GDGB: A BENCHMARK FOR GENERATIVE DYNAMIC TEXT-ATTRIBUTED GRAPH LEARNING

Jie Peng[1] **Jiarui Ji**[1] **Runlin Lei**[1] **Zhewei Wei**[1*] **Yongchao Liu**[2*] **Chuntao Hong**[2]

[1]Renmin University of China [2]Ant Group

{peng_jie, jijiarui, runlin_lei, zhewei}@ruc.edu.cn
{yongchao.ly, chuntao.hct}@antgroup.com

ABSTRACT

Dynamic Text-Attributed Graphs (DyTAGs), which intricately integrate structural, temporal, and textual attributes, are crucial for modeling complex real-world systems. However, most existing DyTAG datasets exhibit poor textual quality, which severely limits their utility for generative DyTAG tasks requiring semantically rich inputs. Additionally, prior work mainly focuses on discriminative tasks on DyTAGs, resulting in a lack of standardized task formulations and evaluation protocols tailored for DyTAG generation. To address these critical issues, we propose Generative DyTAG Benchmark (GDGB), which comprises eight meticulously curated DyTAG datasets with high-quality textual features for both nodes and edges, overcoming limitations of prior datasets. Building on GDGB, we define two novel DyTAG generation tasks: Transductive Dynamic Graph Generation (TDGG) and Inductive Dynamic Graph Generation (IDGG). TDGG transductively generates a target DyTAG based on the given source and destination node sets, while the more challenging IDGG introduces new node generation to inductively model the dynamic expansion of real-world graph data. To enable holistic evaluation, we design multifaceted metrics that assess the structural, temporal, and textual quality of the generated DyTAGs. We further propose GAG-General, an LLM-based multi-agent generative framework tailored for reproducible and robust benchmarking of DyTAG generation. Experimental results demonstrate that GDGB enables rigorous evaluation of TDGG and IDGG, with key insights revealing the critical interplay of structural and textual features in DyTAG generation. These findings establish GDGB as a foundational resource for advancing generative DyTAG research and unlocking further practical applications in DyTAG generation. The dataset and source code are available at https://github.com/Lucas-PJ/GDGB-ALGO.

1 INTRODUCTION

Dynamic graph-structured data is ubiquitous, spanning various domains, such as social networks (Huang et al., 2022; Sun et al., 2022), recommendation systems (Zhang et al., 2022; Tang et al., 2023), and citation networks (Skarding et al., 2021; Geng et al., 2022). Notably, real-world dynamic evolution scenarios inherently contain rich (textual) features. For example, user posts and interactions such as comments and reposts on social networks, and consumer reviews of products on e-commerce platforms. This motivates the construction of dynamic text-attributed graphs (DyTAGs), which systematically integrate structural/temporal dynamics and textual attributes (Zhang et al., 2024a).

Existing representative DyTAG benchmarks, such as DTGB (Zhang et al., 2024a), demonstrate that state-of-the-art Dynamic Graph Neural Networks (DGNNs) achieve significant performance gains in discriminative tasks (e.g., link prediction, node retrieval, and edge classification) by leveraging DyTAGs' textual features. Growing interest in generative tasks spans computer vision (CV) (Wang et al., 2021b; Raut & Singh, 2024) and natural language processing (NLP) (Xu et al., 2024; Blease et al., 2024), with increasing demand for graph generation in domains such as drug discovery (Luo

*Zhewei Wei and Yongchao Liu are the corresponding authors.

et al., 2021; Martinelli, 2022), graph augmentation (Ding et al., 2022; Meng et al., 2023), and privacy-preserving frameworks (Miao et al., 2023; He et al., 2025). However, DyTAG generation remains underexplored. For instance, DTGB focuses narrowly on simplistic textual relation generation while neglecting the rich interplay of textual, structural, and temporal dynamics in graph evolution (Zhang et al., 2024a). This gap underscores the increasing criticality of generating high-quality DyTAGs.

The primary obstacle hindering progress in DyTAG generation lies in the absence of high-quality, text-rich DyTAG benchmarks. Current datasets and benchmarks face two critical limitations: ***1)*** *Lack of high-quality (textual) attributes*: Traditional dynamic graph datasets fundamentally lack node/edge features, relying solely on topological and temporal information (Poursafaei et al., 2022b). This scarcity of attributes poses substantial challenges for generative models that require rich feature inputs. Meanwhile, existing DyTAG datasets from DTGB exhibit poor textual quality, with (source) node texts typically limited to usernames or email addresses, lacking semantic richness (Zhang et al., 2024a), which severely restricts the development of DyTAG learning. ***2)*** *Lack of standardized DyTAG generative task formulations and evaluation protocols*: Current dynamic graph generative models mainly rely on structural and temporal information (Gupta et al., 2022; Dey et al., 2024), necessitating the development of new task formulations and holistic evaluation metrics tailored to DyTAG's textual characteristics. Furthermore, most approaches prioritize direct generation of the final target graph (Zeno et al., 2021; Li et al., 2024), diverging from the incremental and expansive growth patterns observed in real-world dynamic graph generation scenarios (Ji et al., 2025). Hence, to address these issues, we propose the **G**enerative **D**yTA**G** **B**enchmark (**GDGB**) from the following three aspects: 1) dataset construction, 2) task & metric definition, and 3) generative framework design.

Datasets. Our proposed GDGB comprises eight carefully selected and rigorously processed DyTAG datasets covering domains such as e-commerce recommendations, social networks, biographies of celebrities, citation networks, and movie collaboration networks. All datasets include nodes and edges endowed with rich semantic textual information. Thus, the newly proposed datasets resolve the key problems associated with previous dynamic graph datasets, such as poor quality of textual features, which subsequently results in the inability to support DyTAG generation tasks.

Tasks & Metrics. Based on our GDGB datasets, we introduce two novel DyTAG generation tasks: *Transductive Dynamic Graph Generation (TDGG)* and *Inductive Dynamic Graph Generation (IDGG)*. TDGG generates a target DyTAG based on the given source and destination node sets, maintaining the transductive assumption that all nodes are known as prior knowledge. Different from TDGG, the more challenging IDGG extends the transductive setting by introducing inductive modeling of new node generation during graph evolution, thereby successfully modeling the dynamic expansion of real-world graphs. To holistically assess DyTAG generation, we design multifaceted evaluation protocols that jointly consider topological patterns, temporal dynamics, and semantic quality, including: *1) Graph Structural Metric*, *2) Textual Quality Metric*, and *3) Graph Embedding Metric*.

Framework. Given the text-rich nature of DyTAGs, we propose GAG-General, an LLM-based multi-agent framework tailored for DyTAG generation tasks, extending the implementation of GAG (Ji et al., 2025). Unlike the bipartite-graph-specific GAG, GAG-General demonstrates superior generalization across diverse DyTAG scenarios, without requiring domain-specific customization. Furthermore, we implement both TDGG and IDGG tasks within GAG-General, integrating the proposed holistic evaluation metrics to ensure reproducible DyTAG generation and robust benchmarking.

Experimental results of GDGB demonstrate that the proposed datasets, tasks, evaluation protocols, and GAG-General framework collectively enable robust benchmarking for DyTAG generation. Key findings highlight the critical interplay between structural and textual features in DyTAG generation, as well as the practical applicability of DyTAG generation in domains such as e-commerce and social network analysis. While the GAG-General framework achieves competitive performance in both TDGG and IDGG, the generative pipeline still warrants further refinement to address challenges in both structural fidelity and attribute richness, ultimately enabling high-quality DyTAG generation. Our contributions could be summarized as follows:

- **First Generative DyTAG Benchmark.** To the best of our knowledge, GDGB is the first generative DyTAG benchmark, including eight high-quality datasets tailored for DyTAG generation.

- **Novel Generative Tasks, Metric, and Framework.** We introduce two novel DyTAG generation tasks, TDGG and IDGG, along with multifaceted metrics for holistic evaluation. Furthermore, we propose GAG-General to ensure reproducible and robust benchmarking.
- **Empirical Insights.** We find that both structural and textual features are crucial for DyTAG generation, guiding future generative framework refinement and practical applications.

2 Related Work

Discriminative Dynamic Graph Learning. Recent years have witnessed significant advances in dynamic graph learning over discrete-time dynamic graphs (DTDGs) and continuous-time dynamic graphs (CTDGs). Substantial progress has been made through various DGNNs (Rossi et al., 2020; Peng et al., 2025). These models demonstrate strong performance in discriminative tasks, including dynamic link prediction, node retrieval, etc. Concurrently, the community develops diverse benchmarks to standardize evaluation (Poursafaei et al., 2022b; Huang et al., 2024). Notable initiatives include DyGLib (Yu et al., 2023) benchmarking on classical dynamic graph datasets, TGB (Huang et al., 2024) introducing large-scale graphs for scalability challenges, and TGB-Seq (Yi et al., 2025) focusing on complex sequential patterns. However, current benchmarks primarily provide basic structural-temporal information, with limited datasets containing simplistic statistical attributes for nodes or edges (Zhang et al., 2024a). To address this, DTGB (Zhang et al., 2024a) proposes DyTAG datasets and incorporates textual features through BERT embeddings (Devlin et al., 2019), showing measurable performance gains across multiple discriminative tasks (Zhang et al., 2024a). Nevertheless, the information bottleneck caused by transformed BERT embeddings inevitably compromises original semantic richness, leaving open the challenge of effectively utilizing textual semantics for enhanced DyTAG learning.

Generative Dynamic Graph Learning. Compared to its discriminative counterpart, generative dynamic graph learning is still in its nascent stages. Early approaches primarily focus on DTDGs, including (Zeno et al., 2021; Zhou et al., 2020; Dey et al., 2024). Recent advances have shifted toward CTDG generation, enabling more refined temporal modeling that better aligns with real-world application needs (Zhang et al., 2021; Gupta et al., 2022). Contemporary approaches begin exploring feature-supportive generation: VRDAG (Li et al., 2024) implements node attribute generation through graph-based variational autoencoders (VAEs), while DG-Gen (Hosseini et al., 2024) models edge attributes via joint conditional probability distributions. More recently, GAG (Ji et al., 2025), an LLM-based multi-agent framework, enables simulation-based generation for social bipartite DyTAGs. However, this field significantly lacks standardized benchmarks for comprehensive and effective evaluation of dynamic graph generation tasks. First, in terms of task design, existing methods prioritize direct generation of the final target graph (Zeno et al., 2021; Zhou et al., 2020; Li et al., 2024), which deviates from the expansive growth patterns observed in real-world dynamic graph generation scenarios (Ji et al., 2025). Second, the definition of evaluation metrics lacks multidimensional, holistic assessments of graph structure, temporal dynamics, and attribute coherence (Zhou et al., 2020; Li et al., 2024; Ji et al., 2025). Therefore, when extending generative tasks to DyTAG generation, establishing a robust benchmark tailored for DyTAG's dynamic and text-rich characteristics becomes a critical need.

3 Proposed Datasets

To enable high-quality DyTAG generation, this task necessitates datasets where nodes and edges are accompanied by rich textual attributes and evolving structure-temporal information over time.

Table 1: Statistics of proposed DyTAG datasets of GDGB. See Section C.1 for more details.

Dataset	Nodes	Edges	Edge Labels	Timestamps	Domain	Text Attributes	Bipartite
Sephora	210,357/2,274	801,234	5	5,314	E-commerce	Node & Edge	✓
Dianping	158,541/88,118	1,990,409	5	745,151	E-commerce	Node & Edge	✓
WikiRevision	75,622/3,204	2,778,732	2	2,766,153	Web Interaction	Node & Edge	✓
WikiLife	406,148/54,513	1,996,520	24	1,810	Celebrity Biography	Node & Edge	✓
IMDB	125,714	1,534,162	20	122	Movie Collaboration	Node & Edge	✗
WeiboTech	20,767	109,345	2	79,925	Social Network	Node & Edge	✗
WeiboDaily	66,500	354,098	2	293,662	Social Network	Node & Edge	✗
Cora	48,797	110,788	5	8,274	Citation	Node & Edge	✗

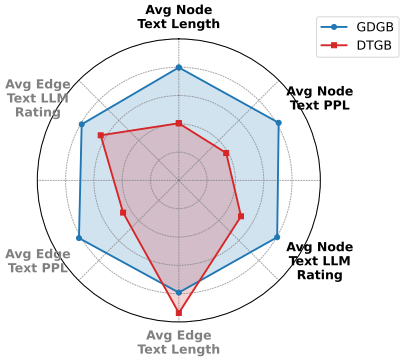

Figure 1: Comparison of node and edge texts across GDGB and DTGB datasets in terms of length, perplexity (PPL (Reversed)), and LLM-based rating.

Limitations of Existing Datasets. Existing dynamic graph datasets, though valuable for structural and temporal modeling, suffer from critical limitations in textual attributes that hinder high-quality DyTAG generation. DTGB (Zhang et al., 2024a) represents the first effort to incorporate both node and edge texts. However, as shown in Figure 1, the node and edge texts in DTGB are notably short and semantically shallow in terms of length, perplexity (PPL), and LLM-based rating. GDGB achieves significant quality advantages over DTGB in five out of six dimensions (see detailed results for each dimension in Section C.5). For instance, in six of DTGB's eight datasets, (source) node texts are typically mere identifiers such as email addresses or usernames (Figure 5, left). While the two Stack-platform datasets attempt to enrich node texts with user locations and bios, half of these fields remain empty or meaningless. Moreover, edge texts in datasets like GDELT and ICEWS1819 are excessively brief (Figure 5, right), providing insufficient context for text generation. Despite DTGB having longer average edge text length, the semantic quality of DTGB's texts—as indicated by PPL and LLM-based rating—is significantly lower than GDGB's. These limitations of current datasets impede progress in DyTAG generation research. A comprehensive comparison of additional existing datasets is provided in Section C.4.

Our Proposed High-Quality DyTAG Datasets. To overcome the aforementioned textual quality issues and enable DyTAG generation tasks, we propose Generative DyTAG Benchmark, GDGB, consisting of eight carefully curated DyTAG datasets. These datasets are sourced from diverse real-world domains, including e-commerce, social networks, biography networks, etc., encompassing both bipartite (4 datasets) and non-bipartite (4 datasets) graphs. Detailed statistics are presented in Table 1. A key focus during the collection and preprocessing of these datasets is to ensure that all source/destination nodes and interaction edges possess rich, semantic textual attributes. For example, Sephora includes: 1) user node texts detailing user appearances and review histories, 2) product node texts describing product brands, ingredients, and ratings, and 3) edge texts containing user detailed reviews. The textual completeness of GDGB provides a robust foundation for developing DyTAG generation models capable of producing both realistic structures and coherent text.

Generative Task Performance Validation. To assess the impact of textual quality on dynamic graph generation, we evaluate two latest feature-supportive models—VRDAG (Li et al., 2024) (node features) and DG-Gen (Hosseini et al., 2024) (edge features)—using their public implementations. Following DTGB's setup (Zhang et al., 2024a), we encode raw texts with BERT-base-uncased (Devlin et al., 2019) and measure structural fidelity via Degree/Spectra MMD (Xiang et al., 2022) to quantify distributional differences between generated and ground-truth graphs. Lower MMD indicates better structural quality when textual features are effective. Results in Figures 6 and 7 show that, on GDGB, incorporating node or edge texts significantly reduces MMD, improving structural similarity. In contrast, on DTGB, textual features degrade performance in half the cases—especially for VRDAG—due to poor text quality. This highlights inherent limitations in DTGB. Beyond these experiments, we further validate GDGB's advantages through applying our generative framework (proposed in Section 4.3) to downstream generation tasks, providing a comprehensive demonstration of the advantages and distinctions of GDGB over DTGB in Section C.5.

4 Generative Tasks, Metrics, and Framework

4.1 Generative Task Design

We consider a DyTAG $\mathcal{G} = (\mathcal{N}, \mathcal{E}, \mathcal{T})$, where $\mathcal{N}$ denotes the set of nodes, $\mathcal{E}$ denotes the set of edges, and $\mathcal{T}$ denotes the set of timestamps. Let $\mathcal{N}^{\text{text}}$, $\mathcal{E}^{\text{text}}$, and $\mathcal{L}$ represent the sets of node texts, edge texts, and edge labels, respectively. For any node $n \in \mathcal{N}$, its associated node text is denoted by $\mathcal{N}_n^{\text{text}}$. Each edge $(u, v) \in \mathcal{E}$ is attached with an edge text $\mathcal{E}_{u,v}^{\text{text}}$, an edge label $\mathcal{L}_{u,v}$, and an interaction timestamp $\mathcal{T}_{u,v}$. Leveraging our proposed GDGB datasets, we introduce two novel DyTAG generation tasks: Transductive Dynamic Graph Generation (TDGG) and Inductive Dynamic Graph Generation (IDGG).

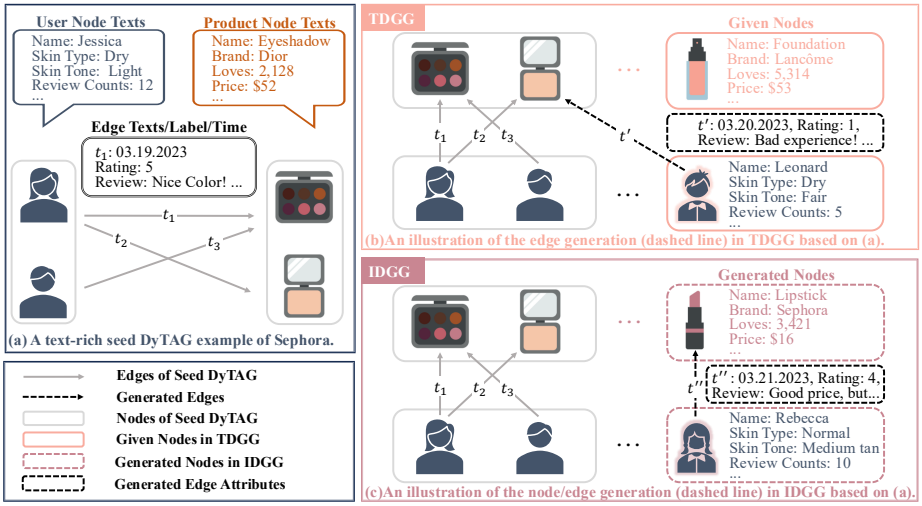

Figure 2: A case study on TDGG and IDGG in the Sephora product reviews scenario.

Transductive Dynamic Graph Generation. Starting with an initial seed DyTAG $\mathcal{G}_0$, TDGG generates a final DyTAG $\mathcal{G}_K$ based on the given source and destination node sets, maintaining the transductive assumption that all nodes are known as prior knowledge. Thus, the goal of TDGG is to conduct destination node selection and edge generation, which naturally integrates traditional discriminative tasks (e.g., node retrieval and edge classification) in a generative paradigm. The generated DyTAG $\mathcal{G}_K$ in TDGG is expected to maintain structural, temporal, and textual similarity to the ground-truth DyTAG, reflecting high-quality generation. The illustration of TDGG is shown in Figure 2 (b), demonstrating that the user will review the powder at t', along with a generated negative review and a rating of 1.

Inductive Dynamic Graph Generation. Different from TDGG, the more challenging IDGG extends the transductive setting by introducing inductive modeling of new node generation during graph evolution. Starting with an initial seed DyTAG $\mathcal{G}_0$, IDGG generates a final DyTAG $\mathcal{G}_K$ based on the dynamically evolving source and destination node sets with newly generated nodes. Due to the incorporation of inductive node generation, the generated DyTAG $\mathcal{G}_K$ in IDGG is expected to preserve the structural properties of the ground-truth DyTAG, while ensuring that all newly added nodes and edges contain high-quality, semantically coherent textual attributes, thereby successfully modeling the dynamic expansion of real-world graph data. The illustration of IDGG is shown in Figure 2 (c), showcasing that the newly generated user will review the newly generated lipstick at t'' with a generated positive review and a rating of 4.

4.2 Proposed Generative Metrics

To ensure a comprehensive assessment of graph structure, temporal dynamics, and textual quality of the generated DyTAG on TDGG and IDGG, we propose multifaceted evaluation protocols for DyTAG generation, integrating structural, textual, and graph embedding-based metrics. Detailed formulations and implementation specifics of the following metrics are provided in Section D.3.

Graph Structural Metric. We evaluate the structural validity of generated DyTAGs using two classical approaches: **1) Degree/Spectra MMD**: We utilize the maximum mean discrepancy (MMD) with a radial basis function (RBF) kernel to measure the distribution distance between generated and ground-truth graphs (e.g., degree/spectral properties)(Xiang et al., 2022; You et al., 2018; Chen et al., 2023). **2) Power-law Analysis**: We assess power-law behavior in degree distributions using the Kolmogorov-Smirnov distance D_k and the power-law exponent α (Clauset et al., 2009). Specifically, we assess the presence of power law validity by evaluating whether $D_k < 0.15$ and $\alpha \in [2, 3]$ for the generated graphs.

Textual Quality Metric. Inspired by the recent role-playing agent research (Wang et al., 2024b; Chen et al., 2024; Wang et al., 2024a), we employ the **LLM-as-Evaluator** framework to assess the text quality in the generated DyTAGs. Specifically, the evaluation framework is conducted

with five scoring criteria (*Contextual Fidelity, Personality Depth, Dynamic Adaptability, Immersive Quality, Content Richness*) tailored for DyTAG generation, scored on a 1–5 scale. Compared to embedding-based metrics used in prior DyTAG benchmarks (e.g., BERTScore (Zhang et al., 2020) in DTGB (Zhang et al., 2024a)), this evaluation framework offers two distinct advantages: 1) Unified multidimensional evaluation across diverse textual attributes, and 2) Preservation of semantic fidelity by avoiding information compression inherent in embedding representations.

Graph Embedding Metric. To enable a holistic and coupled evaluation of structural, temporal, and textual quality in DyTAG generation, we extend the JL-Metric (Hosseini et al., 2025), designed for dynamic graph generation, to a better adaptation on DyTAGs by integrating textual node/edge features, resulting in a **graph embedding-based indicator** that quantifies the generation quality across three critical dimensions: topological patterns, temporal dynamics, and node/edge textual attributes. This metric condenses DyTAGs into a unified embedding space for the pairwise similarity computation between graph embeddings. The resulting scalar score quantifies global fidelity between generated and ground-truth DyTAGs by jointly measuring structural, temporal, and textual characteristics.

4.3 Generative Framework

Limitations of Previous Methods in DyTAG Generation. The novelty of our proposed TDGG and IDGG tasks presents unique challenges, as no existing method can directly address their requirements. For instance, the latest feature-supportive dynamic graph generation models like VRDAG (Li et al., 2024) and DG-Gen (Hosseini et al., 2024) focus solely on generating node or edge representation features, but lack the capability to handle textual contents in DyTAGs. As far as we know, while GAG (Ji et al., 2025) is the only existing baseline that incorporates textual attributes, it is specifically designed for bipartite social graph generation and lacks generalizability to diverse DyTAG structures and domains. This gap motivates the development of a more universal framework tailored for DyTAG generation tasks.

GAG-General. Given the text-rich nature of DyTAGs, we propose GAG-General, an LLM-based multi-agent framework designed for DyTAG generation tasks, building upon the implementation of GAG (Ji et al., 2025). Leveraging the text understanding and generation capabilities of LLMs, our framework extends the original GAG with three key enhancements: **1) Generalization**: Support both bipartite and non-bipartite graph structures universally. **2) Multi-domain Compatibility**: Abstract the DyTAG generation pipeline to enable seamless adaptation across diverse interaction scenarios without domain-specific customization. **3) Standardization**: Establish standardized task formulations for TDGG and IDGG and incorporate holistic evaluation metrics, ensuring reproducible DyTAG generation. See more details of GAG-General in Section D.1.

Framework Details. Following GAG (Ji et al., 2025), GAG-General employs carefully designed LLM-based agents for source and destination nodes, with each node agent maintaining a memory module to record historical neighbor interactions. This memory module effectively captures structural and temporal dynamics from the DyTAG by preserving contextual information about past interactions. We further incorporate an optional memory reflection mechanism, which leverages LLMs to distill the node memories into valuable summaries, akin to the message aggregation process in GNNs (Kipf & Welling, 2016; Peng et al., 2024). For TDGG and IDGG, GAG-General implements an iterative, expansive DyTAG generation pipeline: in each iteration, the source node agent selects destination nodes based on its memory and contextual features. In the case of IDGG, new nodes are first generated and then updated to the node set before this selection step, ensuring the DyTAG's inductive expansion. Subsequently, edges are generated between the selected node pairs. This iterative process continues until the final DyTAG is produced, seamlessly integrating node/edge generation and DyTAG evolution. See more details and pseudocode regarding the implementation of TDGG and IDGG based on GAG-General in Section D.2.

