specific predictive state trained to preserve future-relevant information while the original interaction sequence remains available to the generator. Φ then maps these private continuous states into a shared discrete vocabulary according to their induced future-item predictions rather than latent-space proximity, giving recurring predictive states across users reusable symbolic identities. During generation, the resulting memory token is prepended to the full interaction sequence and also defines a context-dependent predictive route before the target item's native SID. Remarkably, adding only a single predictive memory token can outperform generation from the full history alone, showing that explicit memory provides information complementary to the event sequence itself. These results suggest a simple principle for generative recommendation: *history provides evidence, while memory provides predictive abstraction*. Code is available at https://anonymous.4open.science/r/DeYiPhi-1537.

1 INTRODUCTION

Generative recommendation casts recommendation as autoregressive generation over a user's interaction history (Rajput et al., 2023; Deng et al., 2025). With discrete item representations such as Semantic IDs (SIDs), observed interactions and recommendation targets can be expressed through the model's native token interface (Rajput et al., 2023; Wang et al., 2024). This gives the generator a discrete language for individual items and events. Yet the user state induced by those events remains implicit in the sequence: the history records *what happened*, but does not explicitly represent what those interactions collectively imply about the future. The generator must recover this predictive meaning from event-level tokens whenever it makes a recommendation. This raises a basic question: *does a generative recommender also need an explicit language for memory?*

We argue that *history and memory play complementary roles*. Interaction history provides fine-grained behavioral evidence, whereas memory should expose a higher-level interpretation useful for predicting future behavior. A history can preserve past events without explicitly expressing their predictive meaning. This distinction suggests that the central question is not which past events constitute memory, but rather: *what should an explicit memory represent?* Our answer is **prediction**. A useful memory should preserve the aspects of past behavior that determine what the user is likely to do next, while abstracting away distinctions irrelevant to future behavior.

Making such a memory explicit creates a second challenge. The predictive state distilled from one user's history is naturally continuous and user specific. Yet different users may reach states with essentially the same implications for future behavior. Treating each such state as a distinct memory misses this recurring structure, while conventional vector quantization is not sufficient: two states may

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.

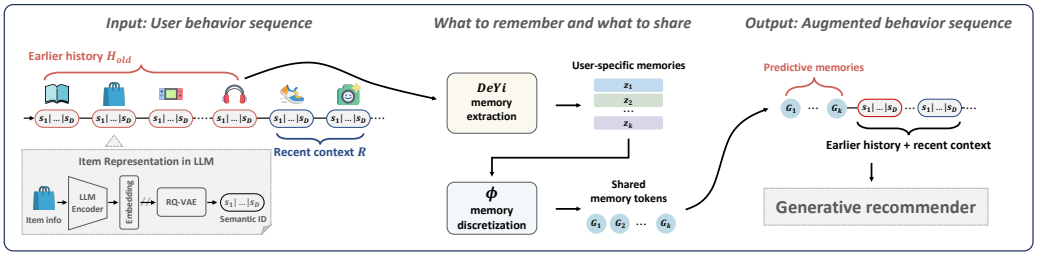


Figure 1: Predictive memory for generative recommendation. Item SIDs preserve event-level interaction evidence, while DEYI distills earlier interactions into an explicit predictive state and Φ assigns recurring predictive states a shared discrete identity. The resulting memory token accompanies the interaction sequence as a predictive abstraction for generation.

be geometrically close but predict different futures, or geometrically distant while inducing nearly the same future-item distribution. We therefore ask a second question: *when should two memories share the same identity?* Again, prediction provides the criterion. Two memories should be considered equivalent when substituting one for the other changes future-item predictions only negligibly. Under this view, prediction determines both the *meaning* of memory and the *identity* through which that meaning can be shared and reused across otherwise different users and interaction histories.

We refer to this principle as *predictive abstraction*, illustrated in Figure 1. Item SIDs preserve event-level evidence, while explicit memory exposes its predictive meaning through a shared discrete identity. We instantiate this view with DEYI $\circ\,\Phi$: DEYI distills earlier interactions into a user-specific predictive state, and Φ maps recurring states into a shared vocabulary according to their induced future-item distributions rather than latent-space proximity. The resulting token denotes a predictive concept reusable across users. During generation, it is prepended to the interaction sequence and participates directly in producing the target item's native SID. Thus, item SIDs provide a discrete language for *events*, while Φ provides one for *memory*.

Our experiments reveal a striking result: explicit predictive memory remains useful even when the complete interaction history is already available to the generator. A single memory token is already sufficient to improve full-history generation, showing that a compact predictive abstraction can complement event-level history. Controlled analyses further validate the roles of predictive state learning and prediction-aligned sharing.

Contributions. Our contributions are threefold. First, we identify a representation gap in generative recommendation and formulate explicit user memory as a *predictive abstraction* of interaction history. Second, we introduce DEYI $\circ\,\Phi$, where DEYI distills history into a user-specific predictive state and Φ assigns predictively equivalent states a shared discrete identity for generation. Third, we show that even a single predictive memory token is sufficient to improve full-history generation, and provide controlled evidence for the importance of predictive abstraction and prediction-aligned sharing.

2 PROBLEM FORMULATION

Setup. Let $\mathcal{I}$ denote the item catalog, where each item $i \in \mathcal{I}$ is represented by a native Semantic ID (SID) $s(i) = (s_1(i), \ldots, s_L(i))$. For a user, we write the observed interaction history as

$$H = (H_{\text{old}}, R),$$

where H_{old} contains earlier interactions and R denotes a short recent context. Let $Y = (y_1, \ldots, y_T)$ denote subsequent interactions used as future supervision. Our goal is to endow the event-level interaction sequence with an explicit memory that captures its future-relevant meaning.

What should memory represent? We seek a memory state

$$m = f(H_{\text{old}}), \qquad m \in \mathcal{M},$$

that abstracts the predictive consequences of earlier behavior rather than reconstructing the events themselves. Conceptually, a predictive memory should satisfy

$$p(Y \mid H_{\text{old}}) \;\approx\; p(Y \mid m), \tag{1}$$

so that the distinctions retained in m are precisely those that matter for future behavior. Equation (1) defines the desired *predictive abstraction* of history.

