Ghidorah: Towards Robust Multi-Scale Information Diffusion Prediction via Test-Time Training

Wenting Zhu[1], Chaozhuo Li[1], Litian Zhang[2], Senzhang Wang[3], Xi Zhang[1]*

[1]Key Laboratory of Trustworthy Distributed Computing and Service (MoE), Beijing University of Posts and Telecommunications, China
[2]School of Cyber Science and Technology, Beihang University, China
[3]School of Computer Science and Engineering, Central South University, China
{zwt, lichaozhuo}@bupt.edu.cn, litianzhang@buaa.edu.cn, szwang@csu.edu.cn, zhangx@bupt.edu.cn

Abstract

Information diffusion prediction (IDP) is a pivotal task for understanding the dynamics of information propagation within social networks. Conventional models typically adhere to a fixed learning-based paradigm, where the trained prediction model remains static during the inference phase. This paradigm presupposes that the data is independent and identically distributed, an assumption that may not hold true due to the inherently open nature of social media and the uncertainty and variability in user behavior. In this paper, we address the novel problem of out-of-distribution (OOD) shifts within IDP tasks and propose a new test-time training-based model for multi-scale IDP tasks, named Ghidorah. Our approach focuses on adapting a subset of model parameters to accommodate the unique characteristics of test samples through self-supervised learning (SSL) tasks. Ghidorah comprises three components: the macroscopic prediction branch, the microscopic prediction branch, and the auxiliary SSL branch. The auxiliary SSL task employs a masked autoencoder-based loss to fine-tune the model for specific test samples prior to prediction. Furthermore, Ghidorah integrates invariant learning to capture robust representations while mitigating spurious correlations. To our knowledge, Ghidorah is the first work to introduce a test-time training framework specifically designed to address the critical yet often overlooked OOD challenges in IDP tasks. Experimental results across several benchmark datasets validate the superiority of our approach.

Introduction

Online social media enables the effortless dissemination and reposting of news. The extensive tracking and recording of information cascades have driven researchers to investigate patterns of information spread within social networks, particularly through the task of information diffusion prediction (IDP). Current IDP research primarily concentrates on predicting future diffusion dynamics, including forecasting diffusion volume from a **macroscopic** perspective to estimate the future popularity of the entire cascade, as well as predicting individual adoption from a **microscopic** perspective to identify the potential next user (Li et al. 2024).

Macroscopic and microscopic predictions are traditionally treated as separate training objectives (Xu et al. 2021; Sun et al. 2022), whereas recent works propose their joint training paradigm as multi-scale prediction (Yang et al. 2021; Jiao et al. 2024). Both existing independent and multi-scale prediction models adhere to the traditional learning-based paradigm. As shown in Fig. 1, the prediction model is first trained on the training set, which is then fixed and evaluated on the test samples. The effectiveness of such paradigm hinges on the assumption that the data is independent and identically distributed (IID), meaning that the training and testing samples are drawn from the same distribution.

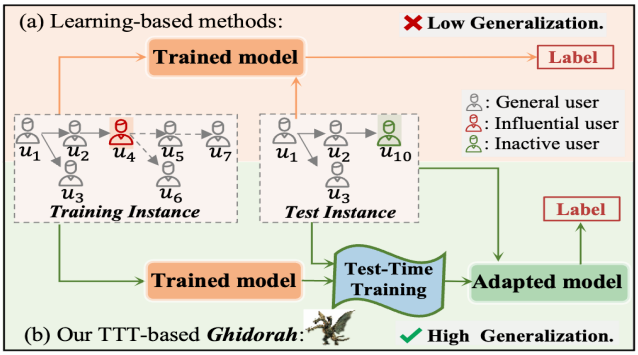

Figure 1: The comparison between the traditional learning-based methods and the proposed TTT-based model.

Nevertheless, the open nature of social media platforms, combined with the inherent uncertainty and variability in user behavior, often undermines the validity of the IID assumption in practical social networks (Li et al. 2018a, 2021a; Jia et al. 2022). For example, even minor modifications to the users within a cascade, given similar initial sharers, can lead to substantial changes in future predictions. As illustrated in Figure 1, two posts with comparable initial sharers may exhibit divergent propagation cascades: one may be amplified by an influential user u_4 as the training sample, while the other may be disseminated by a less active user u_{10} as the testing sample. Similarly, information with comparable diffusion volumes may display notable differences in themes and the demographics of users engaging with it (Zhou et al. 2021). Consequently, the knowledge gained from training cascades may be insufficient for mak-

*Corresponding author: Xi Zhang.

ing accurate predictions on the test set (Zhang et al. 2024a).

In this paper, we investigate the novel problem of out-of-distribution (OOD) shifts within IDP tasks, which arise from potential discrepancies between training and testing cascades due to the inherent uncertainty in social user behavior. A straightforward approach involves applying existing domain generalization techniques, such as adversarial robustness (Ganin et al. 2016) and domain adaptation (Long et al. 2016), to mitigate these shifts by leveraging the topological structure or incorporating data from the test distribution during training. However, these methods, which aim to learn domain-agnostic features during training, may be less effective in real-world social networks with divergent properties (Gulrajani and Lopez-Paz 2020). Moreover, they often neglect the utility of testing data beyond mere evaluation.

Inspired by the success of test-time training (TTT) techniques in computer vision (Sain et al. 2022; Chen et al. 2022; Hatem, Qian, and Wang 2023), we focus on the test phase to develop a novel TTT-based approach for generalization by learning robust characteristics of test cascades. As illustrated in Figure 1, our method integrates an auxiliary self-supervised learning (SSL) task to update model parameters during the test phase, allowing the model to adapt to each test instance. Unlike conventional domain generalization models, which are confined to the training phase and anticipate distribution shifts, TTT extends model adaptation into the test phase by leveraging the available test data.

However, designing an effective TTT framework for IDP tasks poses several challenges. First, the design of the SSL task must be general enough to generate features capable of predicting diffusion patterns across various distributions. Second, TTT's tendency to adjust shared parameters based on a single testing cascade may cause discrepancies between the SSL task and IDP tasks. Third, relying only on TTT to address OOD challenges may be inadequate due to variability in user behaviors. A representation learning module that capitalizes on invariant relationships between cascade features and annotations, while minimizing the impact of spurious correlations, is anticipated to be more effective.

