GRAPHIA: Harnessing Social Graph Data to Enhance LLM-Based Social Simulation

Jiarui Ji[1], **Zehua Zhang**[2], **Zhewei Wei**[1*],
Bin Tong[2], **Guan Wang**[2], **Bo Zheng**[2*]
[1]Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China
[2]Alimama Tech, Taobao & Tmall Group of Alibaba
{jijiarui, zhewei}@ruc.edu.cn
{yuzheng.zzh, tongbin.tb, shangfeng.wg, bozheng}@alibaba-inc.com

Abstract

Large language models (LLMs) have shown promise in simulating human-like social behaviors. Social graphs provide high-quality supervision signals that encode both local interactions and global network structure, yet they remain underutilized for LLM training. To address this gap, we propose Graphia, the first general LLM-based social graph simulation framework that leverages graph data as supervision for LLM post-training via reinforcement learning. With GNN-based structural rewards, Graphia trains specialized agents to predict whom to interact with (destination selection) and how to interact (edge generation), followed by designed graph generation pipelines. We evaluate Graphia under two settings: Transductive Dynamic Graph Generation (TDGG), a micro-level task with our proposed node-wise interaction alignment metrics; and Inductive Dynamic Graph Generation (IDGG), a macro-level task with our proposed metrics for aligning emergent network properties. On three real-world networks, Graphia improves micro-level alignment by 6.1% in the composite destination selection score, 12% in edge classification accuracy, and 27.9% in edge content BERTScore over the strongest baseline. For macro-level alignment, it achieves 35.98% higher structural similarity and 28.71% better replication of social phenomena such as power laws and echo chambers. Our results show that social graphs can serve as high-quality supervision signals for LLM post-training, closing the gap between agent behaviors and network dynamics for LLM-based simulation. Code is available at https://github.com/Ji-Cather/Graphia.git.

1 Introduction

Social simulation with LLM-based agents has emerged as a powerful paradigm in computational social science (Gao et al., 2023, 2024), enabling large-scale exploration of emergent social phenomena such as echo chambers and influence propagation (Piao et al., 2025; Wang et al., 2025a). These macroscopic phenomena arise from microscopic text-based interactions between LLM-based agents. Consequently, realistic social simulation requires modeling the microscopic actions that drive graph evolution (McPherson et al., 2001).

Despite this intrinsic micro-macro link, current social graph generators often decouple these two levels. At the macro level, deep-learning based generators focus on structural evolution using node IDs but ignore the semantic text that catalyzes edge formation (Gupta et al., 2022; Hosseini et al., 2025); while LLM-based generators rely either on qualitative case studies for evaluation (Piao et al., 2025; Wang et al., 2025a) or context-specific pipelines such as Twitter simulation (Ji et al., 2025) that lack generalization. At the micro level, prior work focuses on fine-grained behavioral realism. Zhou et al. (2024) propose SOTOPIA to evaluate how agents interact, while Zhou et al. (2025) focus on predicting who interacts next. Yet these LLM-based generators often overlook the macroscopic evolution of the social graph structure.

Collectively, these approaches face three critical limitations: **(1) Representational Deficiency**: traditional deep-learning based models are confined to modeling graph topology and are incapable of capturing the underlying text-driven activity; **(2) Methodological Gap**: the lack of a generalizable training framework that optimizes micro-level interactions and macro-level structure using social graph data as supervision; and **(3) Evaluation Gap**: the lack of unified metrics to quantitatively measure how well simulated social graphs match real

* Zhewei Wei and Bo Zheng are the corresponding authors.

† The work was partially done at Gaoling School of Artificial Intelligence, Beijing Key Laboratory of Research on Large Models and Intelligent Governance, Engineering Research Center of Next-Generation Intelligent Search and Recommendation, MOE, and Pazhou Laboratory (Huangpu), Guangzhou, Guangdong 510555, China.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 7103–7128

ones in both interactions and structure.

To address these limitations, we build on the Transductive (TDGG) and Inductive Dynamic Graph Generation (IDGG) settings from GDGB (Peng et al., 2025) as a foundation for systematic evaluation. In TDGG, we focus on microscopic alignment metrics that evaluate agent-level interactions. In IDGG, we focus on macroscopic alignment metrics that assess whether simulated graphs reproduce real-world graph properties. Under this paradigm, we formalize three core capabilities for realistic LLM-based social graph simulation: **(1) Destination Selection**: Given a source node, can the LLM predict its next interaction partner? **(2) Edge Generation**: Can the LLM generate socially coherent and contextually grounded micro-interactions between nodes? **(3) Global Structure Fidelity**: Does the generated graph reproduce key macro properties of real graphs?

Guided by these three principles, we propose **Graphia**, a reinforcement learning framework for LLM-based social graph simulation. Our contributions are: (1) The first unified training framework that leverages social graph data as supervision to enhance LLM-based simulation; (2) A micro–macro evaluation paradigm that extends TDGG and IDGG with novel quantitative metrics for joint assessment of interaction fidelity and network realism; (3) improved micro-level performance in TDGG tasks, with 6.1% gain in the composite destination selection score and enhanced edge content quality (+12% edge-classification accuracy, +27.9% BERTScore) over the strongest baseline; (4) enhanced macro-level fidelity in IDGG tasks, achieving 35.98% higher structural similarity and 28.71% better replication of emergent social phenomena (e.g., power laws, echo chambers) when evaluated against ground-truth social graphs.

2 Related Works

2.1 Social Graph Simulation

Existing social graph simulation methods fall into two categories. Structure-driven models (Gupta et al., 2022; Hosseini et al., 2025) which capture temporal and topological network dynamics but cannot generate text-rich interactions. LLM-based simulators (Mou et al., 2024) generate textual interactions but rely on task-specific pipelines and lack training signals from social graphs. For example, SOTOPIA-RL (Yu et al., 2025) trains LLMs using LLM-as-judge rewards on interaction text alone, failing to learn structural properties like degree distributions or homophily. While some works incorporate homophily (Rossetti et al., 2024) or influence propagation (Liu et al., 2024) into evaluation, their simulation pipelines are training-free and lack graph-guided learning. This misalignment between training objectives and empirical network structure limits the realism of simulated social dynamics.

2.2 Social Simulation Evaluation

LLM-based social simulations are typically evaluated at two levels. At the micro level, interaction quality is often assessed using the *LLM-as-a-judge* paradigm, which leverages large language models to score dialogue coherence, goal fulfillment, or social appropriateness (Zhou et al., 2024; Wang et al., 2025b). While scalable, this approach is sensitive to prompts and exhibits inconsistencies (Li et al., 2024; Zhou et al., 2025). At the macro level, current studies rely on qualitative validation of emergent phenomena in specific simulation scenarios, such as attitude shifts (Yao et al., 2025) or information diffusion (Gao et al., 2023) on Twitter. These assessments are tied to specific simulation scenarios or datasets, making them unsuitable for systematic or cross-dataset comparison. Consequently, a unified quantitative framework that jointly evaluates micro-level interactions and macro-level network realism remains lacking.

3 Proposed Framework

We focus on modeling human-like behaviors in dynamic text-attributed social graphs. In this section, we define the social graph data structure, describe the post-training of LLMs for aligning with social behaviors, and introduce our LLM-based framework for social graph simulation.

3.1 Problem Formulation

We consider a directed, dynamic social graph represented as a sequence of time-stamped subgraphs $\{G_t\}_{t=1}^{T}$, where $G_t = (\mathcal{V}_t, \mathcal{E}_t, \mathbf{P}_t, \mathbf{X}_t)$. Here, $\mathcal{V}_t$ is the node set at time t, each representing a person; $\mathcal{E}_t \subseteq \mathcal{V}_t \times \mathcal{V}_t$ is directed edge set, denoting interactions from one person to another. For node attribute, $\mathbf{P}_t = \{p_v \mid v \in \mathcal{V}_t\}$ contains the node profiles, where p_v is a textual description of node v (e.g., interests, role). For edge attribute, $\mathbf{X}_t = \{(m_{u\to v}, y_{u\to v}) \mid (u, v) \in \mathcal{E}_t\}$ is the edge set, where $m_{u\to v}$ is the textual message content and $y_{u\to v} \in \{1, \ldots, Y\}$ is its interaction category (e.g., a post or comment). Given a histori-

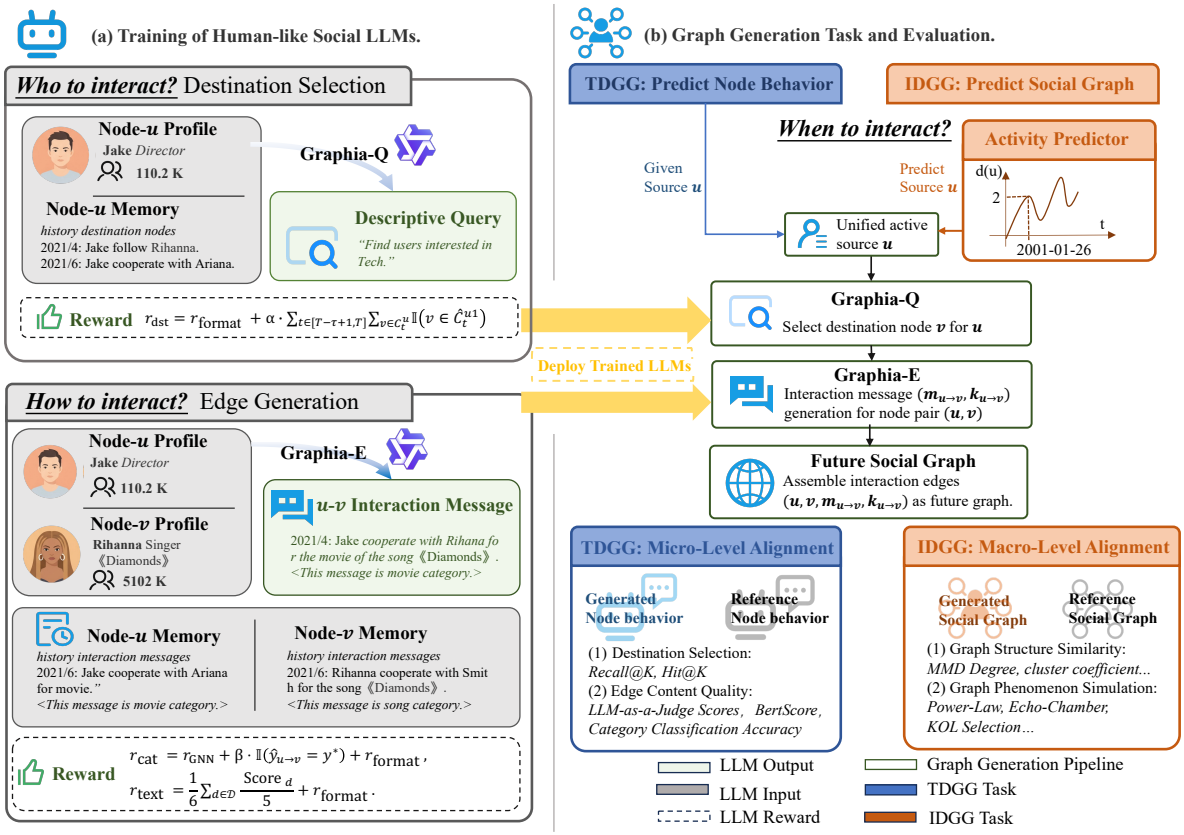


Figure 1: Graphia training, generation, and evaluation pipeline illustrated on a collaboration network. (a) The left panel details the training mechanisms for specialized LLM-based agents: **Graphia-Q for destination selection** (top-left) and **Graphia-E for edge generation** (bottom-left). These agents leverage text-rich node profiles and interaction memories, with rewards designed to optimize respective tasks. (b) The right panel outlines the graph generation pipeline based on trained LLM-based agents for TDGG and IDGG tasks. TDGG focuses on micro node behavior; while IDGG, supported by an activity predictor, models the macro social graph.

cal window of length τ, we define the observed sequence as $\mathcal{G}_{\text{hist}} = \mathcal{G}_{1:T-\tau} = \{G_1, \ldots, G_{T-\tau}\}$, and the goal is to generate the future sequence $\hat{\mathcal{G}}_{\text{fut}} = \hat{\mathcal{G}}_{T-\tau+1:T} = \{\hat{G}_{T-\tau+1}, \ldots, \hat{G}_T\}$. To capture dynamic behavioral context, we define the node memory $\mathcal{M}_t(u)$ for node u, which records its past interactions within the historical window:

$$\mathcal{M}_t(u) = \left\{ \left(p_v, m_{u\to v}, y_{u\to v}\right) \,\middle|\, (u,v) \in \mathcal{E}_{<t} \right\}.$$

This memory includes both the destination nodes' profiles p_v and the previous messages $m_{u\to v}$ with their semantic categories $y_{u\to v}$.

Following GDGB (Peng et al., 2025), we decompose the social graph simulation task into two settings: TDGG and IDGG. The generation of each interaction in the social graph is modeled as a Markov process:

$$p(u, v, m, y \mid \mathcal{G}_{\text{hist}}) = p(u \mid \mathcal{G}_{\text{hist}}) \cdot p(v \mid u, \mathcal{G}_{\text{hist}}) \cdot p(m, y \mid u, v, \mathcal{G}_{\text{hist}}). \quad (1)$$

In the TDGG task, the active source node set is given. The model estimates $p(v \mid u, \mathcal{G}_{\text{hist}})$ for destination selection and $p(m, y \mid u, v, \mathcal{G}_{\text{hist}})$ for edge generation. This task focuses on micro-level evaluation of interaction patterns between u and v.

In the IDGG task, source nodes are not provided; the model must learn $p(u \mid \mathcal{G}_{\text{hist}})$ endogenously. This requires modeling the full generative process of future graph evolution. This task focuses on macro-level evaluation by assessing how well the generated future graph $\hat{\mathcal{G}}_{\text{fut}}$ reproduces realistic social network structures and dynamic patterns.

3.2 Graphia Learning Framework

Building upon Equation (1), we develop a learning framework to train LLMs for simulating human-like node behaviors in dynamic graphs. Based on the trained LLMs, we design a unified graph generation pipeline for TDGG and IDGG tasks. The overall framework is illustrated in Figure 1.

