

HGT4REC: HYPERBOLIC GRAPH TRANSFORMER FOR SEQUENTIAL AND SOCIAL RECOMMENDATION

Rungthip Cobal¹, Jintana Polsri^{1,*}, Phatthira Keawkerd^{1,*}, Nikorn Kannikaklang¹, Poorivat Kampeerapaappat¹, Supasee Duangsai¹, Suratep Pangerd¹, Natratanon Kanraweekultana¹, Wassana Duangmeun¹, Jessadaporn Yanuphrom², Piranun Chantavirod³, Sukanya Kittikhunngam³, Suphattana Tachochalalai², and Chinapat Sakunrasrisuay⁴

Received: October 17, 2025; Revised: February 02, 2026; Accepted: February 05, 2026

Abstract

Sequential behaviors and social ties jointly shape user preferences; however, most prior work models them in isolation and relies on shallow fusion in Euclidean space, which struggles to capture temporal drift and hierarchical social structure. We propose a novel framework; HGT4Rec, a Hyperbolic Graph Transformer for Sequential and Social Recommendation. A graph transformer encodes item-transition dependencies to track evolving preferences along the sequence, while a hyperbolic transformer operates on the social graph to represent long-range and hierarchical influence. We further introduce FusionGRU, an adaptive gating module that integrates the two representations into a unified preference state for Top-K prediction. Experiments on Yelp, iFashion, LastFM show that HGT4Rec delivers substantial improvements over the strongest baseline, achieving +348.22% / +314.86%, +383.21% / +210.19%, and +90.85% / +9.83% in Recall@10 and NDCG@10, respectively. Our results demonstrate the value of combining graph transformers with hyperbolic space modeling and the gated fusion for next-item recommendation in sequential and social settings (our code).

Keywords: Gated Recurrent Unit; Graph Transformer; Hyperbolic Geometry; Sequential Recommendation; Social Recommendation

¹ Major of Information Technology and Digital Business, Faculty of Business Administration, Rajamangala University of Technology Krungthep, Bangkok, Thailand. Email: rungthip.c@mail.rmuk.ac.th, ORCID: 0009-0002-3214-3009; jintana.p@mail.rmuk.ac.th, ORCID: 0009-0007-6139-3767; phatthira.k@mail.rmuk.ac.th, ORCID: 0009-0008-4882-0012; nikorn.k@mail.rmuk.ac.th, ORCID: 0009-0003-9966-3524; poorivat.k@mail.rmuk.ac.th, ORCID: 0009-0001-4981-2195; supasee.d@mail.rmuk.ac.th, ORCID: 0009-0007-8443-367X; suratep.p@mail.rmuk.ac.th, ORCID: 0009-0007-9519-5689; natratanon.k@mail.rmuk.ac.th, ORCID: 0009-0006-0804-291X; wassana.d@mail.rmuk.ac.th, ORCID: 0009-0003-9862-6617

² Major of Finance, Faculty of Business Administration, Rajamangala University of Technology Krungthep, Bangkok, Thailand. Email: jessadaporn.y@mail.rmuk.ac.th, ORCID: 0009-0005-2811-0855; suphattana.t@mail.rmuk.ac.th, ORCID: 0009-0007-4617-5382

³ Major of Accounting, Faculty of Business Administration, Rajamangala University of Technology Krungthep, Bangkok, Thailand. Email: piranun.c@mail.rmuk.ac.th, ORCID: 0009-0008-0917-2809; sukanya.k@mail.rmuk.ac.th, ORCID: 0009-0002-7918-8917

⁴ Department of Information Technology and Computer Innovation, Faculty of Management Science and Information Technology, Nakhon Phanom University, Nakhon Phanom, Thailand. Email: chinapat.sakun@npu.ac.th, ORCID: 0009-0007-8594-4762

* Corresponding author

Introduction

Personalized recommendation underpins modern digital platforms, where users interact with items over time and are simultaneously influenced by their social connections (Wu *et al.*, 2023), (Kannikaklang *et al.*, 2024), (Huang *et al.*, 2024). These two signals—sequential behaviors and social relations—are complementary: the former captures short- and mid-term intent through item-transition patterns, while the latter conveys long-range, often hierarchical influence among users. Effective recommenders must therefore model temporal preference drift and structured social effects jointly, and fuse them into a single, actionable preference state for prediction.

Existing approaches fall short in three ways. First, many methods treat sequential and social signals in isolation, or connect them through shallow fusion (e.g., concatenation or linear blending), which underutilizes their complementary inductive biases (e.g., SSSER (Xu *et al.*, 2019), SBRec (Xie *et al.*, 2023)). Second, most transformer- or GNN-based models operate purely in Euclidean space (e.g., GaGNN (Huang *et al.*, 2024), SASRec (Kang & McAuley, 2018), BiG (Kannikaklang & Wongthanavas, 2025)). While Euclidean attention excels at modeling sequential dependencies, it struggles to represent hierarchical and tree-like structures that are ubiquitous in social graphs without resorting to deep stacks or many heads. Third, when both signals are considered, the geometry mismatch between modules (e.g., Euclidean sequence encoders and geometry-agnostic social encoders) hinders stable optimization and expressive fusion, especially under temporal drift.

Hyperbolic geometry has recently shown promise for representing hierarchical relations with low distortion, suggesting a natural fit for social influence modeling (Nickel & Kiela, 2017). Yet, prior hyperbolic recommenders are typically static or pair hyperbolic embeddings with Euclidean attention, leaving the attention mechanism itself geometry-agnostic. This limits the ability to capture multi-hop, long-range, and time-varying social effects within a unified, end-to-end architecture.

This work proposes HGT4Rec, a Hyperbolic Graph Transformer for sequential and social recommendation. HGT4Rec pairs a graph transformer for sequences with a hyperbolic transformer for social influence, and integrates their outputs via a geometry-aware FusionGRU. Concretely, the sequential branch applies graph-enhanced self-attention to encode item-transition dependencies and track evolving preferences along

the user’s interaction history. The social branch performs manifold-aware self-attention in hyperbolic space using Möbius linear layers and exp/log maps, enabling expressive, curvature-aware aggregation of hierarchical, long-range social signals over time. FusionGRU then aligns manifold-specific representations and adaptively fuses them into a unified preference state for Top-K prediction.

Figure 1 illustrates a toy example of how sequential and social signals jointly drive a next-item recommendation. u_2 is the target user, and u_1/u_3 are u_2 ’s social neighbors. u_2 ’s recent consumption sequence is $i_1 \rightarrow i_2 \rightarrow i_3$. The friends’ sequences are $u_1: i_2 \rightarrow i_3 \rightarrow i_5$ and $u_3: i_4 \rightarrow i_3 \rightarrow i_6$. The model examines what friends are next under similar context windows: after $i_2 \rightarrow i_3$, u_1 proceeded to i_5 , raising i_5 ’s social score; after $i_4 \rightarrow i_3$, u_3 proceeded to i_6 , raising i_6 ’s social score. Social signals are weighted by factors such as sequence similarity, recency, and tie strength; because u_1 shares the subsequence $i_2 \rightarrow i_3$ with u_2 , u_1 ’s influence is stronger than u_3 ’s in this context. Fusing these social cues with u_2 ’s own sequential signal yields the highest combined score for i_5 , which is therefore recommended as the next item.

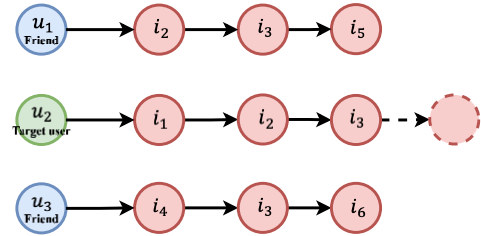


Figure 1. A toy example of how sequential and social signals jointly drive a next-item recommendation

Our main contributions are as follows:

- Graph Transformer (GT) for sequential dynamics; We integrate structural priors (e.g., positional and edge encodings) into a transformer over item-transition graphs to model evolving user intent.
- Hyperbolic Transformer (HT) for social influence; We formulate a transformer block natively in hyperbolic space, with curvature-aware attention and positional/structural encodings, to capture hierarchical and long-range social effects under temporal drift.