5 Experiment

5.1 Experimental Settings

Baselines. 1) As our proposed GAG-General is an LLM-based multi-agent generative framework, for the DyTAG generation in TDGG and IDGG, we benchmark three open-source LLMs: DeepSeek-R1-Distill-Qwen-32B (DeepSeek-AI, 2025) (referred to as DeepSeek), Llama-3-70B-Instruct (AI@Meta,

Table 2: The results on Degree MMD, Spectra MMD, D_k, α, and power-law validity under TDGG with GPT as the LLM backbone. Full results on the other three LLM backbones are available in Table 18, 19, and 20.

Dataset	**Sephora**	**Dianping**	**WikiRevision**	**WikiLife**	**IMDB**	**WeiboTech**	**WeiboDaily**	**Cora**
Degree MMD↓	0.023	0.055	0.108	0.181	0.278	0.243	0.247	0.128
Spectra MMD↓	0.011	0.328	0.156	0.223	0.316	0.297	0.493	0.156
D_k	0.143	0.041	0.056	0.099	0.135	0.030	0.048	0.049
α	2.993	2.234	2.041	2.204	1.720	2.011	1.845	2.378
Power-law Validity	✓	✓	✓	✓	✗	✓	✗	✓

Table 3: The results on average textual quality scores under TDGG. M. and R. denote node memory and reflection mechanism, respectively. Full results on each scoring criterion are available in Table 21-29. The LLM (GPT) used in evaluation is independent of the LLM backbone used by GAG-General itself. See Section E.4 for more details. The best and the runner-up scores are highlighted in bold and underlined fonts.

	DeepSeek			**Llama**			**Qwen**			**GPT**		
	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.
Sephora	4.09	<u>4.10</u>	**4.37**	4.57	<u>4.58</u>	**4.66**	4.61	<u>4.64</u>	**4.70**	4.69	<u>4.69</u>	**4.77**
Dianping	4.29	<u>4.34</u>	**4.41**	4.13	<u>4.18</u>	**4.46**	<u>4.14</u>	4.13	**4.43**	4.32	<u>4.34</u>	**4.71**
IMDB	3.65	<u>3.82</u>	**3.99**	3.97	<u>4.02</u>	**4.32**	4.10	<u>4.18</u>	**4.33**	3.91	<u>4.02</u>	**4.44**
WeiboTech	3.88	<u>3.89</u>	**3.92**	4.49	<u>4.56</u>	**4.86**	<u>4.93</u>	4.91	**4.96**	4.84	<u>4.88</u>	**4.97**

Table 4: The results on the graph embedding metric under TDGG. See Table 30 for full results.

	DeepSeek			**Llama**			**Qwen**			**GPT**		
	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.
Sephora	<u>0.715</u>	**0.758**	0.679	<u>0.675</u>	**0.681**	0.648	0.579	<u>0.630</u>	**0.740**	0.588	**0.671**	<u>0.654</u>
Dianping	0.637	<u>0.666</u>	**0.722**	0.368	<u>0.386</u>	**0.408**	<u>0.408</u>	0.390	**0.413**	0.351	<u>0.368</u>	**0.369**
IMDB	<u>0.534</u>	**0.586**	0.532	0.396	<u>0.482</u>	**0.533**	0.487	<u>0.493</u>	**0.519**	<u>0.432</u>	0.420	**0.435**
WeiboTech	0.415	**0.577**	<u>0.501</u>	0.335	<u>0.417</u>	**0.420**	0.296	<u>0.353</u>	**0.364**	0.327	<u>0.488</u>	**0.698**

2024) (Llama), and Qwen2.5-72B-Instruct (Team, 2024) (Qwen), as well as a closed-source model: GPT-4o-Mini (OpenAI, 2024) (GPT). **2)** For TDGG, which integrates traditional discriminative tasks from DGNNs in a generative paradigm, we compare against five state-of-the-art baselines, including JODIE (Kumar et al., 2019), TGN (Rossi et al., 2020), CAWN (Wang et al., 2021a), GraphMixer (Cong et al., 2023), and DyGFormer (Yu et al., 2023). Performance is assessed on discriminative tasks such as node retrieval (link prediction) and edge classification to benchmark GAG-General against conventional DGNNs. **3)** For IDGG, due to its novelty, the challenge of new node generation, and the text-rich nature of DyTAGs, we select two of the latest dynamic graph generation models that support node/edge feature modeling and generation: DG-Gen (Hosseini et al., 2024), VRDAG (Li et al., 2024), and TIGGER-I (Gupta et al., 2022). See more details in Section E.

Implementation Details. Both TDGG and IDGG share basic settings: the first 1,000 edges and the corresponding nodes serve as the initial seed DyTAG, and we set the edge generation size per round as 50. We employ the metrics introduced in Section 4.2 for evaluating the DyTAG generation quality on TDGG and IDGG, including graph structural metrics, textual quality scores, and a graph embedding-based indicator for joint structural-temporal-textual assessment. See more implementation details in Section E.

5.2 Transductive Dynamic Graph Generation

Results of TDGG. In TDGG, GAG-General generates DyTAGs on GDGB with high structural fidelity, as shown in Table 2. The generated graphs exhibit small deviations from ground-truths—most Degree/Spectra MMDs are below 0.3, and six of eight satisfy the power-law validity criterion, indicating high-quality generation. To assess the role of structural and textual information from historical neighbors, we ablate three configurations: w/o node memory, w/ node memory, and w/ node memory & reflection mechanism. As shown in Tables 3 and 4, both node memory and the reflection mechanism significantly improve textual quality and graph embedding-based metrics across most LLM backbones, due to effective integration and aggregation of historical interaction information. Moreover, we experimentally verify the direct usability of generated graphs from TDGG in downstream tasks compared to the original graphs; experimental details are provided in Section F.5. Scalability analysis for TDGG is provided in Section F.1.

Results of Discriminative Tasks in TDGG. For node retrieval, results on Hit@1 and Hit@10 are reported in Table 36; for edge classification, in Table 39. Although existing DGNNs employ sophisticated modules to model dynamic interactions and achieve strong performance on discriminative tasks (Yu et al., 2023), their effectiveness diminishes when training on only 1,000 edges. Notably, as shown in Table 39, GAG-General outperforms DGNNs on most datasets, while gains on node retrieval are less pronounced (Table 36). This demonstrates GAG-General's superior ability to leverage structural, temporal, and textual information in DyTAGs, whereas DGNNs exhibit limited generalization due to their reliance on large-scale training data. These results further confirm GAG-General's effectiveness in modeling DyTAGs, particularly in capturing dependencies across both discriminative and generative tasks.

5.3 Inductive Dynamic Graph Generation

Results of IDGG. In IDGG, structural quality results are summarized in Table 5. Generated DyTAGs show greater deviation from ground-truths than in TDGG—due to the added complexity of new node generation—yet retain key structural properties. For example, most Degree/Spectra MMDs exceed 0.2, but five of eight graphs still satisfy the power-law criterion, indicating reasonable fidelity. Similar to TDGG, ablation studies (Tables 6 and 7) confirm that node memory and reflection enhance textual and structural quality, underscoring the importance of summarizing historical neighbor information. Moreover, Table 8 compares our framework with existing dynamic graph generation models, revealing their significant limitations: generated graphs exhibit notably lower structural fidelity and poorer node/edge attribute richness. These results highlight the need for dedicated DyTAG generation methods. Additionally, we demonstrate that generated graphs from IDGG can serve as data augmentation to enhance model performance on inductive new nodes; see details in Section F.12. Scalability analysis for IDGG is also included in Section F.1.

Hub Node Analysis in IDGG. The generated DyTAGs in IDGG demonstrate a critical evolutionary pattern: while maintaining structural congruence with ground-truth graphs, they develop highly connected hub nodes with divergent textual attributes. Specifically, for generating a DyTAG with 2,000 edges on Sephora, the top three newly generated hub nodes in the generated DyTAG include products like *Vitamin C Serum*, *Eye Cream*, and *Neck Cream*. In contrast, the ground-truth graph's top three hub nodes consist of products like *Acne Control Clarifying Cleanser*, *Rosebud Perfume*, and *Moisturizing Lotion*. The visualization of the hub node structures in the ground-truth and generated graphs is shown in Figures 10 and 11. This divergence arises because IDGG mimics real-world graph evolution dynamics, ensuring structural fidelity while enabling the creation of new nodes with reasonable attributes that align with the underlying generative patterns of the real world. For instance, in recommendation systems, the hub nodes in the generated DyTAG can represent **emerging products** with high potential for virality, while those in the ground-truth graph correspond to **established bestsellers**. This capability positions DyTAG generation as a strategic tool for proactive decision-making in e-commerce and digital marketing. By identifying potential future hubs, platforms can prioritize resource allocation for product promotion, optimize advertising strategies, and anticipate market trends before they manifest in real-world data.

6 Future Work

Generative Framework Optimization. Leveraging our high-quality, text-rich GDGB datasets, the proposed generative framework achieves groundbreaking performance in the novel DyTAG generation tasks of TDGG and IDGG. Despite this progress, key aspects of the pipeline—especially in IDGG—require further refinement. Future work should focus on improving node generation strategies

Table 5: The results on Degree MMD, Spectra MMD, D_k, α, and power-law validity under IDGG with GPT as the LLM backbone. Full results on the other three LLM backbones are available in Table 40, 41, and 42.

Dataset	**Sephora**	**Dianping**	**WikiRevision**	**WikiLife**	**IMDB**	**WeiboTech**	**WeiboDaily**	**Cora**
Degree MMD↓	0.454	0.300	0.267	0.083	0.321	0.268	0.268	0.136
Spectra MMD↓	0.189	0.453	0.229	0.208	0.423	0.201	0.439	0.244
D_k	0.147	0.069	0.045	0.099	0.252	0.064	0.156	0.091
α	2.057	2.430	2.050	2.204	1.746	1.867	1.732	2.250
Power-law Validity	✓	✓	✓	✓	✗	✗	✗	✓

Table 6: The results on average textual quality scores under IDGG. Full results are available in Table 43-51.

	DeepSeek			Llama			Qwen			GPT		
	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.
Sephora	4.63	4.73	**4.77**	4.58	4.65	**4.74**	4.58	4.68	**4.78**	4.58	4.77	**4.87**
Dianping	4.29	4.44	**4.65**	4.29	4.51	**4.87**	4.04	4.50	**4.68**	4.56	4.71	**4.86**
IMDB	4.13	4.28	**4.39**	4.22	4.31	**4.43**	4.23	4.36	**4.51**	4.19	4.29	**4.49**
WeiboTech	4.60	4.75	**4.85**	3.94	4.00	**4.75**	4.56	4.74	**4.93**	4.60	4.71	**4.93**

Table 7: The results on the graph embedding metric under IDGG. See Table 52 for full results.

	DeepSeek			Llama			Qwen			GPT		
	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.
Sephora	0.621	**0.661**	0.634	0.602	0.612	**0.647**	0.569	0.587	**0.616**	0.601	0.603	**0.628**
Dianping	0.739	0.749	**0.769**	0.426	0.457	**0.512**	0.531	0.542	**0.578**	0.521	**0.527**	0.505
IMDB	0.601	0.616	**0.729**	0.511	0.522	**0.568**	0.502	0.557	**0.588**	0.521	0.535	**0.563**
WeiboTech	**0.604**	0.591	0.562	0.601	0.629	**0.695**	0.531	0.547	**0.549**	0.501	0.514	**0.536**

Table 8: The results on the graph structural and the graph embedding metrics under IDGG of our framework and current feature-supportive dynamic graph generation models. Ours correspond to the best results of our proposed GAG-General among four LLM backbones. See Table 53 for full results.

	Sephora				Dianping				Cora			
	Ours	VRDAG	DG-Gen	TIGGER-I	Ours	VRDAG	DG-Gen	TIGGER-I	Ours	VRDAG	DG-Gen	TIGGER-I
Degree MMD ↓	**0.370**	0.795	0.422	0.622	**0.150**	0.887	0.167	0.446	**0.073**	0.877	0.212	0.372
Spectra MMD ↓	**0.189**	0.847	0.274	0.687	0.351	0.808	**0.245**	0.341	**0.181**	0.760	0.365	0.389
Power-law Validity	✓	✗	✗	✗	✓	✗	✓	✓	✓	✗	✗	✗
Graph Embedding ↑	**0.661**	0.011	0.228	0.085	**0.769**	0.024	0.517	0.580	**0.828**	0.053	0.056	0.197

and selection mechanisms, such as incorporating adaptive node sampling to balance diversity and fidelity, and leveraging existing DGNN techniques to enhance the accuracy and efficiency of candidate node retrieval and ranking. These advancements, grounded in GDGB, will strengthen DyTAG models and provide a solid foundation for generative graph foundation models, addressing challenges in both structural fidelity and attribute richness.

Further Applications. Beyond model optimization, DyTAG generation holds substantial practical value. As shown in Section 5.3, identifying generated future hub nodes—like potential hit products in e-commerce—demonstrates its utility in node-level predictive modeling. At the edge and graph levels, DyTAG generation serves as effective data augmentation tools for sparse graphs, producing synthetic graphs that preserve structure while enriching textual content. This capability extends applications to e-commerce (modeling generative recommendation), social networks (forecasting misinformation spread), and urban planning (predicting infrastructure usage). By integrating dynamic graph analysis with forward-looking textual modeling, DyTAG generation can drive impactful real-world solutions across industries.

7 Conclusion

In this work, we propose **G**enerative **D**yTA**G** **B**enchmark (**GDGB**), which resolves existing limitations in datasets, task formulations, and evaluation metrics for DyTAG generation. GDGB comprises 8 meticulously curated datasets with semantically rich node/edge attributes, enabling rigorous evaluation of DyTAG generation. By defining two novel tasks—TDGG (transductive generation with fixed node sets) and IDGG (inductive generation with new node generation)—we effectively model the dynamic expansion of real-world graphs. Additionally, we propose GAG-General, an LLM-based multi-agent framework that generalizes across diverse DyTAG structures and integrates multifaceted metrics for holistic evaluation in DyTAG generation. Experimental results demonstrate that GDGB establishes robust benchmarks for DyTAG generation, revealing the interplay between structural and textual features as a cornerstone for effective generative models. In summary, GDGB lays the groundwork for future research on advancing DyTAG generation.

Acknowledgments

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. This research was supported in part by National Natural Science Foundation of China (No. 92470128, No. U2241212), by Beijing Outstanding Young Scientist Program No.BJJWZYJH012019100020098, by the National Key Research and Development Plan of China (2023YFB4502305) and Ant Group through CCF-Ant Research Fund. 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

AI@Meta. Llama 3 model card. 2024. URL https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

Charlotte Blease, John Torous, Brian McMillan, Maria Hägglund, and Kenneth D Mandl. Generative language models and open notes: exploring the promise and limitations. *JMIR medical education*, 10:e51183, 2024.

Hongzhan Chen, Hehong Chen, Ming Yan, Wenshen Xu, Xing Gao, Weizhou Shen, Xiaojun Quan, Chenliang Li, Ji Zhang, Fei Huang, and Jingren Zhou. Socialbench: Sociality evaluation of role-playing conversational agents, 2024.

Xiaohui Chen, Jiaxing He, Xu Han, and Liping Liu. Efficient and degree-guided graph generation via discrete diffusion modeling. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 4585–4610. PMLR, 2023. URL https://proceedings.mlr.press/v202/chen23k.html.

Aaron Clauset, Cosma Rohilla Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703, 2009. doi: 10.1137/070710111. URL https://doi.org/10.1137/070710111.

Weilin Cong, Si Zhang, Jian Kang, Baichuan Yuan, Hao Wu, Xin Zhou, Hanghang Tong, and Mehrdad Mahdavi. Do we really need complicated model architectures for temporal networks? In *International conference on learning representations*, 2023.

Hanjun Dai, Azade Nazi, Yujia Li, Bo Dai, and Dale Schuurmans. Scalable deep generative modeling for sparse graphs. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2302–2312. PMLR, 2020. URL http://proceedings.mlr.press/v119/dai20b.html.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 6. long and short papers: Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2019), 2-7 June 2019, Minneapolis, Minnesota, USA*, pp. 4171–4186, Minneapolis, 2019.

Saikat Dey, Sonal Jha, and Wu-chun Feng. G2A2: graph generator with attributes and anomalies. In *Proceedings of the 21st ACM International Conference on Computing Frontiers, CF 2024, Ischia, Italy, May 7-9, 2024*. ACM, 2024. doi: 10.1145/3649153.3649206. URL https://doi.org/10.1145/3649153.3649206.

Kaize Ding, Zhe Xu, Hanghang Tong, and Huan Liu. Data augmentation for deep graph learning: A survey. *ACM SIGKDD Explorations Newsletter*, 24(2):61–77, 2022.

Faezeh Faez, Yassaman Ommi, Mahdieh Soleymani Baghshah, and Hamid R. Rabiee. Deep graph generators: A survey. *IEEE Access*, 9:106675–106702, 2021. doi: 10.1109/ACCESS.2021.3098417. URL https://doi.org/10.1109/ACCESS.2021.3098417.

Dawei Gao, Zitao Li, Xuchen Pan, Weirui Kuang, Zhijian Ma, Bingchen Qian, Fei Wei, Wenhao Zhang, Yuexiang Xie, Daoyuan Chen, Liuyi Yao, Hongyi Peng, Zeyu Zhang, Lin Zhu, Chen Cheng, Hongzhu Shi, Yaliang Li, Bolin Ding, and Jingren Zhou. Agentscope: A flexible yet robust multi-agent platform, 2024. URL https://arxiv.org/abs/2402.14034.

Hao Geng, Deqing Wang, Fuzhen Zhuang, Xuehua Ming, Chenguang Du, Ting Jiang, Haolong Guo, and Rui Liu. Modeling dynamic heterogeneous graph and node importance for future citation prediction. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 572–581, 2022.

Xiaojie Guo and Liang Zhao. A systematic survey on deep generative models for graph generation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(5):5370–5390, 2023. doi: 10.1109/TPAMI.2022.3214832. URL https://doi.org/10.1109/TPAMI.2022.3214832.

Shubham Gupta, Sahil Manchanda, Srikanta Bedathur, and Sayan Ranu. Tigger: Scalable generative modelling for temporal interaction graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(6):6819–6828, June 2022. ISSN 2159-5399. doi: 10.1609/aaai.v36i6.20638. URL http://dx.doi.org/10.1609/aaai.v36i6.20638.

Xinyu He, Dongqi Fu, Hanghang Tong, Ross Maciejewski, and Jingrui He. Temporal heterogeneous graph generation with privacy, utility, and efficiency. In *The Thirteenth International Conference on Learning Representations*, 2025.

Ryien Hosseini, Filippo Simini, Venkatram Vishwanath, and Henry Hoffmann. A deep probabilistic framework for continuous time dynamic graph generation. In *AAAI Conference on Artificial Intelligence*, 2024. URL https://api.semanticscholar.org/CorpusID:274965465.

Ryien Hosseini, Filippo Simini, Venkatram Vishwanath, Rebecca Willett, and Henry Hoffmann. Quality measures for dynamic graph generative models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=8bjspmAMBk.

Shenyang Huang, Farimah Poursafaei, Jacob Danovitch, Matthias Fey, Weihua Hu, Emanuele Rossi, Jure Leskovec, Michael Bronstein, Guillaume Rabusseau, and Reihaneh Rabbany. Temporal graph benchmark for machine learning on temporal graphs. *Advances in Neural Information Processing Systems*, 36, 2024.

Yifan Huang, Clayton Thomas Barham, Eric Page, and PK Douglas. Ttergm: Social theory-driven network simulation. In *NeurIPS 2022 Temporal Graph Learning Workshop*, 2022.

Jiarui Ji, Runlin Lei, Jialing Bi, Zhewei Wei, Xu Chen, Yankai Lin, Xuchen Pan, Yaliang Li, and Bolin Ding. Llm-based multi-agent systems are scalable graph generative models, 2025. URL https://arxiv.org/abs/2410.09824.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Srijan Kumar, Francesca Spezzano, VS Subrahmanian, and Christos Faloutsos. Edge weight prediction in weighted signed networks. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pp. 221–230. IEEE, 2016.

Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1269–1278, 2019.

Fan Li, Xiaoyang Wang, Dawei Cheng, Cong Chen, Ying Zhang, and Xuemin Lin. Efficient dynamic attributed graph generation, 2024. URL https://arxiv.org/abs/2412.08810.

Shitong Luo, Jiaqi Guan, Jianzhu Ma, and Jian Peng. A 3d generative model for structure-based drug design. *Advances in Neural Information Processing Systems*, 34:6229–6239, 2021.

Dominic D Martinelli. Generative machine learning for de novo drug discovery: A systematic review. *Computers in Biology and Medicine*, 145:105403, 2022.

Lin Meng, Hesham Mostafa, Marcel Nassar, Xiaonan Zhang, and Jiawei Zhang. Generative graph augmentation for minority class in fraud detection. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 4200–4204, 2023.

Yangyong Miao, Xiaoding Wang, and Hui Lin. A graph generation network with privacy preserving capabilities. In *International Conference on Algorithms and Architectures for Parallel Processing*, pp. 67–79. Springer, 2023.

OpenAI. Gpt-4o mini: Advancing cost-efficient intelligence. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/, 2024. Accessed: 2025-04-30.

Jie Peng, Runlin Lei, and Zhewei Wei. Beyond over-smoothing: Uncovering the trainability challenges in deep graph neural networks. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 1878–1887, 2024.

Jie Peng, Zhewei Wei, and Yuhang Ye. Tidformer: Exploiting temporal and interactive dynamics makes a great dynamic graph transformer. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 2245–2256, 2025.

Farimah Poursafaei, Shenyang Huang, Kellin Pelrine, and Reihaneh Rabbany. Towards better evaluation for dynamic link prediction. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022a. URL http://papers.nips.cc/paper_files/paper/2022/hash/d49042a5d49818711c401d34172f9900-Abstract-Datasets_and_Benchmarks.html.

Farimah Poursafaei, Shenyang Huang, Kellin Pelrine, and Reihaneh Rabbany. Towards better evaluation for dynamic link prediction. *Advances in Neural Information Processing Systems*, 35: 32928–32941, 2022b.

Gaurav Raut and Apoorv Singh. Generative ai in vision: A survey on models, metrics and applications. *arXiv preprint arXiv:2402.16369*, 2024.

Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. Temporal graph networks for deep learning on dynamic graphs. *arXiv preprint arXiv:2006.10637*, 2020.

Thomas Sauerwald and Luca Zanetti. Random walks on dynamic graphs: Mixing times, hittingtimes, and return probabilities. *arXiv preprint arXiv:1903.01342*, 2019.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

Joakim Skarding, Bogdan Gabrys, and Katarzyna Musial. Foundations and modeling of dynamic networks using dynamic graph neural networks: A survey. *IEEE Access*, 9:79143–79168, 2021.

Li Sun, Zhongbao Zhang, Feiyang Wang, Pengxin Ji, Jian Wen, Sen Su, and Philip S Yu. Aligning dynamic social networks: An optimization over dynamic graph autoencoder. *IEEE Transactions on Knowledge and Data Engineering*, 35(6):5597–5611, 2022.

Haoran Tang, Shiqing Wu, Guandong Xu, and Qing Li. Dynamic graph evolution learning for recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1589–1598, 2023.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

Lei Wang, Jianxun Lian, Yi Huang, Yanqi Dai, Haoxuan Li, Xu Chen, Xing Xie, and Ji-Rong Wen. Characterbox: Evaluating the role-playing capabilities of llms in text-based virtual worlds, 2024a. URL https://arxiv.org/abs/2412.05631.

Yanbang Wang, Yen-Yu Chang, Yunyu Liu, Jure Leskovec, and Pan Li. Inductive representation learning in temporal networks via causal anonymous walks. In *International conference on learning representations*, 2021a.

Zekun Moore Wang, Zhongyuan Peng, Haoran Que, Jiaheng Liu, Wangchunshu Zhou, Yuhan Wu, Hongcheng Guo, Ruitong Gan, Zehao Ni, Jian Yang, Man Zhang, Zhaoxiang Zhang, Wanli Ouyang, Ke Xu, Stephen W. Huang, Jie Fu, and Junran Peng. Rolellm: Benchmarking, eliciting, and enhancing role-playing abilities of large language models, 2024b. URL https://arxiv.org/abs/2310.00746.

Zhengwei Wang, Qi She, and Tomas E Ward. Generative adversarial networks in computer vision: A survey and taxonomy. *ACM Computing Surveys (CSUR)*, 54(2):1–38, 2021b.

