Many Hands Make Light Work: Group-based Information Diffusion Prediction over Long-Context Cascades

Zihan Feng
Tianjin University
Tianjin, China
zihanfeng@tju.edu.cn

Yajun Yang*
Tianjin University
Tianjin, China
yjyang@tju.edu.cn

Xin Huang
Hong Kong Baptist University
Hong Kong, China
xinhuang@comp.hkbu.edu.hk

Xin Wang
Tianjin University
Tianjin, China
wangx@tju.edu.cn

Hong Gao
Zhejiang Normal University
Jinhua, Zhejiang, China
honggao@zjnu.edu.cn

Qinghua Hu
Tianjin University
Tianjin, China
huqinghua@tju.edu.cn

Abstract

Information diffusion prediction aims to forecast the temporal spread of opinions and behaviors by identifying potential adopters. Existing methods typically treat information diffusion as a sequence of individual adoptions and rely on computationally expensive pairwise (one-to-one) influence computations, often restricting predictions to just the next adopter. This individual-level paradigm both misrepresents real-world collective (many-to-many) influences and suffers a critical efficiency trade-off: to remain feasible, such models must truncate long diffusion histories, thereby overlooking early initiators and opinion leaders. To overcome these limitations, we formalize a more practical task: Group-based Information Diffusion Prediction, and propose an effective and scalable GRID framework. Specifically, GRID first learns group-oriented graph embeddings via a task-regularized information bottleneck objective, which amplifies key influence pathways and produces reliable user embeddings for group identification. Built on these embeddings, the core GroupAttn module captures inter-group influence while reducing complexity from quadratic to linear in cascade length. This enables the modeling of ultra-long cascades (exceeding 10,000 users) without truncation while preserving representational fidelity within a provable error bound. Finally, a group-wise objective guides the model to predict semantically meaningful future groups. Extensive experiments on four real-world datasets show that GRID outperforms ten state-of-the-art baselines by an average of 10.65% in accuracy, while achieving an order-of-magnitude gain in efficiency and extending the supported cascade length by up to 10 times.

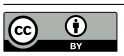
CCS Concepts

• **Information systems** → **Social networks**; *Web mining*.

Keywords

Information Diffusion; Public Opinion; Social Network Analysis

*Corresponding author.



WWW '26, Dubai, United Arab Emirates

ACM ISBN 979-8-4007-2307-0/2026/04
https://doi.org/10.1145/3774904.3792082

ACM Reference Format:
Zihan Feng, Yajun Yang, Xin Huang, Xin Wang, Hong Gao, and Qinghua Hu. 2026. Many Hands Make Light Work: Group-based Information Diffusion Prediction over Long-Context Cascades. In *Proceedings of the ACM Web Conference 2026 (WWW '26), April 13–17, 2026, Dubai, United Arab Emirates.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3774904.3792082

1 Introduction

Information diffusion prediction aims to identify potential users who are likely to be influenced by public opinion, such as sharing or commenting on TikTok videos. Accurate predictions provide crucial insights into user behavior and content virality, benefiting a wide range of downstream applications, from personalized recommendations [11, 28, 36] to public opinion trend analysis [16, 17, 22].

Information diffusion prediction has been an active research area for decades [4, 6, 26, 35, 44], driven by its significance in understanding and improving online platforms as a socio-economic system. A fundamental concept in this domain is the *diffusion cascade*: a temporal sequence of user-time tuples (v_i, t_i), where v_i denotes a participant and t_i the participation time. Each cascade records those who adopt or share information over time. Across prior work, the central task is often framed as *next-participant prediction*: given an observed partial cascade, the goal is to identify which user will be the next to join. Recently, state-of-the-art methods [7, 8, 19, 24, 27, 31, 39, 43] have typically converged on a two-phase framework, termed *GNNs + Sequence Models*. First, Graph Neural Networks (GNNs) are employed on the underlying social network to learn social-aware embeddings for all potential users. These embeddings capture the global context of user relationships. Second, a sequence model, such as a Transformer or RNN, processes the sequential diffusion process in a specific cascade. This step models the sequential dynamics of how influence spreads from one user to the next. By jointly leveraging global social relationships and local diffusion influences, this two-phase framework has achieved favorable performance in predicting the next participant. Despite these advances, the individual-level prediction paradigm suffers from two fundamental limitations:

Limitation 1: Impracticality and Mismatch with Reality: In real-world diffusion scenarios, such as viral TikTok videos, the number of participants can surge into the millions within a short period. For such long-context cascades, predicting the exact sequence of individual adopters is not meaningful and yields limited

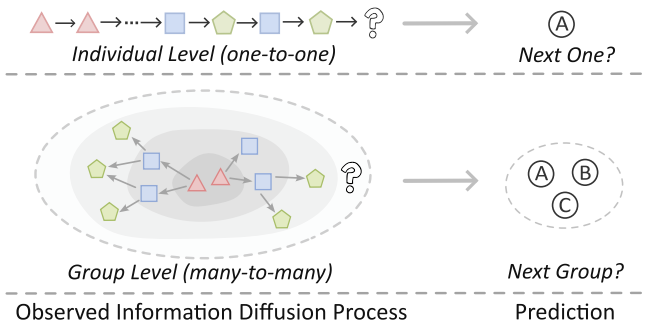


Figure 1: Comparison of diffusion paradigms. Individual-level modeling assumes sequential one-to-one influence, while group-level modeling captures collective, many-to-many adoption. Colors and shapes denote distinct behaviors.

practical insights. More fundamentally, the individual prediction paradigm misrepresents the nature of modern information spread. While classic models assume sequential, one-to-one influence, empirical evidence [5, 18, 20] suggests that diffusion influence is often a *many-to-many* or *group-to-group* process and is driven by collective exposure within influential groups, not just by isolated pairwise interactions. Furthermore, the stochastic nature of user behavior means that small differences in participation time are often inconsequential noise in the overall diffusion process. However, existing methods adopt the *"winner-take-all"* objective and only focus on the next one. For instance, in Fig. 1, if user B joins the diffusion process *just a few seconds* after user A, we should not assume that predicting only user A is right, yet user B is wrong. Consequently, shifting the task objective from predicting only the next user A to identifying *a meaningful set of future participants*, such as $\{A, B, C\}$, is more practical for long-context information diffusion.

Limitation 2: Efficiency-Effectiveness Trade-off: The focus on individual, pairwise interactions has led to a significant trade-off between model effectiveness and computational efficiency. Models based on standard RNNs or Transformer architectures have to process one-to-one pairwise interactions among participants, incurring a computational complexity that is at least quadratic ($O(m^2)$ for a cascade of length m). This cost becomes prohibitive for long-context cascades with numerous participants. To remain feasible, a common practice is to truncate the cascade, considering only the last 200 participants, as in [7–9, 19, 24, 27, 31, 39, 43]. However, this truncation inevitably overlooks long-range dependencies and crucial influences, such as initial authors or key opinion leaders.

Motivated by these observations, we advocate a paradigm shift from individual-level to group-level modeling and formalize the new task of *Group-based Information Diffusion Prediction*. This raises a core question: *How can we effectively identify and model the influence of dynamically formed groups within a diffusion process?* Unlike static communities that persist over time, diffusion groups are transient and context-dependent, emerging within specific cascades as users coordinate around shared interests or events. This dynamic nature renders prior community-based methods inadequate, since fixed communities fail to capture behavioral and temporal coherence. **(1)** The first challenge lies in identifying such groups *directly from noisy and incomplete diffusion evidence*. The behavioral signal (*i.e.*, the cascade) contains stochastic adoptions that may not imply influence, while the structural signal (*i.e.*, the social graph) is rife with spurious or missing ties. Reliable identification, therefore, requires jointly denoising and fusing both local and global views to uncover genuine coordination patterns. **(2)** After group identification, *modeling their influence* remains difficult in both scale and representation. On the scale side, a group's influence may span a long diffusion history, yet existing models must truncate cascades for efficiency, discarding early but decisive influences and biasing the dynamics. On the representation side, group influence is more than the sum of pairwise interactions; it also depends on collective properties such as *group size* and *roles*. For instance, a few opinion leaders may steer attention through credibility, whereas a large crowd of minor participants can collectively amplify visibility—the well-known *"rumors can be frightening"* phenomenon [3, 21, 34]. These aspects highlight that influence arises from *coordinated group behavior over long temporal contexts*, making inter-group modeling both complex and computationally demanding. **(3)** Finally, rather than predicting a single next adopter, the model must forecast a *set of future participants* forming a coherent diffusion group. This requires a group-wise objective that aligns optimization with collective semantics, rather than with isolated individual actions.