- Geometry-aware FusionGRU; We introduce a gating module that aligns Euclidean and hyperbolic representations via exp/log maps and performs adaptive fusion, yielding a stable and interpretable unified preference state.
- End-to-end framework and analysis; We present an efficient training objective for Top-K recommendation and provide ablations that isolate the gains from GT, HT, and FusionGRU.
- Empirical validation; Extensive experiments on multiple public benchmarks demonstrate that HGT4Rec consistently outperforms strong baselines; sensitivity and ablation studies further verify each component's contribution.

Related Works

We review two strands of prior work most relevant to our approach: hyperbolic geometry for recommendation and graph-transformer-based recommendation.

Hyperbolic Geometry for Recommendation

Hyperbolic representation learning has emerged as a powerful tool for modeling hierarchical and tree-like relations with low distortion, e.g., Poincaré embeddings (Nickel & Kiela, 2017), hyperbolic neural networks (Ganea *et al.*, 2018), hyperbolic GCN (Chami *et al.*, 2019), and constant-curvature GNNs (Bachmann *et al.*, 2020). In recommendation, these ideas have been adapted to collaborative filtering and social recommendation by embedding users/items or propagating messages on manifolds, often via exp/log maps and Riemannian optimization, e.g., BiHGCA (Kannikaklang *et al.*, 2024), HGSR (Yang *et al.*, 2024), BiTG4Rec (Kannikaklang & Wongthanavas, 2025), HSR (Li *et al.*, 2022). Such models report gains when user-item or social graphs exhibit hierarchy or long-range dependencies.

Our difference, rather than using hyperbolic space only for embeddings or message passing, HGT4Rec instantiates hyperbolic self-attention inside the transformer block (queries/keys/values via Möbius linear layers with exp/log maps) and makes the social encoder curvature-aware and time-sensitive, enabling multi-hop, long-range social influence modeling within a single end-to-end architecture.

Graph Transformer Recommendation

Self-attention has become a de-facto tool for sequential recommendation (e.g., SASRec (Kang &

McAuley, 2018), and BERT4Rec (Sun *et al.*, 2019)). Graph-based recommenders model collaborative signals with message passing on user-item or session graphs (e.g., NGCF (Wang *et al.*, 2019), LightGCN (He *et al.*, 2020), GCCF (Chen *et al.*, 2020), SGL (Wu *et al.*, 2021), and HCCF (Xia *et al.*, 2022)). Recent GTs inject structural priors (positional/edge encodings, spatial biases) into attention (e.g., Graphormer (Ying *et al.*, 2021)) and have been adapted to recommendation (e.g., GFormer (Li *et al.*, 2023)).

Our difference, we adopt a graph-enhanced transformer specifically for sequence dynamics (item-transition priors and positional/edge encodings) while delegating hierarchical social reasoning to a HT, avoiding the limitations of a single Euclidean geometry for both signals.

Preliminary

In the initial part, we will begin by defining the problem. Next, sequential graph construction and social hyperbolic construction are explained to understand preliminary concept of HGT4Rec.

Problem Definition

Let $\mathcal{U}$ and $\mathcal{I}$ denote the sets of users and items, respectively. For user $u \in \mathcal{U}$, we observe a chronologically ordered interaction sequence $S_u = \langle i_1, i_2, i_3, \dots, i_T \rangle$, $i_t \in \mathcal{I}$, optionally timestamped by $\{t_1 < \dots < t_T\}$. The task is next-item prediction: for each prefix $\langle i_1, i_2, i_3, \dots, i_t \rangle$, learn a scoring function $s(u, t, i)$ over $i \in \mathcal{I}$ so that the ground-truth next item i_{t+1} is ranked near the top. We evaluate with Recall@K and NDCG@K. We denote bold lowercase letters are vectors, and uppercase letters are matrices. The symbols of this article are explained in Table 1.

Sequential Graph Construction

Beyond raw sequences, we build a global item-transition graph $G_{seq} = (\mathcal{I}, \mathcal{E}_{seq})$ that aggregates local order information across users. For a fixed window size w , we add a directed edge ($i \rightarrow j$) whenever j appears within w positions after i in any user's sequence (A_{ij}); its weight counts such transitions:

$$A_{ij} = \sum_{u \in \mathcal{U}} \sum_t 1[i_t = i] \sum_{\tau=1}^w 1[i_{t+\tau} = j] \rho(\tau) \quad (1)$$

where $\rho(\tau) \in (0, 1]$ is a distance decay. We normalize with $\tilde{A} = D^{-1/2} A D^{-1/2}$ where $D_{ii} = \sum_j A_{ij}$. During training, the per-prefix attention is masked by the sequential order (causality), while

global graph G_{seq} supplies structural biases (edge/position encodings) to the GT similar in spirit to Graphormer-style priors (Ying *et al.*, 2021), (Li *et al.*, 2023).

Table 1. Notations and Explanations

Notation	Explanation
$\mathcal{U}, I$	user set and item set
$u \in \mathcal{U}, i \in I$	userID and itemID
G_{seq}	global item-transition graph
A_{ij}	adjacency user’s sequence of item j appearance within window w positions after item i
$\rho(\tau)$	distance-decay for transitions at lag τ
$\mathbb{D}_c^d$	Poincaré ball of curvature $-c$
$exp_0^c(\cdot), log_0^c(\cdot)$	exponential/logarithmic maps at the origin
$d_c(x, y)$	hyperbolic distance
d	the embedding dimension
E_u, E_i	user and item embedding tables
X_t	stacked sequence embeddings up to time t
$\alpha_{ij}^{(\ell, h)}$	attention logit (seq branch) for token $i \leftarrow j$
L_s, L_h	the number of layers in GT / HT
$\beta_{uv}^{(\ell)}$	social attention bias (time/degree aware)
h_t^{seq}	final sequential state at time t (from GT)
h_t^{soc}	final social state at time t (tangent-mapped)
x_t	FusionGRU input
h_{t-1}, h_t	previous and current fused preference state
z_t, r_t	update and reset gates of FusionGRU
$s(u, t, i)$	matching score for user u , state at t , and item i

Social Hyperbolic Construction

To capture hierarchical, power-law and multi-hop effects in social ties, we embed users on a Poincaré ball of curvature $-c$ (with $c > 0$) (Nickel & Kiela, 2017), (Ganea *et al.*, 2018), (Chami *et al.*, 2019), (Bachmann *et al.*, 2020) to capture hierarchical and long-range social effects:

$$\mathbb{D}_c^d = \{x \in \mathbb{R}^d : \|x\| < \frac{1}{\sqrt{c}}\} \quad (2)$$

where $\mathbb{D}_c^d$ denotes Poincaré ball of curvature $-c$. c is Positive curvature parameter (manifold curvature $-c$). $\|\cdot\|$ signifies Euclidean norm.

We use gyrovector operations for hyperbolic geometry (Ganea *et al.*, 2018) to define algebra on the manifold. Möbius addition for $x, y \in \mathbb{D}_c^d$ is:

$$(x \oplus_c y) = \frac{(1 + 2c(x, y) + c\|y\|^2)x + (1 - c\|x\|^2)y}{1 + 2c(x, y) + c^2\|x\|^2\|y\|^2} \quad (3)$$

where $\oplus_c$ represents Möbius addition on $\mathbb{D}_c^d$. Möbius scalar multiplication is:

$$(r \otimes_c x) = \frac{\tanh(r \operatorname{artanh}(\sqrt{c}\|x\|))}{\sqrt{c}\|x\|} x \quad (4)$$

where $r \in \mathbb{R}$. $\otimes_c$ denotes Möbius scalar multiplication. $\tanh(\cdot)$ is hyperbolic tangent activation.