Activity Prediction. To capture which nodes will

become active, i.e., $p(u \mid \mathcal{G}_{\text{hist}})$, we introduce the Activity-Predictor, which is implemented with the Informer architecture (Zhou et al., 2021). For each source node $u \in \mathcal{V}_T$, it takes the historical out-degree sequence $\{d_t(u)\}_{t=1}^{T-\tau}$ as input and predicts the out-degrees over the future horizon: $\{\hat{d}_{T-\tau+1}(u), \ldots, \hat{d}_T(u)\}$. This module is trained to minimize the mean squared error between predicted and actual out-degree:

$$\mathcal{L}_{\text{deg}} = \frac{1}{\tau N} \sum_{u \in \mathcal{V}_T} \sum_{t=T-\tau+1}^{T} \left(d_t(u) - \hat{d}_t(u) \right)^2,$$

where $d_t(u)$ denotes the true out-degree of node u at time t, and $N = |\mathcal{V}_T|$. The predicted out-degrees serve as structural priors for identifying future active source nodes in the IDGG task.

Interaction Policy Learning. For modeling $p(v \mid u, \mathcal{G}_{\text{hist}})$ and $p(m, y \mid u, v, \mathcal{G}_{\text{hist}})$, we treat the LLM as a policy model trained via reinforcement learning. We train two specialized LLMs, Graphia-Q for destination selection and Graphia-E for edge generation, each optimized for its respective task.

(1) Destination Selection. For each source node u, we train a generative LLM, Graphia-Q, to predict the ground-truth destination node set C_t^u at time t. We denote the predicted destination node set as $\hat{C}_t^u$. To retrieve $\hat{C}_t^u$, Graphia-Q generates a descriptive query to constrain the search space. We first retrieve a preliminary candidate set of K_1 by ranking nodes in $\mathcal{V}_T$ based on semantic similarity (BERT embedding cosine similarity) to the query, restricted to historical neighbors satisfying the filter rule. This leads to the first candidate node set: $\hat{C}_t^{u1}$. For graph generation, $\hat{C}_t^{u1}$ is truncated to size $\hat{C}_t^u = (\hat{C}_t^{u1})_{:\text{round}(\hat{d}_t(u))}$, where $\hat{d}_t(u)$ is either given (TDGG) or predicted (IDGG). Specifically, following GAD (Lei et al., 2025), we observe that common neighbors serve as an effective filter function; thus, we re-rank items retrieved via common neighbors. Detailed process is provided in Appendix C.1. To train Graphia-Q, we design a hybrid reward function:

$$r_{\text{dst}} = r_{\text{format}} + \alpha \sum_{t=T-\tau+1}^{T} \sum_{v \in C_t^u} \mathbb{I}(v \in \hat{C}_t^{u1}),$$

where r_{format} is 1 if the generated query conforms to the required format and 0 otherwise, and the second term counts how many true destinations appear in the top-K_1 candidates. The hyperparameter α controls the relative strength of the reward for correctly retrieving true destinations.

(2) Edge Generation. We train Graphia-E, a generative LLM that generates both the message $\hat{m}_{u \to v}$ and interaction category $\hat{y}_{u \to v}$ for each node pair (u, v). To ensure valid output formatting, we include a format reward r_{format}, computed via rule-based parsing, with $r_{\text{format}} = 1$ if valid, else 0. To train Graphia-E, we design separate reward functions for two subtasks: category prediction and message generation. For category prediction, we adopt a curriculum-style reward function that measures prediction accuracy, with the emphasis shifting progressively over training epochs:

$$r_{\text{cat}} = r_{\text{GNN}} + \beta \cdot \mathbb{I}(\hat{y}_{u \to v} = y^*) + r_{\text{format}},$$

where $\mathbb{I}(\cdot)$ is the indicator function, $r_{\text{GNN}} = [\mathbf{z}_{u,v}]_{y^*}$ is the logit score for the ground-truth category $y^* = y_{u \to v}$ from a pre-trained DGNN edge classifier (We adopt GraphMixer (Cong et al., 2023)), serving as a structural prior to guide the model. We set $\beta = \min(\max(0.01s, \beta_{\min}), \beta_{\max})$, which increases with training step s, gradually shifting the reward emphasis from the DGNN's soft guidance to exact category matching.

For message generation, we adopt the *LLM-as-a-judge* paradigm (Zhou et al., 2024; Peng et al., 2025) to assess social and semantic quality. A Qwen3-8B LLM rewarder (Yang et al., 2024) scores each generated message on six dimensions: Goal Fulfillment (GF) from SOTOPIA (Zhou et al., 2024); Contextual Fidelity (CF), Personality Depth (PD), Dynamic Adaptability (DA), Immersive Quality (IQ), and Content Richness (CR) from GDGB (Peng et al., 2025). Each dimension is rated on a [0, 5] scale; missing scores are treated as 0. The final message reward function is normalized and averaged:

$$r_{\text{text}} = \frac{1}{6} \sum_{d \in \mathcal{D}} \frac{\text{Score}_d}{5} + r_{\text{format}},$$

where $\mathcal{D} = \{\text{GF, CF, PD, DA, IQ, CR}\}$. For each domain, we define a task-specific reward with shared format regularization. Training proceeds via domain-interleaved sampling, with ratio typically 1:1 (category: message).

During training of Graphia-Q and Graphia-E, we first fine-tune the backbone LLM using SFT, then optimize both tasks with GRPO (Shao et al., 2024) based on the designed reward function.

Algorithm 1 Graph Generation Pipeline

Require: Historical graph sequence $\mathcal{G}_{\text{hist}} = \{G_1, \ldots, G_{T-\tau}\}$, future horizon τ
Ensure: Generated future graph sequence $\hat{\mathcal{G}}_{\text{fut}} = \{\hat{G}_{T-\tau+1}, \ldots, \hat{G}_T\}$

Stage 1: Activity Prediction
if Task is TDGG, $t = \{T-\tau+1, \ldots, T\}$ **then**
  Given source node set $\mathcal{U}_t$,
else Task is IDGG, $t = \{T-\tau+1, \ldots, T\}$
  Predict out-degrees $\{\hat{d}_t(u), u \in \mathcal{V}_t\}$,
  Source node set $\mathcal{U}_t = \{u \mid \exists t, \hat{d}_t(u) > 0\}$,
end if
Stage 2: Interaction Generation
for each t from $T-\tau+1$ to T **do**
  for each $u \in \mathcal{U}_t$ **do**
    GRAPHIA-Q(p_u, $\mathcal{M}_t(u)$)
      = Query, Filter
    Retrieve destination nodes $\hat{C}_t^u$
    for each $v \in \hat{C}_t^u$ **do**
      GRAPHIA-E(p_u, p_v, $\mathcal{M}_t(u, v)$)
        = $(\hat{m}_{u\to v}, \hat{y}_{u\to v})$
      Add $(u, v, \hat{m}_{u\to v}, \hat{y}_{u\to v})$ to $\hat{\mathcal{E}}_t$
      Add $(\hat{m}_{u\to v}, \hat{y}_{u\to v})$ to $\hat{\mathbf{X}}_t$
    end for
  end for
end for
Assemble $\hat{\mathcal{G}}_{\text{fut}} = \{(\mathcal{V}_t, \hat{\mathcal{E}}_t, \mathbf{P}_t, \hat{\mathbf{X}}_t)\}_{t=T-\tau+1}^{T}$
return $\hat{\mathcal{G}}_{\text{fut}}$

3.3 Graph Generation Pipeline

We design distinct generation pipelines for TDGG and IDGG to reflect their different evaluation focuses: TDGG emphasizes local agent behaviors while IDGG targets social network dynamics. As shown in Alg. 1, the process involves two stages: activity prediction and interaction generation.

TDGG Pipeline. In the transductive setting, the set of future source nodes is given. The full generation pipeline is: (1) For given source node u, condition Graphia-Q on the node profile p_u and node memory $\mathcal{M}_t(u)$ to generate the descriptive query. (2) Retrieve the destination node set $\hat{C}_t^u$ for source node u with the descriptive query. (3) For destination node $v \in \hat{C}_t^u$, condition Graphia-E on p_u, p_v, and node memory $\mathcal{M}_t(u, v)$ to generate the interaction message $(m_{u\to v}, y_{u\to v})$.

IDGG Pipeline. In the inductive setting, no future source nodes are provided. The full generation pipeline is: (1) Use the Activity-Predictor to predict out-degrees $\hat{d}_t(v)$ for all nodes. (2) Select active source nodes with $\hat{d}_t(v) > 0$. (3) For active source node u, apply the Graphia-Q for destination selection and Graphia-E for edge generation. (4) Assemble the full $\hat{\mathcal{G}}_{\text{fut}} = \{\hat{G}_{T-\tau+1}, \ldots, \hat{G}_T\}$.

4 Experiment

4.1 Experimental Setup

We evaluate both TDGG and IDGG tasks for social graph simulation. In our experiments, we adopt Qwen3-8B [*] as the backbone for Graphia.

Micro-Level Alignment Metrics. We propose the TDGG score (S_{TDGG}) to evaluate LLM-based agent's local social behavior. For destination selection, we measure whether Graphia-Q can identify interaction partners for a source node u. Given ground-truth destination set C_t^u at time t, we compute recall at 100 as $\text{R@100} = |C_t^u \cap \hat{C}_t^u|/|C_t^u|$, where $\hat{C}_t^u$ denotes the top-100 predicted destinations. We categorize samples into *Easy* and *Hard* based on the size of the ground-truth destination set $|C_t^u|$; if $|C_t^u|$ exceeds the 70-th percentile across all samples, it is labeled *Easy*, otherwise *Hard*. We report R@100 on Easy, Hard, and All samples. Metrics are normalized and aggregated into a summed average $S_{\text{selection}}$. For edge generation, we assess whether Graphia-E generates valid interaction messages by measuring category prediction accuracy (ACC) of $y_{u\to v}$, and evaluating ROUGE-L and BERTScore-F1 of the generated $\hat{m}_{u\to v}$ against reference message content $m_{u\to v}$. Metrics are normalized and aggregated into a summed average S_{edge}. The final TDGG score is $S_{\text{TDGG}} = 0.5 \cdot S_{\text{selection}} + 0.5 \cdot S_{\text{edge}}$.

Macro-Level Alignment Metrics. We propose the IDGG score (S_{IDGG}) to evaluate Graphia for predicting social graph structure and emergent social phenomena. For structure replication, we use Maximum Mean Discrepancy (MMD) with an RBF kernel to measure distributional distances in degree, clustering, and spectral properties (Peng et al., 2025). We also compute edge overlap, $\text{EO} = |\hat{\mathcal{E}}_{\text{fut}} \cap \mathcal{E}_{\text{fut}}|/|\mathcal{E}_{\text{fut}}|$. Metrics are normalized and aggregated into a summed average $S_{\text{structure}}$. For phenomenon replication, we evaluate: (i) Influencer identification. Following SaGraph (Zhang et al., 2025), we report P@100-KOL: the precision of KOLs in the top-100 degree nodes of $\hat{\mathcal{G}}_{\text{fut}}$. (ii) Echo chamber alignment. We measure the echo-chamber count difference ΔC (Del Vicario et al., 2016) between $\mathcal{G}_{\text{fut}}$ and $\hat{\mathcal{G}}_{\text{fut}}$. (iii) Power-

[*] https://huggingface.co/Qwen/Qwen3-8B

Table 1: Evaluation results for destination selection and edge generation in TDGG tasks. The best and second-best results are highlighted in **bold** and underline, respectively.

Model	Destination Selection					Edge Generation					TDGG	
	R@100-Easy ↑	R@100-Hard ↑	R@100-All ↑	S_{sel} ↑	Rank ↓	ACC ↑	ROUGE-L ↑	BERTScore ↑	S_{edge} ↑	Rank ↓	S_{TDGG} ↑	Rank ↓
Propagate-En												
Qwen3-8B	0.4451	0.3275	0.3634	0.2526	5.17	0.0136	0.6810	0.6157	0.2687	4.67	0.2606	6
Qwen3-8B-SFT	0.4601	0.3275	0.3677	0.5847	3.67	0.0153	0.7332	0.7622	0.6922	2.00	0.6385	2
Qwen3-32B	0.4444	0.3415	0.3718	0.6108	3.33	0.0088	0.6626	0.5117	0.0000	7.00	0.3054	4
DeepSeek-Q-32B	0.4617	0.3125	0.3582	0.1749	5.50	0.0101	0.6668	0.5192	0.0439	6.00	0.1094	7
LLama3.1-70B	0.4418	**0.3437**	0.3735	0.6261	3.00	0.0136	0.6978	0.6794	0.4184	3.33	0.5223	3
Graphia-seq	0.4439	0.3297	0.3641	0.2741	5.33	0.0145	0.6824	0.6177	0.2881	3.67	0.2811	5
Graphia	**0.4763**	0.3319	**0.3761**	**0.8739**	**1.67**	**0.0346**	**0.7421**	**0.7799**	**1.0000**	**1.00**	**0.9370**	**1**
Weibo Tech												
Qwen3-8B	0.2602	0.2301	0.2455	0.5612	4.50	0.6326	0.6014	0.1757	0.5367	3.00	0.5490	2
Qwen3-8B-SFT	0.2422	0.2250	0.2334	0.0455	6.33	0.7325	0.5985	0.2128	0.6920	2.67	0.3687	6
Qwen3-32B	0.2641	0.2346	0.2498	0.8138	2.50	0.5765	0.5762	0.0873	0.0314	6.67	0.4226	5
DeepSeek-Q-32B	**0.2767**	0.2235	0.2518	0.6579	2.83	0.5846	0.5790	0.0656	0.0449	6.33	0.3514	7
LLama3.1-70B	0.2607	0.2283	0.2448	0.5256	4.83	0.6453	0.5929	0.1459	0.4095	4.33	0.4675	4
Graphia-seq	0.2629	0.2232	0.2438	0.4146	5.33	0.6297	0.6007	0.1747	0.5230	4.00	0.4688	3
Graphia	0.2700	**0.2364**	**0.2538**	**0.9292**	**1.50**	**0.8221**	**0.6040**	**0.2963**	**1.0000**	**1.00**	**0.9646**	**1**
Weibo Daily												
Qwen3-8B	0.3326	0.3030	0.3135	0.6719	4.83	0.5456	0.5661	0.1243	0.4201	2.67	0.5460	3
Qwen3-8B-SFT	0.3096	0.3060	0.3072	0.3033	5.50	0.6338	0.5755	0.0735	0.4625	2.67	0.3829	6
Qwen3-32B	0.3344	**0.3159**	**0.3226**	**0.9362**	**1.50**	0.3921	0.5332	0.0142	0.0438	6.67	0.4900	4
DeepSeek-Q-32B	**0.3401**	0.2769	0.2997	0.3675	4.83	0.4127	0.5332	-0.0238	0.0140	6.33	0.1908	7
LLama3.1-70B	0.3259	0.3127	0.3173	0.7470	3.33	0.5332	0.5450	0.0490	0.2315	5.00	0.4892	5
Graphia-seq	0.3325	0.3042	0.3142	0.6844	4.33	0.5422	0.5657	0.1224	0.4138	3.67	0.5491	2
Graphia	0.3379	0.3042	0.3162	0.7411	3.17	**0.8836**	**0.6088**	**0.2652**	**1.0000**	**1.00**	**0.8706**	**1**
Average Performance												
Best Baseline	-	-	-	0.7869	2.44	-	-	-	0.6156	2.44	-	-
Graphia	-	-	-	**0.8481**	**2.11**	-	-	-	**1.0000**	**1.00**	-	-

law fitness. We measure the power-law exponent gap $\Delta\alpha = |\alpha_{\text{ref}} - \alpha_{\text{gen}}|$, where α_{ref} and α_{gen} are fitted exponents for $\mathcal{G}_{\text{fut}}$ and $\hat{\mathcal{G}}_{\text{fut}}$ (Peng et al., 2025). Metrics are normalized and aggregated into a summed average $S_{\text{phenomenon}}$. The final IDGG score is $S_{\text{IDGG}} = 0.5 \cdot S_{\text{structure}} + 0.5 \cdot S_{\text{phenomenon}}$.

