# DREAM SPARSE, CONNECT DENSE: A LIGHTWEIGHT GRAPH PROTOTYPE DREAMING NETWORK FOR COLD-START SEQUENTIAL RECOMMENDATION

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

## ABSTRACT

Sleep replay and dreaming in the human brain reactivate sparse daytime experiences and recombine them into structured neural trajectories. This process consolidates memories and facilitates future behavior anticipation. In sequential recommendation, data sparsity and cold-start issues are equally prevalent. Yet existing methods either rely on costly auxiliary information to compensate for sparse interactions, or employ sophisticated architectures that incur heavy computational overhead. Drawing an analogy to the human dreaming mechanism, we propose a **Light**weight **G**raph **P**rototype **D**reaming network (LightGPD) that translates the dreaming process into a structured data augmentation framework to enhance sequential recommenders in cold-start scenarios. Specifically, we first introduce a prototype learning module that discovers abstract interest prototypes from item embeddings via subspace alignment, providing the fundamental "memory traces" for subsequent augmentation. We then devise a dreaming module that augments sparse interactions through four sequential phases: interaction replay, abstraction, hypothesis generation and verification, and dynamic graph reconstruction. This module selectively reinforces prototype-aligned interactions and discovers new connections based on learned interest patterns, enriching the sparse training signal without requiring auxiliary modalities or user attributes. Extensive experiments on three real-world datasets demonstrate that LightGPD outperforms nine state-of-the-art baselines in both accuracy and efficiency. The code and datasets are available at `https://anonymous.4open.science/r/LightGPD`.

## 1 INTRODUCTION

In cognitive neuroscience, the dreaming process is recognized as an essential mechanism for memory consolidation, where fragmented daily experiences are reactivated and reorganized into structured neural trajectories Maboudi et al. (2024). This mechanism strengthens representations for future behavior anticipation Chang et al. (2025). These findings suggest that structured reorganization of sparse signals may yield robust representations, motivating us to explore an analogous computational mechanism for artificial neural networks. In this work, we attempt to design such a mechanism and apply it to sequential recommendation scenarios Zhang et al. (2025b).

Sequential recommender systems (SRs) provide personalized recommendations based on users' historical behavior sequences, and have been widely adopted across e-commerce Zhang et al. (2025a), multi-media streaming Ye et al. (2025), and online services Ma et al. (2024). Most SRs assume sufficient interaction data are available for model training Zhao et al. (2025). However, new users or items constantly emerge, leading to the cold-start problem where historical interactions are extremely sparse or absent Fang et al. (2025); He et al. (2025). Existing CSSR methods typically focus on content-based methods Pembek et al. (2025), social-enhanced models Du et al. (2022), meta-learning frameworks Liu et al. (2025a), and graph-based networks Lei et al. (2025). By augmenting the user-item sequential graph Zhang et al. (2025b) with auxiliary features, graph-based methods have achieved remarkable performance Li et al. (2022), as it allows the model to infer preferences from richer, interconnected entities Liu et al. (2025a); Han et al. (2025). Despite these efforts, existing methods still face two significant challenges.



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: An illustration of the cold-start sequential recommendation with auxiliary information.

**(1) Heavy reliance on auxiliary information.** Most existing studies require access to auxiliary modal data or user-side features to achieve reasonable performance Lei et al. (2025); Liang et al. (2025); Xu et al. (2024) (as shown in Figure 1). For instance, content-based methods utilize item descriptions or images Pembek et al. (2025), and attribute-based models leverage additionally inputted user features Anjiri et al. (2025). Such dependencies not only increase data collection costs but also limit applicability in scenarios where auxiliary data is unavailable or incomplete.

**(2) High computational complexity.** The incorporation of auxiliary information, coupled with sophisticated network architectures (e.g., multi-layer transformers Zhao et al. (2025) or memory-augmented networks He et al. (2025); Han et al. (2025)), leads to high computational overhead and large parameter scales. It prolongs training and inference time and hinders deployment on resource-constrained edge devices. Hence, developing a lightweight yet effective model remains an open and urgent challenge.

To this end, we propose a **Light**weight **G**raph **P**rototype **D**reaming network (LightGPD) that reinterprets the dreaming mechanism as a structured data augmentation strategy for cold-start sequential recommendation. Specifically, we first introduce a prototype learning module that discovers abstract interest prototypes from item embeddings via a subspace alignment strategy. These prototypes serve as fundamental memory traces, capturing latent preference patterns that provide the basis for subsequent augmentation. This module calculates subspace affinities between items and prototype bases, and utilizes an asymmetric linear attention mechanism for prototype assignment, effectively clustering items with similar latent preferences. Then, we devise a dreaming module that augments sparse interactions through four sequential phases: interaction replay, abstraction, hypothesis generation and verification, and dynamic graph reconstruction. In interaction replay, important interactions are selected based on temporal recency, prototype relevance, and relative frequency. In abstraction, concrete items are converted to abstract interest distributions. In hypothesis generation and verification, hypothetical interest transitions are created and tested. In dynamic graph reconstruction, the graph structure is refined by preserving top-$T$ prototype-based edges for each user, with a residual update mechanism to prevent structural drift. Such a dreaming mechanism enables progressive graph refinement by performing additional graph convolutional layers on the reconstructed graph.

The main contributions of this work are summarized as follows:

- We propose LightGPD, a lightweight sequential recommendation framework that effectively addresses the cold-start problem without relying on auxiliary information.
- We design a prototype learning module that captures abstract interest concepts from sparse interactions, enabling robust user preference modeling.
- We introduce a dreaming module that performs structured data augmentation through prototype-guided replay and dynamic graph reconstruction, enhancing model generalization in data-sparse scenarios.
- We conduct extensive experiments on three datasets. Experimental results demonstrate that LightGPD outperforms state-of-the-art methods in both accuracy and efficiency, even with a deliberately simple base architecture.

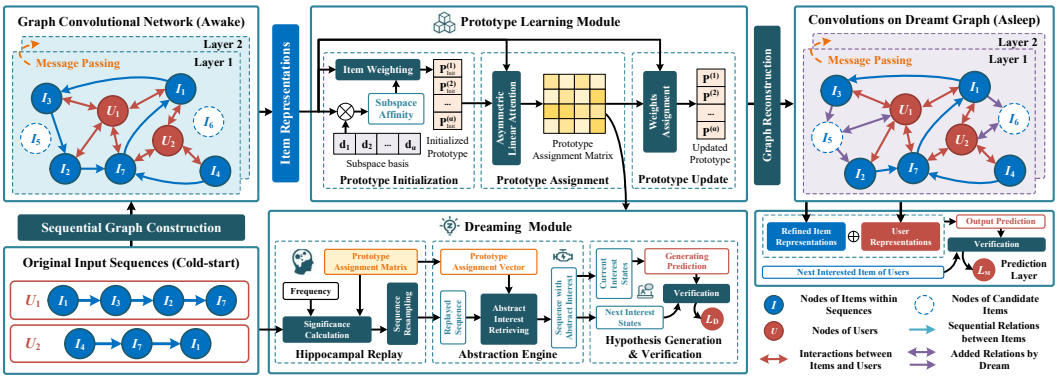



Figure 2: Framework of LightGPD, where $U_1$ and $U_2$ denote two different users, and $\{I_1, I_2, \ldots, I_7\}$ are the historical interactions that compose the sequences for the users.

## 2 Related Work

### 2.1 Sequential Recommendation

Sequential recommender systems (SRs) predict users' next behaviors from their historical interaction sequences Zhang et al. (2025b); Ye et al. (2025). Early methods employed Markov Chains Cai et al. (2017) and RNN/CNN-based models Xu et al. (2019); Yan et al. (2019) to capture sequential patterns, but struggled with long-range dependencies He & McAuley (2016). Subsequently, Transformer-based SRs Wu et al. (2020); Damak et al. (2022); Kang & McAuley (2018); Zhang et al. (2023) achieved strong performance via self-attention mechanisms. Then, Graph Neural Networks (GNNs) have been widely adopted to model structural relationships in sequences Peng et al. (2025); Sang et al. (2025); Zhang et al. (2025b); Liu et al. (2025b); Zhao et al. (2025). Despite their success, most sequential models assume sufficient interaction data Liang et al. (2025), leading to performance degradation in cold-start scenarios Pembek et al. (2025).

