HFAN-ITF: Hierarchical Fusion Wasserstein Generative Adversarial Network for Interbank Network Temporal Forecasting

Huanlan Yan, Yijun Chen, Xinjia Shi, and Zhijun Ding*
School of Computer Science and Technology, Tongji University
No. 4800 Cao'an Road, Shanghai, China, 201804
{2011597,2232918, 2512054, dingzj}@tongji.edu.cn

***Abstract*—Temporal forecasting of interbank networks is crucial for financial system risk assessment, but is challenged by the feature capture of complex dynamic graphs as well as the training instability and convergence difficulties of generative models like Generative Adversarial Networks (GANs). To address these limitations, we introduce HFAN-ITF, a novel generative adversarial framework for interbank network forecasting. Within the generator, a Hierarchical Fusion Network (HFN) captures multi-level structural features, where HFN extracts global and local structural features and captures inter-graph temporal dependencies for spatio-temporal modeling. Furthermore, a novel Self-adaptive Edge Density Control (SAEDC) mechanism, an adaptive Top-K strategy, guides the network toward accelerated learning. The entire process is steered by a Wasserstein distance objective to ensure stable convergence. Through extensive evaluations on our proposed interbank network dataset, we demonstrate that HFAN-ITF consistently outperforms current Widely-adopted techniques across the comprehensive tasks of node attribute, link, and weight forecasting, exhibiting a particularly significant advantage in predicting interbank lending weights.**

***Index Terms*—Interbank Network, Temporal Forecasting, Graph Neural Network, HFAN-ITF, Generative Adversarial Network**

I. Introduction

Interbank networks, functioning as the fundamental arteries for liquidity redistribution, constitute the backbone of the modern financial system [1]. These networks are not merely static conduits for capital; they are complex adaptive systems where the topology of exposures continually evolves in response to exogenous shocks, regulatory shifts, and endogenous liquidity preferences. Such temporal reshaping directly alters the pathways of systemic contagion and the cost of funding. Consequently, the capability to forecast the future state of these networks—encompassing the precise prediction of lending links, weight distributions, and the evolving financial attributes of banking institutions—is a prerequisite for effective macroprudential stress testing and proactive systemic risk management.

However, the task of temporal interbank network forecasting extends beyond traditional link prediction. It requires a generative model capable of jointly predicting the evolving graph topology (links and weights) and the node attributes (balance sheet items). This task faces two primary formidable challenges. First, the topological structure exhibits high complexity, characterized by extreme sparsity, directed anisotropy, and distinct core-periphery patterns that generic graph models fail to capture. Second, the complexity of the prediction task necessitates a stabilized generative framework. The generation of high-dimensional, dynamic graph data is plagued by inherent training instability. Furthermore, the immense search space of potential topologies imposes severe bottlenecks on training efficiency, often leading to slow convergence and difficulties in identifying valid sparse structures without explicit guidance.

Existing computational paradigms struggle to reconcile the discrete nature of graph topology with the continuous dynamics of temporal evolution. Vector-based Temporal Models (e.g., RNNs/LSTMs) [2], [3] attempt to capture dynamics but necessitate the flattening of graph snapshots, a process that fundamentally violates permutation invariance. This reduction strips away structural isomorphism, degrading complex topological evolution into simple low-dimensional regression. Static Graph Neural Networks (GNNs) [4]–[7] restore this structural awareness through message passing, yet they remain mathematically confined by the presupposition of a fixed adjacency matrix. Consequently, they model conditional probabilities rather than joint generative distributions, rendering them incapable of natively originating novel topological structures. Diffusion Models [8], [9] emerge as a potential generative solution, but they grapple with a severe discrete-continuous mismatch. Their core Gaussian assumptions fundamentally conflict with the sparsity of interbank network, while independent noise modeling fails to capture global systemic dependencies. By assuming spatially decoupled noise, the reverse denoising process gravitates toward point-wise local restoration rather than a globally coupled structural reconstruction, leaving the model incapable of recovering the highly synergistic interdependencies inherent in financial networks. This leaves Generative Adversarial Networks (GANs) [10], [11] as the most theoretically viable candidate for modeling such arbitrary distributions. However, in high-dimensional graph spaces, they frequently succumb to Mode Collapse. The adversarial minimax game fails to reach a stable Nash Equilibrium.