Baseline Models. To assess the impact of graph-structured data on training LLMs, we construct sequential data that retains only first-order neighbor edges while discarding higher-order topology. For the TDGG task, which focuses on micro-level behavioral alignment, we compare Graphia against a range of LLMs of varying scale: Qwen3-8B, Qwen3-32B, DeepSeek-R1-Distill-Qwen-32B, and Llama-3.1-70B-Instruct, as well as a fine-tuned Qwen3-8B baseline (Qwen3-SFT). We further evaluate Graphia-seq, a variant of Graphia trained on sequential interaction data based on Qwen3-8B, using destination selection rewards from LIKR (Sakurai et al., 2025) and edge generation rewards from Sotopia (Zhou et al., 2024). For the IDGG task, we compare Graphia with deep-learning and LLM-based social graph generators. First, we adopt DGGen (Hosseini et al., 2025) and TIGGER (Gupta et al., 2022), the only deep-learning models designed for inductive dynamic graph generation; we further note that DGGen uses TGN (Rossi et al., 2020) as its backbone to learn temporal graph embeddings. Following the DGGen framework, we replace the backbone with more recent temporal graph learning models (Zhang et al., 2024a), yielding DGGen variants with GraphMixer (Cong et al., 2023), CAWN (Wang et al., 2021), and Dygformer (Yu et al., 2023) backbones. Second, we construct two hybrid LLM-based simulators by pairing Qwen3-SFT and Graphia-seq with the SA-Graph activity-prediction module (Zhang et al., 2024b), which fits Gaussian distributions to activity patterns and samples active nodes. Finally, we include GAG-General (Peng et al., 2025) with Llama3-8B and Qwen3-8B backbones, an LLM-based multi-agent framework for graph generation.

Table 2: Key statistics of the social network datasets.

Dataset	Propagate-En	Weibo Tech	Weibo Daily
Node Count	5,634	20,768	66,501
Edge Count	16,962	58,930	195,202
Input (Days)	18	5	19
Prediction (Days)	4	1	4
Total (Days)	30	8	31
Category Number	534	2	2

Datasets. We evaluate our framework on three social network datasets: Propagate-En, collected from the Taobao e-commerce platform, and two public datasets: Weibo Tech and Weibo Daily from GDGB (Peng et al., 2025). Table 2 summarizes the dataset statistics, which are aggregated by day based on edge timestamps. Let T_2 be the total time length and $\tau = \lfloor 0.15 \times T_2 \rfloor$. The input length is set to $T - \tau = T_2 - 3\tau$, and the prediction length is

τ. Training, validation, and test sets are chronologically partitioned into intervals of $[0, T]$, $[\tau, T+\tau]$, and $[2\tau, T+2\tau]$. Detailed dataset statistics and preprocessing are provided in Appendix A.

4.2 TDGG: Micro-Level Alignment

Destination Selection. As shown in Table 1 on the left side, Graphia achieves competitive performance in destination selection. First, ablation studies validate the effectiveness of our training framework. Both Qwen3-8B and Qwen3-8B-SFT underperform compared to Graphia. Graphia achieves an average rank of 2.11, outperforming Qwen3-8B-SFT by a margin of 3. Second, despite being built on an 8B-parameter backbone, Graphia achieves performance comparable to or even exceeding that of larger LLMs. On Propagate-En, it outperforms Qwen3-32B (rank: 3.33), DeepSeek-Q-32B (rank: 5.50), and Llama3.1-70B (rank: 3.00), achieving the highest average rank of 1.67. While on Weibo Daily, Graphia ranks behind Qwen3-32B; In the aggregated selection score, Graphia achieves an average of $S_{\text{sel}} = 0.848$ across datasets, surpassing the best baseline Qwen3-32B by 6.1% ($S_{\text{sel}} = 0.787$). These results demonstrate Graphia's superior micro-level alignment capabilities, matching or exceeding the performance of much larger models in destination selection.

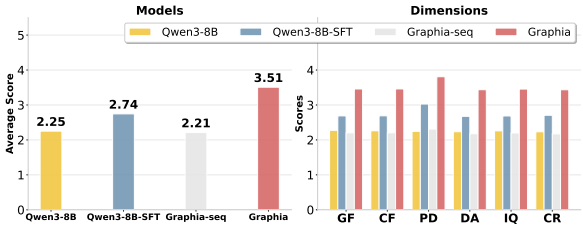


Figure 2: LLM-as-a-judge for edge generation.

Edge Generation. Inspired by SOTOPIA, we first adopt the LLM-as-a-judge for edge evaluation. Using Qwen2-72B as the evaluator, $m_{u\to v}$ is scored across six dimensions (Yang et al., 2024): *Goal Fulfillment* (GF) from SOTOPIA (Zhou et al., 2024), and *Contextual Fidelity* (CF), *Personality Depth* (PD), *Dynamic Adaptability* (DA), *Immersive Quality* (IQ), *Content Richness* (CR) from GDGB (Peng et al., 2025). Ratings range from 1 to 5 (missing values treated as zero). As shown in Figure 2, we evaluate four models: Qwen3-8B, Qwen3-8B-SFT, Graphia-seq, and Graphia. While Graphia-seq uses the SOTOPIA-RL (Yu et al., 2025) reward framework: using LLM-as-a-judge scoring *Goal Fulfillment*, *Relationship Maintenance*, and *Knowledge Seeking* for GRPO training; we find this reward design for sequential data does not improve LLM-judge scores on generated edge messages. In contrast, Graphia achieves the highest average score, outperforming the best baseline by 0.77 points (+28% relative), with the largest gain in *Personality Depth* by 0.78 points (+25.7% relative). Since LLM-as-a-judge is vulnerable to manipulation (Li et al., 2024), we evaluate Graphia with four different LLMs as judge; results consistently show it outperforms baselines (see Appendix D.4, include Qwen3-32B, Qwen2-72B LLama3.1-70B and Llama3.3-70B).

To ensure a more robust evaluation, we complement LLM-as-a-judge scores with automatic metrics, specifically category accuracy (ACC) for $\hat{y}_{u\to v}$ and content similarity metrics for $\hat{m}_{u\to v}$. These metrics are then normalized and aggregated into S_{edge}. As shown in Table 1, Graphia consistently leads on edge generation metrics across datasets. For category accuracy, Graphia improves over the best baseline (Qwen3-8B-SFT) by 1.9% on Propagate-En, 24.98% on Weibo Daily, and 9% on Weibo Tech, resulting in an average improvement of 12% in category prediction accuracy. For message content similarity, Graphia improves ROUGE-L by 0.016 points (+2.5% relative). More notably, it improves BERTScore by an average increase of 0.098 points (+27.9% relative). Graphia maintains a rank of 1 across all evaluation metrics. In the combined metric $S_{\text{edge}} = 1$, Graphia achieves $S_{\text{edge}} = 1$ on average, surpassing the best baseline by 38.4% ($S_{\text{edge}} = 0.6156$).

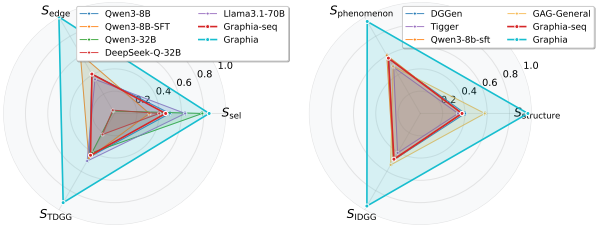
4.3 IDGG: Macro-Level Alignment

Macro Structure Replication. We evaluate all baselines on the three datasets, omitting results for those that encounter out-of-memory (OOM) errors. As shown in Table 3, Graphia consistently attains the lowest MMD.D^2, MMD.C^2 and MMD.S^2 scores across all datasets. While DGGen performs competitively on Propagate-En, LLM-based generators, including Graphia, consistently outperform deep learning baselines on both Weibo datasets. Notably, most deep learning models yield near-zero edge overlap (EO), reflecting a significant gap in edge distribution from reference graphs. In contrast, LLM-based methods generate non-zero EO, with Graphia achieving the highest EO on all datasets. Averaged across datasets, Graphia attains $S_{\text{structure}} = 0.97$, surpassing the best baseline by

Table 3: Evaluation results for social graph structure and social phenomenon replication in IDGG tasks.

Model	Macro Structure						Macro Phenomenon					IDGG	
	MMD.D^2 ↓	MMD.C^2 ↓	MMD.S^2 ↓	EO ↑	$S_{\text{structure}}$ ↑	Rank ↓	P@100-KOL ↑	ΔC ↓	$\Delta\alpha$ ↓	$S_{\text{phenomenon}}$ ↑	Rank ↓	S_{IDGG} ↑	Rank ↓
Propagate-En													
Qwen3-8B-sft	0.3509	0.4128	0.3739	0.0608	0.4054	5.25	0.27	33.0000	1.0884	0.3054	4.67	0.3554	5.00
DGGen	0.1486	**0.3336**	**0.1327**	0.0000	0.6602	2.50	0.02	13.0000	0.1614	0.5283	3.00	0.5942	2.00
DGGen(GraphMixer)	0.1601	0.4315	0.3253	0.0018	0.4531	5.00	0.00	30.0000	1.1210	0.0862	6.00	0.2697	7.00
DGGen(DyGFormer)	0.1801	1.4142	0.4006	0.0000	0.1352	6.75	0.02	18.0000	1.3355	0.1793	5.00	0.1573	8.00
DGGen(CAWN)	0.1495	0.3753	0.3158	0.0073	0.4967	3.75	0.01	31.0000	1.2745	0.0458	6.33	0.2713	6.00
Tigger	0.2067	1.3563	0.2613	0.0000	0.2575	5.50	0.01	21.0000	0.0216	0.4685	4.00	0.3630	3.00
Graphia-seq	0.3406	0.3522	0.3797	0.0608	0.4222	4.50	0.28	33.0000	1.1728	0.2932	5.00	0.3577	4.00
Graphia	**0.0351**	0.3557	0.1981	**0.1022**	**0.9339**	**1.75**	**0.37**	**2.0000**	**0.0100**	**1.0000**	**1.00**	**0.9669**	**1.00**
Weibo Tech													
Qwen3-8B-sft	0.2623	1.2628	0.4772	0.0143	0.2603	4.25	0.30	16.0000	1.0828	0.8186	3.67	0.5395	2.00
DGGen	0.1870	1.2979	0.5472	0.0001	0.2363	5.25	0.01	18.0000	0.5434	0.5909	4.67	0.4136	5.00
DGGen(GraphMixer)	0.1292	1.4142	0.5636	0.0000	0.2236	5.75	0.01	16.0000	1.1252	0.5102	4.67	0.3669	6.00
GAG-General(Qwen3-8B)	0.4422	1.4141	0.3503	0.0000	0.1157	6.00	0.00	112.0000	0.4446	0.2851	5.67	0.2004	8.00
GAG-general(Llama3-8B)	**0.0922**	1.1131	0.1904	0.0000	0.5687	2.75	0.00	66.0000	0.1889	0.4753	5.00	0.5220	4.00
Tigger	0.3349	1.2840	0.1727	0.0000	0.3390	4.75	0.00	14.0000	2.3443	0.3234	5.33	0.3312	7.00
Graphia-seq	0.2692	1.2537	0.4948	0.0144	0.2496	4.25	0.27	16.0000	1.0113	0.7981	3.67	0.5239	3.00
Graphia	0.1467	**0.7668**	**0.1027**	**0.1347**	**0.9611**	**1.50**	**0.32**	**11.0000**	**0.1230**	**1.0000**	**1.00**	**0.9805**	**1.00**
Weibo Daily													
Qwen3-8B-sft	0.3234	0.8353	0.4558	0.0253	0.4444	4.25	**0.44**	**0.0000**	1.0467	0.6830	**2.67**	0.5637	2.00
DGGen	0.2126	0.9267	0.7379	0.0000	0.3087	5.00	0.00	1.0000	0.3315	0.5815	3.00	0.4451	6.00
GAG-General(Qwen3-8B)	0.5698	1.4142	0.3392	0.0000	0.1416	5.50	0.00	123.0000	0.7887	0.1010	5.33	0.1213	7.00
GAG-general(Llama3-8B)	0.1362	0.7363	0.2065	0.0000	0.5869	2.75	0.00	4.0000	0.5478	0.5025	4.33	0.5447	3.00
Tigger	0.2098	1.4102	0.0648	0.0000	0.4171	3.75	0.00	8.0000	**0.0802**	0.6450	3.67	0.5310	5.00
Graphia-seq	0.3342	0.7786	0.4735	0.0262	0.4506	4.25	0.40	3.0000	1.0966	0.6282	4.00	0.5394	4.00
Graphia	**0.0614**	**0.4983**	**0.0338**	**0.0973**	**1.0000**	**1.00**	0.32	3.0000	0.2074	**0.8593**	**2.67**	**0.9296**	**1.00**
Average Performance													
Best Baseline	-	-	-	-	0.6052	2.66	-	-	-	0.6766	3.11	-	-
Graphia	-	-	-	-	**0.9650**	**1.42**	-	-	-	**0.9531**	**1.56**	-	-

35.98% ($S_{\text{structure}} = 0.61$).