### 2.2 Cold-start Recommendation

Cold-start recommendation aims to provide accurate suggestions for new users or items with limited interaction data Anjiri et al. (2025); Du et al. (2022); He et al. (2025); Lei et al. (2025). Recent approaches include content-based methods Pembek et al. (2025); Kim et al. (2024), collaborative filtering with side information Du et al. (2022), and meta-learning frameworks Kim et al. (2023); He et al. (2025). Graph-based methods have also shown promise by constructing local graphs to infer cold-start preferences Li et al. (2022) or multi-view item graphs to explore item correlations Liu et al. (2025a). More recently, LLM-enhanced graph modules have been proposed to strengthen node representations Lei et al. (2025). However, these methods rely heavily on auxiliary information (e.g., modal features or user attributes) Li et al. (2022), which increases computational complexity and limits practical deployment.

## 3 Methodology

### 3.1 Preliminary

**Notation.** Suppose that $\mathcal{U} = \{u_1, \ldots, u_m\}$ denotes the set of users and $\mathcal{I} = \{I_1, \ldots, I_n\}$ denotes the set of items, where $m$ and $n$ are the number of users and items, respectively. For each user $u_i \in \mathcal{U}$, the historical interaction sequence is $S_i = (I_1^i, \ldots, I_{t_i}^i)$, where $I_j^i \in \mathcal{I}$ is the $j$-th interacted item and $t_i$ is the sequence length. Let $\mathcal{A} \in \mathbb{R}^{n \times \alpha}$ be the prototype assignment matrix, where $\mathcal{A}[j, r] = a_{j,r}$ represents the probability that item $I_j$ belongs to prototype $\mathbf{p}_r$.

**Problem Definition.** Given $u_i$ and the historical interaction sequence $S_i$, the task aims to predict the next item $I_{t_i+1}$ that $u_i$ is most likely to interact with.

**Graph Construction.** We construct a sequential graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$ as model input. The node set $\mathcal{N} = \mathcal{U} \cup \mathcal{I}$ contains both user and item nodes. The edge set $\mathcal{E}$ consists of two types of associations, i.e., user-item interactions and item-item sequential relationships. These associations are recorded as the adjacency matrix $\mathbf{M} \in \mathbb{R}^{(m+n)\times(m+n)}$:

$$\mathbf{M} = \begin{bmatrix} \mathcal{M}_{I\to I} & \mathcal{M}_{U\to I} \\ \mathcal{M}_{I\to U} & \mathbf{0} \end{bmatrix}, \tag{1}$$

where $\mathcal{M}_{I\to I} \in \mathbb{R}^{n\times n}$, $\mathcal{M}_{U\to I} \in \mathbb{R}^{m\times n}$, and $\mathcal{M}_{I\to U} \in \mathbb{R}^{n\times m}$ denote the item-item sequential, user-to-item, and item-to-user relationships, respectively.

### 3.2 Overall Framework

As illustrated in Figure 2, LightGPD consists of three key components: (1) a prototype learning module that discovers abstract interest prototypes from item embeddings via subspace alignment; (2) a dreaming module that augments sparse interactions through structured data augmentation; and (3) a prediction layer that fuses multi-view user representations for next-item prediction. We deliberately adopt a two-layer GCN as the base architecture to demonstrate that the proposed prototype-guided dreaming strategy can yield substantial improvements even with minimal model complexity.

### 3.3 Graph Convolutional Module

We adopt LightGCN He et al. (2020) to learn initial node embeddings from the sequential graph $\mathcal{G}$. It retains only the neighborhood aggregation component to achieve lightweight representation learning. The propagation on the $l$-th layer is:

$$\mathbf{E}^{(l)} = \left(\mathbf{D}^{-\frac{1}{2}}\left(\mathbf{M} + \mathbf{I}\right)\mathbf{D}^{-\frac{1}{2}}\right)\mathbf{E}^{(l-1)}, \tag{2}$$

where $\mathbf{E}^{(l)} \in \mathbb{R}^{(m+n)\times d}$ denotes the node representations, $\mathbf{I}$ is the identity matrix, $\mathbf{D}$ is the degree matrix of $\mathbf{M} + \mathbf{I}$, and $d$ is the embedding dimension. We set $L = 2$ to avoid over-smoothing He et al. (2020). After $L$ layers of propagation, we obtain the final node representations $\mathbf{E} = [\mathbf{E}_U; \mathbf{E}_I]$, where $\mathbf{E}_U \in \mathbb{R}^{m\times d}$ and $\mathbf{E}_I \in \mathbb{R}^{n\times d}$ represent the user and item embeddings, respectively.

### 3.4 Prototype Learning Module

We introduce a prototype learning module that discovers $\alpha$ abstract interest prototypes $\mathbf{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_r, \ldots, \mathbf{p}_\alpha\} \in \mathbb{R}^{\alpha\times d}$ from the item embedding matrix $\mathbf{E}_I$. These prototypes serve as memory traces that capture latent preference patterns, providing the basis for the subsequent dreaming module.

**Prototype Initialization.** We employ a subspace alignment strategy to obtain semantically meaningful prototypes. We first initialize the subspace bases $\mathcal{D} = \{\mathbf{d}_1, \ldots, \mathbf{d}_\alpha\} \in \mathbb{R}^{\alpha\times d}$ via K-means clustering on $\mathbf{E}_I$, where each basis vector represents a fundamental preference direction. We then calculate the subspace affinity between each item $\mathbf{e}_j$ and each basis $\mathbf{d}_r$:

$$s_{jr} = \frac{\left\|\mathbf{e}_j^\top \mathbf{d}_r\right\|_{\mathrm{F}}^2 + \eta d}{\sum_{r'=1}^{\alpha}\left(\left\|\mathbf{e}_j^\top \mathbf{d}_{r'}\right\|_{\mathrm{F}}^2 + \eta d\right)}, \tag{3}$$

where $\eta$ is a smoothness parameter (set to $1e^{-4}$ according to Cai et al. (2022)) and $d$ is the embedding dimension. This affinity calculation achieves linear complexity $O(n \times \alpha \times d)$. The $r$-th initialized prototype is obtained by aggregating item embeddings weighted by their subspace affinities:

$$\mathbf{p}_{\text{Init},r} = \frac{\sum_{j=1}^{n} s_{jr} \cdot \mathbf{e}_j}{\sum_{j=1}^{n} s_{jr}}. \tag{4}$$

This subspace-based initialization shifts the focus from the sparse interaction space to a dense semantic interest space.

**Prototype Assignment.** For item $I_j$, we calculate its assignment score $a_{j,r}$ to $\mathbf{p}_{\text{Init},r}$ via an asymmetric linear attention:

$$a_{j,r} = \frac{\phi(\mathbf{e}_{I_j})^\top \psi(\mathbf{p}_{\text{Init},r})}{\sum_{r'=1}^{\alpha} \phi(\mathbf{e}_{I_j})^\top \psi(\mathbf{p}_{\text{Init},r'})}, \tag{5}$$

where $\phi(\mathbf{x}) = \text{ELU}(\mathbf{W}_\phi \mathbf{x} + \mathbf{b}_\phi) + 1$ and $\psi(\mathbf{x}) = \mathbf{W}_\psi \mathbf{x} + \mathbf{b}_\psi$. Here $\mathbf{W}_\phi, \mathbf{W}_\psi \in \mathbb{R}^{d \times d}$ and $\mathbf{b}_\phi, \mathbf{b}_\psi \in \mathbb{R}^d$ are learnable parameters. Compared to standard linear attention Katharopoulos et al. (2020), $\phi(\cdot)$ uses non-linear activation to ensure non-negative outputs, while $\psi(\cdot)$ remains linear for computational efficiency. We then obtain the assignment matrix $\mathcal{A} \in \mathbb{R}^{n \times \alpha}$, where each row $\mathbf{a}_j$ represents the prototype distribution for item $I_j$.