To address the aforementioned challenges, we propose a robust IDP model based on the TTT framework, comprising three primary branches: the macroscopic prediction branch, the microscopic prediction branch, and the auxiliary SSL branch. These branches share a common feature extractor, with each possessing its own prediction head, similar to the multi-headed **Ghidorah** from the movie *Godzilla: King of the Monsters*. Inspired by the Masked Autoencoder (MAE) loss, we introduce an MAE-like loss specifically designed for IDP tasks as the auxiliary task, aimed at capturing the intrinsic correlations among users within a single cascade. Unlike the vanilla TTT framework (Sun et al. 2020), Ghidorah integrates a consistency loss for test-time training, ensuring that all tasks are optimized in a cohesive direction. Furthermore, an invariant learning component is incorporated to learn cascade representations that remain stable across various environments, thereby enhancing Ghidorah's generalization capability. Our proposal is extensively evaluated across four datasets and two tasks, demonstrating superior performance compared to sixteen SOTA baselines.

Our major contributions are summarized as follows:

- To the best of our knowledge, we are the first to introduce the test-time training framework to address the critical yet overlooked OOD challenges within IDP tasks.
- We propose a novel TTT-based model, Ghidorah, specifically designed for IDP tasks. This model incorporates several innovative modules, including invariant learning and consistency loss, to overcome the limitations of vanilla TTTs.
- Comprehensive experiments demonstrate that Ghidorah consistently outperforms sixteen SOTA approaches.

Problem Formulation

Diffusion Cascade Given the user set $\mathcal{U}$, a cascade $c_i = \{(u_1^i, t_1^i), (u_2^i, t_2^i), \ldots, (u_{|c_i|}^i, t_{|c_i|}^i)\}$ records the diffusion process of the information item i in chronological order, where the tuple (u_j^i, t_j^i) indicates that the user u_j forwards the current information i at a certain timestamp t_j. All observed historical cascades are denoted as $\mathcal{C} = \{c_i\}$.

Graph Construction The social graph $\mathcal{G}_S = (\mathcal{U}, \mathcal{E})$ is a directed graph, where $\mathcal{U}$ is the user set and $\mathcal{E}$ is the edge set representing social relationships among users. The observed diffusion cascades $\mathcal{C} = \{c_1, c_2, \ldots, c_M\}$, $|\mathcal{C}| = M$ are split into T subsets based on timestamps to construct sequential diffusion hypergraphs $\mathcal{G}_D = \{\mathcal{G}_D^t | t = 1, 2, \ldots, T\}$, $\mathcal{G}_D^t = (\mathcal{U}^t, \mathcal{E}^t)$, where $\mathcal{U}^t$ is the user set and $\mathcal{E}^t$ is the hyperedge set. In each diffusion hypergraph $\mathcal{G}_D^t$, a hyperedge connects users who participate in the same cascade only during the t-th time interval, which represents the dynamic diffusion process of the cascade.

Definition 1. Macroscopic Prediction Given a social graph $\mathcal{G}_S$, diffusion hypergraphs $\mathcal{G}_D$ and an observed snapshot of cascade c_i at time t_o, we aim to predict the final size $|c_i|$ (a.k.a. popularity) of this cascade, i.e., the total number of users who perform the retweeting action to the original information after t_o.

Definition 2. Microscopic Prediction Given a social graph $\mathcal{G}_S$, diffusion hypergraphs $\mathcal{G}_D$, and an observed cascade snapshot, our goal is to predict which users are likely to express interest and repost in the subsequent step.

Methodology

The framework of the proposed Ghidorah model is illustrated in Figure 2. Given the input diffusion hypergraph $\mathcal{G}_D$ and social graph $\mathcal{G}_S$, the shared feature extractor θ_e is designed to learn user representations that capture both social relations and global interactions among users. These learned representations are then fed into three distinct branches: (1) an auxiliary branch θ_a, which employs self-supervised learning to refine the shared feature extractor for specific test cascades; (2) a main branch θ_p that applies a task-specific prediction head to perform the macroscopic prediction task; and (3) another main branch θ_m that similarly employs a task-specific prediction head to execute the microscopic prediction task. In the auxiliary branch, a masked autoencoder (MAE) is adapted as the self-reconstruction task to effectively mine and learn the underlying diffusion patterns and latent structures within cascades.

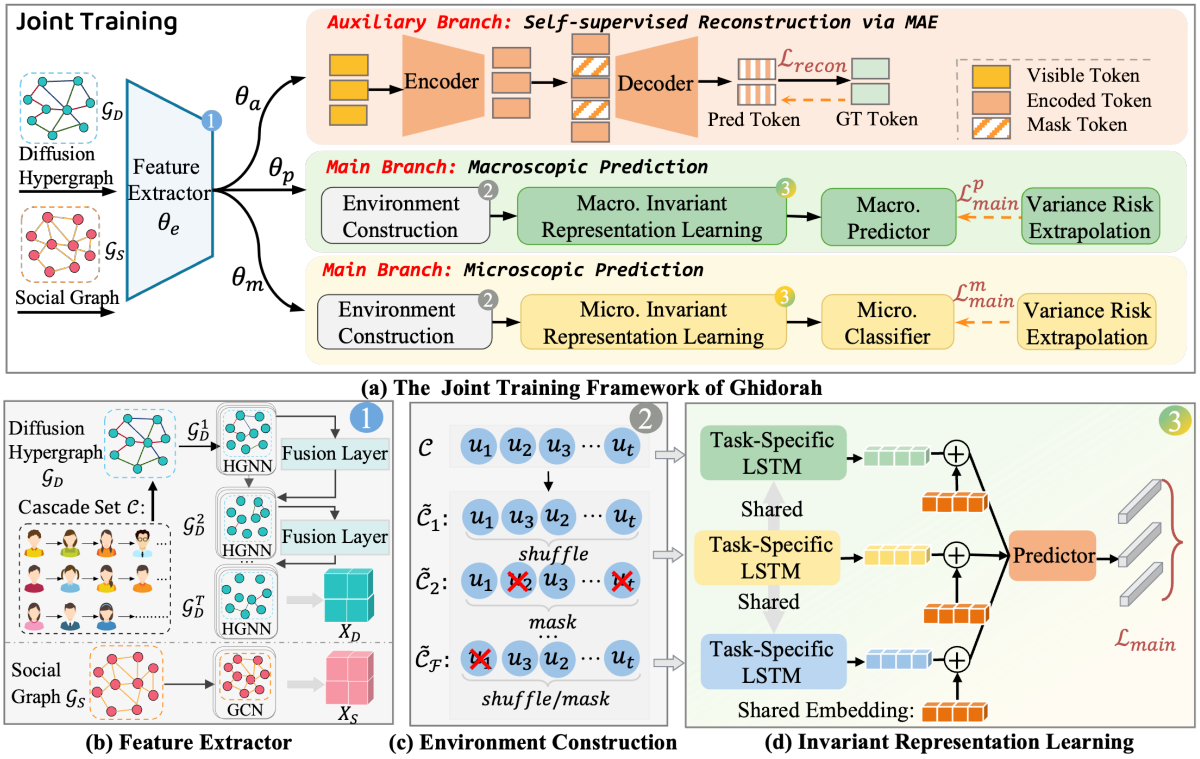

Figure 2: The joint training framework of Ghidorah.