(a) TDGG Evaluation (b) IDGG Evaluation

Figure 3: The social fidelity score for TDGG and IDGG tasks. (a) Graphia outperforms baseline in edge generation, achieves equal performance with 32B in destination selection. (b) Graphia outperforms baselines in graph structure and phenomena replication, achieves best performance compared to both deep-learning based and LLM-based social graph generators.

Macro Phenomenon Replication. To quantitatively assess emergent societal phenomena, we introduce three metrics: P@100-KOL, echo chamber alignment (ΔC), and deviation in power-law exponent ($\Delta\alpha$). These metrics address the limitations of existing LLM-based simulators, which rely on qualitative assessments of phenomena such as echo chambers (Zheng and Tang, 2024; Wang et al., 2025a), power-law distributions (Du et al., 2025; Ji et al., 2025), and influencer selection (Zhang et al., 2025). Graphia achieves the best performance in these metrics on Weibo-Tech and Propagate-En, and ranks 2.67 in Weibo-Daily. Averaged across datasets, Graphia achieves $S_{\text{phenomenon}} = 0.95$, surpassing the best baseline by 28.71% ($S_{\text{phenomenon}} = 0.68$), suggesting it's reliable for exploration of phenomena in social graphs.

5 Conclusion

In this paper, we address two critical limitations in LLM-based social simulation: (i) the absence of a generalizable training framework that leverages graph-structured data to enhance microscopic and macroscopic social simulation realism, and (ii) the lack of unified quantitative metrics to assess the alignment between simulated and real-world social graphs. To bridge these gaps, we make two key contributions. First, we propose Graphia, a general social graph generator that treats social graph as high-quality supervision signals for LLM post-training. Graphia trains specialized LLM-based agents to model human-like interactions by predicting whom to interact and how to interact, followed by carefully designed graph generation pipelines. Second, we establish a unified evaluation paradigm for TDGG and IDGG tasks, with quantitative metrics to assess both micro-level interaction and macro-level realism in social graph simulation. Experiments on three real-world datasets validate both contributions: Graphia improves micro-level

alignment by 6.1% in destination selection, 12% in edge classification accuracy, and 27.9% in edge content BERTScore; simultaneously, it achieves 35.98% higher structural similarity and 28.71% better replication of emergent social phenomena for macro-level alignment. This shows that social graphs are effective supervision signals for LLMs, bridging microscopic agent behaviors and macroscopic network dynamics in social simulation.

Limitations

This paper acknowledges several limitations that future research could address:

Analysis of Learned Policies. Our primary focus is improving alignment between LLM-generated and real-world social graph simulation via reward feedback from social graphs. Our study focuses on who, how, and when agents interact within the simulated social network. The question of why agents interact remains outside the scope of this work.Yet, a core tenet of social network analysis is that agent behaviors stem from interpretable causal mechanisms, and that these micro-level mechanisms give rise to distinct macro-level network phenomena. Future work can build upon Graphia's framework to explicitly investigate the causal drivers of both micro-level agent decisions and emergent macro-level graph structures.

Incorporation of Structural Rewards. While our framework uses GNN-derived rewards to align LLM-generated edges with real-world graph edges, the current implementation relies solely on node-level destination selection and edge-category logits from a pre-trained dynamic GNN. Higher-order topological properties (such as community cohesion, triadic closure) are not explicitly captured in the reward function. Future work could investigate learned filtering mechanisms or graph-structure-aware reward designs that explicitly incorporate complex topological signals, enabling better generalization across diverse graph regimes.

Acknowledgments

This research was supported in part by National Natural Science Foundation of China (No. 92470128, No. U2241212), by Alibaba Group through Alibaba Innovative Research Program. We also wish to acknowledge the support provided by the fund for building world-class universities (disciplines) of Renmin University of China, by Engineering Research Center of Next-Generation Intelligent Search and Recommendation, Ministry of Education, by Intelligent Social Governance Interdisciplinary Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Public Policy and Decision-making Research Lab, and Public Computing Cloud, Renmin University of China.

References

Xiaohui Chen, Jiaxing He, Xu Han, and Liping Liu. 2023. Efficient and degree-guided graph generation via discrete diffusion modeling. In Int. Conf. Machin. Learn., ICML, pages 4585–4610. PMLR.

Weilin Cong, Si Zhang, Jian Kang, Baichuan Yuan, Hao Wu, Xin Zhou, Hanghang Tong, and Mehrdad Mahdavi. 2023. Do we really need complicated model architectures for temporal networks? In Proc. Int. Conf. Learn. Represent., ICLR.

Michela Del Vicario, Alessandro Bessi, Fabiana Zollo, Fabio Petroni, Antonio Scala, Guido Caldarelli, H Eugene Stanley, and Walter Quattrociocchi. 2016. The spreading of misinformation online. Proceedings of the national academy of Sciences, PNAS, 113(3):554–559.

Enjun Du, Xunkai Li, Tian Jin, Zhihan Zhang, Rong-Hua Li, and Guoren Wang. 2025. Graphmaster: Automated graph synthesis via LLM agents in data-limited environments. CoRR, abs/2504.00711.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 82 others. 2024. The llama 3 herd of models. CoRR, abs/2407.21783.

Peter Eigenschink, Thomas Reutterer, Stefan Vamosi, Ralf Vamosi, Chang Sun, and Klaudius Kalcher. 2023. Deep generative models for synthetic data: A survey. IEEE Access, 11:47304–47320.

Chen Gao, Xiaochong Lan, Nian Li, Yuan Yuan, Jingtao Ding, Zhilun Zhou, Fengli Xu, and Yong Li. 2024. Large language models empowered agent-based modeling and simulation: A survey and perspectives. Humanities and Social Sciences Communications, 11(1):1–24.

Chen Gao, Xiaochong Lan, Zhihong Lu, Jinzhu Mao, Jinghua Piao, Huandong Wang, Depeng Jin, and Yong Li. 2023. S^3: Social-network simulation system with large language model-empowered agents. CoRR, abs/2307.14984.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai

Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. Nature, 645(8081):633–638.

Shubham Gupta, Sahil Manchanda, Srikanta Bedathur, and Sayan Ranu. 2022. TIGGER: scalable generative modelling for temporal interaction graphs. In Proc. AAAI Conf. Artif. Intell., AAAI, pages 6819–6828.

Ryien Hosseini, Filippo Simini, Venkatram Vishwanath, and Henry Hoffmann. 2025. A deep probabilistic framework for continuous time dynamic graph generation. In Proc. AAAI Conf. Artif. Intell., AAAI, pages 17249–17257.

Jiarui Ji, Runlin Lei, Jialing Bi, Zhewei Wei, Xu Chen, Yankai Lin, Xuchen Pan, Yaliang Li, and Bolin Ding. 2025. Llm-based multi-agent systems are scalable graph generative models. In Find. Annu. Meet. Assoc. Comput Linguist., ACL, pages 1492–1523.

Runlin Lei, Jiarui Ji, Haipeng Ding, Lu Yi, Zhewei Wei, Yongchao Liu, and Chuntao Hong. 2025. Exploring the potential of large language models as predictors in dynamic text-attributed graphs. CoRR, abs/2503.03258.

Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. 2024. Llms-as-judges: A comprehensive survey on llm-based evaluation methods. CoRR, abs/2412.05579.

Yuhan Liu, Xiuying Chen, Xiaoqing Zhang, Xing Gao, Ji Zhang, and Rui Yan. 2024. From skepticism to acceptance: Simulating the attitude dynamics toward fake news. In Proc. Int. Joint Conf. Artif. Intell., IJCAI.

Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. Annual review of sociology, 27(1):415–444.

Xinyi Mou, Zhongyu Wei, and Xuanjing Huang. 2024. Unveiling the truth and facilitating change: Towards agent-based large-scale social movement simulation. In Find. Annu. Meet. Assoc. Comput Linguist., ACL, pages 4789–4809.

Jie Peng, Jiarui Ji, Runlin Lei, Zhewei Wei, Yongchao Liu, and Chuntao Hong. 2025. GDGB: A benchmark for generative dynamic text-attributed graph learning. CoRR, abs/2507.03267.

Jinghua Piao, Yuwei Yan, Jun Zhang, Nian Li, Junbo Yan, Xiaochong Lan, Zhihong Lu, Zhiheng Zheng, Jing Yi Wang, Di Zhou, Chen Gao, Fengli Xu, Fang Zhang, Ke Rong, Jun Su, and Yong Li. 2025. Agentsociety: Large-scale simulation of llm-driven generative agents advances understanding of human behaviors and society. CoRR, abs/2502.08691.

Giulio Rossetti, Massimo Stella, Rémy Cazabet, Katherine Abramski, Erica Cau, Salvatore Citraro, Andrea Failla, Riccardo Improta, Virginia Morini, and Valentina Pansanella. 2024. Y social: an llm-powered social media digital twin. CoRR, abs/2408.00818.

Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. 2020. Temporal graph networks for deep learning on dynamic graphs. In ICML 2020 Workshop on Graph Representation Learning.

Keigo Sakurai, Ren Togo, Takahiro Ogawa, and Miki Haseyama. 2025. LLM is knowledge graph reasoner: Llm's intuition-aware knowledge graph reasoning for cold-start sequential recommendation. In Proc. Eur. Conf. Inf. Retr., ECIR, volume 15573 of Lecture Notes in Computer Science, pages 263–278.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300.

Chenxi Wang, Zongfang Liu, Dequan Yang, and Xiuying Chen. 2025a. Decoding echo chambers: Llm-powered simulations revealing polarization in social networks. In Proc. Int. Conf. Comput. Linguist., COLING, pages 3913–3923.

Xintao Wang, Heng Wang, Yifei Zhang, Xinfeng Yuan, Rui Xu, Jen-tse Huang, Siyu Yuan, Haoran Guo, Jiangjie Chen, Shuchang Zhou, and 1 others. 2025b. CoSER: Coordinating LLM-based persona simulation of established roles. In Proc. Int. Conf. Machin. Learn., ICML.

Yanbang Wang, Yen-Yu Chang, Yunyu Liu, Jure Leskovec, and Pan Li. 2021. Inductive representation learning in temporal networks via causal anonymous walks. In Proc. Int. Conf. Learn. Represent., ICLR. OpenReview.net.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 40 others. 2025. Qwen3 technical report. CoRR, abs/2505.09388.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. Qwen2 technical report. Preprint, arXiv:2407.10671.

Junchi Yao, Hongjie Zhang, Jie Ou, Dingyi Zuo, Zheng Yang, and Zhicheng Dong. 2025. Social opinions prediction utilizes fusing dynamics equation with llm-based agents. Scientific Reports, 15(1):15472.

Haofei Yu, Zhengyang Qi, Yining Zhao, Kolby Nottingham, Keyang Xuan, Bodhisattwa Prasad Majumder, Hao Zhu, Paul Pu Liang, and Jiaxuan You. 2025. Sotopia-rl: Reward design for social intelligence. CoRR, abs/2508.03905.

Le Yu, Leilei Sun, Bowen Du, and Weifeng Lv. 2023. Towards better dynamic graph learning: New architecture and unified library. In Proc. Adv. neural inf. proces. syst., NeurIPS.

Jiasheng Zhang, Jialin Chen, Menglin Yang, Aosong Feng, Shuang Liang, Jie Shao, and Rex Ying. 2024a. DTGB: A comprehensive benchmark for dynamic text-attributed graphs. In Proc. Adv. neural inf. proces. syst., NeurIPS.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In Proc. Int. Conf. Learn. Represent., ICLR.

Xiaoqing Zhang, Xiuying Chen, Yuhan Liu, Jianzhou Wang, Zhenxing Hu, and Rui Yan. 2024b. A large-scale time-aware agents simulation for influencer selection in digital advertising campaigns. CoRR, abs/2411.01143.

Xiaoqing Zhang, Yuhan Liu, Jianzhou Wang, Zhenxing Hu, Xiuying Chen, and Rui Yan. 2025. Sagraph: A large-scale social graph dataset with comprehensive context for influencer selection in marketing. In Proc. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr., SIGIR, pages 3733–3742.

Wenzhen Zheng and Xijin Tang. 2024. Simulating social network with llm agents: An analysis of information propagation and echo chambers. In Proc. Int. Conf. Learn. Represent., ICLR, pages 63–77. Springer.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In Proc. AAAI Conf. Artif. Intell., AAAI, volume 35, pages 11106–11115.

Lingfeng Zhou, Jialing Zhang, Jin Gao, Mohan Jiang, and Dequan Wang. 2025. Personaeval: Are LLM evaluators human enough to judge role-play? In Proc. Second Conf. Language Modeling, COLM.

Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Haofei Yu, Zhengyang Qi, Louis-Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, and Maarten Sap. 2024. SOTOPIA: interactive evaluation for social intelligence in language agents. In Proc. Int. Conf. Learn. Represent., ICLR.

Appendix: Contents

A Details of Dataset

We provide additional details about the three social network datasets used in our experiments.

- **Propagate-En.** A product-sharing social network collected from an e-commerce platform, where nodes represent taokes (content influencers) and edges indicate forwarding behaviors of promotional content.
- **Weibo Tech.** A subgraph of the Weibo network focused on technology-related topics, capturing information diffusion among tech influencers.
- **Weibo Daily.** A general-topic Weibo network with broader user coverage, reflecting daily social interactions and news propagation.