**Prototype Representation Update.** Each prototype $\mathbf{p}_r$ is iteratively refined as the weighted average of all item embeddings:

$$\mathbf{p}_r = \frac{\sum_{I_j \in \mathcal{I}} a_{j,r} \cdot \mathbf{e}_{I_j}}{\sum_{I_j \in \mathcal{I}} a_{j,r}}, \tag{6}$$

encouraging items with similar latent preferences to cluster around the same prototype, forming coherent abstract interest concepts for the subsequent dreaming module.

### 3.5 Dreaming Module

We devise a dreaming module that augments sparse interactions through structured data augmentation. During training, the module progressively refines the graph via residual updates; at inference, the refined graph is retained for prediction while the stochastic components (i.e., interaction replay and hypothesis generation) are deactivated. The module operates in four phases: (1) Interaction Replay, (2) Abstraction, (3) Hypothesis Generation and Verification, and (4) Dynamic Graph Reconstruction.

**Interaction Replay.** This phase replays and strengthens important interactions. For each $u_i$, we weigh each interaction by its significance. The score $h_{i,j}$ for $I_j^i \in S_i$ is calculated based on temporal position, prototype relevance, and relative frequency:

$$h_{i,j} = \left(\frac{j}{t_i}\right)^\rho \cdot \max_r \left(\mathcal{A}[\text{idx}(I_j^i), r]\right) \cdot \left(1 + \frac{\text{count}(i,j)}{\sum_{j'} \text{count}(i,j')}\right), \tag{7}$$

where $j/t_i$ is the relative temporal position, $\mathcal{A}[\text{idx}(I_j^i), r]$ is the prototype assignment score of the interacted item, and $\text{count}(i,j)$ is the interaction frequency. The temporal decay rate $\rho$ is adaptively determined by dataset sparsity:

$$\rho = 1 + \sigma\left(\log_{10} \frac{|\mathcal{O}|}{m \times n} + 1\right), \tag{8}$$

where $\sigma(\cdot)$ is the sigmoid function and $\mathcal{O}$ is the set of observed interactions. This design ensures that $\rho$ approaches 1.0 for sparse datasets and grows toward 2.0 for denser ones. We then sample a replayed sequence $\tilde{S}_i$ from $S_i$ based on normalized scores:

$$p(I_j^i | u_i) = \frac{h_{i,j}}{\sum_{j'=1}^{t_i} h_{i,j'}}. \tag{9}$$

**Abstraction.** This phase converts concrete interactions into abstract interest concepts. For each $\tilde{I}_j^i$ in the replayed sequence $\tilde{S}_i$, we retrieve its prototype assignment vector from $\mathcal{A}$:

$$\mathbf{x}_j^i = \mathcal{A}[\text{idx}(\tilde{I}_j^i), :], \tag{10}$$

where $\text{idx}(\cdot)$ returns the item index and $\mathbf{x}_j^i \in \mathbb{R}^\alpha$ is the abstract interest state at step $j$. This transforms the sequence from items $S_i$ into interest distributions $\mathbf{X}_i = (\mathbf{x}_1^i, \ldots, \mathbf{x}_{t_i}^i)$.

**Hypothesis Generation and Verification.** This phase generates and tests hypothetical interest transitions. Specifically, we generate a predicted next interest state $\hat{\mathbf{x}}_{j+1}^i$ from the current state $\mathbf{x}_j^i$:

$$\hat{\mathbf{x}}_{j+1}^i = \text{softmax}(\mathbf{W}_g \mathbf{x}_j^i + \mathbf{b}_g), \tag{11}$$

where $\mathbf{W}_g \in \mathbb{R}^{\alpha \times \alpha}$ and $\mathbf{b}_g \in \mathbb{R}^\alpha$ are learnable parameters.

For the verification, the discrepancy between $\hat{\mathbf{x}}_{j+1}^i$ and the real next state $\mathbf{x}_{j+1}^i$ is measured by KL-divergence:

$$\mathcal{L}_{dream} = \frac{1}{|\mathcal{U}|} \sum_{u_i \in \mathcal{U}} \sum_{j=1}^{t_i - 1} D_{KL}(\mathbf{x}_{j+1}^i \| \hat{\mathbf{x}}_{j+1}^i). \tag{12}$$

This auxiliary loss regularizes the learning process, encouraging the prototypes to form coherent and predictable interest patterns.

**Dynamic Graph Reconstruction.** This phase refines the graph structure and node representations based on the abstract interest knowledge. For each $u_i$, we compute the aggregated interest representation from $\tilde{S}_i$:

$$\mathbf{z}_{u_i} = \frac{1}{t_i} \sum_{j=1}^{t_i} \mathcal{A}[\text{idx}(\tilde{I}_j^i), :], \tag{13}$$

where $\mathbf{z}_{u_i} \in \mathbb{R}^\alpha$ is $u_i$'s overall interest distribution, derived from the replayed interactions rather than the original sequence. A projection layer aligns it with the item embedding space:

$$\hat{\mathbf{z}}_{u_i} = \mathbf{W}_z \mathbf{z}_{u_i} + \mathbf{b}_z, \tag{14}$$

where $\mathbf{W}_z \in \mathbb{R}^{d\times\alpha}$ and $\mathbf{b}_z \in \mathbb{R}^d$ are learnable parameters. The affinity between $u_i$ and $I_j$ is:

$$\tilde{w}_{u_i,I_j} = \hat{\mathbf{z}}_{u_i}^\top \mathbf{e}_{I_j}. \tag{15}$$

We reconstruct the graph by selecting the top-$T$ edges per user based on these affinities. This process removes noisy or less relevant interactions and adds new connections to items that align with the user's interest prototypes. The reconstructed graph $\tilde{\mathcal{G}}$ contains:

$$\tilde{\mathcal{E}} = \{(u_i, I_j) \mid \tilde{w}_{u_i,I_j} \text{ is in the top-}T \text{ for } u_i\}. \tag{16}$$

The reconstructed graph $\tilde{\mathcal{G}}$ updates node representations via additional convolutional layers:

$$\tilde{\mathbf{E}}^{(l)} = H(\tilde{\mathbf{E}}^{(l-1)}, \tilde{\mathcal{G}}), \quad l = L{+}1, \ldots, L{+}L_{dream}, \tag{17}$$

where $L_{dream} = 2$ to avoid over-smoothing, and $\tilde{\mathbf{E}}^{(L)} = \mathbf{E}^{(L)}$ is initialized with the original node representations. To prevent the graph structure from drifting too far from the original observations, we adopt a residual update:

$$\mathcal{G}^{(e+1)} = (1-\gamma) \cdot \mathcal{G}^{(e)} + \gamma \cdot \tilde{\mathcal{G}}^{(e)}, \tag{18}$$

where $\gamma \in [0, 1]$ controls the update step size. This mechanism preserves the original graph as an anchor while allowing incremental edge refinement.

### 3.6 Prediction Layer

The prediction fuses three views of user representations:

$$P(I_{t_i+1}|S_i, u_i) = \text{softmax}\Big(\mathbf{W}_f[\mathbf{e}_{u_i}^{(L)}; \tilde{\mathbf{e}}_{u_i}; \mathbf{z}_{u_i}]^\top \mathbf{E}_I^{(L)} + \mathbf{b}_f\Big), \tag{19}$$

where $\mathbf{W}_f \in \mathbb{R}^{d\times(2d+\alpha)}$ is the transformation matrix, $\mathbf{b}_f \in \mathbb{R}^n$ is the bias term, $[\cdot;\cdot]$ denotes concatenation, $\mathbf{e}_{u_i}^{(L)}$ is the original GCN embedding, $\tilde{\mathbf{e}}_{u_i}$ is derived from the refined graph (Eqn. 17), $\mathbf{z}_{u_i}$ is the prototype interest distribution (Eqn. 13), and $\mathbf{E}_I^{(L)}$ is the item embedding matrix. The cross-entropy loss is:

$$\mathcal{L}_{main} = -\frac{1}{|\mathcal{S}|} \sum_{S_i\in\mathcal{S}} \sum_{I_j\in\mathcal{I}} y_{i,j} \cdot \log P(I_j|S_i, u_i), \tag{20}$$

where $y_{i,j}$ is the ground-truth indicator (1 if $I_j$ is the true next item for $u_i$, 0 otherwise) and $\mathcal{S}$ is the set of training sequences. The overall loss is:

$$\mathcal{L} = \mathcal{L}_{main} + \beta \cdot \mathcal{L}_{dream}, \tag{21}$$

where $\beta$ controls the augmentation weight.