When should memories share an identity? A predictive state is naturally user specific, yet different users may induce states with nearly identical consequences for future behavior. Let

$$q : \mathcal{M} \to \mathcal{G}$$

map a memory state to a symbol from a finite shared vocabulary $\mathcal{G}$. For $m \in \mathcal{M}$, define

$$\pi_m(i) \triangleq p(i \mid m), \qquad i \in \mathcal{I},$$

as the future-item distribution induced by that state. We call two memories *predictively equivalent* when their induced future-item distributions are sufficiently similar. The desired abstraction therefore satisfies

$$\pi_m \approx \pi_{m'} \quad \Longrightarrow \quad q(m) = q(m'). \tag{2}$$

Conversely, memories with different predictive consequences should remain distinguishable even when they are close in the underlying representation space.

Equations (1) and (2) define the two requirements studied in this work: prediction determines the *meaning* that memory should encode, while predictive equivalence determines the *identity* through which that meaning can be shared. Section 3 presents our realization of these principles.

3 METHOD

Our method gives the interaction sequence an explicit predictive memory. As illustrated in Figure 2, DEYI first distills earlier history into continuous states whose semantics are defined by future behavior; Φ then maps recurring predictive states across users into a shared discrete vocabulary, whose tokens are exposed to the generative recommender together with the original interaction sequence. The framework therefore follows a simple progression from *history*, to *predictive state*, to *shared memory identity*, and finally to *generation*. Throughout the framework, memory is learned in a frozen item-semantic space: for item i in catalog $\mathcal{I}$, let $v(i)$ be its semantic representation and $e(i) = v(i)/\|v(i)\|_2$ its normalized embedding. Memory learning and predictive quantization operate over $e(i)$, while the generative backbone retains its native Semantic ID $s(i)$ for item generation.

3.1 DEYI: LEARNING PREDICTIVE MEMORY

We first address *what memory should represent*. For user r, DEYI maps the earlier history H_r^{old} into K normalized predictive states

$$Z_r = [z_{r1}, \ldots, z_{rK}] = \text{DEYI}(H_r^{\text{old}}), \qquad \|z_{rk}\|_2 = 1, \tag{3}$$

where z_{rk} is the k-th memory state. Historical events are contextualized by self-attention, after which K learned memory queries retrieve user-specific content through cross-attention. Each state is trained through the future-item distribution it induces:

$$p(i \mid z_{rk}) = \frac{\exp\big(z_{rk}^\top e(i)/\tau\big)}{\sum_{j \in \mathcal{I}} \exp\big(z_{rk}^\top e(j)/\tau\big)}, \tag{4}$$

where τ controls the sharpness of the distribution. Because both z_{rk} and $e(i)$ are normalized, their inner product measures alignment between the memory state and item i in the shared semantic space. The resulting distribution therefore gives the state a functional meaning: it specifies which future items the state considers likely.

For $K > 1$, the recent context R_r is used to compute relevance weights α_{rk} over the predictive states:

$$p(i \mid Z_r, R_r) = \sum_{k=1}^{K} \alpha_{rk}\, p(i \mid z_{rk}), \qquad \sum_{k=1}^{K} \alpha_{rk} = 1. \tag{5}$$

Here, α_{rk} measures the context-dependent relevance of state k, so different memories contribute according to the user's recent behavior. Given a future supervision window Y_r containing subsequent

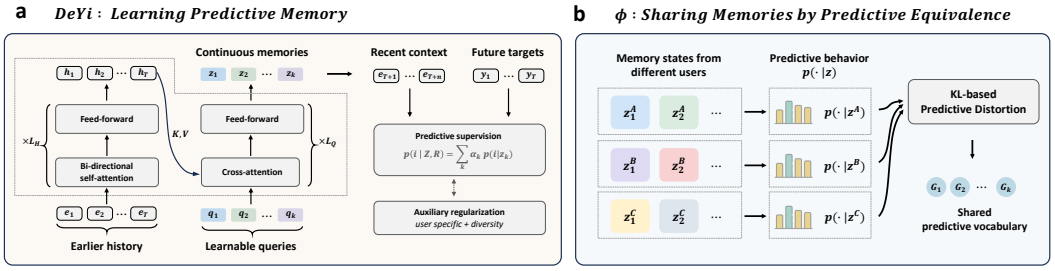


Figure 2: Overview of DEYI ∘ Φ. (a) DEYI distills earlier interactions into user-specific predictive memory while the event-level interaction sequence remains explicit. (b) Φ assigns memories with similar future-item predictions a shared discrete identity. The resulting memory token accompanies the native item sequence and directly participates in autoregressive generation.

interactions of user r, we optimize

$$\mathcal{L}_{\text{pred}} = -\mathbb{E}_r \left[\frac{1}{|Y_r|} \sum_{y \in Y_r} \log p(y \mid Z_r, R_r) \right], \tag{6}$$

where the expectation is taken over training users. The objective encourages memory to retain the structure in earlier behavior that remains informative about subsequent interactions.

3.2 PHI: GIVING MEMORY A SHARED DISCRETE IDENTITY

DEYI produces predictive states, but they remain continuous and user specific. We next address *when memories should share an identity*. As shown in Figure 2(b), Φ compares states through their predictive behavior rather than latent-space geometry. For a state z, let $p_z = p(\cdot \mid z)$ denote its predictive distribution from Eq. (4); a representative c analogously induces $p_c = p(\cdot \mid c)$. We define their predictive distortion as

$$d_{\text{pred}}(z, c) = D_{\text{KL}}\left(p_z \,\|\, p_c\right). \tag{7}$$

A small distortion means that replacing z with c changes future-item predictions little, making c a suitable representative of that predictive state. To evaluate this quantity without repeatedly materializing full catalog distributions, define

$$\mu_z = \mathbb{E}_{i \sim p_z}[e(i)], \qquad A(c) = \log \sum_{i \in \mathcal{I}} \exp\left(c^\top e(i)/\tau\right), \tag{8}$$

where μ_z is the expected item-semantic embedding under the predictions of z, and $A(c)$ is the log-partition term of representative c. The KL divergence then becomes

$$D_{\text{KL}}(p_z \| p_c) = C_z + A(c) - \frac{1}{\tau} c^\top \mu_z, \tag{9}$$

where C_z depends only on z and is constant when comparing representatives. For a fixed state, Eq. (9) therefore favors representatives aligned with the future-item semantics summarized by μ_z, while $A(c)$ accounts for the representative's normalization over the catalog. This makes predictive equivalence computable from compact statistics rather than an explicit state–catalog probability matrix; the derivation is given in Appendix B.

