DAME: DEPTH-AWARE HYBRID STRUCTURAL MIXTURE-OF-EXPERTS FOR GRAPH RECOMMENDATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph neural networks have achieved remarkable success in collaborative recommendation tasks by modeling high-order connections. However, existing methods often force diverse behaviors to follow homogenized aggregation paths, fail to disentangle complex user intents, and lack explicit modeling of the hierarchical semantics inherent in propagation depth. To address this, we propose DAME, a Depth-Aware Mixture-of-Experts framework for graph-based recommendation. DAME employs a structural expert mechanism to achieve node-adaptive graph propagation and incorporates Rotational Embedding Propagation, integrating propagation depth information into node representations through layer-wise orthogonal rotations. This design combines adaptive modeling of interaction structures with explicit depth encoding, facilitating the distinction between shallow and high-order collaborative signals. Experiments on three public datasets demonstrate that DAME achieves state-of-the-art performance. Ablation studies further validate the effectiveness of these two components, while routing analysis reveals the correlation between expert specialization and user interaction patterns.

1 INTRODUCTION

Recommendation systems (Chen et al., 2025b; Anand & Maurya, 2025) have evolved into pivotal information filtering mechanisms, aimed at delivering the most relevant content to users. Within the broader domain of user preference analysis, Graph Neural Networks (GNNs) have emerged as a particularly critical paradigm with applications that span numerous sectors, including e-commerce, social networks, and streaming of content (Zhang et al., 2025b). Consequently, this task has obtained significant attention from researchers, leading to the development of numerous advanced methods (Yan et al., 2025; Shi et al., 2024). Despite these advances, current research in GNN-based recommendation leaves two important aspects insufficiently explored.

First, as shown in Figure 1 (Left), user-item interactions inherently contain signals of different motivations with distinct characteristics, where stable preferences reflect long-term interests, and impulsive clicks reflect short-term dynamics. The recommendation task (Zhao et al., 2025b; Chen et al., 2025c) is mainly to predict future interactions taking the historical graph as input and extracting their collaborative features. Based on the way structural information is utilized during feature extraction, existing methods predominantly operate on a static, global adjacency matrix. These approaches (Liu et al., 2025; Sang et al., 2025) treat all interactions uniformly, forcing diverse behaviors to pass through a homogenized aggregation path. Recent empirical studies (Li et al., 2025b; Cai et al., 2024) reveal that the importance of different interaction patterns varies across users, with each component potentially beneficial or detrimental to forecasting performance depending on the specific context (Ge et al., 2025). This indicates that simply aggregating all neighbors through a unified graph structure may be suboptimal. However, although recent studies have introduced contrastive learning (Wu et al., 2023; Liang et al., 2025a;b) to augment data, they primarily focus on random perturbations, leaving the explicit modeling of structural decoupling and cross-intent interactions unexplored.

Second, the depth of propagation naturally reflects the different strength of collaborative signals. As illustrated in Figure 1 (Right), analysis of node influence across different hops reveals two distinct semantic patterns. One is the direct dependency, where neighbors in one hop represent strong,

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.

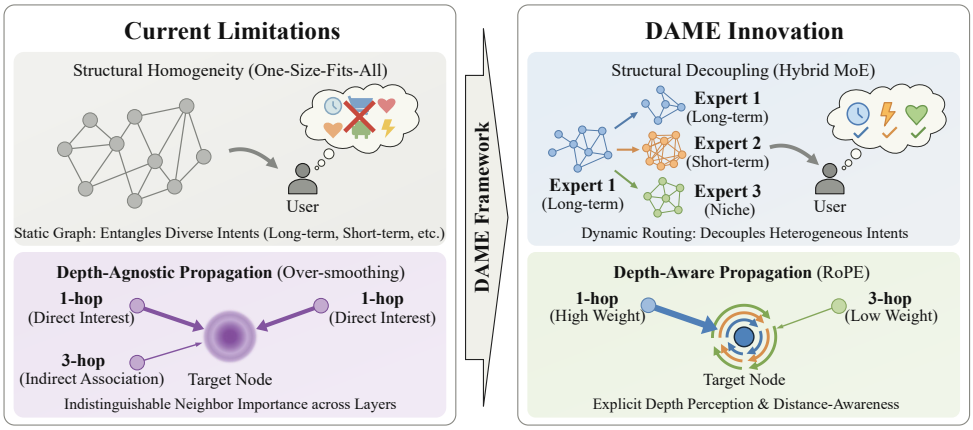

Figure 1: Conceptual comparison of traditional GNN-based recommendation limitations (left) and the proposed DAME framework's solutions for structural decoupling and depth-awareness (right).

explicit user preferences. The other is the indirect correlation, where high-order neighbors provide auxiliary context but carry weaker semantic weight. Recent methods (Chen et al., 2025a; Dang et al., 2025; Wang et al., 2025; Zhao et al., 2025a) have dismissed weaker semantic weights by adaptively identifying and removing redundant edges and unimportant embedding dimensions from the graph to capture high-order signals. Some approaches (Zhang et al., 2025a; Huang et al., 2025; Yang & Liang, 2025) attempt to address the resulting over-smoothing issue through residual connections or layer-wise weighting, yet these methods have shown limited effectiveness in practice as they remain depth-agnostic. Despite these attempts in deep graph modeling, the geometric characteristics inherent in propagation depth remain insufficiently explored. To fully harness the potential of high-order connectivity for performance enhancement, exploring explicit depth perception (Li et al., 2025a) becomes a promising direction.

In this paper, we propose **DAME**, a graph recommendation framework with two components. The Hybrid Structural Mixture-of-Experts module learns multiple sparse expert subgraphs and routes each node to a small set of structural views, preserving shared global signals while enabling node-adaptive denoising. The Rotational Embedding Propagation module applies depth-dependent rotations after each aggregation layer, so representations from different propagation depths remain distinguishable before layer pooling. The core contributions of this work are summarized as follows:

- **A Structural Decoupling Framework via Hybrid Experts:** We innovatively introduce the Mixture-of-Experts paradigm into the graph propagation layer to decompose the message passing task into multiple highly specialized sparse structure experts.
- **Depth-Aware Propagation via RoPE:** We adapt Rotary Embedding Propagation for graph collaborative filtering. By applying depth-dependent rotations after graph aggregation, RoPE explicitly encodes propagation depth without imposing an order on graph nodes.
- **State-of-the-Art Performance and Analysis:** Extensive experiments on three real-world datasets demonstrate that DAME comprehensively outperforms existing state-of-the-art methods. Furthermore, in-depth analysis reveals that through the collaboration of 32 expert bases and a Top-4 routing strategy, DAME successfully disentangles complex structural correlations, significantly expanding the model's capacity to handle large-scale heterogeneous interaction graphs.