## 4 Experiments

We conduct experiments to answer four **R**esearch **Q**uestion**s** (RQs): how LightGPD compares with state-of-the-art methods on cold-start sequential recommendation (**RQ1**); how the key components contribute to its performance (**RQ2**); how LightGPD performs in training efficiency and parameter scale (**RQ3**); and how the key hyperparameters affect its performance (**RQ4**).

### 4.1 EXPERIMENTAL SETTINGS

**Datasets.** We use Amazon Reviews'23 Electronics Wang et al. (2025), MovieLens-1M Zhang et al. (2025b), and KuaiRec Gao et al. (2022). We retain ratings of at least 4 for Electronics and viewing records with a watch ratio of at least 1.0 for KuaiRec, apply iterative 5-core filtering, and keep at most the 52 most recent interactions per user; full preprocessing details and dataset statistics are provided in Appendix A. Electronics is the sparsest (99.91%), followed by MovieLens-1M (98.49%), while KuaiRec (the official small interaction matrix) is comparatively denser (92.75%) and serves to verify performance under relatively rich supervision.

**Baselines.** We compare LightGPD with nine baselines in two categories: (1) **Sequential recommender systems (SRs): SASRec** Kang & McAuley (2018) (attention-based), **FineRec** Zhang et al. (2024) (LLM-enhanced), **CoLA-Former** Zhao et al. (2025) (lightweight Transformer), **EA-GPS** Zhang et al. (2025b) (lightweight graph), and **LS4SRec** He et al. (2026) (self-supervised graph). (2) **Cold-start sequential recommender systems (CSSRs): MetaTL** Wang et al. (2021) (meta-learning), **GitRec** Li et al. (2022) (graph-based interpolation), **CMCLRec** Xu et al. (2024) (contrastive learning), and **CTM** Wang et al. (2025) (counterfactual meta-learning).

**Evaluation Metrics.** We adopt three widely-used metrics for top-10 recommendation: Recall@10 (R@10), MRR@10, and NDCG@10 Kang & McAuley (2018); Zhang et al. (2025b); Liang et al. (2025).

**Implementation Details.** We implement LightGPD in PyTorch on an NVIDIA RTX 3090 GPU. The model uses two graph convolution layers in the original graph encoder and two additional convolution layers on the reconstructed graph within a single dreaming module; each dataset uses its own configuration of embedding size, batch size, learning rate, weight decay, dropout, and $\beta$, selected on the validation set (see Appendix A). We initialize all parameters with Xavier initialization Glorot & Bengio (2010) and optimize with Adam Kingma & Ba (2015). All results are reported as the mean$\pm$std over three runs with random seeds 42, 3407, and 2024, and the significance markers in Table 1 are assessed with paired t-tests over the three seeds ($p < 0.05$).

Table 1: Prediction performance (%) on three datasets (mean$\pm$std over three random seeds). The best results are in **bold** and the second-best results are underlined. Significant improvements over the best baseline are marked with $^\dagger$ (paired t-test, $p < 0.05$).

| Dataset | Electronics | | | MovieLens-1M | | | KuaiRec | | |
|---|---|---|---|---|---|---|---|---|---|
| Metric | R@10 | MRR@10 | NDCG@10 | R@10 | MRR@10 | NDCG@10 | R@10 | MRR@10 | NDCG@10 |
| SASRec | 2.95±0.03 | 1.12±0.01 | 1.55±0.02 | 6.30±0.07 | 2.05±0.04 | 3.01±0.05 | 42.60±0.22 | 28.52±0.15 | 31.93±0.14 |
| FineRec | 3.02±0.04 | 1.16±0.01 | 1.58±0.02 | 6.42±0.10 | 2.10±0.04 | 3.09±0.05 | 43.10±0.22 | 29.05±0.16 | 32.44±0.14 |
| CoLA-Former | 3.11±0.03 | 1.19±0.01 | 1.63±0.02 | 6.55±0.06 | 2.14±0.04 | 3.14±0.06 | 43.90±0.21 | 29.90±0.13 | 33.30±0.14 |
| EA-GPS | 3.13±0.03 | 1.20±0.01 | 1.65±0.01 | 6.58±0.07 | 2.15±0.04 | 3.17±0.05 | 44.05±0.24 | 29.92±0.12 | 33.34±0.13 |
| LS4SRec | 3.07±0.03 | 1.17±0.01 | 1.61±0.01 | 6.47±0.07 | 2.12±0.04 | 3.10±0.05 | 43.70±0.17 | 29.70±0.20 | 33.11±0.19 |
| MetaTL | 3.22±0.04 | 1.24±0.01 | 1.68±0.01 | 6.72±0.06 | 2.19±0.04 | 3.21±0.08 | 44.30±0.18 | 29.98±0.13 | 33.46±0.18 |
| GitRec | 3.26±0.04 | 1.26±0.01 | 1.73±0.01 | 6.80±0.05 | 2.22±0.04 | 3.27±0.07 | 44.45±0.19 | 30.10±0.12 | 33.58±0.18 |
| CMCLRec | 3.31±0.03 | 1.27±0.01 | 1.74±0.02 | 6.87±0.07 | 2.25±0.05 | 3.30±0.04 | 44.55±0.24 | 30.20±0.14 | 33.67±0.19 |
| CTM | 3.38±0.03 | 1.28±0.01 | 1.77±0.01 | 6.92±0.07 | 2.28±0.05 | 3.33±0.06 | 44.60±0.18 | 30.25±0.18 | 33.74±0.20 |
| **LightGPD** | **3.55±0.05**$^\dagger$ | **1.37±0.03**$^\dagger$ | **1.87±0.03**$^\dagger$ | **7.10±0.06**$^\dagger$ | **2.44±0.05**$^\dagger$ | **3.52±0.07**$^\dagger$ | **45.78±0.44**$^\dagger$ | **30.94±0.42**$^\dagger$ | **34.53±0.34**$^\dagger$ |

### 4.2 OVERALL PERFORMANCE (RQ1)

As presented in Table 1: 1) Traditional SRs perform reasonably well on KuaiRec but struggle on the cold-start datasets, especially on MovieLens-1M. This observation indicates the limitations of modeling sequential patterns when cold-start challenges are not explicitly addressed. 2) CSSRs consistently outperform traditional SRs on both cold-start datasets, validating the significance of specialized strategies for sparse data. 3) Among CSSRs, CTM achieves the second-best performance among the CSSR baselines, benefiting from its counterfactual task-augmented meta-learning framework. 4) LightGPD outperforms all baselines across all metrics on all three datasets. Such consistent improvements validate the superiority of the prototype-guided dreaming strategy. 5) LightGPD also achieves the best performance on KuaiRec, demonstrating its applicability to traditional sequential recommendation.

### 4.3 Ablation Study (RQ2)

We design five variants on Electronics and MovieLens-1M: (1) **GPD**$_{w/o\ PL}$ removes the prototype learning module; (2) **GPD**$_{re\ HA}$ replaces interaction replay with uniform random replay; (3) **GPD**$_{w/o\ HG}$ disables hypothesis generation and verification; (4) **GPD**$_{w/o\ DR}$ removes dynamic graph reconstruction; (5) **GPD**$_{w/o\ DM}$ completely disables the dreaming module. As shown in Table 2: 1) GPD$_{w/o\ PL}$ shows the largest performance drop, confirming that prototype learning is the most critical component for clustering sparse interactions into abstract interest concepts. 2) GPD$_{re\ HA}$ underperforms the full model, indicating that frequency- and prototype-aware replay better preserves important interactions than random replay. 3) GPD$_{w/o\ HG}$ degrades performance, validating that hypothesis generation encourages coherent interest transition patterns. 4) GPD$_{w/o\ DR}$ underperforms the full model, confirming that graph reconstruction enhances the structure by reinforcing prototype-aligned edges. 5) GPD$_{w/o\ DM}$ degrades performance, illustrating the vital role of the dreaming module. 6) The full LightGPD consistently achieves the best performance, confirming that all components work synergistically.