To address these challenges, we propose GRID, an efficient and effective framework for **GR**oup-based **I**nformation **D**iffusion prediction. First, to improve group quality, GRID introduces a group-oriented graph representation module that prunes the social graph to highlight true influential pathways and applies a self-supervised objective, inspired by the information bottleneck principle, to learn group-oriented user embeddings driven by *true influential signals rather than spurious correlations*. Second, to model inter-group influence at scale, we design an efficient *GroupAttn* module that operates on a compact set of group representatives, rather than costly pairwise interactions. This reduces computational complexity *from quadratic to linear* in the cascade length, enabling GRID to analyze long-context cascades over 10,000 without truncation, while preserving representational fidelity within a bounded error $O(\delta)$ to group tightness δ. Third, to ensure that the identified groups form coherent behavioral units rather than loose collections of individuals, we reformulate the prediction task itself. We carefully design a group-wise objective to guide the model optimization with both *completeness* and *correctness* of predicted diffusion groups. Experiments on real-world datasets validate the superiority of GRID: *(i) Quality:* It improves group prediction by an average of *10.65%* over ten SOTA competitors. *(ii) Scalability:* It increases the maximum supported cascade length by 10× while baselines fail with out-of-memory errors. *(iii) Efficiency:* It is ~30× faster in inference and uses nearly 60% less GPU memory than competitors.

Overall, the main contributions of this paper are as follows: **(1)** We introduce and formalize the novel problem of group-based information diffusion prediction, providing a more general and practical paradigm. **(2)** We propose GRID, an efficient and effective framework with linear complexity and a provable error bound. The framework is optimized with a group-wise objective to ensure the coherent and meaningful diffusion groups. **(3)** Experiments on four real-world datasets validate that GRID outperforms ten state-of-the-art methods in terms of quality, scalability, and efficiency.

2 Preliminaries

2.1 Concepts

The social network is represented as a directed graph $G = (V, E)$, where V is the set of users (or nodes), with $|V| = N$, and $E \subseteq V \times V$ is the set of directed edges, with $|E| = M$. Each edge signifies a "following" relationship within the social network. For social networks, the terms "user" and "node" are used interchangeably.

Definition 1: (Information Diffusion Cascade). An *information diffusion cascade* C is a sequence of user adoptions, ordered by time: $C = \{(v_1, t_1), (v_2, t_2), \ldots, (v_m, t_m)\}$. Each tuple (v_i, t_i) indicates that user $v_i \in V$ adopted the information at time t_i, with $t_1 \leq t_2 \leq \cdots \leq t_m$. The integer m denotes the length of the diffusion cascade.

Traditional individual-level approaches primarily focus on *next-participant prediction*. Given an observed cascade C, this task aims to predict the next participant v_{m+1}, while subsequent users (*e.g.*, v_{m+2}) are typically disregarded. However, this "winner-take-all" prediction overlooks real-world coordinated group behavior. Motivated by this limitation, we introduce the following concepts to formalize the group-level task.

Definition 2: (Diffusion Group). A *diffusion group* is a non-empty subset of participants within a cascade that exhibit similar patterns. It is defined as $s_i = \{(v_{i,1}, t_{i,1}), (v_{i,2}, t_{i,2}), \ldots, (v_{i,n}, t_{i,n})\} \subseteq C$. The integer n denotes the size of the group s_i.

These diffusion groups are identified dynamically based on shared activity patterns within the cascade, thereby forming behaviorally coherent units. This allows us to reformulate the diffusion process as a sequence of group adoptions rather than individual adoptions.

Definition 3: (Group-based Cascade). A *group-based cascade*, S, is an ordered collection of diffusion groups that form a partition of the original cascade. It is represented as $S = \{s_1, s_2, \ldots, s_k\}$, where each $s_i \subseteq C$ and their union reconstructs the original cascade, *i.e.*, $\bigcup_{i=1}^{k} s_i = C$. The integer k denotes the number of groups.

In contrast to conventional models that treat diffusion as a sequence of isolated individual actions, the group-based cascade framework captures the collective and temporally correlated nature of real-world influence dynamics.

2.2 Problem Definition

Based on these concepts, we can define the problem of Group-based Diffusion Prediction: Given a social network $G = (V, E)$ and an observed diffusion cascade C, the task consists of two mappings: (1) a grouping function $\phi\colon (G, C) \rightarrow S$ that partitions C into a group-based cascade $S = \{s_1, \ldots, s_k\}$; (2) a prediction function $f\colon (G, S) \rightarrow s_{k+1}$ that forecasts the next diffusion group $s_{k+1} \subseteq V$.

Discussion. Existing methods typically predict the next individual user, yet real diffusion often unfolds through coordinated groups rather than isolated individuals. Capturing these collective dynamics yields a more faithful model of real-world influence. However, this shift introduces two core challenges. First, diffusion groups are dynamic and context-specific, not static communities; they must be identified adaptively within each cascade. Second, predicting a set of users with varying composition is harder than predicting a single individual. In other words, accurate multi-step prediction is more challenging than next individual prediction. In practice, our formulation offers *generalization*. When group size $|s_i| = 1$, it reduces to individual prediction; when groups are restricted to predefined communities, it aligns with community-level prediction.

3 Related Work

3.1 Information Diffusion Prediction

Information diffusion prediction aims to identify future infections based on historical diffusion paths. Recent works can be broadly classified into *cascade-based models* and *graph-based models*. Cascade-based models [15, 33, 37] typically use RNNs or Transformer-like architectures to learn the diffusion patterns within the cascades. For graph-based models [7, 8, 24, 25, 27, 31, 38, 39, 43], these methods are based on a general observation that people have common interests with their friends. For example, DyHGCN [39] utilizes both the social network and diffusion graph to learn user embeddings. Recent advancements in hypergraph learning enable more nuanced modeling of user interactions. Methods such as DisenIDP [7], MSHGAT [27], and MINDS [19] employ hypergraphs to capture dynamic user preferences. RotDiff [24] uses hyperbolic space instead of Euclidean space for more refined embeddings. MGCL [8] designs graph contrastive learning objectives to improve the embedding quality and mitigate the data quality issue. GODEN [31] designs an ODE-based GNN to model dynamic relationships, while CARE [43] retrieves similar historical cascades to enhance current cascade learning. Inspired by social psychology, PMRCA [12] and SILN [9] introduce classical propagation patterns into cascade modeling to finely identify the diffusion influence. All of these methods focus on modeling one-to-one pairwise influences, where they have to truncate the cascade length to 200 for computational feasibility. Furthermore, these methods fundamentally target *next-participant prediction*, which is often unreliable in the real-world scenarios.

3.2 Community-level Information Pathways

Community-level information pathways prediction (CLIPP) [10, 20, 23, 32] formalizes a coarse-grained task at community-level: predicting the propagation pathway of multi-modal content across online communities. It constructs a community influence graph (each community as a node) and employs dynamic GNNs to capture evolving inter-community transmission patterns. In CLIPP, communities are predefined and fixed macro-level units, and the modeling objective is restricted to content flow between these communities. Despite its strengths, CLIPP differs fundamentally from our proposed group-based diffusion framework. First, by relying on community units assumed to exist a priori, CLIPP overlooks intra-community heterogeneity, such as the varying roles of key influencers versus peripheral members, and fails to capture fine-grained influence dynamics within each community. Second, the assumption that ground-truth community labels are available is often unrealistic in practice, as structural community detection is often misaligned with actual diffusion behavior. Third, because CLIPP operates on fixed boundaries, it cannot accommodate the dynamic, cascade-specific user groupings that naturally emerge during diffusion. In contrast, GRID identifies diffusion groups dynamically based on shared behavioral patterns within each diffusion cascade.

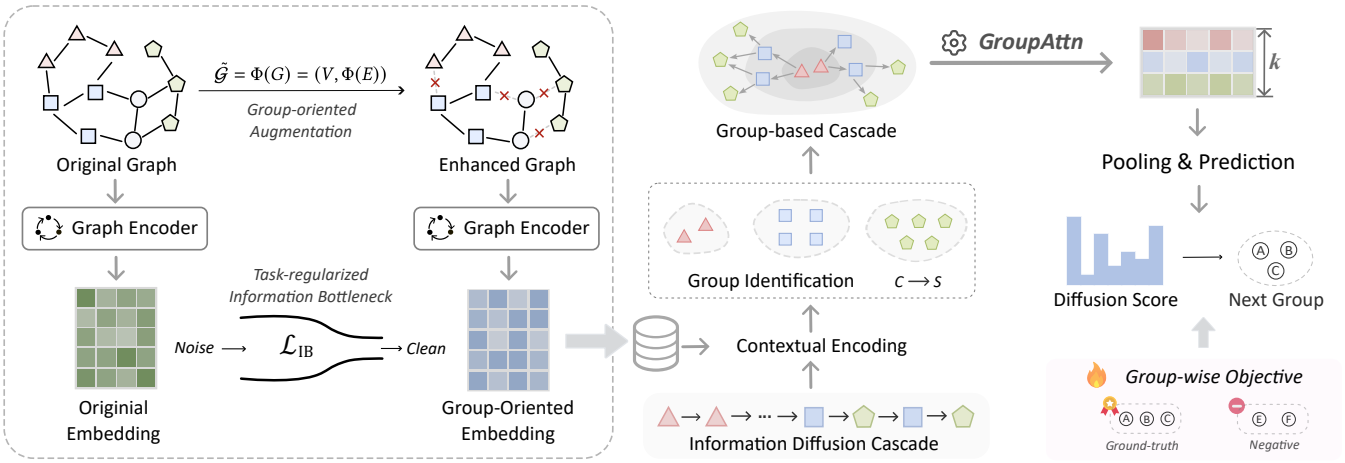


Figure 2: Overview of the proposed GRID framework.