2 Preliminaries

We denote the user set as $\mathcal{U}$ and the item set as $\mathcal{I}$. Observed interactions form a bipartite graph $\mathcal{G} = (\mathcal{U} \cup \mathcal{I}, \mathcal{E})$ with normalized adjacency matrix $\hat{\mathbf{A}}$. Given initial node embeddings $\mathbf{E}^{(0)} \in$

$\mathbb{R}^{(|\mathcal{U}|+|\mathcal{I}|)\times d}$, lightweight graph recommendation models propagate representations by

$$\mathbf{E}^{(l)} = \hat{\mathbf{A}}\mathbf{E}^{(l-1)}, \quad \hat{a}_{ij} = \frac{1}{\sqrt{d_i d_j}}\mathbb{I}((i,j) \in \mathcal{E}). \tag{1}$$

The final embedding is usually obtained by pooling layer representations, e.g., $\mathbf{E}^* = \sum_{l=0}^{L} \mathbf{E}^{(l)}$.

Mixture-of-Experts. A Mixture-of-Experts (MoE) layer combines multiple expert functions with an input-dependent router. For an input representation $\mathbf{x}$, a sparse Top-K MoE can be written as

$$\mathrm{MoE}(\mathbf{x}) = F_s(\mathbf{x}) + \sum_{k\in\mathcal{K}(\mathbf{x})} \bar{g}_k(\mathbf{x}) F_k(\mathbf{x}), \quad \mathcal{K}(\mathbf{x}) = \mathrm{TopK}(\mathrm{Softmax}(\mathbf{W}_g\mathbf{x})), \tag{2}$$

where F_s is a shared expert, F_k is the k-th routed expert, and $\bar{g}_k$ denotes the normalized routing weight among selected experts. In DAME, experts are graph propagation operators over learned sparse subgraphs rather than feed-forward networks.

Rotary Position Encoding. Rotary position encoding (RoPE) injects positional information by rotating every two embedding dimensions with a position-dependent angle. For the r-th two-dimensional pair of a vector at position p, RoPE applies

$$\begin{bmatrix} x'_{2r} \\ x'_{2r+1} \end{bmatrix} = \begin{bmatrix} \cos\phi_{p,r} & -\sin\phi_{p,r} \\ \sin\phi_{p,r} & \cos\phi_{p,r} \end{bmatrix} \begin{bmatrix} x_{2r} \\ x_{2r+1} \end{bmatrix}. \tag{3}$$

Instead of using token order as in sequence models, DAME treats propagation depth as the position, so embeddings from different graph layers remain distinguishable before layer pooling.

3 Proposed Model

In this section, we present **DAME**, a novel framework designed to address structural noise and over-smoothing in graph-based recommendations. The overall architecture is illustrated in Figure 2. We first formulate the recommendation task, then describe the two core components: the *Hybrid Structural Mixture-of-Experts* for adaptive structural denoising and the *Rotational Embedding Propagation* for depth-aware signal encoding. Finally, we introduce the joint training objective.

3.1 Task Formulation

Given the bipartite graph $\mathcal{G} = (\mathcal{U} \cup \mathcal{I}, \mathcal{E})$ defined in preliminaries, our objective is to learn a graph encoder $f_\Theta : \mathcal{G} \to \mathbb{R}^{(|\mathcal{U}|+|\mathcal{I}|)\times d}$. The encoder maps each user or item node $v \in \mathcal{U} \cup \mathcal{I}$ to a low-dimensional embedding $\mathbf{e}_v$ that captures both direct and high-order collaborative signals. For a target user u, the predicted preference score for item i is computed by inner product:

$$\hat{y}_{ui} = \mathbf{e}_u^\top \mathbf{e}_i. \tag{4}$$

The recommendation task is formulated as pairwise ranking, where observed items should be ranked above unobserved items:

$$\hat{y}_{ui} > \hat{y}_{uj}, \quad \forall i \in \mathcal{N}_u, j \notin \mathcal{N}_u \tag{5}$$

where $\mathcal{N}_u$ denotes the observed interaction set of user u. This formulation motivates the following design: DAME first denoises graph propagation through node-wise structural experts, and then uses RoPE to keep different propagation depths distinguishable before the final ranking score is computed.

3.2 Hybrid Structural Mixture-of-Experts

To mitigate the noise amplification inherent in static graph convolution, DAME learns multiple sparse graph propagation operators and dynamically selects a small subset of them for each node. Expert masks learn sparse graph views from the original interaction graph, and the router selects a Top-K mixture of these views at each layer.

Expert Construction. We use $N_{exp} + 1$ structural experts: one shared expert that is always active to preserve global connectivity, and N_{exp} routed experts that learn sparse topology views for node-specific propagation.

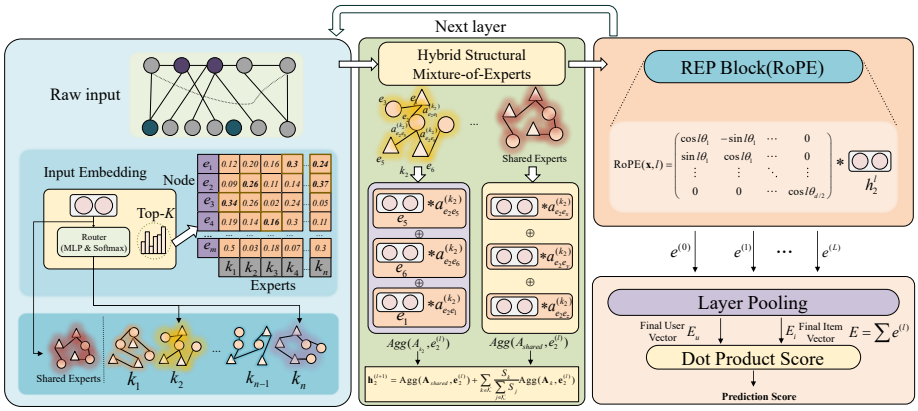

Figure 2: The overall architecture of the proposed DAME framework. The Hybrid Structural Mixture-of-Experts block routes node representations to one shared structural expert and Top-K routed structural experts, where each expert performs graph propagation over a learned sparse view of the original interaction graph. The RoPE block then applies a depth-specific rotation to the aggregated representation. Finally, the layer pooling module aggregates multi-depth embeddings ($\mathbf{E}^{(0)}$ to $\mathbf{E}^{(L)}$) to obtain the final user and item vectors for dot-product prediction.