Table 2: Ablation study (%) on Electronics and MovieLens-1M (mean±std over three random seeds).

| **Dataset** | **Electronics** | | | **MovieLens-1M** | | |
|---|---|---|---|---|---|---|
| **Metric (@10)** | **R** | **MRR** | **NDCG** | **R** | **MRR** | **NDCG** |
| GPD$_{w/o\ PL}$ | 3.36±0.03 | 1.22±0.02 | 1.70±0.01 | 5.54±0.05 | 1.74±0.04 | 2.59±0.04 |
| GPD$_{re\ HA}$ | 3.40±0.03 | 1.27±0.01 | 1.75±0.01 | 6.25±0.06 | 2.02±0.05 | 2.95±0.05 |
| GPD$_{w/o\ HG}$ | 3.43±0.03 | 1.28±0.01 | 1.77±0.03 | 6.49±0.06 | 2.16±0.05 | 3.18±0.05 |
| GPD$_{w/o\ DR}$ | 3.46±0.03 | 1.29±0.01 | 1.78±0.01 | 6.77±0.07 | 2.27±0.05 | 3.32±0.05 |
| GPD$_{w/o\ DM}$ | 3.37±0.03 | 1.26±0.01 | 1.74±0.01 | 6.00±0.06 | 1.91±0.03 | 2.82±0.05 |
| **LightGPD** | **3.55±0.05** | **1.37±0.03** | **1.87±0.03** | **7.10±0.06** | **2.44±0.05** | **3.52±0.07** |

### 4.4 Parameter Scale and Training Efficiency (RQ3)

We vary the training data proportion from 0.2 to 1.0 and compare with four competitive baselines and three ablation variants. As shown in Figure 3: 1) LightGPD exhibits significantly lower training time than other CSSR methods (i.e., CoLA-Former, EA-GPS, CTM) and is comparable to SASRec. This advantage becomes more pronounced as data size increases. 2) LightGPD's training time grows nearly linearly with data volume, confirming its scalability. 3) LightGPD maintains a compact model size (32.45 MB on Electronics, 145.78 MB on MovieLens-1M), substantially smaller than other CSSR methods. 4) The full model introduces minimal parameter overhead over its ablated variants, confirming the lightweight design.

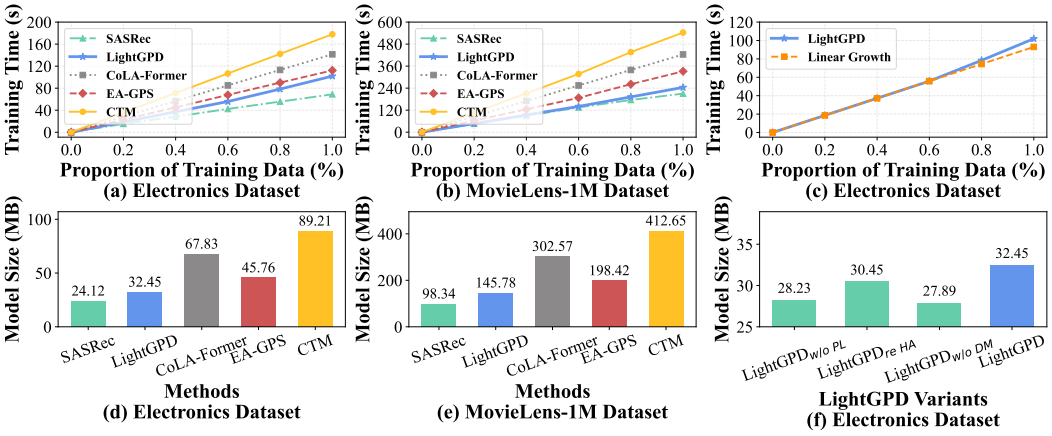



Figure 3: Comparison of training time and parameter scale.

### 4.5 HYPERPARAMETER ANALYSIS (RQ4)

We investigate three key hyperparameters on Electronics and MovieLens-1M, as reported in Figure 4.

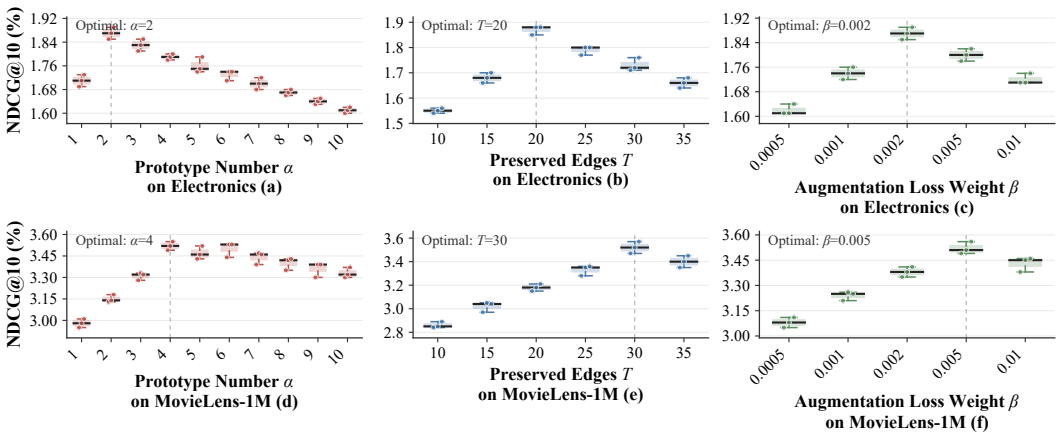



Figure 4: Hyperparameter sensitivity analysis of the prototype number $\alpha$, preserved edges $T$, and augmentation loss weight $\beta$ on Electronics and MovieLens-1M. Each box summarizes the NDCG@10 results over three runs, and the dashed line indicates the optimal setting selected by mean performance.

1) **Number of prototypes** $\alpha$ determines the granularity of the learned interest space. We evaluate $\alpha \in \{1, 2, \ldots, 10\}$: the optimum is 2 on Electronics and 4 on MovieLens-1M, suggesting that cold-start users benefit from coarser interest abstraction, as overly many prototypes would fragment the sparse signals.

2) **Number of preserved prototype edges** $T$ controls how many prototype-based edges are retained during graph reconstruction. Tuning $T \in \{10, 15, 20, 25, 30, 35\}$ per dataset yields optima of 20 on Electronics and 30 on MovieLens-1M, balancing dream-enhanced representations against hallucinated connections.

3) **Augmentation loss weight** $\beta$ balances the main recommendation objective with the auxiliary augmentation regularization. On the logarithmic grid $\beta \in \{0.0005, 0.001, 0.002, 0.005, 0.01\}$, the optimum is 0.002 for Electronics and 0.005 for MovieLens-1M, suggesting that moderately sparse datasets tolerate stronger dreaming supervision than extremely sparse ones.

Beyond the mean trends, the box plots separate systematic effects from seed noise: within any fixed configuration, the per-seed NDCG@10 spread stays below 0.05 points on Electronics and 0.10 points on MovieLens-1M, whereas sweeping $\alpha$, $T$, or $\beta$ moves the mean by up to 0.32 points on Electronics (0.66 points on MovieLens-1M). The graph update rate $\gamma$ is fixed to 0.3 as a design constant, and the temporal decay rate $\rho$ is determined automatically by dataset sparsity via Eq. (8).