Test-time Training Framework of Ghidorah

Following standard TTT model (Sun et al. 2020), the overall workflow consists of three phases: joint training, test-time training, and inference. Joint training occurs before test time, while test-time training and inference are executed repeatedly and sequentially during test time.

The Joint Training Phase During the joint training, we train all model parameters for the two diffusion prediction tasks (main tasks) and the auxiliary self-supervised task (SSL task) based on the training samples. For clarity, we define the objectives of the auxiliary task, macroscopic prediction task, and microscopic prediction task as $\mathcal{L}_{aux}$, $\mathcal{L}^p_{main}$, and $\mathcal{L}^m_{main}$, respectively. Ghidorah is jointly optimized by minimizing a weighted sum of $\mathcal{L}^p_{main}$, $\mathcal{L}^m_{main}$, $\mathcal{L}_{aux}$, and $\mathcal{L}_{align}$ as shown in Eq. (1), where α is a hyper-parameter:

$$\min_{\theta_e,\theta_a,\theta_p,\theta_m} \mathcal{L}^p_{main}(\theta_e,\theta_p) + \mathcal{L}^m_{main}(\theta_e,\theta_m) + \alpha\mathcal{L}_{aux}(\theta_e,\theta_a) + \mathcal{L}_{align}(\theta_e,\theta_a,\theta_p,\theta_m). \quad (1)$$

By minimizing this training loss, we achieve the well-trained parameters θ^*_e, θ^*_a, θ^*_p, and θ^*_m for the related modules.

The Test-time Training Phase As depicted in Figure 3, given a target cascade from the test set, we apply δ steps of gradient descent to fine-tune the parameters of the feature extractor θ^*_e by minimizing the auxiliary self-supervised reconstruction task loss $\mathcal{L}_{aux}$ as in Eq. (2). Note that the parameters of two main branches, θ^*_p and θ^*_m, remain fixed throughout the test-time training process:

$$\min_{\theta^*_e,\theta^*_a} \mathcal{L}_{aux}(\theta^*_e,\theta^*_a). \quad (2)$$

The learned parameters θ'_e and θ'_a are optimized for a specific testing cascade, allowing the model to develop a robust representation that effectively captures complex user interactions, even within previously unseen cascades.

The Inference Phase During the inference phase, the feature extractor θ'_e learned during test-time training, is combined with the main branches θ^*_p and θ^*_m to form two models: (θ'_e, θ^*_p) and (θ'_e, θ^*_m). The model (θ'_e, θ^*_p) is used for macroscopic prediction, while the model (θ'_e, θ^*_m) is used for microscopic prediction.

Shared Feature Extractor

As shown in the Figure 2(b), the shared feature extractor aims to learn representations for input cascades by leveraging both social following relationships and user interactions within diffusion cascades. Following previous works (Yuan et al. 2021; Li et al. 2017b), our framework employs a multi-layer Graph Convolutional Network (GCN) (Kipf and Welling 2016) and a Hypergraph Neural Network (HGNN) (Zhang et al. 2024c, 2025) as its foundational components. The GCN is utilized to capture the relatively stable follow relationships among users within the social graph, while the HGNN complements this by modeling user interactions and diffusion dynamics within sequential diffusion hypergraphs.

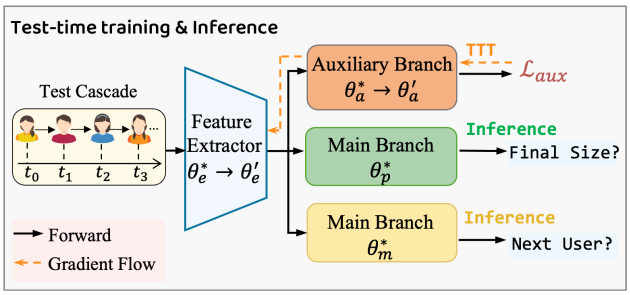

Figure 3: The test-time training and inference of Ghidorah.

Social Relation Encoder Given the social graph $\mathcal{G}_S = (\mathcal{U}, \mathcal{E})$, the final user social relation embeddings $\mathbf{X}_S \in \mathbb{R}^{N \times d}$ can be obtained by stacking multiple layers of GCN, where N denotes the size of user set. The initial embeddings are randomly initialized and viewed as learnable parameters.

Diffusion Interaction Encoder On the basis of the constructed diffusion hypergraphs $\mathcal{G}_D$, we model user interactions across cascades through HGNN at each time interval, where message aggregation involves two main steps: Node-to-Hyperedge aggregation and Hyperedge-to-Node aggregation. Given a diffusion hypergraph $\mathcal{G}_D^t$, the first step updates the embedding $o_{j,t}$ of hyperedge e_j^t by aggregating the embedddings of all its connected nodes. After obtaining embeddings of hyperedges, the second step aims to aggregate the embeddings of hyperedges associated with user u_i^t to update its embedding $x_{i,t}$. This process is defined as follows:

$$
\begin{aligned}
o_{j,t}^{l+1} &= \sigma\left(\sum_{u_i^t \in \mathcal{N}_v(e_j^t)} \frac{\mathbf{W}_v \mathbf{x}_{i,t}^l}{|\mathcal{N}_v(e_j^t)|}\right), \\
\mathbf{x}_{i,t}^{l+1} &= \sigma\left(\sum_{e_j^t \in \mathcal{N}_e(u_i^t)} \frac{\mathbf{W}_e \mathbf{o}_{j,t}^{l+1}}{|\mathcal{N}_e(u_i^t)|}\right),
\end{aligned} \tag{3}
$$

where $\mathcal{N}_v(e_j^t)$ represents the set of nodes connected to hyperedge e_j^t, $\mathcal{N}_e(u_i^t)$ denotes the set of hyperedges connected to node u_i^t and $\mathbf{W}_v$ and $\mathbf{W}_e$ are learnable parameters.

A single HGNN only captures the interactions at a time interval, which is insufficient for fully modeling the evolution of cascades. To address this limitation, we introduce a gated fusion strategy (Zhang, Zhang, and Pan 2022) that sequentially integrates the interactions learned by the HGNN across different time intervals, enabling us to obtain the final user dynamic interaction representations, $\mathbf{X}_D \in \mathbb{R}^{N \times d}$.