4 Methodology

Overview. Figure 2 illustrates the overall framework. GRID first refines the original social graph through a principled pruning strategy $\Phi(G)$, preserving only high-confidence influence pathways. A task-regularized information bottleneck $\mathcal{L}_{\text{IB}}$ then guides the graph encoder to learn group-oriented embeddings, providing a clean foundation for identifying functional groups. Using these embeddings, GRID dynamically partitions each diffusion cascade into coherent groups further based on the similarity of diffusion context. The core *GroupAttn* module (detailed in Figure 3) efficiently captures inter-group influences, reducing complexity from quadratic to linear in cascade length within a bounded error $O(\delta)$. Finally, the group-level representations are pooled for diffusion scoring and optimized end-to-end with a group-wise objective, ensuring that predicted groups remain semantically coherent.

4.1 Group-oriented Social Embedding

In this part, we aim to learn group-oriented user embeddings, providing a foundation for subsequent group identification.

4.1.1 Denoised Graph Augmentation. The inherent noise in social networks leads to a gap between structural connectivity and the behavioral cohesion required for diffusion patterns, which obscures meaningful influence pathways and often impairs subsequent group identification [8, 29]. We first design a group-oriented graph augmentation strategy to enhance meaningful influence pathways by selectively pruning spurious connections based on interaction patterns. Specifically, we define the denoised graph as:

$$\tilde{\mathcal{G}} = \Phi(G) = (V, \Phi(E)), \tag{1}$$

where $\Phi(\cdot)$ is the augmentation operator that filters the original edge set E to produce a refined graph $\tilde{\mathcal{G}}$ that accurately reflects group diffusion dynamics. The core motivation for $\Phi(\cdot)$ is that users who are frequently and temporally co-activated within the same diffusion cascades are more likely to exhibit similar patterns and form a behavioral group. Specifically, we first compute an interaction score based on historical co-activation patterns. For each pair of nodes (v_i, v_j), we offline compute their interaction score $P_{i,j}$ by aggregating their interactions across all historical cascades C: $P_{i,j} = \sum_{C \in \mathcal{C}} \sum_{t_j > t_i} -\lambda(t_j - t_i)$, where (v_i, t_i) denotes the activation of user v_i at time t_i in cascade C, and $\lambda > 0$ is a temporal decay factor. This term penalizes large activation time gaps, thereby emphasizing influence between users who engage in close temporal proximity. The resulting influence matrix $\mathbf{P}$ quantifies the behavioral strength of directed information flow between users. Based on the influence matrix $\mathbf{P}$, for each node v_i, we prune the edge $(v_i, v_j) \in E$ corresponding to the lowest α fraction of scores $P_{i,j}$.

4.1.2 Group-Oriented Embedding Module. Given the augmented graph $\tilde{\mathcal{G}}$, we employ a GNN block [13] as a graph encoder to iteratively update user embeddings via neighborhood aggregation:

$$\mathbf{z}_i^{(l+1)} = \sum_{v_j \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i||\mathcal{N}_j|}} \mathbf{z}_j^{(l)}, \tag{2}$$

where $\mathcal{N}_i$ is the neighbor set of user v_i in $\tilde{\mathcal{G}}$, and $\mathbf{z}_i^{(l)}$ is the embedding of v_i at layer l. The final user representation is obtained by summing the embeddings from all layers: $\mathbf{z}_i = \sum_{l=1}^{L} \mathbf{z}_i^{(l)}$. The embeddings for all users form the matrix $\mathbf{Z} \in \mathbb{R}^{N \times d}$.

To ensure that the produced embeddings $\mathbf{Z}$ capture group-relevant semantics while discarding noise inherent in the initial graph, we introduce a self-supervised objective grounded in the Information Bottleneck (IB) principle [30]. The IB principle posits that an ideal representation $\mathbf{Z}$ should be maximally informative about a target variable Y while being minimally informative about the original input X. This is formally expressed as:

$$\max_{\mathbf{Z}} \ \mathcal{I}(Y; \mathbf{Z}) - \beta \cdot \mathcal{I}(X; \mathbf{Z}), \tag{3}$$

where $\beta > 0$ is a trade-off parameter and $\mathcal{I}(\cdot\,;\cdot)$ denotes mutual information. Here X represents the initial, noisy user features (*e.g.*, embeddings from the original graph); Y corresponds to the unobserved, latent group memberships; and $\mathbf{Z}$ is the refined embedding learned from the denoised graph $\tilde{\mathcal{G}}$. Eq. (3) thus seeks an embedding $\mathbf{Z}$ that is predictive of group structure (maximizing $\mathcal{I}(Y; \mathbf{Z})$) while compressing the initial features (minimizing $\mathcal{I}(X; \mathbf{Z})$).

Since the latent groups Y are unknown, we instantiate this objective using a tailored contrastive learning framework. We construct diffusion-aware positive and negative sets for each user v_i using the influence matrix $\mathbf{P}$. The positive set, $\text{pos}(i)$, comprises the top-r users with the highest interaction scores $P_{i,j}$, who are behaviorally similar in diffusion events. Conversely, the negative set, $\text{neg}(i)$, consists of the bottom-r users with the lowest scores, who likely belong to unrelated groups. We enforce two complementary constraints implemented as follows:

$$\mathcal{L}_{\text{IB}} = \sum_{v_i \in V} \log \frac{\sum_{v_j \in \text{pos}(i)} \text{sim}(\mathbf{z}_i, \mathbf{z}_j)}{\sum_{v_k \in \text{neg}(i)} \text{sim}(\mathbf{z}_i, \mathbf{z}_k) + \beta \cdot \text{sim}(\mathbf{z}_i, \mathbf{x}_i)} + \sum_{v_i \in V} \log \frac{\text{sim}(\mathbf{z}_i, \mathbf{x}_i)}{\sum_{v_k \in \text{neg}(i)} \text{sim}(\mathbf{z}_i, \mathbf{z}_k)}, \tag{4}$$

where $\text{sim}(\mathbf{a}, \mathbf{b}) = \exp(\cos(\mathbf{a}, \mathbf{b})/\tau)$ is the similarity function with temperature τ. $\mathbf{x}_i$ denotes the embedding of user v_i learned from the original graph G. In the first term, the noisy embedding $\mathbf{x}_i$ serves as a task-irrelevant negative sample, weighted by β. The second term treats $\mathbf{x}_i$ as a positive sample, which encourages to maintain individual identity relative to completely irrelevant users in $\text{neg}(i)$.

4.2 Group-based Cascade Modeling

In this part, we focus on a crucial real-world phenomenon: *influence often propagates between behavioral groups of users, not just between one-to-one individuals*. The core premise is that by identifying individual participants into compact group-level representations, we can capture the macro-dynamics of group-based information flow.

4.2.1 From Individuals to Groups. Given a diffusion cascade $C = \{(v_1, t_1), (v_2, t_2), \ldots, (v_m, t_m)\}$ with m ordered participants, our first step is to generate a contextual representation for each user's participation within the cascade. Each user $v_i \in C$ has the group-oriented embedding $\mathbf{z}_i \in \mathbf{Z}$. To enhance temporal (sequential) information, we employ temporal position embeddings, which injects relative temporal information into the embedding space. We then fuse the static embedding and its temporally-injected counterpart, $\mathbf{h}_i = \mathbf{z}_i + \mathbf{z}_i^t \in \mathbb{R}^d$. The entire cascade is thus represented by $\mathbf{H}_c = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_m]^\top \in \mathbb{R}^{m \times d}$.