## 5 CONCLUSION AND FUTURE WORK

In this work, we propose a Lightweight Graph Prototype Dreaming (LightGPD) network for cold-start sequential recommendation. LightGPD augments a deliberately simple two-layer GCN backbone with prototype learning and a structured dreaming module, alleviating the reliance on auxiliary information at low computational cost. The dreaming module refines the interaction graph in a prototype-aligned manner, so that the observed gains can be attributed to structured data augmentation rather than architectural capacity. Extensive experiments on three real-world datasets demonstrate the effectiveness of LightGPD. In the future, we plan to investigate adaptive prototype learning and to apply the structured augmentation mechanism to broader sparse scenarios such as cross-domain and session-based recommendation.

## AI USE STATEMENT

Generative AI tools were used to assist language editing. The authors reviewed and validated all AI-assisted technical content. The authors take full responsibility for the final content of this work.

## REPRODUCIBILITY STATEMENT

All three datasets used in our experiments are publicly available. Electronics is derived from the timestamped Amazon Reviews'23 dataset (Wang et al., 2025), KuaiRec is obtained from the publicly released fully observed recommendation dataset (Gao et al., 2022), and ML-1M is provided by GroupLens (Zhang et al., 2025b). We release our source code, processed datasets, data preprocessing scripts, and instructions for reproducing the experiments at the anonymous repository: `https://anonymous.4open.science/r/LightGPD`.

## REFERENCES

Simon Nandwa Anjiri, Derui Ding, Yan Song, and Ying Sun. A multiplex hypergraph attribute-based graph collaborative filtering for cold-start POI recommendation. *IEEE Transactions on Big Data*, 11(5):2401–2416, 2025.

Chenwei Cai, Ruining He, and Julian J. McAuley. SPMC: socially-aware personalized markov chains for sparse sequential recommendation. In *IJCAI*, pp. 1476–1482, Melbourne, Australia, 2017. ijcai.org.

Jinyu Cai, Jicong Fan, Wenzhong Guo, Shiping Wang, Yunhe Zhang, and Zhao Zhang. Efficient deep embedded subspace clustering. In *CVPR*, pp. 21–30, New Orleans, LA, USA, 2022. IEEE.

Hongyu Chang, Wenbo Tang, Annabella M Wulf, Thokozile Nyasulu, Madison E Wolf, Antonio Fernandez-Ruiz, and Azahara Oliva. Sleep microstructure organizes memory replay. *Nature*, 637 (8048):1161–1169, 2025.

Khalil Damak, Sami Khenissi, and Olfa Nasraoui. Debiasing the cloze task in sequential recommendation with bidirectional transformers. In *SIGKDD*, pp. 273–282, Washington, DC, USA, 2022. ACM.

Jing Du, Zesheng Ye, Lina Yao, Bin Guo, and Zhiwen Yu. Socially-aware dual contrastive learning for cold-start recommendation. In *SIGIR*, pp. 1927–1932, Madrid, Spain, 2022. ACM.

Shuheng Fang, Kangfei Zhao, Yu Rong, Jeffrey Xu Yu, and Zhixun Li. All-in-one: Heterogeneous interaction modeling for cold-start rating prediction. In *ICDE*, pp. 1537–1550, Hong Kong, China, 2025. IEEE.

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 *CIKM*, pp. 540–550, Atlanta, GA, USA, 2022. ACM.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, pp. 249–256, Sardinia, Italy, 2010. JMLR.org.

Di Han, Xiaotian Jing, Yijun Chen, Junmin Liu, Kai Liao, and Wenting Li. Cold-start recommendation based on knowledge graph and meta-learning under positive and negative sampling. *TORS*, 3(3):33:1–33:24, 2025.

Dongxiao He, Jiaqi Cui, Xiaobao Wang, Guojie Song, Yuxiao Huang, and Lingfei Wu. Dual enhanced meta-learning with adaptive task scheduler for cold-start recommendation. *TKDE*, 37(4): 1728–1741, 2025.

Liang He, Wujie Yan, Tingzhou Yi, and Heli Sun. Self-supervised graph neural sequential recommendation with disentangling long and short-term interest. *TORS*, 4(1):10:1–10:25, 2026.

Ruining He and Julian J. McAuley. Fusing similarity models with markov chains for sparse sequential recommendation. In *ICDM*, pp. 191–200, Barcelona, Spain, 2016. IEEE Computer Society.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*, pp. 639–648, Virtual Event, China, 2020. ACM.

Wang-Cheng Kang and Julian J. McAuley. Self-attentive sequential recommendation. In *ICDM*, pp. 197–206, Singapore, 2018. IEEE Computer Society.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *ICML*, pp. 5156–5165, Virtual Event, 2020. PMLR.

Jinri Kim, Eungi Kim, Kwangeun Yeo, Yujin Jeon, Chanwoo Kim, Sewon Lee, and Joonseok Lee. Content-based graph reconstruction for cold-start item recommendation. In *SIGIR*, pp. 1263–1273, Washington DC, USA, 2024. ACM.

Minchang Kim, Yongjin Yang, Jung Hyun Ryu, and Taesup Kim. Meta-learning with adaptive weighted loss for imbalanced cold-start recommendation. In *CIKM*, pp. 1077–1086, Birmingham, United Kingdom, 2023. ACM.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, pp. 1–15, San Diego, CA, US, 2015. openreview.org.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey E. Hinton. Similarity of neural network representations revisited. In *ICML*, pp. 3519–3529, Long Beach, California, USA, 2019. PMLR.

Songyuan Lei, Xinglong Chang, Zhizhi Yu, Dongxiao He, Cuiying Huo, Jianrong Wang, and Di Jin. Feature-structure adaptive completion graph neural network for cold-start recommendation. In *AAAI*, pp. 12022–12030, Philadelphia, PA, USA, 2025. AAAI Press.

Aoran Li, Jiajun Wu, Shuai Xu, Yalei Zang, Yi Liu, Jiayi Lu, Yanchao Zhao, Gaoxu Wang, Qing Meng, and Xiaoming Fu. Graph-based sequential interpolation recommender for cold-start users. In *APWeb-WAIM*, pp. 57–71, Nanjing, China, 2022. Springer.

Xurong Liang, Vu Nguyen, Vuong Le, Paul Albert, and Julien Monteil. In-context learning for addressing user cold-start in sequential movie recommenders. In *RecSys*, pp. 979–982, Prague, Czech Republic, 2025. ACM.

Huiting Liu, Wei Zhang, Pei-Pei Li, Peng Zhao, and Xindong Wu. Causal meta-learning with multi-view graphs for cold-start recommendation. *TKDD*, 19(7):138:1–138:29, 2025a.

Yuwen Liu, Lianyong Qi, Xingyuan Mao, Weiming Liu, Shichao Pei, Fan Wang, Xuyun Zhang, Amin Beheshti, and Xiaokang Zhou. Variational graph auto-encoder driven graph enhancement for sequential recommendation. In *IJCAI*, pp. 3144–3152, Montreal, Canada, 2025b. ijcai.org.

Luyi Ma, Aashika Padmanabhan, Anjana Ganesh, Shengwei Tang, Jiao Chen, Xiaohan Li, Lalitesh Morishetti, Kaushiki Nag, Malay Patel, Jason H. D. Cho, Sushant Kumar, and Kannan Achan. Improving sequential recommender systems with online and in-store user behavior. In *IEEE Big Data*, pp. 3544–3549, Washington, DC, USA, 2024. IEEE.

Kourosh Maboudi, Bapun Giri, Hiroyuki Miyawaki, Caleb Kemere, and Kamran Diba. Retuning of hippocampal representations during sleep. *Nature*, 629(8012):630–638, 2024.

Anton Pembek, Artem Fatkulin, Anton Klenitskiy, and Alexey Vasilev. Let it go? not quite: Addressing item cold start in sequential recommendations with content-based initialization. In *RecSys*, pp. 626–631, Prague, Czech Republic, 2025. ACM.

Tianhao Peng, Haitao Yuan, Yongqi Zhang, Yuchen Li, Peihong Dai, Qunbo Wang, Senzhang Wang, and Wenjun Wu. Tagrec: Temporal-aware graph contrastive learning with theoretical augmentation for sequential recommendation. *TKDE*, 37(5):3015–3029, 2025.