Shared Cascade Representation Learning Based on the input cascade $\mathcal{C}$, a LSTM model is introduced to model the contextual interactions between users within a cascade. Specifically, for each original cascade $c_m \in \mathcal{C}$, we retrieve the corresponding user embeddings from $\mathbf{X}_S$ and $\mathbf{X}_D$, arranging them in the original order of the cascade while disregarding the specific timestamps. This yields $\mathbf{Z}_m^S = [(\mathbf{x}_i)] \in \mathbb{R}^{|c_m| \times d}$, $\mathbf{Z}_m^D = [(\mathbf{x}_i)] \in \mathbb{R}^{|c_m| \times d}$. Both sets of embeddings are then fed into the LSTM at each time step, producing a d-dimensional representation for each user within the cascade. By integrating the structural and dynamic aspects of the embeddings, the LSTM generates a more comprehensive representation, denoted as $\mathbf{H}^{\text{share}} \in \mathbb{R}^{|c_m| \times d}$.

Macroscopic Prediction Branch

Given the user dynamic interaction representations $\mathbf{X}_D$ and the shared embedding $\mathbf{H}^{\text{share}}$, macroscopic prediction branch is designed to predict the ultimate size of the cascade.

Invariant Cascade Representation Learning Geirhos et al. (Geirhos et al. 2020) highlight that deep neural networks are prone to exploit easy-to-fit spurious correlations, i.e. shortcut strategies, when solving problems. Invariant learning, which aims to reveal invariant causal relationships between the features and target labels across different environments while reducing the effect of variant spurious correlations, can achieve satisfactory OOD generalization under distribution shifts (Arjovsky et al. 2019). We adopt Variance Risk Extrapolation (Krueger et al. 2021) to train an invariant model that can generalize to unseen and shifted test data.

As shown in Figure 2(c), multiple training environments are constructed from the original training cascades $\mathcal{C}$. For each diffusion cascade $c_m \in \mathcal{C}$, two typical sequence augmentation strategies (Xie et al. 2022) are adopted: user masking and user shuffling, resulting in the augmented training environments $\mathcal{F}$.

The macroscopic prediction task of various environments are handled independently. For each augmented cascade $\tilde{c}_m$, user embeddings from $\mathbf{X}_D$ form $\mathbf{Y}_m^D = [(x_i)] \in \mathbb{R}^{|\tilde{c}_m| \times d}$, which is input into a task-specific LSTM to produce the cascade representation $\mathbf{H}^{macro}$. This representation is concatenated with H^{share} and passed through an MLP predictor to estimate the final cascade size. The task-specific LSTM and predictor make up the macroscopic prediction branch, parameterized by θ_p.

Training Objective Following previous work (Jiao et al. 2024), the Mean Absolute Error is selected as the loss for a single environment: $\mathcal{R}_e(\theta_e, \theta_p) = \frac{1}{M}\sum_{m=1}^{M} |y_m - \hat{y}_m|$, where $\hat{y}_m = f_{\theta_p}(\tilde{\mathcal{C}}_e; \theta_e)$ denotes the predicted popularity for the cascade $\tilde{c}_m \in \tilde{\mathcal{C}}_e$, y_m represents the true popularity value, and $|M|$ is the number of training cascades. To combine the losses of different environments, we aim to reduce the average training risk while simultaneously increasing the similarity of training risks across different environments (Krueger et al. 2021) as follows:

$$
\begin{aligned}
\mathcal{L}_{main}^p(\theta_e, \theta_p) \doteq \gamma \text{Var}\left(\{\mathcal{R}_1(\theta_e, \theta_p), \ldots, \mathcal{R}_{|\mathcal{F}|}(\theta_e, \theta_p)\}\right) \\
+ \sum_{e=1}^{|\mathcal{F}|} \mathcal{R}_e(\theta_e, \theta_p),
\end{aligned} \tag{4}
$$

where γ is a hyper-parameter to control the balance between reducing average risk and enforcing equality of risks.

Microscopic Prediction Branch

The microscopic prediction branch is similar to the macroscopic branch, except for its learning objective. The classifier in this branch, composed of multiple layers of MLPs, predicts the next user in the cascade. The task-specific LSTM and the classifier together constitute the microscopic prediction branch, parameterized by θ_m.

Training Objective Microscopic prediction is treated as a classification problem with the Cross-Entropy loss function:

$$\mathcal{L}_e(\theta_e, \theta_m) = -\sum_{i=2}^{|\tilde{c}_m|}\sum_{j=1}^{N} y_{ij}\log(\hat{y}_{ij}), \tag{5}$$

where y_{ij} = 1 denotes that the user u_j participate in cascade $\tilde{c}_m$ at position i, otherwise, y_{ij} = 0. Analogous to Eq. (4) , we derive the loss for the microscopic task, $\mathcal{L}^m_{main}$, using the same principle of Risk Extrapolation.

Auxiliary Self-Supervised Branch

The key to TTT lies in selecting an proper auxiliary SSL task (Sun et al. 2020). We propose incorporating an MAE-based SSL task into the auxiliary branch and introducing a consistency loss to align the SSL task with the main tasks.

Self-supervised Learning via MAE As shown in Fig. 2, the MAE employs an asymmetric encoder-decoder structure. Consider a short cascade $c_i = \{u_1, u_2, \ldots, u_Q\}$. A subset of users is randomly masked according to a mask ratio p_m, with their user tokens replaced by a uniform mask token. The sequence is then transformed into user embeddings using the learned shared representations $\mathbf{H}^{\text{share}}$. For a visible user u_j, the user embedding is $\mathbf{e}_j$, while masked users share a common embedding $\hat{\mathbf{m}}$. Position embeddings are then added for each user, resulting in the cascade's representative embeddings, denoted as $\mathbf{S} = [\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2, \ldots, \tilde{\mathbf{e}}_Q]$.

In the encoder module, we input the representative embeddings of visible users $\mathbf{V} = [\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2, \ldots, \tilde{\mathbf{e}}_R]$, where R is the number of visible users. The encoder, composed of N_e Transformer layers, generates individual representations for each visible user, $\mathbf{h}_r$. In the decoder module, N_d Transformer layers use the encoder's output along with the shared embedding of masked users, $[\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_Q]$, to reconstruct the embeddings of all users in the sequence, denoted as $\hat{\mathbf{H}} = [\hat{\mathbf{h}}_1, \hat{\mathbf{h}}_2, \ldots, \hat{\mathbf{h}}_Q]$. Finally, we only use the reconstructed embeddings of the masked users to construct the loss function. Specifically, the reconstruction loss $\mathcal{L}_{aux}$ is the mean squared error between these reconstructed embeddings and the true embeddings from $\mathbf{H}^{\text{share}}$, as follows:

$$\mathcal{L}_{aux}(\theta_e, \theta_a) = \frac{1}{Q-R}\sum_{i=1}^{Q-R}\left(\mathbf{x}_i - \hat{\mathbf{h}}_i\right)^2. \tag{6}$$