Sheng Xiang, Dawei Cheng, Jianfu Zhang, Zhenwei Ma, Xiaoyang Wang, and Ying Zhang. Efficient learning-based community-preserving graph generation. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 1982–1994, 2022. doi: 10.1109/ICDE53745.2022.00194.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, Yang Wang, and Enhong Chen. Large language models for generative information extraction: A survey. *Frontiers of Computer Science*, 18(6):186357, 2024.

Lu Yi, Jie Peng, Yanping Zheng, Fengran Mo, Zhewei Wei, Yuhang Ye, Yue Zixuan, and Zengfeng Huang. Tgb-seq benchmark: Challenging temporal gnns with complex sequential dynamics, 2025. URL https://arxiv.org/abs/2502.02975.

Jiaxuan You, Rex Ying, Xiang Ren, William L. Hamilton, and Jure Leskovec. Graphrnn: Generating realistic graphs with deep auto-regressive models. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5694–5703. PMLR, 2018. URL http://proceedings.mlr.press/v80/you18a.html.

Le Yu, Leilei Sun, Bowen Du, and Weifeng Lv. Towards better dynamic graph learning: New architecture and unified library. *Advances in Neural Information Processing Systems*, 36:67686–67700, 2023.

Giselle Zeno, Timothy La Fond, and Jennifer Neville. Dymond: Dynamic motif-nodes network generative model. WWW '21, pp. 718–729, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383127. doi: 10.1145/3442381.3450102. URL https://doi.org/10.1145/3442381.3450102.

Jiasheng Zhang, Jialin Chen, Menglin Yang, Aosong Feng, Shuang Liang, Jie Shao, and Rex Ying. Dtgb: A comprehensive benchmark for dynamic text-attributed graphs. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 91405–91429. Curran Associates, Inc., 2024a. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/a65d054a407f94c34ecfb598fb540a0d-Paper-Datasets_and_Benchmarks_Track.pdf.

Liming Zhang, Liang Zhao, Shan Qin, Dieter Pfoser, and Chen Ling. Tg-gan: Continuous-time temporal graph deep generative models with time-validity constraints. In *Proceedings of the Web Conference 2021*, pp. 2104–2116, 2021.

Mengqi Zhang, Shu Wu, Xueli Yu, Qiang Liu, and Liang Wang. Dynamic graph neural networks for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 35(5): 4741–4753, 2022.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2020.

Xiaoqing Zhang, Xiuying Chen, Yuhan Liu, Jianzhou Wang, Zhenxing Hu, and Rui Yan. Sagraph: A large-scale text-rich social graph dataset for advertising campaigns, 2024b. URL https://arxiv.org/abs/2403.15105.

Ying Zhang, Xiaofeng Li, Zhaoyang Liu, and Haipeng Zhang. Paths of a million people: Extracting life trajectories from wikipedia, 2024c. URL https://arxiv.org/abs/2406.00032.

Yongfeng Zhang, Min Zhang, Yiqun Liu, Shaoping Ma, and Shi Feng. Localized matrix factorization for recommendation based on matrix block diagonal forms. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, pp. 1511–1520, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450320351. doi: 10.1145/2488388.2488520. URL https://doi.org/10.1145/2488388.2488520.

Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, pp. 83–92, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450322577. doi: 10.1145/2600428.2609579. URL https://doi.org/10.1145/2600428.2609579.

Dawei Zhou, Lecheng Zheng, Jiawei Han, and Jingrui He. A data-driven graph generative model for temporal interaction networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, pp. 401–411, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450379984. doi: 10.1145/3394486.3403082. URL https://doi.org/10.1145/3394486.3403082.

Appendix: Contents **15**

A Ethics Statement

Our proposed GDGB offers both potential benefits and challenges. Positively, it advances DyTAG generation research, enabling applications in e-commerce (e.g., personalized recommendations), social networks (e.g., community governance), and urban planning frameworks. These advancements could enhance data-driven decision-making while promoting DyTAG generation development through standardized evaluation protocols. Negatively, the misuse of generated DyTAGs might risk generating misleading information or amplifying biases in dynamic network scenarios (e.g., illegal user interactions). However, these risks are mitigated under proper usage and regulatory frameworks—such as data anonymization, access control, and ethical guidelines—which ensure responsible deployment in real-world applications. By adhering to these principles, GDGB's technical contributions can safely drive innovation in DyTAG learning.

B Reproducibility Statement

To ensure the reproducibility of our work, we have made the following efforts: Our proposed GDGB comprises eight carefully selected and rigorously processed DyTAG datasets, hosted through a sharing link (`https://www.kaggle.com/datasets/gdgbdataset/gdgb-a-benchmark-for-generative-dytag-learning`). The proposed GDGB benchmark and our generative framework GAG-General are fully described in this paper, including detailed dataset construction processes and model implementation specifications. All code and data preprocessing scripts are publicly available in our code repository (`https://github.com/Lucas-PJ/GDGB-ALGO`), which includes comprehensive instructions for reproducing the TDGG and IDGG experiments. The repository also provides evaluation scripts of our proposed multifaceted metrics to facilitate verification of the reported results. The GDGB website is `https://gdgb-algo.github.io/`. The website allows researchers to submit methods and track performance via leaderboards on TDGG and IDGG.

C Dataset

C.1 Dataset Details

Sephora[1] is a dataset collected from Kaggle, documenting user reviews of beauty and skincare products on the Sephora e-commerce platform. The temporal span of the dataset ranges from August 28, 2008, to March 21, 2023. Specifically, the dataset includes rich textual information about users, such as skin tone, skin type, hair color, eye color, and historical review statistics. Notably, it also contains detailed textual features of beauty products from the Sephora online store, including product and brand names, prices, ingredients, ratings, and all associated attributes, which support the construction of textual features for nodes in this DyTAG. For user reviews of beauty and skincare products, the dataset provides review ratings (ranging from 1 to 5) as edge labels, and detailed textual reviews as edge content. Consequently, the Sephora dataset is represented as a bipartite DyTAG, where users and beauty products serve as nodes with textual features, and an edge represents a user's rating and textual review of a product at a given time.

Dianping[2] (Zhang et al., 2013; 2014) is a business review dataset derived from Dianping, a prominent platform for business recommendations (e.g., restaurants), spanning from July 3, 2009, to February 8, 2012. The dataset records Dianping users' historical review statistics, frequently reviewed cities, and detailed business information, including store names, addresses, cuisine styles, average costs, and historical scores. User reviews of businesses include multi-dimensional ratings (e.g., flavor, environment, service) in addition to overall scores (ranging from 0 to 5) as edge labels. Reviews also contain rich textual content and supplementary evaluations for specific items such as recommended dishes or special services, providing extensive edge textual features. The dataset is structured as a bipartite DyTAG, where Dianping users and businesses are nodes with textual features, and an edge represents a user's detailed textual review of a business at a given time.

[1] `https://www.kaggle.com/datasets/nadyinky/sephora-products-and-skincare-reviews/`
[2] `http://www.dianping.com`

WikiRevision[3] is a dataset cleaned and processed from Wikipedia's official data dumps, recording user revisions and modifications to Wikipedia pages with the selected dumps corresponding to December 1, 2024. The temporal span ranges from January 30, 2001, to December 3, 2024. The dataset includes Wikipedia users' usernames, historical editing statistics, frequently edited pages, and all revised Wikipedia page titles. To enrich textual features for page nodes, we crawl the first paragraph of each corresponding Wikipedia page from Wikipedia. User revisions are categorized into two edge classes (minor vs. non-minor edits), with the accompanying revision comments serving as edge textual features. The dataset is structured as a bipartite DyTAG, where Wikipedia users and pages are nodes with textual features, and an edge represents a user's textual revision summary on a page at a given time.

WikiLife (Zhang et al., 2024c) is a dataset derived from Wikipedia's official data dumps, recording historical records of notable individuals being physically present at specific locations. The temporal span ranges from 202 to 2024. The original dataset (Zhang et al., 2024c) is extracted from English Wikipedia's biography pages as person-time-location triplets. We extend this by crawling the first paragraphs of corresponding Wikipedia pages for persons and locations as extra textual features. Additionally, we utilize 24 life trajectory activity categories (e.g., birth/death, education, career) from (Zhang et al., 2024c) as edge labels and map the original Wikipedia text of triplets to describe the specific activities of individuals at locations. The WikiLife dataset is structured as a bipartite DyTAG, where persons and locations are nodes with textual features, and an edge represents a person's textual life trajectory activity in a location at a given time.

IMDB[4] is a dataset based on IMDB's official data, documenting actor/actress collaboration networks. The temporal span ranges from 1988 to 2031 (including collaboration information of official future films like Avatar 3). The dataset includes actors' and actresses' names, birth/death years, primary professions, and extra biographical information crawled from Wikipedia to enrich node textual features. Edge categories correspond to 20 types of movie genres (e.g., comedy, drama, crime, action), with edge texts comprising the title of the collaborated movie and roles played by actors/actresses, respectively. The IMDB dataset is a non-bipartite DyTAG, where actors/actresses are nodes with textual features, and an edge represents a movie collaboration relationship between them at a given time.

WeiboTech(Zhang et al., 2024b) is a dataset collected from the Weibo social platform, recording user interactions (comments and reposts). The temporal span ranges from December 29, 2023, to January 5, 2024. The original dataset (Zhang et al., 2024b) focuses on advertising strategies for electric toothbrushes. We reprocess the raw data to extract temporal information and restructure it into the DyTAG format. The dataset includes user profiles (e.g., usernames, gender, regions, follower/followee counts, self-introductions) and interaction edges categorized as comment or repost, with edge texts comprising source post content and destination user comments/reposts. Named WeiboTech due to its focus on technology-related topics (e.g., electronics, automobiles), the dataset is a non-bipartite DyTAG, where Weibo users are nodes with textual features, and an edge represents user interactions at a given time.

WeiboDaily(Zhang et al., 2024b) is also a dataset collected from the Weibo social platform, recording user interactions (comments and reposts). Different from WeiboTech, the temporal span ranges from December 1, 2023, to December 31, 2023, and the original dataset (Zhang et al., 2024b) focuses on advertising strategies for ABCReading. Thus, WeiboDaily spans a full month and targets daily life topics (e.g., lifestyle sharing), enabling continuous modeling analysis. Similarly, the dataset includes user profiles (e.g., usernames, gender, regions, follower/followee counts, self-introductions) and interaction edges categorized as comment or repost, with edge texts comprising source post content and destination user comments/reposts. The dataset is a non-bipartite DyTAG, where Weibo users are nodes with textual features, and an edge represents user interactions at a given time.

Cora is an extended version of the classic citation network dataset Cora (Sen et al., 2008), documenting academic paper citation relationships. The temporal span ranges from February 1, 1985, to September 1, 2024. The original Cora dataset records citation relationships and node categories (Sen et al., 2008). We expand the dataset by crawling first-order and second-order paper information via references, enriching the size of the citation network. Node textual features include paper titles, abstracts, authors,

[3] https://dumps.wikimedia.org/enwiki/20241201/

[4] https://datasets.imdbws.com/

and citation counts. Edge texts are extracted from the exact sentence in the citing paper that references the cited paper. With regards to the edge labels, based on the section name of each citation in the citing paper, they are mapped to five categories: 1) Intro & Background, 2) Tech & Methodology, 3) Experiment & Conclusion, 4) Topic-specific, and 5) Others. The Cora dataset is a non-bipartite DyTAG, where papers are nodes with textual features, and an edge indicates a citation relationship at a given time.

C.2 Dataset Licenses

In this section, we provide dataset licenses for each proposed DyTAG.

Sephora: CC BY 4.0 license (Creative Commons Attribution 4.0 International License). The original dataset can be found here.

Dianping: CC BY-SA license (Creative Commons Attribution-ShareAlike License). The original dataset can be found here.

WikiRevision: GFDL (GNU Free Documentation License) and CC BY-SA license (Creative Commons Attribution-ShareAlike License). The original dataset can be found here.

WikiLife: GFDL (GNU Free Documentation License) and CC BY-SA license (Creative Commons Attribution-ShareAlike License). The original dataset can be found here.

IMDB: Subject to your compliance with these Conditions of Use and your payment of any applicable fees, IMDB or its content providers grants you a limited, non-exclusive, non-transferable, non-sublicenseable license to access and make personal and non-commercial use of the IMDB Services, including digital content available through the IMDB Services, and not to download (other than page caching) or modify this site, or any portion of it, except with express written consent of IMDB. Additional license terms may be found in the Terms. The IMDB Services or any portion of such services may not be reproduced, duplicated, copied, sold, resold, visited, or otherwise exploited for any commercial purpose without the express written consent of IMDB. This license does not include any resale or commercial use of any IMDB Service or its contents or any derivative use of this site or its contents. All licenses are non-exclusive, and all rights not expressly granted to you in these Conditions of Use or any applicable Terms are reserved and retained by IMDB or its licensors, suppliers, publishers, rightsholders, or other content providers. You will use all IMDB Services in compliance with all applicable laws. The original dataset can be found here.

WeiboTech: CC-BY 4.0 license (Creative Commons Attribution 4.0 International License). The original dataset can be found here.

WeiboDaily: CC-BY 4.0 license (Creative Commons Attribution 4.0 International License). The original dataset can be found here.

Cora: CC BY-SA license (Creative Commons Attribution-ShareAlike). The original dataset can be found here.

C.3 Dataset Analysis

To provide a comprehensive overview of our GDGB datasets, we present the node degree distribution in Figure 3. Most datasets exhibit a long-tailed distribution, which aligns with the real-world graph growth patterns governed by power-law principles (Clauset et al., 2009). This characteristic reflects the uneven connectivity observed in natural networks, such as social networks or citation graphs, where a few nodes dominate high degrees while the majority remain sparsely connected.

Furthermore, Figure 4 illustrates the edge label distribution of the GDGB datasets. The results reveal that most datasets maintain a balanced label distribution, mitigating the risk of performance degradation in edge classification tasks caused by extreme class imbalances. However, a subset of datasets exhibits relatively skewed label distributions, intentionally introducing realistic challenges akin to practical applications (e.g., rare but critical edge types in life trajectory or movie collaboration scenarios). This dual distribution pattern ensures both diversity and practical relevance, enhancing the dataset's utility for robust model evaluation.

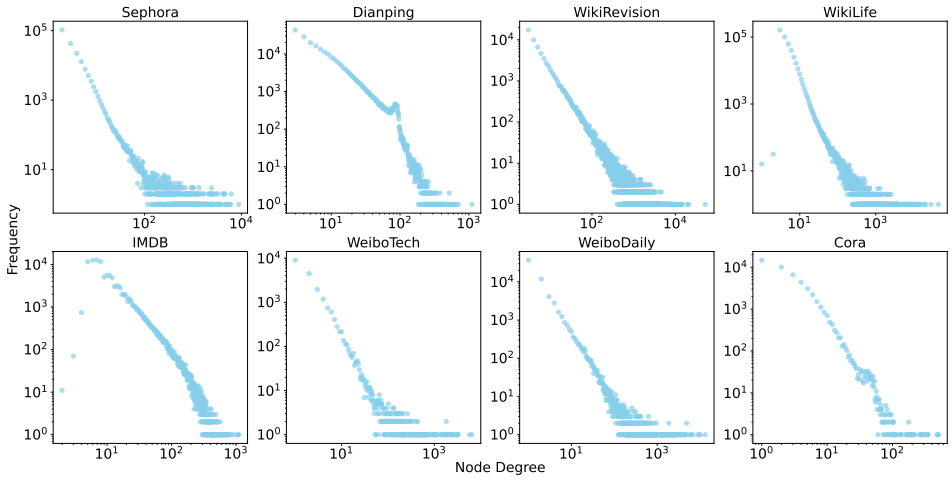

Figure 3: Distribution of node degree on GDGB datasets.

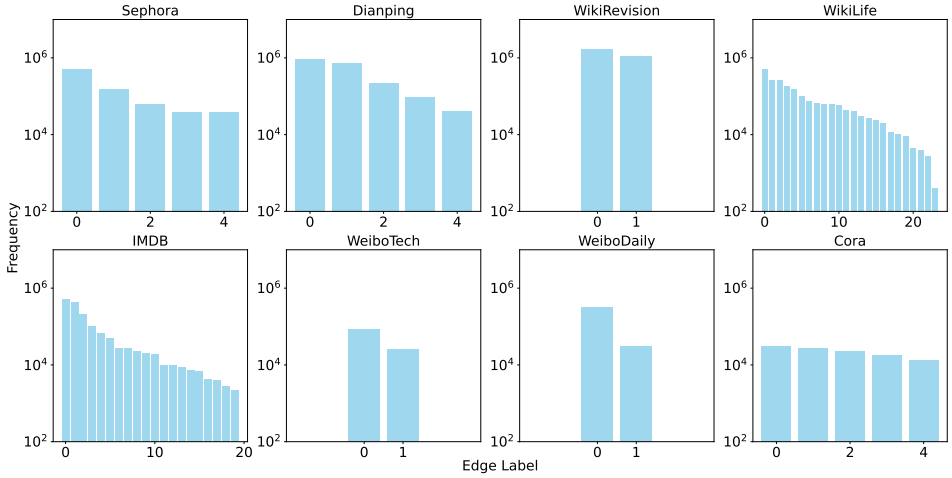

Figure 4: Distribution of the number of edges for each label on GDGB datasets.

C.4 Comparison with Existing Datasets for Graph Generation.

In graph generation research, while significant progress has been made in molecular graph generation, non-molecular graph generation has received comparatively less attention (Faez et al., 2021). Consequently, existing models for non-molecular graph generation are often trained on synthetic datasets such as Grid (Dai et al., 2020), Community, and SBM (You et al., 2018; Guo & Zhao, 2023), or on static, topology-only graphs, such as Ego and Polblogs (Chen et al., 2023). Further, we observe a lack of focus on dynamic graph generation, despite the fact that real-world non-molecular graphs inherently exhibit temporal evolution in both topology and node attributes (Guo & Zhao, 2023). Among the most commonly used continuous-time dynamic graph datasets, such as LastFM and MOOC (Hosseini et al., 2024; Gupta et al., 2022; Poursafaei et al., 2022b), both are derived from interaction networks (Poursafaei et al., 2022a) that inherently include textual attributes of users/items and timestamped interactions. However, these datasets suffer from critical limitations: MOOC provides only numerical features extracted from raw text, while LastFM discards attribute information entirely.

Although widely adopted, existing datasets face notable challenges in modeling the co-evolution of temporal, structural, and textual attributes in graphs. First, static graph datasets lack temporal annotations, making them incompatible with dynamic graph generation tasks. Even discrete-time dynamic graph datasets like Bitcoin-Alpha (Kumar et al., 2016; Gupta et al., 2022), which usually represent graphs as discrete daily snapshots, suffer from sparse temporal distributions of edges, complicating the fine-grained modeling of structural and temporal dynamics. Second, most dynamic

Table 9: Statistics of our proposed datasets and comparison with existing datasets for generative tasks. $|V|_{\max}$ and $|E|_{\max}$ represent the maximum number of nodes and maximum number of edges in the graph list of the dataset, respectively.

	Dataset	Nodes	Edges	Edge Categories	Timestamps	Domain	Text Attributes	Bipartite
Previous Static Graphs	**Grid**	$\|V\|_{\max} = 361$	$\|E\|_{\max} = 684$	\	\	Synthetic	\	✗
	Community	$\|V\|_{\max} = 160$	$\|E\|_{\max} = 1945$	\	\	Synthetic	\	✗
	Ego	$\|V\|_{\max} = 399$	$\|E\|_{\max} = 1071$	\	\	Social Network	\	✗
	Polblogs	1,222	16,714	\	\	Interaction	\	✗
Previous Dynamic Graphs	**MOOC**	7,047/97	411,749	\	345,600	Interaction	\	✓
	LastFM	980/1,000	1,293,103	\	1,283,614	Interaction	\	✓
	Reddit	10,000/984	672,447	\	669,065	Social	\	✓
	Wikipedia	8,227/1,000	157,474	\	152,757	Interaction	\	✗
	Bitcoin-Alpha	3,783	24,186	\	191	Financial Network	\	✗
Previous DyTAGs (DTGB)	**Stack elec**	67,155/330,547	1,262,225	2	5,224	Multi-round dialogue	Node & Edge	✓
	Stack ubuntu	180,261/493,987	1,497,006	2	4,972	Multi-round dialogue	Node & Edge	✓
	Googlemap CT	83,796/27,372	1,380,623	5	55,521	E-commerce	Node & Edge	✓
	Amazon movies	233,459/60,107	3,217,324	5	7,287	E-commerce	Node & Edge	✓
	Yelp	1,987,896/150,346	6,990,189	5	6,036	E-commerce	Node & Edge	✓
	Enron	42,711	797,907	10	1,006	E-mail	Node & Edge	✗
	GDELT	6,786	1,339,245	237	2,591	Knowledge graph	Node & Edge	✗
	ICEWS1819	31,796	1,100,071	266	730	Knowledge graph	Node & Edge	✗
GDGB	**Sephora**	210,357/2,274	801,234	5	5,314	E-commerce	Node & Edge	✓
	Dianping	158,541/88,118	1,990,409	5	745,151	E-commerce	Node & Edge	✓
	WikiRevision	75,622/3,204	2,778,732	2	2,766,153	Web Interaction	Node & Edge	✓
	WikiLife	406,148/54,513	1,996,520	24	1,810	Celebrity Biography	Node & Edge	✓
	IMDB	125,714	1,534,162	20	122	Movie Collaboration	Node & Edge	✗
	WeiboTech	20,767	109,345	2	79,925	Social Network	Node & Edge	✗
	WeiboDaily	66,500	354,098	2	293,662	Social Network	Node & Edge	✗
	Cora	48,797	110,788	5	8,274	Citation	Node & Edge	✗

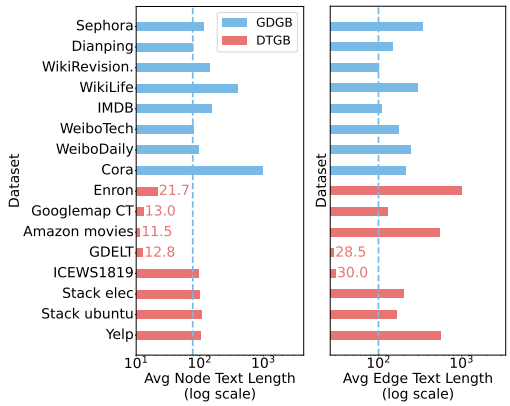

Figure 5: **Left**: Average node text lengths on GDGB and DTGB datasets. In non-bipartite graphs, text lengths are averaged across all nodes. For bipartite graphs, averages are calculated for source nodes. **Right**: Average edge text lengths on each dataset.

graph datasets omit raw textual data, relying instead on preprocessed numeric or categorical features (Hosseini et al., 2024), which limits the exploration of rich textual attributes in real-world scenarios. Finally, more recently, datasets such as the DyTAGs from DTGB (Zhang et al., 2024a), while providing edge-level text features, often lack sufficient and high-quality node-level textual information. For instance, six out of eight datasets of DTGB contain (source) node texts that are typically just identifiers like email addresses or username. This is a critical limitation, as real-world applications frequently require detailed node-wise text attributes to disambiguate entities beyond IDs. The statistics of the above mentioned datasets and our proposed datasets are shown in Table 9.

C.5 Comparison Between GDGB and DTGB Datasets

We provide a comprehensive analysis and empirical comparison between our proposed **GDGB** and the existing **DTGB**. While both benchmarks contain dynamic text-attributed graphs, we demonstrate that GDGB is fundamentally distinct in construction philosophy, data quality, and utility for generative tasks.

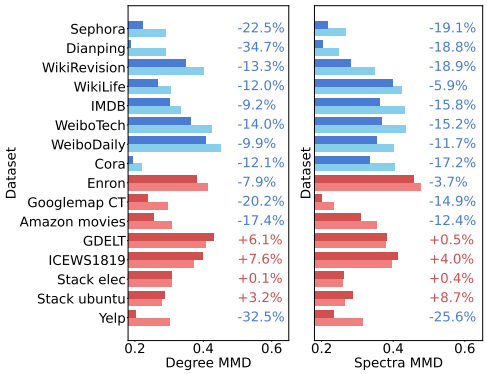

Figure 6: **Left**: Comparison of Degree MMD w/ and w/o text on GDGB datasets and DTGB datasets in DG-Gen. **Right**: Comparison of Spectra MMD w/ and w/o text on GDGB datasets and DTGB datasets in DG-Gen.