Chun-Yan Sang, Ming Gong, Shigen Liao, and Wei Zhou. MA-GCL4SR: improving graph contrastive learning-based sequential recommendation with model augmentation. *TKDD*, 19(5):1–21, 2025.

Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *NIPS*, pp. 6306–6315, Long Beach, CA, USA, 2017.

Jianling Wang, Kaize Ding, and James Caverlee. Sequential recommendation for cold-start users with meta transitional learning. In *SIGIR*, pp. 1783–1787, Virtual Event, Canada, 2021. ACM.

Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *SIGIR*, pp. 165–174, Paris, France, 2019. ACM.

Zhiqiang Wang, Jiayi Pan, Xingwang Zhao, Jianqing Liang, Chenjiao Feng, and Kaixuan Yao. Counterfactual task-augmented meta-learning for cold-start sequential recommendation. In *AAAI*, pp. 12801–12809, Philadelphia, PA, USA, 2025. AAAI Press.

Liwei Wu, Shuqing Li, Cho-Jui Hsieh, and James Sharpnack. SSE-PT: sequential recommendation via personalized transformer. In *RecSys*, pp. 328–337, Virtual Event, Brazil, 2020. ACM.

Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Jiajie Xu, Victor S. Sheng, Zhiming Cui, Xiaofang Zhou, and Hui Xiong. Recurrent convolutional neural network for sequential recommendation. In *WWW*, pp. 3398–3404, San Francisco, CA, USA, 2019. ACM.

Xiaolong Xu, Hongsheng Dong, Lianyong Qi, Xuyun Zhang, Haolong Xiang, Xiaoyu Xia, Yanwei Xu, and Wanchun Dou. Cmclrec: Cross-modal contrastive learning for user cold-start sequential recommendation. In *SIGIR*, pp. 1589–1598, Washington DC, USA, 2024. ACM.

An Yan, Shuo Cheng, Wang-Cheng Kang, Mengting Wan, and Julian J. McAuley. Cosrec: 2d convolutional neural networks for sequential recommendation. In *CIKM*, pp. 2173–2176, Beijing, China, 2019. ACM.

Yuyang Ye, Zhi Zheng, Yishan Shen, Tianshu Wang, Hengruo Zhang, Peijun Zhu, Runlong Yu, Kai Zhang, and Hui Xiong. Harnessing multimodal large language models for multimodal sequential recommendation. In *AAAI*, pp. 13069–13077, Philadelphia, PA, USA, 2025. AAAI Press.

Haoyi Zhang, Guohao Sun, Jinhu Lu, Guanfeng Liu, and Xiu Susie Fang. Delrec: Distilling sequential pattern to enhance llms-based sequential recommendation. In *ICDE*, pp. 1–14, Hong Kong, China, 2025a. IEEE.

Jinyu Zhang, Chao Li, and Zhongying Zhao. Lightweight yet efficient: An external attentive graph convolutional network with positional prompts for sequential recommendation. *TOIS*, 43(3):79:1–79:25, 2025b.

Xiaokun Zhang, Bo Xu, Youlin Wu, Yuan Zhong, Hongfei Lin, and Fenglong Ma. Finerec: Exploring fine-grained sequential recommendation. In *SIGIR*, pp. 1599–1608, Washington DC, USA, 2024. ACM.

Yipeng Zhang, Xin Wang, Hong Chen, and Wenwu Zhu. Adaptive disentangled transformer for sequential recommendation. In *SIGKDD*, pp. 3434–3445, Long Beach, CA, USA, 2023. ACM.

Zhongying Zhao, Jinyu Zhang, Chuanxu Jia, Chao Li, Yanwei Yu, and Qingtian Zeng. Cola-former: Graph transformer using communal linear attention for lightweight sequential recommendation. In *IJCAI*, pp. 3689–3697, Montreal, Canada, 2025. ijcai.org.

# TECHNICAL APPENDIX

## A DATASET DETAILS AND TRAINING PROTOCOL

**Preprocessing and statistics.** To obtain positive implicit-feedback sequences, we retain ratings of at least 4 for Electronics and viewing records with a watch ratio of at least 1.0 for KuaiRec, while all ratings are retained for MovieLens-1M. We further apply iterative 5-core filtering and retain at most the 52 most recent interactions for each user. For Electronics, we construct a reproducible subset by sampling at most 20,000 users with a fixed random seed and retaining at most 15,000 popular items. For KuaiRec, we use the official small interaction matrix. Dataset sparsity is calculated as $1 - \frac{|\mathcal{O}|}{|\mathcal{U}| \times |\mathcal{I}|}$, where $\mathcal{O}$ denotes the set of retained interactions, and $\mathcal{U}$ and $\mathcal{I}$ denote the user and item sets, respectively. After the 5-core filtering, every retained user and item participates in at least five interactions in the complete retained interaction set. The cold-start evaluation in Appendix is instead defined by the number of interactions visible in each evaluation prefix, which can be fewer than five after chronological splitting. For an evaluation prefix with no visible interaction, we initialize the corresponding prototype interest distribution as uniform over all prototypes. The statistics of the resulting datasets are reported in Table 3.

Table 3: Statistics of the preprocessed datasets.

| **Dataset** | **Electronics** | **MovieLens-1M** | **KuaiRec** |
|---|---|---|---|
| Users | 16,236 | 6,040 | 1,411 |
| Items | 8,904 | 3,070 | 706 |
| Interactions | 128,506 | 279,129 | 72,210 |
| Sparsity | 99.91% | 98.49% | 92.75% |

**Training configuration.** Each dataset uses its own configuration: Electronics (embedding size 16, batch size 128, learning rate $2 \times 10^{-4}$, weight decay $10^{-4}$, dropout 0.35, $\beta$=0.002), MovieLens-1M (embedding size 32, batch size 256, learning rate $5 \times 10^{-4}$, weight decay $5 \times 10^{-5}$, dropout 0.25, $\beta$=0.005), and KuaiRec (embedding size 32, batch size 128, learning rate $5 \times 10^{-4}$, weight decay $5 \times 10^{-5}$, dropout 0.20, $\beta$=0.005). Training runs for up to 300, 250, and 200 epochs on the three datasets, with evaluation every 5 epochs; early stopping monitors NDCG@10 with a patience of 8, 6, and 6 evaluations, and the learning rate is halved whenever the monitored metric plateaus. For baselines, we adopt the optimal hyperparameters from their original papers and further fine-tune on our datasets.

## B NECESSITY OF THE DREAMING MODULE

One might question whether the dreaming module is merely a complex regularizer, given that its stochastic components are only active during training. We argue that its role goes beyond regularization in two aspects: (1) *Structured data augmentation*: Unlike standard regularization (e.g., dropout, weight decay) that randomly perturbs the model, the dreaming module performs semantically guided augmentation, as it selectively reinforces prototype-aligned interactions and discovers new edges based on learned interest patterns. This is particularly valuable in cold-start scenarios where observed interactions are too sparse to form reliable user representations. (2) *Iterative graph refinement*: The dreaming module reconstructs the graph structure at each epoch, enabling progressive denoising and edge discovery. This addresses a limitation that neither the prototype learning module nor the graph convolutional backbone can resolve: the former captures abstract interests but cannot generate new training signals, while the latter propagates information along existing edges but cannot correct noisy or missing connections. The dreaming module fills this gap by actively synthesizing prototype-consistent interactions and refining the graph topology, which together produce a denser and cleaner training signal for the backbone to leverage in subsequent epochs.

The remainder of this appendix provides empirical support for this claim along four axes: representation-level evidence (embedding quality), robustness of the reconstructed graph under corruption (noise robustness), behavior on the hardest user population (cold-start users), and positioning

against graph and prototype-based baselines. All appendix results follow the same evaluation protocol as the main text: three random seeds with mean±std reporting.