Expert Subgraph Learning. For each layer l and expert k, every observed edge $e \in \mathcal{E}$ has a learnable mask logit $\alpha_{e,k}^{(l)}$. We convert it into a binary mask with a straight-through estimator (STE):

$$p_{e,k}^{(l)} = \sigma(\alpha_{e,k}^{(l)}), \quad z_{e,k}^{(l)} = \mathbb{I}(p_{e,k}^{(l)} > \tau) - \mathrm{sg}(p_{e,k}^{(l)}) + p_{e,k}^{(l)}, \tag{6}$$

where $\mathrm{sg}(\cdot)$ denotes stop-gradient. The expert-specific adjacency is

$$\tilde{\mathbf{A}}_k^{(l)} = \hat{\mathbf{A}} \odot \mathbf{Z}_k^{(l)}, \tag{7}$$

where $\hat{\mathbf{A}}$ is the normalized original adjacency matrix and $\mathbf{Z}_k^{(l)}$ contains masks on observed edges. Thus, each expert is a learned sparse view of the original topology.

Node-wise Sparse Routing. Given node embeddings $\mathbf{E}^{(l)}$, each expert performs graph propagation on its learned sparse view:

$$\mathbf{M}_k^{(l)} = \tilde{\mathbf{A}}_k^{(l)} \mathbf{E}^{(l)}, \quad k = 1, \ldots, N_{exp}, \quad \mathbf{M}_{shared}^{(l)} = \tilde{\mathbf{A}}_{shared}^{(l)} \mathbf{E}^{(l)}. \tag{8}$$

For each node u, a router computes the routed expert distribution and keeps only Top-K experts:

$$\mathbf{g}_u^{(l)} = \mathrm{Softmax}(\mathbf{W}_r^{(l)} \mathbf{e}_u^{(l)} + \boldsymbol{\epsilon}), \quad \mathcal{K}_u^{(l)} = \mathrm{TopK}(\mathbf{g}_u^{(l)}). \tag{9}$$

The pre-rotation representation mixes the shared expert and selected routed experts:

$$\mathbf{h}_u^{(l+1)} = \mathbf{M}_{shared,u}^{(l)} + \sum_{k \in \mathcal{K}_u^{(l)}} \frac{g_{u,k}^{(l)}}{\sum_{j \in \mathcal{K}_u^{(l)}} g_{u,j}^{(l)}} \mathbf{M}_{k,u}^{(l)}. \tag{10}$$

Unlike NLP-style MoE, where experts are usually feed-forward networks, each DAME expert is a graph propagation operator. Thus, routing selects structural views rather than feature-transformation branches.

3.3 Rotational Embedding Propagation

Standard GNN layer pooling directly sums embeddings from different propagation depths. DAME uses RoPE as a depth stamp after each graph aggregation layer.

Depth Rotation After Aggregation. At layer l, the MoE first produces the pre-rotation embedding matrix $\mathbf{H}^{(l+1)}$:

$$\mathbf{H}^{(l+1)} = \mathrm{MoE}^{(l)}(\mathbf{E}^{(l)}, \hat{\mathbf{A}}), \tag{11}$$

where $\mathbf{E}^{(l)}$ is the input embedding of the current layer. RoPE is applied to $\mathbf{H}^{(l+1)}$, not before message passing. For each node v, we split $\mathbf{h}_v^{(l+1)}$ into two-dimensional pairs. For the r-th pair, where $r = 0, \ldots, d/2 - 1$, the rotation angle is

$$\phi_{l,r} = (l+1) \cdot \theta_{base}^{-2r/d}. \tag{12}$$

The pairwise rotation is computed as:

$$\begin{bmatrix} e_{v,2r}^{(l+1)} \\ e_{v,2r+1}^{(l+1)} \end{bmatrix} = \begin{bmatrix} \cos\phi_{l,r} & -\sin\phi_{l,r} \\ \sin\phi_{l,r} & \cos\phi_{l,r} \end{bmatrix} \begin{bmatrix} h_{v,2r}^{(l+1)} \\ h_{v,2r+1}^{(l+1)} \end{bmatrix}. \tag{13}$$

Stacking all rotated pairs gives $\mathbf{E}^{(l+1)}$, which is used both as the representation of layer $l+1$ and as the input to the next DAME layer. RoPE does not impose any order on graph nodes; the encoded "position" is only the propagation depth. Since each rotation matrix is orthogonal, RoPE preserves embedding norms while placing different depths at different angles before layer pooling.

After L propagation layers, we aggregate the initial and rotated layer representations in a LightGCN-style readout:

$$\mathbf{E}^* = \sum_{l=0}^{L} \mathbf{E}^{(l)}, \tag{14}$$

where $\mathbf{E}^{(0)}$ is the initial embedding and $\mathbf{E}^{(l)}$ for $l \geq 1$ denotes the RoPE-rotated output of the l-th DAME layer. The final prediction score is:

$$\hat{y}_{ui} = \mathbf{e}_u^{*\top} \mathbf{e}_i^*. \tag{15}$$

3.4 Joint Optimization

We optimize the model parameters Θ end-to-end. The objective function consists of the recommendation task loss and an auxiliary regulation loss.