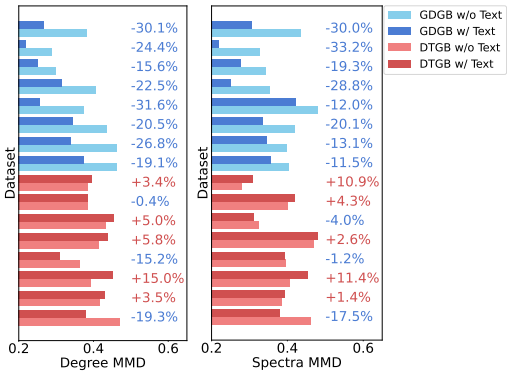

Figure 7: **Left**: Comparison of Degree MMD w/ and w/o text on GDGB datasets and DTGB datasets in VRDAG. **Right**: Comparison of Spectra MMD w/ and w/o text on GDGB datasets and DTGB datasets in VRDAG.

Textual Information Richness and Functional Utility. As illustrated in Figure 5, we compare the average text length of node and edge attributes across datasets in GDGB and DTGB, demonstrating that GDGB datasets exhibit significantly richer textual content. However, text length alone does not guarantee utility in generative modeling. To assess the functional value of textual features, we evaluate two feature-aware dynamic graph generation models—VRDAG (Li et al., 2024) and DG-Gen (Hosseini et al., 2024)—on both benchmarks, with and without the incorporation of BERT-based textual embeddings. As shown in Figures 6 and 7, textual features consistently **improve** generation quality on GDGB, as evidenced by reduced Degree and Spectra MMD scores. In contrast, on DTGB, the inclusion of textual features often **degrades** performance—particularly for VRDAG, which is highly dependent on node feature inputs. This suggests that the textual attributes in DTGB are semantically impoverished or noisy, and thus detrimental to generative tasks, whereas GDGB's textual content is both meaningful and effectively leveraged by learning models.

Intrinsic Text Quality Evaluation. To directly evaluate the linguistic quality of node and edge texts, we conduct intrinsic analysis using two complementary metrics: (i) **Perplexity (PPL)**, where lower values indicate higher fluency and coherence with natural language distributions; and (ii) **LLM-based human-like scoring** (on a 1–5 scale) across five dimensions: *Contextual Fidelity*, *Personality Depth*, *Dynamic Adaptability*, *Immersive Quality*, and *Content Richness*. The aggregated results are summarized in Table 10. GDGB exhibits significantly lower perplexity (average value less than 90), indicating that its textual content is more coherent and aligned with natural language patterns. Furthermore, LLM evaluations confirm the superior semantic quality of GDGB, with average scores exceeding 4.0 across both node and edge texts. In contrast, DTGB achieves notably lower scores for node texts (average: 3.03), particularly in datasets such as Enron, where node attributes often consist of identifiers (e.g., email addresses) lacking meaningful semantics.

Table 10: Intrinsic text quality comparison (averaged across datasets).

Metric	DTGB	GDGB
PPL of node texts ↓	193.58	**75.95**
PPL of edge texts ↓	181.82	**88.17**
LLM rating (node texts) ↑	3.03	**4.21**
LLM rating (edge texts) ↑	3.55	**4.17**

Generative Performance on GAG-General. To further assess the suitability of each benchmark for DyTAG generation, we apply the newly proposed GAG-General framework to all datasets from DTGB and all datasets from GDGB, using GPT as the LLM backbone under both TDGG and IDGG tasks. All models are evaluated under identical configurations in Section E.4 to ensure fair comparison.

The results under TDGG are presented in Table 11, 12, and 13. While, the results under IDGG are presented in Table 14, 15, and 16. The above results reveal consistent advantages for GDGB across structural, textual, and embedding fidelity metrics. For instance, in the graph embedding metric under TDGG (Table 13), DTGB results predominantly fall within the 0.2–0.4 range, while GDGB achieves substantially greater similarity (0.4–0.7), indicating better preservation of global structural and semantic properties. An even more pronounced trend is observed under IDGG: none of the DyTAGs generated on DTGB datasets satisfy the power-law validity criterion (Table 14), and both average textual quality scores (Table 15) and graph embedding metrics (Table 16) are generally lower than those achieved on GDGB datasets. This again highlights the limitations of DTGB's textual content in effectively supporting generative DyTAG tasks. Although the integration of memory and reflection mechanisms in GAG-General yields improvements on DTGB, these gains are constrained by the underlying data limitations, underscoring the dataset's inherent constraints for generative tasks.

Table 11: The results of graph structural quality under TDGG with GPT as the LLM backbone.

Benchmark	Dataset	Degree MMD ↓	Spectra MMD ↓	D_k	α	Power-law Validity
DTGB	Enron	0.331	0.266	0.125	1.982	×
	GDELT	0.252	0.326	0.174	1.829	×
	ICEWS1819	0.302	0.341	0.132	1.794	×
	Stack elec	0.228	0.345	0.133	1.822	×
	Stack ubuntu	0.235	0.359	0.130	1.901	×
	Googlemap CT	0.255	0.375	0.080	2.278	✓
	Amazon movies	0.404	0.350	0.182	1.636	×
	Yelp	0.213	0.224	0.121	2.033	✓
GDGB	Sephora	0.023	0.011	0.143	2.993	✓
	Dianping	0.055	0.328	0.041	2.234	✓
	WikiRevision	0.108	0.156	0.056	2.041	✓
	WikiLife	0.181	0.223	0.099	2.204	✓
	IMDB	0.278	0.316	0.135	1.720	×
	WeiboTech	0.243	0.297	0.030	2.011	✓
	WeiboDaily	0.247	0.493	0.048	1.845	×
	Cora	0.128	0.156	0.049	2.378	✓

Table 12: The results on average textual quality scores under TDGG with GPT as the LLM backbone. M. and R. denote node memory and reflection mechanism, respectively.

Benchmark	Dataset	w/o M.	w/ M.	w/ M. & R.
DTGB	Enron	3.65	3.72	**3.83**
	GDELT	3.70	3.98	**4.23**
	ICEWS1819	3.47	3.87	**4.14**
	Stack elec	3.68	3.70	**3.96**
	Stack ubuntu	3.38	3.56	**3.78**
	Googlemap CT	3.10	3.30	**4.06**
	Amazon movies	3.78	3.79	**4.11**
	Yelp	3.80	4.23	**4.33**
GDGB	Sephora	4.69	4.69	**4.77**
	Dianping	4.32	4.34	**4.71**
	WikiRevision	4.74	4.79	**4.96**
	WikiLife	4.44	4.46	**4.59**
	IMDB	3.91	4.02	**4.44**
	WeiboTech	4.84	4.88	**4.97**
	WeiboDaily	4.80	4.92	**4.99**
	Cora	3.98	4.10	**4.52**

Summary. Collectively, the experimental results demonstrate that GDGB significantly outperforms DTGB across multiple dimensions of generative evaluation. The superior performance on structural metrics, textual quality, and embedding similarity highlights GDGB's enhanced semantic richness and compatibility with generative modeling frameworks. In contrast, the limited and often noisy textual content in DTGB hinders its effectiveness in supporting high-fidelity DyTAG generation. Therefore, GDGB is not merely an extension of DTGB with longer text, but a **purpose-built benchmark for**

Table 13: The results on the graph embedding metric under TDGG with GPT as the LLM backbone.

Benchmark	Dataset	w/o M.	w/ M.	w/ M. & R.
DTGB	Enron	0.201	0.259	**0.400**
	GDELT	0.255	0.348	**0.393**
	ICEWS1819	0.198	0.289	**0.383**
	Stack elec	0.235	**0.319**	0.291
	Stack ubuntu	0.223	0.327	**0.330**
	Googlemap CT	0.271	0.302	**0.462**
	Amazon movies	0.303	0.359	**0.427**
	Yelp	0.279	0.396	**0.411**
GDGB	Sephora	0.588	**0.671**	0.654
	Dianping	0.351	0.368	**0.369**
	WikiRevision	0.438	0.422	**0.454**
	WikiLife	0.459	0.453	**0.463**
	IMDB	0.432	0.420	**0.435**
	WeiboTech	0.327	0.488	**0.698**
	WeiboDaily	0.681	0.689	**0.694**
	Cora	**0.511**	0.446	0.465

Table 14: The results of graph structural quality under IDGG with GPT as the LLM backbone.

Benchmark	Dataset	Degree MMD ↓	Spectra MMD ↓	D_k	α	Power-law Validity
DTGB	Enron	0.412	0.378	0.132	1.742	×
	GDELT	0.489	0.456	0.167	1.891	×
	ICEWS1819	0.503	0.412	0.184	1.623	×
	Stack elec	0.537	0.459	0.181	1.654	×
	Stack ubuntu	0.521	0.488	0.195	1.577	×
	Googlemap CT	0.398	0.342	0.112	1.803	×
	Amazon movies	0.476	0.401	0.143	1.668	×
	Yelp	0.355	0.317	0.078	1.921	×
GDGB	Sephora	0.454	0.189	0.147	2.057	✓
	Dianping	0.300	0.453	0.069	2.430	✓
	WikiRevision	0.267	0.229	0.045	2.050	✓
	WikiLife	0.083	0.208	0.099	2.204	✓
	IMDB	0.321	0.423	0.252	1.746	×
	WeiboTech	0.268	0.201	0.064	1.867	×
	WeiboDaily	0.268	0.439	0.156	1.732	×
	Cora	0.136	0.244	0.091	2.250	✓

generative DyTAG tasks, where high-quality textual attributes are a **core design principle**. These findings establish GDGB as a more suitable foundation for advancing research in generative modeling of dynamic text-attributed graphs.

D Details of Tasks and Metrics

D.1 Details of GAG-General

GAG-General is a generalized agent-based framework designed for both TDGG and IDGG within the DyTAG paradigm. It extends the existing GAG framework (Ji et al., 2025) through key architectural and procedural generalizations, enabling broader applicability and domain adaptability. The full procedure is specified in Algorithms 1 and 2; here we detail its design and core distinctions from GAG.

The framework proceeds in three stages. First, generalized node formulation initializes node agents, each optionally equipped with Node Memory and a Reflection Mechanism to maintain and summarize interaction history. Unlike GAG, this stage supports both bipartite and non-bipartite graph structures, allowing modeling of diverse network types. Second, during abstracted interaction simulation, agents generate interactions (or new nodes in IDGG) over sequential rounds. Guided by an LLM backbone and informed by memory and reflection, agents produce textual content such as relations and attributes.

Table 15: The results on average textual quality scores under IDGG with GPT as the LLM backbone. M. and R. denote node memory and reflection mechanism, respectively.

Benchmark	Dataset	w/o M.	w/ M.	w/ M. & R.
DTGB	Enron	3.51	3.89	**3.93**
	GDELT	3.62	3.82	**4.02**
	ICEWS1819	3.59	3.90	**4.25**
	Stack elec	3.23	3.46	**3.64**
	Stack ubuntu	3.19	3.47	**3.73**
	Googlemap CT	3.22	3.42	**3.87**
	Amazon movies	3.63	3.75	**4.22**
	Yelp	3.53	4.03	**4.16**
GDGB	Sephora	4.58	4.77	**4.87**
	Dianping	4.56	4.71	**4.86**
	WikiRevision	4.39	4.54	**4.71**
	WikiLife	4.28	4.39	**4.47**
	IMDB	4.19	4.29	**4.49**
	WeiboTech	4.60	4.71	**4.93**
	WeiboDaily	4.74	4.83	**4.99**
	Cora	4.27	4.36	**4.57**

Table 16: The results on the graph embedding metric under IDGG with GPT as the LLM backbone.

Benchmark	Dataset	w/o M.	w/ M.	w/ M. & R.
DTGB	Enron	0.221	0.278	**0.387**
	GDELT	0.242	0.359	**0.388**
	ICEWS1819	0.234	0.297	**0.361**
	Stack elec	0.243	0.289	**0.321**
	Stack ubuntu	0.239	**0.319**	0.309
	Googlemap CT	0.289	**0.333**	0.330
	Amazon movies	0.294	0.338	**0.396**
	Yelp	0.281	0.369	**0.417**
GDGB	Sephora	0.601	0.603	**0.628**
	Dianping	0.521	**0.527**	0.505
	WikiRevision	0.589	0.614	**0.629**
	WikiLife	0.510	**0.534**	0.531
	IMDB	0.521	0.535	**0.563**
	WeiboTech	0.501	0.514	**0.536**
	WeiboDaily	0.459	0.481	**0.502**
	Cora	0.541	0.552	**0.572**

Crucially, this workflow is abstracted from domain-specific rules, enabling seamless transfer across datasets without customization—a significant departure from GAG's rigid, task-specific designs. Finally, in the graph projection and evaluation phase, all generated elements are compiled into a dynamic graph and evaluated using our multi-dimensional metrics.

Hence, the key generalizations over GAG lie in three aspects. 1) GAG-General removes the bipartite constraint, supporting arbitrary graph topologies. 2) It replaces GAG's domain-specific simulation logic with a unified, abstracted process, enhancing reusability across domains. 3) While GAG focuses on a single generation mode, GAG-General explicitly supports both TDGG and IDGG, capturing distinct dynamics of temporal evolution and structural growth.

In sum, GAG-General serves as a demonstration framework that validates the feasibility of generative DyTAG modeling. Its generalizations extend beyond the original GAG's limitations, offering a more flexible foundation for future work. While our primary contribution is the GDGB benchmark—encompassing datasets, tasks, and metrics—GAG-General illustrates how such benchmarks can drive the development of more generalizable generative frameworks.

D.2 DETAILS OF TDGG AND IDGG ON GAG-GENERAL

Transductive Dynamic Graph Generation. TDGG operates on a fixed node set $\mathcal{N}$, evolving from an initial seed graph $\mathcal{G}_0 = (\mathcal{N}, \mathcal{E}^0)$ to a final graph $\mathcal{G}_K = (\mathcal{N}, \mathcal{E}^K)$ after K rounds. For instance, during the k-th round ($1 \leq k \leq K$), the DyTAG generation process includes the following three stages:

1. **Node Update**: Based on the given source node set $\mathcal{N}_{\text{src}}^{k-1}$ and destination node set $\mathcal{N}_{\text{dst}}^{k-1}$ from the last round, we first update the node sets as $\mathcal{N}_{\text{src}}^{k} = \mathcal{N}_{\text{src}}^{k-1} \cup \widetilde{\mathcal{N}_{\text{src}}^{k}}$ and $\mathcal{N}_{\text{dst}}^{k} = \mathcal{N}_{\text{dst}}^{k-1} \cup \widetilde{\mathcal{N}_{\text{dst}}^{k}}$, where $\widetilde{\mathcal{N}_{\text{src}}^{k}}$ and $\widetilde{\mathcal{N}_{\text{dst}}^{k}}$ are obtained based on the original DyTAG with a predefined size. We then initialize active source nodes as $\mathcal{N}_{\text{src}}^{k-\text{active}} = \widetilde{\mathcal{N}_{\text{src}}^{k}}$ for the following transducitve generation.
2. **Node Selection**: According to $\mathcal{N}_{\text{src}}^{k-\text{active}}$, we activate source node agents to perform pairwise interactions. LLM-based agents recall and select final destination nodes $\mathcal{N}_{\text{dst}}^{k-\text{select}}$ from $\mathcal{N}_{\text{dst}}^{k}$ by analyzing the source node's textual profile and its memories. The memory reflection mechanism is optionally used to summarize the memories.
3. **Interaction Generation**: After the destination node selection, the active source node agents in the k-th round generate a new edge set $\widetilde{\mathcal{E}^k}$ (between $\mathcal{N}_{\text{src}}^{k-\text{active}}$ and $\mathcal{N}_{\text{dst}}^{k-\text{select}}$), including edge labels $\mathcal{L}_{u,v}$ and edge texts $\mathcal{E}_{u,v}^{\text{text}}$. We then update the edge set as $\mathcal{E}^k = \mathcal{E}^{k-1} \cup \widetilde{\mathcal{E}^k}$.

The destination node selection and edge generation stages in TDGG naturally integrate traditional discriminative tasks (e.g., node retrieval, edge classification) with LLM-driven text understanding and generation, forming a DyTAG-specific generative paradigm. The pseudocode of TDGG on GAG-General is shown in Algorithm 1.

Algorithm 1: Transductive Dynamic Graph Generation

Require: Initial seed DyTAG $\mathcal{G}_0 = (\mathcal{N}, \mathcal{E}^0)$, rounds K, edge generation size S
Ensure: Final generated DyTAG $\mathcal{G}_K = (\mathcal{N}, \mathcal{E}^K)$

Initialize $\mathcal{N}_{\text{src}}^0 = \mathcal{N}_{\text{src}}^{\text{init}}, \mathcal{N}_{\text{dst}}^0 = \mathcal{N}_{\text{dst}}^{\text{init}}, \mathcal{E}^0 = \mathcal{E}_{\text{init}}^0$
for $k = 1$ to K **do**
  Node Update:
  $\widetilde{\mathcal{N}_{\text{src}}^{k}} = \text{Sample}(\mathcal{N}, S), \widetilde{\mathcal{N}_{\text{dst}}^{k}} = \text{Sample}(\mathcal{N}, S)$
  $\mathcal{N}_{\text{src}}^{k} = \mathcal{N}_{\text{src}}^{k-1} \cup \widetilde{\mathcal{N}_{\text{src}}^{k}}, \mathcal{N}_{\text{dst}}^{k} = \mathcal{N}_{\text{dst}}^{k-1} \cup \widetilde{\mathcal{N}_{\text{dst}}^{k}}$
  $\mathcal{N}_{\text{src}}^{k-\text{active}} = \widetilde{\mathcal{N}_{\text{src}}^{k}}$
  Node Selection:
  $\mathcal{N}_{\text{dst}}^{k-\text{select}} = \text{Recall, Select}(\mathcal{N}_{\text{src}}^{k-\text{active}}, \mathcal{N}_{\text{dst}}^{k})$
  Interaction Generation:
  $\widetilde{\mathcal{E}^k} = \text{GenerateEdges}(\mathcal{N}_{\text{src}}^{k-\text{active}}, \mathcal{N}_{\text{dst}}^{k-\text{select}})$
  $\mathcal{E}^k = \mathcal{E}^{k-1} \cup \widetilde{\mathcal{E}^k}$
end for
return $\mathcal{G}_K = (\mathcal{N}, \mathcal{E}^K)$

Inductive Dynamic Graph Generation. IDGG extends TDGG by simultaneously generating new nodes and edges. Starting from seed graph $\mathcal{G}_0 = (\mathcal{N}^0, \mathcal{E}^0)$, it evolves to $\mathcal{G}_K = (\mathcal{N}^K, \mathcal{E}^K)$ through K rounds. Key steps for the k-th round ($1 \leq k \leq K$) include:

1. **Node Generation**: We apply node generator agents to generate $\widetilde{\mathcal{N}_{\text{src}}^{k}}$ (size of R_{src}) and $\widetilde{\mathcal{N}_{\text{dst}}^{k}}$ (size of R_{dst}) along with their textual features, according to the recent active nodes. R_{src} and R_{dst} are predefined by the seed graph based on the average numbers of new source nodes and new destination nodes per round.
2. **Node Update**: After node generation, we update the node sets as $\mathcal{N}_{\text{src}}^{k} = \mathcal{N}_{\text{src}}^{k-1} \cup \widetilde{\mathcal{N}_{\text{src}}^{k}}$ and $\mathcal{N}_{\text{dst}}^{k} = \mathcal{N}_{\text{dst}}^{k-1} \cup \widetilde{\mathcal{N}_{\text{dst}}^{k}}$. We then initialize active source nodes as $\mathcal{N}_{\text{src}}^{k-\text{active}} = \text{RndSample}(\mathcal{N}_{\text{src}}^{k})$ for the following inductive generation, where $\text{RndSample}()$ is a random sampling function.

3. **Node Selection & Interaction Generation**: Similar to TDGG, active source node agents recall and select destination nodes $\mathcal{N}_{\text{dst}}^{k-\text{select}}$ from $\mathcal{N}_{\text{dst}}^{k}$, and generate $\widetilde{\mathcal{E}^k}$ (between $\mathcal{N}_{\text{src}}^{k-\text{active}}$ and $\mathcal{N}_{\text{dst}}^{k-\text{select}}$), including edge labels $\mathcal{L}_{u,v}$ and edge texts $\mathcal{E}_{u,v}^{\text{text}}$. We then update the edge set as $\mathcal{E}^k = \mathcal{E}^{k-1} \cup \widetilde{\mathcal{E}^k}$.

The IDGG above faithfully replicates the evolutionary dynamics of real-world DyTAG through new node and edge generation over time. Although inherently challenging, this paradigm establishes a foundational direction for advancing dynamic graph generation research by simulating open-ended growth processes. The pseudocode of IDGG on GAG-General is shown in Algorithm 2. The source/destination node generation counts $R_{\text{src}}/R_{\text{dst}}$ for IDGG are shown in Table 17).

Algorithm 2: Inductive Dynamic Graph Generation

Require: Initial seed DyTAG $\mathcal{G}_0 = (\mathcal{N}^0, \mathcal{E}^0)$, rounds K, edge generation size S, source/destination node generation numbers $R_{\text{src}}, R_{\text{dst}}$
Ensure: Final generated DyTAG $\mathcal{G}_K = (\mathcal{N}^K, \mathcal{E}^K)$
$\mathcal{N}_{\text{src}}^0 = \mathcal{N}_{\text{src}}^{\text{init}}, \mathcal{N}_{\text{dst}}^0 = \mathcal{N}_{\text{dst}}^{\text{init}}, \mathcal{E}^0 = \mathcal{E}_{\text{init}}^0$
for $k = 1$ to K **do**
  Node Generation:
  $\widetilde{\mathcal{N}_{\text{src}}^k} = \text{GenerateNodes}(R_{\text{src}}), \widetilde{\mathcal{N}_{\text{dst}}^k} = \text{GenerateNodes}(R_{\text{dst}})$
  Node Update:
  $\mathcal{N}_{\text{src}}^k = \mathcal{N}_{\text{src}}^{k-1} \cup \widetilde{\mathcal{N}_{\text{src}}^k}, \mathcal{N}_{\text{dst}}^k = \mathcal{N}_{\text{dst}}^{k-1} \cup \widetilde{\mathcal{N}_{\text{dst}}^k}$
  $\mathcal{N}_{\text{src}}^{k-\text{active}} = \text{RndSample}(\mathcal{N}_{\text{src}}^k, S)$
  Node Selection:
  $\mathcal{N}_{\text{dst}}^{k-\text{select}} = \text{Recall, Select}(\mathcal{N}_{\text{src}}^{k-\text{active}}, \mathcal{N}_{\text{dst}}^k)$
  Interaction Generation:
  $\widetilde{\mathcal{E}^k} = \text{GenerateEdge}(\mathcal{N}_{\text{src}}^{k-\text{active}}, \mathcal{N}_{\text{dst}}^{k-\text{select}})$
  $\mathcal{E}^k = \mathcal{E}^{k-1} \cup \widetilde{\mathcal{E}^k}$
end for
return $\mathcal{G}_K = (\mathcal{N}^K, \mathcal{E}^K)$

Table 17: The source node generation counts R_{src} and the destination node generation counts R_{dst}, which are determined by the seed graph.

Dataset	Sephora	Dianping	WikiRevision	WikiLife	IMDB	WeiboTech	WeiboDaily	Cora
R_{src}	27	33	6	13	17	3	6	32
R_{dst}	4	44	37	12	17	29	19	18

D.3 Details of Metrics

We provide more detailed mathematical formulations and implementation specifics of our evaluation metrics in Section 4.2 as follows.

Graph Structural Metrics. We evaluate the structural quality of generated DyTAGs using two classical approaches:

- **Degree/Spectra MMD.** This metric quantifies the discrepancy between the distribution of graph descriptors (e.g., degree/spectral properties) in the generated and ground-truth graphs (Xiang et al., 2022; You et al., 2018; Chen et al., 2023). We compute the distribution distance, using the maximum mean discrepancy (MMD) with a radial basis function (RBF) kernel. Thus, lower MMD values indicate higher structural fidelity. Specifically, we employ a positive definite RBF kernel with a smoothing parameter v, defined as:

$$k\left(x_i, x_j\right) = \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2v^2}\right),$$

  where x_i and x_j are feature vectors representing graph descriptors (e.g., degrees/spectral properties) for nodes i and j. v denotes smoothing parameter controlling the width of the

RBF kernel. Larger v values reduce sensitivity to local variations. $k(\cdot,\cdot)$ denotes the RBF kernel function measuring similarity between graph descriptors.

We treat nodes as samples, where n and m denote the number of nodes in the generated and ground-truth graphs, respectively. The MMD is computed as:

$$\mathrm{MMD}^2(X,Y) := \frac{1}{n^2}\sum_{i,j=1}^{n}\mathrm{k}\left(x_i,x_j\right) + \frac{1}{m^2}\sum_{i,j=1}^{m}\mathrm{k}\left(y_i,y_j\right) - \frac{2}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}\mathrm{k}\left(x_i,y_j\right)$$

where $X = \{x_1, x_2, \ldots, x_n\}$is the set of feature vectors from the generated DyTAG. $Y = \{y_1, y_2, \ldots, y_m\}$is the set of feature vectors from the ground-truth DyTAG.