## C Comparison with Graph and Prototype-Based Baselines

We additionally compare LightGPD with two representative graph-based baselines, LightGCN He et al. (2020) and NGCF Wang et al. (2019), together with the prototype-related quantization baseline VQ-VAE van den Oord et al. (2017), on all three datasets. As shown in Table 4, these additional baselines are competitive across the three datasets, but none of them closes the gap to LightGPD. Two observations are worth noting. First, on the extremely sparse Electronics dataset the quantization-based VQ-VAE fares best among the three, suggesting that discrete prototype-like codes partially mitigate cold-start sparsity; LightGPD still retains a consistent lead over VQ-VAE (0.28 points in R@10), indicating that structured augmentation adds value on top of prototype-like quantization. Second, LightGPD leads by 0.52 points in R@10 on MovieLens-1M and by 1.28 points on the densest KuaiRec, showing that the dreaming module refines the graph even when the observed topology is reliable. Overall, the prototype-guided augmentation mechanism provides substantial benefits beyond both standard graph convolution and prototype-like quantization across all sparsity regimes.

Table 4: Comparison with graph and prototype-based baselines (mean±std over three random seeds).

| | **Electronics** | | | **MovieLens-1M** | | | **KuaiRec** | | |
|---|---|---|---|---|---|---|---|---|---|
| **Method** | **R@10** | **MRR@10** | **NDCG@10** | **R@10** | **MRR@10** | **NDCG@10** | **R@10** | **MRR@10** | **NDCG@10** |
| LightGCN | 3.12±0.04 | 1.15±0.01 | 1.59±0.01 | 6.28±0.10 | 2.12±0.08 | 3.08±0.05 | 43.10±0.17 | 29.40±0.16 | 32.73±0.16 |
| NGCF | 3.19±0.03 | 1.18±0.01 | 1.63±0.02 | 6.41±0.08 | 2.17±0.05 | 3.15±0.05 | 43.80±0.20 | 29.60±0.19 | 33.04±0.15 |
| VQ-VAE | 3.27±0.04 | 1.22±0.01 | 1.68±0.02 | 6.58±0.11 | 2.24±0.05 | 3.24±0.05 | 44.50±0.24 | 30.05±0.17 | 33.55±0.13 |
| **LightGPD** | **3.55±0.05** | **1.37±0.03** | **1.87±0.03** | **7.10±0.06** | **2.44±0.05** | **3.52±0.07** | **45.78±0.44** | **30.94±0.42** | **34.53±0.34** |

## D Embedding Quality Analysis

To investigate whether the augmentation module improves embedding quality rather than merely acting as a regularizer, we measure the representation similarity between the dreaming-enhanced and original GCN outputs using Centered Kernel Alignment (CKA) Kornblith et al. (2019), and the Euclidean distance between each user's prototype distribution and the prototype assignment vector of their actual next item. Crucially, we contrast LightGPD against its *w/o DM* ablation under the same protocol: if the dreaming module were merely a regularizer, its removal should barely move the representations, and both variants would look alike. As shown in Table 5, the CKA score of the full model (0.94 on Electronics) is high but clearly not saturated, whereas the w/o DM variant sits at 0.99—essentially indistinguishable from the unmodified backbone. This contrast localizes the representation change to the dreaming module itself. The prototype-distance metric sharpens the picture: the full model brings predicted interests substantially closer to ground-truth outcomes than the ablated variant (0.37 vs. 0.52 on Electronics, 0.33 vs. 0.51 on MovieLens-1M). The MovieLens-1M distances are smaller for both variants, consistent with its denser histories yielding better-separated interest prototypes. Together, these results confirm that the dreaming module actively aligns user representations with their true interest patterns rather than passively stabilizing training.

Table 5: Embedding quality on Electronics and MovieLens-1M (mean±std over three random seeds).

| | **Electronics** | | **MovieLens-1M** | |
|---|---|---|---|---|
| **Metric** | **LightGPD** | **w/o DM** | **LightGPD** | **w/o DM** |
| CKA (vs. backbone) | 0.94±0.01 | 0.99±0.00 | 0.95±0.01 | 0.99±0.00 |
| Prototype distance | **0.37±0.02** | 0.52±0.03 | **0.33±0.02** | 0.51±0.03 |

## E Noise Robustness Analysis

Since the augmentation module reconstructs the graph structure, it is important to verify its robustness against noisy interactions. We randomly perturb 10%, 20%, and 30% of the training interactions on Electronics and MovieLens-1M by replacing them with random items, and evaluate NDCG@10; KuaiRec is omitted, as its dense graph is inherently robust to edge corruption and the effect sizes are negligible. The comparison includes SASRec as a conventional sequential baseline, CTM as a competitive cold-start baseline, and LightGPD with dreaming-based graph reconstruction. As shown in Table 6, the three methods form a strict degradation ordering on both datasets: the graph-free SASRec collapses fastest, CTM degrades at an intermediate rate, and LightGPD exhibits the most graceful decay. The gap between LightGPD and CTM widens markedly with noise on Electronics (1.13 points at 0% to 3.27 points at 30%) and grows on MovieLens-1M as well. This ordering supports the robustness of the proposed reconstruction strategy: the dreaming module's prototype-based affinity scoring filters spurious edges during reconstruction, yielding a more graceful degradation under increasing interaction noise.

Table 6: Noise robustness analysis (NDCG@10, mean±std over three random seeds). The 0% column equals the clean main-text results.

| | Electronics | | | | MovieLens-1M | | | |
|---|---|---|---|---|---|---|---|---|
| **Method** | **0%** | **10%** | **20%** | **30%** | **0%** | **10%** | **20%** | **30%** |
| SASRec | 1.55±0.02 | 1.42±0.01 | 1.25±0.01 | 1.06±0.01 | 3.01±0.05 | 2.67±0.04 | 2.27±0.04 | 1.79±0.02 |
| CTM | 1.77±0.01 | 1.66±0.01 | 1.50±0.01 | 1.34±0.01 | 3.33±0.06 | 3.04±0.05 | 2.73±0.08 | 2.31±0.04 |
| LightGPD | **1.87±0.03** | **1.79±0.01** | **1.68±0.02** | **1.51±0.02** | **3.52±0.07** | **3.31±0.05** | **3.02±0.04** | **2.64±0.05** |

## F Cold-Start User Evaluation

To further validate LightGPD's effectiveness for cold-start users, we conduct a targeted evaluation on users with fewer than 5 visible historical interactions in their evaluation prefixes on all three datasets, comparing against the strongest overall baseline CTM and MetaTL, whose meta-learning objective is explicitly designed for cold-start transfer. As shown in Table 7, LightGPD consistently outperforms both baselines on every dataset and metric; notably, it also surpasses the cold-start-specialized MetaTL, indicating that data-level augmentation and task-level transfer are complementary rather than interchangeable. The positive margins persist across the sparsity spectrum: LightGPD leads CTM by 0.23 points in R@10 on the extremely sparse Electronics, 0.27 on the moderately sparse MovieLens-1M, and 1.10 on the dense KuaiRec, confirming that prototype-guided augmentation benefits the hardest user population on every dataset rather than depending on a particular sparsity regime.

Table 7: Performance comparison on cold-start users (fewer than 5 visible historical interactions in the evaluation prefix, mean±std over three random seeds).

| | Electronics | | | MovieLens-1M | | | KuaiRec | | |
|---|---|---|---|---|---|---|---|---|---|
| **Method** | **R@10** | **MRR@10** | **NDCG@10** | **R@10** | **MRR@10** | **NDCG@10** | **R@10** | **MRR@10** | **NDCG@10** |
| MetaTL | 2.89±0.03 | 1.10±0.02 | 1.51±0.01 | 5.28±0.09 | 1.86±0.04 | 2.66±0.05 | 41.86±0.20 | 28.44±0.12 | 31.70±0.17 |
| CTM | 3.02±0.03 | 1.11±0.01 | 1.54±0.02 | 5.56±0.08 | 1.93±0.05 | 2.78±0.04 | 42.66±0.21 | 29.20±0.19 | 32.48±0.22 |
| **LightGPD** | **3.25±0.05** | **1.23±0.01** | **1.69±0.01** | **5.83±0.08** | **2.13±0.09** | **3.01±0.07** | **43.76±0.20** | **29.65±0.18** | **33.07±0.21** |