In previous works [9, 10, 24, 27] with the standard attention mechanism, the interaction strength between any two participants v_i and v_j is computed via a scaled embedding dot-product $A_{ij} \propto \mathbf{h}_i^\top \mathbf{h}_j$. A key observation arises: if two participants v_j and v_k have similar characteristics ($\mathbf{h}_j \approx \mathbf{h}_k$), their interaction scores with any other user v_i will also be nearly identical:

$$A_{ij} \approx A_{ik} \quad \text{when} \quad \mathbf{h}_j \approx \mathbf{h}_k. \tag{5}$$

This property reveals a fundamental diffusion pattern: *users with similar characteristics tend to exhibit similar influence patterns, naturally forming behavioral groups*. Therefore, rather than modeling each participant in isolation, it is both conceptually meaningful and computationally efficient to model influence at the group level.

Motivated by this insight, we propose to explicitly approximate these individual-level influences with a small set of representative group-level patterns. Specifically, we cluster the m cascade participants based on the similarity of their contextual embeddings in $\mathbf{H}_c \in \mathbb{R}^{m \times d}$. We implement a GPU-efficient k-Means algorithm to partition the user embeddings $\{\mathbf{h}_1, \ldots, \mathbf{h}_m\}$ into k groups, denoted $S = \{s_1, \ldots, s_k\}$. To balance effectiveness with efficiency, we empirically set the group number using a predefined k (ensuring $k \ll m$). In experiments, Exp-7 discusses the parameter sensitivity. After that, we derive a representative embedding $\hat{\mathbf{h}}_j$ for each group s_j by calculating the centroid of its members' embeddings:

$$\hat{\mathbf{h}}_j = \frac{1}{|s_j|} \sum_{i \in s_j} \mathbf{h}_i. \tag{6}$$

These k group representatives are then collected into a final group-level embedding matrix, $\mathbf{H}_g = [\hat{\mathbf{h}}_1, \ldots, \hat{\mathbf{h}}_k]^\top \in \mathbb{R}^{k \times d}$. We formalize this entire procedure as a single operator: $\mathbf{H}_g = \text{Grouping}(\mathbf{H}_c)$.

4.2.2 Group-based Attention Module. The Group-based Attention module (GroupAttn) computes influence scores between the identified behavioral groups. We first project the group representative matrix $\mathbf{H}_g$ into query ($\mathbf{Q}$), key ($\mathbf{K}$), and value ($\mathbf{V}$) spaces using learnable weight matrices $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{d \times d}$. The raw, unnormalized attention scores between group representatives are then computed via a scaled dot-product as follows:

$$\mathbf{O} = \frac{(\mathbf{H}_g \mathbf{W}^Q)(\mathbf{H}_g \mathbf{W}^K)^\top}{\sqrt{d}}. \tag{7}$$

A naive application of the standard softmax to the score matrix $\mathbf{O}$ would treat each group as a single entity, ignoring that real influence is jointly determined by *who participates* and *how many participate*. The former factor (role importance) is already encoded in the semantic embeddings of group representatives; the latter (collective magnitude) must be explicitly modeled to avoid underestimating weak but large groups, as discussed *"rumors can be frightening"* phenomenon in Section 1. To incorporate both factors, we apply GroupSoftmax, a size-aware normalization function that faithfully models the aggregated influence of all individuals within each group. Specifically, the contribution of each target group s_μ in the normalization term is weighted by its cardinality $|s_\mu|$:

$$\tilde{A}_{ij} = \text{GroupSoftmax}(\mathbf{O})_{ij} = \frac{|s_j| \exp(O_{ij})}{\sum_{\mu=1}^{k} |s_\mu| \exp(O_{i\mu})}. \tag{8}$$

Hence, GroupSoftmax restores a principled balance between semantic importance (encoded in $\hat{k}_j$) and collective magnitude ($|s_j|$). The resulting size-aware attention scores yield the final context-dependent group representations:

$$\tilde{\mathbf{H}}_g = \tilde{\mathbf{A}}(\mathbf{H}_g \mathbf{W}^V), \tag{9}$$

where $\tilde{\mathbf{H}}_g = [\tilde{\mathbf{h}}_1, \ldots, \tilde{\mathbf{h}}_k]^\top \in \mathbb{R}^{k \times d}$ contains the updated, influence-aware embeddings for all behavioral groups. Finally, to obtain an integrated representation for the entire cascade, we perform a weighted pooling operation over these updated group representations, with weights determined by group sizes:

$$\tilde{\mathbf{h}}^c = \text{Pooling}(\tilde{\mathbf{H}}_g) = \frac{1}{m} \sum_{j=1}^{k} |s_j| \tilde{\mathbf{h}}_j. \tag{10}$$

The resulting vector $\tilde{\mathbf{h}}^c \in \mathbb{R}^d$ serves as the final cascade representation used for the prediction task. We further formalize a theoretical analysis, proving that GroupAttn approximates the standard Self-Attention used in most competitors [7, 8, 24, 27, 31, 39, 43].

Figure 3: Illustration of the Group-based Attention module.

Example 1: As illustrated in Figure 3, users with similar embeddings (*e.g.*, the second and fourth participants) often produce nearly identical rows and columns in the attention matrix. Rather than computing redundant pairwise interactions A_{ij} at the individual level, these participants can be clustered into a diffusion group (*e.g.*, Group 2), represented by the centroid $\hat{\mathbf{h}}_g$. The group-level influence is then computed via the representative interaction $O_{xy} = \hat{\mathbf{h}}_x^\top \hat{\mathbf{h}}_y$, which approximates the original fine-grained scores. This abstraction can reduce the complexity from $O(m^2)$ to $O(k^2)$ with $k \ll m$, dramatically lowering computational costs while preserving the essential diffusion patterns.

Theorem 1: *Let $\mathbf{h}^c$ be the cascade embedding from standard self-attention and $\tilde{\mathbf{h}}^c$ be from our GroupAttn. Let $s(i)$ be the group of participant i, and let $\hat{\mathbf{h}}_{s(i)}$ be its representative embedding. Assume embeddings are on a unit hypersphere ($\|\mathbf{h}\| \le 1$), projections $\boldsymbol{W}^{\{Q,K,V\}}$ are orthogonal, and the intra-group error $\|\mathbf{h}_i - \hat{\mathbf{h}}_{s(i)}\| \le \delta$. Then the error is linearly bounded by δ: $\|\mathbf{h}^c - \tilde{\mathbf{h}}^c\| \le (1 + \frac{2}{\sqrt{d}})\delta = O(\delta)$.*

Corollary 1.1: *[Correctness] If all users within any given group have identical input embeddings, the output embedding from GroupAttn is identical to that of standard self-attention, i.e., $\tilde{\boldsymbol{h}}^c = \boldsymbol{h}^c$.*

Due to the space limitation, refer to the proof in Appendix B.

4.3 Group-wise Prediction

Given the final cascade representation $\tilde{\mathbf{h}}^c \in \mathbb{R}^d$, our goal is to predict the next *meaningful group* of participants. Unlike conventional next-user classification, which forces the model to select a single winner, group evolution is inherently *multi-positive*: multiple users can plausibly co-appear in the next step. Therefore, we formulate next-group prediction as a *ranking* problem, where the model learns to assign higher scores to users from the true future group than to invalid alternatives.

To anchor predictions and stabilize training, we retain a standard cross-entropy loss for next-user prediction as an auxiliary objective, following prior work. The core signal is group-wise and ranking-based. Specifically, we adopt the Bayesian Personalized Ranking (BPR) framework [41] and minimize the loss over triplets $\mathcal{D}$:

$$\mathcal{L}(\mathcal{D}) = -\sum_{(c,i,j)\in\mathcal{D}} \ln \sigma\big(\pi(i,c) - \pi(j,c) - \gamma\big), \tag{11}$$

where $\pi(u,c)$ is a linear scorer that computes the relevance between a candidate user u and the cascade state $\tilde{\mathbf{h}}^c$, and γ is a margin term. Upon convergence, the learned scoring function $\pi(\cdot,\cdot)$ serves as the criterion for *Group Selection*. To facilitate comparison with individual-level methods, we output the top-K scoring users as the predicted future group, which empirically correspond to coherent behavioral group semantics. This ranking objective addresses a key limitation of classification-based cross-entropy: the *winner-take-all* competition among candidates. In particular, cross-entropy concentrates probability mass on a single user, implicitly penalizing other legitimate members of the same future group, which can fragment the predicted set when we later take top-K. In contrast, BPR directly enforces *relative* preferences ($i \succ j$) and naturally supports *multi-positive supervision*: we treat all users in the next group as positives and encourage each of them to outrank negatives. As a result, the model learns a scoring landscape where true group members are consistently lifted together, yielding higher group integrity and more coherent top-K retrieval.

Table 1: Statistics of four evaluation datasets.