Let

$$\mathcal{V}_\Phi = \{g_1, \ldots, g_G\}, \qquad \mathcal{R}_\Phi = \{c_1, \ldots, c_G\}$$

denote the shared memory vocabulary and its G predictive representatives in the same semantic space, where token g_ℓ is associated with c_ℓ. Given training states $\{z_s\}$, Φ learns the vocabulary by minimizing

$$\mathcal{L}_\Phi = \sum_s w_s D_{\text{KL}}\left(p_{z_s} \,\|\, p_{c_{T(z_s)}}\right), \qquad T(z_s) = \arg\min_{\ell \in \{1,\ldots,G\}} D_{\text{KL}}\left(p_{z_s} \,\|\, p_{c_\ell}\right), \tag{10}$$

where the optimization is over $\{c_\ell\}_{\ell=1}^{G}$, $T(z_s)$ assigns each state to its minimum-distortion representative, and w_s balances training users. Thus, the representatives summarize recurring predictive functions, while T determines which shared identity each state receives. Accordingly, state z_{rk} is represented by the discrete token $g_{rk} = g_{T(z_{rk})}$. States from different users share a symbol when they induce similar future-item distributions, turning recurring predictive structure into a shared identity. For large state collections and catalogs, we use a scalable realization described in Appendix C.

3.3 Memory-Augmented Generative Recommendation

After Φ, user r is represented by discrete predictive memories

$$G_r = [g_{r1}, \ldots, g_{rK}],$$

which are prepended to the complete native interaction sequence:

$$X_r = [G_r,\ \mathrm{SID}(H_r^{\mathrm{old}}),\ \mathrm{SID}(R_r)]. \tag{11}$$

Thus, the original event-level evidence remains available while G_r exposes its predictive abstraction in the generator's discrete interface. For the main setting $K = 1$, this simply adds one shared predictive memory token before the interaction sequence.

We also expose the relevant memory identity in the autoregressive target. For target item y, the responsibility

$$\gamma_{rk}(y) = \frac{\alpha_{rk}\, p(y \mid z_{rk})}{\sum_{j=1}^{K} \alpha_{rj}\, p(y \mid z_{rj})} \tag{12}$$

measures how strongly state k accounts for predicting y. When $K > 1$, responsibilities of states sharing the same token are aggregated, and the most relevant shared identity is

$$g^*(r, y) = \arg\max_{g \in \mathcal{V}_\Phi} \sum_{k: g_{rk} = g} \gamma_{rk}(y). \tag{13}$$

For $K = 1$, this reduces directly to $g^*(r, y) = g_{r1}$. The decoder target is

$$[g^*(r, y),\ \mathrm{SID}(y)]. \tag{14}$$

The target item is used only to construct the memory-token supervision during training. At inference, the model autoregressively generates the memory identity before the item's native SID. The recommender retains its standard autoregressive next-token objective.

4 Experiments

Our experiments follow the conceptual progression of the paper. We first ask whether explicit predictive memory is useful when the generator already observes the complete interaction history. We then examine what makes such a memory useful, test whether the discrete symbols learned by Φ correspond to predictive equivalence, and finally study how much explicit memory is needed.

4.1 Experimental Setup

Datasets. We evaluate on the *Product* and *Short Video* tasks from the released OpenOneRec RecIF benchmark. The two domains provide complementary regimes: Product has a moderate catalog and shorter histories, whereas Short Video combines substantially longer histories with a catalog an order of magnitude larger. For each task, we use its single-domain interaction channel, retain users with at least 32 historical interactions and one future target, discard items without valid representations, and truncate histories to the task-specific limit shown in Table 1. The benchmark test files remain held out throughout model development.

Table 1: Dataset statistics after preprocessing.

Dataset	Train	Test	Hist.	Catalog
Product	76.0K	18.7K	100	1.13M
Short Video	151.5K	37.5K	512	12.88M

Table 2: **A shared predictive memory token consistently improves full-history generation.** Results on Product and Short Video with TIGER. All variants receive the same complete interaction sequence; DEYI additionally exposes a continuous predictive state, while DEYI $\circ\ \Phi$ exposes its shared discrete identity through Φ. Memory variants use $K = 1$. Higher is better; the best and second-best results within each dataset are **bolded** and underlined, respectively.

	Pass@k				Recall@k				NDCG@k			
Method	5	10	20	32	5	10	20	32	5	10	20	32
Product												
FULL	0.0679	0.1048	0.1519	0.1879	0.0122	0.0194	0.0295	0.0374	0.0177	0.0180	0.0227	0.0257
DEYI	0.0667	0.1047	0.1532	0.1881	0.0119	0.0195	0.0298	0.0379	0.0169	0.0176	0.0223	0.0253
DEYI $\circ\ \Phi$	**0.0742**	**0.1151**	**0.1666**	**0.2051**	**0.0134**	**0.0213**	**0.0323**	**0.0415**	**0.0189**	**0.0195**	**0.0245**	**0.0281**
Short Video												
FULL	0.0335	0.0519	0.0715	0.0827	0.0038	0.0060	0.0085	0.0099	0.0077	0.0067	0.0081	0.0087
DEYI	0.0281	0.0436	0.0620	0.0730	0.0031	0.0049	0.0072	0.0086	0.0066	0.0057	0.0069	0.0076
DEYI $\circ\ \Phi$	**0.0374**	**0.0553**	**0.0766**	**0.0888**	**0.0042**	**0.0064**	**0.0091**	**0.0107**	**0.0089**	**0.0075**	**0.0090**	**0.0097**

Backbone and evaluation. We use TIGER (Rajput et al., 2023), a representative SID-based generative recommender in which a Transformer encoder–decoder autoregressively generates the target item's Semantic ID. This makes TIGER a natural testbed for our central question: whether the discrete language for items should be accompanied by an explicit discrete language for user memory. All variants retain the same native item representation and complete interaction sequence. We report Pass@k, Recall@k, and NDCG@k for $k \in \{5, 10, 20, 32\}$, using beam search to generate 32 candidates. Unless stated otherwise, the explicit memory contains a single predictive state ($K = 1$).

4.2 SHARED PREDICTIVE MEMORY IMPROVES GENERATION WITH FULL HISTORY

The first question is simple: *does explicit memory remain useful when the complete interaction history is already available to the generator?* Table 2 compares the native full-history generator with continuous predictive memory from DEYI and the shared discrete memory produced by DEYI $\circ\ \Phi$. All three variants observe the same interaction sequence; they differ only in whether the predictive meaning of that history is made explicit and shared across users.