All data are binned into daily snapshots. For temporal splitting, let T_2 denote the total duration (in days). We define $\tau = \lfloor 0.15 \times T_2 \rfloor$ and use:

- **Train Split.** $[0, T)$, where $T = T_2 - 3\tau$
- **Validation Split.** $[T, T + \tau)$, used to predict $[T + \tau, T + 2\tau)$
- **Test Split**: $[T + \tau, T + 2\tau)$, used to predict $[T + 2\tau, T_2)$

Thus, both the training, validation, and test splits use an input history of length T days to predict the next τ days, ensuring a consistent evaluation protocol across all phases. The full statistics, including average edge duration (i.e., the mean lifespan of each edge in days), are shown in Table 4.

B Details of Metric

We provide detailed mathematical formulations and implementation specifics for the IDGG and TDGG social fidelity Scores introduced in Section 4.1.

First, we define the dataset-wise normalization function for different metrics. To map all component metrics to [0,1] with a positive direction (higher is better), we apply min–max normalization per metric within each dataset.

For positive-direction metric value x, its normalized score is:

$$\overline{x} = \frac{x - \min(x)}{\max(x) - \min(x)},$$

where $\min(x)$ and $\max(x)$ are computed over the metric values of across all models evaluated on the same dataset. The positive-direction metrics include: R@100-Easy, R@100-Hard, R@100-All, ACC, ROUGE-L, BERTScore-F1, EO, and P@100-KOL.

For negative-direction metrics (smaller is better) x, we use reversed normalized score as:

$$1 - \overline{x} = 1 - \frac{x - \min(x)}{\max(x) - \min(x)}.$$

The negative-direction metrics include: $\mathrm{MMD.D}^2, \mathrm{MMD.C}^2, \mathrm{MMD.S}^2$, D, and Chambers Diff. This normalization is applied independently to each metric prior to aggregation into the final scores.

B.1 TDGG Social Fidelity Score

The TDGG social fidelity score (S_{TDGG}) evaluates micro-level behavioral authenticity by combining destination selection fidelity and edge generation quality.

Destination Selection. Following PersonaEval, we assess whether models select accurate partners using R@100 (Zhou et al., 2025), the fraction of true destinations ranked among the top-100 predicted nodes. To differentiate difficulty, we categorize target nodes by out-degree $d(u)$: **Easy**: $d(u) >$ 70-th percentile degree (i.e., hub nodes), **Hard**: $d(u) \leq$ 70-th percentile degree (i.e., non-hub nodes). We compute These are normalized and averaged into the selection score:

$$S_{\text{selection}} = \frac{1}{3}(\overline{\text{R@100-Easy}} + \overline{\text{R@100-Hard}} + \overline{\text{R@100-All}}).$$

Edge Generation. To evaluate the quality of generated interaction messages, we assess both the predicted message category $\hat{y}_{u\to v}$ against the reference label $y_{u\to v}$, and the gener ated message text $\hat{m}_{u\to v}$ against the reference message $m_{u\to v}$ using ROUGE-L and BERTScore-F1.

- **Category Accuracy.** Measures the accuracy (ACC) of predicted message types (e.g., question, greeting, request).For the set of evaluated edges $\mathcal{E}$, it is defined as::

$$\text{ACC} = \frac{1}{|\mathcal{E}|} \sum_{(u,v)\in\mathcal{E}} \mathbb{I}\left(\hat{y}_{u\to v} = y_{u\to v}\right),$$

where $\hat{y}_e$ and y_e denote the predicted and true message categories for edge e, respectively, and $\mathbb{I}(\cdot)$ is the indicator function.

Table 4: Extended dataset statistics.

Dataset	Propagate-En	Weibo Tech	Weibo Daily
Node Count	5,634	20,768	66,501
Edge Count	16,962	58,930	195,202
Input Length (Days)	18	5	19
Prediction Length (Days)	4	1	4
Total Length (Days)	30	8	31
Avg. Edge (Days)	565.40	7,366.25	6,296.84
Number of Categories	534	2	2
Input Edge Count (Test)	10,119	45,623	125,073
Prediction Edge Count (Test)	2,283	8,618	47,980

- **ROUGE-L.** Evaluates the similarity between generated and reference edge messages using the longest common subsequence (LCS). It measures n-gram co-occurrence with flexibility in word order, making it robust to syntactic variations. Formally:

$$\text{ROUGE-L} = \frac{\sum_{e\in\mathcal{E}} \text{LCS}(\hat{m}_e, m_e)}{\sum_{e\in\mathcal{E}} |m_e|},$$

  where $\text{LCS}(\hat{m}_e, m_e)$ denotes the length of the longest common subsequence between the generated message $\hat{m}_e$ and its ground truth m_e, and $|m_e|$ is the token length of the reference message. The final score is computed as an average over all edges in the evaluation set.

- **BERTScore-F1.** We adopt BERTScore to compute a contextual F1 score between generated and reference edge messages (Zhang et al., 2020), leveraging pretrained contextual embeddings for more semantically meaningful similarity measurement. Formally, for each edge $e \in \mathcal{E}$, we compare the generated message $\hat{m}_e$ with its ground truth m_e, and compute BERTScore over the entire set of $(m_e, \hat{m}_e)$ pairs.

These metrics are normalized and averaged into the edge sub-score:

$$S_{\text{edge}} = \frac{1}{3}(\overline{\text{ACC}} + \overline{\text{ROUGE-L}} + \overline{\text{BERTScore-F1}}).$$

(3) TDGG Social Fidelity Score. The final score is computed as:

$$S_{\text{TDGG}} = 0.5 \cdot S_{\text{selection}} + 0.5 \cdot S_{\text{edge}},$$

with equal weighting reflecting balanced importance between structural attention and semantic realism.

B.2 IDGG Social Fidelity Score

The IDGG social fidelity score (S_{IDGG}) evaluates macro-level realism through macro-level structural and phenomenological realism.

Macro Structure Fidelity. We assess structural similarity using distributional distances measured by Maximum Mean Discrepancy (MMD) with an RBF kernel:

$$\begin{aligned}\text{MMD}^2(X,Y) = &\frac{1}{n^2}\sum_{i,j=1}^{n} k(x_i, x_j) \\ &+ \frac{1}{m^2}\sum_{i,j=1}^{m} k(y_i, y_j) \\ &- \frac{2}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} k(x_i, y_j),\end{aligned}$$

where $k(\cdot,\cdot)$ is the RBF kernel $k(a,b) = \exp(-\|a-b\|^2/2v^2)$. In our evaluation, we report MMD.D^2 for degree distribution, MMD.C^2 for cluster coefficient, and MMD.S^2 for spectral properties, consistent with common practice in graph generation literature. Additionally, we compute edge overlap ratio for future edges:

$$\text{EO} = \frac{|\hat{\mathcal{E}}_{\text{fut}} \cap \mathcal{E}_{\text{fut}}|}{|\mathcal{E}_{\text{fut}}|}.$$

These metrics are normalized and averaged into the structure fidelity score:

$$\begin{aligned}S_{\text{structure}} = \frac{1}{4}((1 - \overline{\text{MMD.D}^2}) + \\ (1 - \overline{\text{MMD.C}^2}) + \\ (1 - \overline{\text{MMD.S}^2}) + \overline{\text{EO}}).\end{aligned}$$

.

Macro Phenomenon Consistency. We evaluate three canonical social phenomena:

- **Influencer Selection.** Inspired by SaGraph (Zhang et al., 2025), we evaluate the model's ability to predict key opinion leaders (KOLs) using P@100-KOL, which measures the precision in recovering the most influential nodes in the future graph. We define the ground-truth KOLs as the 100 nodes with the highest PageRank scores in the reference future graph $\mathcal{G}_{\text{fut}}$. The predicted KOLs are taken as the top-100 nodes ranked by PageRank in the generated graph $\hat{\mathcal{G}}_{\text{fut}}$. P@100-KOL is then computed as the fraction of true KOLs that appear among the predicted top-100. A higher score indicates better alignment in capturing central influencers.

- **Echo Chamber Alignment.** To quantify ideological polarization, we detect tightly-knit, ideologically homogeneous communities (i.e., echo chambers) in both the reference graph $\mathcal{G}_{\text{fut}}$ and the generated graph $\hat{\mathcal{G}}_{\text{fut}}$, following prior work (Zheng and Tang, 2024; Wang et al., 2025a). The number of such chambers is extracted via community detection under polarization constraints. We then measure the deviation in chamber count:

$$\Delta C = \left| |\text{Chambers}(\mathcal{G}_{\text{fut}})| - |\text{Chambers}(\hat{\mathcal{G}}_{\text{fut}})| \right|.$$

  A smaller ΔC indicates better preservation of macro-level social fragmentation patterns.

- **Power-law Degree Distribution.** Following GDGB (Peng et al., 2025), we assess how well the generated graph preserves the heavy-tailed nature of real-world networks. We fit a power-law distribution $p(k) \sim k^{-\alpha}$ to the degree sequence of the reference graph using maximum likelihood estimation, with $x_{\min} = 2$ for all datasets. The goodness-of-fit is evaluated via the Kolmogorov-Smirnov (KS) distance between the empirical and fitted distributions. Additionally, we compute the deviation in the estimated power-law exponent:

$$\Delta\alpha = |\alpha_{\text{ref}} - \alpha_{\text{gen}}|,$$

  where α_{ref} and α_{gen} are the exponents fitted on $\mathcal{G}_{\text{fut}}$ and $\hat{\mathcal{G}}_{\text{fut}}$, respectively. Smaller $\Delta\alpha$ indicates better reproduction of scale-free characteristics.

These metrics are normalized and averaged into the phenomenon replication score:

$$S_{\text{phenomenon}} = \frac{1}{3}(\overline{\text{P@100-KOL}} + (1 - \overline{\Delta C}) + (1 - \overline{\Delta\alpha})).$$

IDGG Social Fidelity Score. The final score is computed as:

$$S_{\text{IDGG}} = 0.5 \cdot S_{\text{structure}} + 0.5 \cdot S_{\text{phenomenon}},$$

with equal weighting reflecting balanced importance between macro structure alignment and macro phenomenon replication.

B.3 Degree Prediction Metrics

To evaluate the Activity-Predictor in source node degree prediction, we adopt three distributional discrepancy metrics based on histogram comparisons between predicted and ground-truth out-degree distributions. Let $d(v)$ denote the true out-degree of node v, $\hat{d}(v)$ its predicted value, and V the set of all nodes. We first construct empirical histograms over predefined bins:

$$\hat{p}_i = \frac{|\{v \in V : d(v) \in \text{bin}_i\}|}{|V|},$$

$$\hat{q}_i = \frac{\left|\{v \in V : \hat{d}(v) \in \text{bin}_i\}\right|}{|V|},$$

where $\hat{p}$ and $\hat{q}$ represent the normalized frequency distributions of true and predicted degrees, respectively. We then compute the following metrics:

- **Wasserstein Distance.** We measure the Wasserstein distance between the cumulative distribution functions (CDFs) derived from $\hat{p}$ and $\hat{q}$. The CDFs are defined as:

$$\hat{P}_i = \sum_{j=1}^{i} \hat{p}_j, \quad \hat{Q}_i = \sum_{j=1}^{i} \hat{q}_j.$$

  The Wasserstein distance is then given by:

$$W = \frac{1}{n} \sum_{i=1}^{n} \left| \hat{P}_i - \hat{Q}_i \right|,$$

  where n is the number of bins. The Wasserstein distance quantifies the minimum total cost required to transform the distribution $\hat{p}$ into $\hat{q}$, interpreting bin differences as transportation distances. It is sensitive to shifts in distributional location and shape, making it well-suited for comparing degree distributions.

- **KL-Divergence.** We measure the relative entropy from $\hat{q}$ to $\hat{p}$ using:

$$D_{\mathrm{KL}}(\hat{p} \parallel \hat{q}) = \sum_{i=1}^{n} \hat{p}_i \log \left(\frac{\hat{p}_i + \epsilon}{\hat{q}_i + \epsilon} \right),$$

  where $\epsilon = 10^{-10}$ is a small constant added to prevent numerical instability due to zero probabilities. KL divergence quantifies how much information is lost when $\hat{q}$ is used to approximate $\hat{p}$, with lower values indicating better alignment.

- **MMD.OD.** We measure the distribution discrepancy between predicted and true out-degree distributions using the Maximum Mean Discrepancy (MMD) with a Gaussian RBF kernel. While we report MMD^2 in the main IDGG task, for degree prediction we take the square root to obtain a more interpretable scale:

$$\mathrm{MMD.OD} = \sqrt{\mathrm{MMD}^2(\hat{p}, \hat{q})}.$$

  This ensures that MMD.OD has the same units as node degrees, providing an intuitive estimate of distributional divergence in terms of average activity level mismatch.

C Implementation Details

C.1 Implementation of Graphia

Destination Selection. For each source node u at time t, the goal is to select $K_2 = \mathrm{round}\big(\hat{d}_t(u)\big)$ destination nodes based on a query and behavior filter generated by Graphia-Q. The Graphia-Q generates a textual query to constrain the search space. First, we sample fixed constant number K_1 of candidate destination nodes in two epochs.

- First, a textual query is generated to describe the desired characteristics of the target (e.g., "a user interested in fitness gear"). To reflect real-world social dynamics, the system first retrieves top-matching nodes from u's historical neighbors. The candidate node are ranked based on this query and historical neighbor nodes using cosine similarity of BERT embeddings.

- Second, if fewer than K_1 valid neighbors are available, the system expands the search to the general population of profiles. We retrieve and rank nodes from historical neighbors of u using the textual query, scoring via cosine similarity of BERT embeddings.

Then, all candidate lists are merged in order of priority: neighbors first, then filtered general nodes. Duplicates are removed while preserving ranking order, and the final list is truncated to K_2 destination nodes.

Edge Generation. We employ domain interleaved sampling with a fixed 4:1 ratio (category:message) for Propagate-En and a 1:1 ratio (category:message) for Weibo Tech and Weibo Daily. For each domain, we define a task-specific reward function while enforcing a shared output format to ensure structural consistency during generation. To train the GraphMixer as a reward model, we follow the training protocol of DTGB (Zhang et al., 2024a). The detailed training configuration is summarized in Table 6.