Datasets	# Users	# Links	# Cascades	# Avg. Len	# Max Len†
Quora	4,578	55,698	1,267	34.60	494
Twitter	12,627	309,631	3,434	32.64	491
Douban	12,232	396,580	3,475	21.76	489
Weibo	31,061	294,756	1,910	176.47	8,188

Complexity Analysis. Let m be the cascade length, k be the number of groups, d be the embedding dimension. The time complexity of our core *GroupAttn* module is determined by two main steps: (1) Dynamic Grouping: The k-Means algorithm partitions the m user embeddings into k groups. This iterative process has a time complexity of $O(k \cdot m \cdot d)$. (2) Attention Computation: Attention is then computed on the k group representative vectors, resulting in a complexity of $O(k^2 \cdot d + k \cdot d^2)$. In typical diffusion scenarios, the cascade length is the dominant factor, where $m \gg k$ and $m \gg d$. Consequently, the overall complexity is asymptotically dominated by $O(k \cdot m \cdot d)$, which scales linearly with the cascade length m.

5 Experiments

5.1 Experimental Settings

Dataset Description: Experiments are conducted on four real-world datasets containing static social networks and diffusion cascades: *Quora* [25], *Twitter* [14], *Douban* [42], and *Weibo* [40]. Key statistics are presented in Table 1. We adopt a standard temporal splitting methodology, arranging cascades chronologically and partitioning them into training (70%), validation (10%), and testing (20%) sets to prevent look-ahead bias. The influence matrix for our graph augmentation algorithm is constructed using only the training data. The ground truth for a given cascade is constructed by taking the observed last 10% of activated users, ordered chronologically, as the target group s_{k+1}. †**In Exp-3, we additionally synthesize much longer sequences only for scalability evaluation.**

Evaluation Metrics: We evaluate the prediction as a ranking task and use two standard metrics, **Recall@K** and **NDCG@K**, which are averaged across all users in the ground-truth set. Recall@K measures the hit rate in the top-K list, while NDCG@K is a rank-aware metric that rewards higher placement of correct users.

Table 2: *Next Group Prediction* on four datasets. Best results are in boldface, and second-best results are underlined.

Methods	Quora				Twitter				Douban				Weibo			
	R@50	N@50	R@100	N@100	R@50	N@50	R@100	N@100	R@50	N@50	R@100	N@100	R@50	N@50	R@100	N@100
DyHGCN	0.2168	0.0981	0.3026	0.1174	0.1075	0.0895	0.1255	0.1063	0.1623	0.1092	0.2209	0.1378	0.0723	0.0465	0.0871	0.0511
MSHGAT	0.2255	0.1014	0.3118	0.1208	0.1103	0.0908	0.1375	0.1072	0.1686	0.1041	0.2265	0.1320	0.0738	0.0481	0.0900	0.0515
DisenIDP	0.2332	0.1048	0.3191	0.1242	0.1327	0.1020	0.1597	0.1077	0.1730	0.1173	0.2339	0.1471	0.0759	0.0492	0.0923	0.0535
RotDiff	0.2393	0.1089	0.3260	0.1277	0.1345	0.1024	0.1609	0.1085	0.1787	0.1196	0.2297	0.1421	0.0775	0.0506	0.0946	0.0539
MGCL	0.2254	0.0931	0.3123	0.1100	0.1269	0.1042	0.1532	0.1094	0.1841	0.1326	0.2361	0.1554	0.0791	0.0521	0.0990	0.0555
MINDS	0.2326	0.0958	0.3185	0.1231	0.1183	0.0943	0.1564	0.1006	0.1884	0.1345	0.2300	0.1487	0.0804	0.0537	0.1034	0.0574
GODEN	0.2391	0.1086	0.3253	0.1321	0.1408	0.1051	0.1700	0.1113	0.1900	0.1465	0.2442	0.1504	0.0722	0.0543	0.1076	0.0593
CARE	0.2230	0.0996	0.3104	0.1273	0.1424	0.1063	0.1722	0.1115	0.1847	0.1373	0.2372	0.1523	0.0744	0.0553	0.1098	0.0602
PMRCA	0.2404	0.1114	0.3312	0.1312	0.1556	0.1131	0.1869	0.1210	0.2003	0.1467	0.2490	0.1598	0.0873	0.0535	0.1121	0.0633
SILN	0.2463	0.1131	0.3342	0.1343	0.1570	0.1125	0.1904	0.1218	0.1955	0.1477	0.2483	0.1608	0.0903	0.0560	0.1101	0.0604
GRID (ours)	**0.2697**	**0.1232**	**0.3528**	**0.1416**	**0.1851**	**0.1438**	**0.2117**	**0.1519**	**0.2187**	**0.1579**	**0.2624**	**0.1690**	**0.0961**	**0.0619**	**0.1216**	**0.0685**
Improv.	+9.50%	+8.93%	+5.57%	+5.44%	+17.90%	+27.14%	+11.19%	+24.51%	+9.19%	+6.90%	+5.38%	+5.10%	+6.43%	+10.54%	+8.47%	+8.22%

Compared Methods: We benchmark GRID against 10 state-of-the-art baselines: *DyHGCN* [39], *MSHGAT* [27], *DisenIDP* [7], *RotDiff* [24], *MINDS* [19], *MGCL* [8], *GODEN* [31], *CARE* [43], PMRCA [12], and SILN [9]. Note that we do not compare with the community-based *CLIPP* [20], as it aims to predict video-content inter-community propagation with multi-modal characteristics. In addition, to perform a fair and controlled analysis of computational efficiency, we compare GRID against its own ablated variant, `GRID-SA`. This variant replaces our GroupAttn with the standard Self-Attention mechanism used ubiquitously by the competitor models [7, 8, 24, 27, 31, 39, 43]. More implementation details and model configurations are provided in Appendix A.

5.2 Overall Performance

Exp-1: Next Group Prediction. As shown in Table 2, we first evaluate GRID against ten state-of-the-art competitors on the core task of predicting the next diffusion group. Since all competitors predict the next participant, we make necessary modifications of data processing and predictor to enable the group-based predictions, while keeping their graph and cascade encoders. GRID consistently and significantly outperforms all competitors across four real-world datasets. Compared to the strongest competitors, GRID achieves an average relative improvement of 10.65%, validating the effectiveness of group-based diffusion modeling. GRID can capture a longer diffusion context and model the many-to-many dynamics of real-world diffusion. Moreover, all competitors designed for next-user prediction struggle to adapt to group-level inference, leading to degraded performance. GRID is purpose-built for this task, with a group-wise objective for predicting coherent diffusion groups.

5.3 Efficiency and Scalability

Exp-2: Computational Efficiency. All models were evaluated on the same NVIDIA A100 GPU (80GB). Reported GPU memory corresponds to the peak usage during training. In Table 3, we compare GRID against the strongest baseline and its own variant `GRID-SA`, which replaces GroupAttn with the standard Self-Attention widely adopted in [9, 12, 27, 43]. For fair comparison, the maximum cascade length is capped at 2000, and the maximum group number is fixed to $k = 30$ for GRID (best performance). In terms of theoretical complexity, GRID requires only 1.07 GFLOPs, reflecting an order-of-magnitude reduction in floating-point operations over the strongest competitors. In practice, GRID attains an inference throughput of 670.11 samples/s, nearly 30× faster than SILN under the same hardware configuration. The peak GPU memory drops from 9.08 GB (SILN) to only 3.83 GB, showing the model's ability to handle long cascades with limited resources. Moreover, GRID reduces the per-batch inference latency to 47.63 ms, whereas SILN requires more than 1.3 s on average. Taken together, GRID offers a practical solution for long-context information diffusion analysis.

Table 3: Efficiency comparison on the largest Weibo dataset.

Methods	FLOPs ↓	GPU Mem ↓	Runtime / batch ↓	Throughput / s ↑
SILN	32.57 G	9.08 GB	1359.13 ms	22.95
GRID-SA	8.77 G	5.44 GB	519.46 ms	61.45
GRID	**1.07 G**	**3.83 GB**	**47.63 ms**	**670.11**

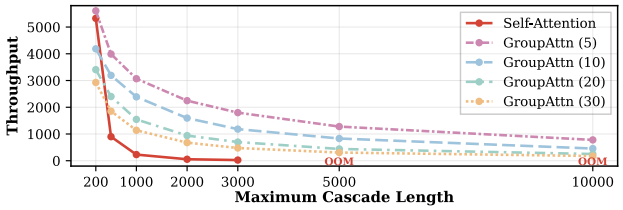


Figure 4: Scalability on long cascades. For `length` = 10000, GroupAttn (k=30) can still handle ~176 cascades per second.