Exponential map at the origin is:

$$\exp_0^c(v) = \tanh(\sqrt{c} \|v\|) \frac{v}{\sqrt{c}\|v\|} \quad (5)$$

where $\exp_0^c(\cdot)$ is exponential map at the origin. Logarithmic map at the origin is:

$$\log_0^c(x) = \frac{1}{\sqrt{c}} \operatorname{artanh}(\sqrt{c} \|x\|) \frac{x}{\|x\|} \quad (6)$$

where $\log_0^c(\cdot)$ is logarithmic map at the origin. The hyperbolic distance can be written via

Möbius addition as:

$$d_c(x, y) = \frac{2}{\sqrt{c}} \operatorname{artanh}(\sqrt{c} \|(-x) \oplus_c y\|) \quad (7)$$

where $d_c(x, y)$ represents hyperbolic distance on $\mathbb{D}_c^d$.

We attach optional temporal weight to social edges (u, v) when timestamp τ_{uv} are available:

$$w_{uv} = \exp(-\lambda(t - \tau_{uv})) \quad (8)$$

where λ is temporal decay coefficient for social edges: $\lambda \geq 0$, which will act as biases in hyperbolic attention. w_{uv} denotes time-decayed social edge weight for (u, v) . τ_{uv} as timestamp of the interaction/edge (u, v) . Manifold parameters are updated with Riemannian optimization (Riemannian Adam) while Euclidean parameter use Adam (Ganea *et al.*, 2018), (Chami *et al.*, 2019).

Methodology

Our research introduces the innovative HGT4Rec framework, illustrated in Figure 2, consisting of five main layers: 1) the embedding layer, 2) the GT for sequential dynamics (Euclidean branch), 3) the HT for social influence (manifold branch), 4) the FusionGRU for geometry-aware temporal fusion, and 5) the prediction layer. We train the model end-to-end with a sampled-softmax objective plus an interior-ball regularizer for manifold stability. The report is organized into six subsections, outlining a new method to address evolving user preferences. The first subsection introduces the embedding layer. The second subsection presents the GT for sequential recommendation. The third subsection explores HT for social recommendation. The fourth subsection presents the novel FusionGRU, which integrates sequential and social preferences. The

fifth subsection details the process of predicting user ratings using specific formulas. The sixth subsection explains the model's optimization to fine-tune through relevant equations.

Let $E_u \in \mathbb{R}^{|U| \times d}$, $E_i \in \mathbb{R}^{|I| \times d}$ be user/item embeddings; and optional temporal embeddings $E_t \in \mathbb{R}^{T \times d}$. We denote by X_t the stacked item-token embeddings up to step t .

Embedding Layer

For a training triple $(u, \langle i_1, \dots, i_t \rangle, i_{t+1})$:

- 1) Sequential branch inputs. Item tokens are embedded as:

$$x_k = E_i[i_k] + p_k \quad (9)$$

where x_k is token embedding for item i_k (with position code). p_k is a learnable position code (absolute and/or relative). Stack $X_t = [x_1, \dots, x_t] \in \mathbb{R}^{t \times d}$.

- 2) Social branch maps users from the tangent (Euclidean) space to the Poincaré ball $\mathbb{D}_c^d$ via the exponential map:

$$z_u^{(0)} = \exp_0^c(E_u[u]), \quad z_v^{(0)} = \exp_0^c(E_u[v]) \quad (10)$$

where $\forall v \in \mathcal{N}(u)$ (Ganea *et al.*, 2018), (Bachmann *et al.*, 2020). $z_u^{(0)}$ and $z_v^{(0)}$ represents the user embeddings mapped to $\mathbb{D}_c^d$.

Graph Transformer for Sequential Recommendation (Euclidean)

We apply L_s layer graph-aware transformer to X_t . Multi-head projections at layer ℓ , head h , we project tokens to queries/keys/values:

$$Q = X^{(\ell-1)} W_Q^{(\ell,h)}, K = X^{(\ell-1)} W_K^{(\ell,h)}, V = X^{(\ell-1)} W_V^{(\ell,h)} \quad (11)$$

where W_Q, W_K, W_V are projection matrices for attention (Euclidean/tangent).

The attention logits incorporate structural biases from G_{seq} and relative positions:

$$\alpha_{ij}^{(\ell,h)} = \frac{\langle q_i, k_j \rangle}{\sqrt{d_h}} + b_p^{pos} + \phi^{(\ell,h)}(\tilde{A}_{i_t, i_j}) \quad (12)$$

where $\alpha_{ij}^{(\ell,h)}$ is attention logit (sequential branch) with position/graph biases for token $i \leftarrow j$. $\langle \cdot, \cdot \rangle$ denotes Euclidean inner product.

We also define relative-position bias b_p^{pos} for $p = j - i$ and edge biases $b_{ij}^{edge} = \phi^{(\ell,h)}(\tilde{A}_{i_t, i_j})$ where $\phi^{(\ell,h)}(\cdot)$ is a learnable scalar map of the normalized adjacency $\tilde{A}$; both will enter the attention logits.

Then, the attention weight becomes the per-head output at token i :

$$Head^{(\ell,h)}(i) = \sum_{j \leq t} \text{softmax}_j(\alpha_{ij}^{(\ell,h)}) v_j \quad (13)$$

where $Head^{(\ell,h)}(i)$ is per-head attended value at token i .

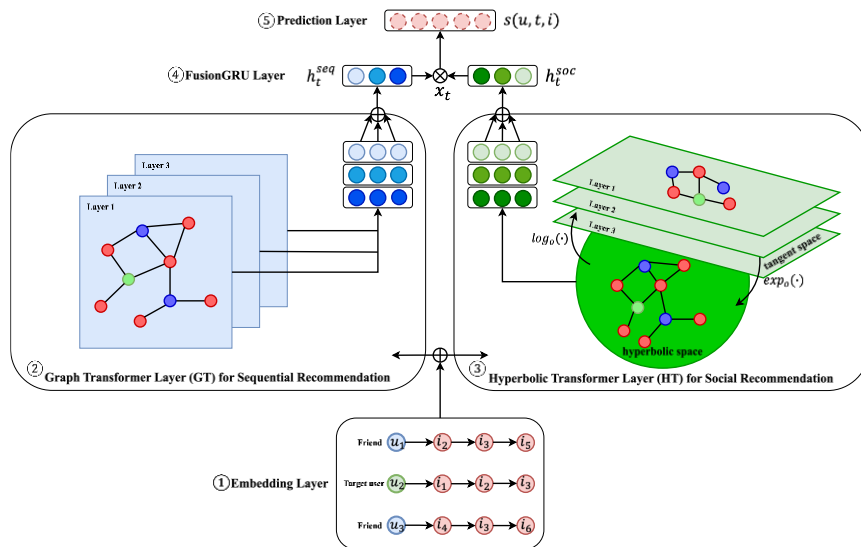


Figure 2. The architecture of Hyperbolic Graph Transformer for sequential and social Recommendation (HGT4Rec). HGT4Rec consists of five principal layers: 1. the embedding layer, 2. Graph Transformer layer, 3. Hyperbolic Transformer layer, 4. the FusionGRU layer, and 5. the prediction layer

Heads are concatenated and linearly projected by $W_o^{(l)}$, followed by residuals, LayerNorm and a position-wise feed-forward network (FFN): $X^{(l)} = \text{FFN}(\text{LayerNorm}(X^{(l-1)} + \text{Concat}_h(\text{Head}^{(l,h)}W_o^{(l)})))$. The final sequential state at time t is the last token representation:

$$h_t^{\text{seq}} = [GT(X_t)]_t \quad (14)$$

where h_t^{seq} denotes final sequential state at time t (from GT). $GT(X_t) \in \mathbb{R}^{t \times d}$ represents the matrix of contextualized token embeddings for all positions $1, \dots, t$.