To address these limitations, we introduce HFAN-ITF, a

This work is partially supported by the Special Fund of Fundamental Scientific Research Business Expense for Higher School of Central Government under Grant Number 22120240563.

novel generative adversarial framework for robust interbank network forecasting. Our main contributions are threefold:

- A Novel Framework for a New Task: To our knowledge, this is the first work to address the task of interbank network generation. We innovatively propose the HFAN-ITF method for this task. We hope to facilitate and standardize future research on this new task by providing the first public dataset and a novel method.
- Specialized Architectural Innovations: We design a set of modules tailored for the unique challenges of financial networks:
  - To capture complex spatio-temporal dynamics, our generator features a HFN. This module employs parallel branches to balance different structural priors: a GCN branch for global structural smoothing via normalized graph convolutions, and a GAT branch to capture heterogeneous neighbor importance through attention mechanisms. By extracting these complementary structural features and integrating them with temporal dependencies via a Transformer, the HFN constructs a comprehensive and robust node representation.
  - To address the training instability and convergence difficulties inherent in generating complex graph data, our framework integrates two key components. First, a Wasserstein GAN provides a stable learning signal to counteract mode collapse and vanishing gradients, ensuring robust convergence. Second, a Self-adaptive Edge Density Control (SAEDC) mechanism, an adaptive Top-K strategy, selects a network structure with a density similar to the historical distribution by dynamically adjusting the edge filtering threshold, thereby accelerating model training.
- Extensive Empirical Validation: We conduct fair comparisons against Widely-adopted methods, demonstrating that HFAN-ITF is consistently superior to existing approaches.

II. Related Work

Interbank Network Prediction Based on Machine Learning. Recent approaches combine ML with traditional theories [12]. For instance, XGBoost [13] and GCN-LSTM hybrids [14] have been applied to link prediction. Takahashi et al. [15] further employed self-attention with GCNs for dynamic link and flow prediction. However, these methods typically focus on predicting the existence of edges for key nodes rather than generating complete network structures (links, weights, and attributes) for future.

Deep Generative Models for Graphs. Generative models such as NetGAN [16], LGGAN [17], and GraphGAN [18] have demonstrated success in general graph tasks. More recent temporal methods like GCN-GAN [19], TSGG-GAN [20], and GT-GAN [21] attempt to address dynamic evolution [22]. Yan et al. [23] introduced a causal information bottleneck to enhance adversarial defense. However, these methods are prone to gradient instability or mode collapse when trained on novel tasks. To address this bottleneck, Wasserstein GAN (WGAN) [24] offers a robust solution by providing more reliable gradient information. Nevertheless, applying these directly to financial networks remains challenging due to the complexities of dense and time-varying graphs.

III. Task Definition

We define the temporal interbank network as $G = \{G_1, \dots, G_T\}$, where G_t is the interbank network at time t, $G_t = (A_t, X_t)$, where A_t is the adjacency matrix of the interbank lending network among N banks. $A_t^{ij} = [i, j, w]$ indicates that bank i lends an amount w to bank j. Here, $X_t = [x_t^1, \dots, x_t^N]$ represents the vectors of the nodes' attributes in the network. The task of this paper is the temporal prediction of the interbank network, including the prediction of the interbank lending network A_{t+1} and the prediction of the attributes X_{t+1} of the bank nodes. For snapshot t, and the l time slices before this time slice (using data from l look-back time steps as input), these data are used as input to predict the network $\tilde{G}_{t+1}$ at time $t+1$. This is specifically represented as Eq. 1.

$$\tilde{G}_{t+1} = f(G_{t-l}, G_{t-l+1}, \dots, G_t) \tag{1}$$

where $f(\cdot)$ refers to the HFAN-ITF constructed in this paper, and $\tilde{G}_{t+1}$ represents the temporal prediction result.

IV. Our Method

A. Data Generation

Due to the confidentiality of granular interbank exposure data, no public datasets are available for this task. To bridge this gap, we constructed a synthetic dataset using the Minimum Density(MD) method [25], a standard approach in financial systemic risk research. The MD method reconstructs the network topology by minimizing total lending costs while respecting the observed asset and liability constraints of each bank. To improve model robustness and simulate real-world reporting noise, we applied a dual augmentation strategy:

- **Parameter Augmentation:** We randomly perturbed the MD cost and entropy parameters (c, θ) by $\delta \sim \mathcal{U}(-\alpha, \alpha)$ to generate diverse topological structures. Where $\alpha = 0.5$, c denotes the transaction cost, and θ represents the transaction entropy.
- **Attribute Perturbation:** We injected Gaussian noise into the bank node attributes ($X_i' = X_i(1+\eta_i), \eta \sim \mathcal{N}(0, \sigma^2)$) and reconstructed the networks.

The data used in this paper is sourced from the *CSMAR*[1], which spanning 2007-2023 for 507 banks. The training set uses data from 2007 to 2019, while the test set spans 2020 to 2023.