Recommendation Task. We utilize the Bayesian Personalized Ranking (BPR) loss:

$$\mathcal{L}_{rec} = \sum_{(u,i,j)\in\mathcal{D}} -\ln\sigma(\hat{y}_{ui} - \hat{y}_{uj}) \tag{16}$$

Load Balancing. To prevent expert collapse (where the router assigns all nodes to a single expert), we adopt a standard load-balancing loss to encourage different experts to receive comparable routing traffic:

$$\mathcal{L}_{aux} = N_{exp} \sum_{k=1}^{N_{exp}} f_k \cdot P_k \tag{17}$$

where N_{exp} denotes the total number of experts. The term f_k represents the fraction of training samples dispatched to expert k, while P_k is the average routing probability assigned to expert k across the batch. And the final objective is :

$$\mathcal{L} = \mathcal{L}_{rec} + \lambda\mathcal{L}_{aux} + \gamma\|\Theta\|_2^2 \tag{18}$$

where Θ denotes the set of all trainable model parameters, and λ and γ are hyperparameters controlling the strength of load balancing and L_2 regularization, respectively.

4 Experiments

In this section, we present a comprehensive empirical evaluation of DAME to substantiate its effectiveness. Our experiments are designed to address the following four research questions (RQs):

- **RQ1 (Comparative Analysis):** How does DAME perform against state-of-the-art GNN-based recommendation paradigms across diverse data scenarios?

- **RQ2 (Ablation Study):** What are the individual contributions of the proposed Hybrid Structural MoE and RoPE mechanisms?
- **RQ3 (Hyper-parameter Sensitivity):** How do key hyper-parameters, specifically the expert count (N_{exp}) and routing budget (K), govern the model's learning dynamics?
- **RQ4 (Routing Interpretability):** Does the learned routing mechanism exhibit healthy load balancing? Furthermore, can we observe interpretable structural specialization across different propagation depths driven by RoPE?

In the following, we first describe the experimental settings used to evaluate DAME.

4.1 Experimental Setup

Datasets. We conduct experiments on three benchmark datasets: **Gowalla**(Cho et al., 2011), **Yelp**, and **Amazon**(He & McAuley, 2016). These datasets vary significantly in scale and sparsity, providing a robust testbed for evaluating generalization capability(Chen et al., 2025a). Table 1 summarizes the statistics.

Table 1: Statistics of the datasets.

Dataset	#Users	#Items	#Interactions
Gowalla	25,557	19,747	294,983
Yelp	42,712	26,822	182,357
Amazon	76,469	83,761	966,680

Evaluation Metrics. To comprehensively evaluate the recommendation performance, we employ two widely used metrics: Recall@K and NDCG@K, where K is set to $\{20, 40\}$. Instead of random splitting, we utilize the provided training and testing sets for each dataset to ensure a fair and consistent comparison with existing baselines (Chen et al., 2025a). To guarantee reproducibility, all reported results are based on a fixed random seed (1234).

Compared Baselines. We compare DAME with 19 baseline models, providing a comprehensive evaluation against the landscape of recommendation research. These encompass conventional methods like BiasMF (Koren et al., 2009) and NCF (He et al., 2017); GNN-based architectures including AutoR (Sedhain et al., 2015), PinSage (Ying et al., 2018), STGCN (Zhang et al., 2019), GCMC (Van Den Berg et al., 2017), NGCF (Wang et al., 2019), GCCF (Chen et al., 2020), LightGCN (He et al., 2020), and DGCF (Wang et al., 2020); efficient distillation methods SLRec (Yao et al., 2021), GLT (Chen et al., 2021), and UnKD (Chen et al., 2023); as well as state-of-the-art graph contrastive learning methods: SGL (Wu et al., 2021), NCL (Lin et al., 2022), HCCF (Xia et al., 2022), SimGCL (Yu et al., 2022), SimRec (Xia et al., 2023), and LightGNN (Chen et al., 2025a).

Implementation Details. All experiments are implemented in PyTorch 1.12.1 and conducted on a single NVIDIA A100 GPU. We optimize the model using the Adam optimizer to facilitate efficient training. The learning rate is tuned within $\{1e^{-3}, 3e^{-3}\}$ and the weight decay is searched in $\{1e^{-6}, 1e^{-8}\}$. Based on the sensitivity analysis, we adopt $N_{exp} = 32$ experts and Top-$K = 4$ routing as the default setting. The RoPE base frequency is set to $\theta = 10000.0$.

4.2 Performance Comparison (RQ1)

Quantitative Analysis. Table 2 reports the overall performance comparison, where **bold** and underline denote the best and runner-up results, respectively. **DAME** achieves consistent superiority across all datasets, establishing a new state-of-the-art.

Robustness in Sparse Scenarios: On the highly sparse Amazon dataset, DAME achieves a remarkable improvement of approximately 3.52 percentage points in Recall@40 compared to the strongest baseline. This substantial gain validates our "conditional computation" hypothesis: unlike standard GNNs that propagate noise via symmetric aggregation, DAME's expert routing acts as a *dynamic structural filter*. By selectively activating only the Top-K relevant experts, the model effectively

Table 2: Overall performance comparison on Amazon, Gowalla, and Yelp datasets. The best and second-best results are highlighted in **bold** and underline, respectively.