This design follows self-attentive sequential recommendation with graph priors (Kang & McAuley, 2018), (Sun *et al.*, 2019), (C. Ying *et al.*, 2021), (C. Li *et al.*, 2023).

Hyperbolic Transformer for Social Recommendation

We compute attention in the tangent space and aggregate on the manifold at each layer $\ell = 1, \dots, L_h$:

- 1). Map to tangent (Euclidean) space at the origin:

$$y_v^{(\ell-1)} = \log_0^c(z_v^{(\ell-1)}) \quad (15)$$

where $v \in \{u\} \cup \mathcal{N}(u)$. $y_v^{(\ell-1)}$ is tangent-space vector.

- 2). Scaled dot-product attention in the tangent space (Euclidean computation, but biased by social structure and time):

$$Q = YW_Q^{(\ell)}, K = YW_K^{(\ell)}, V = YW_V^{(\ell)}, Y = [y_v^{(\ell-1)}]_v \quad (16)$$

For an edge (u, v) , we add time/degree-aware bias:

$$\beta_{uv}^{(\ell)} = \psi^{(\ell)}(w_{uv}) + \eta^{(\ell)}(\text{deg}(v)) \quad (17)$$

where $\beta_{uv}^{(\ell)}$ represents social attention bias (time/degree aware). $\psi(\cdot)$ and $\eta(\cdot)$ are learnable scalars for biasing social attention.

Then, leading to weights and messages:

$$a_{uv}^{(\ell)} = \text{softmax}_v\left(\frac{\langle q_u, k_v \rangle}{\sqrt{d}} + \beta_{uv}^{(\ell)}\right) \quad (18)$$

$$m_u^{(\ell)} = \sum_v a_{uv}^{(\ell)} y_v \quad (19)$$

where $a_{uv}^{(\ell)}$ is social attention weights in tangent space. $m_u^{(\ell)}$ represents message in tangent aggregated from neighbors.

- 3) Map back and aggregate on the manifold using gyrovectors operations:

$$\hat{m}_u^{(\ell)} = \exp_0^c(m_u^{(\ell)}) \quad (20)$$

$$z_u^{(\ell)} = z_u^{(\ell-1)} \oplus_c (\gamma^{(\ell)} \otimes_c \hat{m}_u^{(\ell)}) \quad (21)$$

where $\hat{m}_u^{(\ell)}$ is message mapped back to manifold. $z_u^{(\ell)}$ is manifold residual update (gyrovectors algebra) at layer ℓ . $\gamma^{(\ell)}$ signifies residual gain for manifold aggregation.

A manifold FFN is applied as $\exp_0^c(\text{FFN}(\log_0^c(\cdot)))$ (Ganea *et al.*, 2018), (Chami *et al.*, 2019), (Bachmann *et al.*, 2020).

After L_h layers, we align to the tangent space for fusion and obtain the final social representation (h_t^{soc}) for u :

$$h_t^{\text{soc}} = \log_0^c(z_u^{(L_h)}) \quad (22)$$

Remarks: Computing attention in the tangent space preserves transformer expressively while the aggregation and residuals occur on the manifold, which respects curvature and retains hierarchical inductive bias (Nickel & Kiela, 2017), (Ganea *et al.*, 2018), (Chami *et al.*, 2019), (Bachmann *et al.*, 2020), (Yang *et al.*, 2024), (A. Li *et al.*, 2022).

FusionGRU (Geometry - Aware Temporal Fusion)

For each encoder, we derive corresponding sequential and social representations. These representations capture various viewpoints of similar semantics, encompassing valuable representations from multiple viewpoints. To merge two sources of representations, simple and crude fusion approaches include pooling or concatenation. However, these fusions are suboptimal to our method as follows: 1) social and sequential representations originate from different sources, resulting in heterogeneous representations, 2) the social and sequential representations are produced by distinct encoders, creating a semantic gap in the hidden vector features, 3) social influence may not always enhance each sequential influence, potentially disrupting sequential influence and reducing overall performance.

Simple and crude fusion assume equivalent priority for all preferences, which may not hold true for social and sequential representations. Some preferences might be more significant, resulting in the loss of crucial representation. This issue prompts the suggestion of a FusionGRU, which is geometry-aware temporal fusion to effectively balance and utilize the advantages of two sources of representations. It fuses dynamic preferences in distinct spaces, modeling their balanced representations and improving optimization.

To adaptively integrate sequential and social cues at each timestep, we use a GRU that consumes both h_t^{seq} and h_t^{soc} (the latter already geometry-aligned via $\log_0(\cdot)$):

$$x_t = [h_t^{seq} || h_t^{soc}] \quad (23)$$

where x_t represents the input of FusionGRU at time step t .

GRU (X. Chen *et al.*, 2019) is a robust and optimized tool to capture social and sequential representations. The GRU's operations at each time step (t) are as follows:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \quad (24)$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \quad (25)$$

$$h'_t = \tanh(W_h x_t + U_h (r_t \odot h_{t-1}) + b_h) \quad (26)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot h'_t \quad (27)$$

where z_t denotes update gate and r_t represents reset gate. The recent memory content is symbolized as h'_t , h_t is the unified preference state, and h_{t-1} is the fused state from the previous step $t - 1$. The element-wise product is denoted as $\odot$. GRU utilizes sigmoid function $\sigma(\cdot)$ and hyperbolic tangent activation function $\tanh(\cdot)$, with the corresponding weight matrices are symbolized as W_z, W_r, W_h .

The reset gate r_t controls the former information to delete, while the update gate z_t controls the balance between retaining the former information h_{t-1} and amalgamating the current memory information h'_t . Intuitively, z_t learns when to rely more on sequential dynamics and social context under drift. GRU variants are widely adopted for fusion due to their efficiency and effectiveness over both predecessors; in contrast, our novelty is geometry-aware alignment before fusion.

Prediction Layer

Given candidate item i , the score is:

$$s(u, t, i) = h_t^T E_i[i] + b_i \quad (28)$$

where $s(u, t, i)$ denotes matching score for user u , state at t , and item i . b_i represents item bias term.

We rank all $i \in I$ by $s(u, t, i)$ for Top- K retrieval.

The algorithm of HGT4Rec can be detailed as shown in Algorithm 1.

Algorithm 1 HGT4Rec

Input: user embeddings E_u , item embeddings E_i , and timestamp embeddings E_t .

Output: Top- K recommendation.

#Graph Transformer for sequential recommendation
 1: encode item sequence X_t with graph-aware multi-head attention
 2: apply position/edge biases from global transition graph $\tilde{A}$
 3: obtain final sequential state h_t^{seq}
#Hyperbolic Transformer for social recommendation
 4: map user and neighbors to hyperbolic space via exp/log maps
 5: perform tangent-space attention with social/time biases
 6: aggregate on manifold, return social state h_t^{soc}
#Geometry-Aware Fusion (FusionGRU)
 7: form input $x_t \leftarrow [h_t^{seq} || h_t^{soc}]$
 8: update fused preference state h_t via update z_t and reset r_t gates
#Scoring and Ranking
 9: for each candidate item i , compute $s(u, t, i)$
 10: **return** Top- K items ranked by $s(u, t, i)$

Model Optimization

We train with a sampled softmax (cross-entropy) that is standard for next-item ranking. For each (u, t) , let $i^+ = i_{t+1}$ and $\mathcal{N}_{u,t}$ be a set of negative:

$$\mathcal{L}_{CE} = -\log \frac{\exp(s(u, t, i^+)/\tau)}{\exp(s(u, t, i^+)/\tau) + \sum_{i \in \mathcal{N}_{u,t}} \exp(s(u, t, i)/\tau)} \quad (29)$$

where $\mathcal{L}_{CE}$ represents cross-entropy objective loss. τ denotes temperature in sampled softmax with $\tau > 0$.