Training Details. Our training pipeline consists of two stages: supervised fine-tuning (SFT) followed by task-specific reinforcement learning via GRPO (Shao et al., 2024). For destination selection, the input is $\mathcal{M}_t(u) + p_u$ (interaction history and source node profile), and the target output is the ground-truth destination node profile p_v. For edge generation, the input is $\mathcal{M}_t(u, v) + p_u + p_v$, and the model is trained to generate the actual edge message $m_{u \to v}$. In the SFT stage, we perform full-parameter fine-tuning for edge generation. In the SFT stage, we fully fine-tune the edge generation model, but skip SFT stage for destination selection due to negligible gains observed in abalation experiments. In the RL stage, we optimize each task separately using GRPO with reward shaping based on domain-specific metrics. Training proceeds until convergence, with early stopping triggered by performance on a validation set. In the RL stage, we optimize each task separately using GRPO with reward shaping based on domain-specific metrics. Training proceeds until convergence, with early stopping determined by performance on a validation set. To ensure the validation set reflects a meaningful range of task difficulty, we adopt a difficulty-aware sampling strategy based on the out-degree of source nodes. Specifically, we stratify source nodes into three difficulty tiers—low (1), medium (2), and high (3)—using the 30th and 70th percentiles of the out-degree distribution. Only nodes in the low and medium difficulty tiers (levels 1 and 2) are included in the validation set. During RL optimization, we monitored the validation accuracy

Table 5: Training Hyperparameter Configuration for Graphia RL Components

Component	Step	K_1	α	Component	Step	β_{min}	β_{max}	GNN Rewarder	LLM Rewarder	Interleave Ratio (GNN:LLM)
Propagate-En										
Graphia-Q	300	$3K_2$	5	Graphia-E	100	1	5	GraphMixer	Qwen3-8B	4:1
Weibo-Tech										
Graphia-Q	50	100	1	Graphia-E	100	1	5	GraphMixer	Qwen3-8B	1:1
Weibo-Daily										
Graphia-Q	100	1000	5	Graphia-E	100	1	5	GraphMixer	Qwen3-8B	1:1

Table 6: Training Hyperparameter Configuration for GraphMixer

Parameter Type	Configuration
Model Architecture	
Number of GNN Layers	2
Dropout Rate	0.1
Sampling Strategy	
Number of Neighbors	20
Sampling Method	Recent
Training Parameters	
Batch Size	2048
Patience	5

and applied early stopping once the performance plateaued. Detailed training hyperparameters for Graphia-Q and Graphia-E are summarized in Table 5.

C.2 Implementation of Baselines

To systematically evaluate the role of graph-structured inputs in enhancing LLM-based social simulation, we introduce a *sequentialized* data format. In sequentialized data, all higher-order graph structures (e.g., multi-hop neighborhoods, global topology) are removed; each node's context is represented solely as a flat sequence of its one-hop neighbors. This allows us to isolate the contribution of explicit graph modeling by comparing performance between Graphia and its sequential variant Graphia-seq.

We design two evaluation tasks targeting different levels of social dynamics:

- **TDGG-task.** Evaluates the fidelity of local agent behaviors generated by LLMs.
- **IDGG-task.** Assesses system-level accuracy in predicting future social network evolution over time.

Baselines for TDGG. In the TDGG task, we examine both model scale and training paradigm. Evaluated models include:

- Qwen3-8B, Qwen3-32B (Yang et al., 2025)
- DeepSeek-R1-Distill-Qwen-32B (Guo et al., 2025)
- Llama-3.1-70B-Instruct (Dubey et al., 2024)

We additionally include a supervised fine-tuned version of Qwen3-8B (denoted Qwen3-SFT) to analyze the effect of direct behavioral cloning without reinforcement learning.

To study the impact of input data structure, we train Graphia-seq on the sequentialized dataset. The model follows the same architecture and training procedure as Graphia with sequential data. For reward design, we adopt established approaches from prior work on sequential data: destination selection is guided by a reward function adapted from LIKR (Sakurai et al., 2025), and edge generation is guided by a reward function adapted from Sotopia (Zhou et al., 2024).

Baselines for IDGG. In the IDGG task, we benchmark against a range of representative social simulators spanning different modeling paradigms. This setup allows for a comparative analysis of approaches, from purely neural models to those incorporating LLM-based components:

- **Deep Learning-Based Graph Generators.** These models are specifically designed to capture the evolution of temporal graph structures and serve as strong non-LLM baselines. (i) DGGen (Hosseini et al., 2025): A temporal GNN-based model for step-ahead graph prediction. (ii) TIGGER (Gupta et al., 2022): Uses probabilistic rules learned from historical interactions. (iii) DGGen variants: we further note that DGGen uses

Table 7: Metrics for the destination selection task. The best and second-best results are highlighted in **bold** and underline, respectively.

Dataset	Model	Easy		Hard		All	
		H@100	R@100	H@100	R@100	H@100	R@100
Propagate-En	Qwen3-8b	0.7450	0.4451	0.3275	0.3275	0.4550	0.3634
	Qwen3-8B-SFT	0.7828	0.4601	0.3275	0.3275	0.4655	0.3677
	Qwen3-32B	0.7606	0.4444	0.3415	0.3415	0.4648	0.3718
	DeepSeek-Q-32B	0.7667	0.4617	0.3125	0.3125	0.4515	0.3582
	Llama3.1-70B	0.7513	0.4418	**0.3437**	**0.3437**	0.4676	0.3735
	Graphia-seq	0.7449	0.4439	0.3297	0.3297	0.4547	0.3641
	Graphia	**0.7910**	**0.4763**	0.3319	0.3319	**0.4726**	**0.3761**
Weibo Tech	Qwen3-8b	0.5971	0 0.2602	0.2301	0.2301	0.4188	0.2455
	Qwen3-8B-SFT	0.5592	0.2422	0.2250	0.2250	0.3878	0.2334
	Qwen3-32B	0.6152	0.2641	0.2346	0.2346	0.4303	0.2498
	DeepSeek-Q-32B	**0.6246**	**0.2767**	0.2235	0.2235	**0.4372**	0.2518
	Llama3.1-70B	0.6025	0.2607	0.2283	0.2283	0.4186	0.2448
	Graphia-seq	0.6051	0.2629	0.2232	0.2232	0.4212	0.2438
	Graphia	0.6152	0.2700	**0.2364**	**0.2364**	0.4329	**0.2538**
Weibo Daily	Qwen3-8b	0.5800	0.3326	0.3030	0.3030	0.4011	0.3135
	Qwen3-8B-SFT	0.5453	0.3096	0.3060	0.3060	0.3850	0.3072
	Qwen3-32B	0.5827	0.3344	**0.3159**	**0.3159**	**0.4123**	**0.3226**
	DeepSeek-Q-32B	**0.5918**	**0.3401**	0.2769	0.2769	0.3906	0.2997
	Llama3.1-70B	0.5747	0.3259	0.3127	0.3127	0.4044	0.3173
	Graphia-seq	0.5792	0.3325	0.3042	0.3042	0.4012	0.3142
	Graphia	0.5800	0.3379	0.3042	0.3042	0.4028	0.3162

TGN (Rossi et al., 2020) as its backbone to learn temporal graph embeddings. Following the DGGen framework, we replace the backbone with more recent temporal graph learning models (Zhang et al., 2024a), yielding DGGen variants with GraphMixer (Cong et al., 2023), CAWN (Wang et al., 2021), and Dygformer (Yu et al., 2023) backbones.

- **Hybrid LLM-based Social Simulators.** We implement a baseline based on SA-Graph (Zhang et al., 2025), which predicts daily activity levels by fitting Gaussian distributions to historical node activity. The predicted number of active edges determines how many source nodes are sampled per day. Given active source nodes, we use two LLMs to generate interaction: Qwen3-SFT and Graphia-seq.

- **Pure LLM-Based Simulators:** We also compare with GAG-General (Peng et al., 2025), a recent LLM-based system designed for general-purpose graph generation. It uses prompt engineering and ReAct to simulate agent decisions and network growth. We set the seed graph length to 10000 edges, (closet to the prediction time), We implement GAG-General based on Llama3-8B [†] and Qwen3-8B [‡] backbone.

We train all baseline models on the training split and generate future graphs matching the temporal extent of the test split. Since GAG-General only provides graph generation pipelines for Weibo Tech and Weibo Daily, we do not report its performance on the Propagate-En dataset.

D Supplementary Experiments

D.1 TDGG Experiments

Destination Selection. As shown in Table 7, we report Hit@100 (H@100) and Recall@100 (R@100) for destination selection on three datasets: Propagate-En, Weibo Daily, and Weibo Tech.

On Propagate-En, Graphia achieves the highest performance on the full test set with H@100 = 0.7910 and R@100 = 0.4763, outperforming both the base Qwen3-8B model and its SFT variant. It also surpasses larger models such as Qwen3-32B and Llama3.1-70B, suggesting that the RL-based training contributes to improved prediction accuracy. Gains are modest on the hard subset but more evident in the easy and overall settings. On Weibo Daily, Qwen3-32B (H@100 = 0.5827) and DeepSeek-Q-32B (H@100 = 0.5918) achieve the best Hit@100 scores. Graphia performs comparably in H@100 (0.5800) and attains

[†]https://huggingface.co/meta-llama/Meta-Llama-3-8B

[‡]https://huggingface.co/Qwen/Qwen3-8B

Table 8: LLM-as-a-judge for the edge generation task, we adopt Qwen2-72B as the evaluation LLM. The best and second-best results are highlighted in **bold** and underline, respectively.

Dataset	Model	GF	CF	PD	DA	IQ	CR	Average
Propagate-En	Qwen3	1.9106	1.8655	1.8322	1.8541	1.8708	1.8708	1.8674
	Qwen-3-SFT	2.7647	2.7647	3.8235	2.7647	2.7647	**2.8824**	2.9608
	Qwen3-8B+TGN	2.6167	2.6180	3.5817	2.5948	2.6180	2.6211	2.7751
	Qwen3-8B+DyGFormer	2.6307	2.6303	3.5738	2.6124	2.6290	2.6321	2.7847
	Qwen3-8B+GraphMixer	2.7665	2.7678	3.8445	2.7464	2.7678	2.7700	2.9439
	Graphia	**2.7877**	**2.7877**	**3.9147**	**2.7817**	**2.7857**	2.7996	**2.9762**
Weibo Tech	Qwen3	2.4308	2.4320	2.4192	2.3941	2.4267	2.3806	2.4139
	Qwen3-8B-SFT	2.7564	2.7597	2.7401	2.7379	2.7538	2.7223	2.7450
	Qwen3-8B+TGN	2.3660	2.3669	2.3559	2.3576	2.3631	2.3504	2.3600
	Qwen3-8B+DyGFormer	2.3697	2.3707	2.3626	2.3604	2.3684	2.3550	2.3645
	Qwen3-8B+GraphMixer	2.3110	2.3116	2.3062	2.3029	2.3094	2.2945	2.3059
	Graphia	**3.3692**	**3.3705**	**3.3449**	**3.3478**	**3.3666**	**3.3343**	**3.3555**
Weibo Daily	Qwen3	2.4834	2.4859	2.4820	2.4541	2.4800	2.4453	2.4718
	Qwen3-8B-SFT	2.5405	2.5427	2.5157	2.5226	2.5382	2.5114	2.5285
	Qwen3-8B+TGN	2.4066	2.4083	2.3925	2.4012	2.4058	2.3983	2.4021
	Qwen3-8B+DyGFormer	2.4560	2.4571	2.4389	2.4489	2.4543	2.4466	2.4503
	Qwen3-8B+GraphMixer	2.4192	2.4207	2.4044	2.4123	2.4178	2.4107	2.4142
	Graphia	**4.2161**	**4.2177**	**4.1638**	**4.1799**	**4.2120**	**4.1711**	**4.1934**

the highest R@100 (0.3379), indicating slightly better coverage of true destinations despite similar ranking performance. On Weibo Tech, Graphia matches the top H@100 score (0.6152, shared with Qwen3-32B), achieves the highest R@100 (0.2700), and performs best on the hard subset (H@100 = 0.2364), suggesting effectiveness in identifying interaction partners.

The Graphia-seq variant, which is trained using reinforcement learning on sequential data without incorporating structural feedback, performs similarly to the SFT baseline. The superior performance of Graphia underscores that incorporating graph-structured data can effectively boost LLMs' ability to select appropriate interaction partners.

Edge Generation. Structural prediction assesses whether the generated graph structure aligns with the reference graph, but textual content is the core focus in our evaluation. To strengthen the evaluation of Graphia's ability to generate interaction content, we introduce a new baseline: DGNN+LLM. Specifically, we leverage the state-of-the-art dynamic graph neural networks (DGNN) from DTGB (Zhang et al., 2024a): TGN (Rossi et al., 2020), DyGFormer (Yu et al., 2023), GraphMixer (Cong et al., 2023). The DGNNs are used to predict edge types, and LLMs are used to generate the edge content for predicted edge type.

Table 8 presents LLM-as-a-judge scores across six dimensions: Goal Fulfillment (GF) from SOTOPIA (Zhou et al., 2024), Contextual Fidelity (CF), Personality Depth (PD), Dynamic Adaptability (DA), Immersive Quality (IQ), and Content Richness (CR) from GDGB (Peng et al., 2025) for edge message generation on Propagate-En, Weibo Daily, and Weibo Tech.

First, the results suggest that edge generation is a relatively accessible task compared to destination selection. Even the base Qwen3-8B model achieves moderate performance, with average scores ranging from 2.41 to 2.47 on the Weibo datasets and 1.87 on Propagate-En. Moreover, DGNN provides some auxiliary benefit to the LLM: on the Propagate-EN dataset, DGNN+LLM achieves text quality comparable to the SFT-finetuned model, but fails to improve performance on Weibo-Daily and Weibo-Tech. In comparison, Graphia achieves the best performance on all three datasets.These values are substantially higher than typical baseline performance in node retrieval tasks, indicating that generating plausible interaction text benefits heavily from pre-trained language priors.