A single shared predictive memory token consistently improves generation across both domains, evaluation cutoffs, and metric families. The contrast with continuous DEYI is revealing: a user-specific predictive state alone provides limited benefit, whereas Φ turns recurring states with similar future consequences across users into reusable memory symbols. The generator therefore receives the original behavioral evidence together with a shared predictive concept that can be reused across users whenever different histories imply similar future behavior.

This result establishes the central empirical observation of the paper: event-level history and explicit memory play complementary roles in generation. History records what happened, whereas the memory token exposes a reusable predictive abstraction of what those events imply about the future. Sections 4.3 and 4.4 next examine why this shared representation is useful and whether its learned symbols indeed correspond to recurring future-behavior patterns.

4.3 WHAT MAKES EXPLICIT MEMORY USEFUL?

The main result establishes the value of a shared predictive memory, but leaves open what property of the representation makes it useful. Table 3 follows the two questions that motivated the method: *what should memory represent*, and *when should memories share an identity?* The analysis keeps the full-history TIGER interface unchanged and varies the form of the auxiliary memory.

A memory must carry meaningful structure. We begin with the weakest possible interventions. A learned DUMMY token supplies an additional trainable position without encoding the user's history, while POOLING provides a generic summary of earlier item representations. Neither changes the full-

Table 3: **What makes explicit memory useful?** Controlled analysis on TIGER + Product. Every variant retains the complete interaction sequence, TIGER backbone, and native item representation. The three blocks examine whether auxiliary memory alone is sufficient, what information memory should preserve, and how predictive states should share a discrete identity. Higher is better.

Method	Memory organized by	Representation	Pass@32	Recall@32	NDCG@32
FULL	–	None	0.1879	0.0374	0.0257
Does an auxiliary representation suffice?					
+ DUMMY	None	Continuous	0.1875	0.0374	0.0257
+ POOLING	Generic summary	Continuous	0.1860	0.0372	0.0251
What should memory represent?					
+ CAUSE-Core	Category	Continuous	0.1875	0.0375	0.0248
+ HiCoGen-Core	Semantics	Continuous	0.1877	0.0376	0.0252
+ Chronicle-Core	Time	Continuous	0.1853	0.0370	0.0249
+ DEYI	Prediction	Continuous	0.1881	0.0379	0.0253
When should memories share an identity?					
+ DEYI ∘ Uniform VQ	Latent quantization	Discrete	0.1949	0.0394	0.0266
+ DEYI ∘ Geometric VQ	Cosine geometry	Discrete	0.1942	0.0391	0.0264
+ **DEYI** ∘ Φ	Predictive equivalence	Discrete	**0.2051**	**0.0415**	**0.0281**

history result meaningfully. Simply adding another representation, or compressing history without specifying what information the summary should preserve, is therefore insufficient. The useful signal must come from how history is abstracted.

Prediction provides the strongest continuous abstraction. We next organize continuous memory according to four different views of the same history. CAUSE-Core (Liu et al., 2026b) uses category structure, HiCoGen-Core (Li et al., 2025) uses semantic similarity, Chronicle-Core (Huang et al., 2026) uses temporal structure, and DEYI defines memory through future prediction. Among these continuous alternatives, the prediction-trained state gives the strongest result across the reported metrics. The comparison connects directly to the formulation in Section 2: a useful memory is better characterized by the future behavior it preserves than by a particular partition of the observed past.

A shared identity is most useful when it preserves predictive meaning. The final block starts from DEYI predictive states and changes how those states acquire a discrete identity. Uniform VQ and Geometric VQ already improve upon the private continuous representation, suggesting that reusable discrete identities are valuable to the generator. More importantly, the sharing criterion determines what structure is reused across users: Φ assigns the same symbol to recurring predictive states with similar future-item distributions, allowing a common predictive concept to be shared across otherwise different histories. This prediction-aligned sharing gives the strongest result, indicating that the benefit of discretization is greatest when the shared identity captures recurring future consequences rather than generic latent-space proximity.

Taken together, the three blocks trace the same progression as the method: history must first be distilled into a predictive state, and recurring predictive states must then be given a shared symbolic identity. The large gain appears only when these two distinct but complementary roles are combined.

4.4 MEMORY TOKENS FORM PREDICTIVE REGIONS

The previous experiment shows that prediction-aligned sharing is useful for recommendation. Here we test the stronger representation-level claim behind Φ: if a memory token denotes a predictive identity, then users assigned the same token should exhibit similar future behavior.

We evaluate this property on held-out users that do not participate in memory codebook construction. For a memory token g, let $S_{\text{intra}}(g)$ denote the average future similarity among users assigned to g, and let $S_{\text{inter}}(g)$ denote the corresponding similarity between those users and users assigned to other tokens. A token that captures a coherent predictive region should satisfy $S_{\text{intra}}(g) > S_{\text{inter}}(g)$. Figure 3 examines this relation from two complementary views. At the token level, each point

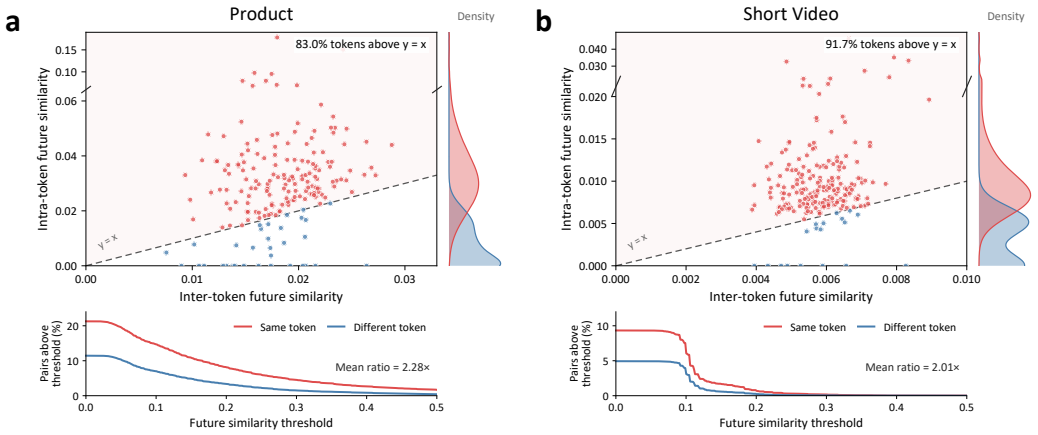


Figure 3: **Memory tokens form predictive regions across domains.** Each column corresponds to one dataset. *Top:* each point represents a learned memory token, with average inter-token future similarity on the x-axis and average intra-token future similarity on the y-axis. Points above the dashed $y = x$ line indicate stronger future agreement within a token than across tokens; the right marginal shows the corresponding density. *Bottom:* fraction of user pairs whose future similarity exceeds a varying threshold, separated into same-token and different-token pairs. In both domains, same-token users exhibit systematically stronger agreement in future behavior.