Exp-3: Scalability on Long-context Diffusion. Figure 4 shows inference throughput against cascade length to evaluate scalability. The self-attention baseline (*i.e.*, GRID-SA), with its $O(m^2)$ complexity, shows a severe quadratic decay, dropping to ~25 cascades/sec at length 3,000 before failing with an Out-of-Memory (OOM) error at 5,000. In contrast, our GroupAttn (with varying k) exhibits more-linear scalability due to its $O(k \cdot m \cdot d)$ complexity where $k \ll m$. While the practical k-Means implementation causes a deviation from perfect linearity, this efficiency allows GRID to maintain a high throughput of ~176 cascades/sec at length 10,000, where the baseline GRID-SA fails. At the *equivalent throughput*, GroupAttn can increase the maximum supported cascade length by 10× than Self-Attention for long-context cascades (*e.g.*, 1k → 10k).

Table 4: Ablation studies of key components.

Dataset	Quora		Weibo	
Metrics	Recall@100	NDCG@100	Recall@100	NDCG@100
w/o DG	0.3386	0.1374	0.1174	0.0634
w/o IB	0.3324	0.1294	0.1160	0.0611
w/o GI	0.3352	0.1308	0.1145	0.0621
w/o GS	0.3401	0.1311	0.1183	0.0642
w/o GO	0.3286	0.1254	0.1158	0.0609
GI→CD	0.3379	0.1291	0.1166	0.0616
GRID	**0.3528**	**0.1416**	**0.1216**	**0.0685**

5.4 Ablation Study

Exp-4: Ablation on Graph Modules. In Table 4, removing the *Denoised Graph (DG)* causes a severe performance drop, demonstrating that filtering noisy connections from the raw social graph is fundamental to identifying coherent groups. Discarding the *Information Bottleneck (IB)* objective also hurts performance. By constraining representations to retain only group-relevant information, the IB term preserves meaningful behavioral regularities shared among group members, improving group identification.

Exp-5: Ablation on Cascade Modules. Replacing our *Group Identification (GI)* with fixed-budget random-sampling lowers performance, proving the superiority of dynamic grouping over a generic historical context. Similarly, substituting our *GroupSoftmax (GS)* with standard Softmax degrades performance, validating its importance for modeling group-internal dynamics. Finally, using a conventional next-user loss instead of our *Group-wise Objective (GO)* proves the inapplicability of the individual assumption.

Exp-6: Comparison with Detected Community. We design a *GI→CD* variant, which replaces our dynamic grouping with predefined communities from the classical Louvain algorithm [2]. The sharp performance decline shows that static communities are poor proxies for transient, cascade-specific diffusion groups, validating the need for dynamic grouping within each cascade.

5.5 Sensitivity Analysis

Exp-7: Impact of Group Tightness. Figure 5 analyzes the maximum group number k, which also controls the intra-group tightness δ used in Theorem 1. Intuitively, increasing k decreases intra-group variance and thus reduces δ. We observe that a modest k (about 20–30) yields competitive performance, matching full self-attention while remaining substantially more efficient (see Table 3). Interestingly, although a smaller k theoretically increases the bound $O(\delta)$, in practice this reduction sometimes acts as a beneficial regularizer that filters noisy, low-signal individual fluctuations.

Exp-8: Impact of Denoised Pruning α. This parameter controls the aggressiveness of graph augmentation. As shown in Fig. 6(a), performance generally improves as more low-interactive edges are pruned. Our experiments indicate that pruning up to 10% of edges consistently yields performance benefits. Moreover, pruning 20% of edges maintains comparable performance while improving efficiency due to the sparser graph removing spurious connections.

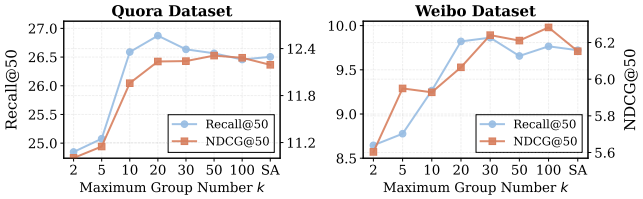


Figure 5: Sensitivity to maximum group number k. "SA" indicates Self-Attention. A few groups can be satisfiable.

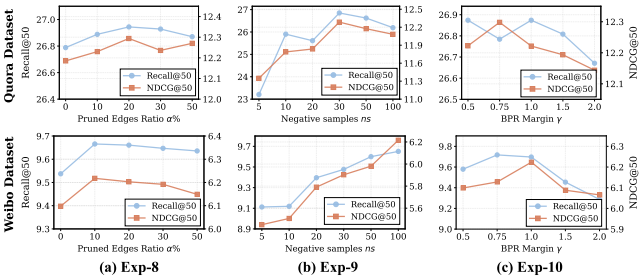


Figure 6: Hyperparameter sensitivity on $\{\alpha, |ns|, \gamma\}$.

Exp-9: Impact of Negative Sample Size ns. The number of negative samples is crucial for our BPR-based group-wise loss. Fig. 6(b) shows performance peaking at a moderate value. Too few samples provide insufficient contrastive signal for the discriminative function, while too many can lead to diminishing returns at a high computational cost. Notably, we find that larger datasets like Weibo, with their vast candidate pools, benefit from more negative samples to ensure a sufficiently diverse set of non-target instances.

Exp-10: Impact of BPR Margin γ. The margin γ sets the desired separation between positive and negative sample scores. As shown in Fig. 6(c), a margin of 1.0 is optimal in our experiments. A small margin provides a weak learning signal, resulting in a lack of discriminative power. Conversely, an overly large margin imposes an excessively strict discrimination that causes training instability.

6 Conclusion and Future Work

This paper introduces a paradigm shift to group-based information diffusion prediction and presents a purpose-built GRID framework. At its core, the *GroupAttn* module captures the influence in linear-time, supported by a theoretical error guarantee. This innovation, combined with a noise-resilient graph embedding module and a group-wise optimization objective, enables GRID to analyze long-context cascades without truncation. Extensive experiments confirm its superiority in terms of quality, scalability, and efficiency.

Acknowledgments

This work is partially supported by the National Key R&D Program of China (No. 2023YFB4503602), the National Natural Science Foundation of China (Nos. 62472304, 62436001), and the Hong Kong RGC Projects (No. 12200424). We would also like to express our gratitude to the anonymous reviewers for their valuable feedback.

References

[1] Jay J Van Bavel, Katherine Baicker, Paulo S Boggio, Valerio Capraro, Aleksandra Cichocka, Mina Cikara, Molly J Crockett, Alia J Crum, Karen M Douglas, James N Druckman, et al. 2020. Using social and behavioural science to support COVID-19 pandemic response. *Nature human behaviour* 4, 5 (2020), 460–471.

[2] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008 (2008), P10008. https://api.semanticscholar.org/CorpusID:334423

[3] Alexandre Bovet and Hernán A Makse. 2019. Influence of fake news in Twitter during the 2016 US presidential election. *Nature communications* 10, 1 (2019), 7.

[4] Jan B Broekaert, Davide La Torre, and Faizal Hafiz. 2022. Competing control scenarios in probabilistic SIR epidemics on social-contact networks. *Annals of Operations Research* (2022), 1–24.

[5] Damon Centola. 2010. The spread of behavior in an online social network experiment. *Science* 329, 5996 (2010), 1194–1197.

[6] Jiangzhuo Chen, Stefan Hoops, Achla Marathe, and etc. 2022. Effective Social Network-Based Allocation of COVID-19 Vaccines. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*. ACM, 4675–4683.

[7] Zhangtao Cheng, Wenxue Ye, Leyuan Liu, Wenxin Tai, and Fan Zhou. 2023. Enhancing Information Diffusion Prediction with Self-Supervised Disentangled User and Cascade Representations. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 3808–3812.

[8] Zihan Feng, Rui Wu, Yajun Yang, Hong Gao, Xin Wang, Xueli Liu, and Qinghua Hu. 2024. Multi-level Contrastive Learning on Weak Social Networks for Information Diffusion Prediction. In *Database Systems for Advanced Applications - 29th International Conference, DASFAA 2024*.

[9] Zihan Feng, Yajun Yang, Xin Huang, Hong Gao, Liping Jing, and Qinghua Hu. 2025. Efficient Sphere-Effect Based Information Diffusion Prediction on Large-scale Social Networks. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25)*. New York, NY, USA, 615–625.

[10] Jinfei Gao, Xiao Wang, Tian Gan, Jianhua Yin, Chuanchen Luo, and Liqiang Nie. 2025. Social Context-Aware Community-Level Propagation Prediction. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25)*. 1995–2005.

[11] Jiewen Guan, Xin Huang, and Bilian Chen. 2023. Community-Aware Social Recommendation: A Unified SCSVD Framework. *IEEE Trans. Knowl. Data Eng.* 35, 3 (2023), 2379–2393.

[12] Weikang He, Yunpeng Xiao, Mengyang Huang, Xuemei Mou, Rong Wang, and Qian Li. 2025. A Pattern-Driven Information Diffusion Prediction Model Based on Multisource Resonance and Cognitive Adaptation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25)*. 592–601.