Ball-boundary penalty ($\mathcal{R}_{ball}$): We regularize Euclidean parameters with ℓ_2 and constrain manifold point via a ball penalty. To keep manifold points strictly inside $\mathbb{D}_c^d$, we add a ball penalty:

$$\mathcal{R}_{ball} = \sum_u \max \left(0, \|z_u^{(L_h)}\| - \frac{(1-\epsilon)}{\sqrt{c}} \right)^2 \quad (30)$$

where ϵ signifies margin to keep points inside $\mathbb{D}_c^d$.

Total loss ($\mathcal{L}$) is:

$$\mathcal{L} = \sum_{u,t} [\mathcal{L}_{CE} + \lambda_2 \|\Theta_{Euc}\|_2^2 + \lambda_b \mathcal{R}_{ball}] \quad (31)$$

where Θ_{Euc} is set of Euclidean parameters (all non-manifold). λ_2 and λ_b are ℓ_2 regularizer and ball-penalty coefficients.

We optimize Euclidean parameter Θ_{Euc} with Adam; manifold parameters (e.g., $z_u^{(\ell)}$, curvature if learnable) with Riemannian Adam and re-

projection to $\mathbb{D}_c^d$ as needed (Ganea *et al.*, 2018), (Chami *et al.*, 2019), (Bachmann *et al.*, 2020). Dropout and label smoothing are applied to stabilize training (Kang & McAuley, 2018), (Sun *et al.*, 2019), (C. Li *et al.*, 2023).

Implementation notes: Typical settings that worked well in our experiments: $d = 64$, $L_s = L_h = 3$, 4 heads, dropout = 0.1; window $w \in [2, 5]$. Bias functions $\phi(\cdot)$, $\psi(\cdot)$, $\eta(\cdot)$ are 1-layer MLPs to keep attention logits in a comparable numeric range. Curvature c can be fixed $c = 1$ or learned per layer (Bachmann *et al.*, 2020); we found fixing c simplifies optimization.

Results and Discussion

This section begins by outlining the research questions and introducing three real-world datasets, thirteen comparative baselines, and two evaluation metrics. Following this, we address four specific research questions through our experiments.

Research Questions

- **RQ1:** Does HGT4Rec achieve statistically significant improvements over state-of-the-art (SOTA) baselines?
- **RQ2:** Are HGT4Rec’s gains consistent across vary Top- K thresholds ($K \in \{10, 20, 40\}$)?
- **RQ3:** Which architectural components of HGT4Rec drive performance?
- **RQ4:** How sensitive is HGT4Rec to hyperparameters?

Datasets

We evaluate our method on three popular real-world benchmarks for recommender systems: Yelp (Asghar, 2016), iFashion (W. Chen *et al.*, 2019), and LastFM (Schedl, 2016). The Yelp corpus contains user–business interactions for venue recommendation and is sourced from the Yelp platform. iFashion is a fashion-outfit dataset curated by Alibaba. LastFM records user activities with music applications and online radio services. A summary of the key statistics for these datasets appears in Table 2. Here, #Users, #Items, and #Ratings denote the counts of unique users, unique items, and observed interactions, respectively, and Sparsity is computed as $1 - \frac{\text{\#Ratings}}{\text{\#Users} \times \text{\#Items}}$.

Table 2. Statistics of Datasets

Dataset	#Users	#Items	#Ratings	Sparsity
Yelp ¹	42,712	26,822	182,357	99.98%
iFashion ²	31,668	38,048	618,629	99.95%
LastFM ³	1,889	15,376	51,987	99.82%

Comparison methods

To rigorously evaluate HGT4Rec, we benchmark against many baseline methods spanning three families: 1) conventional non-sequential approaches; 2) GNN-based recommenders without SSL; and 3) recent SSL recommenders.

1). Conventional non-sequential baselines.

- **BiasMF** (Koren *et al.*, 2009): This matrix-factorization approach embeds users and items in a shared latent space and explicitly models user and item bias terms.
- **NCF** (He *et al.*, 2017): This deep neural CF model stacks fully connected layers to capture nonlinear user–item interaction patterns.
- **AutoRec** (Sedhain *et al.*, 2015): This autoencoder-style recommender learns representations by reconstructing observed interaction vectors.

2). GNN-based without SSL.

- **PinSage** (R. Ying *et al.*, 2018): This model employs graph convolution with random-walk–guided neighborhood sampling to enable efficient message passing on large bipartite graphs.
- **NGCF** (Wang *et al.*, 2019): This method stacks graph convolution layers to encode multi-hop collaborative signals into user and item embeddings.
- **LightGCN** (He *et al.*, 2020): This streamlined variant of NGCF removes feature transformations and nonlinearities, relying on lightweight graph convolution for effective representation learning.
- **GCCF** (L. Chen *et al.*, 2020): This approach refines GCN-based CF by omitting nonlinear transformations and introducing residual connections to stabilize training and enhance performance.

3) SSL-enhanced Recommenders.

- **EGLN** (Yang *et al.*, 2021): This model couples a node-embedding learner with a graph-structure learner and encourages mutual guidance between the two modules via self-supervision.
- **SLRec** (Yao *et al.*, 2021): This method applies contrastive learning on node features to regularize and improve recommendation representations.
- **NCL** (Lin *et al.*, 2022): This model first clusters users with an EM procedure and then performs neighborhood-aware contrastive learning within each cluster.

- **SGL** (Wu *et al.*, 2021): This approach constructs multiple graph views via random augmentations (e.g., edge dropping, node dropping, random walks) and trains with a contrastive objective.
- **HCCF** (Xia *et al.*, 2022): This hypergraph-based method forms global and local hypergraph views and optimizes a contrastive loss to capture global relations.
- **GFormer** (Li *et al.*, 2023): This graph-transformer baseline integrates a generative, rationale-aware self-supervised module that automatically synthesizes high-quality augmentations by distilling informative user-item interaction patterns, and it performs task-adaptive invariant rationalization to preserve global user-item relations.

Evaluation metrics

For each dataset, we partition the observed interactions into training, validation, and test splits in a 0.70 : 0.05 : 0.25 ratio. We repeat this process with five different random splits and report the average results. Following (Wu *et al.*, 2021), (Xia *et al.*, 2022), evaluation uses an all-items ranking protocol that scores each user against the full candidate set, thereby mitigating biases from negative sampling. Model quality is measured with two standard metrics: Recall@K and NDCG@K at $K \in \{10, 20, 40\}$.

Overall comparison (RQ1)

Table 3 summarizes a head-to-head comparison between HGT4Rec and all baselines. For readability, the overall best result in each column is boldfaced, while the strongest baseline is underlined. The Improvement row reports

HGT4Rec’s relative gain over the second-best method in that column.

We evaluate Recall@10 and NDCG@10 on Yelp, iFashion, and LastFM. Baselines span three families: 1) non-GNN CF models (BiasMF, NCF, AutoRec), 2) GNN recommenders without SSL (PinSage, NGCF, LightGCN, GCCF), and 3) SSL-enhanced methods (EGLN, SLRec, NCL, SGL, HCCF). We also include GFormer as a strong recent baseline. Statistical tests indicate significance at $p < 0.05$.

Main findings: 1) HGT4Rec attains the best performance on every dataset and metric. Its gains over the runner-up are large: Yelp (348.22% Recall@10; 314.86% NDCG@10), iFashion (383.21% Recall@10; 210.19% NDCG@10), and LastFM (90.85% Recall@10; 9.83% NDCG@10), and 2) The advantage holds against both non-SSL and SSL baselines. Even relative to strong contrastive approaches (SGL, HCCF, NCL), HGT4Rec remains decisively better, indicating benefits that go beyond augmentation-based self-supervision.

The strong point of HGT4Rec: 1) The GT (sequential module) captures long-range, high-order dependencies within user histories, improving coverage at the top ranks, 2) The HT (social module) better represents hierarchical and power-law characteristics of social ties than Euclidean counterparts, sharpening position-sensitive ranking quality, and 3) FusionGRU adaptively fuses sequential and social signals at each timestep, emphasizing the source most informative for a given user-context-particularly helpful under sparsity and long-tail interactions (e.g., Yelp, iFashion).