- **Power-law Analysis** Real-world networks often exhibit power-law degree distributions. We assess power-law behavior in degree distributions using the Kolmogorov-Smirnov (KS) distance:

$$D_k = \max_{k \geq k_{\min}} |F(k) - H(k)|,$$

where $F(k)$ is the empirical cumulative distribution function (CDF) of the degree data, and $H(k)$ is the CDF of the best-fitting power-law model. $k_{\min}$ denotes the minimum degree value used for fitting, set to $k_{\min} = 2$ to ensure robustness. The power-law exponent α indicates adherence to a power-law distribution, typically valid if $\alpha \in [2, 3]$ (Clauset et al., 2009). A DyTAG is power-law valid if $D_k < 0.15$ and the power-law exponent $\alpha \in [2, 3]$.

Textual Quality Metrics. Inspired by recent role-playing agent research (Wang et al., 2024b; Chen et al., 2024; Wang et al., 2024a), we employ the LLM-as-Evaluator framework to assess the text quality in the generated DyTAGs. Considering the targets and requirements of DyTAG generation tasks, we define five scoring criteria:

1. **Contextual Fidelity**: Consistency of node/edge texts with historical interactions.
2. **Personality Depth**: Richness of semantic and stylistic diversity in node profiles.
3. **Dynamic Adaptability**: Temporal coherence of evolving textual content.
4. **Immersive Quality**: Engagement and realism of generated narratives.
5. **Content Richness**: Information density and relevance of edge texts.

Each criterion is scored on a 1–5 scale by LLM evaluators, ensuring a multidimensional assessment of textual quality. For the LLM-as-Evaluator component, we consistently employ GPT as the underlying LLM backbone for all baselines, with hyperparameters (e.g., temperature, top-p, and repetition penalty) kept identical to those specified in Section E.4. The prompts used in the LLM-as-Evaluator framework are provided in Section G.3. This approach outperforms embedding-based metrics like BERTScore (Zhang et al., 2020) by avoiding information loss during feature compression.

Graph Embedding Metric. To jointly evaluate structural, temporal, and textual fidelity, we extend the JL-Metric (Hosseini et al., 2025) to a better adaptation on DyTAGs with textual node feature integration to a graph embedding-based indicator. Specifically, the metric is calculated through three stages:

- **Node Embedding Construction.** For node j, we concatenate textual attributes and temporal information of historical interactions into a contextualized node embedding:

$$\mathbf{v}_j = \left[\tilde{c}(\mathcal{T}_{j,1}) \| \tilde{c}(\mathcal{T}_{j,2}) \| \ldots \| \tilde{c}(\mathcal{T}_{j,m_j}) \| \mathcal{N}_j^{\text{text}}\right],$$

where $\tilde{c}(\mathcal{T}_{j,i}) = (\mathcal{T}_{j,i}, \mathcal{E}_{j,i}^{\text{text}}(t_i), \mathcal{N}_i^{\text{text}})$ encodes the i-th historical interaction event (including timestamp $\mathcal{T}_{j,i}$, textual edge feature $\mathcal{E}_{j,i}^{\text{text}}(t_i)$, and textual interacted node feature $\mathcal{N}_i^{\text{text}}$), and m_j is the total historical interaction count for node j. Additionally, we append the textual attribute of node j as the final component of the embedding vector, denoted by $\mathcal{N}_j^{\text{text}}$.

- **Random Projection Mapping.** Leveraging the Johnson-Lindenstrauss theorem for transforming varying dimensionality with bounded error guarantees (Hosseini et al., 2025), we project high-dimensional embeddings into a consistent space with random matrices, which enables cross-graph alignment despite different node counts and interaction histories. The projected graph embedding is:

$$\hat{\mathcal{G}} = \{\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \ldots, \hat{\mathbf{v}}_j, \ldots\},$$

where $\hat{\mathbf{v}}_j$ denotes the j-th projected node embedding.

- **Similarity Measurement.** We compute the cosine distance between the generated graph $\hat{\mathcal{G}}_g$ and ground-truth graph $\hat{\mathcal{G}}_o$:

$$\rho(\hat{\mathcal{G}}_g, \hat{\mathcal{G}}_o) = 1 - \frac{\langle \hat{\mathcal{G}}_g, \hat{\mathcal{G}}_o \rangle_F}{\|\hat{\mathcal{G}}_g\|_F \cdot \|\hat{\mathcal{G}}_o\|_F},$$

where $\|\cdot\|_F$ is the Frobenius norm and $\langle \cdot, \cdot \rangle_F$ is the Frobenius inner product. Lower ρ values indicate higher global fidelity.

E Supplementary Experimental Details

E.1 Large Language Model Baselines

LLMs possess strong capabilities in language understanding and generation, demonstrating impressive performance across a wide range of natural language processing tasks. To comprehensively evaluate the performance of different models, we selected several representative LLMs for comparison in our experiments. These include open-source models: DeepSeek-R1-Distill-Qwen-32B (DeepSeek-AI, 2025), Llama-3-70B-Instruct (AI@Meta, 2024), and Qwen2.5-72B-Instruct (Team, 2024). In addition, we include a closed-source model, GPT-4o-Mini (OpenAI, 2024), as a reference baseline.

E.2 Dynamic Graph Neural Network Baselines

JODIE (Kumar et al., 2019). This model is a representation learning framework for nodes in temporal networks that captures dynamic embedding trajectories based on sequences of interactions. JODIE uses two coupled recurrent neural networks to update the states of interacting entities and employs a projection operation to forecast future embedding trajectories.

TGN (Rossi et al., 2020). This method models sequences of time-stamped events. By combining memory modules with graph-based operators, TGN effectively captures temporal and structural dependencies in evolving graphs. This innovative architecture enables TGNs to maintain historical context while updating node representations in real time.

CAWN (Wang et al., 2021a). This approach inductively represents temporal networks through an anonymization strategy based on sampled walks. These walks explore the causal structure of network dynamics and generate inductive node representations. A neural network encodes and aggregates the sampled walks to produce the final node embeddings.

GraphMixer (Cong et al., 2023). This model is a simple yet effective architecture for temporal link prediction, composed of three key components: a multi-layer perceptron (MLP)-based link encoder to capture information from temporal links, a node encoder using neighbor mean-pooling, and an MLP-based link classifier for prediction.

DyGFormer (Yu et al., 2023). This method is a Transformer-based model for dynamic graph learning that leverages nodes' historical first-hop interactions to learn meaningful representations. It introduces a neighbor co-occurrence encoding scheme to capture correlations between source and destination nodes based on their interaction histories. To handle longer sequences efficiently, DyGFormer uses a patching technique that divides each historical sequence into smaller segments, allowing the Transformer to process long-term dependencies effectively.

E.3 Dynamic Graph Generation Model Baselines

DG-Gen (Hosseini et al., 2024). This model is a generative framework for CTDG that enables assumption-free and inductive graph generation. Built as an encoder-decoder model, it learns conditional probabilities of temporal interactions to model topological evolution effectively. By relying on temporal embeddings instead of node IDs, DG-Gen supports inductive learning, allowing it to generate graphs with unseen nodes, timestamps, and edge features.

VRDAG (Li et al., 2024). This model is a novel framework for simultaneously generating dynamic graph topology and node attributes in a data-driven manner. It uses a bi-flow graph encoder to preserve directional message flow and attribute information in node embeddings. A GRU-based

module captures temporal dependencies, updating hidden node states across time. VRDAG also parameterizes a flexible prior distribution to sample latent variables at future timesteps, enabling it to model complex dependencies within and across topology and attribute evolution. This variational approach effectively captures both structural and attribute dynamics in evolving graphs.

TIGGER Gupta et al. (2022). This model is a scalable generative model for continuous-time dynamic graphs that overcomes the transductive limitations and node identity leakage of prior methods. Its inductive variant, TIGGER-I, learns a distribution over node embeddings rather than node IDs, enabling generation of graphs with unseen nodes. Notably, TIGGER-I employs a GAN-based module to jointly generate realistic features and graph structure, supporting flexible up- or down-sampling of graph size without requiring one-to-one node mapping—making it suitable for privacy-sensitive and feature-aware dynamic graph synthesis.

E.4 Implementation Details

Implementation Details of TDGG. For TDGG, we set the size of the target expanded DyTAG as 10,000 edges. To ensure fairness in discriminative task comparisons, we adopt a dataset split of 1,000/1,000/8,000 (train/val/test) for DGNNs based on the implementation of DyGLib (Yu et al., 2023; Zhang et al., 2024a), and compare with the last 8,000 edges in the generated DyTAG from GAG-General. For edge classification, we evaluate the models using weighted Precision, Recall, and F1 scores (Zhang et al., 2024a). For the discriminative task comparisons in TDGG, following DTGB (Zhang et al., 2024a), node/edge texts are encoded using the BERT-base-uncased model (Devlin et al., 2019) as initialization for each DGNN. In terms of discriminative task-specific metrics, we use Hit@1 and Hit@10 for node retrieval (link prediction) tasks. Specifically, for the node retrieval task, we set the number of negative samples to 100 for each positive sample. Notably, the Hit@1 metric can be considered a more challenging form of link prediction, as it involves a positive-to-negative sample ratio of 1:100, compared to the easier 1:1 ratio.

Implementation Details of IDGG. For IDGG, we set the size of the target expanded DyTAG as 2,000 edges, due to the additional stage of node generation. To ensure fair comparisons with dynamic graph generation baselines, we use the seed DyTAG with 1,000 edges as the input data for training DG-Gen and VRDAG and adjust the generation ratio in DG-Gen to generate 2,000 edges using its publicly available code (Hosseini et al., 2024), whereas VRDAG does not offer explicit generation size control (Li et al., 2024). Therefore, evaluations are conducted based on the graphs actually generated by VRDAG. As the current dynamic graph generation models' lack of support for DyTAG's textual features, our focus is on comparing the quality of generated graph structures and graph embeddings, rather than the texts. Similar to TDGG, following DTGB (Zhang et al., 2024a), node/edge texts are encoded using the BERT-base-uncased model (Devlin et al., 2019) as initialization for each dynamic graph generation baseline.

Implementation Details of GAG-General on TDGG and IDGG. Following GAG (Ji et al., 2025), we maintain a memory module for each node in DyTAG to record the information of its historical interacted neighbors, effectively incorporating structural and temporal dynamics from the DyTAG. In our proposed GAG-General, the node memory module is implemented via random walks (Sauerwald & Zanetti, 2019). We set the number of random walks for each node as 10, with a random walk length of 10. The node memory is capped at a maximum memory context length of 1,000. Additionally, our generative framework includes an optional memory reflection mechanism, which leverages the LLMs to distill node memories into valuable summaries, akin to the message aggregation process in GNNs (Kipf & Welling, 2016; Peng et al., 2024). During the destination node selection process for the active source node agent, for both TDGG and IDGG, we set the number of recalled candidate destination nodes for each source node to 10. For the inference parameter configuration of benchmarked LLMs, we set the temperature as 0.8, the top-p (nucleus sampling) as 0.9, the repetition penalty as 1.1, and a maximum token limit as 2,000 to balance randomness, diversity, and redundancy suppression. With regarding to the LLM prompt design for TDGG and IDGG, we incorporates detailed node descriptions, edge descriptions, memory reflection instructions, interaction instructions, and node generation instructions in our proposed GAG-General. The example full templates on Sephroa (bipartite) and WeiboTech (non-bipartite) are provided in Section G. As for the LLM-as-Evaluator framework of the textual quality metric, based on the implementation of CharacterBox (Wang et al., 2024a), we use GPT as the LLM backbone for the evaluation framework. The hyperparameters (temperature, top-p, repetition penalty, etc.) of the LLM used in the textual quality metric are kept

identical to those used in generation. Notably, although multiple LLMs serve as backbones in GAG-General for generation, all generated outputs are evaluated by the same GPT-based evaluator with fixed hyperparameters, ensuring robustness and reproducibility of the assessment. The prompts used in the LLM-as-Evaluator framework are provided in Section G.3. Experiments are conducted on NVIDIA A800 with 80 GB memory.

F Supplementary Experimental Results

F.1 Scalability Analysis

Following GAG (Ji et al., 2025), we adopt parallel processing techniques (Gao et al., 2024) in implementing GAG-General to mitigate idle inference time in LLMs during DyTAG generation. Agents are grouped into distinct clusters: each active source node agent collaborates with its corresponding destination node agent, and these groups execute in parallel across CPU cores with multiple ports, enabling an efficient DyTAG generation pipeline.

As shown in Figure 9, we evaluate the time costs of generating a DyTAG with 10,000 edges in TDGG and a DyTAG with 2,000 edges in IDGG, using a 1,000-edge seed graph across diverse datasets with GAG-General applying GPT as the LLM backbone. Under the parallel architecture, the TDGG task achieves an average generation time of 1.61 hours, demonstrating satisfying scalability for DyTAG generation. For IDGG, the additional computational overhead of generating new nodes increases total time, which is jointly determined by both node and edge counts. Across eight datasets, the IDGG task achieves an average generation time of 1.11 hours, indicating moderate scalability with room for further optimization. Furthermore, to rigorously assess the scalability of GAG-General, we conduct experiments on generating large-scale DyTAGs in both TDGG and IDGG tasks. As illustrated in Figure 8, for instance, generating a DyTAG with 100,000 edges in TDGG on Sephora takes 16.3 hours, while generating a DyTAG with 50,000 edges in IDGG requires 35.7 hours. Although the current time costs for large-scale DyTAG generation remain relatively high, our GDGB datasets—comprising multiple million-edge-scale DyTAGs—provide a robust foundation for future research on scalability challenges in DyTAG generation.

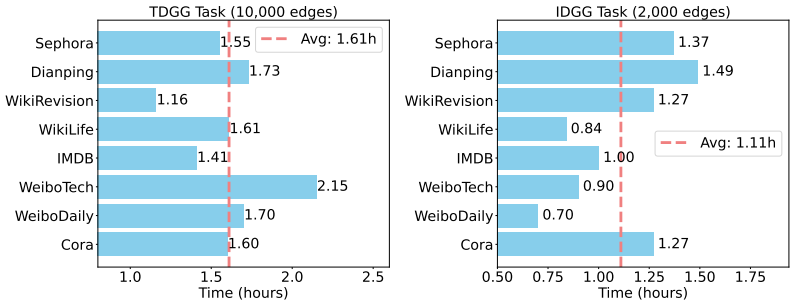

Figure 8: The time required to generate DyTAGs under TDGG (10,000 edges) and IDGG task (2,000 edges) from a 1,000-edge seed graph across GDGB datasets.

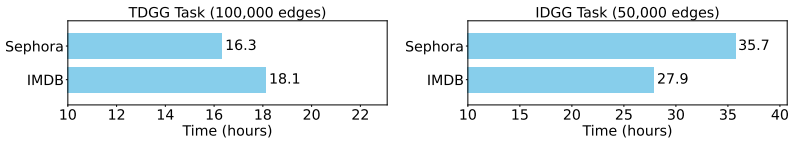

Figure 9: The time required to generate DyTAGs under TDGG (100,000 edges) and IDGG task (50,000 edges) from a 1,000-edge seed graph on Sephora and IMDB.

F.2 Results of TDGG on Graph Structural Metrics

Comprehensive results on Degree MMD, Spectra MMD, D_k, α, and power-law validity under TDGG using three additional LLM backbones (Deepseek, Llama, and Qwen) are available in Table 18, 19, and 20. The results demonstrate that the generated DyTAGs on most datasets exhibit low

Degree/Spectra MMD values and satisfy the power-law validity criterion, confirming high-quality generation in TDGG with GAG-General.

Table 18: The results on Degree MMD, Spectra MMD, D_k, α, and power-law validity under TDGG with Deepseek as the LLM backbone.

Dataset	**Sephora**	**Dianping**	**WikiRevision**	**WikiLife**	**IMDB**	**WeiboTech**	**WeiboDaily**	**Cora**
Degree MMD↓	0.050	0.033	0.209	0.112	0.295	0.110	0.123	0.219
Spectra MMD↓	0.142	0.188	0.149	0.257	0.331	0.177	0.273	0.220
D_k	0.161	0.057	0.040	0.099	0.099	0.076	0.018	0.052
α	2.382	2.808	2.246	2.204	2.065	2.255	2.028	2.562
Power-law Validity	✗	✓	✓	✓	✓	✓	✓	✓

Table 19: The results on Degree MMD, Spectra MMD, D_k, α, and power-law validity under TDGG with Llama as the LLM backbone.

Dataset	**Sephora**	**Dianping**	**WikiRevision**	**WikiLife**	**IMDB**	**WeiboTech**	**WeiboDaily**	**Cora**
Degree MMD↓	0.043	0.062	0.018	0.164	0.285	0.188	0.259	0.109
Spectra MMD↓	0.010	0.399	0.048	0.217	0.316	0.257	0.421	0.126
D_k	0.135	0.044	0.108	0.099	0.125	0.051	0.039	0.047
α	2.976	2.331	2.328	2.204	1.750	2.145	1.979	2.328
Power-law Validity	✓	✓	✓	✓	✗	✓	✗	✓

Table 20: The results on Degree MMD, Spectra MMD, D_k, α, and power-law validity under TDGG with Qwen as the LLM backbone.

Dataset	**Sephora**	**Dianping**	**WikiRevision**	**WikiLife**	**IMDB**	**WeiboTech**	**WeiboDaily**	**Cora**
Degree MMD↓	0.034	0.041	0.098	0.181	0.248	0.222	0.293	0.085
Spectra MMD↓	0.013	0.216	0.141	0.225	0.307	0.307	0.486	0.127
D_k	0.139	0.026	0.055	0.099	0.149	0.020	0.054	0.045
α	2.978	2.177	2.065	2.204	1.723	2.020	1.844	2.367
Power-law Validity	✓	✓	✓	✓	✗	✓	✗	✓

F.3 Results of TDGG on Textual Quality Metrics

Full results on average textual quality scores under TDGG are shown in Table 21. Detailed results for each scoring criterion—*Contextual Fidelity, Personality Depth, Dynamic Adaptability, Immersive Quality, and Content Richness*—rated on a 1–5 scale, are available in Table 22–29. Full descriptions of these criteria are provided in Section D.3. The experimental results demonstrate that node memory or reflection mechanisms consistently improve textual generation quality compared to their absence, highlighting the necessity of these components and the value of integrating structural and textual information in DyTAG generation.

Table 21: The full results on **average** textual quality scores under TDGG. M. and R. denote node memory and reflection mechanism, respectively. The best and the runner-up scores are highlighted in bold and underlined fonts.

	DeepSeek			**Llama**			**Qwen**			**GPT**		
	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.
Sephora	4.09	<u>4.10</u>	**4.37**	4.57	<u>4.58</u>	**4.66**	4.61	<u>4.64</u>	**4.70**	4.69	<u>4.69</u>	**4.77**
Dianping	4.29	<u>4.34</u>	**4.41**	4.13	<u>4.18</u>	**4.46**	<u>4.14</u>	4.13	**4.43**	4.32	<u>4.34</u>	**4.71**
WikiRevision	4.19	<u>4.19</u>	**4.36**	4.21	<u>4.31</u>	**4.48**	4.66	<u>4.73</u>	**4.93**	4.74	<u>4.79</u>	**4.96**
WikiLife	<u>4.30</u>	4.23	**4.34**	<u>4.31</u>	4.25	**4.55**	4.31	<u>4.33</u>	**4.40**	4.44	<u>4.46</u>	**4.59**
IMDB	3.65	<u>3.82</u>	**3.99**	3.97	<u>4.02</u>	**4.32**	4.10	<u>4.18</u>	**4.33**	3.91	<u>4.02</u>	**4.44**
WeiboTech	3.88	<u>3.89</u>	**3.92**	4.49	<u>4.56</u>	**4.86**	<u>4.93</u>	4.91	**4.96**	4.84	<u>4.88</u>	**4.97**
WeiboDaily	3.95	<u>4.22</u>	**4.25**	4.51	<u>4.58</u>	**4.90**	4.88	<u>4.88</u>	**4.98**	4.80	<u>4.92</u>	**4.99**
Cora	4.31	<u>4.40</u>	**4.60**	4.12	<u>4.12</u>	**4.43**	4.21	<u>4.24</u>	**4.42**	3.98	<u>4.10</u>	**4.52**

Table 22: The results on each scoring criterion of generated textual quality under TDGG on **Sephora**.

	DeepSeek			Llama			Qwen			GPT		
	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.
Contextual Fidelity	4.23	4.28	**4.48**	4.63	4.63	**4.71**	4.69	4.72	**4.78**	4.73	4.74	**4.81**
Personality Depth	3.86	3.89	**4.21**	4.49	4.52	**4.65**	4.52	4.57	**4.67**	4.61	4.62	**4.74**
Dynamic Adaptability	4.07	4.02	**4.36**	4.52	4.53	**4.58**	4.55	4.58	**4.62**	4.65	4.64	**4.71**
Immersive Quality	4.11	4.12	**4.38**	4.59	4.59	**4.69**	4.64	4.67	**4.74**	4.71	4.72	**4.79**
Content Richness	4.17	4.17	**4.42**	4.60	4.61	**4.67**	4.66	4.67	**4.70**	4.73	4.72	**4.77**
Average	4.09	4.10	**4.37**	4.57	4.58	**4.66**	4.61	4.64	**4.70**	4.69	4.69	**4.77**

Table 23: The results on each scoring criterion of generated textual quality under TDGG on **Dianping**.

	DeepSeek			Llama			Qwen			GPT		
	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.
Contextual Fidelity	4.42	4.44	**4.54**	4.13	4.21	**4.48**	4.16	4.13	**4.46**	4.33	4.33	**4.71**
Personality Depth	4.13	4.23	**4.28**	4.02	4.06	**4.47**	4.03	3.99	**4.36**	4.19	4.21	**4.66**
Dynamic Adaptability	4.24	4.28	**4.34**	4.09	4.16	**4.38**	4.06	4.07	**4.33**	4.25	4.30	**4.68**
Immersive Quality	4.28	4.35	**4.42**	4.15	4.19	**4.47**	4.16	4.15	**4.45**	4.34	4.34	**4.72**
Content Richness	4.40	4.43	**4.50**	4.23	4.30	**4.51**	4.30	4.30	**4.54**	4.50	4.51	**4.78**
Average	4.29	4.34	**4.41**	4.13	4.18	**4.46**	4.14	4.13	**4.43**	4.32	4.34	**4.71**

Table 24: The results on each scoring criterion of generated textual quality under TDGG on **WikiRevision**.

	DeepSeek			Llama			Qwen			GPT		
	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.
Contextual Fidelity	4.27	4.26	**4.42**	4.64	4.66	**4.70**	4.74	4.77	**4.95**	4.82	4.86	**4.97**
Personality Depth	4.16	4.16	**4.32**	4.12	4.21	**4.33**	4.59	4.62	**4.92**	4.67	4.72	**4.95**
Dynamic Adaptability	4.12	4.11	**4.32**	4.02	4.10	**4.50**	4.55	4.60	**4.89**	4.66	4.72	**4.94**
Immersive Quality	4.21	4.21	**4.38**	3.98	4.06	**4.21**	4.69	4.73	**4.94**	4.77	4.82	**4.97**
Content Richness	4.21	4.20	**4.38**	4.30	4.51	**4.66**	4.72	4.73	**4.94**	4.79	4.81	**4.96**
Average	4.19	4.19	**4.36**	4.21	4.31	**4.48**	4.66	4.73	**4.93**	4.74	4.79	**4.96**

Table 25: The results on each scoring criterion of generated textual quality under TDGG on **WikiLife**.

	DeepSeek			Llama			Qwen			GPT		
	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.
Contextual Fidelity	**4.43**	4.34	**4.43**	4.47	4.44	**4.65**	4.40	4.44	**4.48**	4.54	4.58	**4.66**
Personality Depth	4.29	4.19	**4.31**	4.31	4.27	**4.59**	4.29	4.31	**4.39**	4.41	4.43	**4.58**
Dynamic Adaptability	4.08	4.04	**4.19**	4.10	4.10	**4.35**	4.14	4.13	**4.24**	4.25	4.28	**4.45**
Immersive Quality	**4.39**	4.26	4.38	4.42	4.40	**4.64**	4.39	4.41	**4.47**	4.54	4.54	**4.65**
Content Richness	4.33	4.30	**4.38**	4.26	4.25	**4.54**	4.31	4.34	**4.42**	4.46	4.46	**4.61**
Average	4.30	4.23	**4.34**	4.31	4.25	**4.55**	4.31	4.33	**4.40**	4.44	4.46	**4.59**

Table 26: The results on each scoring criterion of generated textual quality under TDGG on **IMDB**.