Consistency Loss for TTT As the correlation of gradients between the auxiliary and main task loss functions is crucial for TTT success (Sun et al. 2020), we propose a consistency loss to align updates across all three tasks. Based on Eqs. (4) and (6), the anticipated gradients for the shared graph encoder parameters θ_e, derived from $\mathcal{L}^p_{main}$, $\mathcal{L}^m_{main}$, and $\mathcal{L}_{aux}$, are denoted as $\mathbf{g}^p_{main}$, $\mathbf{g}^m_{main}$, and $\mathbf{g}_{aux}$, respectively. These gradients are expected to be strongly correlated to ensure consistent optimization of the shared parameters and to improve overall performance (Zhao et al. 2021). To achieve this, we introduce a consistency loss that enforces equality among the standardized versions of $\mathbf{g}^p_{main}$, $\mathbf{g}^m_{main}$, and $\mathbf{g}_{aux}$:

$$\mathcal{L}_{align}(\theta_e, \theta_a, \theta_p, \theta_m) = |\hat{\mathbf{g}}^p_{main} - \hat{\mathbf{g}}_{aux}| + |\hat{\mathbf{g}}^m_{main} - \hat{\mathbf{g}}_{aux}|, \tag{7}$$

Datasets	Christianity	Android	Memetracker	Douban
# Users	2,897	9,958	4,709	12,232
# Links	35,624	48,573	209,194	396,992
# Cascades	589	679	12,661	3,475
Avg. Length	22.9	33.3	16.24	21.76

Table 1: The statistics of four datasets.

where the standardized gradient $\hat{\mathbf{g}}^p_{main} = \frac{\mathbf{g}^p_{main} - E(\mathbf{g}^p_{main})}{\sigma(\mathbf{g}^p_{main})}$, and similarly for $\hat{\mathbf{g}}^m_{main}$ and $\hat{\mathbf{g}}_{aux}$.

Experiment

Experimental Settings

Datasets Following previous works (Yang et al. 2021; Sun et al. 2022), we evaluate the proposed framework on four datasets collected from real-world platforms: Christianity, Android, Memetracker (Jiao et al. 2024), and Douban. We randomly sample 80% of the cascades for training, 10% for validation, and the remaining 10% for testing. The detailed statistics of the datasets are presented in Table 1.

Baselines The proposed model is compared with sixteen state-of-the-art (SOTA) baselines. For the macroscopic baselines, four models are selected: DeepCas (Li et al. 2017a), DeepHawkes (Cao et al. 2017), CasCN (Chen et al. 2019b), and CasFlow (Xu et al. 2021). For the microscopic baselines, we select six models: TopoLSTM (Wang et al. 2017), NDM (Yang et al. 2019), Inf-VAE (Sankar et al. 2020), DyHGCN (Yuan et al. 2021), MS-HGAT (Sun et al. 2022), and CE-GCN (Wang et al. 2022). For the multi-scale prediction baselines, we select three models: FOREST (Yang et al. 2021), DMT-LIC (Chen et al. 2019a), and MINDS (Wang, Xu, and Zhang 2023). For the domain generalization baselines, we select three models: MLDG (Li et al. 2018b), MMD-AAE (Li et al. 2018c), and SFA (Li et al. 2021b).

Evaluation Metrics Following previous work (Xu et al. 2021), we use MSLE for macroscopic prediction. For microscopic prediction, we adopt two popular ranking metrics, MAP@k and Hits@k, with k set to 10, 50, and 100.

Implementation Details Our model is implemented in PyTorch. The results for Ghidorah are presented as the mean of five runs to ensure reliable evaluation. The performance of the baseline models represents the best outcomes reported in published papers, alongside our local experiments. All hyperparameters are selected through a grid search algorithm based on validation set performance, with final results reported on the test set.

Performance Comparison

Table 2 and Table 3 report the results for microscopic prediction, while Table 4 summarizes the results for macroscopic prediction. (1) As shown in Tables 2 and 3, Ghidorah consistently surpasses all SOTA baselines, exhibiting an impressive improvement of over 10.97% in Hits@100 and MAP@100 scores. Compared to the separated training models like CE-GCN, Ghidorah's superiority stems from the collaborative reinforcement provided by multi-task learning for cascade modeling, as well as the efficacy of TTT. (2) Table

Models	Christianity			Android			Memetracker			Douban		
Hits@k ↑	**@10**	**@50**	**@100**	**@10**	**@50**	**@100**	**@10**	**@50**	**@100**	**@10**	**@50**	**@100**
TopoLSTM	0.1559	0.3653	0.4777	0.0460	0.1318	0.2103	0.1908	0.3687	0.4683	0.0306	0.0143	0.0184
NDM	0.0464	0.1145	0.1461	0.0170	0.0423	0.0555	0.0931	0.1228	0.1279	0.0388	0.0506	0.0528
Inf-VAE	0.0767	0.2569	0.3853	0.0318	0.0938	0.1452	0.1165	0.3096	0.4200	0.1364	0.2361	0.3039
DyHGCN	0.2380	0.4689	0.5923	0.0748	0.1746	0.2596	0.2522	0.4603	0.5710	0.1438	0.2648	0.3329
MS-HGAT	0.2880	0.4714	0.5562	0.1041	0.2031	0.2755	0.2584	0.4743	0.5832	0.2133	0.3525	0.4275
CE-GCN	0.2806	0.5250	0.6394	0.0886	0.1992	0.2727	0.3701	0.5604	0.6509	0.1885	0.3272	0.4047
FOREST	0.2746	0.4665	0.5603	0.0866	0.1739	0.2314	0.2648	0.4502	0.5499	0.1106	0.1986	0.2559
DMT-LIC	0.2768	0.4442	0.5669	0.0932	0.1639	0.2315	0.2746	0.4619	0.5656	0.1465	0.2506	0.3054
MINDS	0.3214	0.4978	0.6250	0.1096	0.1989	0.2766	0.2819	0.4760	0.5790	0.1956	0.3087	0.3641
Ghidorah	**0.3917**	**0.6639**	**0.8059**	**0.1310**	**0.2365**	**0.3214**	**0.4202**	**0.6224**	**0.7276**	**0.2471**	**0.3964**	**0.4845**

Table 2: Experimental results for microscopic prediction are reported in terms of *Hits@k*, where higher scores indicate better performance. The improvement over the best-performing baseline methods is statistically significant (sign test, $p < 0.01$).