Table 3. The Comparison of Top-10 Recommendation

Methods	Yelp		iFashion		LastFM	
	<i>Recall@10</i>	<i>NDCG@10</i>	<i>Recall@10</i>	<i>NDCG@10</i>	<i>Recall@10</i>	<i>NDCG@10</i>
BiasMF	0.0122	0.0070	0.0302	0.0281	0.0609	0.0696
NCF	0.0166	0.0101	0.0268	0.0253	0.0574	0.0645
AutoRec	0.0230	0.0133	0.0309	0.0264	0.0543	0.0599
PinSage	0.0278	0.0171	0.0291	0.0276	0.0899	0.1046
NGCF	0.0438	0.0269	0.0375	0.0350	0.1257	0.1489
GCCF	0.0484	0.0296	0.0373	0.0352	0.1230	0.1452
LightGCN	0.0422	0.0254	0.0437	0.0416	0.1490	0.1739
EGLN	0.0458	0.0278	0.0473	0.0438	0.1133	0.1263
SLRec	0.0418	0.0258	0.0373	0.0353	0.1175	0.1384
NCL	0.0493	0.0301	0.0474	0.0446	0.1491	0.1745
HCCF	0.0518	0.0318	0.0489	0.0464	0.1502	0.1773
SGL	0.0522	0.0319	0.0512	0.0487	0.1496	0.1775
GFormer	0.0562	0.0350	0.0542	0.0520	0.1573	0.1831
HGT4Rec	0.2519	0.1452	0.2619	0.1613	0.3002	0.2011
<i>Improvement</i>	348.22%	314.86%	383.21%	210.19%	90.85%	9.83%

Bold numbers are the highest result in each column, while underlined numbers are the second-highest result in each column. Improvement row represents the percentage gain in improvement over the second-highest model, with statistically significant results at the 0.05 level. The results are shown as percentages.

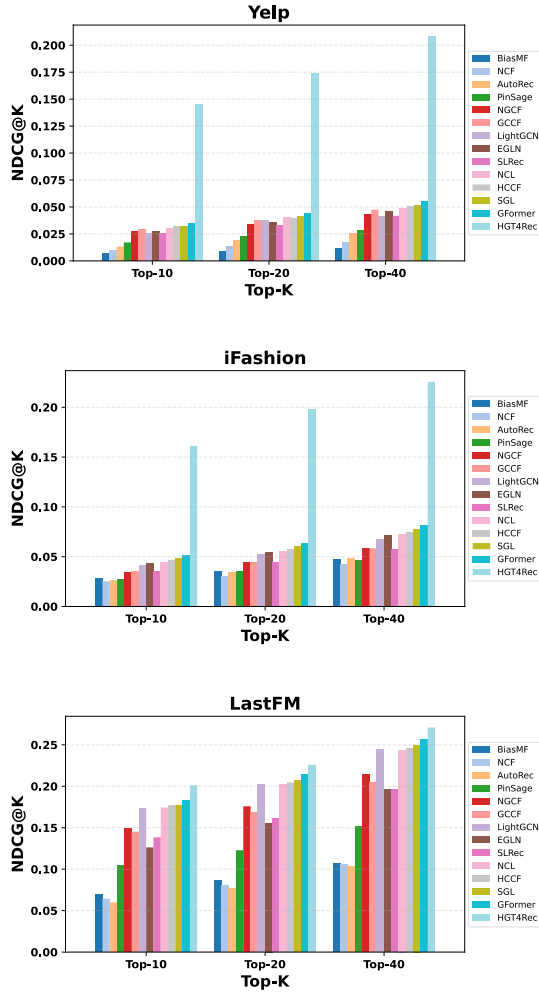


Figure 3. NDCG@K of varying Top-K values. To evaluate steady performance across all Top-K categories, NDCG@K is determined for Top-10, Top-20, and Top-40

In conclusion, across three benchmarks and two metrics, HGT4Rec consistently and significantly outperforms all compared methods. The combination of a GT backbone for sequences, a HT for social influence, and an adaptive FusionGRU underpins the observed, robust improvements.

Top-K study (RQ2)

We evaluate $NDCG@K$ at $K \in \{10, 20, 40\}$ on Yelp, iFashion, and LastFM. In every dataset and at every K , HGT4Rec achieves the highest NDCG, confirming consistent superiority as the recommendation list grows (Figure 3).

Key observations: 1) On the sparser Yelp and iFashion datasets, HGT4Rec’s lead over the strongest baseline expands with K , indicating sustained relevance in longer lists, 2) On the denser LastFM, the advantage is smaller but stable, showing robustness even when interactions are abundant.

Conclusion, these results decisively answer RQ2: HGT4Rec outperforms all baselines across all Top-K settings, with particularly large gains on sparser datasets. We attribute this to its heterogeneous GT, which captures high-order cross-relation signals, and attention mechanisms that prioritize high-confidence items, preserving ranking quality as K increases.

Ablation study (RQ3)

To examine the design rationale of HGT4Rec, we perform an ablation over five configurations (Table 4): 1) vanilla Transformer baseline (T) for single component aspect and no graph/social modeling, 2) HT for social recommendation, 3) GT for sequential recommendation, 4) HGT4Rec without FusionGRU (w/o FusionGRU) for fusing GT and HT using simple concatenation and MLP, and 5) complete HGT4Rec. Across all datasets, HGT4Rec outperforms all variants, confirming the benefit of each component and their integration.

Findings; 1) GT consistently improves over T and often by a larger margin than HT: the graph-aware encoder captures higher-order item transitions and strengthens sequential signals beyond both self-attention alone of Transformer-based and social influence, 2) HT also outperforms T: modeling social relations in hyperbolic space better self-attention alone of Transformer-based, 3) Combining GT and HT-even with a simple MLP-outperforms either component alone, showing the complementarity of sequential and social signals, 4)

Table 4. Ablation Study

Dataset	Metrics	T	HT	GT	w/o FusionGRU	HGT4Rec
Yelp	<i>Recall@10</i>	0.2267	0.2285	0.2296	0.2308	0.2519
	<i>NDCG@10</i>	0.1136	0.1152	0.1162	0.1179	0.1452
iFashion	<i>Recall@10</i>	0.2188	0.2197	0.2205	0.2213	0.2619
	<i>NDCG@10</i>	0.1185	0.1192	0.1201	0.1209	0.1613
LastFM	<i>Recall@10</i>	0.2949	0.2966	0.2988	0.2995	0.3002
	<i>NDCG@10</i>	0.1988	0.1993	0.2002	0.2008	0.2011

Replacing static MLP fusion with FusionGRU yields consistent, modest gains: the gated recurrent fusion dynamically weights sequential and social cues over time, improving both retrieval and ranking.

Conclusion, the ablation validates our design. Graph-based sequential modeling (GT) and hyperbolic social modeling (HT) each confer clear benefits; combining them is better than either alone; and FusionGRU provides the best overall performance by learning a temporal, context-aware integration of the two influences.

Hyperparameter study (RQ4)

We implemented HGT4Rec in TensorFlow with Python and ran all experiments on Google Colab Pro. All hyperparameter tuning is performed first on a validation split, and all subsequent experiments (overall comparison, ablation, and Top- K study) use the best-found settings. We performed a grid search over key hyperparameters. For the embedding dimension D and the batch size, we evaluated $\{8, 16, 32, 64, 128\}$ and selected 64 for both. To study model depth, we varied the number of layers in the GT and HT components from 1 to 5 (Figure 4), while keeping FusionGRU at a layer for simplicity. For attention, we tested $\{2, 4, 6, 8\}$ heads (Figure 5) and the best was 4 heads. The best training settings were a learning rate of 0.001 and a dropout rate of 0.1 (Figure 6). All models were trained with the Adam optimizer for 100 epochs.