	DeepSeek			Llama			Qwen			GPT		
	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.
Contextual Fidelity	3.93	4.15	**4.30**	4.22	4.31	**4.50**	4.42	4.48	**4.55**	4.14	4.28	**4.61**
Personality Depth	3.49	3.62	**3.80**	3.83	3.86	**4.38**	3.95	4.03	**4.30**	3.78	3.88	**4.48**
Dynamic Adaptability	3.45	3.63	**3.78**	3.76	3.80	**4.04**	3.88	3.95	**4.06**	3.69	3.82	**4.12**
Immersive Quality	3.72	3.88	**4.04**	4.07	4.13	**4.47**	4.22	4.29	**4.46**	4.01	4.15	**4.58**
Content Richness	3.64	3.83	**4.03**	3.97	4.00	**4.24**	4.06	4.14	**4.27**	3.92	3.97	**4.38**
Average	3.65	3.82	**3.99**	3.97	4.02	**4.32**	4.10	4.18	**4.33**	3.91	4.02	**4.44**

Table 27: The results on each scoring criterion of generated textual quality under TDGG on **WeiboTech**.

	DeepSeek			Llama			Qwen			GPT		
	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.
Contextual Fidelity	4.09	4.08	**4.12**	4.59	4.66	**4.88**	4.94	4.92	**4.97**	4.86	4.90	**4.99**
Personality Depth	3.96	3.95	**3.97**	4.50	4.54	**4.90**	4.91	4.89	**4.95**	4.83	4.86	**4.97**
Dynamic Adaptability	3.62	3.63	**3.70**	4.40	4.50	**4.82**	4.93	4.91	**4.95**	4.83	4.88	**4.97**
Immersive Quality	3.95	3.95	**4.00**	4.55	4.62	**4.88**	4.93	4.92	**4.96**	4.85	4.88	**4.98**
Content Richness	3.79	3.83	**3.83**	4.43	4.50	**4.82**	4.93	4.91	**4.96**	4.82	4.88	**4.96**
Average	3.88	3.89	**3.92**	4.49	4.56	**4.86**	4.93	4.91	**4.96**	4.84	4.88	**4.97**

Table 28: The results on each scoring criterion of generated textual quality under TDGG on **Weibo-Daily**.

	DeepSeek			Llama			Qwen			GPT		
	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.
Contextual Fidelity	4.16	4.38	**4.40**	4.61	4.66	**4.92**	4.90	4.89	**4.98**	4.83	4.93	**5.00**
Personality Depth	4.05	4.27	**4.33**	4.53	4.58	**4.93**	4.87	4.86	**4.98**	4.79	4.90	**4.99**
Dynamic Adaptability	3.65	**3.98**	3.97	4.44	4.51	**4.87**	4.88	4.87	**4.97**	4.78	4.91	**4.97**
Immersive Quality	4.02	4.30	**4.37**	4.58	4.63	**4.92**	4.90	4.89	**4.98**	4.83	4.92	**5.00**
Content Richness	3.90	**4.18**	**4.18**	4.41	4.50	**4.86**	4.88	4.87	**4.97**	4.77	4.91	**4.97**
Average	3.95	4.22	**4.25**	4.51	4.58	**4.90**	4.88	4.88	**4.98**	4.80	4.92	**4.99**

Table 29: The results on each scoring criterion of generated textual quality under TDGG on **Cora**.

	DeepSeek			Llama			Qwen			GPT		
	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.
Contextual Fidelity	4.40	4.49	**4.69**	4.21	4.19	**4.51**	4.28	4.32	**4.49**	4.08	4.18	**4.59**
Personality Depth	4.25	4.32	**4.53**	4.15	4.13	**4.45**	4.21	4.25	**4.43**	3.99	4.11	**4.52**
Dynamic Adaptability	4.23	4.28	**4.47**	3.88	3.93	**4.23**	4.04	4.01	**4.25**	3.73	3.89	**4.33**
Immersive Quality	4.33	4.42	**4.64**	4.20	4.18	**4.49**	4.26	4.32	**4.48**	4.06	4.17	**4.58**
Content Richness	4.35	4.49	**4.69**	4.18	4.17	**4.49**	4.25	4.30	**4.46**	4.05	4.16	**4.59**
Average	4.31	4.40	**4.60**	4.12	4.12	**4.43**	4.21	4.24	**4.42**	3.98	4.10	**4.52**

F.4 Results of TDGG on Graph Embedding Metrics

Full results on the graph embedding metric under TDGG are shown in Table 30. Similar to the findings on textual quality, the use of node memory and reflection mechanisms significantly enhances generation quality, underscoring the critical role of integrating structural and textual information in DyTAG generation.

Table 30: The results on the graph embedding metric under TDGG.

	DeepSeek			Llama			Qwen			GPT		
	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.
Sephora	0.715	**0.758**	0.679	0.675	**0.681**	0.648	0.579	0.630	**0.740**	0.588	**0.671**	0.654
Dianping	0.637	0.666	**0.722**	0.368	0.386	**0.408**	0.408	0.390	**0.413**	0.351	0.368	**0.369**
WikiRevision	0.609	**0.641**	0.624	0.567	0.582	**0.636**	0.458	0.429	**0.464**	0.438	0.422	**0.454**
WikiLife	**0.599**	0.554	0.588	0.429	0.462	**0.477**	0.443	0.478	**0.490**	0.459	0.453	**0.463**
IMDB	0.534	**0.586**	0.532	0.396	0.482	**0.533**	0.487	0.493	**0.519**	0.432	0.420	**0.435**
WeiboTech	0.415	**0.577**	0.501	0.335	0.417	**0.420**	0.296	0.353	**0.364**	0.327	0.488	**0.698**
WeiboDaily	0.471	0.424	**0.494**	0.403	**0.524**	0.494	**0.437**	0.377	0.389	0.681	0.689	**0.694**
Cora	**0.649**	0.556	0.581	**0.689**	0.499	0.483	0.485	0.450	**0.536**	**0.511**	0.446	0.465

F.5 Results of TDGG on Direct Usability in Downstream Tasks

To evaluate the direct usability of DyTAGs generated under the TDGG task, we assess their effectiveness as substitutes for original graphs in standard transductive downstream tasks (edge classification and link prediction), following the conventional experimental settings in dynamic graph learning (Yu et al., 2023; Zhang et al., 2024a; Peng et al., 2025). We employ four established dynamic graph neural network (DGNN) models—TGN (Rossi et al., 2020), CAWN (Wang et al., 2021a), GraphMixer (Cong et al., 2023), and DyGFormer (Yu et al., 2023)—to perform edge classification

and link prediction on both the generated (Gen.) and original (Ori.) graphs across four datasets: Sephora, WikiLife, Cora, and WeiboTech.

For edge classification, the results in Table 31 show consistently small performance gaps between the generated and original graphs. The absolute differences in Precision ($|\Delta P|$), Recall ($|\Delta R|$), and F1-score ($|\Delta F|$) are uniformly below 0.04 across all models and datasets, with most values under 0.02. This indicates that the generated graphs preserve sufficient structural and textual fidelity to support accurate classification. Similarly, in link prediction (transductive setting in dynamic graph learning), the Average Precision (AP) differences ($|\Delta AP|$) between using generated and original graphs are minimal, as summarized in Table 32. The maximum observed $|\Delta AP|$ is 0.025, further confirming that the generated DyTAGs accurately capture the temporal evolution patterns of the original graphs.These results demonstrate that DyTAGs generated by our framework in the TDGG task are of high quality and can be directly deployed in downstream applications with negligible performance degradation.

Table 31: The result of edge classification performance on generated graphs from TDGG and original graphs.

Model	Dataset	Type	Precision	Recall	F1	$\lvert\Delta P\rvert \downarrow$	$\lvert\Delta R\rvert \downarrow$	$\lvert\Delta F\rvert \downarrow$
TGN	Sephora	Gen.	0.556 ± 0.009	0.490 ± 0.012	0.523 ± 0.004	0.022	0.004	0.011
		Ori.	0.534 ± 0.009	0.494 ± 0.036	0.512 ± 0.026			
	WikiLife	Gen.	0.176 ± 0.024	0.299 ± 0.017	0.191 ± 0.033	0.033	0.017	0.030
		Ori.	0.209 ± 0.016	0.282 ± 0.035	0.221 ± 0.010			
	Cora	Gen.	0.219 ± 0.016	0.241 ± 0.004	0.206 ± 0.027	0.014	0.009	0.016
		Ori.	0.233 ± 0.021	0.232 ± 0.017	0.190 ± 0.033			
	WeiboTech	Gen.	0.781 ± 0.008	0.790 ± 0.019	0.782 ± 0.012	0.004	0.019	0.026
		Ori.	0.777 ± 0.009	0.771 ± 0.024	0.756 ± 0.012			
CAWN	Sephora	Gen.	0.526 ± 0.017	0.690 ± 0.016	0.563 ± 0.014	0.005	0.006	0.007
		Ori.	0.531 ± 0.029	0.696 ± 0.003	0.556 ± 0.023			
	WikiLife	Gen.	0.190 ± 0.015	0.292 ± 0.016	0.206 ± 0.023	0.009	0.008	0.022
		Ori.	0.199 ± 0.003	0.284 ± 0.033	0.228 ± 0.026			
	Cora	Gen.	0.212 ± 0.014	0.278 ± 0.038	0.188 ± 0.032	0.010	0.009	0.016
		Ori.	0.222 ± 0.020	0.269 ± 0.031	0.204 ± 0.013			
	WeiboTech	Gen.	0.783 ± 0.018	0.787 ± 0.017	0.784 ± 0.019	0.013	0.002	0.017
		Ori.	0.770 ± 0.020	0.789 ± 0.020	0.767 ± 0.031			
GraphMixer	Sephora	Gen.	0.576 ± 0.024	0.690 ± 0.003	0.563 ± 0.018	0.023	0.005	0.012
		Ori.	0.553 ± 0.020	0.685 ± 0.017	0.575 ± 0.033			
	WikiLife	Gen.	0.227 ± 0.015	0.315 ± 0.005	0.218 ± 0.021	0.014	0.014	0.009
		Ori.	0.213 ± 0.015	0.301 ± 0.013	0.209 ± 0.008			
	Cora	Gen.	0.222 ± 0.014	0.268 ± 0.015	0.222 ± 0.021	0.013	0.008	0.002
		Ori.	0.235 ± 0.009	0.260 ± 0.033	0.224 ± 0.012			
	WeiboTech	Gen.	0.776 ± 0.018	0.787 ± 0.036	0.775 ± 0.026	0.005	0.007	0.002
		Ori.	0.771 ± 0.009	0.780 ± 0.007	0.773 ± 0.006			
DyGFormer	Sephora	Gen.	0.476 ± 0.011	0.690 ± 0.004	0.563 ± 0.029	0.007	0.008	0.003
		Ori.	0.469 ± 0.025	0.698 ± 0.007	0.560 ± 0.017			
	WikiLife	Gen.	0.242 ± 0.024	0.311 ± 0.014	0.194 ± 0.029	0.003	0.000	0.002
		Ori.	0.239 ± 0.029	0.311 ± 0.020	0.196 ± 0.017			
	Cora	Gen.	0.209 ± 0.018	0.274 ± 0.039	0.206 ± 0.023	0.010	0.024	0.007
		Ori.	0.219 ± 0.021	0.250 ± 0.024	0.213 ± 0.030			
	WeiboTech	Gen.	0.806 ± 0.010	0.803 ± 0.024	0.780 ± 0.014	0.020	0.013	0.013
		Ori.	0.786 ± 0.007	0.790 ± 0.006	0.767 ± 0.006			

F.6 Results of Node Retrieval

Comprehensive results on the node retrieval task, including Hit@1 and Hit@10 metrics, are presented in Table 33. Detailed results for all LLM backbones and DGNN baselines are provided in Table 34–36. The experimental findings reveal that DGNNs maintain a significant advantage in node retrieval performance. However, with reduced training data availability, the GAG-General model achieves superior results than DGNNs on the Sephora and IMDB datasets under a generative paradigm. This highlights the strong dependency of DGNNs on large-scale training data and their limited generalization capability in low-data scenarios.

Table 32: The results of link prediction performance (transductive) on generated graphs from TDGG and original graphs.

Model	Dataset	Type	Average Precision	$\lvert\Delta AP\rvert \downarrow$
TGN	Sephora	Gen.	0.663 ± 0.025	0.010
		Ori.	0.673 ± 0.016	
	WikiLife	Gen.	0.807 ± 0.017	0.017
		Ori.	0.790 ± 0.023	
	Cora	Gen.	0.621 ± 0.022	0.025
		Ori.	0.596 ± 0.012	
	WeiboTech	Gen.	0.859 ± 0.009	0.021
		Ori.	0.838 ± 0.027	
CAWN	Sephora	Gen.	0.731 ± 0.017	0.001
		Ori.	0.730 ± 0.015	
	WikiLife	Gen.	0.831 ± 0.003	0.019
		Ori.	0.850 ± 0.023	
	Cora	Gen.	0.616 ± 0.011	0.024
		Ori.	0.592 ± 0.017	
	WeiboTech	Gen.	0.862 ± 0.020	0.013
		Ori.	0.849 ± 0.030	
GraphMixer	Sephora	Gen.	0.729 ± 0.019	0.018
		Ori.	0.711 ± 0.010	
	WikiLife	Gen.	0.813 ± 0.002	0.018
		Ori.	0.795 ± 0.030	
	Cora	Gen.	0.626 ± 0.030	0.002
		Ori.	0.624 ± 0.021	
	WeiboTech	Gen.	0.868 ± 0.028	0.010
		Ori.	0.858 ± 0.011	
DyGFormer	Sephora	Gen.	0.631 ± 0.007	0.013
		Ori.	0.644 ± 0.027	
	WikiLife	Gen.	0.800 ± 0.022	0.004
		Ori.	0.796 ± 0.012	
	Cora	Gen.	0.824 ± 0.020	0.004
		Ori.	0.828 ± 0.027	
	WeiboTech	Gen.	0.800 ± 0.020	0.001
		Ori.	0.799 ± 0.009	

Table 33: The results on Hit@1 and Hit@10 under the node retrieval task. Ours correspond to the best results of our proposed GAG-General among four LLM backbones.

	Hit@1					**Hit@10**				
	Ours	TGN	CAWN	GraphMixer	DyGFormer	Ours	TGN	CAWN	GraphMixer	DyGFormer
Sephora	**0.285**	0.045	0.023	0.065	0.028	**0.386**	0.259	0.206	0.309	0.184
Dianping	0.176	0.040	0.033	0.036	**0.534**	0.178	0.188	0.193	0.207	**0.585**
WikiRevision	0.105	0.031	0.067	0.099	**0.439**	0.230	0.164	0.287	0.272	**0.507**
WikiLife	0.139	0.119	0.116	**0.175**	0.160	0.315	0.388	0.379	**0.511**	0.264
IMDB	**0.142**	0.015	0.009	0.005	0.008	**0.159**	0.123	0.110	0.105	0.111
WeiboTech	0.051	0.183	0.128	0.169	**0.672**	0.056	0.371	0.448	0.378	**0.678**
WeiboDaily	0.091	0.138	0.267	**0.445**	0.154	0.100	0.295	0.615	**0.663**	0.438
Cora	0.122	0.025	0.016	0.064	**0.465**	0.237	0.193	0.175	0.299	**0.321**

Table 34: The results on Hit@1 under the node retrieval task with our framework.

	DeepSeek			**Llama**			**Qwen**			**GPT**		
	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.
Sephora	0.002	**0.285**	0.048	0.000	0.039	**0.073**	0.019	0.046	**0.064**	0.016	0.066	**0.077**
Dianping	0.000	**0.176**	0.094	0.001	0.002	**0.003**	0.001	0.008	**0.041**	0.001	0.002	**0.013**
WikiRevision	0.003	**0.013**	0.005	0.084	0.092	**0.105**	**0.013**	0.012	0.009	0.013	0.013	**0.015**
WikiLife	0.029	**0.135**	0.034	0.086	0.094	**0.097**	0.067	0.073	**0.075**	0.107	0.126	**0.139**
IMDB	0.000	**0.142**	0.020	0.001	**0.022**	0.005	0.043	0.043	**0.053**	0.001	0.017	**0.019**
WeiboTech	0.002	**0.051**	0.021	0.001	**0.002**	0.001	0.004	0.004	**0.005**	0.000	0.007	**0.008**
WeiboDaily	0.009	**0.091**	0.040	0.003	**0.006**	0.003	0.009	0.012	**0.013**	0.002	**0.014**	0.004
Cora	0.002	0.069	**0.082**	0.000	0.024	**0.033**	0.110	0.108	**0.122**	0.000	0.045	**0.115**

Table 35: The results on Hit@10 under the node retrieval task with our framework.

	DeepSeek			Llama			Qwen			GPT		
	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.
Sephora	0.064	**0.386**	0.169	0.090	0.164	**0.314**	0.085	0.191	**0.278**	0.071	0.291	**0.369**
Dianping	0.004	**0.178**	0.108	0.003	0.004	**0.014**	0.002	0.015	**0.061**	0.004	0.009	**0.024**
WikiRevision	0.031	**0.041**	0.031	0.143	0.177	**0.230**	**0.067**	0.061	0.047	0.074	0.065	**0.077**
WikiLife	0.096	**0.205**	0.093	0.216	0.255	**0.260**	0.145	0.156	**0.178**	0.238	0.275	**0.315**
IMDB	0.035	**0.159**	0.063	0.015	**0.033**	0.021	0.055	0.054	**0.064**	0.012	0.025	**0.029**
WeiboTech	0.006	**0.056**	0.028	**0.006**	0.005	0.005	0.007	0.008	**0.008**	0.005	0.011	**0.026**
WeiboDaily	0.021	**0.100**	0.049	0.015	**0.024**	0.012	**0.022**	0.021	0.019	**0.036**	0.022	0.016
Cora	0.064	0.128	**0.160**	0.111	0.135	**0.174**	0.205	0.209	**0.226**	0.128	0.172	**0.237**

Table 36: The results on Hit@1 and Hit@10 under the node retrieval task with DGNN baselines.

	Hit@1					Hit@10				
	JODIE	TGN	CAWN	GraphMixer	DyGFormer	JODIE	TGN	CAWN	GraphMixer	DyGFormer
Sephora	0.018	0.045	0.023	**0.065**	0.028	0.120	0.259	0.206	**0.309**	0.184
Dianping	0.419	0.040	0.033	0.036	**0.534**	0.427	0.188	0.193	0.207	**0.585**
WikiRevision	0.284	0.031	0.067	0.099	**0.439**	0.375	0.164	0.287	0.272	**0.507**
WikiLife	0.049	0.119	0.116	**0.175**	0.160	0.410	0.388	0.379	**0.511**	0.264
IMDB	0.013	**0.015**	0.009	0.005	0.008	0.111	**0.123**	0.110	0.105	0.111
WeiboTech	0.632	0.183	0.128	0.169	**0.672**	0.650	0.371	0.448	0.378	**0.678**
WeiboDaily	0.027	0.138	0.267	**0.445**	0.154	0.440	0.295	0.615	**0.663**	0.438
Cora	0.017	0.025	0.016	0.064	**0.465**	0.129	0.193	0.175	0.299	**0.321**

F.7 Results of Edge Classification

Full results on Precision, Recall, and F1 metric under the edge classification task are available in Table 37. Detailed results on all LLM backbones and DGNN baselines are available in Table 38 and 39. The experimental findings demonstrate that our GAG-General model achieves excellent performance on this discriminative task under the generative paradigm. This success stems from the strong correlation between edge labels and the semantic information encoded in both node texts and generated edge texts. For instance, in e-commerce recommendation scenarios, if a negative review is generated as the edge text, the corresponding edge label (e.g., low rating) aligns closely with this sentiment. In contrast, existing DGNNs struggle to effectively capture and utilize the semantic richness of node/edge texts in DyTAG datasets, resulting in significant performance gaps on edge classification tasks.

Table 37: The results on Precision, Recall, and F1 metric under the edge classification task. Ours correspond to the best results of our proposed GAG-General among four LLM backbones.

Datasets	Models	Ours	JODIE	TGN	CAWN	GraphMixer	DyGFormer
Sephora	Precision	**0.798**	0.463	0.452	0.442	0.443	0.471
	Recall	**0.790**	0.661	0.672	0.661	0.684	0.687
	F1	**0.790**	0.555	0.529	0.532	0.546	0.559
Dianping	Precision	**0.572**	0.359	0.345	0.409	0.325	0.217
	Recall	**0.533**	0.437	0.436	0.459	0.466	0.466
	F1	**0.522**	0.341	0.279	0.330	0.298	0.296
WikiRevision	Precision	**0.804**	0.291	0.474	0.739	0.756	0.401
	Recall	**0.829**	0.378	0.387	0.444	0.382	0.431
	F1	**0.782**	0.209	0.232	0.341	0.217	0.303
WikiLife	Precision	**0.562**	0.151	0.171	0.199	0.194	0.093
	Recall	**0.451**	0.255	0.288	0.290	0.291	0.284
	F1	**0.484**	0.163	0.188	0.209	0.172	0.128
IMDB	Precision	**0.698**	0.423	0.442	0.470	0.473	0.463
	Recall	0.666	0.680	0.672	0.677	0.650	**0.682**
	F1	**0.672**	0.531	0.542	0.551	0.546	0.533
WeiboTech	Precision	**0.726**	0.615	0.479	0.479	0.503	0.533
	Recall	**0.779**	0.687	0.692	0.690	0.692	0.692
	F1	**0.740**	0.587	0.566	0.566	0.566	0.568
WeiboDaily	Precision	**0.980**	0.853	0.853	0.867	0.853	0.898
	Recall	**0.990**	0.924	0.924	0.922	0.924	0.925
	F1	**0.985**	0.887	0.887	0.891	0.887	0.890
Cora	Precision	**0.566**	0.186	0.149	0.149	0.153	0.099
	Recall	**0.550**	0.273	0.280	0.279	0.279	0.280
	F1	**0.539**	0.152	0.148	0.134	0.131	0.126

Table 38: The results on Precision, Recall, and F1 metric under the edge classification task on all LLM backbones.

		DeepSeek			Llama			Qwen			GPT		
		w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.
Sephora	Precision	0.714	0.744	**0.798**	0.631	**0.638**	0.634	0.550	**0.571**	0.546	0.542	**0.572**	**0.572**
	Recall	0.682	0.742	**0.790**	0.628	0.628	**0.637**	0.386	0.397	**0.400**	**0.593**	0.586	0.591
	F1	0.686	0.734	**0.790**	0.620	0.620	**0.625**	0.405	0.416	**0.420**	0.566	0.564	**0.570**
Dianping	Precision	0.541	**0.572**	0.531	0.427	**0.462**	0.455	0.410	0.422	**0.425**	**0.409**	0.396	0.393
	Recall	0.529	0.330	**0.533**	0.346	0.352	**0.380**	0.408	**0.432**	0.422	**0.374**	0.343	0.354
	F1	0.521	0.411	**0.522**	0.260	0.271	**0.291**	0.384	**0.411**	0.397	**0.311**	0.286	0.288
WikiRevision	Precision	**0.750**	0.715	0.736	0.712	0.717	**0.804**	0.587	0.605	**0.625**	0.681	0.678	**0.689**
	Recall	**0.741**	0.709	0.727	0.739	0.731	**0.829**	0.489	0.496	**0.505**	0.497	0.493	**0.503**
	F1	**0.727**	0.681	0.712	0.695	0.686	**0.782**	0.429	0.437	**0.444**	0.410	0.404	**0.420**
WikiLife	Precision	0.411	0.414	**0.438**	0.537	**0.561**	0.537	0.393	**0.416**	0.414	0.544	0.549	**0.562**
	Recall	0.387	0.386	**0.415**	0.240	**0.254**	0.243	0.246	0.249	**0.253**	0.442	0.441	**0.451**
	F1	0.381	0.380	**0.409**	0.292	**0.306**	0.297	0.210	**0.219**	0.215	0.472	0.475	**0.484**
IMDB	Precision	0.640	0.666	**0.698**	0.634	**0.686**	0.647	**0.683**	0.681	0.647	0.550	0.617	**0.606**
	Recall	0.573	0.615	**0.657**	0.431	0.529	**0.566**	**0.666**	0.664	0.665	0.427	0.526	**0.576**
	F1	0.598	0.633	**0.672**	0.497	0.572	**0.595**	**0.626**	0.620	0.621	0.466	0.547	**0.577**
WeiboTech	Precision	0.676	0.675	**0.688**	0.629	**0.701**	0.631	0.644	**0.693**	0.642	0.626	0.636	**0.726**
	Recall	**0.735**	0.728	0.733	0.697	**0.745**	0.694	**0.704**	**0.704**	0.703	0.703	0.704	**0.779**
	F1	0.685	0.679	**0.693**	0.589	**0.646**	0.574	0.583	0.583	**0.595**	0.590	0.584	**0.740**
WeiboDaily	Precision	0.918	0.913	**0.923**	0.902	0.925	**0.927**	0.888	**0.935**	0.906	0.980	0.902	**0.980**
	Recall	0.918	0.916	**0.931**	0.924	**0.948**	0.921	0.930	0.931	**0.931**	0.990	0.931	**0.990**
	F1	0.918	0.914	**0.927**	0.891	**0.923**	0.883	0.896	0.897	**0.898**	0.985	0.899	**0.985**
Cora	Precision	**0.566**	0.493	0.542	**0.528**	0.495	0.460	0.427	0.445	**0.456**	**0.480**	0.377	0.410
	Recall	**0.550**	0.474	0.520	**0.473**	0.431	0.414	0.414	0.427	**0.436**	**0.421**	0.377	0.386
	F1	**0.539**	0.454	0.508	**0.436**	0.385	0.367	0.382	0.398	**0.403**	**0.370**	0.335	0.334

Table 39: The results on Precision, Recall, and F1 metric under the edge classification task on all DGNN baselines.