Models	Christianity			Android			Memetracker			Douban		
MAP@k ↑	**@10**	**@50**	**@100**	**@10**	**@50**	**@100**	**@10**	**@50**	**@100**	**@10**	**@50**	**@100**
TopoLSTM	0.0523	0.0619	0.0635	0.0166	0.0202	0.0213	0.0870	0.0955	0.0969	0.0354	0.0824	0.0884
NDM	0.0144	0.0177	0.0182	0.0059	0.0070	0.0072	0.0463	0.0480	0.0481	0.0141	0.0824	0.0884
Inf-VAE	0.0172	0.0254	0.0272	0.0076	0.0103	0.0110	0.0425	0.0509	0.0525	0.0543	0.0588	0.0598
DyHGCN	0.1062	0.1167	0.1184	0.0392	0.0434	0.0446	0.1410	0.1502	0.1518	0.0801	0.0856	0.0865
MS-HGAT	0.1744	0.1827	0.1840	0.0639	0.0687	0.0696	0.1408	0.1504	0.1519	0.1172	0.1252	0.1260
CE-GCN	0.1467	0.1586	0.1602	0.0477	0.0524	0.0534	0.2154	0.2243	0.2256	0.1103	0.1164	0.1175
FOREST	0.1569	0.1658	0.1672	0.0628	0.0667	0.0675	0.1429	0.1514	0.1528	0.0655	0.0694	0.0702
DMT-LIC	0.1649	0.1728	0.1746	0.0622	0.0652	0.0662	0.1496	0.1581	0.1595	0.0812	0.0856	0.0897
MINDS	0.1955	0.2037	0.2054	0.0677	0.0716	0.0727	0.1535	0.1623	0.1638	0.1142	0.1199	0.1213
Ghidorah	**0.2468**	**0.2526**	**0.2673**	**0.0809**	**0.0834**	**0.0841**	**0.2439**	**0.2489**	**0.2521**	**0.1345**	**0.1405**	**0.1417**

Table 3: Experimental results for microscopic prediction are reported in terms of *MAP@k*, where higher scores indicate better performance. The improvement over the best-performing baseline methods is statistically significant (sign test, $p < 0.01$).

4 indicates that Ghidorah consistently outperforms all SOTA methods for macroscopic prediction, including CasCN and CasFlow, with a relative reduction of over 15.89% in MSLE. This substantial enhancement can be attributed to Ghidorah's capability to capture more fine-grained and context-aware diffusion dependencies within cascades. (3) Compared to unified frameworks such as FOREST, DMT-LIC, and MINDS, our approach uses TTT to better capture the intrinsic features of each test sample, resulting in more robust and satisfactory OOD generalization. (4) Table 4 shows that Ghidorah not only competes favorably but often surpasses domain generalization methods like MLDG, MMD-AAE, and SFA, which highlights its superior ability to generalize across diverse domains.

Ablation Study

We conduct a series of ablation studies on the Christianity and Douban datasets to evaluate the importance of each module within Ghidorah. Table 5 presents the results. The first row of Table 5 reveals that the exclusion of HGNN significantly impairs model performance, underscoring the critical role of capturing global user dynamic interactions. The absence of TTT results in a marked deterioration in performance, accentuating the challenge of OOD scenarios and the efficacy of TTT in enhancing the model's generalization capabilities. Moreover, the omission of consistency loss leads to a performance drop, highlighting the essential need to align optimization objectives between auxiliary and primary tasks. The absence of the invariant learning component further diminishes generalization performance, demonstrating its pivotal role in maintaining stable and invariant cascade representations. In the TTT framework, the selection of the auxiliary task is paramount. Contrastive learning (Xie et al. 2022; Zhang et al. 2024d) and Bootstrap Your Own Latent (BYOL) (Grill et al. 2020) are introduced for comparison. The results, detailed in the 5th row of Table 5, indicate that MAE is the most effective auxiliary task for IDP.

Hyperparameter Sensitivity Analysis

Mask Ratio p_m in MAE The mask ratio p_m specifies the percentage of users masked in the cascade. As illustrated in Fig. 4(a), Ghidorah performs best with a mask ratio of 0.4, which masks 40% of users. Lower mask ratios might necessitate masking a greater proportion of users to reduce redundancy, while higher ratios may leave too few visible users to effectively capture context-dependent interactions.

Model (*MSLE* ↓)	Christ.	Android	Meme.	Douban
DeepCas	1.446	2.122	2.231	2.122
DeepHawkes	1.111	1.971	1.143	1.725
CasCN	1.046	0.981	0.967	1.476
CasFlow	0.765	1.041	0.535	0.465
FOREST	1.726	0.556	0.621	0.825
DMT-LIC	1.692	0.201	0.701	0.741
MINDS	0.572	0.151	0.506	0.404
MLDG	0.461	0.142	0.476	0.387
MMD-AAE	0.415	0.136	0.453	0.352
SFA	0.384	0.130	0.438	0.326
Ghidorah	**0.356**	**0.127**	**0.414**	**0.290**

Table 4: Results of macroscopic prediction are evaluated by MSLE. Lower values indicate better performance.

Models	*MSLE* ↓		*MAP@100* ↑	
	Christ.	Douban	Christ.	Douban
w/o HGNN	1.121	0.594	0.1965	0.1092
w/o TTT	0.485	0.371	0.2214	0.1286
w/o align	0.372	0.308	0.2622	0.1379
w/o invariant	0.386	0.325	0.2586	0.1326
CL	0.441	0.353	0.2302	0.1254
BYOL	0.412	0.337	0.2471	0.1295
Ghidorah	**0.356**	**0.290**	**0.2673**	**0.1417**

Table 5: Ablation study for two IDP prediction tasks.

Auxiliary Task Loss Weight α To assess the impact of α, which balances the main and auxiliary task losses as described in Eq. (1), we perform a qualitative analysis within the Ghidorah framework (see Fig. 4(b)). Optimal performance is achieved when α is set to 0.5.

Number of Constructed Environment $\mathcal{F}$ As in Fig. 4(c), the optimal number of constructed environments is 6. Increasing this number significantly reduces performance, likely because distribution shifts in diffusion prediction are not solely associated with a single spurious domain. An excessive number of environments may hinder the learning of a stable invariant representation.

Gradient Steps δ during Test-Time Training The parameter δ determines the number of gradient steps applied during test-time training to fine-tune the shared feature extractor for a test sample. As shown in Fig. 4(d), performance improves with an increasing number of steps, peaks at 15 gradient updates, and then begins to decline. When the number of steps exceeds 30, the shared encoder may overfit to the target distribution, leading to poor classifier performance as the representations deviate from the learned latent space.