[13] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *SIGIR 2020*. ACM, 639–648.

[14] Nathan Oken Hodas and Kristina Lerman. 2013. The Simple Rules of Social Contagion. *CoRR* abs/1308.5015 (2013). http://arxiv.org/abs/1308.5015

[15] Mohammad Raihanul Islam, Sathappan Muthiah, Bijaya Adhikari, B. Aditya Prakash, and Naren Ramakrishnan. 2018. DeepDiffuse: Predicting the 'Who' and 'When' in Cascades. In *IEEE International Conference on Data Mining, ICDM 2018*. 1055–1060.

[16] Shuo Ji, Xiaodong Lu, Mingzhe Liu, Leilei Sun, Chuanren Liu, Bowen Du, and Hui Xiong. 2023. Community-based Dynamic Graph Learning for Popularity Prediction. In *KDD 2023*. ACM, 930–940.

[17] Xueqi Jia, Jiaxing Shang, Dajiang Liu, Haidong Zhang, and Wancheng Ni. 2022. HeDAN: Heterogeneous diffusion attention network for popularity prediction of online content. *Knowl. Based Syst.* 254 (2022), 109659.

[18] Runhao Jiang, Renchi Yang, and Wenqing Lin. 2025. Community-Aware Social Community Recommendation. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM '25)*. 1179–1189.

[19] Pengfei Jiao, Hongqian Chen, Qing Bao, Wang Zhang, and Huaming Wu. 2024. Enhancing Multi-Scale Diffusion Prediction via Sequential Hypergraphs and Adversarial Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 8571–8581.

[20] Yiqiao Jin, Yeon-Chang Lee, Kartik Sharma, Meng Ye, Karan Sikka, Ajay Divakaran, and Srijan Kumar. 2023. Predicting information pathways across online communities. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1044–1056.

[21] Junliang Li, Yajun Yang, Qinghua Hu, Xin Wang, and Hong Gao. 2023. Public Opinion Field Effect Fusion in Representation Learning for Trending Topics Diffusion. In *Thirty-seventh Conference on Neural Information Processing Systems*.

[22] Xiaodong Lu, Shuo Ji, Le Yu, Leilei Sun, Bowen Du, and Tongyu Zhu. 2023. Continuous-Time Graph Learning for Cascade Popularity Prediction. In *IJCAI 2023*. 2224–2232.

[23] Mingyu Derek Ma, Alexander K Taylor, Nuan Wen, Yanchen Liu, Po-Nien Kung, Wenna Qin, Shicheng Wen, Azure Zhou, Diyi Yang, Xuezhe Ma, et al. 2024. MIDDAG: Where Does Our News Go? Investigating Information Diffusion via Community-Level Information Pathways. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 23811–23813.

[24] Hongliang Qiao, Shanshan Feng, Xutao Li, Huiwei Lin, Han Hu, Wei Wei, and Yunming Ye. 2023. RotDiff: A Hyperbolic Rotation Representation Model for Information Diffusion Prediction. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 2065–2074.

[25] Aravind Sankar, Xinyang Zhang, Adit Krishnan, and Jiawei Han. 2020. Inf-VAE: A Variational Autoencoder Framework to Integrate Homophily and Influence in Diffusion Prediction. In *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, 2020*. ACM, 510–518.

[26] Wenbo Shang, Zihan Feng, Yang Yajun, and Xin Huang. 2025. Make Information Diffusion Explainable: LLM-based Causal Framework for Diffusion Prediction. In *Proceedings of the 2025 Conference on Neural Information Processing Systems (NeurIPS 2025)*. https://neurips.cc/virtual/2025/poster/117336

[27] Ling Sun, Yuan Rao, Xiangbo Zhang, Yuqian Lan, and Shuanghe Yu. 2022. MS-HGAT: Memory-Enhanced Sequential Hypergraph Attention Network for Information Diffusion Prediction. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022*. AAAI Press, 4156–4164.

[28] Xiangguo Sun, Hong Cheng, Bo Liu, Jia Li, Hongyang Chen, Guandong Xu, and Hongzhi Yin. 2023. Self-Supervised Hypergraph Representation Learning for Sociological Analysis. *IEEE Trans. Knowl. Data Eng.* 35, 11 (2023), 11860–11871.

[29] Youchen Sun, Zhu Sun, Yingpeng Du, Jie Zhang, and Yew Soon Ong. 2024. Self-Supervised Denoising through Independent Cascade Graph Augmentation for Robust Social Recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2806–2817.

[30] Naftali Tishby, Fernando C Pereira, and William Bialek. 2000. The information bottleneck method. *arXiv preprint physics/0004057* (2000).

[31] Ding Wang, Wei Zhou, and Songlin Hu. 2024. Information Diffusion Prediction with Graph Neural Ordinary Differential Equation Network. In *MM 2024*. ACM, 9699–9708.

[32] Yuhang Wang, Wei Zhou, Ziang Hu, Jizhong Han, and Tao Guo. 2024. CSFI for Social Media: Understanding and Predicting Cross-Community Information Propagation. In *2024 IEEE 36th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 264–269.

[33] Zhitao Wang, Chengyao Chen, and Wenjie Li. 2018. A Sequential Neural Information Diffusion Model with Structure Attention. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018*. ACM, 1795–1798.

[34] L. Weng, A. Flammini, A. Vespignani, and F. Menczer. 2012. Competition among Memes in a World with Limited Attention. *Scientific Reports* 2, 335 (2012).

[35] Jiyoung Woo, Jaebong Son, and Hsinchun Chen. 2011. An SIR model for violent topic diffusion in social media. In *2011 IEEE International Conference on Intelligence and Security Informatics, ISI 2011*. IEEE, 15–19.

[36] Lianghao Xia, Yizhen Shao, Chao Huang, Yong Xu, Huance Xu, and Jian Pei. 2023. Disentangled Graph Social Recommendation. In *39th IEEE International Conference on Data Engineering, ICDE 2023*. IEEE, 2332–2344.

[37] Cheng Yang, Maosong Sun, Haoran Liu, Shiyi Han, Zhiyuan Liu, and Huanbo Luan. 2021. Neural Diffusion Model for Microscopic Cascade Study. *IEEE Trans. Knowl. Data Eng.* 33, 3 (2021), 1128–1139.

[38] Cheng Yang, Jian Tang, Maosong Sun, Ganqu Cui, and Zhiyuan Liu. 2019. Multi-scale Information Diffusion Prediction with Reinforced Recurrent Networks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019*. 4033–4039.

[39] Chunyuan Yuan, Jiacheng Li, Wei Zhou, Yijun Lu, Xiaodan Zhang, and Songlin Hu. 2020. DyHGCN: A Dynamic Heterogeneous Graph Convolutional Network to Learn Users' Dynamic Preferences for Information Diffusion Prediction. In *ECML PKDD 2020*, Vol. 12459. Springer, 347–363.

[40] Jing Zhang, Biao Liu, Jie Tang, Ting Chen, and Juanzi Li. 2013. Social Influence Locality for Modeling Retweeting Behaviors. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence*. 2761–2767.

[41] Yingqi Zhao, Haiwei Zhang, Qijie Bai, Changli Nie, and Xiaojie Yuan. 2024. DHMAE: A Disentangled Hypergraph Masked Autoencoder for Group Recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 914–923.

[42] Erheng Zhong, Wei Fan, Junwei Wang, Lei Xiao, and Yong Li. 2012. ComSoc: adaptive transfer of user behaviors over composite social network. In *KDD '12*. ACM, 696–704.

[43] Ting Zhong, Jienan Zhang, Zhangtao Cheng, Fan Zhou, and Xueqin Chen. 2024. Information Diffusion Prediction via Cascade-Retrieved In-Context Learning. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2472–2476.

[44] Huaijie Zhu, Wei Liu, Jian Yin, Mengxiang Wang, Jianliang Xu, Xin Huang, and Wang-Chien Lee. 2022. Continuous Geo-Social Group Monitoring over Moving Users. In *38th IEEE International Conference on Data Engineering, ICDE 2022, Kuala Lumpur, Malaysia, May 9-12, 2022*. IEEE, 312–324.

Table 5: Model Configurations of GRID.

Parameter	Value	Description
d	{128, 256}	The embedding dimension.
L	{1, 2}	Number of the GNN layers.
B	{4, 6, 8}	Number of attention heads.
k	{10, 30, 50}	The maximum number of groups.
τ	[0.1, 1.0]	Temperature coefficient in Eq.(5).
β	[1, 5]	The weight of denoising trade-off parameter.
γ	[0.5, 2.0]	The margin for the BPR-based loss function.
$\|ns\|$	[30, 100]	Number of negative samples per positive instance.
Batch Size	{16, 32}	The number of samples per training batch.
Learning Rate	1e-3	The learning rate for the Adam optimizer.