Method	Amazon				Gowalla				Yelp			
	R@20	**N@20**	**R@40**	**N@40**	**R@20**	**N@20**	**R@40**	**N@40**	**R@20**	**N@20**	**R@40**	**N@40**
BiasMF	0.0324	0.0211	0.0578	0.0293	0.0867	0.0579	0.1269	0.0695	0.0198	0.0094	0.0307	0.0120
NCF	0.0367	0.0234	0.0600	0.0306	0.1019	0.0674	0.1563	0.0833	0.0304	0.0143	0.0487	0.0187
AutoR	0.0525	0.0318	0.0826	0.0415	0.1477	0.0690	0.2511	0.0985	0.0491	0.0222	0.0692	0.0268
PinSage	0.0486	0.0317	0.0773	0.0402	0.0985	0.0809	0.1882	0.0994	0.0510	0.0245	0.0743	0.0315
STGCN	0.0583	0.0377	0.0908	0.0478	0.1574	0.1042	0.2318	0.1252	0.0562	0.0282	0.0856	0.0355
GCMC	0.0837	0.0579	0.1196	0.0692	0.1863	0.1151	0.2627	0.1390	0.0584	0.0280	0.0891	0.0360
NGCF	0.0551	0.0353	0.0876	0.0454	0.1757	0.1135	0.2586	0.1367	0.0681	0.0336	0.1019	0.0419
GCCF	0.0772	0.0501	0.1175	0.0625	0.2012	0.1282	0.2903	0.1532	0.0742	0.0365	0.1151	0.0466
LightGCN	0.0868	0.0571	0.1285	0.0697	0.2230	0.1433	0.3181	0.1670	0.0761	0.0373	0.1175	0.0474
DGCF	0.0617	0.0372	0.0912	0.0468	0.2055	0.1312	0.2929	0.1555	0.0700	0.0347	0.1072	0.0437
SLRec	0.0742	0.0480	0.1123	0.0598	0.2001	0.1298	0.2863	0.1540	0.0665	0.0327	0.1032	0.0418
NCL	0.0955	0.0623	0.1409	0.0764	0.2283	0.1478	0.3232	0.1745	0.0806	0.0402	0.1230	0.0505
SGL	0.0874	0.0569	0.1312	0.0704	0.2332	0.1509	0.3251	0.1780	0.0803	0.0398	0.1226	0.0502
HCCF	0.0885	0.0578	0.1335	0.0716	0.2293	0.1482	0.3258	0.1751	0.0789	0.0391	0.1210	0.0492
SimGCL	0.0921	0.0605	0.1367	0.0730	0.2328	0.1506	0.3276	0.1772	0.0788	0.0395	0.1213	0.0498
GLT	0.0901	0.0585	0.1355	0.0725	0.2324	0.1464	0.3269	0.1730	0.0812	0.0400	0.1249	0.0507
UnKD	0.0947	0.0607	0.1376	0.0745	0.2331	0.1496	0.3301	0.1766	0.0819	0.0392	0.1202	0.0493
SimRec	0.1067	0.0734	0.1535	0.0879	0.2434	0.1592	0.3399	0.1865	0.0823	0.0414	0.1251	0.0519
LightGNN	0.1189	0.0820	0.1677	0.0969	0.2610	0.1684	0.3597	0.1962	0.0879	0.0443	0.1328	0.0553
DAME	**0.1468**	**0.1048**	**0.2029**	**0.1242**	**0.2712**	**0.1771**	**0.3742**	**0.2059**	**0.0895**	**0.0451**	**0.1353**	**0.0563**

decouples noisy interactions from core user interests, preventing the “noise amplification" common in static graphs.

Superiority over Contrastive Methods: Furthermore, DAME consistently outperforms graph contrastive learning baselines. This suggests that explicitly encoding high-order connectivity via RoPE is a more robust strategy than stochastic data augmentations (such as edge dropout). While random augmentation may disrupt the fragile topology of sparse graphs, RoPE preserves the deterministic structural integrity while still providing necessary regularization.

Alleviating Over-smoothing: On the denser Gowalla dataset, DAME maintains a clear lead. This demonstrates that RoPE successfully mitigates the over-smoothing issue. By injecting “relative depth" information into the complex vector space, DAME ensures that representations remain distinguishable even after multiple propagation layers, effectively capturing deep collaborative signals without homogenization.

4.3 Ablation Study (RQ2)

To dissect the efficacy of our core contributions, we conduct an ablation study by systematically modifying key components. Note that removing both RoPE and MoE degrades the model to the standard GNN architecture, which has already been compared in RQ1. Therefore, we focus on the two essential variants to verify the individual impact of depth awareness and structural denoising:

- **w/o RoPE:** This configuration replaces the rotary embedding propagation with standard linear aggregation, treating neighbors at different hops indistinguishably.
- **w/o MoE:** This configuration disables the router and expert selection, forcing the model to aggregate information from the full, static adjacency matrix without dynamic pruning.

The results in Table 3 yield several insightful observations regarding the correlation between model components and data characteristics:

- **Impact of RoPE (Depth-Awareness):** As shown in Table 3, removing RoPE results in a consistent performance decline, which is particularly pronounced on the Gowalla dataset. Specifically, Recall@20 drops by approximately 0.59 percentage points (compared to a 0.37 percentage-point drop for w/o MoE). In dense graphs with rich high-order connectivity, explicitly encoding "relative depth" prevents information homogenization, ensuring the model can distinguish direct user interests from broad community trends.
- **Impact of MoE (Structural Denoising):** The impact of the MoE component is evident from the comparison between the DAME (Full) and the w/o MoE variant. As shown in Table 3, removing the MoE module leads to a consistent performance drop across all three datasets. This observation underscores the effectiveness of the mixture-of-experts architecture in capturing diverse user interests and complex item correlations by leveraging multiple specialized experts.

Table 3: Ablation study on key components (MoE and RoPE).

Variant	Gowalla		Amazon		Yelp	
	R@20	**N@20**	**R@40**	**N@40**	**R@40**	**N@40**
DAME (Full)	**0.2712**	**0.1771**	**0.2029**	**0.1242**	**0.1353**	**0.0563**
w/o RoPE (MoE only)	0.2653	0.1734	0.2021	0.1237	0.1345	0.0558
w/o MoE (RoPE only)	0.2675	0.1750	0.2017	0.1238	0.1350	0.0561

4.4 Parameter Sensitivity Analysis (RQ3)

We further investigate the impact of the expert capacity (N_{exp}) and the routing budget (K). The experimental results on Gowalla are summarized in Table 4.

As illustrated in Table 4, increasing the number of experts initially boosts performance, peaking at $N_{exp} = 32$. However, further scaling to 64 experts yields diminishing returns or even regression. Similarly, regarding the routing strategy, we identify $K = 4$ as the optimal point.

This phenomenon points to a **"Bounded Collaboration"** mechanism: efficient graph representation learning requires a balance between sufficient structural capacity and focused aggregation. Excessive experts or routing paths may introduce structural noise, diluting the user's core interests.

Table 4: Sensitivity analysis of expert count N_{exp} and routing budget K across different datasets.