Related Work

Information diffusion prediction, which aims to forecast future diffusion dynamics based on observed interactive records, can be categorized into macroscopic prediction and microscopic prediction (Li et al. 2024; Wang, Xu, and Zhang 2023). While most previous studies have primarily focused on single tasks (Xu et al. 2021; Sun et al. 2022; Wang et al. 2022; Feng et al. 2022; Lu et al. 2023; Ji et al. 2023), recent efforts have introduced unified frameworks to address both tasks simultaneously by leveraging their shared features (Chen et al. 2019b; Yang et al. 2021; Jiao et al. 2024). Despite their considerable success, these approaches overlook the distribution shifts between training and testing cascades, arising from the inherent uncertainty of user behaviors (Zhang et al. 2023), which hampers the model's generalization to unknown test environments.

The primary objective of test-time training (TTT) is to adapt the trained model to a single test sample through an auxiliary self-supervised learning (SSL) task. TTT has been widely applied across various domains, including image (Wang et al. 2024; Hatem, Qian, and Wang 2023), graph (Zhang et al. 2024b), and reinforcement learning (Hansen et al. 2020). Extensive experiments highlight the promising potential of TTT in enhancing the model's generalization capabilities. However, this paradigm is heavily dependent on the selection of an appropriate auxiliary task (Gandelsman et al. 2022), and unconstrained TTT can sometimes lead to undesirable performance degradation in the main task. Therefore, developing an effective TTT framework for information diffusion prediction (IDP) tasks is not trivial.

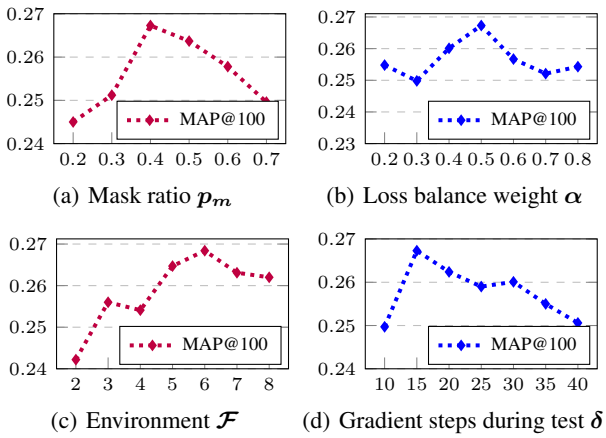

Figure 4: Hyperparameter sensitivity results on Christianity.

Conclusion

In this paper, we addressed overlooked OOD generalization within IDP tasks and proposed Ghidorah, a novel model based on the test-time training paradigm. The auxiliary branch introduces an MAE-like loss to optimize model parameters to specific test samples, while the main branch uses invariant learning to capture stable cascade representations across environments. These innovations enhance the model's generalization. Experiments demonstrate that Ghidorah consistently outperforms state-of-the-art approaches.

Acknowledgments

This work is supported by the National Key Research and Development Program (Grant No. 2023YFC3303800).

References

Arjovsky, M.; Bottou, L.; Gulrajani, I.; and Lopez-Paz, D. 2019. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*.

Cao, Q.; Shen, H.; Cen, K.; Ouyang, W.; and Cheng, X. 2017. Deephawkes: Bridging the gap between prediction and understanding of information cascades. In *CIKM*, 1149–1158.

Chen, L.; Zhang, Y.; Song, Y.; Wang, J.; and Liu, L. 2022. Ost: Improving generalization of deepfake detection via one-shot test-time training. In *NeurIPS*, volume 35, 24597–24610.

Chen, X.; Zhang, K.; Zhou, F.; Trajcevski, G.; Zhong, T.; and Zhang, F. 2019a. Information cascades modeling via deep multi-task learning. In *SIGIR*, 885–888.

Chen, X.; Zhou, F.; Zhang, K.; Trajcevski, G.; Zhong, T.; and Zhang, F. 2019b. Information diffusion prediction via recurrent cascades convolution. In *ICDE*, 770–781.

Feng, S.; Zhao, K.; Fang, L.; Feng, K.; Wei, W.; Li, X.; and Shao, L. 2022. H-Diffu: hyperbolic representations for information diffusion prediction. *TKDE*, 35(9): 8784–8798.

Gandelsman, Y.; Sun, Y.; Chen, X.; and Efros, A. 2022. Test-time training with masked autoencoders. In *NeurIPS*, volume 35, 29374–29385.

Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; March, M.; and Lempitsky, V. 2016. Domain-adversarial training of neural networks. *JMLR*, 17(59): 1–35.

Geirhos, R.; Jacobsen, J.-H.; Michaelis, C.; Zemel, R.; Brendel, W.; Bethge, M.; and Wichmann, F. A. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11): 665–673.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Avila Pires, B.; Guo, Z.; Gheshlaghi Azar, M.; et al. 2020. Bootstrap your own latent-a new approach to self-supervised learning. In *NeurIPS*, volume 33, 21271–21284.

Gulrajani, I.; and Lopez-Paz, D. 2020. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*.

Hansen, N.; Jangir, R.; Sun, Y.; Alenyà, G.; Abbeel, P.; Efros, A. A.; Pinto, L.; and Wang, X. 2020. Self-supervised policy adaptation during deployment. *arXiv preprint arXiv:2007.04309*.

Hatem, A.; Qian, Y.; and Wang, Y. 2023. Point-TTA: Test-Time Adaptation for Point Cloud Registration Using Multitask Meta-Auxiliary Learning. In *CVPR*, 16494–16504.

Ji, S.; Lu, X.; Liu, M.; Sun, L.; Liu, C.; Du, B.; and Xiong, H. 2023. Community-based dynamic graph learning for popularity prediction. In *SIGKDD*, 930–940.

Jia, X.; Shang, J.; Liu, D.; Zhang, H.; and Ni, W. 2022. HeDAN: Heterogeneous diffusion attention network for popularity prediction of online content. *KBS*, 254: 109659.

Jiao, P.; Chen, H.; Bao, Q.; Zhang, W.; and Wu, H. 2024. Enhancing Multi-Scale Diffusion Prediction via Sequential Hypergraphs and Adversarial Learning. In *AAAI*, 8571–8581.

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

Krueger, D.; Caballero, E.; Jacobsen, J.-H.; Zhang, A.; Binas, J.; Zhang, D.; Le Priol, R.; and Courville, A. 2021. Out-of-distribution generalization via risk extrapolation (rex). In *ICML*, 5815–5826.

Li, C.; Ma, J.; Guo, X.; and Mei, Q. 2017a. Deepcas: An end-to-end predictor of information cascades. In *WWW*, 577–586.

Li, C.; Pang, B.; Liu, Y.; Sun, H.; Liu, Z.; Xie, X.; Yang, T.; Cui, Y.; Zhang, L.; and Zhang, Q. 2021a. Adsgnn: Behavior-graph augmented relevance modeling in sponsored search. In *SIGIR*, 223–232.

Li, C.; Wang, S.; Yang, D.; Li, Z.; Yang, Y.; Zhang, X.; and Zhou, J. 2017b. PPNE: property preserving network embedding. In *DASFAA*, 163–179.