Datasets	Models	JODIE	TGN	CAWN	GraphMixer	DyGFormer
Sephora	Precision	0.463	0.452	0.442	0.443	**0.471**
	Recall	0.661	0.672	0.661	0.684	**0.687**
	F1	0.555	0.529	0.532	0.546	**0.559**
Dianping	Precision	0.359	0.345	**0.409**	0.325	0.217
	Recall	0.437	0.436	0.459	0.466	**0.466**
	F1	**0.341**	0.279	0.330	0.298	0.296
WikiRevision	Precision	0.291	0.474	0.739	**0.756**	0.401
	Recall	0.378	0.387	**0.444**	0.382	0.431
	F1	0.209	0.232	**0.341**	0.217	0.303
WikiLife	Precision	0.151	0.171	**0.199**	0.194	0.093
	Recall	0.255	0.288	0.290	**0.291**	0.284
	F1	0.163	0.188	**0.209**	0.172	0.128
IMDB	Precision	0.423	0.442	0.470	**0.473**	0.463
	Recall	0.680	0.672	0.677	0.650	**0.682**
	F1	0.531	0.542	**0.551**	0.546	0.533
WeiboTech	Precision	**0.615**	0.479	0.479	0.503	0.533
	Recall	0.687	0.692	0.690	0.692	**0.692**
	F1	0.587	0.566	0.566	0.566	**0.568**
WeiboDaily	Precision	0.853	0.853	0.867	0.853	**0.898**
	Recall	0.924	0.924	0.922	0.924	**0.925**
	F1	0.887	0.887	**0.891**	0.887	0.890
Cora	Precision	**0.186**	0.149	0.149	0.153	0.099
	Recall	0.273	**0.280**	0.279	0.279	**0.280**
	F1	**0.152**	0.148	0.134	0.131	0.126

F.8 Results of IDGG on Graph Structural Metrics

Full results on Degree MMD, Spectra MMD, D_k, α, and power-law validity under IDGG using other three LLM backbones are available in Table 40, 41, and 42. Compared to the TDGG task, DyTAGs generated under IDGG exhibit slightly higher Degree/Spectra MMD values, primarily due to the introduction of new node generation, which increases the complexity of the modeling process. Nevertheless, over half of the generated DyTAG graphs still satisfy the power-law distribution criterion, demonstrating the robustness of the GAG-General framework in maintaining structural quality even under more challenging generation conditions.

Table 40: The results on Degree MMD, Spectra MMD, D_k, α, and power-law validity under IDGG with Deepseek as the LLM backbone.

Dataset	**Sephora**	**Dianping**	**WikiRevision**	**WikiLife**	**IMDB**	**WeiboTech**	**WeiboDaily**	**Cora**
Degree MMD	0.370	0.150	0.191	0.087	0.329	0.212	0.219	0.073
Spectra MMD	0.250	0.351	0.158	0.206	0.440	0.172	0.397	0.181
D_k	0.099	0.056	0.029	0.099	0.238	0.059	0.131	0.084
α	2.135	3.010	2.185	2.204	1.805	1.953	1.839	2.337
Power-law Validity	✓	✗	✓	✓	✗	✗	✗	✓

Table 41: The results on Degree MMD, Spectra MMD, D_k, α, and power-law validity under IDGG with Llama as the LLM backbone.

Dataset	**Sephora**	**Dianping**	**WikiRevision**	**WikiLife**	**IMDB**	**WeiboTech**	**WeiboDaily**	**Cora**
Degree MMD	0.411	0.236	0.102	0.078	0.145	0.231	0.246	0.123
Spectra MMD	0.243	0.413	0.105	0.223	0.379	0.215	0.385	0.232
D_k	0.098	0.067	0.103	0.099	0.226	0.029	0.088	0.130
α	2.077	2.480	2.300	2.204	1.972	2.099	1.968	2.174
Power-law Validity	✓	✓	✓	✓	✗	✓	✗	✓

Table 42: The results on Degree MMD, Spectra MMD, D_k, α, and power-law validity under IDGG with Qwen as the LLM backbone.

Dataset	**Sephora**	**Dianping**	**WikiRevision**	**WikiLife**	**IMDB**	**WeiboTech**	**WeiboDaily**	**Cora**
Degree MMD	0.512	0.321	0.286	0.083	0.303	0.237	0.295	0.095
Spectra MMD	0.234	0.436	0.193	0.218	0.415	0.189	0.461	0.192
D_k	0.174	0.076	0.039	0.099	0.207	0.050	0.152	0.093
α	2.015	2.497	2.114	2.204	1.783	1.900	1.757	2.247
Power-law Validity	✗	✓	✓	✓	✗	✗	✗	✓

F.9 Results of IDGG on Textual Quality Metrics

Full results on average textual quality scores under IDGG are shown in Table 43. Detailed results on each scoring criterion are available in Table 44-51. Similar to the findings under the TDGG task, the experimental results demonstrate that incorporating node memory or reflection mechanisms significantly enhances the textual quality of generated DyTAGs, highlighting the importance of integrating structural and textual information in DyTAG generation.

Table 43: The results on **average** textual quality scores under IDGG. M. and R. denote node memory and memory reflection mechanism, respectively.

	DeepSeek			**Llama**			**Qwen**			**GPT**		
	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.
Sephora	4.63	4.73	**4.77**	4.58	4.65	**4.74**	4.58	4.68	**4.78**	4.58	4.77	**4.87**
Dianping	4.29	4.44	**4.65**	4.29	4.51	**4.87**	4.04	4.50	**4.68**	4.56	4.71	**4.86**
WikiRevision	4.21	4.46	**4.63**	3.98	4.21	**4.30**	4.37	4.53	**4.71**	4.39	4.54	**4.71**
WikiLife	4.15	4.30	**4.44**	4.12	4.26	**4.38**	4.11	4.26	**4.29**	4.28	4.39	**4.47**
IMDB	4.13	4.28	**4.39**	4.22	4.31	**4.43**	4.23	4.36	**4.51**	4.19	4.29	**4.49**
WeiboTech	4.60	4.75	**4.85**	3.94	4.00	**4.75**	4.56	4.74	**4.93**	4.60	4.71	**4.93**
WeiboDaily	4.68	4.81	**4.92**	4.32	4.43	**4.82**	4.75	4.87	**4.98**	4.74	4.83	**4.99**
Cora	4.27	4.44	**4.56**	4.26	4.36	**4.54**	4.29	4.44	**4.53**	4.27	4.36	**4.57**

Table 44: The results on each scoring criterion of generated textual quality under IDGG on **Sephora**.

	DeepSeek			**Llama**			**Qwen**			**GPT**		
	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.
Contextual Fidelity	4.63	4.80	**4.87**	4.64	4.73	**4.83**	4.51	4.78	**4.92**	4.77	4.84	**4.93**
Personality Depth	4.53	4.56	**4.61**	4.55	4.52	**4.63**	4.42	4.46	**4.60**	4.20	4.61	**4.78**
Dynamic Adaptability	4.60	4.67	**4.70**	4.41	4.59	**4.66**	4.55	4.61	**4.69**	4.54	4.71	**4.82**
Immersive Quality	4.71	4.83	**4.87**	4.71	4.73	**4.84**	4.77	4.81	**4.91**	4.69	4.86	**4.95**
Content Richness	4.66	4.79	**4.81**	4.59	4.67	**4.74**	4.66	4.74	**4.79**	4.72	4.80	**4.86**
Average	4.63	4.73	**4.77**	4.58	4.65	**4.74**	4.58	4.68	**4.78**	4.58	4.77	**4.87**

Table 45: The results on each scoring criterion of generated textual quality under IDGG on **Dianping**.

	DeepSeek			Llama			Qwen			GPT		
	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.
Contextual Fidelity	4.40	4.50	**4.85**	4.52	4.64	**4.95**	3.89	4.57	**4.77**	4.56	4.77	**4.93**
Personality Depth	4.03	4.21	**4.32**	4.33	4.21	**4.72**	4.01	4.21	**4.45**	4.33	4.42	**4.69**
Dynamic Adaptability	4.21	4.37	**4.50**	4.28	4.39	**4.82**	3.98	4.43	**4.61**	4.50	4.69	**4.80**
Immersive Quality	4.47	4.57	**4.81**	4.11	4.67	**4.95**	4.22	4.60	**4.79**	4.70	4.82	**4.94**
Content Richness	4.33	4.57	**4.79**	4.20	4.65	**4.93**	4.10	4.68	**4.80**	4.69	4.83	**4.92**
Average	4.29	4.44	**4.65**	4.29	4.51	**4.87**	4.04	4.50	**4.68**	4.56	4.71	**4.86**

Table 46: The results on each scoring criterion of generated textual quality under IDGG on **WikiRevision**.

	DeepSeek			Llama			Qwen			GPT		
	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.
Contextual Fidelity	4.88	4.86	**4.94**	4.22	4.76	**4.82**	4.82	4.90	**4.98**	4.82	4.92	**4.98**
Personality Depth	4.01	3.81	**4.09**	3.68	3.65	**3.75**	3.81	3.90	**4.17**	3.89	3.91	**4.20**
Dynamic Adaptability	3.92	4.28	**4.44**	3.76	3.91	**4.01**	4.21	4.32	**4.56**	4.10	4.33	**4.55**
Immersive Quality	4.23	4.81	**4.91**	4.20	4.60	**4.69**	4.62	4.86	**4.96**	4.83	4.89	**4.96**
Content Richness	4.01	4.53	**4.76**	4.05	4.14	**4.21**	4.40	4.66	**4.86**	4.29	4.64	**4.86**
Average	4.21	4.46	**4.63**	3.98	4.21	**4.30**	4.37	4.53	**4.71**	4.39	4.54	**4.71**

Table 47: The results on each scoring criterion of generated textual quality under IDGG on **WikiLife**.

	DeepSeek			Llama			Qwen			GPT		
	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.
Contextual Fidelity	4.52	4.67	**4.83**	4.53	**4.63**	**4.63**	4.30	**4.58**	4.57	4.62	4.74	**4.79**
Personality Depth	3.63	3.77	**3.90**	3.62	3.75	**4.07**	3.69	3.73	**3.81**	3.73	3.85	**4.00**
Dynamic Adaptability	4.04	4.03	**4.11**	3.92	4.02	**4.11**	3.92	4.02	**4.06**	4.02	4.10	**4.17**
Immersive Quality	4.34	4.64	**4.81**	4.42	4.60	**4.62**	4.32	4.56	**4.57**	4.63	4.73	**4.75**
Content Richness	4.24	4.38	**4.57**	4.13	4.28	**4.46**	4.33	4.39	**4.43**	4.42	4.55	**4.66**
Average	4.15	4.30	**4.44**	4.12	4.26	**4.38**	4.11	4.26	**4.29**	4.28	4.39	**4.47**

Table 48: The results on each scoring criterion of generated textual quality under IDGG on **IMDB**.

	DeepSeek			Llama			Qwen			GPT		
	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.
Contextual Fidelity	4.53	4.71	**4.88**	4.76	4.83	**4.84**	4.82	4.80	**4.88**	4.65	4.70	**4.78**
Personality Depth	3.72	3.81	**3.84**	3.67	3.76	**4.03**	3.78	3.87	**4.17**	3.72	3.82	**4.19**
Dynamic Adaptability	3.89	4.06	**4.10**	4.01	4.04	**4.16**	4.02	4.17	**4.25**	3.90	4.08	**4.23**
Immersive Quality	4.30	4.65	**4.82**	4.58	4.72	**4.80**	4.33	4.71	**4.85**	4.56	4.65	**4.77**
Content Richness	4.20	4.18	**4.29**	4.10	4.20	**4.30**	4.19	4.26	**4.40**	4.11	4.18	**4.48**
Average	4.13	4.28	**4.39**	4.22	4.31	**4.43**	4.23	4.36	**4.51**	4.19	4.29	**4.49**

Table 49: The results on each scoring criterion of generated textual quality under IDGG on **WeiboTech**.

	DeepSeek			Llama			Qwen			GPT		
	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.
Contextual Fidelity	4.77	4.89	**4.94**	4.22	4.23	**4.83**	4.89	4.90	**4.99**	4.82	4.91	**4.99**
Personality Depth	4.30	4.50	**4.72**	3.65	3.76	**4.68**	4.22	4.49	**4.83**	4.33	4.43	**4.84**
Dynamic Adaptability	4.42	4.75	**4.84**	3.95	3.96	**4.70**	4.35	4.77	**4.94**	4.65	4.73	**4.92**
Immersive Quality	4.83	4.87	**4.93**	4.02	4.16	**4.82**	4.76	4.89	**4.98**	4.67	4.88	**4.99**
Content Richness	4.69	4.72	**4.83**	3.86	3.87	**4.70**	4.57	4.66	**4.91**	4.52	4.60	**4.89**
Average	4.60	4.75	**4.85**	3.94	4.00	**4.75**	4.56	4.74	**4.93**	4.60	4.71	**4.93**

Table 50: The results on each scoring criterion of generated textual quality under IDGG on **Weibo-Daily**.

	DeepSeek			Llama			Qwen			GPT		
	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.
Contextual Fidelity	4.82	4.90	**4.96**	4.34	4.57	**4.86**	4.82	4.93	**4.99**	4.82	4.90	**4.99**
Personality Depth	4.52	4.69	**4.87**	4.19	4.29	**4.79**	4.72	4.79	**4.97**	4.69	4.72	**4.98**
Dynamic Adaptability	4.64	4.80	**4.91**	4.42	4.38	**4.79**	4.82	4.86	**4.97**	4.71	4.85	**4.99**
Immersive Quality	4.82	4.90	**4.96**	4.32	4.56	**4.86**	4.72	4.94	**4.99**	4.82	4.90	**4.99**
Content Richness	4.62	4.77	**4.91**	4.32	4.34	**4.79**	4.68	4.85	**4.97**	4.67	4.80	**4.98**
Average	4.68	4.81	**4.92**	4.32	4.43	**4.82**	4.75	4.87	**4.98**	4.74	4.83	**4.99**

Table 51: The results on each scoring criterion of generated textual quality under IDGG on **Cora**.

	DeepSeek			Llama			Qwen			GPT		
	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.
Contextual Fidelity	4.64	4.84	**4.94**	4.61	4.77	**4.92**	4.63	4.82	**4.91**	4.72	4.78	**4.93**
Personality Depth	3.69	3.71	**3.80**	3.57	3.60	**3.74**	3.51	3.69	**3.76**	3.68	3.58	**3.80**
Dynamic Adaptability	4.25	4.26	**4.36**	4.20	4.21	**4.44**	4.24	4.28	**4.40**	4.11	4.19	**4.43**
Immersive Quality	4.26	4.81	**4.92**	4.65	4.73	**4.89**	4.76	4.80	**4.88**	4.36	4.72	**4.89**
Content Richness	4.53	4.59	**4.80**	4.25	4.50	**4.71**	4.31	4.59	**4.70**	4.49	4.51	**4.78**
Average	4.27	4.44	**4.56**	4.26	4.36	**4.54**	4.29	4.44	**4.53**	4.27	4.36	**4.57**

F.10 Results of IDGG on Graph Embedding Metrics

Full results on the graph embedding metric under IDGG are provided in Table 52, exhibiting the importance of incorporating structural and textual information in DyTAG generation.

Table 52: The results on the graph embedding metric under IDGG.

	DeepSeek			Llama			Qwen			GPT		
	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.	w/o M.	w/ M.	w/ M.R.
Sephora	0.621	**0.661**	0.634	0.602	0.612	**0.647**	0.569	0.587	**0.616**	0.601	0.603	**0.628**
Dianping	0.739	0.749	**0.769**	0.426	0.457	**0.512**	0.531	0.542	**0.578**	0.521	**0.527**	0.505
WikiRevision	0.673	0.682	**0.699**	0.533	0.579	**0.658**	0.578	0.608	**0.621**	0.589	0.614	**0.629**
WikiLife	0.542	0.555	**0.804**	0.501	0.518	**0.536**	0.541	0.553	**0.549**	0.510	**0.534**	0.531
IMDB	0.601	0.616	**0.729**	0.511	0.522	**0.568**	0.502	0.557	**0.588**	0.521	0.535	**0.563**
WeiboTech	**0.604**	0.591	0.562	0.601	0.629	**0.695**	0.531	0.547	**0.549**	0.501	0.514	**0.536**
WeiboDaily	0.541	**0.567**	0.555	0.620	0.623	**0.638**	0.498	**0.513**	0.503	0.459	0.481	**0.502**
Cora	0.610	0.623	**0.828**	**0.655**	0.642	0.635	0.526	0.599	**0.634**	0.541	0.552	**0.572**

F.11 Results of VRDAG, DG-Gen, and TIGGER-I

Comprehensive results on the graph structural and the graph embedding metrics under IDGG of GAG-General, VRDAG, DG-Gen, and TIGGER-I are shown in Table 53. Since VRDAG (Li et al., 2024), DG-Gen (Hosseini et al., 2024), and TIGGER-I (Gupta et al., 2022) do not support text generation, textual quality comparisons are not feasible. Furthermore, node/edge representations generated by VRDAG and DG-Gen are directly used as node/edge features when computing graph embeddings. The experimental results demonstrate that GAG-General significantly outperforms VRDAG, DG-Gen, and TIGGER-I in both graph structural quality and graph embedding quality. Notably, VRDAG and TIGGER-I struggles to generate high-quality datasets across all eight GDGB datasets, while DG-Gen achieves marginally better performance due to its explicit control over the size of generated graphs. These findings highlight the limitations of existing dynamic graph generation models in the DyTAG generation task and underscore the need for future models to effectively balance structural fidelity, temporal dynamics, and textual richness.

Table 53: The results on the graph structural and the graph embedding metrics under IDGG of our framework and current feature-supportive dynamic graph generation models. Ours correspond to the best results of our proposed GAG-General among four LLM backbones.

Datasets	Models	Ours	VRDAG	DG-Gen	TIGGER-I
Sephora	Degree MMD ↓	**0.370**	0.795	0.422	0.622
	Spectra MMD ↓	**0.189**	0.847	0.274	0.687
	Power-law Validity	✓	✗	✗	✗
	Graph Embedding ↑	**0.661**	0.011	0.228	0.085
Dianping	Degree MMD ↓	**0.150**	0.887	0.167	0.446
	Spectra MMD ↓	0.351	0.808	**0.245**	0.341
	Power-law Validity	✓	✗	✓	✓
	Graph Embedding ↑	**0.769**	0.024	0.517	0.580
WikiRevision	Degree MMD ↓	**0.102**	0.807	0.197	0.658
	Spectra MMD ↓	**0.105**	0.776	0.159	0.606
	Power-law Validity	✓	✗	✓	✗
	Graph Embedding ↑	**0.699**	0.089	0.121	0.105
WikiLife	Degree MMD ↓	**0.078**	0.464	0.283	0.418
	Spectra MMD ↓	**0.206**	0.236	0.288	0.254
	Power-law Validity	✓	✗	✗	✗
	Graph Embedding ↑	**0.804**	0.088	0.303	0.156
IMDB	Degree MMD ↓	**0.145**	0.882	0.373	0.679
	Spectra MMD ↓	**0.379**	0.710	0.537	0.653
	Power-law Validity	✗	✗	✗	✗
	Graph Embedding ↑	**0.729**	0.095	0.293	0.159
WeiboTech	Degree MMD ↓	0.212	0.867	**0.211**	0.313
	Spectra MMD ↓	**0.172**	0.778	0.311	0.384
	Power-law Validity	✓	✗	✓	✓
	Graph Embedding ↑	**0.695**	0.085	0.292	0.289
WeiboDaily	Degree MMD ↓	**0.219**	0.871	0.259	0.465
	Spectra MMD ↓	**0.385**	0.791	0.553	0.589
	Power-law Validity	✗	✗	✓	✗
	Graph Embedding ↑	**0.638**	0.108	0.294	0.241
Cora	Degree MMD ↓	**0.073**	0.877	0.212	0.372
	Spectra MMD ↓	**0.181**	0.760	0.365	0.389
	Power-law Validity	✓	✗	✗	✗
	Graph Embedding ↑	**0.828**	0.053	0.056	0.197

F.12 Results of IDGG on Utility in Data Augmentation for Inductive Learning

We further investigate the utility of DyTAGs generated under the IDGG task for data augmentation in inductive learning scenarios, which simulate real-world cold-start problems such as recommending new items to users. We augment the training data of the original graph with the generated DyTAG (Aug.) and evaluate performance on an inductive link prediction task, where the goal is to predict future links involving nodes not observed during training. The DGNN models and the experimental settings that we used for evaluation are identical to those in Section F.5.

As shown in Table 54, augmenting training data with IDGG-generated graphs consistently improves model performance across all datasets and models. The improvement in Average Precision (ΔAP) ranges from 0.027 to 0.108, with particularly notable gains on WikiLife and Cora. For example, CAWN achieves a ΔAP of 0.108 on WikiLife, and DyGFormer shows a 0.046 improvement on Cora. These results demonstrate that IDGG-generated DyTAGs provide valuable structural and textual context for unseen nodes, effectively enriching the training signal and enhancing model generalization in cold-start settings. This validates the practical value of our generative framework in improving downstream model robustness through synthetic data augmentation.

F.13 Visualizations of the Hub Node Structures

The visualization of hub node structures in both the ground-truth and generated graphs from IDGG is presented in Figures 10 and 11. The top-3 hub product nodes are highlighted in dark red. As described in Section 5.3, both graphs exhibit similar hub node patterns, yet the generated graph contains hub nodes formed through inductive generation, featuring entirely distinct textual profiles.

This divergence arises because IDGG emulates real-world graph evolution dynamics, preserving structural fidelity while generating new nodes with plausible attributes that align with the underlying

Table 54: The results of link prediction (inductive) performance with data augmentation from the graphs generated from IDGG.

Model	Dataset	Type	Average Precision	$\Delta AP \uparrow$
TGN	Sephora	Aug.	0.673 ± 0.010	0.037
		Ori.	0.636 ± 0.002	
	WikiLife	Aug.	0.667 ± 0.020	0.099
		Ori.	0.568 ± 0.024	
	Cora	Aug.	0.552 ± 0.030	0.034
		Ori.	0.518 ± 0.019	
	WeiboTech	Aug.	0.741 ± 0.022	0.041
		Ori.	0.700 ± 0.027	
CAWN	Sephora	Aug.	0.750 ± 0.014	0.050
		Ori.	0.700 ± 0.013	
	WikiLife	Aug.	0.741 ± 0.010	0.108
		Ori.	0.633 ± 0.006	
	Cora	Aug.	0.651 ± 0.017	0.043
		Ori.	0.608 ± 0.010	
	WeiboTech	Aug.	0.740 ± 0.009	0.033
		Ori.	0.707 ± 0.009	
GraphMixer	Sephora	Aug.	0.710 ± 0.006	0.027
		Ori.	0.683 ± 0.025	
	WikiLife	Aug.	0.696 ± 0.004	0.031
		Ori.	0.665 ± 0.030	
	Cora	Aug.	0.575 ± 0.003	0.040
		Ori.	0.535 ± 0.018	
	WeiboTech	Aug.	0.720 ± 0.006	0.058
		Ori.	0.662 ± 0.029	
DyGFormer	Sephora	Aug.	0.691 ± 0.009	0.051
		Ori.	0.640 ± 0.003	
	WikiLife	Aug.	0.734 ± 0.022	0.039
		Ori.	0.695 ± 0.023	
	Cora	Aug.	0.599 ± 0.019	0.046
		Ori.	0.553 ± 0.020	
	WeiboTech	Aug.	0.736 ± 0.008	0.040
		Ori.	0.696 ± 0.016	

generative mechanisms of real systems. For example, in recommendation systems, hub nodes in the generated DyTAG may represent emerging products with high virality potential, whereas those in the ground-truth graph correspond to established bestsellers. This capability establishes DyTAG generation as a strategic tool for proactive decision-making in e-commerce and digital marketing. By identifying potential future hubs, platforms can prioritize resource allocation for product promotion, optimize advertising strategies, and anticipate market trends before they emerge in real-world data.