Second, despite the low barrier to entry for LLM-based agents, structured training plays a critical role in performance. The Graphia-seq variant, which uses only sequential interaction data without explicit topological modeling, performs worse than the SFT baseline in most cases. In contrast, Graphia, which incorporates both graph-structural context and structured reward modeling during reinforcement learning, achieves significant improvements, with an average score improvement of 1.66 over SFT on Weibo Daily and 0.61 on Weibo Tech. These gains are consistent across all evaluation

dimensions, demonstrating that grounding agent behavior in structural dynamics leads to better contextual understanding and more socially coherent interactions.

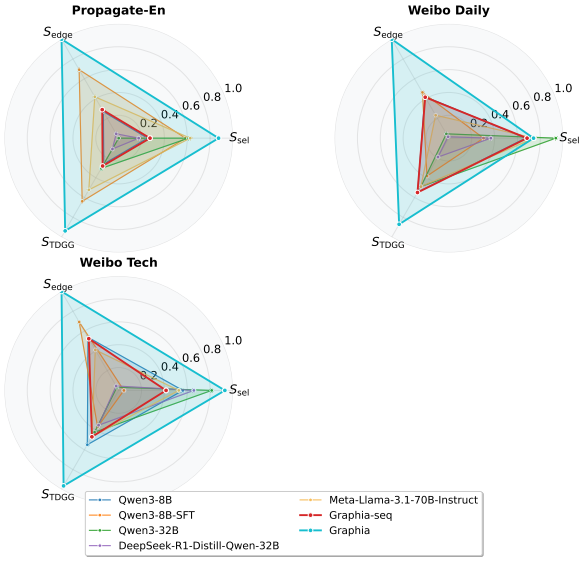


Figure 4: The social fidelity score for the TDGG task on three social network datasets.

Overall Comparison. As shown in Figure 4, we report the destination selection score (S_{sel}), edge generation score (S_{edge}), and overall TDGG performance (S_{TDGG}) across multiple models on three datasets. Compared to its base model Qwen3-8B, Graphia achieves substantial improvements in edge generation through reinforcement learning on social graph data, while yielding more modest gains in destination selection. This suggests that generating semantically coherent edges is more readily optimized via reward-guided training than accurately predicting high-level node destinations.

When comparing all models, the ranking in S_{sel} is approximately: Graphia, Qwen3-32B, Llama-3.1-70B-Instruct, DeepSeek-R1-Distill-Qwen-32B, Qwen3-8B, and Qwen3-8B-SFT. Larger-parameter models generally outperform smaller ones, indicating that destination selection benefits from increased model capacity and world knowledge. In contrast, the ranking for S_{edge} follows a different pattern: Graphia, Qwen3-8B-SFT, Qwen3-8B, Graphia-seq, Llama-3.1-70B-Instruct, DeepSeek-R1-Distill-Qwen-32B, and Qwen3-32B. Notably, fine-tuned smaller models outperform even the largest LLMs, and Graphia significantly surpasses all baselines. This highlights the effectiveness of RL-based alignment with structural feedback in improving edge content quality beyond what scale or supervised fine-tuning alone can achieve.

These results indicate that while edge generation can be effectively enhanced through targeted reinforcement learning, destination selection remains a more challenging task that demands deeper reasoning over long-term human behavior. This finding aligns with recent observations in PersonaEval (Zhou et al., 2025).

D.2 Ablation Experiment on Filter

As shown in Table 9, applying a post-hoc filter to Graphia's generated edges leads to a measurable improvement in destination selection performance. The gain primarily comes from enforcing that nodes sharing many common neighbors are more likely to be connected, which is a structural property commonly observed in real-world graphs.

D.3 Ablation Experiment on Reward

To evaluate the effectiveness of our reward design, we conduct ablation studies on the components of the reward function. For Graphia(w/o cat), we remove the category prediction reward r_{cat} during training and use only the text generation reward r_{text}. For Graphia(w/o GNN), we retain the dual-domain, domain-interleaved sampling strategy with a fixed 1:1 ratio (category:message), but remove the GNN-based structural reward component. Specifically, we modify r_{cat} to:

$$r_{\text{cat}} = \mathbb{I}(\hat{y}_{u \to v} = y^*) + r_{\text{format}},$$

where the indicator term rewards correct category prediction and r_{format} ensures output formatting correctness, without incorporating graph-level consistency signals from the GNN. We train the variants using the same number of steps as Graphia-E (100 steps).

As shown in Table 10, for the Weibo-Tech dataset, compared to the base models Qwen3-8B and Qwen3-8B-SFT, we observe that supervised fine-tuning alone improves category prediction accuracy by approximately 10%. However, Graphia(w/o cat) achieves only a marginal gain of 0.5% over Qwen3-8B-SFT; Graphia(w/o GNN) achieves a 4.65% increase in accuracy over Qwen3-8B-SFT. As shown in Table 11, for the Weibo-Daily dataset, a similar trend is observed. Qwen3-8B-SFT improves accuracy by nearly 9% over Qwen3-8B. Graphia(w/o cat) yields a modest 1.85% improvement over Qwen3-8B-SFT, Graphia(w/o GNN) attains a 24.63% increase

in accuracy over Qwen3-8B-SFT; Graphia still achieves the best performance with 24.98% over Qwen3-8B-SFT. This demonstrate that optimizing r_{cat} during reinforcement learning provides measurable benefits. Nevertheless, Graphia which incorporates GNN-as-reward, delivers a substantial 8.96% improvement in category prediction accuracy, highlighting the critical role of graph-structural feedback in aligning agent behavior with ground-truth interaction patterns.

Table 9: Ablation study on the effect of applying the filtering mechanism during evaluation for the destination selection task.

	H@100-ALL		R@100-ALL	
	w/ filter	wo. filter	w/ filter	wo. filter
Propagate-En				
Qwen3-8B	0.455	0.127	0.363	0.077
Graphia	0.473	0.253	0.376	0.178
Weibo Tech				
Qwen3-8B	0.419	0.079	0.246	0.021
Graphia	0.433	0.079	0.254	0.010
Weibo Daily				
Qwen3-8B	0.401	0.078	0.314	0.045
Graphia	0.403	0.035	0.316	0.012

D.4 Ablation on Evaluation LLMs

To mitigate potential bias arising from using a reward model (Qwen3-8B) that may favor outputs in its own linguistic style—thereby introducing unfairness in evaluation, we conduct an ablation study on the impact of different LLM-as-a-judge models on the final assessment. Specifically, we select four distinct large language models as judges: Llama-3.1-70B, Llama-3.3-70B, Qwen3-32B, and Qwen2-72B. These models independently evaluate the edge generation task. The results are summarized in Table 12.

As shown in the table, while the absolute scores vary across different judges, the overall performance trend remains consistent: GRAPHIA consistently outperforms both Qwen3 and Qwen3-SFT across the majority of settings. This demonstrates that GRAPHIA's advantage in edge generation is robust and not dependent on a particular evaluation model. Notably, we observe that Llama-3.1-70B frequently produces malformed outputs during evaluation, leading to lower and less reliable scores. This highlights the importance of selecting stable and well-behaved models when deploying LLM-as-a-judge protocols.

Table 10: Performance comparison of Graphia variants on the Weibo Tech dataset for the edge generation task. The best and second-best results are highlighted in **bold** and underline, respectively.

Model	ACC ↑	ROUGE-L ↑	BERTScore ↑
Qwen3-8B	0.6326	0.6014	0.1757
Qwen3-8B-SFT	0.7325	0.5985	0.2128
Graphia(w/o cat)	0.7362	0.6012	0.2111
Graphia(w/o GNN)	0.7790	0.6018	0.2737
Graphia	**0.8221**	**0.6040**	**0.2963**

Table 11: Performance comparison of Graphia variants on the Weibo Daily dataset for the edge generation task. The best and second-best results are highlighted in **bold** and underline, respectively.

Model	ACC ↑	ROUGE-L ↑	BERTScore ↑
Qwen3-8B	0.5456	0.5661	0.1243
Qwen3-8B-SFT	0.6338	0.5755	0.0735
Graphia(w/o cat)	0.6523	0.5723	0.0260
Graphia(w/o GNN)	0.8801	0.5901	0.1594
Graphia	**0.8836**	**0.6088**	**0.2652**

D.5 IDGG Experiments

Activity Prediction. As shown in Table 13, we evaluate the accuracy of source node out-degree prediction by binning degree values and computing distributional distances between the generated and reference graphs. The metrics of Wasserstein distance, KL-divergence, and MMD are all lower-is-better (↓), indicating how closely the predicted degree distribution matches the ground truth.

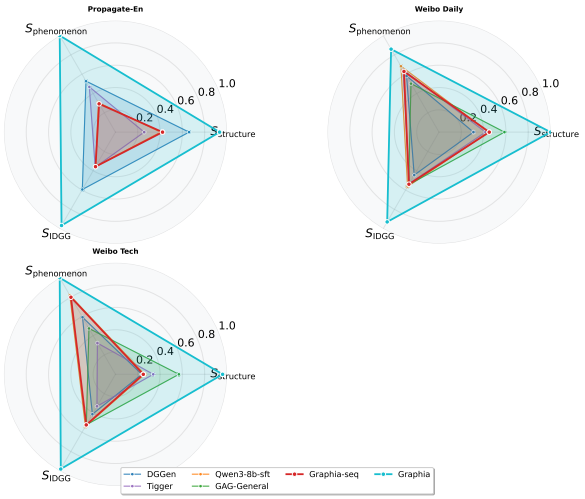


Figure 5: The social fidelity score for the IDGG task on three social network datasets.

The results show that Graphia's activity predictor significantly outperforms DGGen and GAG-General, which rely on random sampling strategies

Table 12: Ablation study on different LLMs-as-a-judge. We report the scores on six dimensions: Goal Fulfillment (GF), Contextual Fidelity (CF), Personality Depth (PD), Dynamic Adaptability (DA), Immersive Quality (IQ), and Content Richness (CR). The best and second-best results are highlighted in **bold** and underline, respectively.

LLM-as-a-judge	Model	GF	CF	PD	DA	IQ	CR	Average
Propagate-En								
Llama31-70B	Qwen3	1.6216	1.5747	1.4630	1.3049	1.5164	1.4520	1.4888
	Qwen3-SFT	2.0280	**1.9711**	**1.7823**	**1.5852**	**1.8747**	**1.8046**	**1.8410**
	Graphia	**2.0368**	1.9483	1.7407	1.5331	1.8625	1.7613	1.8138
Llama33-70B	Qwen3	1.5103	1.4700	1.3631	**1.1993**	**1.1756**	**1.0166**	1.2892
	Qwen3-SFT	1.6364	1.5629	1.4503	1.1253	0.9912	0.7718	1.2563
	Graphia	**1.6575**	**1.6001**	**1.4792**	1.1848	1.0320	0.7819	**1.2892**
Qwen3-32B	Qwen3	1.8647	1.8287	1.5090	1.5545	1.6404	1.5957	1.6655
	Qwen3-SFT	**2.2698**	2.3031	1.7367	**1.8760**	1.9242	**1.9474**	2.0096
	Graphia	2.2523	**2.3149**	**1.7429**	1.8585	**1.9439**	1.9461	**2.0098**
Qwen2-72B	Qwen3	1.9106	1.8655	1.8322	1.8541	1.8708	1.8708	1.8674
	Qwen3-SFT	2.7647	2.7647	3.8235	2.7647	2.7647	**2.8824**	2.9608
	Graphia	**2.7877**	**2.7877**	**3.9147**	**2.7817**	**2.7857**	2.7996	**2.9762**
Weibo Tech								
Llama31-70B	Qwen3	1.7852	1.6641	1.5688	1.3179	1.6105	1.4545	1.5668
	Qwen3-SFT	1.9383	1.8265	1.7613	1.5155	1.7844	1.6821	1.7514
	Graphia	**2.1057**	**1.9651**	**1.9069**	**1.6036**	**1.9119**	**1.8324**	**1.8876**
Llama33-70B	Qwen3	2.2075	2.0964	1.8625	1.6595	1.9860	1.7091	1.9202
	Qwen3-SFT	2.1691	2.0832	1.9698	1.8241	1.9356	1.7682	1.9583
	Graphia	**2.4125**	**2.3022**	**2.1855**	**2.0081**	**2.1365**	**1.9416**	**2.1644**
Qwen3-32B	Qwen3	2.3356	2.3640	1.9314	2.0303	2.1692	1.8684	2.1165
	Qwen3-SFT	2.7140	2.7404	2.3013	2.4148	2.5553	2.2794	2.5009
	Graphia	**3.2165**	**3.2577**	**2.7452**	**2.8738**	**3.0502**	**2.7412**	**2.9808**
Qwen2-72B	Qwen3	2.4308	2.4320	2.4192	2.3941	2.4267	2.3806	2.4139
	Qwen3-SFT	2.7564	2.7597	2.7401	2.7379	2.7538	2.7223	2.7450
	Graphia	**3.3692**	**3.3705**	**3.3449**	**3.3478**	**3.3666**	**3.3343**	**3.3555**
Weibo Daily								
Llama31-70B	Qwen3	1.9201	1.7534	1.6956	1.3937	1.6998	1.5490	1.6686
	Qwen3-SFT	2.1431	1.9504	1.9060	1.5975	1.9067	1.7533	1.8762
	Graphia	**3.1950**	**2.8984**	**2.7752**	**2.2418**	**2.8354**	**2.6460**	**2.7653**
Llama33-70B	Qwen3	2.7762	2.6386	2.4059	2.1984	2.4679	2.2207	2.4513
	Qwen3-SFT	2.3057	2.2256	2.1454	1.9926	2.0469	1.9035	2.1033
	Graphia	**3.6851**	**3.5156**	**3.0989**	**2.8741**	**3.2346**	**2.8684**	**3.2128**
Qwen3-32B	Qwen3	2.3332	2.3215	1.9612	2.0209	2.1872	1.9146	2.1231
	Qwen3-SFT	2.3845	2.3751	2.0603	2.1335	2.2671	2.0380	2.2097
	Graphia	**3.9917**	**4.0042**	**3.2503**	**3.4254**	**3.8050**	**3.2621**	**3.6231**
Qwen2-72B	Qwen3	2.4834	2.4859	2.4820	2.4541	2.4800	2.4453	2.4718
	Qwen3-SFT	2.5405	2.5427	2.5157	2.5226	2.5382	2.5114	2.5285
	Graphia	**4.2161**	**4.2177**	**4.1638**	**4.1799**	**4.2120**	**4.1711**	**4.1934**

Table 13: Node degree prediction metrics for different datasets. The best and second-best results are highlighted in **bold** and underline, respectively.