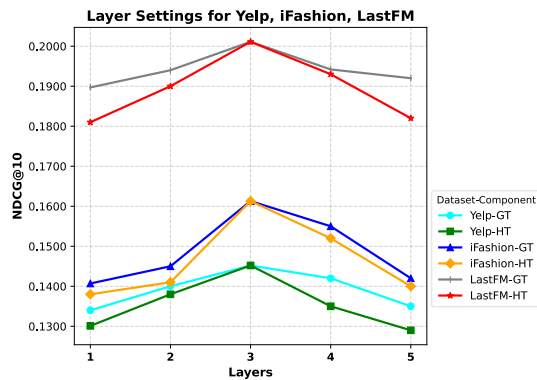


Figure 4. NDCG@10 versus the number of layers for GT and HT on Yelp, iFashion, and LastFM (x-axis: layers, y-axis: NDCG@10)

The number of layers of HT and GT is analyzed. The results indicate that the optimal performance is consistently achieved with three layers for all components across all datasets.

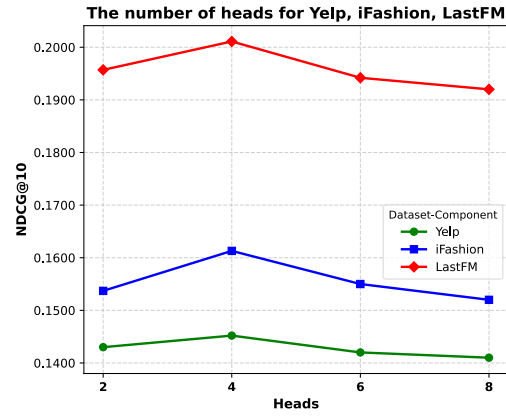


Figure 5. The number of heads

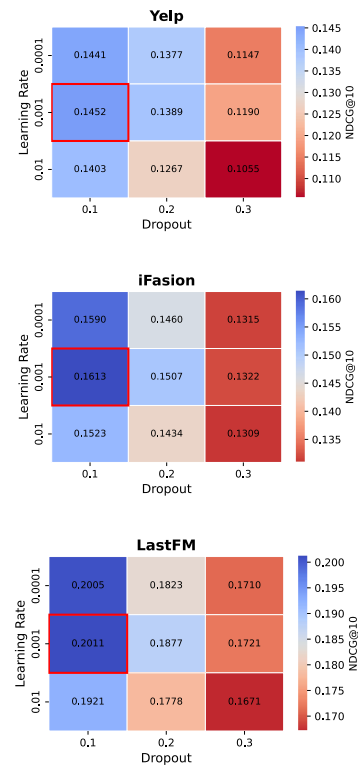


Figure 6. The learning rate versus dropout

Conclusions

This work introduced HGT4Rec, a hyperbolic graph transformer that jointly models sequential dynamics and social influence within a single, end-to-end architecture. The model couples a GT for item-transition sequences with a HT that captures long-range, hierarchical effects on the social graph, and integrates them through a geometry-aware

FusionGRU. By aligning Euclidean (sequential) and hyperbolic (social) representations via exponential/logarithm maps before gated fusion, HGT4Rec yields a unified preference state that adapts over time and under drift.

Across three public benchmarks, HGT4Rec achieves state-of-the-art results on Recall@ K and NDCG@ K , with large margins over strong baselines, including recent graph-transformer and contrastive methods. Top- K analyses further show that the gains persist and often widen as K increases, especially on sparser datasets, indicating that the model maintains ranking quality in longer recommendation lists. Ablations confirm that each component is necessary: the sequential GT and the social HT contribute complementary signals, their naïve concatenation underperforms the proposed FusionGRU, and the full system is consistently best. A sensitivity study offers practical guidance (e.g., three layers for both encoders, four heads, 64-dimensional embeddings, 0.1 dropout).

Limitations and future work, HGT4Rec relies on the availability and quality of social ties and (when used) their timestamps; missing or noisy edges may attenuate benefits. Although hyperbolic operations are efficient in modern libraries, full-batch social attention can be costly on very large graphs; scalable sampling, pruning, or distillation strategies are promising directions. In its current form, HGT4Rec does not explicitly address the cold-start problem for new users or items. In future work, HGT4Rec could incorporate cold-start remedies such as metadata-based warm-start embeddings, social-neighbor priors for users with few interactions, or pretraining/meta-learning strategies that generalize to unseen entities. Future extensions include learnable or node-adaptive curvature, dynamic social graphs with temporal edge formation, multi-behavior and multimodal context (e.g., text or images), explainability for the fused decisions, privacy-preserving training, and online/streaming evaluation with bandit feedback.

In summary, HGT4Rec demonstrates that explicitly combining Euclidean sequence modeling with hyperbolic social reasoning, and fusing them through geometry-aware gating, is a principled and empirically powerful recipe for next-item recommendation. The results suggest a general path forward for hybrid-geometry recommenders that respect the distinct inductive biases of sequential and social signals while enabling stable, interpretable integration.

Funding

The authors received no specific financial support for the research, authorship, and/or publication of this article.

Author Contributions

Conceptualization: R.C., J.P., Ph.K., Ni.K., Po.K., S.D., S.P., Na.K., W.D., J.Y., P.C., S.K., S.T., C.S.; Methodology: R.C., J.P., Ph.K., Ni.K., Po.K., S.D., S.P., Na.K., W.D., J.Y., P.C., S.K., S.T., C.S.; Validation: R.C., J.P., Ph.K., Ni.K., Po.K., S.D., S.P., Na.K., W.D., J.Y., P.C., S.K., S.T., C.S.; Formal Analysis: R.C., J.P., Ph.K., Ni.K., Po.K., S.D., S.P., Na.K., W.D., J.Y., P.C., S.K., S.T., C.S.; Investigation: R.C., J.P., Ph.K., Ni.K., Po.K., S.D., S.P., Na.K., W.D., J.Y., P.C., S.K., S.T., C.S.; Data Curation: R.C., J.P., Ph.K., Ni.K., Po.K., S.D., S.P., Na.K., W.D., J.Y., P.C., S.K., S.T., C.S.; Writing – Original Draft: R.C., J.P., Ph.K., Ni.K., Po.K., S.D., S.P., Na.K., W.D., J.Y., P.C., S.K., S.T., C.S.; Writing – Review & Editing: R.C., J.P., Ph.K., Ni.K., Po.K., S.D., S.P., Na.K., W.D., J.Y., P.C., S.K., S.T., C.S. All authors have read and agreed to the published version of the manuscript.

Table 5. The Selected Hyperparameters for Baseline Methods

Method	Embedding size	Layers	Batch size	Learning rate	Dropout
BiasMF	64	-	256	1e-3	-
NCF	64	3 MLP	256	1e-3	0.1
AutoRec	128	1 AE	256	1e-3	0.1
PinSage	64	2 GCN	256	1e-3	0.1
NGCF	64	3 GCN	256	1e-3	0.1
GCCF	64	3 GCN	256	1e-3	0.1
LightGCN	64	3 GCN	256	1e-3	0
EGLN	64	2 GNN	256	1e-3	0.1
SLRec	64	2 GNN	256	1e-3	0.1
NCL	64	3 GNN	256	1e-3	0.1
HCCF	64	2HGNN	256	1e-3	0.1
SGL	64	3 GCN	256	1e-3	0.1
GFormer	64	2 GT	256	1e-3	0.1

Ethics Statement

Ethical approval was not required for this study because it did not involve human participants, animals, identifiable personal data, or materials requiring institutional ethical review.

Declaration of Generative AI Use

During the preparation of this manuscript, the authors used ChatGPT (OpenAI) solely to assist with language editing, including grammar checking, improving clarity, and enhancing the overall flow and readability of the manuscript. ChatGPT was not used to generate scientific content or to contribute to the study design, methodology, data analysis, interpretation of results, or conclusions. All AI-assisted revisions were critically reviewed, verified, and further revised by the authors as necessary. The authors take full responsibility for the originality, accuracy, integrity, and scientific content of the manuscript.