Li, C.; Wang, S.; Yu, P. S.; Zheng, L.; Zhang, X.; Li, Z.; and Liang, Y. 2018a. Distribution distance minimization for unsupervised user identity linkage. In *CIKM*, 447–456.

Li, D.; Yang, Y.; Song, Y.; and Hospedales, T. M. 2018b. Learning to Generalize: Meta-Learning for Domain Generalization. In *AAAI*, 3490–3497.

Li, H.; Pan, S. J.; Wang, S.; and Kot, A. C. 2018c. Domain generalization with adversarial feature learning. In *CVPR*, 5400–5409.

Li, H.; Xia, C.; Wang, T.; Wang, Z.; Cui, P.; and Li, X. 2024. Grass: Learning spatial–temporal properties from chainlike cascade data for microscopic diffusion prediction. *TNNLS*, 35: 16313–16327.

Li, P.; Li, D.; Li, W.; Gong, S.; Fu, Y.; and Hospedales, T. M. 2021b. A simple feature augmentation for domain generalization. In *ICCV*, 8886–8895.

Long, M.; Zhu, H.; Wang, J.; and Jordan, M. I. 2016. Unsupervised domain adaptation with residual transfer networks. In *NeurIPS*, volume 29, 136–144.

Lu, X.; Ji, S.; Yu, L.; Sun, L.; Du, B.; and Zhu, T. 2023. Continuous-time graph learning for cascade popularity prediction. *arXiv preprint arXiv:2306.03756*.

Sain, A.; Bhunia, A. K.; Potlapalli, V.; Chowdhury, P. N.; Xiang, T.; and Song, Y.-Z. 2022. Sketch3T: Test-Time Training for Zero-Shot SBIR. In *CVPR*, 7462–7471.

Sankar, A.; Zhang, X.; Krishnan, A.; and Han, J. 2020. Inf-VAE: A variational autoencoder framework to integrate homophily and influence in diffusion prediction. In *WSDM*, 510–518.

Sun, L.; Rao, Y.; Zhang, X.; Lan, Y.; and Yu, S. 2022. MS-HGAT: memory-enhanced sequential hypergraph attention network for information diffusion prediction. In *AAAI*, volume 36, 4156–4164.

Sun, Y.; Wang, X.; Liu, Z.; Miller, J.; Efros, A.; and Hardt, M. 2020. Test-time training with self-supervision for generalization under distribution shifts. In *ICML*, 9229–9248.

Wang, D.; Wei, L.; Yuan, C.; Bao, Y.; Zhou, W.; Zhu, X.; and Hu, S. 2022. Cascade-enhanced graph convolutional network for information diffusion prediction. In *DASFAA*, 615–631.

Wang, J.; Zheng, V. W.; Liu, Z.; and Chang, K. C.-C. 2017. Topological recurrent neural network for diffusion prediction. In *ICDM*, 475–484.

Wang, R.; Xu, X.; and Zhang, Y. 2023. Multiscale information diffusion prediction with minimal substitution neural network. *TNNLS*.

Wang, Z.; Huang, H.; Zheng, A.; and He, R. 2024. Heterogeneous Test-Time Training for Multi-Modal Person Re-identification. In *AAAI*, volume 38, 5850–5858.

Xie, X.; Sun, F.; Liu, Z.; Wu, S.; Gao, J.; Zhang, J.; Ding, B.; and Cui, B. 2022. Contrastive learning for sequential recommendation. In *ICDE*, 1259–1273.

Xu, X.; Zhou, F.; Zhang, K.; Liu, S.; and Trajcevski, G. 2021. Casflow: Exploring hierarchical structures and propagation uncertainty for cascade prediction. *TKDE*, 35(4): 3484–3499.

Yang, C.; Sun, M.; Liu, H.; Han, S.; Liu, Z.; and Luan, H. 2019. Neural diffusion model for microscopic cascade study. *TKDE*, 33(3): 1128–1139.

Yang, C.; Wang, H.; Tang, J.; Shi, C.; Sun, M.; Cui, G.; and Liu, Z. 2021. Full-scale information diffusion prediction with reinforced recurrent networks. *TNNLS*, 34(5): 2271–2283.

Yuan, C.; Li, J.; Zhou, W.; Lu, Y.; Zhang, X.; and Hu, S. 2021. DyHGCN: A dynamic heterogeneous graph convolutional network to learn users' dynamic preferences for information diffusion prediction. In *ECML-PKDD*, 347–363.

Zhang, H.; Liu, X.; Yang, Q.; Yang, Y.; Qi, F.; Qian, S.; and Xu, C. 2024a. T3RD: Test-Time Training for Rumor Detection on Social Media. In *WWW*, 2407–2416.

Zhang, J.; Wang, Y.; Yang, X.; and Zhu, E. 2024b. A Fully Test-Time Training Framework for Semi-Supervised Node Classification on Out-of-Distribution Graphs. *TKDD*, 18(7): 172.

Zhang, L.; Zhang, X.; Li, C.; Zhou, Z.; Liu, J.; Huang, F.; and Zhang, X. 2024c. Mitigating Social Hazards: Early Detection of Fake News via Diffusion-Guided Propagation Path Generation. In *MM*, 2842–2851.

Zhang, L.; Zhang, X.; and Pan, J. 2022. Hierarchical cross-modality semantic correlation learning model for multimodal summarization. In *AAAI*, volume 36, 11676–11684.

Zhang, L.; Zhang, X.; Zhou, Z.; Zhang, X.; Wang, S.; Philip, S. Y.; and Li, C. 2024d. Early Detection of Multimodal Fake News via Reinforced Propagation Path Generation. *TKDE*.

Zhang, L.; Zhang, X.; Zhou, Z.; Zhang, X.; Yu, P. S.; and Li, C. 2025. Knowledge-aware multimodal pre-training for fake news detection. *Information Fusion*, 114: 102715.

Zhang, P.; Guo, J.; Li, C.; Xie, Y.; Kim, J. B.; Zhang, Y.; Xie, X.; Wang, H.; and Kim, S. 2023. Efficiently leveraging multi-level user intent for session-based recommendation via atten-mixer network. In *WSDM*, 168–176.

Zhao, J.; Li, C.; Wen, Q.; Wang, Y.; Liu, Y.; Sun, H.; Xie, X.; and Ye, Y. 2021. Gophormer: Ego-graph transformer for node classification. *arXiv preprint arXiv:2110.13094*.

Zhou, F.; Xu, X.; Trajcevski, G.; and Zhang, K. 2021. A survey of information cascade analysis: Models, predictions, and recent advances. *CSUR*, 54(2): 1–36.