compares intra-token and inter-token future similarity. Across both Product and Short Video, the large majority of tokens lie above the equality line, and the marginal distributions are correspondingly shifted toward stronger within-token agreement. At the pair level, we sweep the future-similarity threshold and measure the fraction of same-token and different-token pairs that remain above it. The same-token curve dominates throughout the informative range, with an average prevalence roughly twice that of different-token pairs. The two views tell the same story at different resolutions. Individual tokens define internally coherent future-behavior regions, and this structure persists when the vocabulary is examined over user pairs rather than token averages. The discrete identities learned by Φ therefore exhibit the predictive equivalence that motivated their construction: a shared memory symbol corresponds to recurring consequences for future behavior.

4.5 One Predictive Memory Token Is Sufficient

The preceding analyses establish what memory should represent and what its shared discrete identity means. We finally ask how much explicit predictive memory the generator needs. On Product, we vary the number of predictive states $K \in \{1, 2, 4, 8\}$ while keeping the interaction history, item space, predictive objective, Φ vocabulary, TIGER backbone, and evaluation protocol unchanged. The corresponding Short Video analysis is reported in Appendix A.

Figure 4 reveals a clear pattern. Across Pass@32, Recall@32, and NDCG@32, shared predictive memory remains beneficial throughout the tested range, but the largest improvement is already obtained with $K = 1$. Increasing the number of states does not yield a monotonic gain, while continuous DEYI memory remains near or below the full-history reference.

This behavior is consistent with the role of memory in our formulation. Because the complete interaction sequence already preserves fine-grained behavioral evidence, explicit memory is most useful as a compact predictive abstraction rather than an additional store of detail. With $K = 1$, future-relevant information is concentrated into a single predictive state whose shared identity can recur across users. Larger K increases capacity, but can fragment this recurring predictive structure across multiple states and introduce additional ambiguity in selecting which state should guide prediction. The capacity study therefore reinforces the central design principle: the benefit of explicit memory comes from exposing a concise, reusable predictive abstraction. Remarkably, a single shared token suffices to instantiate this predictive abstraction within the generator's discrete interface.

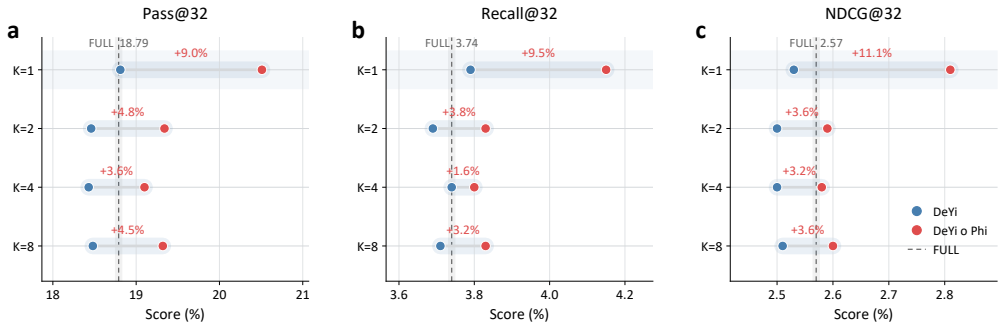


Figure 4: **Memory-capacity analysis on Product.** The panels report Pass@32, Recall@32, and NDCG@32 as K varies over $\{1, 2, 4, 8\}$. Blue denotes continuous DEYI memory, red denotes the shared discrete memory of DEYI $\circ$ Φ, and the dashed line marks the FULL reference. The strongest improvement is obtained at $K = 1$, while increasing memory capacity provides no monotonic gain.

5 Related Work

Generative Recommendation and Discrete Item Languages. Generative recommendation casts recommendation as token generation, making item representation part of the model interface. LLM-based approaches use textual identifiers, attributes, descriptions, or multimodal content (Geng et al., 2022; Bao et al., 2023; Lin et al., 2024; Tan et al., 2024; Zheng et al., 2024; Liu et al., 2024; Kim et al., 2024; 2026). RQ-VAE-style Semantic IDs enable autoregressive generation in TIGER (Lee et al., 2022; Rajput et al., 2023), with later tokenizers incorporating collaborative, contextual, order-agnostic, parallel, or multi-aspect signals (Wang et al., 2024; Hou et al., 2025b; Lin et al., 2025; Hou et al., 2025a; Liu et al., 2026c). Recent work further advances scalable generative recommendation (Zhai et al., 2024; Deng et al., 2025; Huang et al., 2025; Zhou et al., 2025). These works refine the discrete language for *items*, while user memory remains implicit. We introduce a corresponding memory language: DEYI extracts a predictive state, and Φ maps recurring states with similar future consequences across users to a shared identity.

User Memory and Predictive Representation. Recommendation models have long constructed compact representations beyond interaction events. Candidate-aware methods retrieve target-relevant behaviors (Qin et al., 2020; Pi et al., 2020; Chen et al., 2021; Chang et al., 2023), while candidate-independent approaches summarize histories through memory, clustering, interests, or scalable sequence modeling (Pi et al., 2019; Ren et al., 2019; Yan et al., 2024; Wang et al., 2025; Chai et al., 2025). Generative recommenders organize history through hierarchical interests, categorical aggregation, temporal structure, or context-aware representations (Li et al., 2025; Liu et al., 2026b; Zhou et al., 2026; Liu et al., 2026a; Huang et al., 2026). These approaches differ in how the past is organized. We instead define memory by its *predictive consequences*: DEYI learns what history implies about future behavior, while Φ uses predictive equivalence to determine when states across users should share an identity. The resulting vocabulary turns recurring cross-user predictive structure into reusable symbols that can be shared across otherwise different interaction histories.