F.14 Semantic-drift Analysis for IDGG Evaluation

To assess the long-term semantic consistency of generated DyTAGs in the IDGG task, we introduce a cross-snapshot semantic drift detection evaluation. Specifically, we partition the generated 10K-edge DyTAG into 10 temporal snapshots $\{\mathcal{G}_t\}_{t=1}^{10}$, each containing 1K newly added edges and corresponding newly generated nodes. For each snapshot $\mathcal{G}_t$, we compute a global semantic representation $\mathbf{h}_t$ by encoding all textual node and edge attributes using BERT-base-uncased(Devlin et al., 2019) and aggregating via mean pooling:

$$\mathbf{h}_t = \frac{1}{|\mathcal{V}_t| + |\mathcal{E}_t|} \left(\sum_{v \in \mathcal{V}_t} \text{BERT}(v.\text{text}) + \sum_{e \in \mathcal{E}_t} \text{BERT}(e.\text{text}) \right), \tag{1}$$

where $\mathcal{V}_t$ and $\mathcal{E}_t$ denote the sets of nodes and edges introduced in the current snapshot t. We then measure inter-temporal semantic stability via cosine similarity between consecutive representations:

$$s_t = \cos(\mathbf{h}_t, \mathbf{h}_{t+1}), \quad t = 1, \ldots, 9. \tag{2}$$

Thus, higher s_t indicates greater semantic coherence over time.

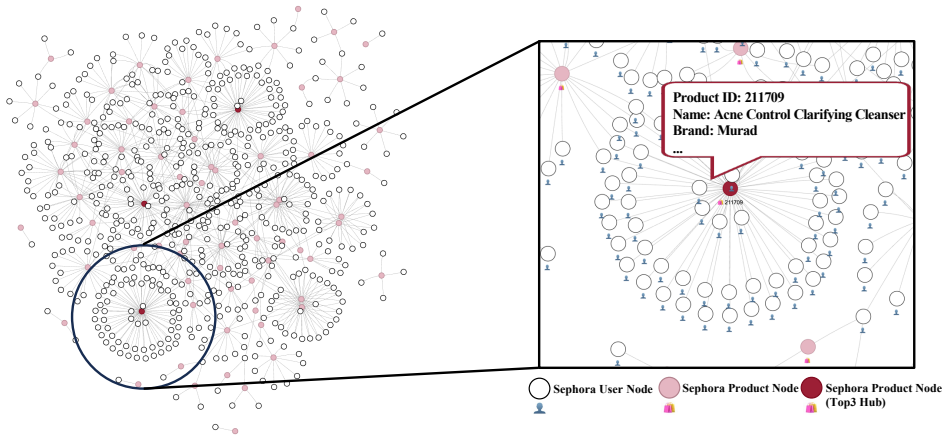

Figure 10: The visualization of the hub node structures in the ground-truth graph.

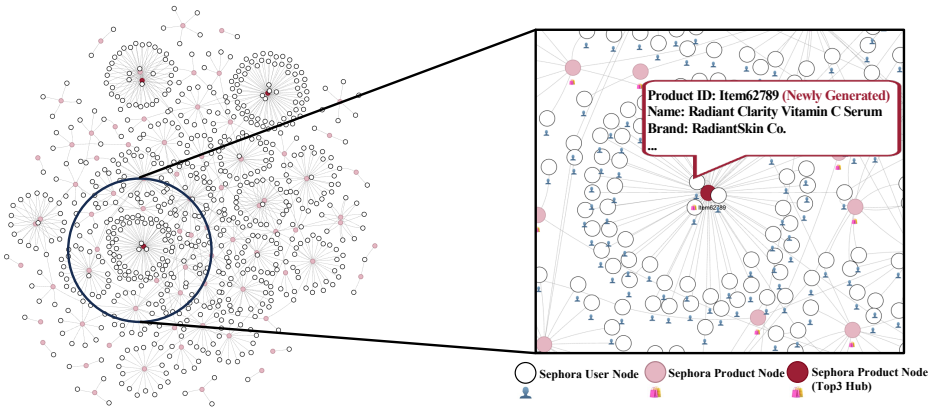

Figure 11: The visualization of the hub node structures in the generated graph.

On the Sephora dataset, as shown in Figure 12, BERT-based similarity metrics reveal a gradual decline after 5K edges under GAG-General—suggesting emerging semantic drift (e.g., generation of out-of-domain items like "makeup bag"). Importantly, we further analyze the semantic drift under guided generation. As shown in Figure 12, we find that the drift is significantly reduced when domain-specific constraints (e.g., "only generate reviews of makeup products (edge generation)/core makeup products (node generation), which are possible to be shown on Sephora platform") are incorporated into the prompts for edge generation (Table 59) or node generation (Table 60), demonstrating the effectiveness of guided generation in future work.

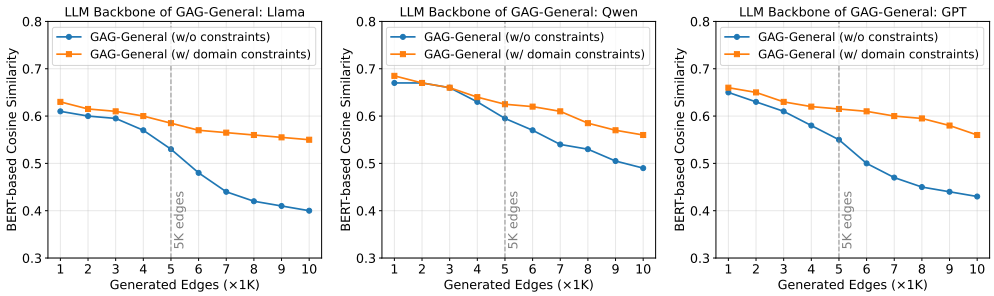

Figure 12: Cross-snapshot semantic consistency over three LLM backbones in IDGG on Sephora.

F.15 HUMAN-AMENITY CHECK FOR IDGG EVALUATION

To evaluate the practical usability and naturalness of generated content, we conduct a small-scale human evaluation on the IDGG-generated DyTAGs. Five skilled annotators independently assessed 50 randomly sampled newly generated nodes and their interactions (e.g., reviews, revisions, citations, etc.). Each item was rated on a 5-point Likert scale across three dimensions: *naturalness*, *plausibility*, and *consistency* with platform context.

The results are shown in Table 55. For instance, the results on Sephora with GPT as the LLM backbone show that generated nodes (users/products) achieve an average score of 3.9, while generated edges (e.g., user reviews) receive a higher average of 4.3. These scores indicate that the majority of generated content is perceived as realistic, contextually appropriate, and aligned with real-world user behavior on the target platform—supporting the human amenity of our framework in practical deployment scenarios.

Table 55: Human evaluation average scores (5-point Likert scale) for generated nodes and edges under IDGG across different LLM backbones and datasets. Higher scores indicate better naturalness, plausibility, and consistency.

	DeepSeek		**Llama**		**Qwen**		**GPT**	
	Node	Edge	Node	Edge	Node	Edge	Node	Edge
Sephora	3.7	4.2	3.6	4.1	3.8	4.3	**3.9**	**4.3**
Dianping	3.5	4.0	**4.0**	**4.5**	3.6	4.1	3.8	4.2
WikiRevision	3.3	3.7	3.2	3.6	**3.6**	**4.0**	3.5	3.9
WikiLife	3.6	4.1	3.5	4.0	3.7	4.2	**3.8**	**4.3**
IMDB	**3.8**	**4.3**	3.3	3.8	3.5	4.0	3.6	4.1
WeiboTech	3.2	3.6	3.1	3.5	3.3	3.7	**3.4**	**3.8**
WeiboDaily	3.3	3.8	**3.8**	**4.3**	3.4	3.9	3.5	4.0
Cora	3.5	4.0	**3.9**	**4.5**	3.6	4.1	3.7	4.2

G PROMPTS

G.1 PROMPT TEMPLATES OF BIPARTITE GRAPHS

Table 56: The prompt template for node description in bipartite graphs (e.g. Sephora).

Node (Sephora User): Author ID: {node_id}, Skin Tone: {skin_tone}, Eye Color: {eye_color}, Skin Type: {skin_type}, Hair Color: {hair_color}, Total Negative Feedback Count: {total_neg_feedback_count}, Total Positive Feedback Count: {total_pos_feedback_count}
Node (Sephora Product): Product ID: {node_id} Product Name: {product_name} Brand Name: {brand_name} Primary Category: {primary_category} Secondary Category: {secondary_category} Ingredients of the Product: {ingredients} Loves Count of the Product: {loves_count} Product Rating: {rating} Number of the Product Reviews: {reviews} Size of the Product: {size} Product Price (USD): {price_usd}

Table 57: The prompt template for edge description in bipartite graphs (e.g. Sephora).

Edge (Sephora Review): Review Time (yyyy-mm-dd): {timestamp} Rating: {rating} Review Title: {review_title} Review Text: {review_text} Total Negative Feedback Count: {total_neg_feedback_count} Total Positive Feedback Count: {total_pos_feedback_count}

Table 58: The prompt template for memory reflection in bipartite graphs (e.g. Sephora).

As a customer of the Sephora online shopping platform, you can review Sephora products based on the provided information and your own situation: {node_info}
Here are your previous review history: {node_memory}
Now, based on your personal description and past review history, progressively refine your memory into a concise version, ensuring it reflects your personal preferences.
Respond.

Table 59: The prompt template for edge generation in bipartite graphs (e.g. Sephora).

Query:
You are a customer of the Sephora online shopping platform, you can review Sephora products based on the provided information and your own situation: {node_info}
Here's your past reviews history: {node_memory}
FIRST, you should Search for candidate prodcuts using the provided tools.
Respond.
Action:
You are a customer of the Sephora online shopping platform, you can review Sephora products based on the provided information and your own situation: {node_info}
Here's your past reviews history: {node_memory}
Here's the candidate prodcuts you can review: {node_items}
Here's the example of how you should proceed with your review: {interaction_example}
You should review ONE product. You can review the chosen product with detailed text and rate it. Additionally, you should predict how many positive/negative feedbacks will be received for this review. The predicted time of the review should be firmly related to the time in your past reviews history (relatively later than or equal to them). Respond using the following detailed JSON format for ONE product:

```
{
   "review":{
      item_id: (str, "The ID of the product you want to review. Be sure to be one of the Item IDs mentioned above!"),
      timestamp: (str, "The time of review (yyyy-mm-dd)"),
      rating: (int, "The overall rating given to the product (From 1 to 5)"),
      review_title: (str, "The title of your review"),
      review_text: (str, "Your detailed review text"),
      total_neg_feedback_count: (int, "Number of negative feedback received for this review"),
      total_pos_feedback_count: (int, "Number of positive feedback received for this review")
   }
}
```

Respond.

Table 60: The prompt template for node generation in bipartite graphs (e.g. Sephora).

Node (Sephora Author):
Now we have a Sephora dataset, which records Sephora users' reviews on several Sephora products. Here's information of the recent active Sephora author nodes: {recent_node_info}
You are expected to generate ONE new Sephora author node for the Sephora dataset, and ensure that the generated new node is somewhat different from the existing nodes. Respond using the following detailed JSON format for ONE new Sephora author:

```
{
   "sephora_author":{
      node_id: (str, "The ID of the generated author (Format: G + 6-digit random number)"),
      node_type: sephora_author,
      skin_tone: (str, "The skin tone of the generated author (e.g. light, fair, mediumTan, tan, olive, etc.)"),
      eye_color: (str, "The eye color of the generated author (e.g. brown, green, hazel, blue, etc.)"),
      skin_type: (str, "The skin type of the generated author (e.g. oily, dry, combination, normal, etc.)"),
      hair_color: (str, "The hair color of the generated author (e.g. brown, black, blonde, auburn, etc.)"),
      total_neg_feedback_count: (int, "The number of total negative feedback received from other authors of the generated author"),
      total_pos_feedback_count: (int, "The number of total active feedback received from other authors of the generated author"),
   }
}
```

Respond.

Node (Sephora Product):
Now we have a Sephora dataset, which records Sephora users' reviews on several Sephora products. Here's information of the recent active Sephora product nodes: {recent_node_info}
You are expected to generate ONE new Sephora product node for the Sephora dataset, and ensure that the generated new node is somewhat different from the existing nodes. Respond using the following detailed JSON format for ONE new Sephora product:

```
{
   "sephora_product":{
      node_id: (str, "The ID of the generated product (Format: G + 5-digit random number)"),
      node_type: sephora_product,
      product_name: (str, "The name of the generated product"),
      brand_name: (str, "The name of the brand of the generated product"),
      primary_category: (str, "The primary category of the generated product"),
      secondary_category: (str, "The secondary category of the generated product"),
      ingredients: (str, "The ingredients of the generated product"),
      loves_count: (int, "The loves count from the users of the generated product"),
      rating: (float, "The avg rating from the users of the generated product"),
      reviews: (int, "The number reviews from the users of the generated product"),
      size: (str, "The size the generated product"),
      price_usd: (float, "The price the generated product"),
   }
}
```

Respond.

G.2 PROMPT TEMPLATES OF NON-BIPARTITE GRAPHS

Table 61: The prompt template for node description in non-bipartite graphs (e.g. WeiboTech).

Node (Weibo User): User ID: {node_id} User Name: {user_name} User Source: {user_source} User Gender: {user_gender} User Location: {user_location} Number of the User's Followers: {user_followers} Number of the User's Followees: {user_friends} User Description: {user_description}

Table 62: The prompt template for edge description in non-bipartite graphs (e.g. WeiboTech).

Edge (Weibo Interact): Interaction Time (yyyy-mm-dd hh-mm-ss): {timestamp} Interaction Type: {label} Source User Text: {src_text} Destination User Text: {dst_text}

Table 63: The prompt template for memory reflection in non-bipartite graphs (e.g. WeiboTech).

As a Weibo user, you can search for other Weibo users who may interact with you on the online social media Weibo platform: {node_info}
Here are your previous interactions history: {node_memory}
Now, based on your personal description and past interactions history, progressively refine your memory into a concise version, ensuring it reflects your personal preferences.
Respond.

Table 64: The prompt template for edge generation in non-bipartite graphs (e.g. WeiboTech).

Query:
As a Weibo user, you need to search for other Weibo users who may interact with you on the online social media Weibo platform: {node_info}
Here are your previous interactions history: {node_memory}
First, utilize the provided tools to search for potential Weibo users who may interact with you.
Respond.
Action:
As a Weibo user, you need to search for other Weibo users who may interact with you on the online social media Weibo platform: {node_info}
Here are the potential Weibo users you can choose from: {node_items}
Here is your previous interaction history: {node_memory}
Here's the example of how to interact with others: {interaction_example}
You should select ONE destination user, you're tend to select the one you have interacted with before. Respond using the following detailed JSON format:

```
{
   "interact":{
      item_id: (str, "The ID of the destination user. Be sure to be one of the Item IDs mentioned above!")
   }
}
```

Respond.
Request:
As a Weibo user, you need to search for other Weibo users who may interact with you on the online social media Weibo platform: {node_info}
Here is your previous interaction history: {node_memory}
The chosen destination Weibo user who may interact with you is: {item_info}
Here's the interaction history of this chosen Weibo user: {item_memory}
Here's the example of how to interact with others: {interaction_example}
You should select ONE destination user. You should post texts as the source user and the selected destination user should interact with you in detailed text. Additionally, you should label the type of the interaction (comment or repost). The predicted time of the interaction should be firmly related to the time in your previous interaction history (relatively later than or equal to them). Respond using the following detailed JSON format:

```
{
   "interact":{
      item_id: (str, "The ID of the destination user"),
      timestamp: (str, "The time of the interaction (yyyy-mm-dd hh-mm-ss)"),
      label: (str, "The type of interaction (TWO TYPE: 1.comment, 2.repost)"),
      src_text: (str, "The text from the source user"),
      dst_text: (str, "The text from the destination user")
   }
}
```

Respond.

Table 65: The prompt template for node generation in non-bipartite graphs (e.g. WeiboTech).

Node (Weibo User):
Now we have a weibo dataset, which records the interaction history between Weibo users. Here's information of the recent active user nodes: {recent_node_info}
You are expected to generate ONE new user node for the weibo dataset, and ensure that the generated new node is somewhat different from the existing nodes. Respond using the following detailed JSON format for ONE new user:

```
{
   "weibo_user":{
      node_id: (str, "The ID of the generated user (Format: G + 5-digit random number)"),
      node_type: weibo_user,
      user_name: (str, "The name of the generated user"),
      user_source: (str, "The source(IP/location/device) of the generated user"),
      user_gender: (str, "The gender of the generated user"),
      user_location: (str, "The location of the generated user"),
      user_followers: (int, "The number of the followers of the generated user"),
      user_friends: (int, "The number of the followees of the generated user"),
      user_description: (int, "The description of the generated user"),
   }
}
```

Respond.

G.3 PROMPT TEMPLATES OF LLM-AS-EVALUATOR IN TEXTUAL QUALITY METRICS

Table 66: The prompt template of evaluation criteria for LLM-as-Evaluator.

Please evaluate the role-playing ability of the ACTOR NODE based on its actions with ITEM NODES across multiple turns based on scene, ACTOR NODE information, action history and evaluation criteria.

[Environment Description]: {environment}

[ACTOR NODE Description]: {actor_node}

[Multi-turn Actions]: {actions}

Strict Evaluation Criteria:
1. Factual Accuracy: Identify and point out any elements that do not accurately match the historical or factual backdrop.
2. Behavior Consistency: Explicitly highlight inconsistencies between the actor nodes' actions and their traits.
3. Logical Coherence: Point out any logical fallacies or actions that contradict the established context or logic.
4. Content Redundancy: Identify repetitions in actions that could detract from engagement and realism.
5. Emotional Expression: Assess whether emotional responses and expressions are appropriate and convincingly portrayed, highlighting any discrepancies.
6. Interaction Adaptability: Critique the actor node's interactions with item nodes, noting any unnatural or contextually inappropriate responses.
7. Creativity and Originality: Evaluate the creativity of responses and actions, pointing out generic or unoriginal content.
8. Detail Handling: Scrutinize the level of detail in scene setting and actor node enactment, marking areas lacking depth or accuracy.
9. Style Consistency: Ensure that the linguistic style remains consistent, identifying any deviations.
10. Fluency and Quality: Critically assess the smoothness and quality of the text, highlighting any grammatical errors or awkward phrasings.

Table 67: The prompt template of scoring criteria for LLM-as-Evaluator.

[Scoring Criteria]:
1. Contextual Fidelity:
- 1 Point: Responses are often incorrect or irrelevant, significantly inconsistent with the actor node's background.
- 3 Points: Responses are generally accurate, though there are occasional errors or some details are not very relevant to the actor node's background.
- 5 Points: Responses are always accurate and highly relevant to the actor node's historical or professional background, demonstrating deep knowledge and skills.

2. Personality Depth:
- 1 Point: The displayed personality traits often conflict with or lack consistency with the actor node's setup.
- 3 Points: Personality traits generally match the actor node's setup, though there are occasional inconsistencies.
- 5 Points: Consistently displays behavior and language choices that match the core personality traits of the actor node, showcasing the actor node's uniqueness.

3. Dynamic Adaptability:
- 1 Point: The actor node struggles to adapt to new scenarios over time, producing rigid or illogical responses.
- 3 Points: Adaptation occurs in most situations but occasionally falters in handling unexpected turns or maintaining coherence.
- 5 Points: The actor node fluidly adapts to novel contexts and introduces innovative solutions and maintains consistency even under challenging conditions.

4. Immersive Quality:
- 1 Point: Actor node portrayal is often inconsistent, making it difficult for users to immerse or understand the actor node.
- 3 Points: Actor node is mostly consistent, but occasional contradictions slightly affect immersion.
- 5 Points: Actor node portrayal is always consistent, enhancing user immersion and effectively showing self-awareness and actor node limitations.

5. Content Richness:
- 1 Point: Output is sparse, lacking depth, detail, or meaningful interaction with item nodes.
- 3 Points: Content is adequate but occasionally superficial, missing opportunities for nuanced or layered interactions.
- 5 Points: Output is exceptionally dense with relevant details, creative ideas, and meaningful engagement with item nodes. It showcases a high level of expertise and creativity, providing users with an abundance of valuable information and interactive elements that significantly enhance the overall experience.

Table 68: The prompt template of evaluation steps and response formats for LLM-as-Evaluator.

[Evaluation Steps]:
1. Contextual Understanding: Examine the actor node's profile and background information thoroughly to fully grasp the nuances of their context, motivations, and historical background.
2. Behavioral Observation: Monitor how the actor node reacts across different scenarios, paying special attention to their decisions and interactions.
3. Criteria-Based Assessment: Strictly analyze each observed behavior using the above criteria to systematically evaluate the consistency and effectiveness of the actor node's portrayal.

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]:
Contextual Fidelity: [1-5]
Personality Depth: [1-5]
Dynamic Adaptability: [1-5]
Immersive Quality: [1-5]
Content Richness: [1-5]

[Response Format Example]:
Contextual Fidelity: 3
Personality Depth: 3
Dynamic Adaptability: 3
Immersive Quality: 3
Content Richness: 3

[Response]:

Table 69: The prompt template of dataset descriptions for LLM-as-Evaluator.

Sephora:
General Description: This is a dataset recording users' reviews of products on the Sephora e-commerce platform, including information about each user's age, skin type, eye color, information about each product, and users' reviews of each product. Main Actors: The users who review products on the Sephora e-commerce platform. Main Activities of Actors: Sephora users review various products, rating products, and providing additional user experiences.
Dianping:
General Description: This is a dataset recording users' reviews of businesses on the Dianping platform, including information about each user, information about each business, and users' reviews of each business. Main Actors: The users who review businesses on the Dianping platform. Main Activities of Actors: Dianping users review various businesses, rating businesses, and providing additional user consumption experiences.
WikiRevision:
General Description: This is a dataset recording users' revisions on Wikipedia pages, including information about each wiki page and comments submitted by users during revision. Main Actors: The users who revise the wiki pages. Main Activities of Actors: Wiki users revise various wiki pages (correcting errors in wiki pages, adding new content, removing redundant content, and summarizing the wiki pages) with a comment or brief summary in text.
WikiLife:
General Description: This is a dataset recording the life trajectories of several people (celebrities), including basic information about each person, information about the location of each life trajectory, and the events happened at those locations for each person. Main Actors: The people who physically present in some locations for various categories of life trajectories. Main Activities of Actors: People (celebrities) physically present in the specified location for various life trajectories, such as being born, studying, working, getting married, having children, and passing away.
IMDB:
General Description: This is a dataset recording the collaboration relationships of movie actors/actresses, including information about each movie actor/actress, information about the movies they collaborated on, and the roles they played in those movies. Main Actors: The movie actors/actresses who collaborate with each other. Main Activities of Actors: Movie actors/actresses collaborate with others and playing different roles in movies.
WeiboTech/WeiboDaily:
General Description: This is a dataset recording users' interaction on the Weibo platform, including information about each user and the text information attached during user interactions such as comments and reposts. Main Actors: The users who interact with each other on the Weibo platform. Main Activities of Actors: Weibo users make multiple posts, and other users interacting with them by commenting or reposting the posts.
Cora:
General Description: This is a dataset recording paper citation relationships, including information about each paper and the specific text attached during citation in a specific section of the paper. Main Actors: The academic papers which cite other related papers. Main Activities of Actors: Academic papers cite other related papers with specific citation sentences in the text.

H USE OF LARGE LANGUAGE MODELS (LLMS)

LLMs were used in two specific aspects of this work. First, we employ LLMs to polish the manuscript text and refine figure captions for clarity and presentation quality. Second, our proposed generative framework, GAG-General, is built upon an LLM-based multi-agent architecture that coordinates multiple specialized agents to generate structurally and semantically coherent DyTAGs. The LLMs in this framework are used to model textual attributes and support agent reasoning during graph

generation. No other parts of the research, including problem formulation, method design, or analysis, involved significant LLM assistance.