N_{exp}	K	Gowalla		Yelp		Amazon	
		R@20	N@20	R@40	N@40	R@40	N@40
8	4	0.2658	0.1715	0.1335	0.0554	0.1932	0.1169
16	4	0.2667	0.1728	0.1341	0.0558	0.1952	0.1186
32	4	**0.2712**	**0.1771**	**0.1353**	**0.0563**	**0.2029**	**0.1242**
64	4	0.2655	0.1716	0.1346	0.0558	0.1950	0.1180
32	1	0.2670	0.1713	0.1352	0.0559	0.1952	0.1182
32	2	0.2665	0.1710	0.1346	0.0558	0.1980	0.1201
32	8	0.2663	0.1709	0.1346	0.0558	0.1989	0.1206

4.5 Routing Interpretability and Expert Specialization (RQ4)

A common challenge in sparse Mixture-of-Experts is expert collapse or purely capacity-driven specialization. We therefore evaluate whether users routed to the same expert are semantically closer than users routed to different experts. Table 5 reports three post-hoc metrics at the deepest layer. Same-expert users share substantially more items than cross-expert users, low- and high-activity groups show different routing distributions, and the top items served by different experts have small

overlap. These results suggest that routing learns meaningful behavioral specialization rather than only increasing model capacity. Additional layer-wise analyses and expert-load visualizations are provided in Appendix B and Appendix E.

Table 5: Routing specialization at the deepest layer. Larger Intra/Inter and JSD/Random indicate stronger specialization; lower Top-50 overlap indicates less expert redundancy.

Dataset	Intra/Inter	JSD/Random	Top-50 Overlap
Gowalla	$5.7\times$	$1.45\times$	5.5%
Amazon	$5.7\times$	$1.50\times$	5.6%
Yelp	$31.5\times$	$1.16\times$	3.0%

4.6 Stability and Depth Analysis

We further validate the robustness of DAME over five random seeds. Table 6 reports the mean and standard deviation, showing that performance remains stable across datasets and evaluation cutoffs.

Table 6: DAME performance over five random seeds.

Dataset	Metric@K	Recall	NDCG
Gowalla	@20	0.2698 ± 0.0015	0.1759 ± 0.0011
Yelp	@40	0.1346 ± 0.0010	0.0559 ± 0.0004
Amazon	@20	0.1479 ± 0.0016	0.1057 ± 0.0016

To directly examine whether RoPE becomes more useful in deeper propagation, we measure inter-layer cosine similarity on Gowalla. As shown in Table 7, the reduction grows from 0.5% at two layers to 31.1% at eight layers, confirming that RoPE increasingly helps keep high-order representations distinguishable as depth increases.

Table 7: Inter-layer cosine similarity under increasing depth on Gowalla. Lower values indicate less layer collapse.

Layers	GCN Cos.	DAME Cos.	Reduction
2	0.887	0.882	0.5%
3	0.848	0.791	6.7%
4	0.808	0.722	10.7%
5	0.788	0.623	21.0%
6	0.771	0.582	24.5%
8	0.736	0.507	31.1%

5 Conclusion

This paper proposes a novel framework called DAME, aiming to address the dual challenges of modeling highly heterogeneous graph interactions and mitigating the oversmoothing of GNNs. Specifically, DAME proactively filters structural noise using diverse expert subgraphs, while incorporating relative position encoding to preserve inter-layer dependencies and enhance deep feature representations. Extensive experiments demonstrate that DAME significantly outperforms state-of-the-art baselines in recommendation performance and robustness to heterogeneity, all while maintaining controllable computational overhead. In future work, we will leverage the conditional computation features of DAME to explore lower-cost training and inference strategies, paving the way for efficient, robust, and scalable graph recommendation systems.

AI USE STATEMENT

We used generative AI tools to assist with language polishing and improve the clarity and readability of the manuscript. The authors have reviewed the AI-assisted revisions and take full responsibility for the final content of this work.

REFERENCES

Vineeta Anand and Ashish Kumar Maurya. A survey on recommender systems using graph neural network. *ACM Transactions on Information Systems*, 43(1):1–49, 2025.

Wanlin Cai, Yuxuan Liang, Xianggen Liu, Jianshuai Feng, and Yuankai Wu. Msgnet: Learning multi-scale inter-series correlations for multivariate time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 11141–11149, 2024.

Gang Chen, Jiawei Chen, Fuli Feng, Sheng Zhou, and Xiangnan He. Unbiased knowledge distillation for recommendation. In *Proceedings of the sixteenth ACM international conference on web search and data mining*, pp. 976–984, 2023.

Guoxuan Chen, Lianghao Xia, and Chao Huang. Lightgnn: Simple graph neural network for recommendation. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, pp. 549–558, 2025a.

Haibo Chen, Xin Wang, Zeyang Zhang, Haoyang Li, Ling Feng, and Wenwu Zhu. AutoGFM: Automated graph foundation model with adaptive architecture customization. In *Forty-second International Conference on Machine Learning*, 2025b.

Lei Chen, Le Wu, Richang Hong, Kun Zhang, and Meng Wang. Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 27–34, 2020.

Tianlong Chen, Yongduo Sui, Xuxi Chen, Aston Zhang, and Zhangyang Wang. A unified lottery ticket hypothesis for graph neural networks. In *International conference on machine learning*, pp. 1695–1706. PMLR, 2021.

Ziheng Chen, Jin Huang, Fabrizio Silvestri, Yongfeng Zhang, Hongshik Ahn, and Gabriele Tolomei. Joint factual and counterfactual explanations for top-k gnn-based recommendations. *ACM Transactions on Recommender Systems*, 2025c.

Eunjoon Cho, Seth A Myers, and Jure Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1082–1090, 2011.

Yizhou Dang, Yuting Liu, Enneng Yang, Guibing Guo, Linying Jiang, Jianzhe Zhao, and Xingwei Wang. Efficient and adaptive recommendation unlearning: A guided filtering framework to erase outdated preferences. *ACM Transactions on Information Systems*, 43(2):1–25, 2025.

Chendi Ge, Xin Wang, Ziwei Zhang, Yijian Qin, Hong Chen, Haiyang Wu, Yang Zhang, Yuekui Yang, and Wenwu Zhu. Behavior importance-aware graph neural architecture search for cross-domain recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 11708–11716, 2025.

Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pp. 507–517, 2016.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, pp. 173–182, 2017.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pp. 639–648, 2020.

Xiaoyu Huang, Weidong Chen, Bo Hu, and Zhendong Mao. Graph mixture of experts and memory-augmented routers for multivariate time series anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 17476–17484, 2025.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

Qian Li, Minghui Ouyang, Tian Ding, Yuyi Wang, Qingjiang Shi, and Ruoyu Sun. Towards explaining the power of constant-depth graph neural networks for structured linear programming. In *The Thirteenth International Conference on Learning Representations*, 2025a.

Wenxin Li, Xiao Song, and Yuchun Tu. Graphdrl: Gnn-based deep reinforcement learning for interactive recommendation with sparse data. *Expert Systems with Applications*, 273:126832, 2025b.

Jianqing Liang, Xinkai Wei, Min Chen, Zhiqiang Wang, and Jiye Liang. Gnn-transformer cooperative architecture for trustworthy graph contrastive learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 18667–18675, 2025a.

Shunpan Liang, Qianjin Kong, Yu Lei, and Chen Li. Graphical contrastive learning for multi-interest sequential recommendation. *Expert Systems with Applications*, 259:125285, 2025b.

Zihan Lin, Changxin Tian, Yupeng Hou, and Wayne Xin Zhao. Improving graph collaborative filtering with neighborhood-enriched contrastive learning. In *Proceedings of the ACM web conference 2022*, pp. 2320–2329, 2022.

Ziyang Liu, Chaokun Wang, Shuwen Zheng, Cheng Wu, Kai Zheng, Yang Song, and Na Mou. Pone-gnn: integrating positive and negative feedback in graph neural networks for recommender systems. *ACM Transactions on Recommender Systems*, pp. 1–23, 2025.

Lei Sang, Weichen Fei, Yi Zhang, Yuee Huang, and Yiwen Zhang. Heterogeneous adaptive preference learning for recommendation. *ACM Transactions on Recommender Systems*, pp. 1–25, 2025.

Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th International Conference on World Wide Web*, pp. 111–112, 2015.

Lei Shi, Jiapeng Yang, Pengtao Lv, Lu Yuan, Feifei Kou, Jia Luo, and Mingying Xu. Self-derived knowledge graph contrastive learning for recommendation. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 7571–7580, 2024.

Rianne Van Den Berg, N Kipf Thomas, and Max Welling. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263*, 2(8):9, 2017.

Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, pp. 165–174, 2019.

Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. Disentangled graph collaborative filtering. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pp. 1001–1010, 2020.

Yuhan Wang, Qing Xie, Zhifeng Bao, Mengzi Tang, Lin Li, and Yongjian Liu. Enhancing transferability and consistency in cross-domain recommendations via supervised disentanglement. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, pp. 104–113, 2025.

Cheng Wu, Chaokun Wang, Jingcao Xu, Ziyang Liu, Kai Zheng, Xiaowei Wang, Yang Song, and Kun Gai. Graph contrastive learning with generative adversarial network. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 2721–2730, 2023.

Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pp. 726–735, 2021.

Lianghao Xia, Chao Huang, Yong Xu, Jiashu Zhao, Dawei Yin, and Jimmy Huang. Hypergraph contrastive collaborative filtering. In *Proceedings of the 45th International ACM SIGIR conference on research and development in information retrieval*, pp. 70–79, 2022.

Lianghao Xia, Chao Huang, Jiao Shi, and Yong Xu. Graph-less collaborative filtering. In *Proceedings of the ACM web conference 2023*, pp. 17–27, 2023.

Meng Yan, Cai Xu, Xujing Wang, Ziyu Guan, Wei Zhao, and Yuhang Zhou. Conf-gnnrec: Quantifying and calibrating the prediction confidence for gnn-based recommendation methods. In *Companion Proceedings of the ACM on Web Conference 2025*, pp. 1460–1464, 2025.

Zhe Yang and Tiantian Liang. Multi-graph co-training for capturing user intent in session-based recommendation. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 1377–1386, 2025.

Tiansheng Yao, Xinyang Yi, Derek Zhiyuan Cheng, Felix Yu, Ting Chen, Aditya Menon, Lichan Hong, Ed H Chi, Steve Tjoa, Jieqi Kang, et al. Self-supervised learning for large-scale item recommendations. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pp. 4321–4330, 2021.

Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 974–983, 2018.

Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pp. 1294–1303, 2022.

Jiani Zhang, Xingjian Shi, Shenglin Zhao, and Irwin King. Star-gcn: Stacked and reconstructed graph convolutional networks for recommender systems. *arXiv preprint arXiv:1905.13129*, 2019.

Zhipeng Zhang, Youfu Zhu, Mianxiong Dong, Kaoru Ota, Yao Zhang, and Yonggong Ren. Embedding guarantor: Knowledge-enhanced graph learning for new item cold-start recommendation. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2025a.

Zhongjian Zhang, Xiao Wang, Huichi Zhou, Yue Yu, Mengmei Zhang, Cheng Yang, and Chuan Shi. Can large language models improve the adversarial robustness of graph neural networks? In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 2008–2019, 2025b.

Chongjie Zhao, Jinyan Wang, Linlin Su, Zeming Gan, and Ziyang Zhou. Robust heterogeneous gnns via semantic attention and contrastive learning. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pp. 4326–4335, 2025a.

Yushang Zhao, Yike Peng, Dannier Li, Yuxin Yang, Chengrui Zhou, and Jing Dong. Research on personalized financial product recommendation by integrating large language models and graph neural networks. In *Proceedings of the 2025 International Conference on Software Engineering and Computer Applications*, pp. 53–59, 2025b.

A Technical appendices and supplementary material

This appendix contains additional analyses, diagnostic experiments, and implementation details that support the main paper. The main submission is designed to stand alone; these materials provide optional evidence for routing behavior, RoPE's effect on over-smoothing, dataset sparsity, and model complexity.

Roadmap.

- Appendix B: Expert specialization and routing interpretability
- Appendix C: RoPE and over-smoothing analysis
- Appendix D: Dataset sparsity and Top-K sensitivity
- Appendix E: Additional visualizations
- Appendix F: Complexity analysis
- Appendix G: Limitations

B Expert Specialization and Routing Interpretability

This section provides post-hoc analyses for whether routed experts learn meaningful behavioral groups rather than merely increasing model capacity. We analyze users assigned to the same expert versus users assigned to different experts at the deepest layer unless otherwise specified.