A Experimental Setup

Datasets and Preprocessing. Following standard practice [11, 27, 40], we preprocess the data by filtering cascades with fewer than 10 participants and retaining only the first adoption for any user appearing multiple times in a cascade. Furthermore, we establish a standardized protocol for our group-based task. Each cascade is partitioned chronologically: (1) *Observed History (C):* The first 90% of participants, used as model input. (2) *Ground-Truth Group (s_{k+1}):* The final 10% of participants, used as the prediction target. This 90/10 temporal split aligns our evaluation with a well-documented temporal-clustering phenomenon [1, 9, 16, 17], in which the practical goal of forecast the next *wave* of adopters. In Exp-3, we synthesize extended cascades with longer sequences to evaluate the scalability through random padding users. These synthetic cascades are used **only** for scalability experiments.

Model Configurations. Table 5 lists the key hyperparameters of GRID. All predictive experiments are conducted on NVIDIA 4090 (24GB) GPUs, while efficiency experiments for handling ultra-long cascades are performed on A100 (80GB) GPUs. The model is trained using the Adam optimizer. To ensure robust results, we employ an early stopping strategy based on the validation, with a patience of 5 epochs and a maximum of 30 training epochs.

B Theoretical Analysis

We assume the projection matrices $\mathbf{W}^Q$, $\mathbf{W}^K$, and $\mathbf{W}^V$ are orthogonal. While these matrices are not strictly orthogonal in practice, this assumption provides *analytical tractability* by ensuring that distances and norms are preserved during projection, allowing us to isolate the error introduced by our GroupAttn approximation.

Theorem 1: *Let $\mathbf{h}^c$ be the cascade embedding from standard self-attention and $\tilde{\mathbf{h}}^c$ be from our GroupAttn. Let $s(i)$ denote the group of participant i, and let $\hat{\mathbf{h}}_{s(i)}$ be its representative embedding. Assume embeddings lie on a unit hypersphere ($\|\mathbf{h}\| \le 1$) and that the intra-group error is bounded by $\|\mathbf{h}_i - \hat{\mathbf{h}}_{s(i)}\| \le \delta$. Under the idealized condition that the projections $\mathbf{W}^Q$, $\mathbf{W}^K$, and $\mathbf{W}^V$ are orthogonal, the error is linearly bounded by δ: $\|\mathbf{h}^c - \tilde{\mathbf{h}}^c\| \le \left(1 + \frac{2}{\sqrt{d}}\right)\delta = O(\delta)$.*

The proof proceeds via two key lemmas that bound the error in the attention logits and the resulting context vectors.

Lemma 1.1: *[Logit Error Bound] Under the assumptions of Theorem 1, the absolute difference between the attention logit for any pair (i, j) and its GroupAttn approximation is bounded by 2δ:*

$$\left|\mathbf{q}_i^\top \mathbf{k}_j - \hat{\mathbf{q}}_{s(i)}^\top \hat{\mathbf{k}}_{s(j)}\right| \le 2\delta.$$

Proof. By adding and subtracting $\hat{\mathbf{q}}_{s(i)}^\top \mathbf{k}_j$ and applying the triangle and Cauchy–Schwarz inequalities, we have:

$$\begin{aligned}\left|\mathbf{q}_i^\top \mathbf{k}_j - \hat{\mathbf{q}}_{s(i)}^\top \hat{\mathbf{k}}_{s(j)}\right| &= \left|(\mathbf{q}_i - \hat{\mathbf{q}}_{s(i)})^\top \mathbf{k}_j + \hat{\mathbf{q}}_{s(i)}^\top (\mathbf{k}_j - \hat{\mathbf{k}}_{s(j)})\right| \\ &\le \|\mathbf{q}_i - \hat{\mathbf{q}}_{s(i)}\|\,\|\mathbf{k}_j\| + \|\hat{\mathbf{q}}_{s(i)}\|\,\|\mathbf{k}_j - \hat{\mathbf{k}}_{s(j)}\|.\end{aligned}$$

By the orthogonality of $\mathbf{W}^Q$ and $\mathbf{W}^K$, distances are preserved, hence $\|\mathbf{q}_i - \hat{\mathbf{q}}_{s(i)}\| = \|\mathbf{h}_i - \hat{\mathbf{h}}_{s(i)}\| \le \delta$ and $\|\mathbf{k}_j - \hat{\mathbf{k}}_{s(j)}\| = \|\mathbf{h}_j - \hat{\mathbf{h}}_{s(j)}\| \le \delta$. Since all embeddings lie on a unit hypersphere, $\|\mathbf{k}_j\| \le 1$ and $\|\hat{\mathbf{q}}_{s(i)}\| \le 1$. Substituting these bounds yields:

$$\left|\mathbf{q}_i^\top \mathbf{k}_j - \hat{\mathbf{q}}_{s(i)}^\top \hat{\mathbf{k}}_{s(j)}\right| \le \delta \cdot 1 + 1 \cdot \delta = 2\delta.$$

□

Lemma 1.2: *[Context Vector Error Bound] The norm of the difference between the context vector $\mathbf{c}_i$ and its approximation $\hat{\mathbf{c}}_i$ is bounded by*

$$\|\mathbf{c}_i - \hat{\mathbf{c}}_i\| \le \left(1 + \frac{2}{\sqrt{d}}\right)\delta. \tag{12}$$

Proof. We decompose the error using the triangle inequality:

$$\|\mathbf{c}_i - \hat{\mathbf{c}}_i\| \le \underbrace{\left\|\sum_{j=1}^{m}(A_{ij} - \hat{A}_{ij})\mathbf{v}_j\right\|}_{\text{Attention Error}} + \underbrace{\left\|\sum_{j=1}^{m}\hat{A}_{ij}(\mathbf{v}_j - \hat{\mathbf{v}}_{s(j)})\right\|}_{\text{Value Error}}.$$

(1) Value Error. Since $\sum_j \hat{A}_{ij} = 1$ and $\hat{A}_{ij} \ge 0$, this term is a convex combination of the value errors. By orthogonality of $\mathbf{W}^V$, we have $\|\mathbf{v}_j - \hat{\mathbf{v}}_{s(j)}\| = \|\mathbf{h}_j - \hat{\mathbf{h}}_{s(j)}\| \le \delta$, thus Value Error is bounded by δ.
(2) Attention Error. Since $\|\mathbf{v}_j\| \le 1$, this term is bounded by the L_1 distance between attention distributions:

$$\|\mathbf{A}_{i,:} - \hat{\mathbf{A}}_{i,:}\|_1.$$

The softmax function with scaling $1/\sqrt{d}$ is Lipschitz continuous, mapping the L_∞ norm on its inputs to the L_1 norm on its outputs. Hence,

$$\|\mathbf{A}_{i,:} - \hat{\mathbf{A}}_{i,:}\|_1 \le \frac{1}{\sqrt{d}} \max_j \left|\mathbf{q}_i^\top \mathbf{k}_j - \hat{\mathbf{q}}_{s(i)}^\top \hat{\mathbf{k}}_{s(j)}\right|.$$

Applying Lemma 1.1, the Attention Error is bounded by $2\delta/\sqrt{d}$. Combining both bounds completes the proof:

$$\|\mathbf{c}_i - \hat{\mathbf{c}}_i\| \le \delta + \frac{2\delta}{\sqrt{d}} = \left(1 + \frac{2}{\sqrt{d}}\right)\delta.$$

□

Proof of Theorem 1. The total error equals the norm of the average difference of the context vectors. By the triangle inequality:

$$\|\mathbf{h}^c - \tilde{\mathbf{h}}^c\| = \left\|\frac{1}{m}\sum_{i=1}^{m}(\mathbf{c}_i - \hat{\mathbf{c}}_i)\right\| \le \frac{1}{m}\sum_{i=1}^{m}\|\mathbf{c}_i - \hat{\mathbf{c}}_i\| \le \max_i \|\mathbf{c}_i - \hat{\mathbf{c}}_i\|.$$

Applying Lemma 1.2 yields the desired result:

$$\|\mathbf{h}^c - \tilde{\mathbf{h}}^c\| \le \left(1 + \frac{2}{\sqrt{d}}\right)\delta.$$

□

Proof of Corollary 1.1. The identical condition implies that $\delta = \max_i \|\mathbf{h}_i - \hat{\mathbf{h}}_{s(i)}\| = 0$. Substituting $\delta = 0$ into the bound from Theorem 1 yields $\|\mathbf{h}^c - \tilde{\mathbf{h}}^c\| \le 0$, proving the identical outputs. □