LEARNING PREDICTIVE EVENT STATES FOR SEQUENTIAL MODELING AND RECOMMENDATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Next-token prediction (NTP) trains causal sequence models for sequential recommendation, but its next-interaction target does not specify which event-level information should be accessible from each causal state. We introduce Event State Prediction (ESP), which augments NTP with two predictive relations over contiguous interaction groups. Intra-event prediction trains token states formed from observed prefixes to predict contextual summaries of their enclosing events; inter-event prediction trains pooled states at event completion to predict the next event before its interactions are observed. Both relations use history-conditioned targets from an exponential-moving-average encoder and single-positive contrastive matching, supervising individual states and event-level aggregates. ESP improves NDCG@10 and NDCG@20 across four datasets while retaining the original NTP inference path. Frozen-state probes on three datasets indicate greater accessibility of future and event-level information on the tested tasks. Geometry analyses find higher effective rank and broader state dispersion, and ablations support the complementary contributions of the two objectives. The code is available at `https://anonymous.4open.science/r/ESP-Code/`.

1 INTRODUCTION

Causal sequence models trained by next-token prediction (NTP) are a common foundation for sequential recommendation, where user interactions are represented as tokens and predicted from their behavioral histories (Rajput et al., 2023; Zhai et al., 2024). At each position, the backbone uses only the observed prefix to form a hidden state, and an output head predicts the next interaction from that state. These states can also serve as reusable representations for downstream tasks (Yuan et al., 2021; 2020; Chen et al., 2025), so what they retain beyond the next interaction matters. We study histories organized into events (Quadrana et al., 2017): contiguous interaction groups defined by supplied boundaries or a predefined rule, such as a calendar-time window. An event may contain multiple interactions, yet NTP supervises each state through only the next-interaction target. We ask how to train causal states to expose information about ongoing and subsequent events that is predictable from their prefixes, while preserving next-interaction recommendation.

Prior work has sought to enrich causal states through broader predictive supervision, either by expanding what is predicted or by changing how prediction targets are represented. We consider these approaches from two complementary perspectives: *supervision granularity* and *state geometry* (Figure 1). Supervision granularity concerns not only what target is predicted, but also which causal state is required to predict it. Discrete-target objectives broaden token-level supervision to multiple future tokens (Gloeckle et al., 2024) or items in a subsequent session (Huang et al., 2025), while representation-target objectives replace discrete observations with encoded targets: NextLat (Teoh et al., 2025) learns transitions to the next token state, NONTP (Li et al., 2026) predicts states at multiple future offsets, and future-summary prediction (Mahajan et al., 2026) targets compact representations of later observations. Target type alone, however, does not determine the requirement imposed on a causal state. Learning a transition after observing the next token differs from requiring a state within an unfolding behavioral episode to capture information about that episode, or requiring a state at its completion to anticipate what follows. Moreover, fixed token offsets need not coincide with meaningful behavioral boundaries. Events provide a natural contextual unit for expressing these distinctions: their boundaries separate information belonging to the current behavioral context from information that remains genuinely future.

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.

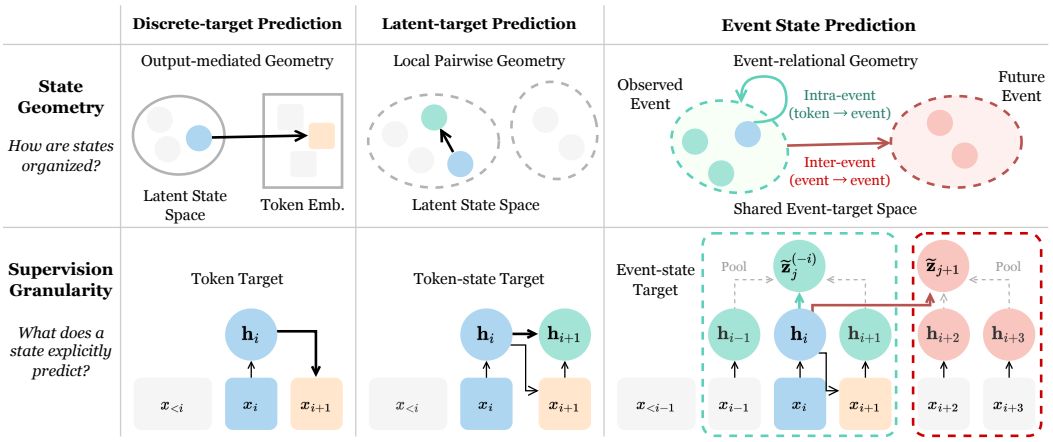


Figure 1: State geometry and supervision granularity in predictive learning. ESP uses token-to-event and event-to-next-event prediction; the latter starts from a pooled contextual event state. Geometric sketches are schematic.

The second perspective, *state geometry*, concerns how predictive objectives organize the hidden representation space. Discrete-token prediction constrains token probabilities through a learned output head, so accurate probabilities do not by themselves imply useful similarities among backbone states; representation degeneration has also been observed in sequential recommendation (Qiu et al., 2022). Representation-target objectives define comparisons with encoded observations, often through prediction heads. Their targets are commonly local token transitions, fixed future offsets, or summaries of later observations. Such targets need not reflect the higher-level structure of a behavioral history: states produced at different positions within the same event may share a common context, while transitions across event boundaries may correspond to a qualitatively different predictive requirement. Consequently, existing objectives may broaden future supervision without explicitly requiring the corresponding causal states to retain event-aligned information at the positions where it is needed. We learn these relations through representation matching and assess the geometry of the retained states independently of the training heads.

To impose these requirements on the corresponding causal representations, we use the event partition to define two complementary predictive relations at different source levels. Intra-event prediction assigns each eligible token state a summary of its enclosing event, including later observations when the source lies inside that event. Inter-event prediction treats a completed event as a unit: its pooled contextual state predicts the next event before any destination-event token is observed. With its prediction head fixed, a pooled query cannot distinguish token-state changes that preserve its event mean, whereas predicting enclosing-event content does not explicitly require prediction across the event boundary. Combining the two relations therefore supervises both individual states within events and their contextual aggregates at completion. The partition defines the supervision groups without assuming that each group expresses a single latent intent.

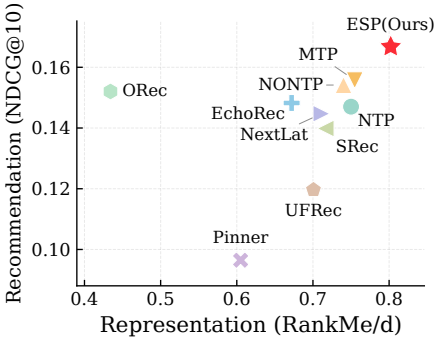


Figure 2: Frozen-state geometry versus next-token recommendation. Each point reports the mean RankMe/d and NDCG@10 across MerRec, RetailRocket, and KuaiRec.

We instantiate these two predictive relations as Event State Prediction (ESP), a training framework that augments NTP with two contrastive objectives over contextual event targets. ESP encourages causal states to capture information about their enclosing and subsequent events. A causal online encoder produces token-level and pooled event source states, while an exponential moving average (EMA) target encoder produces the corresponding event targets. Both encoders retain preceding context through global causal attention; event boundaries define pooling groups rather than restricting attention. For intra-event prediction, leave-one-out pooling excludes the source position's teacher

state from the event average. Separate prediction heads map the two source types into the shared target space, where a single-positive contrastive objective matches each prediction to its assigned event target against alternative event instances. The two objectives are jointly optimized with NTP. After training, the target encoder and event-prediction heads are discarded, leaving the original causal backbone and NTP inference path unchanged.

Our experiments examine whether ESP improves recommendation and whether its supervision enriches the learned causal states beyond the immediate next-token objective. ESP improves recommendation across four datasets, while frozen-state probes on MerRec, KuaiRec, and AppRec assess the accessibility of future and event-level information. On MerRec, the same frozen ESP encoder achieves the highest mean Macro-F1 on all eight future-category tasks, including queries before an event ends. These findings indicate that the benefits extend beyond the trained NTP head. As summarized in Figure 2, ESP combines strong recommendation performance with better representation quality, reflected here by the higher effective rank of its frozen states. Further analyses and ablations support the benefits of event-level supervision and the complementary roles of the two predictive objectives, without changing the deployed inference path. Overall, our contributions are as follows:

- We formulate event-level predictive requirements at token positions and event boundaries, and analyze their distinct constraints on individual and pooled representations.
- We introduce ESP, which combines intra-event and inter-event prediction through contextual momentum targets and contrastive matching while retaining the original NTP inference path.
- We show that event-supervised training improves recommendation and the accessibility of future behavioral information in frozen states, with geometric diagnostics and ablations characterizing the effects and limits of the learned representations.

2 Related Work

2.1 Token- and Event-Level Predictive Supervision

In sequential recommendation, Kang & McAuley (2018) learn a history representation by predicting the next item, while Sun et al. (2019) reconstruct selected observations from bidirectional context through masked-item pretraining. For generative retrieval, Rajput et al. (2023) predict tokenized item identifiers autoregressively, and Zhai et al. (2024) extend predictive modeling to heterogeneous behavioral streams with large-scale sequential transducers.

Multi-token prediction (MTP) (Gloeckle et al., 2024) trains several future-token heads from a shared context. In recommendation, UFRec (Cui et al., 2026) weights multi-step item supervision by predictive uncertainty, while EchoRec (Ma et al., 2026) connects future-item prediction branches and aligns their aggregate preference with the decoding representation. PinnerFormer (Pinner) (Pancha et al., 2022) supervises long-term engagement within a future window. SessionRec (SRec) (Huang et al., 2025) aggregates interactions into session representations and predicts the positively interacted items in the next session. S^3-Rec (Zhou et al., 2020) also uses subsequence-level supervision through a segment-prediction objective. Sessions, subsequences, and future windows thus already provide grouped supervision, with different target constructions and source representations.

Continuous targets further extend the supervised content. Future-summary prediction (Mahajan et al., 2026) includes targets encoded by a reverse language model, and Oracle4Rec (ORec) (Xia et al., 2024) aligns past representations with a future-information encoder. NONTP (Li et al., 2026) supervises a future trajectory through offset-specific predictions of individual momentum token states. ESP instead pools contextual states into event targets and assigns them to two causal sources: a token state predicts its enclosing event, and a completed event's pooled state predicts the next event before any destination token is observed. These event-level objectives augment NTP. ESP's distinction lies in the contextual event targets and their assignment to token and event source states; recommendation retains the causal encoder and NTP head at inference.

2.2 Latent Prediction and State Geometry

Littman et al. (2001) characterize a state through predictions of future observations. CPC (van den Oord et al., 2018) learns such representations by discriminating future latent observations; data2vec (Baevski et al., 2022) predicts contextualized teacher states, and I-JEPA (Assran et al.,

2023) predicts target-block representations from visible image context. For autoregressive sequences, NextLat (Teoh et al., 2025) learns a latent transition conditioned on the current state and the next token, including multi-step training. MoCo (He et al., 2020) uses momentum encoders to provide slowly changing keys, while BYOL (Grill et al., 2020) and DINO (Caron et al., 2021) learn from stopped-gradient teacher targets updated from the online network.

The matching objective shapes representation geometry as well as prediction accuracy. For normalized embeddings, Wang & Isola (2020) analyze contrastive learning through positive-pair alignment and distributional uniformity. Chen et al. (2020) use a learned projection head to separate the features used by the contrastive loss from those evaluated downstream. Which representations should align depends on the positive relation. CL4SRec (Xie et al., 2022) pairs augmented views of a sequence. DuoRec (Qiu et al., 2022) addresses representation degeneration using dropout views and histories sharing the same next-item target. ICLRec (Chen et al., 2022) aligns sequence views with latent intent prototypes inferred by clustering.

ESP defines positives by an assigned event instance and a predictive relationship: each query predicts its enclosing or subsequent event target and is contrasted with targets from other event instances. These event-defined pairs connect supervision granularity to the similarities learned through prediction heads, while the geometry of the retained causal states is evaluated independently.

3 Problem Formulation

We study how a given event partition can enrich predictive supervision for causal sequence models.

3.1 Causal States and Event Partitions

Let $X = (x_1, \ldots, x_n)$ be a sequence of observations represented as tokens, with $x_i \in \mathcal{X}$. In our behavioral sequence setting, tokens encode items, actions, or their attributes. The sequence is partitioned into m ordered, contiguous events. Let $e_i \in \{1, \ldots, m\}$ denote the event assignment of token x_i, and let $G_j = \{i : e_i = j\}$ contain the positions of event j. We use a given partition whose boundaries are supplied by the data or a predefined grouping rule, such as a calendar-time window.

A causal sequence encoder produces a token state from the observed prefix:

$$\mathbf{h}_i = f_\theta(x_{1:i}) \in \mathbb{R}^d. \tag{1}$$

An output head $p_\phi(\cdot \mid \mathbf{h}_i)$ predicts the next token. The next-token prediction (NTP) objective is

$$\mathcal{L}_{\text{NTP}} = -\frac{1}{|\mathcal{I}|} \sum\nolimits_{i \in \mathcal{I}} \log p_\phi(x_{i+1} \mid \mathbf{h}_i), \tag{2}$$

where $\mathcal{I}$ contains positions with valid next-token targets.

We also consider a contextual event state $\mathbf{z}_j$ formed from the token states of G_j after event j is complete. Writing $b_j = \max G_j$, this state has access only to $x_{1:b_j}$. Section 4.1 specifies the aggregation used by ESP.

3.2 Predictive Requirements

NTP constrains the distribution of x_{i+1} predicted from $\mathbf{h}_i$, but does not explicitly require that state to preserve all prefix information useful for an enclosing-event target. In an attention-based encoder, later states can revisit the prefix, so their predictions do not impose the same information requirement on $\mathbf{h}_i$. Appendix B formalizes this distinction through predictive sufficiency.

We therefore consider two additional predictive requirements. At a token position $i \in G_j$, $\mathbf{h}_i$ should retain prefix information useful for predicting the contextual content of its enclosing event j. At event completion b_j, the contextual event state $\mathbf{z}_j$ should retain information predictive of event $j+1$. The first requirement includes prediction from interior positions, while the second excludes all destination-event observations from its source.

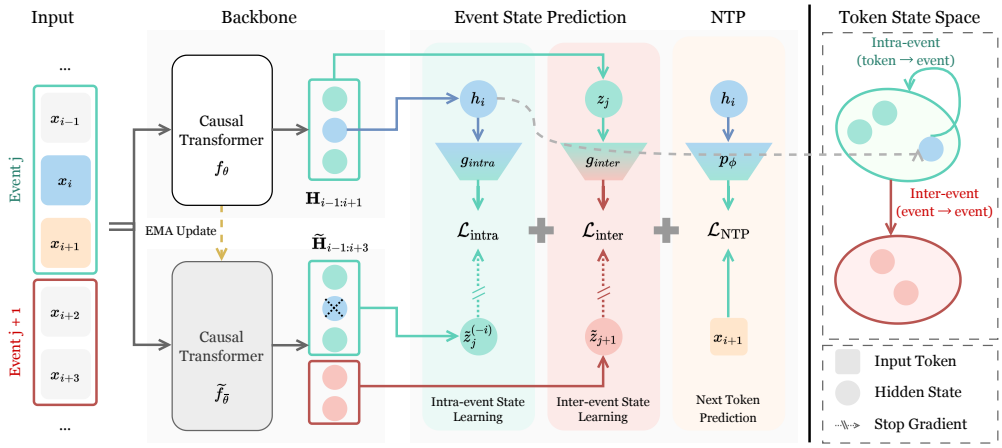


Figure 3: ESP training framework. Alongside NTP, token states predict leave-one-out targets within events, while pooled contextual event states predict the next event. A shared EMA encoder supplies the event targets. The right-hand sketch illustrates the predictive relations; contrastive scores use the prediction-head outputs and event targets.

4 Event State Prediction

ESP instantiates the predictive requirements in Section 3 by matching causal source states to contextual event targets (Figure 3). A shared EMA target encoder supplies representations for enclosing and subsequent events, while separate prediction heads implement the two source–target relations.

4.1 Contextual Event States and Targets

We represent each event in its preceding behavioral context. The online encoder f_θ uses global causal attention: each token can attend to preceding positions across event boundaries, while future positions are masked. A target encoder $\widetilde{f}_{\bar{\theta}}$ has the same architecture and attention pattern, and produces $\widetilde{\mathbf{h}}_i = \widetilde{f}_{\bar{\theta}}(x_{1:i})$. Its parameters track the online encoder through an exponential moving average:

$$\bar{\theta} \leftarrow \mu\bar{\theta} + (1-\mu)\theta, \qquad \mu \in [0, 1). \tag{3}$$

This branch supplies slowly evolving targets shared by both predictive relations (He et al., 2020; Grill et al., 2020).

For a nonempty set of positions G, define mean pooling as

$$\mathrm{Pool}(\mathbf{H}, G) = \frac{1}{|G|}\sum\nolimits_{i\in G} \mathbf{h}_i, \qquad \mathbf{H} = (\mathbf{h}_1, \ldots, \mathbf{h}_n). \tag{4}$$

Writing $\widetilde{\mathbf{H}} = (\widetilde{\mathbf{h}}_1, \ldots, \widetilde{\mathbf{h}}_n)$, we pool both branches over the same event groups:

$$\mathbf{z}_j = \mathrm{Pool}(\mathbf{H}, G_j), \qquad \widetilde{\mathbf{z}}_j = \mathrm{Pool}(\widetilde{\mathbf{H}}, G_j). \tag{5}$$

4.2 Intra- and Inter-Event Prediction

Two independent prediction heads, g_{intra} and g_{inter}, map the source states into the shared target space. Both are bias-free $d \times d$ linear maps initialized to the identity. We write sg for stop-gradient.

Intra-event prediction. To supervise individual token states with enclosing-event information, we assign each eligible $\mathbf{h}_i$ a contextual target pooled from the other positions in G_j. For $i \in G_j$ with $|G_j| > 1$, define

$$\widetilde{\mathbf{z}}_j^{(-i)} = \mathrm{Pool}(\widetilde{\mathbf{H}}, G_j \setminus \{i\}). \tag{6}$$

The query and its paired key are

$$\mathbf{q}_i^{\mathrm{intra}} = g_{\mathrm{intra}}(\mathbf{h}_i), \qquad \mathbf{k}_i^{\mathrm{intra}} = \mathrm{sg}\left(\widetilde{\mathbf{z}}_{e_i}^{(-i)}\right). \tag{7}$$

The source accesses only $x_{1:i}$. At interior positions $i < b_j$, the target includes later positions in the same event; at $i = b_j$, it summarizes enclosing-event content already observed by the source. Leave-one-out excludes $\widetilde{\mathbf{h}}_i$ from pooling but retains x_i in the teacher input, so later teacher states can still depend on x_i.

Inter-event prediction. For each adjacent pair $(j, j+1)$ in the same sequence, with $j < m$, the completed event's pooled state predicts its successor:

$$\mathbf{q}_j^{\text{inter}} = g_{\text{inter}}(\mathbf{z}_j), \qquad \mathbf{k}_j^{\text{inter}} = \text{sg}(\widetilde{\mathbf{z}}_{j+1}). \tag{8}$$

The source is available at b_j, before any token in G_{j+1} is observed. Destination-event observations enter only the contextual training target.

The two relations constrain different source representations. With teacher targets and prediction heads fixed, token-state changes satisfying $\sum_{i \in G_j} \Delta \mathbf{h}_i = 0$ preserve the event mean and leave the inter-event query unchanged. Intra-event prediction acts on individual states and can constrain such changes when they alter relative scores among its candidate targets. Conversely, its enclosing-event target does not explicitly impose prediction of the successor event. The relations therefore complement each other in source granularity and prediction scope.

4.3 Contrastive Training and Inference

Given these source–target assignments, we train each prediction to distinguish its assigned event target from targets of other event instances. For relation $r \in \{\text{intra}, \text{inter}\}$ and anchor a, let $\mathcal{B}_a^r$ contain its paired key $\mathbf{k}_a^r$ and negative keys from other event instances. An event instance is identified by its sequence and local event index. Each anchor has one positive, even when the batch contains several anchors from its target event. The InfoNCE (van den Oord et al., 2018) objective is

$$\ell_a^r = -\log \frac{\exp\big((\mathbf{q}_a^r)^\top \mathbf{k}_a^r / \tau\big)}{\sum_{\mathbf{k} \in \mathcal{B}_a^r} \exp((\mathbf{q}_a^r)^\top \mathbf{k} / \tau)}, \tag{9}$$

where $\tau > 0$ is the temperature. With fixed linear heads and teacher keys, the local curvature depends on the candidate-key covariance projected back to the source representation (Appendix B). Section 5.5 evaluates the global geometry of the retained backbone states.

Let $\mathcal{A}_{\text{intra}}$ contain selected token positions in non-singleton events and $\mathcal{A}_{\text{inter}}$ contain selected events with a successor in the same sequence. Eligible anchors may be subsampled to limit contrastive computation. For each nonempty anchor set, we average

$$\mathcal{L}_r = \frac{1}{|\mathcal{A}_r|} \sum\nolimits_{a \in \mathcal{A}_r} \ell_a^r, \qquad r \in \{\text{intra}, \text{inter}\}. \tag{10}$$

The joint objective augments NTP from Equation 2:

$$\mathcal{L}_{\text{ESP}} = \mathcal{L}_{\text{NTP}} + \lambda_{\text{intra}} \mathcal{L}_{\text{intra}} + \lambda_{\text{inter}} \mathcal{L}_{\text{inter}}, \tag{11}$$

where $\lambda_{\text{intra}}, \lambda_{\text{inter}} \geq 0$ weight the event losses. We train the NTP term using sampled softmax with 128 sampled negatives and evaluate recommendation over the full catalog. Appendix D gives the shared training settings and ESP hyperparameters.

NTP and both event losses update the online backbone, with each prediction head optimized by its own objective. The target encoder receives no gradients and updates only through Equation 3. After training, we retain the online backbone and NTP head for recommendation. Frozen-state evaluations use token representations before the training heads, including $\mathbf{h}_{b_j}$ for End probes (Section 5.4). Algorithm 1 in Appendix A summarizes training.

5 Experiments

Our experiments examine whether ESP improves next-token recommendation (RQ 1), whether the gains extend to more distant tokens (RQ 2), and whether frozen states expose more future and event-level information (RQ 3). We then analyze frozen-state geometry (RQ 4) and the contributions of the event objectives and contrastive matching (RQ 5).

Table 1: Downstream next-token prediction (NTP) results on four recommendation datasets. N@K and H@K denote NDCG@K and HR@K. Best and second-best results are shown in **bold** and underlined; yellow and blue gradients indicate the top 4 and bottom 4 results. † and ‡ denote statistically significant improvements over NTP and the best baseline ($p < 0.05$).

Dataset	Metric	NTP	MTP	NextLat	NONTP	Pinner	SRec	EchoRec	ORec	UFRec	ESP
MerRec	N@10	0.2048	0.2176	0.1769	0.2217	0.1927	0.1819	0.1956	0.2049	0.1737	**0.2307**‡
	N@20	0.2204	0.2321	0.1919	0.2364	0.2062	0.1969	0.2110	0.2214	0.1904	**0.2459**‡
	H@10	0.3187	0.3351	0.2777	0.3326	0.3047	0.2966	0.3029	0.3633	0.2840	**0.3803**‡
	MRR	0.2031	0.2137	0.1813	**0.2151**	0.1918	0.1808	0.1968	0.1868	0.1750	0.2140†
RetailRocket	N@10	0.1200	0.1356	0.1434	0.1247	0.0720	0.1251	0.1333	0.1348	0.0696	**0.1520**‡
	N@20	0.1247	0.1411	0.1483	0.1295	0.0776	0.1287	0.1396	0.1415	0.0732	**0.1563**‡
	H@10	0.1577	0.1756	0.1826	0.1586	0.1016	0.1533	**0.1886**	0.1841	0.0924	0.1853†
	MRR	0.1486	0.1628	0.1704	0.1545	0.1061	0.1568	0.1551	0.1587	0.1061	**0.1807**‡
KuaiRec	N@10	0.1163	0.1139	0.1138	0.1161	0.0247	0.1125	0.1158	0.1164	0.1157	**0.1177**‡
	N@20	0.1276	0.1255	0.1250	0.1274	0.0345	0.1224	0.1271	0.1276	0.1271	**0.1288**‡
	H@10	0.1572	0.1546	0.1532	0.1568	0.0515	0.1496	0.1569	0.1583	0.1563	**0.1587**†
	MRR	0.1450	0.1429	0.1432	0.1450	0.0626	0.1426	0.1445	0.1447	0.1446	**0.1463**‡
AppRec	N@10	0.6328	0.6500	0.6460	0.6543	0.5304	0.6311	0.6487	0.5671	0.6640	**0.6747**‡
	N@20	0.6455	0.6622	0.6585	0.6660	0.5499	0.6420	0.6600	0.5840	0.6745	**0.6854**‡
	H@10	0.8413	0.8674	0.8556	0.8658	0.7713	0.8447	0.8538	0.7818	0.8696	**0.8794**‡
	MRR	0.5747	0.5875	0.5854	0.5938	0.4664	0.5707	0.5907	0.5105	0.6050	**0.6155**‡

5.1 Experimental Setup

We evaluate MerRec (Li et al., 2024), RetailRocket (Zykov, 2017), KuaiRec (Gao et al., 2022), and the anonymized industrial dataset AppRec. Chronologically ordered interactions form non-overlapping sequences of at most 128 tokens. The public datasets use a deterministic 70/30 user split, while AppRec has a test set of 256k users. Within each dataset, methods share the sequences, vocabulary, and full-catalog candidates. Appendix C describes preprocessing and dataset sizes.

The main comparisons evaluate training objectives on a shared HSTU backbone (Zhai et al., 2024). As a method label, NTP denotes this backbone trained with the next-token objective alone; the other methods are MTP (Gloeckle et al., 2024), NextLat (Teoh et al., 2025), NONTP (Li et al., 2026), Pinner (Pancha et al., 2022), SRec (Huang et al., 2025), EchoRec (Ma et al., 2026), ORec (Xia et al., 2024), and UFRec (Cui et al., 2026). ESP uses the objective in Equation 11, and recommendation uses the resulting backbone and NTP head. ESP's default event partition is the natural hour. Appendix D specifies the architecture and implementation settings, with experiment-specific configurations given alongside supplementary results.

Following Huang et al. (2023), we report NDCG@10, NDCG@20, HR@10, and MRR for next-token recommendation. Frozen linear probes separately test whether future category labels, excluded from backbone inputs, are accessible through a learned linear map. We then measure width-normalized RankMe (Garrido et al., 2023), IsoScore (Rudman et al., 2022), uniformity (Wang & Isola, 2020), and mean-direction anisotropy (Ethayarajh, 2019) on final valid causal states before task-specific heads. Each geometry evaluation uses 16,384 states, or all 2,081 held-out sequences on RetailRocket. Model training uses seed 1234. Probe error bars measure variation across three probe-head initializations, rather than variation across independently trained backbones. Following Xin et al. (2026), we use a permutation test (Smucker et al., 2007) to assess statistical significance.

5.2 RQ 1: Next-Token Recommendation

We first compare next-token recommendation on the shared HSTU backbone (Table 1). (1) Event supervision consistently improves next-token ranking. ESP outperforms the NTP baseline on every reported dataset and metric, and leads both NDCG measures on all four datasets. Each NDCG lead over the strongest competing method passes the reported randomization test. ESP retains only the causal backbone and NTP head at inference. The gains therefore suggest that its event-level training makes the deployed causal states more useful for ranking the next interaction. (2) The benefits of other future objectives vary across datasets. NONTP is competitive on MerRec but less so on RetailRocket, while NextLat stands out on RetailRocket and trails NTP on MerRec. UFRec is the

strongest competing method on AppRec but falls below NTP on MerRec and RetailRocket. These shifts show that broader future supervision does not consistently help the same causal backbone. ESP's two event assignments specify both the target and the causal state required to predict it: token states predict their enclosing events, and completed event states predict subsequent events. (3) ESP's advantage is most consistent on NDCG. It also leads HR@10 and MRR on three datasets each, but EchoRec has the best HR@10 on RetailRocket and NONTP has the best MRR on MerRec. The NDCG@10 margin over the closest baseline is widest on RetailRocket and narrowest on KuaiRec. The size of the gain varies by dataset and metric, with the most stable evidence coming from the two NDCG measures. Appendix F.7 examines a later temporal snapshot.

5.3 RQ 2: Future-Token Retrieval

We next test whether ESP's gains extend beyond the immediate next token. For each held-out MerRec sequence, we select the ninth token from the end as the causal query. Its state and the trained retrieval path remain fixed while the target changes to offset $h \in \{1, 2, 4, 8\}$. Evaluation ranks the full catalog, without horizon-specific fine-tuning or rollout. The $h = 1$ result therefore concerns one selected query per sequence, whereas Table 1 evaluates every valid transition. ESP leads NDCG@10 at every tested horizon, and its advantage over MTP widens beyond the immediate target (Figure 4). Accuracy still declines with distance for all compared methods, but ESP declines less than the strongest comparator.

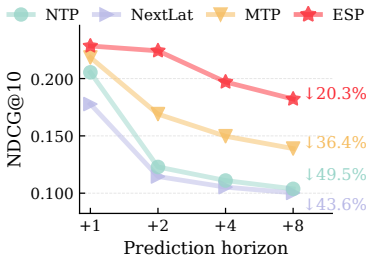


Figure 4: MerRec future-token NDCG@10 from a fixed causal query.

5.4 RQ 3: Future and Event Information in Frozen States

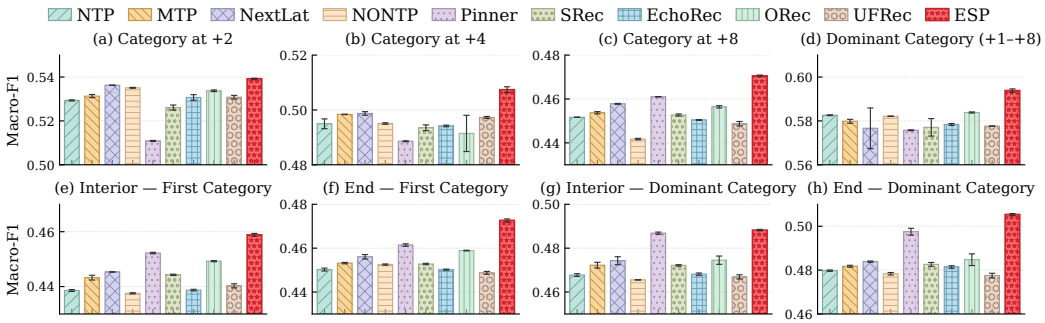


Figure 5: Frozen linear-probe Macro-F1 on MerRec future-category tasks. Top: fixed-offset and future-window targets. Bottom: final-event targets from Interior and End queries in the penultimate event.

To assess future information independently of the trained retrieval head, we freeze the backbone and embeddings, discard the training heads, and fit a separate 17-way linear classifier for each MerRec task. Category labels come from a separate product catalog and are excluded from backbone inputs.

One protocol uses the ninth token from the end as a common query for category prediction at offsets 2, 4, and 8 and for the dominant category among the next eight tokens, using 106,400 held-out sequences. A second protocol predicts the first or dominant category of the final natural-hour event from the penultimate event. Interior uses a deterministically sampled non-final position; End uses its final position. Both views contain the same 76,834 test sequences and encode only the prefix through the query. All eight tasks use the same ESP checkpoint, frozen throughout probe fitting. Appendix E.1 provides probe training details and the complete metric table.

ESP leads Macro-F1 at each tested token offset and for the dominant category of the next-eight-token window (Figure 5). Its Macro-F1 decreases with prediction distance. ESP leads both final-event targets from Interior and End queries. Its End readouts are stronger, while the narrow Interior dominant-category margin still shows that future-event information is linearly accessible before the

Table 2: Frozen representation quality on MerRec. RM/d, Iso., Unif., and Aniso. denote RankMe/d, IsoScore, uniformity, and mean-direction anisotropy.

Metric	NTP	MTP	NextLat	NONTP	Pinner	SRec	EchoRec	ORec	UFRec	**ESP**
RM/d ↑	0.7006	0.6723	0.6133	0.6693	0.5692	0.6813	0.6165	0.3128	0.6615	**0.7316**‡
Iso. ↑	**0.2483**	0.2287	0.1701	0.2361	0.1666	0.1345	0.2157	0.0064	0.1651	0.2443
Unif. ↓	-2.9855	-2.6854	-3.0107	-2.7537	-3.1724	-2.7113	-3.5247	-0.9320	-3.2629	**-3.5928**‡
Aniso. ↓	0.2261	0.3051	0.2092	0.2811	0.1637	0.2798	0.0822	0.4948	0.1400	**0.0675**‡

Table 3: ESP ablations on MerRec. Parentheses report relative-change magnitudes from full ESP computed from the displayed values.

Variant	**Representation Quality**				**Downstream Retrieval**	
	RankMe/d ↑	**IsoScore** ↑	**Unif.** ↓	**Aniso.** ↓	**NDCG@10** ↑	**HR@10** ↑
ESP	0.7316	0.2443	-3.5928	0.0675	0.2307	0.3803
w/o $\mathcal{L}_{\text{intra}}$	0.7071 (↓3.35%)	0.1713 (↓29.88%)	-3.3493 (↑6.78%)	0.1211 (↑79.41%)	0.2056 (↓10.88%)	0.3460 (↓9.02%)
w/o $\mathcal{L}_{\text{inter}}$	0.7267 (↓0.67%)	0.2405 (↓1.56%)	-3.5542 (↑1.07%)	0.0767 (↑13.63%)	0.2202 (↓4.55%)	0.3345 (↓12.04%)
w/ cosine	0.7214 (↓1.39%)	0.2123 (↓13.10%)	-3.0338 (↑15.56%)	0.1958 (↑190.07%)	0.2033 (↓11.88%)	0.3345 (↓12.04%)

event boundary. Appendices E.1.6 and E.1.7 examine other targets; Appendix F.2 examines retrieval of complete event content.

5.5 RQ 4: Frozen-State Geometry

We next examine frozen-state geometry on MerRec (Table 2). ESP has the highest RankMe/d and the lowest uniformity and mean-direction anisotropy. NTP has slightly higher IsoScore, so the two spectral diagnostics do not rank these models identically. Lower uniformity indicates broader dispersion of normalized states. The diagnostic order does not determine recommendation performance: NTP leads IsoScore but trails ESP on NDCG@10, while ORec has weak spectral scores and a strong HR@10. Appendix E.1.3 reports RetailRocket, KuaiRec, and AppRec.

5.6 RQ 5: Ablation Study

To assess the contribution of ESP's objectives, we remove each event objective in turn or replace InfoNCE with cosine regression. All variants retain NTP. The full-ESP reference matches the main recommendation and geometry tables. On MerRec, removing either event objective lowers NDCG@10 and HR@10 (Table 3). Intra-event removal causes the larger NDCG@10 loss, whereas inter-event removal causes the larger HR@10 loss. The full objective leads both recommendation metrics within this ablation suite. Cosine regression causes the largest NDCG@10 loss. Appendix E.2 reports the RetailRocket ablations, and Appendix F.3 examines model width, event partitions, and EMA momentum.

6 Conclusion

We introduced ESP, which augments next-token prediction with explicit event-state supervision. Token states predict their enclosing events, and completed contextual event states predict subsequent events. Contrastive matching against EMA targets learns both event relationships. Across four datasets, ESP achieves better recommendation performance, while its frozen states have the highest RankMe/d and lowest uniformity and mean-direction anisotropy. The same states also lead Macro-F1 on frozen future-category probes. Ablations show that both event objectives contribute to NDCG@10 and that replacing contrastive matching with cosine regression weakens recommendation. Together, these findings support event-level predictive supervision as a way to improve recommendation and enrich the future information accessible in causal states.

AI USE STATEMENT

Generative AI tools were used to assist with code writing, manuscript organization and revision, and literature discovery and citation verification against primary sources. The authors remain responsible for the implementation and all manuscript content.

ETHICS STATEMENT

This study uses public recommendation datasets and an anonymized industrial interaction dataset. As described in Appendix C, AppRec contains anonymized user and application identifiers, and we report only aggregate dataset statistics and model-level evaluation results. The experiments are conducted offline and do not intervene in user behavior. As with other models learned from behavioral logs, the resulting representations may reflect historical exposure and popularity biases. Their predictions should not be interpreted as causal measurements of user intent. Any deployment should assess privacy, fairness, and feedback-loop risks in its application context.

REPRODUCIBILITY STATEMENT

The main text defines the event construction, target representations, and ESP training objective. The full implementation can be found in our source code. Appendix C documents preprocessing, data splits, and dataset statistics. Appendix D specifies the shared HSTU architecture, baseline adaptations, optimization settings, and ESP hyperparameters, while Appendix D.5 defines the full-catalog ranking protocol. Appendix E.1 describes frozen-feature extraction and linear-probe training. We report the random-seed policy and distinguish variation across probe-head initializations from variation across independently trained encoders.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael G. Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629, 2023.

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. data2vec: A general framework for self-supervised learning in speech, vision and language. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 1298–1312, 2022.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021. doi: 10.48550/arXiv.2104.14294.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709, 2020.

Xiangyi Chen, Kousik Rajesh, Matthew Lawhon, Zelun Wang, Hanyu Li, Hao-Miao Li, Saurabh Vishwas Joshi, Pong Eksombatchai, Jaewon Yang, Yi-Ping Hsu, Jiajing Xu, and Chuck Rosenberg. Pinfm: Foundation model for user activity sequences at a billion-scale visual discovery platform. *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, 2025.

Yongjun Chen, Zhiwei Liu, Jia Li, Julian McAuley, and Caiming Xiong. Intent contrastive learning for sequential recommendation. *Proceedings of the ACM Web Conference 2022*, 2022.

Ziqiang Cui, Xing Tang, Peiyang Liu, Xiaokun Zhang, Shiwei Li, Xiuqiang He, and Chen Ma. Looking farther with confidence: Uncertainty-guided future learning for sequential recommendation. *arXiv preprint arXiv:2605.28493*, 2026. doi: 10.48550/arXiv.2605.28493.

Kawin Ethayarajh. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 55–65, 2019. doi: 10.18653/v1/D19-1006.

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, 2022. doi: 10.1145/3511808.3557220.

Quentin Garrido, Randall Balestriero, Laurent Najman, and Yann LeCun. RankMe: Assessing the downstream performance of pretrained self-supervised representations by their rank. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 10929–10974, 2023.

Fabian Gloeckle, Badr Youbi Idrissi, Baptiste Roziere, David Lopez-Paz, and Gabriel Synnaeve. Better & faster large language models via multi-token prediction. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 15706–15734, 2024.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 21271–21284, 2020.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9726–9735, 2020.

Chengkai Huang, Shoujin Wang, Xianzhi Wang, and Lina Yao. Modeling temporal positive and negative excitation for sequential recommendation. In *Proceedings of the ACM Web Conference 2023*, pp. 1252–1263, 2023. doi: 10.1145/3543507.3583463.

Lei Huang, Hao Guo, Linzhi Peng, Long Zhang, Xiaoteng Wang, Daoyuan Wang, Shichao Wang, Jinpeng Wang, Lei Wang, and Sheng Chen. SessionRec: Next session prediction paradigm for generative sequential recommendation. *arXiv preprint arXiv:2502.10157*, 2025. doi: 10.48550/arXiv.2502.10157.

Wang-Cheng Kang and Julian J. McAuley. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining*, pp. 197–206, 2018. doi: 10.1109/ICDM.2018.00035.

Changhao Li, Shuli Wang, Junwei Yin, Senjie Kou, Yinqiu Huang, Chi Wang, Yinhua Zhu, Haitao Wang, and Xingxing Wang. Not only NTP: Extending training signal coverage for generative recommendation. *arXiv preprint arXiv:2607.12277*, 2026. doi: 10.48550/arXiv.2607.12277.

Lichi Li, Zainul Abi Din, Zhen Tan, Sam London, Tianlong Chen, and Ajay Daptardar. MerRec: A large-scale multipurpose Mercari dataset for consumer-to-consumer recommendation systems. *arXiv preprint arXiv:2402.14230*, 2024. doi: 10.48550/arXiv.2402.14230.

Michael L. Littman, Richard S. Sutton, and Satinder Singh. Predictive representations of state. In *Neural Information Processing Systems*, 2001.

Haokai Ma, Aoqi Hu, Yueao Xing, Ruobing Xie, Yonghui Yang, Teng Tu, Lei Meng, and Tat-Seng Chua. EchoRec: Multi-item prediction-empowered generative recommendation via cycle-consistent preference alignment. *arXiv preprint arXiv:2608.14011*, 2026. doi: 10.48550/arXiv.2608.14011.

Divyat Mahajan, Sachin Goyal, Badr Youbi Idrissi, Mohammad Pezeshki, Ioannis Mitliagkas, David Lopez-Paz, and Kartik Ahuja. Beyond multi-token prediction: Pretraining LLMs with future summaries. In *The Fourteenth International Conference on Learning Representations*, 2026.

Nikil Pancha, Andrew Zhai, Jure Leskovec, and Charles Rosenberg. PinnerFormer: Sequence modeling for user representation at Pinterest. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3702–3712, 2022. doi: 10.1145/3534678.3539156.

Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. Contrastive learning for representation degeneration problem in sequential recommendation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pp. 813–823, 2022. doi: 10.1145/3488560.3498433.

Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. Personalizing session-based recommendations with hierarchical recurrent neural networks. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pp. 130–137, 2017. doi: 10.1145/3109859.3109896.

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, 2023. doi: 10.52202/075280-0452.

William Rudman, Nate Gillman, Taylor Rayne, and Carsten Eickhoff. IsoScore: Measuring the uniformity of embedding space utilization. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 3325–3339, 2022. doi: 10.18653/v1/2022.findings-acl.262.

Mark D. Smucker, James Allan, and Ben Carterette. A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management*, pp. 623–632, 2007. doi: 10.1145/1321440.1321528.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 1441–1450, 2019. doi: 10.1145/3357384.3357895.

Jayden Teoh, Manan Tomar, Kwangjun Ahn, Edward S. Hu, Pratyusha Sharma, Riashat Islam, Alex Lamb, and John Langford. Next-latent prediction transformers learn compact world models. *ArXiv*, abs/2511.05963, 2025.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. doi: 10.48550/arXiv.1807.03748.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 9929–9939, 2020.

Jiafeng Xia, Dongsheng Li, Hansu Gu, Tun Lu, Peng Zhang, Li Shang, and Ning Gu. Oracle-guided dynamic user preference modeling for sequential recommendation. *arXiv preprint arXiv:2412.00813*, 2024. doi: 10.48550/arXiv.2412.00813.

Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. Contrastive learning for sequential recommendation. In *2022 IEEE 38th International Conference on Data Engineering*, pp. 1259–1273, 2022. doi: 10.1109/ICDE53745.2022.00099.

Haidong Xin, Zhenghao Liu, Sen Mei, Yukun Yan, Shi Yu, Shuo Wang, Zulong Chen, Yu Gu, Ge Yu, and Chenyan Xiong. LISRec: Modeling user preferences with learned item shortcuts for sequential recommendation. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, pp. 1650–1661, 2026. doi: 10.1145/3770854.3780337.

Fajie Yuan, Xiangnan He, Alexandros Karatzoglou, and Liguang Zhang. Parameter-efficient transfer from sequential behaviors for user modeling and recommendation. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020.

Fajie Yuan, Guoxiao Zhang, Alexandros Karatzoglou, Joemon Jose, Beibei Kong, and Yudong Li. One person, one model, one world: Learning continual user representation without forgetting. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, pp. 696–705, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380379. doi: 10.1145/3404835.3462884.

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.

Xiangdong Zhang, Debing Zhang, Shaofeng Zhang, Xiaohan Qin, Yu Cheng, and Junchi Yan. NITP: Next implicit token prediction for LLM pre-training. *arXiv preprint arXiv:2605.24956*, 2026. doi: 10.48550/arXiv.2605.24956.

Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. S^3-Rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, pp. 1893–1902, 2020. doi: 10.1145/3340531.3411954.

Roman Zykov. Retailrocket recommender system dataset. Kaggle, 2017.

A ESP TRAINING ALGORITHM

Algorithm 1 follows the construction in Sections 4.1–4.3: contextual encoding and pooling, source–target assignment, contrastive matching, and joint optimization. Superscript u identifies a sequence in the mini-batch, so (u, j) identifies an event instance; t_a^r records the assigned target event for anchor a in relation r. Both encoders use global causal attention within each sequence. The loss averages below apply to nonempty selected anchor sets, as in Equation 10.

Algorithm 1: Event State Prediction (ESP) training

Input : Mini-batches $\{X^u, \{G_j^u\}_{j=1}^{m_u}\}_u$ with given event partitions; temperature τ, loss weights $\lambda_{\text{intra}}, \lambda_{\text{inter}}$, and EMA momentum μ.

Output : Trained causal encoder f_θ and NTP head p_ϕ.

Initialize f_θ and p_ϕ; set $\bar{\theta} \leftarrow \theta$;
Initialize independent bias-free linear heads $g_{\text{intra}}, g_{\text{inter}}$ to the identity;
foreach *training mini-batch* **do**
  `// Contextual states and targets: one shared EMA branch`
  foreach *sequence* u **do**
    $\mathbf{H}^u \leftarrow f_\theta(X^u)$; $\widetilde{\mathbf{H}}^u \leftarrow \widetilde{f}_{\bar{\theta}}(X^u)$ without gradient tracking;
    foreach *event* $j = 1, \ldots, m_u$ **do**
      $\mathbf{z}_j^u \leftarrow \text{Pool}(\mathbf{H}^u, G_j^u)$; $\widetilde{\mathbf{z}}_j^u \leftarrow \text{Pool}(\widetilde{\mathbf{H}}^u, G_j^u)$;
    end
  end
  `// Eligible sources and anchor selection`
  $\mathcal{E}_{\text{intra}} \leftarrow \{(u,i) : |G^u_{e_i^u}| > 1\}$; $\mathcal{E}_{\text{inter}} \leftarrow \{(u,j) : 1 \le j < m_u\}$;
  Select $\mathcal{A}_r \subseteq \mathcal{E}_r$ for each relation r, optionally subsampling eligible anchors;
  `// Intra-event: leave-one-out pooling, teacher input unchanged`
  foreach *anchor* $a = (u,i) \in \mathcal{A}_{\text{intra}}$ **do**
    $j \leftarrow e_i^u$; $\mathbf{q}_a^{\text{intra}} \leftarrow g_{\text{intra}}(\mathbf{h}_i^u)$;
    $\mathbf{k}_a^{\text{intra}} \leftarrow \text{sg}\Big(\text{Pool}(\widetilde{\mathbf{H}}^u, G_j^u \setminus \{i\})\Big)$; $t_a^{\text{intra}} \leftarrow (u,j)$;
  end
  `// Inter-event: pooled source and its same-sequence successor`
  foreach *anchor* $a = (u,j) \in \mathcal{A}_{\text{inter}}$ **do**
    $\mathbf{q}_a^{\text{inter}} \leftarrow g_{\text{inter}}(\mathbf{z}_j^u)$; $\mathbf{k}_a^{\text{inter}} \leftarrow \text{sg}(\widetilde{\mathbf{z}}_{j+1}^u)$; $t_a^{\text{inter}} \leftarrow (u, j+1)$;
  end
  `// Single-positive matching and joint training`
  foreach *relation* $r \in \{\text{intra}, \text{inter}\}$ **do**
    Form each $\mathcal{B}_a^r$ from its assigned positive $\mathbf{k}_a^r$ and negative keys from event instances other than t_a^r;
    Compute dot-product InfoNCE losses ℓ_a^r (Eq. 9); average over $\mathcal{A}_r$ (Eq. 10);
  end
  Compute the sampled-softmax NTP term corresponding to Eq. 2 (Appendix D);
  $\mathcal{L}_{\text{ESP}} \leftarrow \mathcal{L}_{\text{NTP}} + \lambda_{\text{intra}}\mathcal{L}_{\text{intra}} + \lambda_{\text{inter}}\mathcal{L}_{\text{inter}}$;
  Update the parameters of $f_\theta, p_\phi, g_{\text{intra}}, g_{\text{inter}}$ by minimizing $\mathcal{L}_{\text{ESP}}$;
  Update the target encoder only by $\bar{\theta} \leftarrow \mu\bar{\theta} + (1-\mu)\theta$;
end
Discard the target encoder, event-target construction, and event-prediction heads for inference;
return *f_θ and p_ϕ*;

B SUFFICIENCY AND REPRESENTATION CONSTRAINTS OF EVENT PREDICTION

This appendix develops the predictive requirements in Section 3.2 and the source-level complementarity in Section 4.2. We first characterize predictive sufficiency at an ideal population optimum, then analyze the linear-query contrastive loss in Section 4.3 for fixed candidates. Throughout, the target encoder and event groups are fixed, and each source state is treated as a deterministic function of its available prefix.

B.1 Predictive requirements and event-target sufficiency

Following the predictive-sufficiency viewpoint of Teoh et al. (2025), we ask which target distributions must be recoverable from each state. For the full-vocabulary NTP criterion in Equation 2, an unrestricted prediction head gives the excess population risk

$$\mathsf{H}(x_{i+1} \mid \mathbf{h}_i) - \mathsf{H}(x_{i+1} \mid x_{1:i}) = I(x_{i+1}; x_{1:i} \mid \mathbf{h}_i), \tag{12}$$

where H denotes conditional entropy. This identity characterizes the population likelihood criterion; the sampled-softmax training objective is described in Appendix D. The gap can vanish while $\mathbf{h}_i$ omits prefix information relevant to an event target. For example, independent components of the prefix may determine the next token and later event content, respectively. Retaining the first component suffices for NTP. Later attention-based states can revisit the prefix, so their predictions do not require the earlier state to retain both components.

ESP assigns an enclosing-event target to an individual token state and a successor-event target to a completed event's pooled state. Using (C, S, K) for the available prefix, source state, and positive target, the two assignments are

$$(C, S, K) = \begin{cases} (x_{1:i}, \mathbf{h}_i, \widetilde{\mathbf{z}}_j^{(-i)}) & i \in G_j,\ |G_j| > 1, \\ (x_{1:b_j}, \mathbf{z}_j, \widetilde{\mathbf{z}}_{j+1}) & j < m. \end{cases} \tag{13}$$

We consider each assignment at a fixed eligible source position. The following result characterizes the information required to match the full prefix's optimal contrastive risk.

Proposition 1 (Event-target sufficiency)**.** *Consider a finite target support and a source state S that is a deterministic function of C. A candidate list contains one positive drawn from $p(K \mid C)$ and $N-1$ negatives, where $N \geq 2$. Negatives are drawn independently from a fixed full-support distribution ν, independently of the prefix and the positive. If the optimal population contrastive risk given S equals that given C, then $p(K \mid S) = p(K \mid C)$ almost surely. Optimal risks are infima over measurable pairwise scoring functions. Applied separately to the two assignments in Equation 13, this gives*

$$p(\widetilde{\mathbf{z}}_j^{(-i)} \mid \mathbf{h}_i) = p(\widetilde{\mathbf{z}}_j^{(-i)} \mid x_{1:i}), \qquad i \in G_j,\ |G_j| > 1, \tag{14}$$

$$p(\widetilde{\mathbf{z}}_{j+1} \mid \mathbf{z}_j) = p(\widetilde{\mathbf{z}}_{j+1} \mid x_{1:b_j}), \qquad j < m, \tag{15}$$

whenever the corresponding risks are equal.

The Bayes contrastive classifier uses the ratio of the target's conditional probability to its negative-sampling probability (van den Oord et al., 2018). Matching the full-prefix risk requires recovering these ratios, and hence the target conditional distribution. Section B.2 gives the proof. This population construction uses independent negatives and unrestricted scores. The event-instance candidates and linear heads in Section 4.3 need not satisfy these conditions or attain this optimum. Section B.3 instead analyzes their stated dot-product loss with the selected candidates fixed.

The two equalities concern different causal sources and prediction scopes. At an interior position $i < b_j$, the intra-event target includes later teacher states; at $i = b_j$, its content is already observed by the source. Leave-one-out pooling excludes $\widetilde{\mathbf{h}}_i$ without removing x_i from later teacher prefixes. Inter-event prediction assigns the next event's contextual target to $\mathbf{z}_j$ before any destination-event observations enter the source. Sufficiency here is relative to these history-conditioned targets, rather than isolated event labels or a latent intent.

B.2 Proof of event-target sufficiency

For the generic triple (C, S, K) in Proposition 1, let $\nu(k) > 0$ on the finite target support. Randomly place the positive key at a uniform index $J \in \{1, \ldots, N\}$, independently of C. Given (C, J), draw $K_J \sim p(K \mid C)$ and draw the other keys independently from ν. Write $\mathcal{K} = (K_1, \ldots, K_N)$. Randomizing the positive index leaves the pairwise-scoring contrastive risk unchanged.

Following van den Oord et al. (2018), Bayes' rule yields

$$p(J = l \mid C, \mathcal{K}) = \frac{p(K_l \mid C)/\nu(K_l)}{\sum_{b=1}^{N} p(K_b \mid C)/\nu(K_b)}. \tag{16}$$

The same identity holds with S in place of C. These classifiers are realizable as limits of softmax pairwise scores, using the log density ratio and sending scores with zero conditional probability to $-\infty$. Hence the optimal risks are the conditional entropies of J. Since S is a deterministic function of C,

$$\begin{aligned} L^*(S) - L^*(C) &= \mathsf{H}(J \mid S, \mathcal{K}) - \mathsf{H}(J \mid C, \mathcal{K}) \\ &= \mathbb{E}\, D_{\mathrm{KL}}(p(J \mid C, \mathcal{K}) \,\|\, p(J \mid S, \mathcal{K}))\,. \end{aligned} \tag{17}$$

Equality of the risks therefore implies equality of these candidate-index posteriors almost surely.

Fix a prefix c and its state s outside a null set. Choose a key k_0 with $p(k_0 \mid c) > 0$; then $p(k_0 \mid s) > 0$ almost surely as well. For any key k, a candidate list containing k_0 and k has positive probability under c, since k_0 can be the positive and ν has full support. Comparing the posterior probabilities of these two positions in Equation 16 gives

$$\frac{p(k \mid c)}{p(k_0 \mid c)} = \frac{p(k \mid s)}{p(k_0 \mid s)}. \tag{18}$$

This also covers $p(k \mid c) = 0$. Thus $p(k \mid c) = \alpha(c, s)p(k \mid s)$ for all k. Summing over k gives $\alpha(c, s) = 1$, proving $p(K \mid C) = p(K \mid S)$. Applying the two substitutions in Equation 13 proves Equations 14 and 15.

B.3 Source-representation constraints and complementarity

We now condition on a selected anchor, its positive assignment, and its finite candidate list $\mathcal{B}_a^r$ from Section 4.3. Following the local-curvature approach of Zhang et al. (2026), we differentiate with respect to the source representation while holding the teacher keys and prediction head fixed. For the linear query $g_r(\mathbf{s}) = A_r\mathbf{s}$ and dot-product logits $(A_r\mathbf{s})^\top \mathbf{k}/\tau$ in Equation 9, let $\pi_a^r(\mathbf{k})$ be the softmax candidate weight and $\bar{\mathbf{k}} = \sum_{\mathbf{k}} \pi_a^r(\mathbf{k})\mathbf{k}$. Direct differentiation gives

$$\nabla_{\mathbf{s}} \ell_a^r = \tau^{-1} A_r^\top (\bar{\mathbf{k}} - \mathbf{k}_a^r). \tag{19}$$

Differentiating the softmax-weighted mean yields

$$\nabla^2_{\mathbf{h}_i} \ell_i^{\mathrm{intra}} = \tau^{-2} A_{\mathrm{intra}}^\top \operatorname{Cov}_{\pi_i^{\mathrm{intra}}}(\mathbf{k}) A_{\mathrm{intra}}, \tag{20}$$

$$\nabla^2_{\mathbf{z}_j} \ell_j^{\mathrm{inter}} = \tau^{-2} A_{\mathrm{inter}}^\top \operatorname{Cov}_{\pi_j^{\mathrm{inter}}}(\mathbf{k}) A_{\mathrm{inter}}. \tag{21}$$

The covariance uses the selected candidate entries and their softmax weights. These identities apply to any fixed candidate list, including dependent keys. In contrast to Proposition 1, they require no independent-negative assumption.

For a source perturbation $\mathbf{v}$, the directional curvature is

$$\mathbf{v}^\top \nabla^2_{\mathbf{s}} \ell_a^r \mathbf{v} = \tau^{-2} \operatorname{Var}_{\pi_a^r}(\mathbf{k}^\top A_r \mathbf{v}) \geq 0. \tag{22}$$

It is positive exactly when the projected perturbation changes relative scores among the finite-logit candidates. Thus, the target differences and prediction head determine which source directions receive curvature.

Mean pooling determines how the inter-event constraint reaches individual token states. Writing $n_j = |G_j|$, the chain rule gives, for each $i \in G_j$,

$$\nabla_{\mathbf{h}_i} \ell_j^{\mathrm{inter}} = \frac{1}{n_j} \nabla_{\mathbf{z}_j} \ell_j^{\mathrm{inter}}, \qquad \Delta\mathbf{z}_j = \frac{1}{n_j} \sum_{i \in G_j} \Delta\mathbf{h}_i. \tag{23}$$

Inter-event supervision therefore backpropagates to all states in its pooled source. For an event with more than one token, however, nonzero token perturbations can cancel in the mean:

$$\sum_{i \in G_j} \Delta\mathbf{h}_i = 0 \quad \Longrightarrow \quad \Delta\mathbf{z}_j = 0 \quad \Longrightarrow \quad \Delta\mathbf{q}_j^{\mathrm{inter}} = 0. \tag{24}$$

The corresponding inter-event loss is unchanged with its keys and head fixed.

Table 4: Processed dataset statistics. Target counts are distinct retained prediction units. – denotes no corresponding probe. Training and evaluation partitions are user-disjoint.

	AppRec	MerRec	RetailRocket	KuaiRec
Prediction unit	application–action	product–action	item–action	video
Users	10,496,000	211,414	6,057	7,176
Targets	98,019	861,584	77,930	10,724
Probe classes	6	17	–	3
Interactions	3,631,902,735	31,542,538	355,992	8,021,397
Training sequences	29,877,509	248,674	4,924	45,327
Evaluation sequences	746,775	106,400	2,081	19,935
Mean sequence length	118.60	88.83	50.82	122.91
Mean 1h events per sequence	27.14	15.34	12.47	20.78
Mean evaluation 1h event length	4.37	5.77	4.22	6.04

Intra-event prediction evaluates each selected token source separately. A mean-preserving perturbation can therefore have positive intra-event curvature when, for at least one selected anchor i,

$$\mathrm{Var}_{\pi_i^{\mathrm{intra}}}\left(\mathbf{k}^\top A_{\mathrm{intra}} \Delta \mathbf{h}_i\right) > 0. \tag{25}$$

This is a conditional source-granularity distinction: intra-event supervision can constrain directions invisible to the pooled query when its head and candidate differences detect them. Conversely, its enclosing-event target does not explicitly require prediction of the successor event. Inter-event supervision supplies that prediction scope at the completed event's aggregate.

For fixed nonempty anchor sets, Equation 10 averages these per-anchor losses. The calculations above concern source representations, with the selected candidates, teacher keys, and heads fixed. NTP, shared encoder parameters, other anchors, and subsequent EMA updates also shape joint training. Global backbone-state geometry and information accessible to linear probes remain empirical outcomes evaluated in Sections 5.5 and 5.4.

C Dataset Details and Statistics

Table 4 summarizes the four datasets after the preprocessing used in our experiments. The public-dataset counts describe our materialized subsets rather than the complete original releases; they reflect the filtering, windowing, and user-level split described below.

AppRec. AppRec is constructed from an anonymized 14-day snapshot of industrial mobile application interactions. Each prediction target is the joint identity of an application package and its associated action; temporal context is recorded as hour of week. We retain at most the 512 most recent interactions per user and form tail-aligned, non-overlapping sequences of at most 128 interaction tokens, discarding residual sequences shorter than 20. The resulting materialization contains 10,240,000 training users and 256,000 evaluation users, corresponding to 29,877,509 and 746,775 sequences. The two partitions contain 3,543,344,129 and 88,558,606 interactions, respectively. User and application identifiers are anonymized, and the two partitions are disjoint by user. For category probing, AppRec provides six non-exclusive application-category labels; an interaction may carry multiple labels or none of them.

MerRec. MerRec was collected from the Mercari consumer-to-consumer marketplace over six months in 2023 (Li et al., 2024). The complete release contains 5,569,367 users, 1,274,814,848 action events, 83,078,407 item identifiers, and 1,403,098 product identifiers. Its six buyer-side event types span item views, likes, cart additions, offers, and the initiation and completion of purchases. For tractable experiments, we use 60 consecutive Parquet shards from the May 2023 partition, totaling 4.45 GB and 33,653,934 source rows. After sorting interactions by timestamp, we map each product–action pair to a deterministic dense identifier, construct non-overlapping windows of at most 128 interaction tokens, and retain windows with at least 21 tokens. This yields 211,414 users, 355,074 sequences, and 31,542,538 interactions. A deterministic user-hash split assigns 147,989 users (248,674 sequences) to training and 63,425 users (106,400 sequences) to evaluation.

RetailRocket. RetailRocket is a public collection of hashed behavioral logs from a real-world e-commerce website (Zykov, 2017). The original event file spans 4.5 months and contains 2,756,101 interactions from 1,407,580 visitors: views, cart additions, and transactions. We encode each item–action pair as one target token, sort each visitor's interactions chronologically, and apply the same non-overlapping 128-token windowing policy, retaining windows with at least 21 tokens. The processed data contain 6,057 visitors, 7,005 sequences, 355,992 interactions, and 77,930 distinct targets. Views, cart additions, and transactions account for 90.94%, 6.24%, and 2.82% of retained interactions, respectively. The user-hash split produces 4,924 training sequences from 4,239 visitors and 2,081 evaluation sequences from 1,818 visitors.

KuaiRec. KuaiRec comprises a near-fully observed Small Matrix and a larger Big Matrix of short-video feedback (Gao et al., 2022). The Small Matrix contains 1,411 users, 3,327 videos, and 4,676,570 interactions at 99.6% density, whereas the Big Matrix contains 7,176 users, 10,728 videos, and 12,530,806 interactions at 16.3% density. We use the Big Matrix. We retain interactions with `watch_ratio` ≥ 0.5, which leaves 8,031,791 qualified interactions, and then chronologically construct non-overlapping windows of at most 128 tokens. After removing short residuals, the materialization contains 8,021,397 interactions, 65,262 sequences, and 10,724 distinct video targets. The deterministic user-level split assigns 5,023 users (45,327 sequences) to training and 2,153 users (19,935 sequences) to evaluation.

D Baseline and Implementation Details

D.1 Shared Architecture, Training, and Evaluation

All compared methods are instantiated on the same causal HSTU encoder (Zhai et al., 2024). The encoder has hidden size 256, four attention heads, four sequence layers, and a maximum sequence length of 128. The methods share the same input and target embedding parameterization, positional and temporal features, preprocessed sequences, sampled-softmax interface, and full-catalog evaluator. The NTP baseline uses this encoder without auxiliary objectives. Training-only prediction heads, projectors, and teacher models are removed for evaluation. Geometry diagnostics and linear probes use the frozen 256-dimensional encoder states, while recommendation uses the trained causal encoder and NTP head.

The common NTP loss uses 128 sampled negatives during training. At evaluation, the causal state is scored against every target in the materialized vocabulary; none of the reported metrics uses sampled candidates. The main experiments use a per-GPU batch size of 64 on eight Tesla V100-SXM2-32GB GPUs, for a global batch size of 512. Model training runs use FP32 and seed 1234. We optimize with Adam, $(\beta_1, \beta_2) = (0.9, 0.98)$, $\epsilon = 10^{-8}$, and cosine decay after 5% linear warmup to a minimum learning-rate ratio of 0.1. For MerRec, RetailRocket, and KuaiRec, the learning rate is 10^{-3} and weight decay is 0.01. AppRec uses learning rate 10^{-5} and weight decay 0.02. These optimization settings are shared across methods within each dataset. Model training lasts 10 epochs, and we report the final checkpoint.

D.2 Statistical Significance Tests

Following Xin et al. (2026), we compare ESP's fixed-checkpoint recommendation results with NTP and the best baseline for each dataset and metric using two-sided user-paired Monte Carlo randomization tests (Smucker et al., 2007). For representation metrics, we use the same test against the best baseline and a paired user-cluster bootstrap against NTP. We mark improvements as significant only when their Holm-adjusted p-values are below 0.05.

D.3 Baseline Objectives

NTP. This baseline uses dense next-token prediction at every valid transition (Zhai et al., 2024).

Multi-Token Prediction (MTP). MTP trains the shared state to predict several future tokens with independent output heads (Gloeckle et al., 2024). We retain the ordinary NTP loss and add the equally

weighted mean of three auxiliary prediction losses at offsets 2, 3, and 4. Each auxiliary loss uses only positions whose target remains inside the sequence.

NextLat. NextLat augments NTP with a learned latent-dynamics objective (Teoh et al., 2025). A dynamics module predicts the subsequent hidden state from the current state and the observed next-token embedding. We use a residual three-layer GELU MLP of hidden width 512 acting on the layer-normalized concatenation of these two inputs. The next state is a stopped-gradient Smooth-L_1 target; a second loss matches the target and predicted token distributions through a detached output head. Both auxiliary coefficients are 1.0. The original method defines KL over the output vocabulary; our HSTU adaptation computes it on the deduplicated candidate support available to the sampled retrieval head, with temperature 0.05 and at most 512 latent pairs per update.

PinnerFormer (Pinner). Pinner's auxiliary objective trains selected causal states to retrieve positive items from a future engagement window (Pancha et al., 2022). We use a 28-day maximum future target window within the observed sequence, select at most 32 positions per sequence, and apply the sequence-balanced dense-all-action loss. The retrieval head uses an MLP projection with hidden-width multiplier 4, L_2 normalization, a learned temperature lower-bounded by 0.01, in-batch negatives gathered across devices, and 5,000 random negatives. We mask positives from the same sequence in the negative pool and use the batch log-Q correction.

SessionRec (SRec). SRec replaces an exclusively item-wise future target with next-session prediction (Huang et al., 2025). Consecutive interactions are segmented by a 30-minute inactivity threshold. A session-level state predicts the set of positive items in the following session through a multi-positive retrieval objective. In the shared-backbone adaptation, the session predictor is linear, the retrieval temperature is 0.1, and each update retains at most 256 session pairs, 32 positive items per session, and 1,024 random negatives.

EchoRec. EchoRec introduces multi-item prediction branches at different horizons and aligns their trajectory with a holistic representation of future preference (Ma et al., 2026). The training objective combines horizon-specific prediction with trajectory and cycle-consistency terms. We use the original three-horizon structure: the main branch predicts the next item and two chained auxiliary branches predict the second and third future items, each with unit prediction weight. The trace and cycle coefficients are 0.5 and 0.3, respectively; the holistic projection width is 128, the projector hidden width is 256, and auxiliary branch FFNs use a width multiplier of 4.

Oracle4Rec (ORec). ORec transfers information from future behavior into the causal user encoder (Xia et al., 2024). Our shared-HSTU implementation first trains the future encoder, then initializes the causal past encoder and trains it with oracle guidance. We retain 10 future interactions, attenuation coefficient 0.05, frequency quantile 0.75, and one frequency-filter layer. The future objective has unit weight and the guidance loss has weight 0.01 in the HSTU adaptation.

UFRec. UFRec uses predictive uncertainty to weight multi-step future targets and adds a future-aware contrastive objective (Cui et al., 2026). The uncertainty term modulates future supervision, while the contrastive term aligns the current representation with its own aggregated future trajectory against other users'. We use maximum future step $K = 2$, entropy-weighting temperature 4.0, and contrastive coefficient 0.1. The future-trajectory contrastive temperature is 1.0; at most 1,024 valid positions contribute per update. The entropy weight is detached, as in the original objective.

NONTP. NONTP supplements NTP with temporal and cross-domain representation objectives (Li et al., 2026). Temporal contrastive learning (TCL) aligns a causal state with a future trajectory represented by a momentum target. We retain the source paper's $K = 3$ offsets, independent linear predictor per offset, EMA momentum 0.999, cosine InfoNCE temperature 0.07, and TCL and TDL coefficients of 0.1 each. TCL uses at most 2,048 valid positions per update and excludes other positions from the same sequence from its negative pool. Trans-domain learning (TDL) pools strictly preceding states whose domain differs from the next target's domain, then predicts that target through the shared NTP head. TDL uses action types as domains on MerRec, RetailRocket, and AppRec. No TDL domain mapping was configured for KuaiRec, so its NONTP adaptation uses NTP+TCL with $\lambda_{\mathrm{TDL}} = 0$.

D.4 ESP SETTINGS

ESP uses EMA momentum 0.999, FP32 event-state pooling, contrastive temperature 0.1, and $\lambda_{\text{intra}} = \lambda_{\text{inter}} = 0.1$. Each loss retains at most 512 eligible anchor–key pairs per GPU rank, approximately evenly spaced in flattened pair order. InfoNCE uses unnormalized prediction-head outputs and event targets, with only the retained local keys and no cross-GPU negatives. After each optimizer step, EMA updates only the sequence backbone; the teacher uses the current shared input embeddings without gradients.

D.5 FULL-CATALOG RETRIEVAL PROTOCOL

All methods rank the complete materialized target vocabulary at every valid held-out next-token transition. Candidate scores use the causal state and target embeddings through the trained NTP retrieval path. Training may use sampled softmax, but evaluation does not sample negative candidates. Within each dataset, the held-out sequences, prediction units, and candidate vocabulary are shared across methods.

HR@K is the fraction of evaluated transitions whose ground-truth target appears among the top K candidates. NDCG@K measures the logarithmically discounted gain of that target within the same cutoff, assigning zero gain when it falls outside the top K. Each transition has one ground-truth target. MRR averages the reciprocal rank of the ground-truth target over evaluated transitions. Comparisons are made within datasets because catalog sizes and prediction units differ. The fixed-query evaluation in Section 5.3 uses the same ranking definitions but evaluates one query per sequence and changes only the future target offset.

D.5.1 MERREC EVENT-CONTENT RETRIEVAL

The event-retrieval analysis in Section F.2 uses one query per eligible MerRec sequence. We divide each sequence into consecutive natural-hour events and retain sequences with at least two events. The last event is withheld from the query forward pass, and the raw token state $\mathbf{h}_{b_{m-1}}$ at the immediately preceding position is used as the query. To construct the candidate side, each target event is encoded independently by the same frozen backbone. Candidate timestamps are shifted so that the first token occurs at time zero, and the resulting token states are mean-pooled. This yields 105,378 queries.

Candidate identities are defined by ordered target-event token tuples. After deduplication, the pool contains 87,540 identities; representations from repeated occurrences of the same identity are averaged and normalized to form one prototype. Queries and prototypes are ℓ_2-normalized and ranked by cosine similarity. We report NDCG@10 using the rank of the exact ordered target event, breaking score ties by candidate index. The evaluation trains neither a retrieval head nor the backbone. This readout compares a token query with independently encoded event content. During ESP training, inter-event prediction instead uses the pooled source $\mathbf{z}_{m-1}$, a prediction head, and a history-conditioned EMA target.

E SUPPLEMENTARY EXPERIMENTS

E.1 FROZEN-STATE EVALUATION

E.1.1 FEATURE EXTRACTION

We evaluate the 256-dimensional outputs of the causal HSTU encoder before the NTP, intra-event, and inter-event heads. All parameters are frozen during feature extraction for the geometry diagnostics; these diagnostics fit no probe. For each held-out sequence, we retain the final valid causal state. The EMA target branch is not used. For each method, we extract states from the trained backbone used for recommendation.

We use at most 16,384 states per method, sampled without replacement with seed 1234. This limit is reached on MerRec, KuaiRec, and AppRec. RetailRocket has 2,081 held-out sequences, all of which are used. Every evaluated state has finite entries and nonzero norm. The same extraction positions and sample count are used for all methods within a dataset.

Table 5: Frozen representation quality on RetailRocket, KuaiRec, and AppRec.

Dataset	Metric	NTP	MTP	NextLat	NONTP	Pinner	SRec	EchoRec	ORec	UFRec	ESP
RetailRocket	**RM/d** ↑	0.8486	0.8669	0.8640	0.8171	0.6227	0.8388	0.8131	0.5706	0.7009	**0.9257**‡
	Iso. ↑	0.4182	0.5010	0.4573	0.4031	0.0445	0.3389	0.3670	0.0647	0.1261	**0.5868**‡
	Unif. ↓	-3.9252	-3.9214	-3.9231	-3.9145	-2.8053	-3.8972	-3.9150	-2.2202	-3.7478	**-3.9449**‡
	Aniso. ↓	0.0018	0.0046	0.0042	0.0033	0.0834	0.0030	0.0018	0.3309	0.0048	**0.0013**‡
KuaiRec	**RM/d** ↑	0.7009	0.7254	0.6563	0.7339	0.6232	0.6297	0.5864	0.4190	0.7396	**0.7491**‡
	Iso. ↑	0.0577	0.0730	0.0573	0.0692	0.0793	0.0229	0.0432	0.0503	0.0764	**0.1927**‡
	Unif. ↓	-2.8947	-2.9530	-2.9105	-3.0602	-3.2174	-2.1925	-2.6257	-2.5162	-3.1839	**-3.6112**‡
	Aniso. ↓	0.1325	0.1355	0.1202	0.1060	0.0765	0.2604	0.1702	0.2245	0.0873	**0.0437**‡
AppRec	**RM/d** ↑	0.6617	0.6532	0.6842	0.6126	0.4794	0.6357	0.5265	0.1372	0.6813	**0.7499**‡
	Iso. ↑	0.1746	0.1777	0.1889	0.1729	0.1358	0.1154	0.1166	0.0174	0.1557	**0.2692**‡
	Unif. ↓	-2.6428	-2.7580	-2.7691	-2.9573	-1.2730	-2.0303	-2.4272	-0.4170	-2.9046	**-3.3783**‡
	Aniso. ↓	0.2895	0.2662	0.2665	0.2189	0.6642	0.4463	0.3338	0.8602	0.2249	**0.1294**‡

E.1.2 Global Geometry Diagnostics

RankMe measures the entropy-based effective rank of the uncentered state matrix (Garrido et al., 2023). We report RankMe/d, normalized by the 256-dimensional feature width; higher values indicate broader spectral utilization. IsoScore measures the balance of variance across dimensions using the centered sample covariance (Rudman et al., 2022). Its values are clipped to the unit interval for numerical precision, with higher values indicating a more even covariance spectrum. Both metrics are computed in FP64.

For directional diagnostics, we normalize each state using a norm floor of 10^{-12}. Uniformity measures the dispersion of normalized states on the unit hypersphere (Wang & Isola, 2020); lower values indicate broader dispersion. We sample distinct state pairs with seed 1234, using the smaller of 250,000 pairs and eight times the number of states. Following the cosine-based anisotropy diagnostic of Ethayarajh (2019), we measure the squared magnitude of the mean normalized state vector. This equals the mean pairwise cosine similarity when self-pairs are included; lower values indicate weaker concentration around a shared direction.

E.1.3 Additional Geometry Results

Table 5 reports the geometry diagnostics on RetailRocket, KuaiRec, and AppRec.

ESP has the highest RankMe/d and the lowest uniformity and mean-direction anisotropy on all three datasets. It also leads IsoScore on each, with a particularly large IsoScore advantage on KuaiRec relative to its effective-rank advantage. On RetailRocket, several baselines already have near-zero anisotropy; the clearer difference is spectral.

E.1.4 Future-Category Linear Probes

The tasks in Figure 5 use the encoder output at the final observed position of a truncated prefix. The backbone and embeddings remain frozen, and method-specific training heads are not used; the only learned component is a linear map from the 256-dimensional state to the 17 MerRec catalog categories. Category labels are constructed from a separate product catalog and are absent from the encoder input.

For the fixed-horizon protocol, a length-n sequence is truncated immediately after its ninth token from the end. The resulting state is shared across the offset-2, offset-4, offset-8, and next-eight dominant-category tasks. The train and test sets contain 248,674 and 106,400 sequences, respectively. The offset-1 task is omitted from the figure because it is the immediate-target view; the experiment focuses on semantic information beyond that position.

For the event-position protocol, natural-hour events are formed from calendar-aligned one-hour windows. We retain sequences with at least two events and at least two tokens in the penultimate event. A deterministic hash of the sequence ID with seed 1234 selects the Interior query from all but the last position of the penultimate event; the End query uses its last position. Both views encode only their observed prefixes and predict either the first or the dominant category of the same final

Table 6: MerRec frozen-representation linear-probe results for the eight category-prediction tasks.

Task	Metric	NTP	MTP	NextLat	NONTP	Pinner	SRec	EchoRec	ORec	UFRec	ESP
(a)	**Macro-F1**	0.5293 (±0.0002)	0.5313 (±0.0007)	0.5363 (±0.0000)	0.5351 (±0.0002)	0.5109 (±0.0002)	0.5261 (±0.0012)	0.5307 (±0.0013)	0.5337 (±0.0004)	0.5308 (±0.0008)	**0.5392** (±0.0002)
	Macro-AP	0.4849 (±0.0003)	0.4987 (±0.0006)	0.5051 (±0.0003)	0.4954 (±0.0001)	0.4799 (±0.0004)	0.4881 (±0.0046)	0.4900 (±0.0041)	0.5031 (±0.0003)	0.4909 (±0.0011)	**0.5100** (±0.0007)
(b)	**Macro-F1**	0.4949 (±0.0018)	0.4984 (±0.0001)	0.4987 (±0.0007)	0.4951 (±0.0003)	0.4886 (±0.0002)	0.4935 (±0.0010)	0.4942 (±0.0003)	0.4914 (±0.0066)	0.4973 (±0.0004)	**0.5074** (±0.0011)
	Macro-AP	0.4437 (±0.0039)	0.4606 (±0.0011)	0.4628 (±0.0004)	0.4426 (±0.0013)	0.4536 (±0.0000)	0.4463 (±0.0009)	0.4462 (±0.0001)	0.4536 (±0.0052)	0.4497 (±0.0003)	**0.4683** (±0.0017)
(c)	**Macro-F1**	0.4517 (±0.0001)	0.4537 (±0.0005)	0.4578 (±0.0002)	0.4417 (±0.0003)	0.4610 (±0.0001)	0.4527 (±0.0005)	0.4504 (±0.0002)	0.4564 (±0.0005)	0.4487 (±0.0010)	**0.4705** (±0.0004)
	Macro-AP	0.3925 (±0.0002)	0.4045 (±0.0005)	0.4108 (±0.0006)	0.3785 (±0.0002)	**0.4219** (±0.0002)	0.4006 (±0.0003)	0.3911 (±0.0003)	0.4076 (±0.0003)	0.3947 (±0.0028)	0.4194 (±0.0003)
(d)	**Macro-F1**	0.5826 (±0.0002)	0.5798 (±0.0009)	0.5766 (±0.0093)	0.5821 (±0.0001)	0.5757 (±0.0002)	0.5771 (±0.0040)	0.5784 (±0.0004)	0.5838 (±0.0003)	0.5776 (±0.0000)	**0.5938** (±0.0007)
	Macro-AP	0.5607 (±0.0007)	0.5677 (±0.0023)	0.5649 (±0.0135)	0.5573 (±0.0004)	0.5681 (±0.0026)	0.5595 (±0.0070)	0.5561 (±0.0044)	0.5772 (±0.0000)	0.5629 (±0.0008)	**0.5869** (±0.0015)
(e)	**Macro-F1**	0.4386 (±0.0003)	0.4433 (±0.0009)	0.4453 (±0.0001)	0.4375 (±0.0002)	0.4522 (±0.0003)	0.4443 (±0.0002)	0.4387 (±0.0003)	0.4493 (±0.0002)	0.4403 (±0.0008)	**0.4589** (±0.0005)
	Macro-AP	0.3829 (±0.0006)	0.3951 (±0.0024)	0.3934 (±0.0005)	0.3770 (±0.0002)	0.4109 (±0.0003)	0.3912 (±0.0001)	0.3823 (±0.0001)	0.4010 (±0.0006)	0.3841 (±0.0006)	**0.4115** (±0.0003)
(f)	**Macro-F1**	0.4502 (±0.0007)	0.4532 (±0.0003)	0.4561 (±0.0010)	0.4525 (±0.0003)	0.4614 (±0.0005)	0.4528 (±0.0003)	0.4502 (±0.0002)	0.4589 (±0.0002)	0.4488 (±0.0007)	**0.4727** (±0.0007)
	Macro-AP	0.3934 (±0.0005)	0.4063 (±0.0001)	0.4085 (±0.0020)	0.3931 (±0.0001)	0.4174 (±0.0000)	0.3996 (±0.0001)	0.3906 (±0.0001)	0.4129 (±0.0002)	0.3965 (±0.0004)	**0.4221** (±0.0004)
(g)	**Macro-F1**	0.4677 (±0.0006)	0.4722 (±0.0013)	0.4743 (±0.0018)	0.4656 (±0.0001)	0.4868 (±0.0005)	0.4721 (±0.0003)	0.4682 (±0.0005)	0.4745 (±0.0019)	0.4669 (±0.0010)	**0.4882** (±0.0003)
	Macro-AP	0.4156 (±0.0017)	0.4297 (±0.0038)	0.4334 (±0.0047)	0.4049 (±0.0003)	**0.4538** (±0.0001)	0.4254 (±0.0002)	0.4188 (±0.0005)	0.4363 (±0.0011)	0.4162 (±0.0005)	0.4509 (±0.0001)
(h)	**Macro-F1**	0.4797 (±0.0003)	0.4817 (±0.0004)	0.4839 (±0.0003)	0.4783 (±0.0006)	0.4975 (±0.0016)	0.4826 (±0.0009)	0.4814 (±0.0006)	0.4847 (±0.0027)	0.4775 (±0.0011)	**0.5054** (±0.0004)
	Macro-AP	0.4285 (±0.0002)	0.4442 (±0.0002)	0.4436 (±0.0002)	0.4189 (±0.0024)	0.4601 (±0.0014)	0.4334 (±0.0014)	0.4227 (±0.0006)	0.4478 (±0.0022)	0.4275 (±0.0010)	**0.4675** (±0.0003)

event. This protocol yields 178,735 training and 76,834 test sequences. Across both protocols, a dominant-category target is the most frequent category in the corresponding future window or event, with ties resolved by the latest occurrence and then the smaller category id.

Each probe is trained for 30 epochs with inverse-square-root-frequency weighted cross-entropy and AdamW, using learning rate 10^{-3}, weight decay 10^{-4}, and batch size 4096. We evaluate the final checkpoint. Reported values are the mean $\pm$ standard deviation over head seeds 1234, 1235, and 1236. These error bars quantify probe-head variation; all frozen encoders use training seed 1234.

The complete readout tables report Macro-F1 and Macro-AP, with equal weight for each category. Macro-F1 evaluates discrete predictions, while Macro-AP averages per-category average precision computed from continuous prediction scores.

E.1.5 Complete MerRec Readout Results

The same frozen ESP encoder leads mean Macro-F1 on every MerRec readout task (Table 6). Pinner leads Macro-AP at the longest token offset and for the Interior dominant-category target, so the discrete-prediction advantage does not hold for every score-ranking measure.

E.1.6 KuaiRec Future-Category Linear Probes

We repeat the two linear-probe protocols on KuaiRec. Here, a category denotes one of three watch-depth intervals, defined by the ratio of watch time to video duration: $[0.5, 1)$, $[1, 2)$, and $[2, +\infty)$. Thus, the probes measure whether a frozen history state linearly exposes the intensity of future engagement rather than video content metadata. The target positions and their labels are excluded from the encoder input in every task.

For fixed-horizon prediction, we again use the ninth token from the end as the query and predict the watch-depth category at offsets 2, 4, and 8, or the dominant category among the next eight interactions. The four tasks share 45,327 training sequences and 19,935 test sequences. For event-position prediction, interactions are grouped into consecutive natural-hour events. We retain sequences with at least two events and at least two interactions in the penultimate event. Following the deterministic query rule defined above, Interior selects a non-final state in the penultimate event and End selects its final state; each state is used to predict either the first or dominant watch-depth category in the final event. These tasks contain 38,906 training and 17,285 test sequences. Dominant categories use the same frequency and tie-breaking rule as in the MerRec evaluation.

The encoder and embeddings remain frozen, and the only learned component is a linear map from the 256-dimensional history state to three logits. We use the same probe optimization, final-checkpoint protocol, and probe-head seeds as above. Macro-F1 and Macro-AP are computed across the three watch-depth categories.

ESP leads Macro-F1 across the tested KuaiRec token offsets and for the future-window summary (Figure 6; Table 7). Its scores vary little across offsets, unlike the clearer distance-related decline on MerRec.

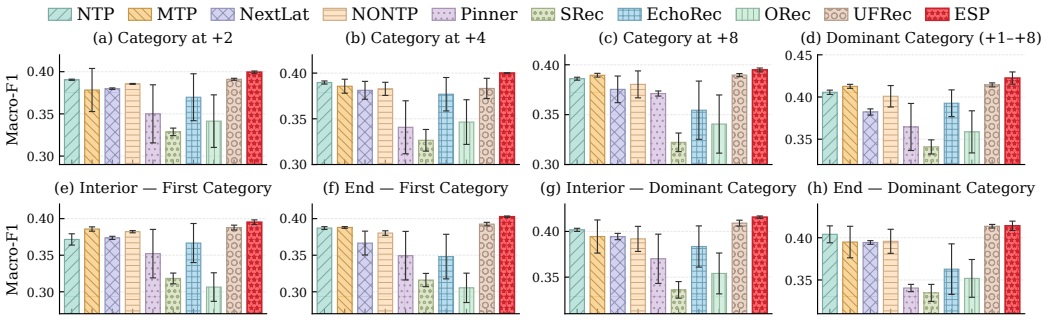


Figure 6: Frozen linear-probe Macro-F1 on KuaiRec watch-depth targets. Top: token offsets and a future window. Bottom: final-event targets from Interior and End queries. Error bars show ±1 standard deviation across three probe-head seeds.

Table 7: KuaiRec frozen-representation linear-probe results for the eight watch-depth prediction tasks.

Task	Metric	NTP	MTP	NextLat	NONTP	Pinner	SRec	EchoRec	ORec	UFRec	ESP
(a)	**Macro-F1**	0.3904 (±0.0007)	0.3783 (±0.0255)	0.3799 (±0.0009)	0.3855 (±0.0005)	0.3500 (±0.0344)	0.3287 (±0.0045)	0.3697 (±0.0278)	0.3414 (±0.0312)	0.3910 (±0.0012)	**0.3995** (±0.0014)
	Macro-AP	0.4020 (±0.0005)	0.3984 (±0.0207)	0.3998 (±0.0007)	0.4049 (±0.0009)	0.3756 (±0.0184)	0.3681 (±0.0095)	0.3907 (±0.0162)	0.3723 (±0.0208)	0.3993 (±0.0004)	**0.4076** (±0.0006)
(b)	**Macro-F1**	0.3897 (±0.0017)	0.3856 (±0.0077)	0.3812 (±0.0098)	0.3828 (±0.0072)	0.3407 (±0.0290)	0.3265 (±0.0119)	0.3769 (±0.0183)	0.3464 (±0.0244)	0.3832 (±0.0111)	**0.4001** (±0.0002)
	Macro-AP	0.4040 (±0.0011)	0.4033 (±0.0139)	0.4021 (±0.0071)	0.4046 (±0.0087)	0.3716 (±0.0153)	0.3681 (±0.0022)	0.3956 (±0.0105)	0.3758 (±0.0199)	0.3959 (±0.0123)	**0.4095** (±0.0006)
(c)	**Macro-F1**	0.3860 (±0.0015)	0.3895 (±0.0019)	0.3754 (±0.0133)	0.3804 (±0.0135)	0.3712 (±0.0026)	0.3222 (±0.0093)	0.3545 (±0.0292)	0.3406 (±0.0292)	0.3897 (±0.0015)	**0.3948** (±0.0020)
	Macro-AP	0.4020 (±0.0006)	**0.4076** (±0.0004)	0.3954 (±0.0096)	0.3954 (±0.0130)	0.3850 (±0.0007)	0.3606 (±0.0017)	0.3763 (±0.0204)	0.3719 (±0.0197)	0.4000 (±0.0008)	0.4050 (±0.0011)
(d)	**Macro-F1**	0.4056 (±0.0025)	0.4126 (±0.0024)	0.3823 (±0.0037)	0.4009 (±0.0127)	0.3645 (±0.0279)	0.3409 (±0.0084)	0.3926 (±0.0158)	0.3587 (±0.0250)	0.4144 (±0.0024)	**0.4224** (±0.0073)
	Macro-AP	0.4249 (±0.0031)	**0.4386** (±0.0007)	0.4081 (±0.0138)	0.4249 (±0.0133)	0.3919 (±0.0203)	0.3676 (±0.0029)	0.4194 (±0.0068)	0.3813 (±0.0304)	0.4263 (±0.0014)	0.4317 (±0.0057)
(e)	**Macro-F1**	0.3716 (±0.0077)	0.3858 (±0.0030)	0.3737 (±0.0021)	0.3822 (±0.0015)	0.3521 (±0.0332)	0.3183 (±0.0073)	0.3667 (±0.0267)	0.3064 (±0.0195)	0.3878 (±0.0033)	**0.3951** (±0.0031)
	Macro-AP	0.3918 (±0.0024)	**0.4050** (±0.0009)	0.3939 (±0.0008)	0.3987 (±0.0011)	0.3752 (±0.0140)	0.3590 (±0.0023)	0.3785 (±0.0245)	0.3511 (±0.0062)	0.3936 (±0.0009)	0.3999 (±0.0019)
(f)	**Macro-F1**	0.3871 (±0.0019)	0.3879 (±0.0012)	0.3666 (±0.0163)	0.3803 (±0.0031)	0.3494 (±0.0332)	0.3161 (±0.0091)	0.3481 (±0.0305)	0.3054 (±0.0201)	0.3925 (±0.0024)	**0.4026** (±0.0010)
	Macro-AP	0.3997 (±0.0006)	**0.4075** (±0.0005)	0.3901 (±0.0116)	0.4012 (±0.0011)	0.3749 (±0.0150)	0.3583 (±0.0008)	0.3724 (±0.0209)	0.3509 (±0.0067)	0.3984 (±0.0003)	0.4054 (±0.0011)
(g)	**Macro-F1**	0.4017 (±0.0017)	0.3944 (±0.0181)	0.3944 (±0.0034)	0.3918 (±0.0137)	0.3701 (±0.0270)	0.3361 (±0.0091)	0.3835 (±0.0225)	0.3541 (±0.0223)	0.4091 (±0.0032)	**0.4157** (±0.0013)
	Macro-AP	0.4232 (±0.0015)	0.4231 (±0.0191)	0.4233 (±0.0008)	0.4175 (±0.0159)	0.3973 (±0.0148)	0.3714 (±0.0024)	0.4115 (±0.0114)	0.3855 (±0.0252)	0.4197 (±0.0005)	**0.4272** (±0.0014)
(h)	**Macro-F1**	0.4041 (±0.0100)	0.3949 (±0.0187)	0.3944 (±0.0022)	0.3958 (±0.0143)	0.3404 (±0.0043)	0.3347 (±0.0101)	0.3628 (±0.0300)	0.3519 (±0.0224)	0.4137 (±0.0021)	**0.4142** (±0.0055)
	Macro-AP	0.4195 (±0.0131)	0.4284 (±0.0126)	0.4243 (±0.0010)	0.4238 (±0.0112)	0.3819 (±0.0063)	0.3731 (±0.0037)	0.3982 (±0.0189)	0.3866 (±0.0255)	0.4236 (±0.0008)	**0.4298** (±0.0033)

ESP's first-category readout improves from Interior to End, whereas its dominant-category readout declines slightly. The End dominant-category result is nearly tied with UFRec, so later context does not improve every KuaiRec target.

ESP leads mean Macro-F1 in every KuaiRec task, but MTP leads Macro-AP at the longest offset, for the future-window summary, and for both event first-category targets. Better discrete predictions therefore do not consistently imply better category-score ranking.

E.1.7 AppRec Future-Category Linear Probes

We apply the two protocols to the six non-exclusive application-category labels in AppRec. Because an interaction may carry multiple category labels or none of the six, each target is represented as a six-dimensional binary vector. The target labels are stored separately and are excluded from the encoder input. For fixed-horizon prediction, the query is again the ninth token from the end, and the four tasks predict the category-label vector at offsets 2, 4, and 8, or the dominant category over the next eight interactions. For the latter task, the dominant category is the most frequently occurring label, with ties broken by its earliest occurrence and then by label index. The four tasks share 9,768 test sequences.

For event-position prediction, interactions are grouped into natural-hour events. We predict either the category-label vector attached to the first interaction of the final event or its dominant category from a state in the penultimate event. Interior uses a deterministically selected non-final position, whereas End uses the final position of that event. The Interior and End protocols contain 6,731 and 9,941 test sequences, respectively; the difference arises because Interior additionally requires a non-final position in the penultimate event. Every view uses 40,000 probe-training and 10,000 validation sequences, with disjoint users across training, validation, and test.

The encoder and embeddings are frozen, and a linear map from the 256-dimensional history state to six logits is trained with binary cross-entropy. We use the same optimizer settings, batch size, and

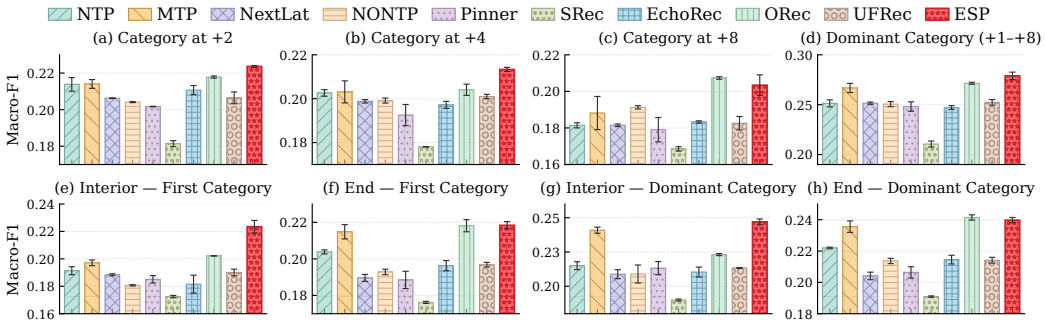


Figure 7: Frozen linear-probe Macro-F1 on AppRec future-category targets. Top: token offsets and a future window. Bottom: final-event targets from Interior and End queries. Error bars show ± 1 standard deviation across three probe-head seeds. Interior and End use different test cohorts.

Table 8: AppRec frozen-representation linear-probe results for the eight category-prediction tasks. Macro-AP is threshold-free, whereas Macro-F1 uses calibrated thresholds.

Task	Metric	NTP	MTP	NextLat	NONTP	Pinner	SRec	EchoRec	ORec	UFRec	ESP
(a)	**Macro-F1**	0.2138 (±0.0037)	0.2141 (±0.0024)	0.2063 (±0.0002)	0.2041 (±0.0003)	0.2017 (±0.0001)	0.1814 (±0.0016)	0.2107 (±0.0025)	0.2178 (±0.0006)	0.2065 (±0.0032)	**0.2236** (±0.0005)
	Macro-AP	0.1936 (±0.0011)	0.1886 (±0.0008)	0.1877 (±0.0009)	0.1909 (±0.0018)	0.1790 (±0.0005)	0.1699 (±0.0004)	0.1881 (±0.0017)	0.1884 (±0.0006)	0.1860 (±0.0009)	**0.1987** (±0.0009)
(b)	**Macro-F1**	0.2026 (±0.0015)	0.2031 (±0.0050)	0.1988 (±0.0008)	0.1991 (±0.0011)	0.1925 (±0.0048)	0.1780 (±0.0001)	0.1972 (±0.0016)	0.2041 (±0.0026)	0.2010 (±0.0010)	**0.2133** (±0.0009)
	Macro-AP	0.1803 (±0.0008)	0.1797 (±0.0007)	0.1781 (±0.0008)	0.1819 (±0.0013)	0.1734 (±0.0006)	0.1610 (±0.0003)	0.1757 (±0.0014)	0.1729 (±0.0005)	0.1786 (±0.0007)	**0.1852** (±0.0002)
(c)	**Macro-F1**	0.1814 (±0.0014)	0.1882 (±0.0091)	0.1815 (±0.0007)	0.1913 (±0.0009)	0.1790 (±0.0067)	0.1685 (±0.0012)	0.1833 (±0.0006)	**0.2073** (±0.0007)	0.1826 (±0.0037)	0.2034 (±0.0056)
	Macro-AP	0.1636 (±0.0008)	0.1642 (±0.0005)	0.1627 (±0.0006)	0.1640 (±0.0005)	0.1620 (±0.0004)	0.1508 (±0.0001)	0.1598 (±0.0007)	0.1699 (±0.0015)	0.1626 (±0.0005)	**0.1754** (±0.0007)
(d)	**Macro-F1**	0.2513 (±0.0035)	0.2668 (±0.0046)	0.2515 (±0.0011)	0.2506 (±0.0025)	0.2481 (±0.0047)	0.2105 (±0.0030)	0.2471 (±0.0019)	0.2714 (±0.0010)	0.2523 (±0.0028)	**0.2789** (±0.0037)
	Macro-AP	0.2325 (±0.0015)	0.2316 (±0.0006)	0.2325 (±0.0009)	0.2360 (±0.0015)	0.2266 (±0.0011)	0.2017 (±0.0006)	0.2265 (±0.0016)	0.2352 (±0.0013)	0.2304 (±0.0012)	**0.2488** (±0.0010)
(e)	**Macro-F1**	0.1914 (±0.0029)	0.1971 (±0.0021)	0.1884 (±0.0007)	0.1807 (±0.0005)	0.1851 (±0.0027)	0.1725 (±0.0008)	0.1815 (±0.0065)	0.2022 (±0.0002)	0.1901 (±0.0024)	**0.2235** (±0.0046)
	Macro-AP	0.1654 (±0.0008)	0.1778 (±0.0005)	0.1636 (±0.0006)	0.1668 (±0.0007)	0.1627 (±0.0007)	0.1490 (±0.0003)	0.1613 (±0.0006)	0.1634 (±0.0005)	0.1649 (±0.0005)	**0.1916** (±0.0021)
(f)	**Macro-F1**	0.2038 (±0.0011)	0.2148 (±0.0039)	0.1896 (±0.0019)	0.1928 (±0.0015)	0.1885 (±0.0047)	0.1762 (±0.0006)	0.1963 (±0.0028)	0.2181 (±0.0033)	0.1968 (±0.0012)	**0.2184** (±0.0020)
	Macro-AP	0.1792 (±0.0008)	0.1774 (±0.0002)	0.1686 (±0.0008)	0.1733 (±0.0009)	0.1666 (±0.0007)	0.1583 (±0.0003)	0.1696 (±0.0006)	0.1755 (±0.0011)	0.1719 (±0.0005)	**0.1863** (±0.0015)
(g)	**Macro-F1**	0.2150 (±0.0029)	0.2409 (±0.0022)	0.2088 (±0.0033)	0.2089 (±0.0066)	0.2132 (±0.0048)	0.1900 (±0.0007)	0.2104 (±0.0036)	0.2231 (±0.0007)	0.2133 (±0.0004)	**0.2469** (±0.0021)
	Macro-AP	0.1836 (±0.0010)	0.1957 (±0.0005)	0.1841 (±0.0010)	0.1829 (±0.0076)	0.1838 (±0.0009)	0.1655 (±0.0004)	0.1805 (±0.0008)	0.1836 (±0.0008)	0.1827 (±0.0007)	**0.2185** (±0.0019)
(h)	**Macro-F1**	0.2219 (±0.0005)	0.2356 (±0.0037)	0.2041 (±0.0024)	0.2136 (±0.0017)	0.2064 (±0.0036)	0.1909 (±0.0005)	0.2145 (±0.0028)	**0.2414** (±0.0017)	0.2141 (±0.0018)	0.2397 (±0.0017)
	Macro-AP	0.1925 (±0.0011)	0.1910 (±0.0003)	0.1839 (±0.0008)	0.1881 (±0.0010)	0.1840 (±0.0009)	0.1693 (±0.0004)	0.1824 (±0.0007)	0.1913 (±0.0012)	0.1846 (±0.0006)	**0.2030** (±0.0009)

epoch cap as the MerRec probes. Checkpoints are selected by validation Macro-AP with a patience of five epochs. After selection, one decision threshold per label is chosen on the validation split to maximize classwise F1. We report test Macro-F1 and Macro-AP averaged across the six categories, with the mean and standard deviation computed over probe-head seeds 1234, 1235, and 1236.

ESP leads Macro-F1 at the shorter AppRec offsets and for the future-window dominant category (Figure 7; Table 8). ORec leads at the longest single-position offset, so the future-window advantage does not extend to every distant target.

ESP leads both Interior targets, while its End first-category result is nearly tied with ORec and its End dominant-category result trails ORec. Interior and End use different eligible test cohorts, so their score difference cannot isolate the effect of query position. The Interior results nevertheless show accessible future-event information before the boundary.

ESP leads mean Macro-AP on every AppRec task, including the longest offset and End dominant-category targets where it does not lead Macro-F1. Its advantage in category-score ranking is therefore more consistent than its advantage after validation-calibrated thresholding.

E.2 RetailRocket Ablation Results

Table 9 reports the RetailRocket ablations using the variants defined in Section 5.6. In the cosine-regression variant, each contrastive term is replaced by $1 - \cos(q, \mathrm{sg}(k))$, with the remaining configuration unchanged.

Removing either event objective lowers NDCG@10 and HR@10. Intra-event removal causes the larger NDCG@10 loss, whereas inter-event removal causes the larger HR@10 loss.

Table 9: ESP ablations on RetailRocket.

Variant	Representation Quality				Downstream Retrieval	
	RankMe/d ↑	IsoScore ↑	Unif. ↓	Aniso. ↓	NDCG@10 ↑	HR@10 ↑
ESP	0.9257	0.5868	-3.9449	0.0013	0.1520	0.1853
w/o $\mathcal{L}_{\text{intra}}$	0.9046 (↓2.28%)	0.5065 (↓13.68%)	-3.9377 (↑0.18%)	0.0011 (↓15.38%)	0.1315 (↓13.49%)	0.1746 (↓5.77%)
w/o $\mathcal{L}_{\text{inter}}$	0.9192 (↓0.70%)	0.5570 (↓5.08%)	-3.9431 (↑0.05%)	0.0018 (↑38.46%)	0.1337 (↓12.04%)	0.1678 (↓9.44%)
w/ cosine	0.6664 (↓28.01%)	0.1209 (↓79.40%)	-3.0825 (↑21.86%)	0.1462 (↑11146.15%)	0.0841 (↓44.67%)	0.1153 (↓37.78%)

Cosine regression causes the largest recommendation loss and a sharp decline in spectral diagnostics. Intra-event removal slightly improves anisotropy while reducing recommendation performance. No single geometry diagnostic explains the ablation outcomes.

F Further Discussion

Building on the main experiments, we examine whether frozen states support future intent prediction (RQ 6) and direct event-content retrieval (RQ 7), how performance depends on model width, event partitions, EMA momentum, and loss weights (RQ 8), whether the gains extend to another backbone (RQ 9), how joint training affects the NTP loss (RQ 10), what training and inference costs ESP incurs (RQ 11), and whether its gains persist under temporal shift (RQ 12). The supplementary ablations in Appendix E.2 further examine objective contributions.

F.1 Future User Intent Prediction on AppRec

We evaluate future user intent prediction on AppRec using six intent classes derived from real user click feedback. Rules based on application names, interaction duration, and related behavioral attributes map feedback within a future 15-minute event segment to intent labels. These labels characterize user intent within that segment and are distinct from the application-category labels used in the eight probes in Appendix E.1.7. The detailed labeling rules are proprietary to the private dataset and cannot be disclosed.

The query state is the causal encoder output at the last trigger position in the observed interaction history. We encode only the prefix through this position and predict the intent labels of the subsequent 15-minute event segment. As in the eight AppRec probes in Appendix E.1.7, the backbone and embeddings remain frozen, and the training heads are discarded. We train only a linear classification head that maps the 256-dimensional query state to six intent logits, using the same readout training settings as those probes. Figure 8 reports Macro-F1 across the six intent classes.

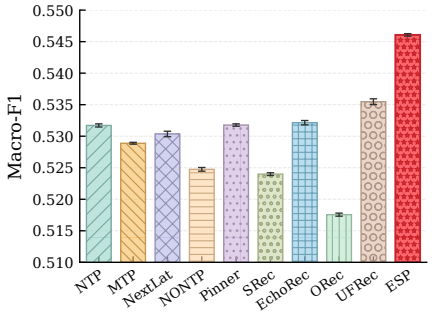


Figure 8: Frozen linear-probe Macro-F1 for future user intent prediction on AppRec. The query uses the last trigger position, and the six intent classes are derived from real user click feedback in the subsequent 15-minute event segment.

ESP achieves the highest Macro-F1, with UFRec as the strongest baseline (Figure 8). Its frozen states support more accurate linear readout of future intent than those learned by the other predictive objectives in this comparison. This result complements the category probes with a separate user-intent target derived from behavior within a future event segment.

F.2 Event-Content Retrieval

We withhold the final natural-hour event from each eligible MerRec sequence and use the causal state immediately before it as the query. Each candidate event is encoded separately by the same frozen backbone, with timestamps shifted to a common origin and token states mean-pooled. The evaluation contains 105,378 queries and 87,540 unique ordered event contents. Repeated occurrences

of an identical token tuple are averaged into one candidate prototype, and normalized queries and prototypes are ranked by cosine similarity. No event-retrieval head is trained. All methods use the same query sequences, candidate-event contents, and cosine-ranking protocol. Appendix D.5 gives the candidate construction and ranking details.

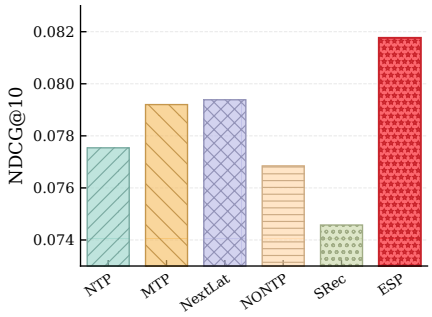


Figure 9: MerRec event-content retrieval from frozen states. Q0 denotes the state immediately before the target event.

ESP leads NDCG@10 for exact ordered-event retrieval from frozen states (Figure 9). The gain over NextLat is small and reflects direct state–event similarity without an NTP head; this evaluation does not isolate a training component.

F.3 Sensitivity to Model and Supervision Settings

F.3.1 Hidden Size

We compare NTP, MTP, NextLat, and ESP at $d \in \{64, 128, 256, 384\}$ using the common MerRec training setup while varying hidden size.

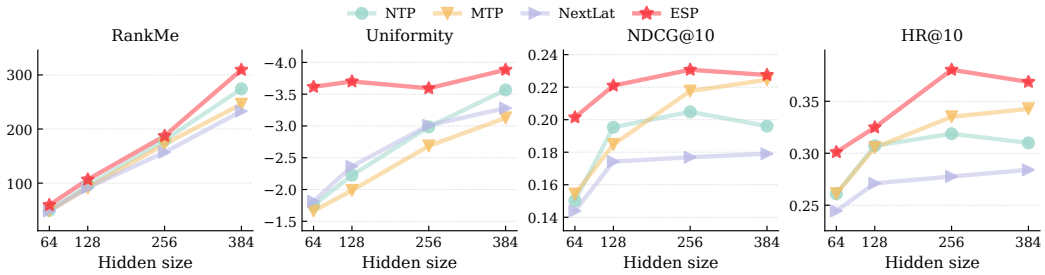


Figure 10: MerRec results across hidden sizes. RankMe is unnormalized effective rank, and the uniformity axis is inverted.

ESP leads NDCG@10 and HR@10 at every tested width, but its NDCG@10 margin over the strongest comparator shrinks as width increases (Figure 10). Its absolute effective rank rises with width, whereas rank normalized by width falls through $d = 256$ before recovering at $d = 384$.

ESP's NDCG@10 and HR@10 peak at $d = 256$ and decline at $d = 384$. The wider model also has the lowest uniformity, so greater state dispersion does not translate into better recommendation in this sweep.

F.3.2 Event Granularity

We evaluate ESP on KuaiRec with time-based event windows of 30 minutes, 1 hour, 4 hours, 12 hours, 1 day, and 2 days. Figure 11 compares these choices with fixed-count partitions of 8, 16, and 32 consecutive tokens, denoted Fix-8, Fix-16, and Fix-32. The partition determines which tokens share an event target and which event follows each source event.

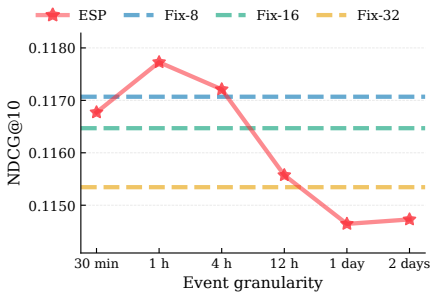


Figure 11: KuaiRec NDCG@10 across time-window and fixed-count event partitions.

Table 10: MerRec full-catalog next-token results across partition settings. The idle rule starts a new group after 30 minutes of inactivity; fixed rules use one-hour or 12-hour windows.

Setting	Method	NDCG@10	NDCG@20	HR@10	MRR
30m idle	SRec	0.1819	0.1969	0.2966	0.1808
	ESP	0.2286	0.2440	0.3793	0.2118
1h fixed	SRec	0.1871	0.2028	0.3224	0.1780
	ESP	0.2307	0.2459	0.3803	0.2140
12h fixed	SRec	0.1811	0.1975	0.3222	0.1702
	ESP	0.2152	0.2308	0.3663	0.1988

The natural-hour window gives the highest NDCG@10 in this KuaiRec sweep, with only a small advantage over other short windows and Fix-8 (Figure 11). Longer time windows and larger fixed-count groups yield lower scores.

F.3.3 Session Partition Choice on MerRec

Table 10 compares SRec and ESP on MerRec under a 30-minute inactivity split and fixed one-hour and 12-hour windows. Both methods use each partition rule; the main-comparison entries use the 30-minute idle rule for SRec and the one-hour rule for ESP (Table 1).

ESP leads SRec on all reported ranking metrics under each tested MerRec partition. Both methods lose NDCG@10 with the longest window; ESP changes little between the idle and one-hour rules. Its advantage over SRec therefore persists across these partition choices.

F.3.4 EMA Momentum

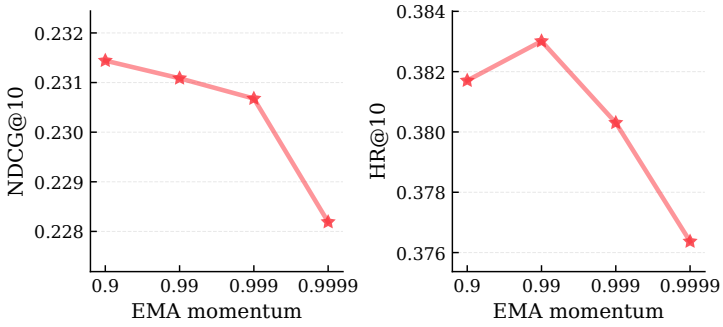


Figure 12: ESP recommendation on MerRec across EMA momenta.

We vary the target-encoder momentum on MerRec under the standard training setup (Figure 12). NDCG@10 favors the lowest tested momentum, whereas HR@10 peaks at the next higher setting. Both metrics reach their lowest values at the largest momentum. Higher momentum smooths target

updates but also slows their response to the online backbone (Equation 3); the decline is consistent with targets adapting too slowly within the fixed training budget. The common $\mu = 0.999$ used in the main comparisons is therefore not optimal in this sweep.

F.3.5 Event-Loss Weights

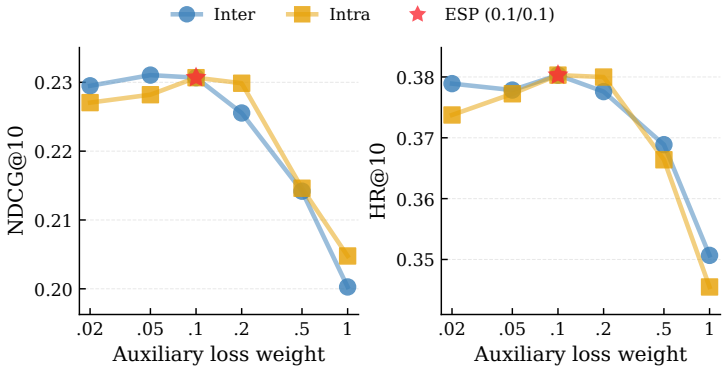


Figure 13: ESP recommendation on MerRec across event-loss weights. Stars mark the default setting.

Figure 13 compares intra- and inter-event loss-weight sweeps on MerRec. The default $\lambda_{\text{intra}} = \lambda_{\text{inter}} = 0.1$ is near the NDCG@10 peak and attains the highest HR@10 in both sweeps. Larger weights reduce both metrics, favoring moderate contributions from the event objectives.

F.4 ESP with a SASRec Backbone

We replace the HSTU backbone with SASRec (Kang & McAuley, 2018) and compare NTP-only training with the same ESP objectives on MerRec. All other settings follow the main experiment. Figure 14 reports NDCG@10, NDCG@20, HR@10, and full-ranking MRR.

Adding ESP improves all four metrics in this MerRec comparison (Figure 14). MRR rises from 0.1977 to 0.2081, while the relative gain is largest for HR@10. The recommendation benefit therefore also appears when SASRec replaces HSTU.

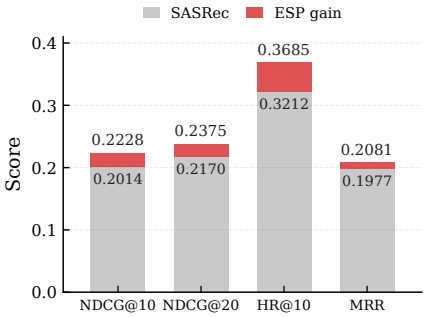


Figure 14: MerRec next-token recommendation under the SASRec backbone. Gray bars show NTP-only scores; red segments show the gain from adding ESP.

F.5 Training Dynamics

Figure 15 tracks optimization under the standard MerRec training setup. All three ESP loss components decrease, with most of the reduction occurring early in training. On the shared NTP objective, ESP and NTP-only training follow similar early trajectories, while ESP reaches a lower loss in later epochs. The auxiliary objectives thus remain compatible with fitting the next-token target. The lower late-stage NTP loss complements the held-out recommendation gains in Table 1.

F.6 Training and Inference Efficiency

We profile MerRec training in FP32 on eight Tesla V100-SXM2-32GB GPUs at a global batch size of 512. After 100 warm-up steps, throughput is computed from the median of four consecutive 10-step windows. ORec throughput includes both training stages. Relative time is the inverse-throughput

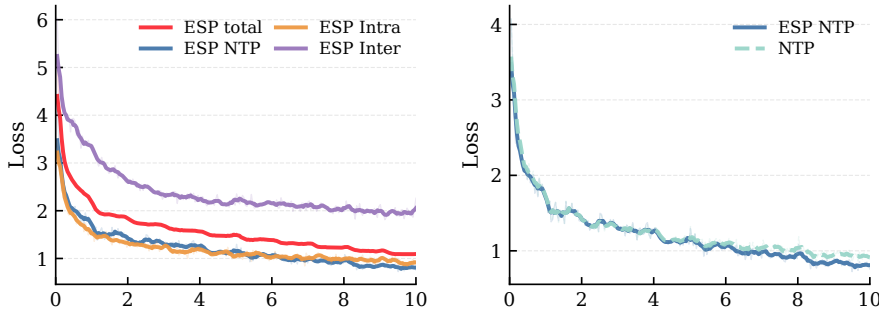


Figure 15: MerRec training losses over ten epochs. Left: ESP's total objective and its components; event losses are shown before weighting. Right: the NTP loss under ESP and NTP-only training. Both horizontal axes show training epoch.

Table 11: MerRec training throughput and peak allocated memory. Relative time is normalized to NTP; ORec (†) includes both training stages.

Method	Throughput (k seq/s) ↑	Relative time ↓	Peak allocation (GiB) ↓
NTP	9.18	1.00×	2.51
MTP	6.13	1.50×	4.92
NextLat	7.95	1.15×	2.53
NONTP	5.81	1.58×	3.28
Pinner	5.78	1.59×	1.32
SRec	7.92	1.16×	2.52
EchoRec	4.70	1.95×	4.52
ORec†	2.87	3.20×	3.75
UFRec	7.05	1.30×	3.34
ESP	6.68	1.37×	2.53

ratio normalized to NTP. Peak allocation is measured on rank 0, using the larger stage-wise value for ORec. Each entry comes from one profiling run.

ESP trains more slowly than NTP with nearly the same peak allocation (Table 11). Its additional computation is confined to training; the deployed query path matches NTP.

F.7 Forward Temporal OOD Recommendation on AppRec

Table 12: Forward temporal OOD recommendation on AppRec.

Metric	NTP	MTP	NextLat	NONTP	Pinner	SRec	EchoRec	ORec	UFRec	ESP
N@10	0.6729	0.6657	0.6713	0.6684	0.5619	0.6496	0.6626	0.5918	0.6763	**0.6849**
N@20	0.6835	0.6767	0.6825	0.6789	0.5784	0.6593	0.6727	0.6071	0.6857	**0.6948**
H@10	0.8687	0.8646	0.8652	0.8626	0.7794	0.8430	0.8506	0.7827	0.8655	**0.8745**
MRR	0.6169	0.6091	0.6156	0.6132	0.5043	0.5954	0.6099	0.5425	0.6224	**0.6305**

We evaluate temporal generalization by training on an earlier AppRec snapshot (14 days) and testing the resulting checkpoints on a later snapshot (7 days) without parameter updates. User segments are concatenated in chronological order, the latest 128 interactions are retained, and users with fewer than 20 interactions are excluded. The evaluation uses one sequence per sampled user. All compared methods use the same evaluation sequences and candidates. Comparisons are made within this forward OOD set, since its sequence materialization differs from the in-domain evaluation.

ESP leads all reported metrics on the later AppRec snapshot (Table 12). Its NDCG@10 gain over UFRec is modest, and the same checkpoint is evaluated without adaptation to the later data.