Table 8: Expert specialization at the deepest layer. Intra/Inter measures the ratio between item overlap of same-expert user pairs and cross-expert user pairs.

Dataset	Intra/Inter	KL(Low, High)	JSD/Random	Top-50 Overlap
Gowalla	$5.7\times$ ($p < 10^{-71}$)	0.76	$1.45\times$	5.5%
Amazon	$5.7\times$ ($p < 10^{-22}$)	0.51	$1.50\times$	5.6%
Yelp	$31.5\times$ ($p < 10^{-14}$)	0.84	$1.16\times$	3.0%

Table 9: Layer-wise intra/inter Jaccard similarity ratio. Larger values indicate stronger same-expert preference clustering.

Dataset	L0	L1	L2	L3
Gowalla	$2.8\times$	$3.7\times$	$6.3\times$	$5.7\times$
Amazon	$2.7\times$	$3.9\times$	$7.2\times$	$5.7\times$
Yelp	$6.0\times$	$5.7\times$	$8.5\times$	$31.5\times$

Table 10: Example routing divergence across user activity groups on Gowalla. Cold and active users select different dominant experts.

User Group	Top-3 Experts (L0)	Top-3 Experts (L1)
Cold (≤ 3 interactions)	E22, E16, E18	E30, E5, E3
Active (≥ 16 interactions)	E3, E2, E30	E19, E12, E31

C RoPE and Over-Smoothing Analysis

We measure over-smoothing by inter-layer cosine similarity, where lower similarity indicates more distinguishable representations across propagation depths. Table 12 compares DAME with and without RoPE across datasets, complementing the depth-wise analysis in Section 4.6.

D Dataset Sparsity and Top-K Sensitivity

The relatively flat Top-K sensitivity on Yelp is partly explained by user sparsity. As shown in Table 13, most Yelp users have very few interactions, making fine-grained routing less visible in aggregate ranking metrics. Active users still exhibit stronger intent clustering than sparse users.

Table 11: Representative expert roles at layer L1 on Gowalla.

Expert	#Users	Avg. Degree	Interpreted Role
E5	1630	6.8	Cold-start
E4	1164	19.8	Active
E3	667	7.1	Cold-start
E2	580	8.0	Low-activity

Table 12: Inter-layer cosine similarity with and without RoPE. Lower values indicate less layer collapse.

Dataset	With RoPE	Without RoPE	Reduction
Gowalla	0.765	0.889	14.0%
Amazon	0.798	0.911	12.4%
Yelp	0.938	0.967	3.0%

E Additional Visualizations

This section provides the expert load distributions for all datasets. These figures complement the quantitative routing analyses in Appendix B.

F Complexity Analysis

The dominant cost of DAME comes from sparse graph propagation. In a fully sparse dispatch implementation, each node consumes only the Top-K routed expert outputs. In our full-batch implementation, an expert is evaluated if it is selected by at least one node in the batch, giving the worst-case per-layer propagation cost

$$\mathcal{O}((1 + N_{exp})|\mathcal{E}|d). \tag{19}$$

The router and node-wise mixture add $\mathcal{O}((|\mathcal{U}| + |\mathcal{I}|)N_{exp} + (|\mathcal{U}| + |\mathcal{I}|)Kd)$ operations. The extra parameters consist of router matrices and edge-mask logits. For L layers, the edge masks require $\mathcal{O}(L(N_{exp} + 1)|\mathcal{E}|)$ scalar gates over existing edges.

G Limitations

DAME may be less effective when user intents are highly ambiguous or when interaction histories are extremely sparse, because edge masks and routers receive limited evidence for reliable structural specialization. The current implementation also stores expert-specific edge gates, adding storage overhead relative to LightGCN. Incorporating side information, designing lower-overhead sparse dispatch, and improving expert interpretability are promising directions for future work.

Table 13: Cross-dataset sparsity and active-user intent clustering.

Dataset	Sparse Users (≤ 3)	Active Users (≥ 10)	Active/Sparse Clustering
Gowalla	24.3%	34.7%	3.44×
Amazon	9.8%	40.6%	2.43×
Yelp	61.6%	7.1%	1.84×

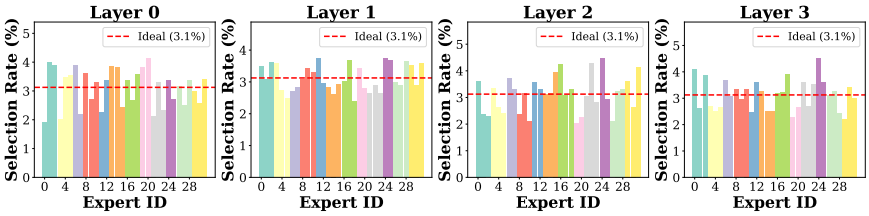

Figure 3: The chosen expert distribution on Amazon test users.

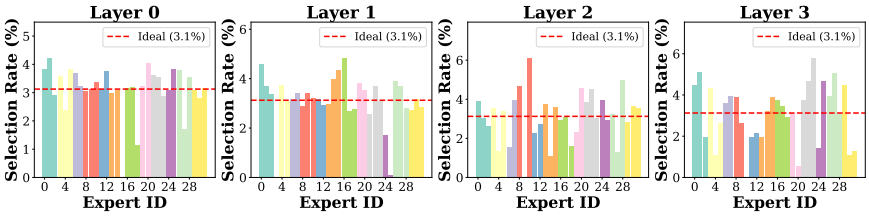

Figure 4: The chosen expert distribution on Yelp test users.

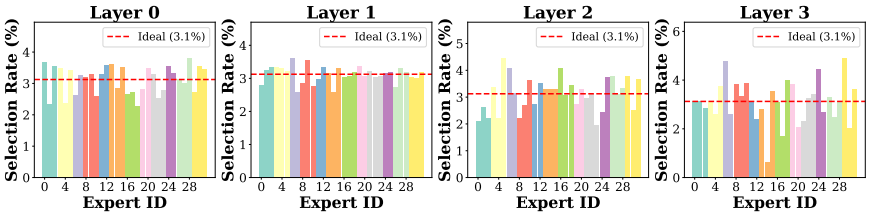

Figure 5: The chosen expert distribution on Gowalla test users.