6 Conclusion

We identify a representation gap in generative recommendation: items have a discrete language, while user memory remains implicit in the interaction sequence. DEYI $\circ$ Φ addresses this gap by distilling history into future-predictive states and assigning recurring predictive states across users a shared discrete identity. Experiments show that this explicit memory complements the complete event sequence, with a single shared token already sufficient to improve generation. Our analyses further show that prediction provides both the meaning of memory and the criterion by which that meaning should be shared. Together, these results suggest that a generative recommender benefits from a discrete language not only for what happened, but also for what events imply about the future.

AI USE STATEMENT

Generative AI tools were used for code assistance, literature organization, figure and table preparation, and language editing. The authors reviewed all AI-assisted content, verified cited sources and experimental results, and take full responsibility for the final paper.

ETHICS STATEMENT

This work studies representation learning for generative recommendation and introduces no new data collection or user-facing deployment. As with recommender systems more broadly, responsible use should consider potential issues related to privacy, bias, and fairness.

REPRODUCIBILITY STATEMENT

Implementation details are available at: `https://anonymous.4open.science/r/DeYiPhi-1537`. The experimental setup is described in Section 4; Appendices B and C provide the derivation of predictive-function quantization and the scalable realization of Φ, respectively.

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.

Zheng Chai, Qin Ren, Xijun Xiao, Huizhi Yang, Bo Han, Sijun Zhang, Di Chen, Hui Lu, Wenlin Zhao, Lele Yu, Xionghang Xie, Shiru Ren, Xiang Sun, Yaocheng Tan, Peng Xu, Yuchao Zheng, and Di Wu. Longer: Scaling up long sequence modeling in industrial recommenders. In *Proceedings of the 19th ACM Conference on Recommender Systems*, pp. 247–256, 2025. doi: 10.1145/3705328.3748065.

Jianxin Chang, Chenbin Zhang, Zhiyi Fu, Xiaoxue Zang, Lin Guan, Jing Lu, Yiqun Hui, Dewei Leng, Yanan Niu, Yang Song, and Kun Gai. Twin: Two-stage interest network for lifelong user behavior modeling in ctr prediction at kuaishou. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3785–3794, 2023. doi: 10.1145/3580305.3599922.

Qiwei Chen, Changhua Pei, Shanshan Lv, Chao Li, Junfeng Ge, and Wenwu Ou. End-to-end user behavior retrieval in click-through rate prediction model. *arXiv preprint arXiv:2108.04468*, 2021.

Jiaxin Deng, Shiyao Wang, Kuo Cai, Lejian Ren, Qigen Hu, Weifeng Ding, Qiang Luo, and Guorui Zhou. Onerec: Unifying retrieve and rank with generative recommender and iterative preference alignment. *arXiv preprint arXiv:2502.18965*, 2025.

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.

Yupeng Hou, Jiacheng Li, Ashley Shin, Jinsung Jeon, Abhishek Santhanam, Wei Shao, Kaveh Hassani, Ning Yao, and Julian McAuley. Generating long semantic ids in parallel for recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 956–966, 2025a. doi: 10.1145/3711896.3736979.

Yupeng Hou, Jianmo Ni, Zhankui He, Noveen Sachdeva, Wang-Cheng Kang, Ed H. Chi, Julian McAuley, and Derek Zhiyuan Cheng. Actionpiece: Contextually tokenizing action sequences for generative recommendation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 24004–24024, 2025b.

Chengkai Huang, Yubin Sheng, Liang Guo, Haoxi Liu, Junwei Pan, Shangyu Zhang, Zhixiang Feng, Chao Zhou, Chengguo Yin, Lina Yao, Haijie Gu, and Jie Jiang. Chroniclerec: Pre-training temporally anchored tokens for lifelong user modeling. *arXiv preprint arXiv:2609.12375*, 2026.

Yanhua Huang, Yuqi Chen, Xiong Cao, Rui Yang, Mingliang Qi, Yinghao Zhu, Qingchang Han, Yaowei Liu, Zhaoyu Liu, Xuefeng Yao, Yuting Jia, Leilei Ma, Yinqi Zhang, Taoyu Zhu, Liujie Zhang, Lei Chen, Weihang Chen, Min Zhu, Ruiwen Xu, and Lei Zhang. Towards large-scale generative ranking. *arXiv preprint arXiv:2505.04180*, 2025.

Kibum Kim, Sein Kim, HongSeok Kang, Jiwan Kim, Heewoong Noh, Yeonjun In, Kanghoon Yoon, Jinoh Oh, Julian McAuley, and Chanyoung Park. Token-efficient item representation via images for llm recommender systems. In *International Conference on Learning Representations*, 2026.

Sein Kim, Hongseok Kang, Seungyoon Choi, Donghyun Kim, Minchul Yang, and Chanyoung Park. Large language models meet collaborative filtering: An efficient all-round llm-based recommender system. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1395–1406, 2024. doi: 10.1145/3637528.3671931.

Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11523–11532, 2022. doi: 10.1109/CVPR52688.2022.01123.

Wuchao Li, Shiyao Wang, Kuo Cai, Jiaxin Deng, Xingmei Wang, Qigen Hu, Defu Lian, and Guorui Zhou. Taming ultra-long behavior sequence in session-wise generative recommendation. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pp. 5839–5846, 2025. doi: 10.1145/3746252.3761564.

Xinyu Lin, Wenjie Wang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. Bridging items and language: A transition paradigm for large language model-based recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1816–1826, 2024. doi: 10.1145/3637528.3671884.

Xinyu Lin, Haihan Shi, Wenjie Wang, Fuli Feng, Qifan Wang, See-Kiong Ng, and Tat-Seng Chua. Order-agnostic identifier for large language model-based generative recommendation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1923–1933, 2025. doi: 10.1145/3726302.3730053.

Chang Liu, Changfa Wu, Hui Qian, Binbin Cao, Jian Wu, Yuliang Yan, Han Zhu, and Bo Zheng. Sparc: Sequence-aware progressive attribute routing and compression framework for generative recommendation. *arXiv preprint arXiv:2607.25339*, 2026a.

Qijiong Liu, Nuo Chen, Tetsuya Sakai, and Xiao-Ming Wu. Once: Boosting content-based recommendation with both open- and closed-source large language models. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pp. 452–461, 2024. doi: 10.1145/3616855.3635845.

Qijiong Liu, Lu Fan, Zhongzhou Liu, Xiaoyu Dong, Yuankai Luo, Guoyuan An, Nuo Chen, Wei Guo, Yong Liu, and Xiao-Ming Wu. Accelerating generative recommendation via simple categorical user sequence compression. In *Proceedings of the 19th ACM International Conference on Web Search and Data Mining*, pp. 1195–1199, 2026b. doi: 10.1145/3773966.3779399.

Qijiong Liu, Jieming Zhu, Zhaocheng Du, Lu Fan, Zhou Zhao, and Xiao-Ming Wu. Learning multi-aspect item palette: A semantic tokenization framework for generative recommendation. In *Advances in Knowledge Discovery and Data Mining*, pp. 395–406, 2026c. doi: 10.1007/978-981-92-1465-5_31.

Qi Pi, Weijie Bian, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. Practice on long sequential user behavior modeling for click-through rate prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2671–2679, 2019. doi: 10.1145/3292500.3330666.

Qi Pi, Guorui Zhou, Yujing Zhang, Zhe Wang, Lejian Ren, Ying Fan, Xiaoqiang Zhu, and Kun Gai. Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, pp. 2685–2692, 2020. doi: 10.1145/3340531.3412744.

Jiarui Qin, Weinan Zhang, Xin Wu, Jiarui Jin, Yuchen Fang, and Yong Yu. User behavior retrieval for click-through rate prediction. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2347–2356, 2020. doi: 10.1145/3397271.3401440.

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

Kan Ren, Jiarui Qin, Yuchen Fang, Weinan Zhang, Lei Zheng, Weijie Bian, Guorui Zhou, Jian Xu, Yong Yu, Xiaoqiang Zhu, and Kun Gai. Lifelong sequential modeling with personalized memorization for user response prediction. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 565–574, 2019. doi: 10.1145/3331184.3331230.

Juntao Tan, Shuyuan Xu, Wenyue Hua, Yingqiang Ge, Zelong Li, and Yongfeng Zhang. Idgenrec: Llm-recsys alignment with textual id learning. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 355–364, 2024. doi: 10.1145/3626772.3657821.

Wenjie Wang, Honghui Bao, Xinyu Lin, Jizhi Zhang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. Learnable item tokenization for generative recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 2400–2409, 2024. doi: 10.1145/3627673.3679569.

Xingmei Wang, Shiyao Wang, Wuchao Li, Jiaxin Deng, Song Lu, Defu Lian, and Guorui Zhou. Transformers are good clusterers for lifelong user behavior sequence modeling. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pp. 3123–3132, 2025. doi: 10.1145/3746252.3761294.

Jing Yan, Liu Jiang, Jianfei Cui, Zhichen Zhao, Xingyan Bin, Feng Zhang, and Zuotao Liu. Trinity: Syncretizing multi-/long-tail/long-term interests all in one. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 6095–6104, 2024. doi: 10.1145/3637528.3671651.

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, 2024.

Zhi Zheng, Wenshuo Chao, Zhaopeng Qiu, Hengshu Zhu, and Hui Xiong. Harnessing large language models for text-rich sequential recommendation. In *Proceedings of the ACM Web Conference 2024*, pp. 3207–3216, 2024. doi: 10.1145/3589334.3645358.

Sashuai Zhou, Weinan Gan, Qijiong Liu, Ke Lei, Jieming Zhu, Hai Huang, Yan Xia, Ruiming Tang, Zhenhua Dong, and Zhou Zhao. Recbase: Generative foundation model pretraining for zero-shot recommendation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 15587–15599, 2025. doi: 10.18653/v1/2025.emnlp-main.786.

Yu Zhou, Chengcheng Guo, Kuo Cai, Ji Liu, Qiang Luo, Ruiming Tang, Han Li, Kun Gai, and Guorui Zhou. Gems: Breaking the long-sequence barrier in generative recommendation with a multi-stream decoder. *arXiv preprint arXiv:2602.13631*, 2026.

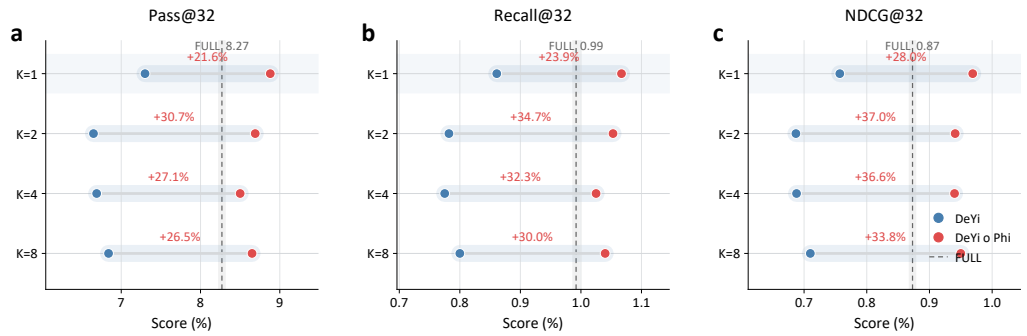


Figure 5: **Memory-capacity analysis on Short Video.** The panels report Pass@32, Recall@32, and NDCG@32 as K varies over $\{1, 2, 4, 8\}$. Blue denotes continuous DEYI memory, red denotes the shared discrete memory of DEYI $\circ\, \Phi$, and the dashed line marks the FULL reference.

A ADDITIONAL EXPERIMENTS ON SHORT VIDEO

We complement the Product analyses in Section 4.3 and Section 4.5 with the corresponding experiments on Short Video. This domain provides a substantially different regime, with longer interaction histories and a much larger item catalog. We retain the same TIGER backbone, complete interaction sequence, memory construction, and evaluation protocol as in the main experiments.

How much predictive memory is needed? We also vary the number of predictive states over $K \in \{1, 2, 4, 8\}$ while keeping the full interaction history available to the generator. Figure 5 shows that the continuous DEYI representation remains below the full-history reference throughout the tested range, whereas the shared discrete memory produced by DEYI $\circ\, \Phi$ consistently exceeds it. Importantly, a single predictive token is already sufficient to improve over full-history generation on all three metrics. Unlike Product, however, Short Video benefits from a modest increase in memory capacity: the largest gains occur at $K = 2$, reaching 30.7% on Pass@32, 34.7% on Recall@32, and 37.0% on NDCG@32 relative to FULL. Increasing capacity further to $K = 4$ or $K = 8$ does not produce additional improvement. Taken together, the Short Video experiments reinforce the main conclusion while adding an important nuance to the capacity analysis. Explicit predictive memory remains useful even when the complete interaction history is available, and its benefit arises from a shared predictive representation rather than from memory size alone. A single token is sufficient to expose this predictive abstraction, while the longer and more heterogeneous Short Video histories can benefit from a small amount of additional capacity.