Dataset	Model	Wasserstein Distance ↓	KL-Divergence ↓	MMD.OD ↓
Propagate-En	DGGen	0.0152	**1.0525**	0.0016
	Tigger	0.0225	1.2222	**0.0006**
	Graphia	**0.0109**	3.2884	0.0013
Weibo Tech	GAG-General	0.6696	19.3922	0.0008
	DGGen	0.0803	6.6602	0.0040
	Tigger	0.0840	10.5437	0.0032
	Graphia	**0.0338**	**2.7597**	**0.0003**
Weibo Daily	GAG-General	0.4921	11.3634	0.0012
	DGGen	0.2023	20.4695	0.0036
	Tigger	0.0909	5.4737	0.0006
	Graphia	**0.0134**	**1.0696**	**0.0003**

for node activation. It also surpasses TIGGER, a method based on temporal point processes for modeling event timing. This performance advantage arises from Graphia's structure-aware design: the activity predictor is built upon an Informer architecture (Zhou et al., 2021) that explicitly integrates historical interaction patterns of evolving node degrees. By jointly modeling temporal dependencies and topological signals, Graphia captures more realistic user engagement dynamics, leading to more accurate activation patterns in social simulations.

Overall Comparison. We report the Macro Structure Fidelity score ($S_{\text{structure}}$), Macro Phenomenon Consistency score ($S_{\text{phenomenon}}$), and overall IDGG performance (S_{IDGG}) across multiple models on three datasets in Figure 5. We select GAG-General with Llama3 backbone for demonstration, as it outperforms the Qwen3 variant. Overall, Graphia significantly outperforms all baselines in both structural fidelity and phenomenological consistency. By integrating reinforcement learning with structural feedback, Graphia achieves the top rank in S_{IDGG} on all three datasets. This demonstrates its ability to generate dynamic graphs that simultaneously align with macroscopic topological properties and capture emergent social phenomena.

Notably, DGGen is a purely structure-driven dynamic graph generator that does not utilize textual content. It achieves the best performance among non-LLM approaches, highlighting the effectiveness of dynamic graph neural network architectures in preserving topological dynamics. In contrast, among all baselines, GAG-General and Qwen3-8B-SFT emerge as the strongest performers, surpassing traditional deep-learning-based models such as DGGen and TIGGER. This underscores the advantage of LLM-based approaches in capturing high-level interaction patterns and their potential for generating realistic social graphs.

D.6 Ablation on Graphia Components

We conduct an ablation study to investigate the influence of two critical components in the Graphia IDGG generation pipeline:

(1) whether reinforcement learning (RL) is applied during training;

(2) whether the activity predictor (AP) needs to be trained to provide prior information about source node degrees.

As shown in Table 14, we identify three key findings:

(1) Both RL and AP contribute positively to the overall quality of generated graphs. In particular, on the *edge overlap* (EO) metric, the combination of RL and AP achieves the best performance consistently across all three datasets;

(2) For small-scale graphs such as Weibo-Tech and Propagate-En, RL yields more substantial improvements, with consistent gains observed across multiple evaluation metrics;

(3) For large-scale graphs, the marginal benefit of RL diminishes. In these cases, a well-trained AP that provides reliable degree priors is sufficient to generate high-quality graphs.

Overall, these results highlight that for large-scale graph generation, training an effective AP is crucial to incorporating strong structural priors, while RL training is particularly advantageous in small-scale scenarios.

D.7 Simulation of Platform Incentives

Through TDGG and IDGG alignment experiments, we show that the discrepancy between the Graphia-generated graph $\hat{\mathbf{G}}_{\text{fut}}$ and the reference graph $\mathbf{G}_{\text{fut}}$ remains within a controllable range. Building on this, we run counterfactual, platform interventions to test whether network shifts plausibly to incentives. We inject a single broadcast into every person's memory $\mathcal{M}_t(u)$: a comment-focused incentive on Weibo Daily and a repost-focused incentive on Weibo Tech.

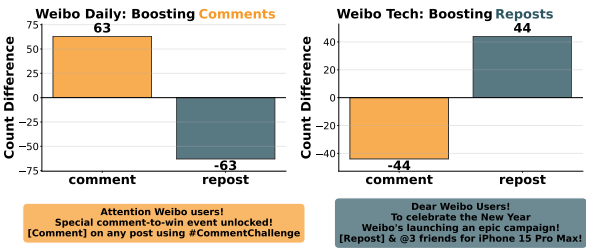


Figure 6: Impact of broadcast incentives on message propagation in the Weibo networks.

As shown in Figure 6, $\hat{\mathbf{G}}_{\text{fut}}$ build on Weibo Daily shift toward comments ($+63$) with a symmetric drop in reposts (-63), whereas $\hat{\mathbf{G}}_{\text{fut}}$ build on Weibo Tech shifts toward reposts ($+44$) with a symmetric drop in comments (-44). These results indicate that platform-level incentives can effectively steer community evolution in social graphs by reshaping the interaction patterns, which demonstrate that Graphia can support plausible counterfactual simulations.

Table 14: Ablation study on the training components of Graphia. We compare the effects of reinforcement learning (RL) and activity predictor (AP) training across three datasets. The best and second-best results are highlighted in **bold** and underline, respectively.

Model	Macro Structure				Macro Phenomenon		
	MMD.D^2 ↓	MMD.C^2 ↓	MMD.S^2 ↓	EO ↑	ΔC ↓	P@100-KOL ↑	$\Delta\alpha$ ↓
Propagate-En							
Qwen3-SFT(w/o AP)	0.3509	0.4128	0.3739	0.0608	33	0.27	1.0884
Qwen3-RL(w/o AP)	0.3594	0.301	0.3604	0.0608	33	0.32	1.1183
Qwen3-SFT(w/ AP)	0.0526	**0.2539**	0.2027	0.0882	4	**0.37**	0.0259
Qwen3-RL(w/ AP)	**0.0351**	0.3557	**0.1981**	**0.1022**	**2**	**0.37**	**0.01**
Weibo Tech							
Qwen3-SFT(w/o AP)	0.2623	1.2628	0.4772	0.0143	16	0.3	1.0828
Qwen3-RL(w/o AP)	0.2713	1.2345	0.5028	0.0137	16	0.28	1.0091
Qwen3-SFT(w/ AP)	0.1599	1.139	0.121	0.0678	9	0.31	0.3736
Qwen3-RL(w/ AP)	**0.1467**	**0.7668**	**0.1027**	**0.1347**	**8**	**0.32**	**0.123**
Weibo Daily							
Qwen3-SFT(w/o AP)	0.3234	0.8353	0.4558	0.0253	1	0.44	1.0467
Qwen3-RL(w/o AP)	0.6509	0.6874	0.1958	0.0269	**0**	0.27	1.1465
Qwen3-SFT(w/ AP)	**0.0337**	**0.4526**	**0.0314**	0.0911	**0**	**0.49**	**0.1097**
Qwen3-RL(w/ AP)	0.0614	0.4983	0.0338	**0.0973**	1	0.32	0.2074

Table 15: Runtime (hours:minutes) of Graphia components across datasets. the overall training time sumed by steps for rl training, and epochs for Activity predictor training.

Model	Dataset	Inference (h:m)	Train (h:m) 50 steps/epochs	Train (h:m) 100 steps/epochs
Graphia-Q	Propagate-En	0:16	3:03	6:06
	Weibo-Tech	0:23	6:46	13:31
	Weibo-Daily	0:28	6:14	12:28
Graphia-E	Propagate-En	0:22	4:26	8:51
	Weibo-Tech	0:39	4:17	8:33
	Weibo-Daily	1:10	8:27	16:53
Activity Predictor	Propagate-En	0:00	0:04	0:08
	Weibo-Tech	0:00	0:05	0:10
	Weibo-Daily	0:00	0:08	0:16

E Scalability of Graphia

We present two aspects of scalability analysis.

Theoretical Analysis Most deep learning-based graph generative models (GGMs) capture high-order dependencies but incur super-linear time complexity, typically $O(N^2)$ (Eigenschink et al., 2023), which limits their applicability to small-scale graphs such as molecular structures. Only a limited number of approaches achieve linear complexity with respect to the number of edges, i.e., $O(M)$ (Chen et al., 2023). Our method is inspired by EDGE (Chen et al., 2023): it first predicts node degrees using an $O(N)$ activity predictor, followed by $O(N)$ training for Graphia-Q and $O(M)$ training for Graphia-E, resulting in an overall time complexity of $O(M)$.

Empirical Comparison We benchmark the training and inference time of Graphia against GAG-General. We acknowledge that, unlike the training-free GAG-General, Graphia incurs additional training overhead. Nevertheless, we apply several engineering optimizations, including distributed training across eight H20 GPUs using the Ray framework and asynchronous inference on four GPUs via vLLM. The Activity Predictor component is trained separately on a single A800 GPU. Crucially, Graphia decouples prompt processing from token generation, thereby eliminating the prompt preprocessing latency that GAG-General incurs during inference. As shown in Tables 15, Graphia demonstrates consistently lower end-to-end generation latency across all evaluated datasets (Propagate-EN, Weibo-Tech, and Weibo-Daily). In terms of total training time, its overall cost remains comparable to that of GAG-General for large graphs like

Weibo-Daily.

F Graph Data Construction

In social graph simulation, we consider a sequence of time-stamped graph snapshots, $\{G_t\}_{t=1}^{T}$, where each $G_t = (\mathcal{V}_t, \mathcal{E}_t, \mathbf{P}_t, \mathbf{X}_t)$. Taking the Weibo social network as an example:

- $\mathcal{V}_t$ denotes the set of Weibo users at time t;
- $\mathcal{E}_t$ denotes the set of interaction edges between users at time t;
- $\mathbf{P}_t$ represents the collection of user profile texts at time t;
- $\mathbf{X}_t$ represents the collection of interaction texts (e.g., comments or reposts) between users at time t.

Example: G_t **at** $t = 3$

```
Users: [Alice, Bob, Charlie]
Profiles:
    Alice: "I love tech"
    Bob:   "Coffee lover"
    Charlie: "Student"
Edges:
    Alice → Bob
    Bob → Charlie
Interactions:
    Alice → Bob: "Check out this paper!"
    Bob → Charlie: "Hey, need help?"
```

To encode graph context for LLM input, for a source node u at time t, we construct the prompt as:

$$
\begin{aligned}
[p_u] + [\mathcal{M}_t(u)] &\rightarrow \text{LLM} \rightarrow [\text{Query}], \\
[\text{Query}] &\rightarrow \hat{C}_t^u, \\
[p_v, p_u, \mathcal{M}_t(u), \mathcal{M}_t(v)] &\rightarrow \text{LLM} \rightarrow [\hat{m}_{u\to v}, \hat{y}_{u\to v}].
\end{aligned}
$$

where p_v is the profile of candidate destination node v, $\mathcal{M}_t(u)$ and $\mathcal{M}_t(v)$ are the memory banks of source node u and destination node v respectively, containing their historical interactions up to time t. Detailed prompt templates are provided in Tables 16, 17, and 18.

G Online Resources

Our code, data, model checkpoints, and baseline implementations are publicly available at https://anonymous.4open.science/r/Graphia.

H Use of Large Language Models

LLMs are employed in two specific aspects of this work. First, we use LLMs as a writing aid to polish the manuscript text and refine figure captions, improving clarity and presentation quality. Second, our proposed framework is built upon LLM with reinforcement learning. No other parts of the research involved significant LLM assistance.

Table 16: The template of Graphia-Q for destination selection.

You should act as a src node in the network. You are given a list of dst nodes and their node texts. Your task is to predict the profile of dst nodes. You should think about the dst nodes you are going to interact with.
** Objective **
You should maximize the chances to retrieve desired dst nodes with your query text.
Your task is to depict node text of dst nodes for the src node <src id>
You're about to interact with <dx src> dst nodes in the network.
<environment description>
[For src-node (<src id>):]
<src node text>
Here's your interaction history with other destination nodes in the network:
<memory dst texts>
Here's your friends' interaction history with other destination nodes in the network:
<neighbor dst texts>

Table 17: The template of Graphia-E for edge generation.

You should generate the edge attributes for the edge (relation/action between src and dst node). You should think about the edge attribute.
** Objective **
You should first predict the edge LABEL. Then generate the edge TEXT, consistent with src node history edges.
[For src-node (<src id>):]
<src node text>
[For dst-node (<dst id>):]
<dst node text>
Src-node(<src id>) past edges:
<memory edge texts>

Table 18: The template of LLM-as-ajudge for edge generation evaluation.

You are an expert judge evaluating the quality of a response to a given prompt. Please evaluate the role-playing ability of the ACTOR NODE based on its actor actions consistency with reference actions.
The social goal for the actor is:
<goal>
[Prompt]:
<prompt>
[ACTOR Action]:
<response>
[Reference Action]:
<reference>
Scoring Logic
- GOAL Fulfillment(GF):
1 (Frequent mismatches with the goal),3 (Mostly aligned, with minor inconsistencies),5 (Fully aligned with the goal)
- Contextual Fidelity(CF):
1 (Frequent inconsistencies),3 (Minor inconsistencies),5 (Deep contextual mastery)
- Personality Depth(PD):
1 (Contradictory traits),3 (Occasional deviations),5 (Nuanced embodiment)
- Dynamic Adaptability(DA):
1 (Rigid responses),3 (Context-dependent adaptation),5 (Creative innovation)
- Immersive Quality(IQ):
1 (Disruptive inconsistencies),3 (Minor immersion breaks),5 (Seamless portrayal)
- Content Richness(CR):
1 (Superficial/output),3 (Adequate detail),5 (Rich, layered interactions)
Your response must follow the format provided below/ Please note that only when the content quality is extremely good can 5 Points be given.
[Response Format]:
GF: [1-5]
CF: [1-5]
PD: [1-5]
DA: [1-5]
IQ: [1-5]
CR: [1-5]
[Response]: