# MARKOVIAN PRE-TRAINED TRANSFORMER FOR NEXT-ITEM RECOMMENDATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

We introduce the Markovian Pre-trained Transformer (MPT) for next-item recommendation, a transferable model fully pre-trained on *synthetic* Markov chains, yet capable of achieving state-of-the-art performance by fine-tuning a lightweight adaptor. This counterintuitive success can be attributed to the observed 'Markovian' characteristics: advanced sequential recommenders tend to rely predominantly on the latest interaction for prediction, while historical interactions primarily provide contextual signals for inferring the user's *general, non-sequential* identity. These characteristics necessitate the capabilities of a universal recommendation model to ❶ effectively summarize the user sequence, with ❷ particular emphasis on the latest interaction. MPT acquires these inherent capabilities from carefully constructed synthetic data. On the one hand, when trained to predict the next state of an *arbitrary* Markov chain, it must learn to ❶ estimate transition probabilities from contextual information (one adaptive manner for sequence summarization), and ❷ attend to the most recent state to ensure an accurate restart point. On the other hand, unlike the heterogeneous interaction data, an unlimited number of controllable Markov chains is available to boost the model capacity. Extensive experiments on five public datasets from three distinct platforms validate the superiority of Markovian pre-training over traditional recommendation pre-training (+39% on average) and recent language pre-training (+24% on average) paradigms. Our code is available at https://anonymous.4open.science/r/MPT.

## 1 INTRODUCTION

The pre-training and fine-tuning workflow has become the *de facto* paradigm across diverse domains (Chen et al., 2020; Devlin et al., 2019; Radford et al., 2021), allowing for significant computational savings through efficient knowledge transfer. In next-item recommendation, a model trained over users' historical interactions is unfortunately domain-specific even when leveraging text embeddings as input. Consequently tedious retraining must be performed over and over to keep adaptive to the latest scenarios. Although considerable efforts (Ding et al., 2021; Hou et al., 2022; Li et al., 2023a) have been devoted to developing a universal pre-trained recommender, the performance still lags far behind that of models trained from scratch. This discrepancy could be attributed to the heterogeneity of public datasets, ranging from diverse user behaviors to non-negligible domain gaps (Ju et al., 2025; Ye et al., 2026). Some researchers have therefore turned to foundation models that are not specifically pre-trained for recommendation tasks, *e.g.*, Large Language Models (LLMs) (Li et al., 2023b; Liao et al., 2024); however, these approaches remain inadequate in terms of accuracy and, more critically, efficiency. In this paper, we aim to pre-train a universal Transformer-based (Vaswani et al., 2017) model for next-item recommendation by addressing the following two fundamental questions: **(RQI)** Which capabilities are transferable across diverse recommendation scenarios? **(RQII)** What types of data are sufficient to stimulate the required capabilities?

We first empirically identify the 'Markovian' characteristics of next-item recommendation through a systematic evaluation of sequential recommenders, including GRU4Rec (Hidasi et al., 2016), SASRec (Kang & McAuley, 2018), and HSTU (Zhai et al., 2024), across several public datasets. Given a chronologically ordered sequence of historical interactions $[v_1, v_2, \ldots, v_t]$, we compare in Figure 1 how NDCG varies when the input sequences are either partially shuffled (with the latest interaction $v_t$ fixed) or completely shuffled. If the chronological order is essential for a sequential model to perform complex yet more accurate inference (*e.g.*, involving logical reasoning), both the



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.

shuffled variants would be expected to cause substantial performance degradation. Surprisingly, our results show that, **as long as the latest interaction remains fixed, performance remains largely unchanged across datasets or platforms**. These observations suggest that current advanced sequential recommenders tend to make next-item predictions in a Markovian fashion: they are capable of ❶ summarizing non-sequential user preferences from historical interactions, while ❷ placing particular emphasis on the most recently interacted item. Importantly, these capabilities are transferable across diverse datasets and platforms, pointing to a promising direction for developing specialized pre-trained recommendation models.

Analogous to the fundamental role of mathematical data in stimulating the logical reasoning ability of LLMs (Huan et al., 2025; Liu et al., 2024), it is worthwhile to ask whether specific types of data can similarly stimulate the required recommendation capabilities. The Markovian Pre-trained Transformer (MPT), fully pre-trained on synthetic Markov chains, can accurately predict the next possible state of an arbitrary trajectory, even if the sampled trajectory has never been encountered during training. This implies that MPT must possess the capabilities to ❶ estimate intrinsic transition probabilities from the context (one adaptive manner for sequence summarization) and ❷ attend to the last state to ensure an accurate restart point. They correspond to the aforementioned recommendation capabilities, one by one, also suggesting a novel **Markovian pre-training and recommendation fine-tuning paradigm**. Unlike the heterogeneous interaction data, the synthetic Markov chains are both unlimited and controllable. We thoroughly investigate the superiority of MPT across five public datasets from three distinct platforms. In most scenarios, state-of-the-art performance can be achieved using the input adaptor alone, yielding relative improvements of approximately 8% (on average) over domain-specific recommenders and 24% (on average) over pre-trained methods. Notably, such improvements are progressively realized after pre-training on 10B tokens.

Our contributions to the community are three-fold: **(I)** We systematically identify two transferable capabilities for next-item recommendation, rather than limiting our analysis to obscure concepts. **(II)** While prior work on recommendation pre-training primarily focuses on leveraging heterogeneous interaction data or resorting to language pre-training, the success of MPT highlights a fundamentally novel paradigm: exploiting synthetic data to endow recommenders with the required capabilities. **(III)** We observe a clear improvement in downstream performance as more synthetic data is incorporated to foster those transferable capabilities. Notably, unlike the well-known scaling laws for LLMs (Kaplan et al., 2020; Hoffmann et al., 2022), Markovian pre-training converges to a theoretical limit governed by the Bayes estimator (Berger, 2013).

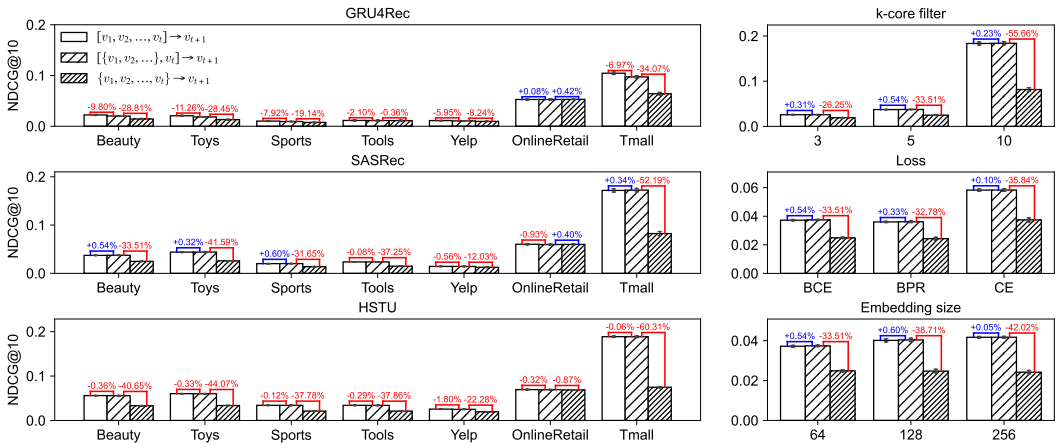
## 2 Markovian Characteristic of Next-Item Prediction

Given a user's chronological interaction sequence $[v_1, v_2, \ldots, v_t]$, the next-item recommendation task aims to predict the subsequent item $\hat{v}_{t+1} \in \mathcal{V}$ that the user is likely to engage with. As expected, architectures such as RNNs (Hidasi et al., 2016) and Transformers (Kang & McAuley, 2018) have demonstrated superior performance in this field owing to their strong capability to capture sequential dependencies. They typically encode the interaction sequence into a vector representation and compute relevance scores over the item set $\mathcal{V}$ using inner product or cosine similarity. The top-ranked items are then considered as candidates for recommendation to the user. To facilitate the examination of the 'Markovian' characteristics, we formulate the next-item prediction as a process of maximizing the conditional probability:

$$\hat{v}_{t+1} = \operatorname*{argmax}_{v \in \mathcal{V}} \mathbb{P}\left(v \mid v_t, v_{t-1}, \ldots, v_1; \Theta\right), \tag{1}$$

where $\mathbb{P}(\cdot \mid \cdot; \Theta)$ gives the probability estimated by a sequential model conditioned on the interaction sequence and $\Theta$ represents its learnt parameters. For instance, GRU4Rec recurrently incorporates items; SASRec captures sequential dependencies in parallel via causal attention; and HSTU further exploits pointwise aggregated attention and time-based positional encoding.

As increasingly powerful architectures are employed for next-item recommendation, an important question arises: do they perform next-item prediction in an increasingly complex fashion? Here, we are to demonstrate that sophisticated models, particularly Transformers, tend to make predictions in a more straightforward manner. To substantiate this argument, we examine three variants whose




(a) Model & Dataset (b) Filter & Loss & Size

Figure 1: Comparisons of the sequence ordering (during inference). $[v_1, v_2, \ldots, v_t]$: chronologically ordered sequence; $[\{v_1, v_2, \ldots\}, v_t]$: partially shuffled sequence; $\{v_1, v_2, \ldots, v_t\}$: completely shuffled sequence. **(a)** Performance across various models and datasets. **(b)** Performance on Beauty across filtering strategies, training objectives, and embedding sizes.

accessible sequential information is progressively disrupted by different shuffling strategies:

$$\begin{aligned} \text{① Chronologically ordered:} \quad & \mathbb{P}\left(v_{t+1} \mid v_t, v_{t-1}, \ldots, v_1; \Theta\right), \\ \text{② Partially shuffled:} \quad & \mathbb{P}\left(v_{t+1} \mid v_t, \{v_1, v_2, \ldots\}; \Theta\right), \\ \text{③ Completely shuffled:} \quad & \mathbb{P}\left(v_{t+1} \mid \{v_1, v_2, \ldots, v_t\}; \Theta\right). \end{aligned}$$

Here, $\{\cdot\}$ denotes a permuted set (implemented through a random shuffle operation). The first keeps the original chronological order, thereby allowing the model to capture all sequential patterns. The second is partially shuffled except for the last item; consequently, the remaining interactions primarily serve as the non-sequential context for inferring the user's general identity. The third strategy applies a completely random shuffle, indeed degenerating into non-sequential prediction. It is noteworthy that the three variants are trained under the same conventional protocol (*i.e.*, sharing the same $\Theta$), differing only in the input sequence order during inference.

We examine in Figure 1 how NDCG@10 changes as the sequence order is progressively disrupted. To ensure the reliability of this evaluation, Figure 1 presents results across three classical models and seven publicly available datasets spanning diverse domains and platforms, while accounting for variations in embedding sizes, training objectives, and dataset filtering strategies. We have the following key observations:

- ① $\approx$ ②: Unlike the fully chronological order, which allows for the exploitation of all potential sequential patterns, in the partially shuffled variant only the most recently interacted item $v_t$ preserves minimal temporal relevance *w.r.t.* the target item. If advanced sequential recommenders truly rely on sequential information for complex yet more accurate inference, such shuffling would be expected to cause an irreversible performance degradation. However, although SASRec and HSTU are more expressive than GRU4Rec and thus achieve greater performance gains, both models appear rather insensitive to this disruption. Moreover, even the introduction of time-based positional encoding (HSTU) fails to alter this situation.
- ①/② $\gtrapprox$ ③: Randomly shuffling the entire sequence completely disrupts the involved sequential information. Consequently, these 'sequential' models typically fail to achieve competitive performance under such conditions. This suggests that in most practical scenarios, at least the latest interaction still plays an indispensable role for next-item prediction. GRU4Rec exhibits an exception due to its recurrent mechanism, which inherently biases the model toward historical information and lacks the capability to place particular emphasis on the most recent interaction (see Appendix C.2 for a detailed discussion).

The observations above collectively suggest a somewhat counterintuitive conclusion: state-of-the-art sequential recommenders do not perform very complex inference to make use of sequential

dependencies. Instead, they primarily summarize general user identity while emphasizing the latest interaction. Formally,

$$\mathbb{P}\left(v_{t+1} \mid v_t, v_{t-1}, \ldots, v_1; \Theta\right) \approx \mathbb{P}(v_{t+1} \mid v_t, \underbrace{\{v_1, v_2, \ldots\}}_{\text{non-sequential context}}; \Theta). \tag{2}$$

This suggests some 'Markovian' characteristics of next-item recommendation: conditioning on the entire interaction sequence is essentially equivalent to conditioning on the most recent interaction along with a non-sequential context.

**Remark 1.** *It is possible that the same phenomenon observed in Figure 1 arises when sequential recommenders strictly follow a Markovian formulation of $\mathbb{P}(v_{t+1} \mid v_t; \Theta)$. We further demonstrate in Appendix C.3 that both the most recent interaction and the non-sequential context are essential.*

**Remark 2.** *We do not intend to claim that the 'Markovian' property holds universally across all scenarios. In fact, we identify a scenario in which the sequential structure is not user-driven but rather platform-driven, causing the 'Markovian' characteristics to break down (see Appendix C.4).*

## 3 Markovian Pre-Trained Transformer

The above studies shed light on the transferable capabilities necessary for developing universal pre-trained recommenders: ❶ effectively summarizing users' general identity based on historical interactions; ❷ placing particular emphasis on the most recently interacted item. The remaining challenge lies in how to effectively stimulate these capabilities. Public recommendation datasets, which have already been extensively explored in prior studies, might not be a feasible choice in this context due to their heterogeneity ranging from diverse user behaviors to non-negligible domain gaps. Motivated by the fundamental role of mathematical data in enhancing the logical reasoning ability of LLMs, we turn to synthetic Markov chains. Recall that a Markov chain is a discrete stochastic process in which the next probable state, $s_{t+1}$, is determined solely by the current state, $s_t$:

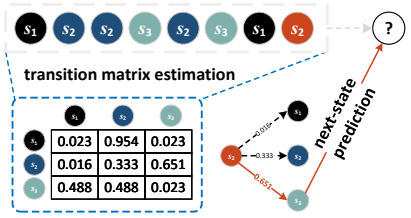



Figure 2: What MPT is expected to do in order to predict the most probable next state of an arbitrary Markov trajectory.

$$\mathbb{P}(s_{t+1} \mid s_t, s_{t-1}, \ldots, s_1) = \mathbb{P}(s_{t+1} \mid s_t), \quad s_n \in \mathcal{S}, \quad \forall n = 1, 2, \ldots, t+1,$$

where $\mathcal{S}$ denotes the state space. To predict the most probable next state of an *arbitrary* sampled trajectory, as shown in Figure 2, one must (I) identify the current state $s_t \in \mathcal{S}$ and (II) estimate the transition probabilities based on the observed frequencies from $s_t$ to the other states. If a sequential model is capable of next-state prediction solely based on the context rather than memorization, it must possess at least the following two capabilities: ❶ adaptively summarizing the transition matrix from the sampled trajectory alone; ❷ placing particular attention on the current state. These precisely correspond to the required recommendation capabilities. Fortunately, the Transformer architecture has been shown to exhibit such in-context learning capabilities in this regard (Müller et al., 2022; Hollmann et al., 2025; Lepage et al., 2025). Thus, it is reasonable to expect that MPT could serve as a universal pre-trained model for next-item recommendation. In the following, we provide a detailed introduction to the Markovian pre-training and recommendation fine-tuning paradigm.

### 3.1 Markovian Pre-Training

As illustrated in Figure 3 (and Algorithm 1), MPT is fully pre-trained on synthetic Markov chains via a next-state prediction task:

$$\mathcal{L}_{\text{NSP}}(\Theta) = \mathop{\mathbb{E}}_{\mathbf{P} \sim \text{Dir}(\boldsymbol{\alpha})} \mathop{\mathbb{E}}_{\{s_t\}_{t=2}^{T} \sim \mathbf{P} \mid s_1} - \sum_{t=1}^{T-1} \log \mathbb{P}(s_{t+1} \mid s_t, \ldots, s_1; \Theta). \tag{3}$$

At each training step, a batch of trajectories is independently sampled based on the transition probability matrices $\mathbf{P}$, which are generated from a Dirichlet prior $\text{Dir}(\boldsymbol{\alpha})$. There are some key components affecting the behaviors of the Markov chains:

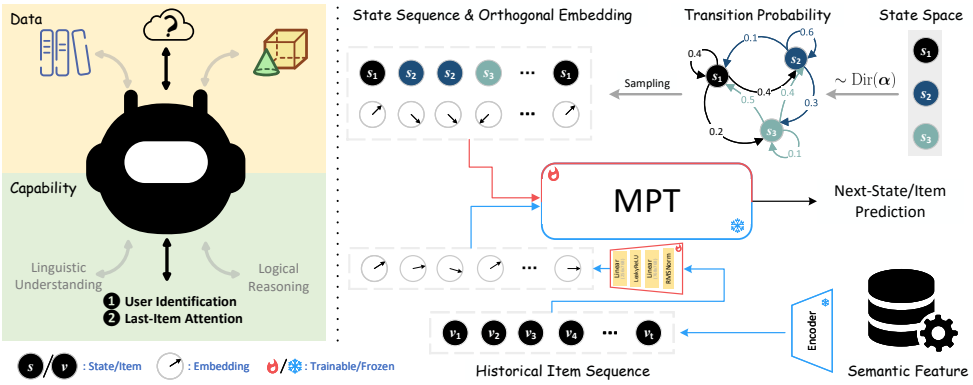



Figure 3: Overview of MPT. **Left:** Analogous to how linguistic corpora and mathematical data stimulate language understanding and logical reasoning, we aim to identify what types of data can effectively activate the capabilities necessary for next-item recommendation. **Right:** Pipeline of Markovian pre-training and recommendation fine-tuning.

- **State space size** $|\mathcal{S}|$. Accurately estimating transition probabilities requires a sufficient number of observed transitions for each state $s \in \mathcal{S}$. Given the fixed maximum input length of a Transformer (*e.g.*, $T = 1024$ in this paper), it is necessary to control the number of states to ensure that most predictions remain tractable; otherwise, MPT will merely learn to make uninformed predictions.
- **Dirichlet distribution Dir**$(\alpha)$. The concentration vector $\boldsymbol{\alpha} \in \mathbb{R}_+^{|\mathcal{S}|}$ controls the uniformity of the transition probabilities. For simplicity, this vector is assigned the identical value of $\alpha$ across all components. As $\alpha$ increases, the current state $s_t$ is more likely to transition to other states with roughly equal probability; conversely, a smaller $\alpha$ causes some states to be revisited frequently while others are seldom visited. For next-item recommendation, we find that pre-training MPT with a slightly smaller $\alpha$ (*e.g.*, 0.05) yields better results. On the one hand, considering the $|\mathcal{S}|^2$ possible combinations, the estimation of a '*dense*' transition matrix becomes an over-parameterization problem if every combination carries a non-negligible probability. On the other hand, item-to-item relationships in practical scenarios tend to be highly sparse.

Recall that MPT in this study is expected to serve as a universal model for various downstream recommendation scenarios. Therefore, we promote insensitivity to the specific input space by generating orthogonal vectors as the state representations, rather than relying on a conventional learnable embedding table. This operation is also performed independently for each sampled trajectory, ensuring that MPT never encounters the same input twice.

**Scaling laws**, which describe how loss scales as a power law *w.r.t.* data volume and model size, have been extensively investigated in both large language models (Kaplan et al., 2020; Hoffmann et al., 2022) and recommender systems (Zhai et al., 2024; Zhang et al., 2024). However, the scaling behavior has an inherent limit in Markovian pre-training. Given a Dirichlet prior Dir$(\alpha)$, the posterior transition probabilities also follow a Dirichlet distribution (Lin, 2016) augmented by the observed occurrences (please refer to Appendix A for details). Consequently, Markovian pre-training admits a well-defined Bayes estimator (Berger, 2013), also implying that the achievable performance is inherently governed by this theoretical limit. As shown in Section 4.2, MPT with four layers and a hidden size of 256 nearly reaches this limit. Further increasing the model capacity yields marginal gains and is therefore not cost-effective.

### 3.2 Recommendation Fine-Tuning

Although extensive Markovian pre-training endows the model with the required capabilities, recommendation fine-tuning remains necessary due to the mismatch between the pre-training and the downstream task space. In line with prior pre-trained recommenders, we use semantic features as the raw item representations. For instance, for the Amazon Reviews dataset, each item's *title*, *categories*, and *brand* are jointly encoded by a sentence transformer (*sentence-T5-xl* in this study) to obtain a

Table 1: Overall performance comparison. The best results of baselines and ours are separately marked in **bold** . $\Delta_1$ ($\Delta_2$): the relative performance gap of MPT with respect to ① (②).

| | | ① Training from scratch | | | | | ② Pre-training & Fine-tuning | | | | | MPT | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | FPMC | FMLPRec | HSTU | SASRec+ | GLINT-RU | UniSRec | RecFormer | E4SRec | LLM2Rec | Qwen2.5 | Adaptor | +LoRA | $\Delta_1$ | $\Delta_2$ |
| Beauty | HR@10 | 0.0722 | 0.0775 | 0.0902 | **0.0982** | 0.0857 | 0.0803 | 0.0676 | 0.0773 | **0.0853** | 0.0826 | **0.1118** | 0.1100 | 13.9% | 31.1% |
| | HR@20 | 0.0991 | 0.1072 | 0.1246 | **0.1348** | 0.1154 | **0.1233** | 0.1017 | 0.1046 | 0.1222 | 0.1200 | **0.1531** | 0.1514 | 13.6% | 24.2% |
| | NDCG@10 | 0.0440 | 0.0454 | 0.0535 | **0.0591** | 0.0530 | 0.0382 | 0.0333 | **0.0469** | 0.0457 | 0.0464 | **0.0673** | 0.0658 | 13.8% | 43.3% |
| | NDCG@20 | 0.0508 | 0.0529 | 0.0621 | **0.0683** | 0.0604 | 0.0490 | 0.0419 | 0.0538 | 0.0552 | **0.0558** | **0.0777** | 0.0763 | 13.7% | 39.1% |
| Toys | HR@10 | 0.0787 | 0.0855 | 0.0981 | **0.1040** | 0.0944 | 0.0829 | 0.0842 | 0.0798 | **0.0902** | 0.0784 | **0.1202** | 0.1192 | 15.5% | 33.3% |
| | HR@20 | 0.1022 | 0.1124 | 0.1288 | **0.1393** | 0.1223 | 0.1235 | 0.1221 | 0.1061 | **0.1315** | 0.1114 | **0.1617** | 0.1611 | 16.1% | 23.0% |
| | NDCG@10 | 0.0503 | 0.0521 | 0.0610 | **0.0645** | 0.0610 | 0.0380 | 0.0405 | **0.0492** | 0.0436 | 0.0433 | **0.0705** | 0.0701 | 9.4% | 43.3% |
| | NDCG@20 | 0.0563 | 0.0589 | 0.0687 | **0.0733** | 0.0680 | 0.0483 | 0.0501 | **0.0558** | 0.0541 | 0.0515 | **0.0809** | 0.0806 | 10.4% | 45.0% |
| Sports | HR@10 | 0.0398 | 0.0413 | 0.0550 | **0.0576** | 0.0493 | 0.0446 | 0.0345 | 0.0387 | 0.0392 | **0.0473** | **0.0651** | 0.0644 | 12.9% | 37.7% |
| | HR@20 | 0.0562 | 0.0593 | 0.0788 | **0.0824** | 0.0684 | 0.0699 | 0.0555 | 0.0563 | 0.0594 | **0.0712** | 0.0924 | **0.0927** | 12.5% | 30.2% |
| | NDCG@10 | 0.0238 | 0.0238 | 0.0317 | **0.0337** | 0.0298 | 0.0221 | 0.0167 | 0.0218 | 0.0204 | **0.0256** | **0.0373** | 0.0365 | 10.8% | 45.7% |
| | NDCG@20 | 0.0279 | 0.0283 | 0.0377 | **0.0399** | 0.0346 | 0.0285 | 0.0220 | 0.0262 | 0.0255 | **0.0316** | **0.0442** | 0.0436 | 10.7% | 39.7% |
| Yelp | HR@10 | 0.0262 | 0.0334 | 0.0453 | **0.0466** | 0.0407 | 0.0343 | 0.0247 | 0.0291 | 0.0124 | **0.0411** | 0.0461 | **0.0495** | 6.3% | 20.5% |
| | HR@20 | 0.0463 | 0.0606 | 0.0773 | **0.0796** | 0.0686 | 0.0567 | 0.0429 | 0.0513 | 0.0180 | **0.0682** | 0.0783 | **0.0832** | 4.4% | 21.9% |
| | NDCG@10 | 0.0127 | 0.0138 | **0.0215** | 0.0209 | 0.0193 | 0.0170 | 0.0121 | 0.0139 | 0.0064 | **0.0204** | 0.0224 | **0.0241** | 12.0% | 17.8% |
| | NDCG@20 | 0.0177 | 0.0207 | **0.0295** | 0.0292 | 0.0263 | 0.0226 | 0.0166 | 0.0195 | 0.0082 | **0.0272** | 0.0305 | **0.0325** | 10.1% | 19.6% |
| Retail | HR@10 | 0.1390 | 0.1333 | 0.1452 | **0.1589** | 0.1327 | 0.1424 | 0.1407 | **0.1551** | 0.1298 | 0.1503 | **0.1608** | 0.1568 | 1.2% | 3.7% |
| | HR@20 | 0.2234 | 0.2237 | 0.2304 | **0.2575** | 0.2130 | 0.2275 | 0.2194 | **0.2528** | 0.2018 | 0.2356 | **0.2572** | 0.2502 | -0.1% | 1.8% |
| | NDCG@10 | 0.0651 | 0.0548 | 0.0656 | **0.0719** | 0.0607 | 0.0662 | 0.0683 | **0.0749** | 0.0641 | 0.0731 | 0.0741 | **0.0746** | 3.8% | -0.4% |
| | NDCG@20 | 0.0863 | 0.0775 | 0.0870 | **0.0968** | 0.0809 | 0.0877 | 0.0881 | **0.0995** | 0.0822 | 0.0945 | **0.0982** | 0.0980 | 1.5% | -1.3% |

unified vector representation. Remark that MPT is technically agnostic to textual, visual, and other forms of item representation. Compared with conventional learnable item embeddings, semantic features have been shown to yield comparable performance while being particularly well suited for cold-start items (Yuan et al., 2023). This is more consistent with the principles of transferability and universality pursued in this work.

As shown in Figure 3, the recommendation fine-tuning pipeline employs only a lightweight input adaptor, consisting of an RMSNorm (Zhang & Sennrich, 2019) followed by a two-layer MLP with a LeakyReLU activation. One may explore other parameter-efficient tuning techniques (*e.g.*, LoRA); however, it should be emphasized that the learnt capabilities of MPT might be disrupted once its inherent parameters are re-tuned. This issue stems from the heterogeneity present in certain recommendation datasets, where adapting MPT to highly specific scenarios would sometimes degrade its general capabilities. In other words, in most scenarios, we only need to focus on aligning the item space with MPT. This not only significantly reduces the overhead of fine-tuning but also suggests the generalizability of the aforementioned capabilities.

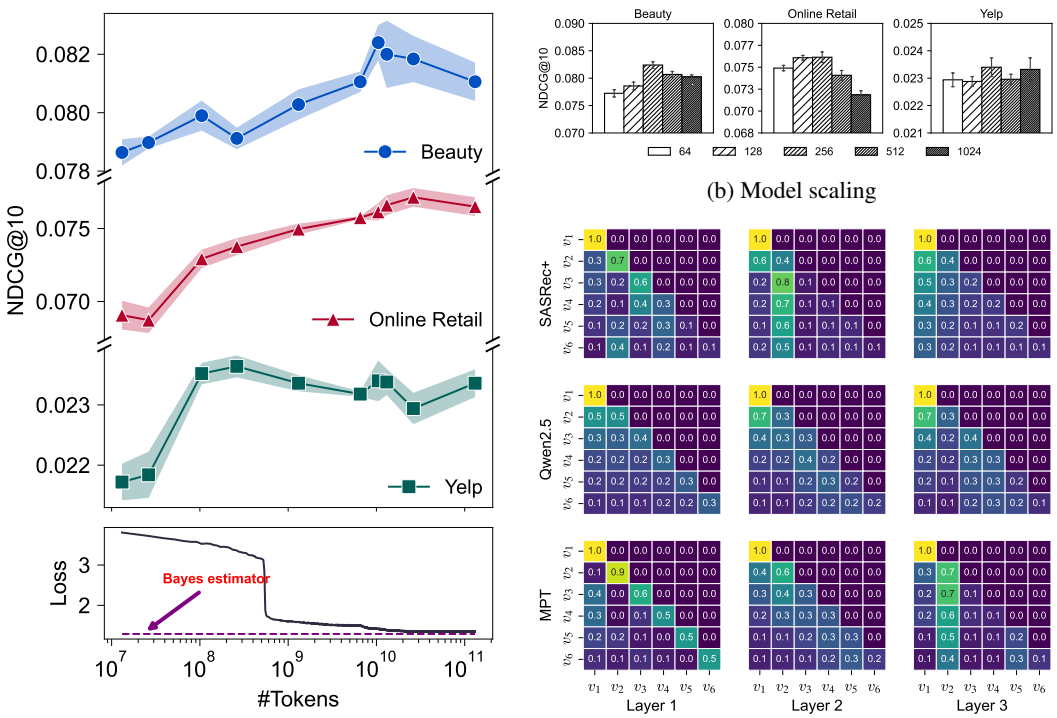
MPT is fine-tuned using a next-item prediction objective:

$$\mathcal{L}_{\text{NIP}}(\theta) = -\sum_{t=1}^{T-1} \log \mathbb{P}(v_{t+1} \mid v_t, \ldots, v_1; \Theta, \theta). \tag{4}$$

Here, $\theta$ denotes the additional parameters involved in the adaptor. In particular, we employ cosine similarity rather than the inner product for scoring, and incorporate a temperature hyperparameter to achieve appropriate rescaling. This is a common practice in sequential recommendation (Zhai et al., 2024; Hou et al., 2022). Note that $\mathcal{L}_{\text{NIP}}$ intrinsically corresponds to a cross-entropy loss computed over the entire item corpus, which may raise memory concerns in practical deployments. Although memory-efficient alternatives (Klenitskiy & Vasilev, 2023; Xu et al., 2024) to cross-entropy exist, we retain this objective by default to avoid introducing additional engineering factors.

## 4 Experiments

To evaluate the transferability of MPT, we consider several representative recommendation-oriented pre-trained models, including UniSRec (Hou et al., 2022) and RecFormer (Li et al., 2023a), as well as LLM-based approaches such as E4SRec (Li et al., 2023b) and LLM2Rec (He et al., 2025). To ensure a fair comparison, Qwen-2.5-7B (Yang et al., 2025) is configured under the same training protocol as MPT via an input adaptor. In addition, we include several traditional sequential models trained from scratch, among which FPMC (Rendle et al., 2010) represents a method utilizing personalized Markov chains. Following prior work, SASRec+ (Klenitskiy & Vasilev, 2023) is also included to control for




(a) Data scaling

(b) Model scaling

(c) Attention map

Figure 4: **(a)** How the $\mathcal{L}_{\text{NSP}}$ loss and validation NDCG@10 performance evolve as more tokens are learnt. **(b)** The validation NDCG@10 performance under varying hidden dimension sizes. **(c)** Attention maps of SASRec+, Qwen2.5, and MPT over a user sequence. The inference backbones of both Qwen2.5 and MPT are not fine-tuned using any recommendation data.

differences in training objectives. All experiments are conducted on five public datasets from three distinct platforms. Further experimental details are provided in Appendix B.

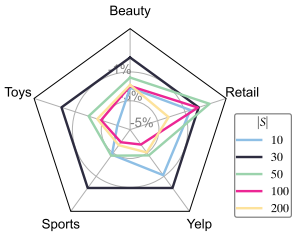
### 4.1 Overall Comparison

In this section, we investigate the universality of MPT by comparing its performance across five public datasets. We also report the results of generative recommendation models (Rajput et al., 2023) in Appendix C.7. From Table 1, we derive the following key observations:

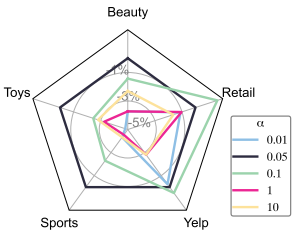
- UniSRec and RecFormer, pre-trained on Amazon Reviews, underperform not only specialized models trained from scratch but also methods such as E4SRec and Qwen2.5, which are pre-trained solely on textual corpora. This suggests that recommendation-oriented pre-training struggles to learn transferable capabilities, likely due to heterogeneous user behaviors and domain gaps. Although E4SRec and Qwen2.5 achieve competitive results, their substantial inference cost limits practical efficiency (see Appendix C.1).
- MPT achieves strong performance by fine-tuning only a lightweight input adaptor, outperforming SASRec+ by approximately 12%. On datasets such as Online Retail that include only item titles for semantic representations, its advantage over learnable ID-based methods is less pronounced. By incorporating learnable ID embeddings, the NDCG@10 of MPT on Online Retail will improve from 0.0746 to 0.0766. Furthermore, both MPT and Qwen2.5 are pre-trained without any recommendation-specific guidance and are fine-tuned in exactly the same manner; however, despite its much smaller model size, MPT surpasses Qwen2.5 by about 32%, indicating that larger capacity and higher cost do not necessarily yield better performance.
- LoRA-based fine-tuning generally degrades MPT performance relative to using only the input adaptor, suggesting that the pre-trained representations are already sufficiently expressive. While similar observations have been reported in other domains (Zhai et al., 2022; Tschannen et al., 2025),

| | Beauty | Yelp | Retail |
|---|---|---|---|
| MF-BPR | 0.0474 | 0.0189 | 0.0739 |
| SASRec | 0.0591 | 0.0190 | 0.0717 |
| AlphaRec | 0.0657 | 0.0228 | 0.0755 |
| T5-xl | 0.0673 | 0.0224 | 0.0741 |

(a) Item embedding sources




(b) State space size $|\mathcal{S}|$




(c) $\alpha$ of Dirichlet distribution

Figure 5: Ablation studies (NDCG@10 metrics). **(a)** MPT performance over various embedding sources. **(b)/(c)** Sensitivity analysis of state space size $|\mathcal{S}|$ and $\alpha$ of Dirichlet distribution. For clarity, we report the performance gaps relative to the default setting (*i.e.*, $|\mathcal{S}| = 30$ and $\alpha = 0.05$).

this is, to our knowledge, the first evidence in recommendation. Of course, the capabilities acquired through Markovian pre-training should not be expected to perfectly adapt to all scenarios. An exception is the Yelp dataset, where LoRA improves performance, implying that more complex sequence modeling is required in such cases.

The following experiments are conducted under the input-adaptor-only setting, as it has been shown to be sufficiently effective.

### 4.2 Empirical Analysis

**Data and model scaling.** We illustrate in Figure 4a and Figure 4b how scaling the data volume and model size influences downstream recommendation performance. Recall that, owing to the existence of a Bayes-optimal estimator, the performance of Markovian pre-training is theoretically upper-bounded. The overall performance increases progressively until reaching a peak, after which it either stabilizes or declines slightly. Although the number of tokens required to obtain the optimal performance varies across scenarios, typically 10B tokens (corresponding to 9,600,000 trajectories of length 1,024) with a hidden dimension size of 256 generally yield satisfactory results for all cases. In addition, the loss dynamics differ markedly from those of conventional recommendation objectives. Several sudden drops can be observed in the loss curve. This phenomenon has also been reported in prior work (Gopalani et al., 2024), signifying the emergence of specialized capabilities.

**Inference mechanisms.** In Section 3, we qualitatively explained why the capabilities acquired through Markovian pre-training can transfer to next-item recommendation. Here, we further support this argument by comparing the attention maps of SASRec+, Qwen2.5, and MPT. As shown in Figure 4c, SASRec+ and MPT exhibit clear layer-wise variations in attention patterns. Notably, MPT emphasizes recent interactions in the first layer, forming an approximately diagonal pattern consistent with the expected capability ❷. By contrast, although neither Qwen2.5 nor MPT is fine-tuned on recommendation data, Qwen2.5 shows little variation across Layers 1–3, suggesting limited ability to infer users' general preferences. This limitation results in inferior recommendation performance compared with SASRec+ and MPT. Overall, this case study demonstrates the advantage of Markovian pre-training over language pre-training for next-item recommendation.

### 4.3 Ablation Study

**Item embedding sources.** By design, MPT can accommodate item embeddings from diverse sources, including those obtained from pre-trained recommenders (*e.g.*, MF-BPR, SASRec) as well as semantic embeddings generated by existing encoders (*e.g.*, AlphaRec, T5-xl). As illustrated in Figure 5a, MPT demonstrates generally robust performance across a variety of embedding sources. Notably, semantic embeddings tend to be more favorable, owing to their superior generalizability.

**Sensitivity analysis.** Markovian pre-training involves two primary hyperparameters that affect the behaviors of the Markov chains: the state space size $|\mathcal{S}|$ and the concentration parameter $\alpha$ of the prior Dirichlet distribution. Both directly determine the degree of over-parameterization. Specifically, excessively large values result in an overly challenging task under the maximum input length of 1,024, whereas overly small values render the Markovian pre-training task too trivial to stimulate the

required capabilities. Consequently, as illustrated in Figure 5b and Figure 5c, optimal performance in most scenarios is achieved around the default setting of $|\mathcal{S}| = 30$ and $\alpha = 0.05$.

**Training paradigms.** It is of interest to examine the capacity of the employed MPT architecture when trained from scratch. As shown in Appendix C.6, MPT trained from scratch still lags substantially behind the Markovian pre-training and recommendation fine-tuning paradigm. This indicates that the performance gains of MPT primarily stem from the superiority of a novel training paradigm, rather than from the model architecture itself.

## 5 Related Work

**Next-item recommendation** aims to predict a user's subsequent interaction based on historical sequences, which are assumed to reflect underlying preferences. Given the sequential nature of this task, prior work has predominantly relied on sequential architectures, including RNN-based models (Hidasi et al., 2016) and Transformer-based approaches (Kang & McAuley, 2018). Moreover, modeling both short-term and long-term user interests has been a central theme in advancing model design (Yu et al., 2019; An et al., 2019; Xiao et al., 2024; Zheng et al., 2022). FPMC (Rendle et al., 2010) and STAMP (Liu et al., 2018) represent earlier efforts that leverage Markovian characteristics by explicitly emphasizing the last item from an architectural perspective. Our findings offer a complementary perspective: the model can be encouraged, through Markovian pre-training, to adaptively summarize sequential context while appropriately attending to the most recent interaction.

**Pre-training and fine-tuning** have become a standard paradigm across domains. Pre-training improves generalization by exposing models to diverse data distributions, enabling efficient transfer to downstream tasks. In recommendation, this paradigm promises to reduce retraining costs and facilitate rapid adaptation via lightweight fine-tuning. Prior studies have explored universal pre-trained recommenders, including ZESRec (Ding et al., 2021), UniSRec (Hou et al., 2022), and RecFormer (Li et al., 2023a). More recently, LLMs have been leveraged for recommendation due to their strong semantic and reasoning capabilities (Wang et al., 2025b). Pre-training in recommendation is also related to multi-task learning (Geng et al., 2022; Wang et al., 2023) and cross-domain recommendation (Tang et al., 2012; Wu et al., 2025). The former targets generalization across heterogeneous tasks, while the latter exploits shared information across domains. In contrast, our work focuses on adapting a pre-trained recommender to diverse next-item recommendation scenarios without requiring overlapping users or items.

**In-context learning (ICL)** is a notable emergent capability of language models (Brown et al., 2020), and substantial efforts (Xie et al., 2021; Min et al., 2022) have been devoted to explaining its underlying mechanisms, such as implicit Bayesian inference. Prior-Data Fitted Networks (PFNs) (Müller et al., 2022; Hollmann et al., 2025) demonstrate the effectiveness of ICL in tabular datasets. More recently, Lepage et al. (Lepage et al., 2025) showed that Transformers can perform next-state prediction through ICL rather than memorization. The feasibility of Markovian pre-training followed by recommendation fine-tuning is closely related to this phenomenon. We posit that Markov chain data can effectively stimulate the transferable capabilities required for sequential recommendation.

## 6 Conclusion and Future Work

We analyze the inference mechanisms of advanced sequential recommenders and examine whether Markovian pre-training can induce the required capabilities. To the best of our knowledge, this is the first work to pre-train a recommender entirely on synthetic data and achieve state-of-the-art performance with lightweight fine-tuning. Compared with conventional recommendation and language pre-training, Markovian pre-training offers clear advantages in both effectiveness and efficiency, suggesting a promising paradigm for universal recommendation.

Nevertheless, Markov chain data may not capture the full diversity of recommendation scenarios, motivating richer forms of synthetic data. For example, Yin et al. (Yin et al., 2024) proposed dataset regeneration to improve cross-architecture generalizability, while recent studies have leveraged LLMs to simulate user behavior and generate more controllable, high-quality interactions (Wang et al., 2025a; Liu et al., 2025b). Together, these advances point to a promising direction for recommendation simulation (Ie et al., 2019; Shi et al., 2019).

### AI USE STATEMENT

We used generative AI tools to polish the manuscript and identify related work on in-context learning. Beyond these uses, generative AI tools were not used for research ideation or execution. The synthetic Markov chains were sampled using Algorithm 1, without LLM involvement, and the theoretical limit of MPT follows directly from Dirichlet-multinomial conjugacy.

### REPRODUCIBILITY STATEMENT

Algorithm 1 details the synthetic data generation and pre-training procedure, while Appendix B provides the datasets, preprocessing steps, evaluation protocol, baseline configurations, and hyperparameters. Results for RecFormer, E4SRec, and Qwen2.5 are averaged over three independent runs due to their substantial computational costs; results for all other methods are averaged over five independent runs. Our implementation is available at `https://anonymous.4open.science/r/MPT`.

## REFERENCES

Mingxiao An, Fangzhao Wu, Chuhan Wu, Kun Zhang, Zheng Liu, and Xing Xie. Neural news recommendation with long-and short-term user representations. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 336–345, 2019.

James O Berger. *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media, 2013.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:1877–1901, 2020.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, pp. 1597–1607. PmLR, 2020.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 4171–4186, 2019.

Hao Ding, Yifei Ma, Anoop Deoras, Yuyang Wang, and Hao Wang. Zero-shot recommender systems. *arXiv preprint arXiv:2105.08318*, 2021.

Junchen Fu, Xuri Ge, Alexandros Karatzoglou, Ioannis Arapakis, Suzan Verberne, Joemon M Jose, and Zhaochun Ren. Differentiable semantic id for generative recommendation. *arXiv preprint arXiv:2601.19711*, 2026.

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *ACM Conference on Recommender Systems (RecSys)*, pp. 299–315, 2022.

Pulkit Gopalani, Ekdeep S Lubana, and Wei Hu. Abrupt learning in transformers: A case study on matrix completion. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:55053–55085, 2024.

F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TIIS)*, 5(4):1–19, 2015.

Yingzhi He, Xiaohao Liu, An Zhang, Yunshan Ma, and Tat-Seng Chua. Llm2rec: Large language models are powerful embedding models for sequential recommendation. In *ACM International Conference on Knowledge Discovery & Data Mining (KDD)*, pp. 896–907, 2025.

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. In *International Conference on Learning Representations (ICLR)*, 2016.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 01 2025. doi: 10.1038/s41586-024-08328-6. URL https://www.nature.com/articles/s41586-024-08328-6.

Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. Towards universal sequence representation learning for recommender systems. In *ACM International Conference on Knowledge Discovery & Data Mining (KDD)*, pp. 585–593, 2022.

Maggie Huan, Yuetai Li, Tuney Zheng, Xiaoyu Xu, Seungone Kim, Minxin Du, Radha Poovendran, Graham Neubig, and Xiang Yue. Does math reasoning improve general llm capabilities? understanding transferability of llm reasoning. *arXiv preprint arXiv:2507.00432*, 2025.

Eugene Ie, Chih-wei Hsu, Martin Mladenov, Vihan Jain, Sanmit Narvekar, Jing Wang, Rui Wu, and Craig Boutilier. Recsim: A configurable simulation platform for recommender systems. *arXiv preprint arXiv:1909.04847*, 2019.

Clark Mingxuan Ju, Leonardo Neves, Bhuvesh Kumar, Liam Collins, Tong Zhao, Yuwei Qiu, Qing Dou, Sohail Nizam, Sen Yang, and Neil Shah. Revisiting self-attention for cross-domain sequential recommendation. In *ACM International Conference on Knowledge Discovery & Data Mining (KDD)*, pp. 1094–1105, 2025.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *IEEE International Conference on Data Mining (ICDM)*, pp. 197–206. IEEE, 2018.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Anton Klenitskiy and Alexey Vasilev. Turning dross into gold loss: Is bert4rec really better than sasrec? In *ACM Conference on Recommender Systems (RecSys)*, pp. 1120–1125, 2023.

Ariel N Lee, Cole J Hunter, and Nataniel Ruiz. Platypus: Quick, cheap, and powerful refinement of llms. *arXiv preprint arXiv:2308.07317*, 2023.

Simon Lepage, Jeremie Mary, and David Picard. Markov chain estimation with in-context learning. *arXiv preprint arXiv:2508.03934*, 2025.

Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley. Text is all you need: Learning language representations for sequential recommendation. In *ACM International Conference on Knowledge Discovery & Data Mining (KDD)*, pp. 1258–1267, 2023a.

Xinhang Li, Chong Chen, Xiangyu Zhao, Yong Zhang, and Chunxiao Xing. E4srec: An elegant effective efficient extensible solution of large language models for sequential recommendation. *arXiv preprint arXiv:2312.02443*, 2023b.

Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang, and Xiangnan He. Llara: Large language-recommendation assistant. In *ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 1785–1795, 2024.

Jiayu Lin. On the dirichlet distribution. *Department of Mathematics and Statistics, Queens University*, 40, 2016.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Enze Liu, Bowen Zheng, Cheng Ling, Lantao Hu, Han Li, and Wayne Xin Zhao. Generative recommender with end-to-end learnable item tokenization. In *ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 729–739, 2025a.

Fei Liu, Xinyu Lin, Hanchao Yu, Mingyuan Wu, Jianyu Wang, Qiang Zhang, Zhuokai Zhao, Yinglong Xia, Yao Zhang, Weiwei Li, et al. Recoworld: Building simulated environments for agentic recommender systems. *arXiv preprint arXiv:2509.10397*, 2025b.

Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. Stamp: Short-term attention/memory priority model for session-based recommendation. In *ACM International Conference on Knowledge Discovery & Data Mining (KDD)*, pp. 1831–1839, 2018.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 11048–11064, 2022.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference. In *International Conference on Learning Representations*, 2022.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pp. 8748–8763. PmLR, 2021.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:10299–10315, 2023.

Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *ACM Web Conference (WWW)*, pp. 811–820, 2010.

Leheng Sheng, An Zhang, Yi Zhang, Yuxin Chen, Xiang Wang, and Tat-Seng Chua. Language representations can be what recommenders need: Findings and potentials. In *International Conference on Learning Representations (ICLR)*, 2025.

Jing-Cheng Shi, Yang Yu, Qing Da, Shi-Yong Chen, and An-Xiang Zeng. Virtual-taobao: Virtualizing real-world online retail environment for reinforcement learning. In *Conference on Artificial Intelligence (AAAI)*, volume 33, pp. 4902–4909, 2019.

Jie Tang, Sen Wu, Jimeng Sun, and Hang Su. Cross-domain collaboration recommendation. In *ACM International Conference on Knowledge Discovery & Data Mining (KDD)*, pp. 1285–1293, 2012.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017.

Qiang Wan, Ze Yang, Dawei Yang, Ying Fan, Xin Yan, Siyang Liu, Yicong Liu, Chenwei Zhang, Wei Xu, Jiahao Qin, et al. R3-vae: Reference vector-guided rating residual quantization vae for generative recommendation. *arXiv preprint arXiv:2604.11440*, 2026.

Lei Wang, Jingsen Zhang, Hao Yang, Zhi-Yuan Chen, Jiakai Tang, Zeyu Zhang, Xu Chen, Yankai Lin, Hao Sun, Ruihua Song, et al. User behavior simulation with large language model-based agents. *ACM Transactions on Information Systems (TOIS)*, 43(2):1–37, 2025a.

Minhao Wang, Yunhang He, Cong Xu, Zhangchi Zhu, and Wei Zhang. Beyond semantic understanding: Preserving collaborative frequency components in llm-based recommendation. *arXiv preprint arXiv:2508.10312*, 2025b.

Yuhao Wang, Ha Tsz Lam, Yi Wong, Ziru Liu, Xiangyu Zhao, Yichao Wang, Bo Chen, Huifeng Guo, and Ruiming Tang. Multi-task deep recommender systems: A survey. *arXiv preprint arXiv:2302.03525*, 2023.

Daniel Woolridge, Sean Wilner, and Madeleine Glick. Sequence or pseudo-sequence? an analysis of sequential recommendation datasets. In *Perspectives@ RecSys*, 2021.

Yiqing Wu, Ruobing Xie, Zhao Zhang, Xu Zhang, Fuzhen Zhuang, Leyu Lin, Zhanhui Kang, Zhulin An, and Yongjun Xu. Id-centric pre-training for recommendation. *ACM Transactions on Information Systems (TOIS)*, 43(5):1–29, 2025.

Yunpeng Xiao, Wanjing Zhao, Yuyang Huang, Tun Li, and Qian Li. A joint learning recommendation model for e-commerce platforms integrating long-term and short-term interests. *IEEE Transactions on Services Computing (TSC)*, 17(4):1326–1339, 2024.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 2021.

Cong Xu, Zhangchi Zhu, Mo Yu, Jun Wang, Jianyong Wang, and Wei Zhang. Are llm-based recommenders already the best? simple scaled cross-entropy unleashes the potential of traditional sequential recommenders. *arXiv preprint arXiv:2408.14238*, 2024.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2025.

Xiaoxin Ye, Chengkai Huang, Hongtao Huang, and Lina Yao. Beyond negative transfer: Disentangled preference-guided diffusion for cross-domain sequential recommendation. In *Conference on Artificial Intelligence (AAAI)*, 2026.

Mingjia Yin, Hao Wang, Wei Guo, Yong Liu, Suojuan Zhang, Sirui Zhao, Defu Lian, and Enhong Chen. Dataset regeneration for sequential recommendation. In *ACM International Conference on Knowledge Discovery & Data Mining (KDD)*, pp. 3954–3965, 2024.

Zeping Yu, Jianxun Lian, Ahmad Mahmoody, Gongshen Liu, and Xing Xie. Adaptive user modeling with long and short-term preferences for personalized recommendation. In *International Joint Conference on Artificial Intelligence (IJCAI)*, volume 7, pp. 4213–4219, 2019.

Zheng Yuan, Fajie Yuan, Yu Song, Youhua Li, Junchen Fu, Fei Yang, Yunzhu Pan, and Yongxin Ni. Where to go next for recommender systems? id-vs. modality-based recommender models revisited. In *ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 2639–2649, 2023.

Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Michael He, et al. Actions speak louder than words: Trillion-parameter sequential transducers for generative recommendations. In *International Conference on Machine Learning (ICML)*, 2024.

Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18123–18133, 2022.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019.

Buyun Zhang, Liang Luo, Yuxin Chen, Jade Nie, Xi Liu, Daifeng Guo, Yanli Zhao, Shen Li, Yuchen Hao, Yantao Yao, et al. Wukong: Towards a scaling law for large-scale recommendation. *arXiv preprint arXiv:2403.02545*, 2024.

Sheng Zhang, Maolin Wang, Wanyu Wang, Jingtong Gao, Xiangyu Zhao, Yu Yang, Xuetao Wei, Zitao Liu, and Tong Xu. Glint-ru: Gated lightweight intelligent recurrent units for sequential recommender systems. In *ACM International Conference on Knowledge Discovery & Data Mining (KDD)*, pp. 1948–1959, 2025.

Yu Zheng, Chen Gao, Jianxin Chang, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. Disentangling long and short-term interests for recommendation. In *ACM Web Conference (WWW)*, pp. 2256–2267, 2022.

Kun Zhou, Hui Yu, Wayne Xin Zhao, and Ji-Rong Wen. Filter-enhanced mlp is all you need for sequential recommendation. In *ACM Web Conference (WWW)*, pp. 2388–2399, 2022.

## Appendix Contents

**A Theoretical Limit of MPT** **14**

**B Experimental Details** **15**

B.1 Figure 1 . . . 15
B.2 Synthetic Markov Chain Data . . . 15
B.3 Datasets . . . 15
B.4 Evaluation Metrics . . . 16
B.5 Baselines . . . 16
B.6 Hyperparameters . . . 17

**C Additional Experimental Results** **17**

C.1 Computational Costs . . . 17
C.2 Exceptions of GRU4Rec . . . 18
C.3 Sequence Ordering . . . 18
C.4 'Non-Markovian' Scenarios . . . 19
C.5 Data Efficiency . . . 19
C.6 Training Paradigms . . . 19
C.7 Comparison with Generative Recommendation Models . . . 20
C.8 Attention Maps . . . 21

## A Theoretical Limit of MPT

A Markov chain can be regarded as a collection of $|\mathcal{S}|$ independent multinomial distributions $\{\text{Cat}(|\mathcal{S}|; \mathbf{p}_i)\}_{i=1}^{|\mathcal{S}|}$. In this study, each ground-truth transition probability vector $\mathbf{p}_i$ is drawn from a Dirichlet prior $\text{Dir}(\boldsymbol{\alpha})$. Because the Dirichlet distribution is conjugate to the multinomial distribution (Lin, 2016), the posterior distribution of the transition probabilities remains Dirichlet. Specifically,

$$\mathbf{p}_i \mid [s_1, s_2, \ldots, s_T] \sim \text{Dir}(c_{i,1} + \alpha_1,\ c_{i,2} + \alpha_2,\ \ldots,\ c_{i,|\mathcal{S}|} + \alpha_{|\mathcal{S}|}),$$

where $c_{i,j}$ counts the number of transitions from state $i$ to state $j$ along the entire trajectory $[s_1, s_2, \ldots, s_T]$. Hence, for an arbitrary trajectory, we can derive the Bayes-optimal estimator (Berger, 2013)

$$\hat{p}_{ij} = \mathbb{E}\left[p_{ij} | [s_1, s_2, \ldots, s_T]\right] = \frac{c_{i,j} + \alpha_j}{\sum_{j=1}^{|\mathcal{S}|}(c_{i,j} + \alpha_j)}.$$

This optimal estimator provides a well-defined theoretical limit for Markovian pre-training under the assumed prior. We also empirically observe in Figure 4a marginal performance gains as this limit is approached.

# B Experimental Details

## B.1 Figure 1

We report the results in Figure 1 based on the official setup. All results are averaged over five independent runs; consequently, models with partially shuffled sequences may occasionally outperform those with chronologically ordered sequences due to random variation. The experiments involving filtering strategies, training objectives, and embedding sizes are conducted on the Beauty dataset. Note that, in Figure 1, the models exhibit inferior performance compared to those in Table 1. This discrepancy arises because, in the latter, the embedding dimensionality is fixed at 256 for all models to ensure a fair comparison.

## B.2 Synthetic Markov Chain Data

**Algorithm 1:** Pre-training of MPT on Synthetic Markov Chains

**Input :** Number of training steps $N_{\text{steps}}$, batch size $B$, maximum trajectory length $T$, Dirichlet prior $\text{Dir}(\boldsymbol{\alpha})$ with $\alpha$ shared across components, state space size $|\mathcal{S}|$

**Output :** Trained MPT parameters $\Theta$

**for** *step* $= 1$ *to* $N_{steps}$ **do**
  Initialize empty batch $\mathcal{B} = \emptyset$;
  **for** $b = 1$ *to* $B$ **do**
    Sample transition matrix $\mathbf{P} \in [0,1]^{|\mathcal{S}|\times|\mathcal{S}|} \sim \text{Dir}(\boldsymbol{\alpha})$ ; `// each row sums to 1`
    Generate trajectory $\{s_t\}_{t=2}^{T}$ by iteratively sampling $s_{t+1} \sim \mathbf{P}(\cdot|s_t)$;
    Store trajectory $\{s_t\}_{t=1}^{T}$ in $\mathcal{B}$;
  **end**
  Sample random orthogonal state representations for each trajectory;
  Compute and update according to the $\mathcal{L}_{\text{NSP}}$ loss.
**end**

We detail the Markovian pre-training pipeline in Algorithm 1. The random orthogonal state representations are obtained by first sampling Gaussian noise and then applying orthogonal decomposition.

## B.3 Datasets

We consider five public datasets originating from three distinct platforms. The first three, Beauty, Toys, and Sports, are derived from the Amazon Reviews data, each processed using a 5-core filter that removes users and items with fewer than five interactions. Yelp is a real-world dataset comprising user reviews and check-ins; in this work, we use the 2018 version with a 10-core filter. Online Retail contains transaction records from a UK-based e-commerce platform, and we directly adopt the preprocessed version provided by (Hou et al., 2022). Note that the Tmall dataset is not considered here because it lacks textual features required by most pre-trained baselines. Following prior work (Kang & McAuley, 2018; Hou et al., 2022), we adopt a *leave-one-out* fashion for dataset partitioning, where the final interaction is reserved for testing and the second-to-last interaction is used for validation. The dataset statistics are presented in Table 2.

Table 2: Dataset statistics

| Dataset | #Users | #Items | #Interactions | Avg. Len. |
|---|---|---|---|---|
| Beauty | 22,363 | 12,101 | 198,502 | 8.88 |
| Toys | 19,412 | 11,924 | 167,597 | 8.63 |
| Sports | 35,598 | 18,357 | 296,337 | 8.32 |
| Yelp | 77,277 | 45,638 | 2,103,896 | 27.23 |
| Online Retail | 16,520 | 3,469 | 519,906 | 31.47 |

### B.4 Evaluation Metrics

The predicted scores over *all items* are first ranked in descending order to yield a top-$N$ user-wise candidate list. We employ the widely used HR@$N$ (Hit Rate) and NDCG@$N$ (Normalized Discounted Cumulative Gain) to assess the recommendation quality. The former measures the rate of successful hits among the top-$N$ recommended candidates, while the latter further accounts for positional relevance by assigning higher weights to correctly predicted items that appear earlier in the ranking.

### B.5 Baselines

Here, we primarily introduce the involved baselines and explain how their results in Table 1 were obtained.

- **FPMC** (Rendle et al., 2010) models user interaction sequences as a form of personalized Markov chains and aims to estimate user-specific transition matrices via tensor factorization. Accordingly, FPMC can be regarded as a baseline that leverages Markovian characteristics from an architectural design perspective.
- **GRU4Rec** (Hidasi et al., 2016) was the first to apply an RNN architecture to session-based recommendation. It consists of a GRU (Cho et al., 2014) kernel followed by a linear output projection layer.
- **SASRec** (Kang & McAuley, 2018) is a classic Transformer-based recommender that employs a causal attention mechanism and learnable positional encoding. Following the official implementation, a binary cross-entropy loss is used by default.
- **FMLPRec** (Zhou et al., 2022) is an MLP-based recommender capable of achieving competitive performance through the incorporation of learnable filters.
- **HSTU** (Zhai et al., 2024) is known for its generative recommendation mechanism. Here, we employ the simplified variant (with learnable ID embeddings only) provided by the official implementation.
- **SASRec+** (Klenitskiy & Vasilev, 2023; Xu et al., 2024) differs from SASRec in its training objective. The cross-entropy loss instead of point-wise objectives is employed for superior performance.
- **GLINT-RU** (Zhang et al., 2025) is a lightweight sequential recommender that replaces heavy Transformer architectures with a single-layer dense selective GRU, enabling efficient modeling of user behavior while preserving high-quality semantic and positional representations.
- **UniSRec** (Hou et al., 2022) proposed a well-known universal recommender pre-trained across five categories from Amazon Reviews: 'Grocery and Gourmet Food', 'Home and Kitchen', 'CDs and Vinyl', 'Kindle Store', and 'Movies and TV'. Consistent with MPT, we fine-tune only its MoE-enhanced adaptor for downstream tasks.
- **RecFormer** (Li et al., 2023a) performs next-item recommendation based on item sentences. Unlike pure language pre-training, RecFormer incorporates recommendation-specific information in the pre-training stage. It is pre-trained across seven categories from Amazon Reviews: 'Automotive', 'Cell Phones and Accessories', 'Clothing, Shoes and Jewelry', 'Electronics', 'Grocery and Gourmet Food', 'Home and Kitchen', and 'Movies and TV'.
- **E4SRec** (Li et al., 2023b) connects pre-trained item ID embeddings directly to an instruction-tuned LLM (Platypus2-7B (Lee et al., 2023) in this study) through a linear adaptor and LoRA-based fine-tuning.

Table 3: Hyperparameters of MPT during different stages.

| | | Beauty | Toys | Sports | Yelp | Online Retail |
|---|---|---|---|---|---|---|
| Pre-training | Learning Rate | 3e-4 | 3e-4 | 3e-4 | 3e-4 | 3e-4 |
| | Weight Decay | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| | Max Len. | 1024 | 1024 | 1024 | 1024 | 1024 |
| | $\alpha$ | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 |
| | $\lvert\mathcal{S}\rvert$ | 30 | 30 | 30 | 30 | 30 |
| | #Layers | 4 | 4 | 4 | 4 | 4 |
| | #Heads | 2 | 2 | 2 | 2 | 2 |
| Adaptor | Learning Rate | 1e-3 | 1e-3 | 1e-3 | 1e-3 | 1e-3 |
| | Weight Decay | 0.1 | 0.1 | 0.1 | 0.1 | 1 |
| | Max Len. | 50 | 50 | 50 | 50 | 50 |
| | Dropout Rate | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |
| | Temperature | 0.07 | 0.07 | 0.07 | 0.07 | 0.07 |
| +LoRA | LoRA_Rank | 16 | 16 | 16 | 16 | 16 |
| | LoRA_Alpha | 16 | 16 | 16 | 16 | 16 |
| | LoRA_Dropout | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |

- **LLM2Rec** (He et al., 2025) represents an alternative protocol for leveraging the rich knowledge in LLMs by generating item embeddings for downstream tasks. To ensure that these embeddings capture both semantic and collaborative information, collaborative supervised fine-tuning and item-level embedding modeling are conducted on LLMs. We adopt the recommended SASRec as the downstream sequential model for evaluation.
- **Qwen2.5-7B** (Yang et al., 2025) is one of the most powerful open-source LLMs. We fine-tune it in the same manner as MPT to examine the differences between Markovian pre-training and language pre-training.
- **AlphaRec** (Sheng et al., 2025) explores the potential of deriving item representations from causal LLMs. In this study, we focus on the specialized item embedding source provided by AlphaRec (based on Qwen2.5-7B), for the ablation analysis in Figure 5a.

For traditional sequential recommendation methods, we carefully tune key hyperparameters, including the learning rate, weight decay, and dropout rate, to ensure competitive performance with an embedding dimension of 256. For other pre-trained methods, we follow the fine-tuning protocols specified in their respective original papers.

### B.6 Hyperparameters

We adopt a Llama-style architecture with four hidden layers as the backbone of MPT. The model is pre-trained using the AdamW (Loshchilov & Hutter, 2019) optimizer with a fixed learning rate of 3e-4 and a weight decay of 0.1. By default, convergence is achieved after learning approximately 10B tokens, corresponding to 9,600,000 trajectories of length 1,024. During the recommendation fine-tuning stage, all datasets use a unified learning rate of 0.001 and an attention dropout rate of 0.2. Except for the Online Retail dataset, which uses a slightly larger weight decay of 1 to prevent overfitting, all other datasets employ a weight decay of 0.1. For a fair comparison, we fix the embedding size at 256 for MPT as well as for all traditional sequential models. Table 3 summarizes the key hyperparameters.

## C Additional Experimental Results

### C.1 Computational Costs

As shown in Table 4 and Table 5, MPT achieves superior efficiency in both fine-tuning and inference. Since a four-block MPT is sufficient to reach the theoretical limit, its computational cost during fine-tuning and inference is comparable to that of standard SASRec models. In contrast, both

Table 4: Training/fine-tuning time. The wall time (in seconds) is measured on an Intel(R) Xeon(R) Gold 6133 CPU paired with a single NVIDIA A800 GPU.

| | Beauty | Toys | Sports | Yelp | Online Retail |
|---|---|---|---|---|---|
| GRU4Rec | 2073.58 | 1666.63 | 2823.03 | 15966.44 | 2809.25 |
| SASRec | 387.60 | 348.05 | 582.82 | 1838.89 | 290.04 |
| FMLPRec | 8358.79 | 6163.90 | 16476.14 | 151648.99 | 33337.11 |
| HSTU | 1456.16 | 1653.19 | 2192.82 | 7145.66 | 1654.16 |
| UniSRec | 5940.00 | 8160.00 | 11400.00 | 27042.00 | 3119.00 |
| E4SRec | 45349.00 | 39397.00 | 35417.00 | 463999.00 | 98736.00 |
| Qwen2.5 | 198756.00 | 102500.00 | 115041.00 | 521734.00 | 64330.00 |
| MPT | 615.35 | 1103.68 | 1544.00 | 4641.52 | 705.98 |

Table 5: Inference time.

| | Beauty | Toys | Sports | Yelp | Online Retail |
|---|---|---|---|---|---|
| GRU4Rec | 1.09 | 1.11 | 1.55 | 3.82 | 1.10 |
| SASRec | 1.20 | 1.04 | 1.69 | 3.62 | 0.90 |
| FMLPRec | 1.88 | 1.60 | 2.80 | 6.86 | 1.49 |
| HSTU | 1.92 | 1.63 | 2.97 | 7.71 | 1.95 |
| UniSRec | 1.85 | 1.59 | 2.54 | 4.72 | 2.09 |
| E4SRec | 251.06 | 274.17 | 397.60 | 1138.52 | 186.17 |
| Qwen2.5 | 299.56 | 158.53 | 332.80 | 759.32 | 151.74 |
| MPT | 1.55 | 1.35 | 2.24 | 5.18 | 1.31 |

recommendation-oriented pre-trained models (*e.g.*, UniSRec) and language-based models (*e.g.*, E4SRec and Qwen2.5) incur substantially higher computational overhead due to their more complex architectures. It is worth noting that MPT requires an additional 18,838 seconds for one-time pre-training with a single A800 GPU; however, this cost remains significantly lower than that of LLM pre-training, which is typically infeasible without the resources of large-scale organizations.

### C.2 Exceptions of GRU4Rec

Table 6: GRU4Rec with skip connections exhibits the same ①/② $\gtrapprox$ ③ phenomenon.

| | Ordering | Beauty | Yelp | Online Retail |
|---|---|---|---|---|
| GRU4Rec | ① $[v_1, v_2, \ldots, v_t] \rightarrow v_{t+1}$ | 0.0223 | 0.0111 | 0.0527 |
| +skip connection | ① $[v_1, v_2, \ldots, v_t] \rightarrow v_{t+1}$ | 0.0279 | 0.0120 | 0.0433 |
| +skip connection | ② $[\{v_1, v_2, \ldots\}, v_t] \rightarrow v_{t+1}$ | 0.0279 | 0.0119 | 0.0433 |
| +skip connection | ③ $\{v_1, v_2, \ldots, v_t\} \rightarrow v_{t+1}$ | 0.0168 | 0.0094 | 0.0429 |

The original GRU4Rec shown in Figure 1 violates the Markovian characteristics. This may be attributed to the fact that GRU4Rec captures sequential information through a one-by-one recurrent mechanism. However, this design inherently biases the model toward emphasizing the historical interactions, compared with Transformer-based architectures. As shown in Table 6, when a skip connection is introduced from the input to the output to enable direct utilization of the last item, GRU4Rec again exhibits the ①/② $\gtrapprox$ ③ phenomenon. Moreover, this simple modification also improves GRU4Rec performance on both Beauty and Yelp, further demonstrating the transferability of the Markovian characteristics.

Table 7: Further comparisons of sequence ordering: ④ $[v_t] \rightarrow v_{t+1}$: retain only the most recent interaction; ⑤ $[v_1, \ldots, v_{t-1}] \rightarrow v_{t+1}$: remove the most recent interaction; ⑥ $[\{v_*\}, v_{t-1}, v_*] \rightarrow v_{t+1}$: fix the second-to-last item; ⑦ $[\{v_*\}, v_{t-2}, v_*, v_*] \rightarrow v_{t+1}$: fix the third-to-last item. NDCG@10 results are reported as 'mean$_{(\text{std})}$'.

| | Beauty | Yelp | Online Retail |
|---|---|---|---|
| ① $[v_1, v_2, \ldots, v_t] \rightarrow v_{t+1}$ | $0.0372_{(.0004)}$ | $0.0144_{(.0005)}$ | $0.0603_{(.0011)}$ |
| ④ $[v_t] \rightarrow v_{t+1}$ | $0.0349_{(.0010)}$ | $0.0087_{(.0007)}$ | $0.0392_{(.0005)}$ |
| ⑤ $[v_1, \ldots, v_{t-1}] \rightarrow v_{t+1}$ | $0.0250_{(.0005)}$ | $0.0126_{(.0007)}$ | $0.0587_{(.0011)}$ |
| ⑥ $[\{v_*\}, v_{t-1}, v_*] \rightarrow v_{t+1}$ | $0.0243_{(.0006)}$ | $0.0123_{(.0006)}$ | $0.0594_{(.0008)}$ |
| ⑦ $[\{v_*\}, v_{t-2}, v_*, v_*] \rightarrow v_{t+1}$ | $0.0248_{(.0004)}$ | $0.0124_{(.0005)}$ | $0.0592_{(.0009)}$ |

Table 8: Examination of the 'Markovian' characteristic on MovieLens-1M, whose recorded interaction order may be influenced by platform-specific data-collection mechanisms. The NDCG@10 results are reported as 'mean$_{(\text{std})}$'.

| | GRU4Rec | SASRec | HSTU |
|---|---|---|---|
| ① $[v_1, v_2, \ldots, v_t] \rightarrow v_{t+1}$ | $0.0913_{(.0019)}$ | $0.1143_{(.0019)}$ | $0.1575_{(.0024)}$ |
| ② $[\{v_1, v_2, \ldots\}, v_t] \rightarrow v_{t+1}$ | $0.0415_{(.0018)}$ | $0.0968_{(.0023)}$ | $0.1399_{(.0042)}$ |
| ③ $\{v_1, v_2, \ldots, v_t\} \rightarrow v_{t+1}$ | $0.0234_{(.0005)}$ | $0.0375_{(.0011)}$ | $0.0364_{(.0008)}$ |

### C.3 Sequence Ordering

We have empirically demonstrated the 'Markovian' characteristics in Section 2. A potential alternative explanation for the observed relationships among (①, ②, ③) is that the most recent interaction $v_t$ entirely dominates the prediction already. To examine this hypothesis, we further report in Table 7 the NDCG@10 performance when only the latest interaction is retained. A comparison between ① and ④ shows that relying solely on the most recent interaction is insufficient. Moreover, the effect of the latest interaction cannot be substituted by the second-to-last (⑥) or third-to-last (⑦) interactions. Therefore, Eq. equation 2 cannot be further simplified to $\mathbb{P}(v_{t+1} \mid v_t, v_{t-1}, \ldots, v_1; \Theta) \approx \mathbb{P}(v_{t+1} \mid v_t; \Theta)$. Both the most recent interaction $v_t$ and the non-sequential context $\{v_1, \ldots, v_{t-1}\}$ are essential.

### C.4 'Non-Markovian' Scenarios

We do not intend to claim that the observed 'Markovian' characteristics necessarily generalize to all recommendation scenarios. Indeed, the results reported in Table 8 are consistent with the conventional understanding of sequential recommendation, *i.e.*, ① > ② > ③. However, we emphasize that this pattern may largely stem from the pseudo-sequential nature of MovieLens. Both Harper & Konstan (2015) and Woolridge et al. (2021) point out that the sequential structure in MovieLens is substantially shaped by the underlying recommendation and data-collection mechanisms, rather than faithfully reflecting users' actual watch–then–rate behavior. In practice, users may rate multiple movies within a short period, such that the resulting interaction order does not necessarily correspond to their true consumption sequence. Consequently, datasets with such platform-driven or procedurally induced sequences may be inherently less suitable for next-item recommendation, where the temporal ordering of interactions is assumed to carry meaningful behavioral information. Similar concerns may also arise in other domains, such as music and short-form video recommendation, where interaction sequences can be strongly influenced by non-user-driven presentation orders.

### C.5 Data Efficiency

Pre-trained models are known for their data efficiency, achieving comparable performance with significantly less data. As shown in Figure 6, MPT also exhibits this property. However, it is important to emphasize that, in contrast to conventional pre-trained approaches, these benefits stem from transferable capabilities rather than from transferred knowledge.

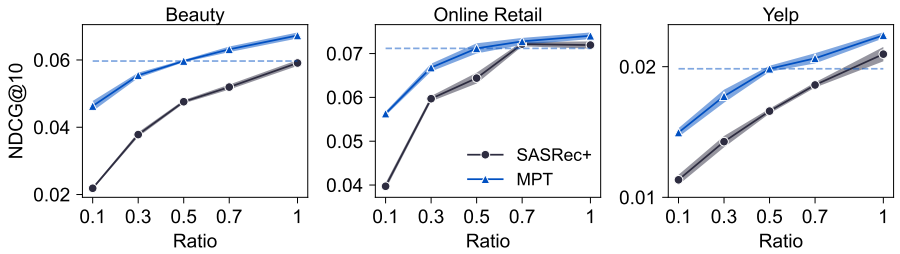



Figure 6: How recommendation performance evolves as an increasing number of user sequences are learnt during training. The dashed line indicates the performance of MPT trained with only half of the data, which is nearly comparable to that of SASRec+ trained on the full dataset.

### C.6 Training Paradigms

Table 9: Comparison of MPT recommendation performance (NDCG@10) under the 'Training from scratch' and 'Pre-training & Fine-tuning' paradigms.

| MPT | Beauty | Toys | Sports | Yelp | Retail |
|---|---|---|---|---|---|
| Training from scratch | $0.0623_{(.0006)}$ | $0.0664_{(.0007)}$ | $0.0336_{(.0005)}$ | $0.0221_{(.0003)}$ | $0.0679_{(.0008)}$ |
| Pre-training & Fine-tuning | $0.0673_{(.0008)}$ | $0.0705_{(.0007)}$ | $0.0373_{(.0004)}$ | $0.0224_{(.0002)}$ | $0.0741_{(.0006)}$ |

We compare, in Table 9, the recommendation performance of MPT under random initialization (*i.e.*, training from scratch) versus initialization via Markovian pre-training. This comparison directly validates the effectiveness of the pre-training stage introduced in this work. Notably, under both paradigms, MPT takes the same semantic embeddings as input and differs only in initialization. As expected, Markovian pre-training yields significant performance improvements in most scenarios. Yelp constitutes an exception, which can be attributed to the greater complexity of its underlying inference patterns that are not fully captured by Markovian characteristics.

### C.7 Comparison with Generative Recommendation Models

Although MPT is inherently a dense retrieval method, we additionally compare it with generative recommendation models, given the considerable attention they have recently received in the recommendation community (Deng et al., 2025). Following the seminal work TIGER (Rajput et al., 2023), generative recommendation typically consists of an item tokenization stage followed by an autoregressive recommendation stage. We reproduce all generative recommendation baselines in Table 10 using

Table 10: Comparison with generative recommendation models.

| | | TIGER | ETEGRec | DIGER | R3-VAE | MPT |
|---|---|---|---|---|---|---|
| Beauty | Recall@10 | $0.0669_{(.0019)}$ | $0.0600_{(.0019)}$ | $0.0664_{(.0055)}$ | $0.0710_{(.0029)}$ | $0.1118_{(.0014)}$ |
| | Recall@20 | $0.0914_{(.0018)}$ | $0.0831_{(.0022)}$ | $0.0879_{(.0082)}$ | $0.0963_{(.0037)}$ | $0.1531_{(.0010)}$ |
| | NDCG@10 | $0.0371_{(.0008)}$ | $0.0325_{(.0011)}$ | $0.0365_{(.0028)}$ | $0.0390_{(.0017)}$ | $0.0673_{(.0008)}$ |
| | NDCG@20 | $0.0434_{(.0007)}$ | $0.0384_{(.0011)}$ | $0.0420_{(.0035)}$ | $0.0454_{(.0019)}$ | $0.0777_{(.0007)}$ |
| Toys | Recall@10 | $0.0612_{(.0015)}$ | $0.0506_{(.0024)}$ | $0.0587_{(.0042)}$ | $0.0661_{(.0019)}$ | $0.1202_{(.0020)}$ |
| | Recall@20 | $0.0834_{(.0028)}$ | $0.0671_{(.0037)}$ | $0.0767_{(.0066)}$ | $0.0894_{(.0027)}$ | $0.1617_{(.0018)}$ |
| | NDCG@10 | $0.0342_{(.0008)}$ | $0.0284_{(.0013)}$ | $0.0326_{(.0021)}$ | $0.0374_{(.0012)}$ | $0.0705_{(.0007)}$ |
| | NDCG@20 | $0.0398_{(.0010)}$ | $0.0326_{(.0016)}$ | $0.0373_{(.0027)}$ | $0.0434_{(.0012)}$ | $0.0809_{(.0006)}$ |
| Sports | Recall@10 | $0.0391_{(.0041)}$ | $0.0395_{(.0017)}$ | $0.0378_{(.0014)}$ | $0.0430_{(.0015)}$ | $0.0651_{(.0007)}$ |
| | Recall@20 | $0.0557_{(.0052)}$ | $0.0550_{(.0016)}$ | $0.0513_{(.0014)}$ | $0.0603_{(.0017)}$ | $0.0924_{(.0006)}$ |
| | NDCG@10 | $0.0206_{(.0023)}$ | $0.0214_{(.0009)}$ | $0.0207_{(.0010)}$ | $0.0229_{(.0008)}$ | $0.0373_{(.0004)}$ |
| | NDCG@20 | $0.0248_{(.0026)}$ | $0.0254_{(.0009)}$ | $0.0241_{(.0010)}$ | $0.0273_{(.0008)}$ | $0.0442_{(.0004)}$ |

the same semantic item embeddings as MPT. As shown in Table 10, both TIGER and its follow-up methods, such as ETEGRec (Liu et al., 2025a), DIGER (Fu et al., 2026), and R3-VAE (Wan et al., 2026), substantially underperform dense retrieval methods. Nevertheless, generative recommendation remains a promising research direction, particularly due to its potential for end-to-end optimization and effective scaling.

### C.8 Attention Maps

We illustrate more attention maps of various user interaction sequences over the Beauty dataset below.

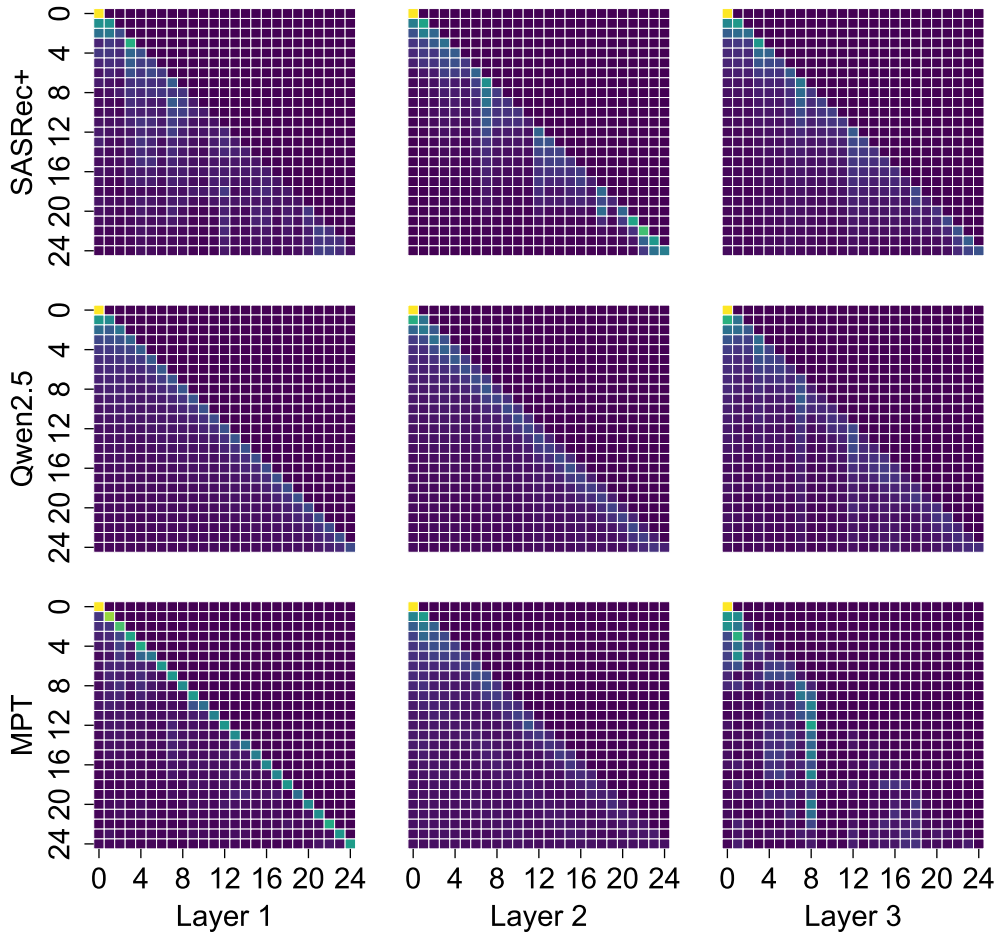



Figure 7: Attention maps for the interaction sequence of user ‘A1NOV41485TE0D’.

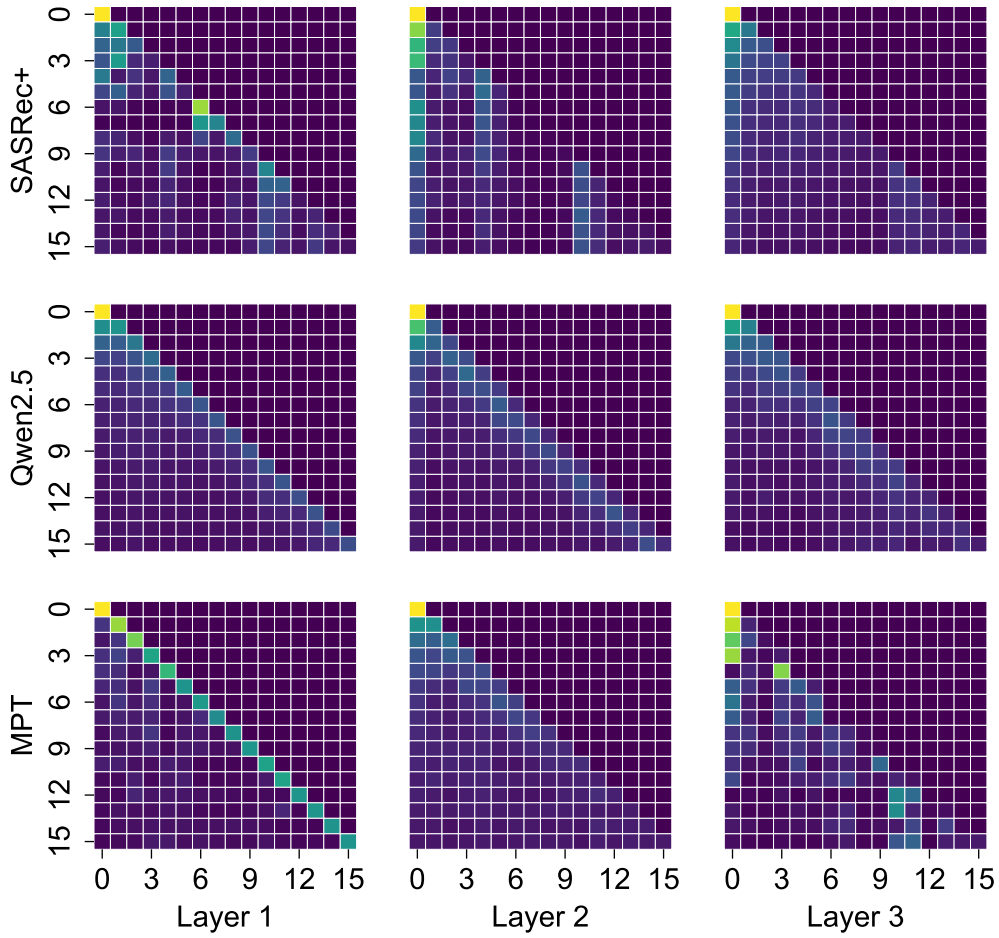



Figure 8: Attention maps for the interaction sequence of user ‘A1WA6BR7VO71QQ’.

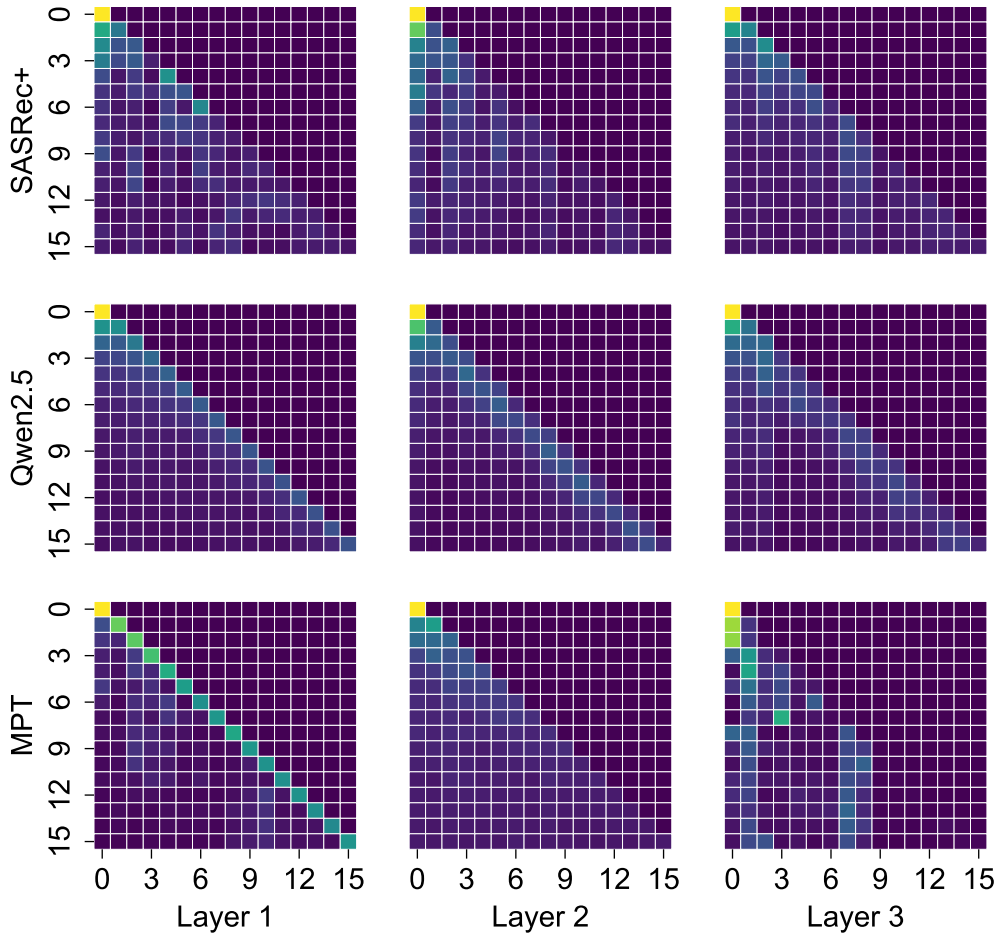



Figure 9: Attention maps for the interaction sequence of user ‘A2GT1S2L7YX426’.