B PREDICTIVE-FUNCTION QUANTIZATION

This section gives the mathematical details behind Eq. (9). For a normalized state z, write

$$p_z(i) = \exp\left(\frac{z^\top e(i)}{\tau} - A(z)\right), \qquad A(z) = \log \sum_{j \in \mathcal{C}} \exp\left(\frac{z^\top e(j)}{\tau}\right). \tag{15}$$

For two states z and c,

$$D_{\mathrm{KL}}(p_z \| p_c) = \mathbb{E}_{i \sim p_z}\left[\log p_z(i) - \log p_c(i)\right] \tag{16}$$

$$= \frac{1}{\tau}(z - c)^\top \mathbb{E}_{i \sim p_z}[e(i)] - A(z) + A(c) \tag{17}$$

$$= C_z + A(c) - \frac{1}{\tau} c^\top \mu_z, \tag{18}$$

where

$$\mu_z = \mathbb{E}_{i \sim p_z}[e(i)], \qquad C_z = \frac{1}{\tau} z^\top \mu_z - A(z). \tag{19}$$

Since C_z does not depend on the representative c, assignment under predictive KL reduces to

$$T(z) = \arg\min_g \left[A(c_g) - \frac{1}{\tau} c_g^\top \mu_z \right]. \tag{20}$$

The same identity yields a simple interpretation of prototype optimization. For the states assigned to token g, let

$$\bar{\mu}_g = \frac{\sum_{s:T(z_s)=g} w_s \mu_{z_s}}{\sum_{s:T(z_s)=g} w_s}. \tag{21}$$

Updating its representative amounts to

$$c_g = \arg\min_c \left[A(c) - \frac{1}{\tau} c^\top \bar{\mu}_g \right], \tag{22}$$

whose stationary point satisfies

$$\mathbb{E}_{i \sim p_{c_g}}[e(i)] = \bar{\mu}_g. \tag{23}$$

Thus each codeword is the canonical predictive representative of the states assigned to it.

C Scalable Realization of Phi

The exact formulation in Section 3.2 requires predictive statistics for every state over the full catalog, leading to $O(|\mathcal{S}||\mathcal{C}|d)$ state–catalog computation. Our large-scale implementation changes the computation, not the underlying notion of predictive equivalence: we estimate predictive geometry accurately on a bounded, user-balanced state reservoir and distill the resulting partition to the remaining states.

Predictive statistics on a reservoir. Let

$$\mathcal{S}_{\text{code}} \subset \mathcal{S}_{\text{fit}}, \qquad |\mathcal{S}_{\text{code}}| \ll |\mathcal{S}_{\text{fit}}|.$$

For each $z \in \mathcal{S}_{\text{code}}$, an approximate maximum-inner-product search retrieves a high-probability head H_z. Writing

$$a_z(i) = \exp(z^\top e(i)/\tau),$$

we evaluate this head exactly and estimate the remaining tail by importance sampling:

$$\widehat{Z}_z = \sum_{i \in H_z} a_z(i) + \frac{1}{N} \sum_{n=1}^{N} \mathbf{1}[j_n \notin H_z] \frac{a_z(j_n)}{q(j_n)}, \tag{24}$$

$$\widehat{M}_z = \sum_{i \in H_z} a_z(i) e(i) + \frac{1}{N} \sum_{n=1}^{N} \mathbf{1}[j_n \notin H_z] \frac{a_z(j_n) e(j_n)}{q(j_n)}, \tag{25}$$

with $j_n \sim q$. We then use

$$\widehat{A}_z = \log \widehat{Z}_z, \qquad \widehat{\mu}_z = \frac{\widehat{M}_z}{\widehat{Z}_z} \tag{26}$$

in the predictive-KL assignment of Eq. (20). The deterministic head captures the dominant predictive mass, while importance sampling corrects the remaining tail.

Predictive medoids and router distillation. To avoid repeated catalog-dependent prototype optimization, the large-scale codebook uses predictive medoids:

$$c_g \in \{z_s : s \in \mathcal{S}_{\text{code}}\}. \tag{27}$$

Assignments and medoid updates can then be carried out using cached predictive statistics.

Rather than recomputing these statistics for every exported memory, we distill the resulting assignments into a lightweight router $R_\phi(g \mid z)$:

$$\mathcal{L}_{\text{router}} = -\sum_{s \in \mathcal{S}_{\text{code}}} w_s \log R_\phi(g_s \mid z_s), \qquad g_s = T(z_s). \tag{28}$$

After distillation, all memories are tokenized by

$$g_{rk} = \arg\max_{g} R_\phi(g \mid z_{rk}), \tag{29}$$

without accessing the item catalog.

For route construction, we precompute the canonical normalizer of each final token,

$$A_g = \log \sum_{i \in \mathcal{C}} \exp\left(\frac{c_g^\top e(i)}{\tau}\right). \tag{30}$$

If several user memories share token g, their current relevance is aggregated as

$$\bar{\alpha}_{rg} = \sum_{k: g_{rk} = g} \alpha_{rk}. \tag{31}$$

The scalable route label is then obtained directly from the canonical token distributions:

$$g^*(r, y) = \arg\max_{g \in \mathcal{G}_r} \left[\log \bar{\alpha}_{rg} + \frac{c_g^\top e(y)}{\tau} - A_g\right], \qquad \mathcal{G}_r = \mathrm{Unique}\{g_{r1}, \ldots, g_{rK}\}. \tag{32}$$

Consequently, expensive catalog-aware computation is restricted to the reservoir and the small set of final token representatives. Ignoring ANN lookup overhead, the dominant catalog-dependent cost changes from

$$O(|\mathcal{S}||\mathcal{C}|d)$$

to approximately

$$O(|\mathcal{S}_{\mathrm{code}}|(M+N)d) + O(G|\mathcal{C}|d),$$

while full-data tokenization is independent of catalog size. The scalable realization introduces approximation through predictive statistics estimation, medoid restriction, and router distillation; we evaluate these approximations separately from downstream recommendation quality.