Conflict of Interest

The authors declare that they have no conflicts of interest related to this work.

Data Availability Statement

The datasets analyzed in this study are publicly available. The Yelp dataset can be accessed through the Yelp Open Dataset (<https://business.yelp.com/data/resources/open-dataset/>), the iFashion dataset through the POG GitHub repository (<https://github.com/wenyuer/POG>), and the LastFM dataset through Kaggle (<https://www.kaggle.com/datasets/harshal19t/lastfm-dataset>). No new datasets were generated in this study. Use of the datasets is subject to the terms and conditions of the respective data providers.

Acknowledgement

The authors thank Major of Information Technology and Digital Business, Faculty of Business Administration, Rajamangala University of Technology Krungthep, Bangkok, Thailand for their supports.

Appendix

Table 5 summarizes the best hyperparameter settings for each baseline on the validation split; the same configuration is then used to report test performance in Table 3.

References

- Asghar, N. (2016). Yelp dataset challenge: Review rating prediction. *arXiv*. <https://doi.org/10.48550/arXiv.1605.05362>
- Bachmann, G., Bécigneul, G., & Ganea, O. (2020). Constant curvature graph convolutional networks. In H. Daumé III & A. Singh (Eds.), *Proceedings of the 37th International Conference on Machine Learning* (Vol. 119, pp. 486-496). PMLR.
- Chami, I., Ying, R., Ré, C., & Leskovec, J. (2019). Hyperbolic graph convolutional neural networks. In *Advances in Neural Information Processing Systems 32*. Curran Associates.
- Chen, L., Wu, L., Hong, R., Zhang, K., & Wang, M. (2020). Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(1), 27-34. <https://doi.org/10.1609/aaai.v34i01.5330>
- Chen, W., Huang, P., Xu, J., Guo, X., Guo, C., Sun, F., Li, C., Pfadler, A., Zhao, H., & Zhao, B. (2019). POG: Personalized outfit generation for fashion recommendation at Alibaba iFashion. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 2662-2670). Association for Computing Machinery. <https://doi.org/10.1145/3292500.3330652>
- Chen, X., Zhang, Y., & Qin, Z. (2019). Dynamic explainable recommendation based on neural attentive models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(1), 53-60. <https://doi.org/10.1609/aaai.v33i01.330153>
- Ganea, O.-E., Bécigneul, G., & Hofmann, T. (2018). Hyperbolic neural networks. In *Advances in Neural Information Processing Systems 31*. Curran Associates.
- He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., & Wang, M. (2020). 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). Association for Computing Machinery. <https://doi.org/10.1145/3397271.3401063>
- He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T.-S. (2017). Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web* (pp. 173-182). International World Wide Web Conferences Steering Committee. <https://doi.org/10.1145/3038912.3052569>
- Huang, Z., Sun, Z., Liu, J., & Ye, Y. (2024). Group-aware graph neural networks for sequential recommendation. *Information Sciences*, 670(120623). <https://doi.org/10.1016/j.ins.2024.120623>
- Kang, W.-C., & McAuley, J. (2018). Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)* (pp. 197-206). IEEE. <https://doi.org/10.1109/ICDM.2018.00035>

- Kannikaklang, N., & Wongthanavasu, S. (2025). BiTG4Rec: Bidirectional transformer graphs for sequential-social recommendation. *IEEE Access*, 13, 175190-175211. <https://doi.org/10.1109/ACCESS.2025.3616206>
- Kannikaklang, N., Thamviset, W., & Wongthanavasu, S. (2024). BiHGCA: A novel SRS-based bidirectional hyperbolic graph capsule co-attention network for user preference drift. *IEEE Access*, 12, 105831-105849. <https://doi.org/10.1109/ACCESS.2024.3436016>
- Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8), 30-37. <https://doi.org/10.1109/MC.2009.263>
- Li, A., Yang, B., Hussain, F. K., & Huo, H. (2022). HSR: Hyperbolic social recommender. *Information Sciences*, 585, 275-288. <https://doi.org/10.1016/j.ins.2021.11.040>
- Li, C., Xia, L., Ren, X., Ye, Y., Xu, Y., & Huang, C. (2023). Graph transformer for recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 1680-1689). Association for Computing Machinery. <https://doi.org/10.1145/3539618.3591723>
- Lin, Z., Tian, C., Hou, Y., & Zhao, W. X. (2022). Improving graph collaborative filtering with neighborhood-enriched contrastive learning. In *Proceedings of the ACM Web Conference 2022* (pp. 2320-2329). Association for Computing Machinery. <https://doi.org/10.1145/3485447.3512104>
- Nickel, M., & Kiela, D. (2017). Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems 30*. Curran Associates.
- Schedl, M. (2016). The LFM-1b dataset for music retrieval and recommendation. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval* (pp. 103-110). Association for Computing Machinery. <https://doi.org/10.1145/2911996.2912004>
- Sedhain, S., Menon, A. K., Sanner, S., & Xie, L. (2015). AutoRec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th International Conference on World Wide Web Companion* (pp. 111-112). Association for Computing Machinery. <https://doi.org/10.1145/2740908.2742726>
- Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., & Jiang, P. (2019). BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (pp. 1441-1450). Association for Computing Machinery. <https://doi.org/10.1145/3357384.3357895>
- Wang, X., He, X., Wang, M., Feng, F., & Chua, T.-S. (2019). Neural graph collaborative filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 165-174). Association for Computing Machinery. <https://doi.org/10.1145/3331184.3331267>
- Wu, B., Zhong, L., & Ye, Y. (2023). Graph-Augmented Social Translation Model for Next-Item Recommendation. *IEEE Transactions on Industrial Informatics*, 19(11), 10913-10922. <https://doi.org/10.1109/TII.2023.3242809>
- Wu, J., Wang, X., Feng, F., He, X., Chen, L., Lian, J., & Xie, X. (2021). 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). Association for Computing Machinery. <https://doi.org/10.1145/3404835.3462862>
- Xia, L., Huang, C., Xu, Y., Zhao, J., Yin, D., & Huang, J. (2022). Hypergraph contrastive collaborative filtering. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 70-79). Association for Computing Machinery. <https://doi.org/10.1145/3477495.3532058>
- Xie, H., Zhou, Y., He, J., & Li, T. (2023). A Sequential Recommendation Model Based on Social Behavior. *2023 2nd International Joint Conference on Information and Communication Engineering (JCICE)*, 141-145. <https://doi.org/10.1109/JCICE59059.2023.00037>
- Xu, Y., Li, X., Li, J., Wang, C., Gao, R., & Yu, Y. (2019). SSSER: Spatiotemporal Sequential and Social Embedding Rank for Successive Point-of-Interest Recommendation. *IEEE Access*, 7, 156804-156823. <https://doi.org/10.1109/ACCESS.2019.2950061>
- Yang, Y., Wu, L., Hong, R., Zhang, K., & Wang, M. (2021). Enhanced graph learning for collaborative filtering via mutual information maximization. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 71-80). Association for Computing Machinery. <https://doi.org/10.1145/3404835.3462928>
- Yang, Y., Wu, L., Zhang, K., Hong, R., Zhou, H., Zhang, Z., Zhou, J., & Wang, M. (2024). Hyperbolic graph learning for social recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 36(12), 8488-8501. <https://doi.org/10.1109/TKDE.2023.3343402>
- Yao, T., Yi, X., Cheng, D. Z., Yu, F., Chen, T., Menon, A., Hong, L., Chi, E. H., Tjoa, S., Kang, J., & Ettinger, E. (2021). Self-supervised learning for large-scale item recommendations. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (pp. 4321-4330). Association for Computing Machinery. <https://doi.org/10.1145/3459637.3481952>
- Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y., & Liu, T.-Y. (2021). Do transformers really perform bad for graph representation? In *Advances in Neural Information Processing Systems 34*. Curran Associates.
- Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., & Leskovec, J. (2018). 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). Association for Computing Machinery. <https://doi.org/10.1145/3219819.3219890>