[1]https://data.csmar.com

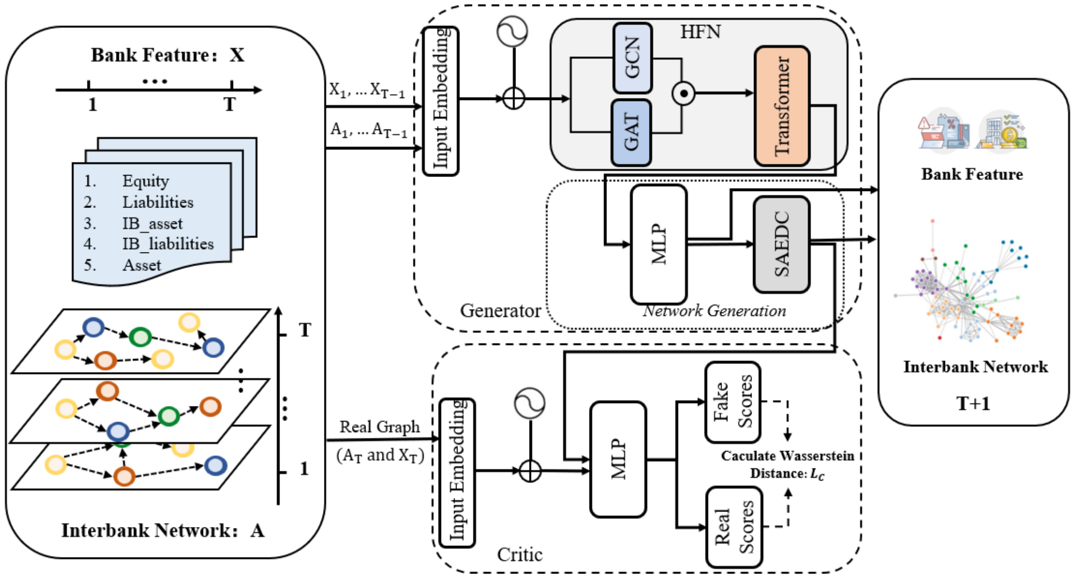

Fig. 1. The overall framework of HFAN-ITF. The Generator takes historical node attributes (X) and interbank networks (A) as input. Its core HFN module extracts spatio-temporal features, which feed a Network Generation module to produce a dense amount matrix. This matrix undergoes density control (SAEDC) to generate the network prediction for T+1. The Critic distinguishes real from generated graphs by computing the Wasserstein distance, guiding stable adversarial training.

B. HFAN-ITF

We propose HFAN-ITF, a generative adversarial framework for interbank network forecasting. As illustrated in Figure 1, the model consists of a Generator, which predicts the network for the next time step, and a Critic, which distinguishes between real and generated graphs to guide the training

The generator produces a complete, weighted, directed graph $\tilde{G}_{t+1} = (\tilde{A}_{t+1}, \tilde{X}_{t+1})$ from historical data. This process involves three key parts: Spatio-Temporal Feature Extraction, Network Generation and the implementation of SAEDC.

(1) HFN for Spatio-Temporal Feature Extraction.

a) Embedding Module.

In this paper, we utilize a `YearEncoder` to process the year information from the input data, and apply the MLP for feature mapping, defined as:

$$T_t = \text{MLP}(\text{YearEncoder}(t)) \tag{2}$$

The MLP consists of two stacked modules, each containing a `Linear` layer, `LayerNorm`, and a `GELU` activation function. After these two modules, an additional `LayerNorm` and `Dropout` operation are applied. Similarly, for node encoding of the input data G_t:

$$\text{GEncoding}(G_t) = \text{MLP}(G_t) \tag{3}$$

b) HFN Module.

Parallel Feature Extraction. The GCN branch emphasizes global structural smoothing. It takes the normalized adjacency matrix $\hat{A}$ and node features X_t to compute the representation $H_{GCN}^{(t)}$:

$$H_{GCN}^{(t)} = \text{ReLU}(\hat{D}^{-1/2}\hat{A}_t\hat{D}^{-1/2}X_t\Theta_{GCN}) \tag{4}$$

Simultaneously, the GAT branch utilizes attention mechanisms to capture heterogeneous neighbor importance. It independently processes the same input (A_t, X_t) to generate $H_{GAT}^{(t)}$:

$$H_{GAT}^{(t)} = \text{GAT}(A_t, X_t) = \Big\Vert_{k=1}^{K} \sigma\left(\sum_{j\in\mathcal{N}_i} \alpha_{ij}^{(k)} \mathbf{\Theta}_{GAT} x_j\right) \tag{5}$$

where $\Vert$ denotes the concatenation of K attention heads.

Fusion and Temporal Modeling. To integrate these complementary features, we concatenate the outputs from both branches and project them via an MLP fusion layer:

$$H_t' = \text{MLP}(H_{GCN}^{(t)} \oplus H_{GAT}^{(t)}) \tag{6}$$

Finally, the sequence of fused representations $\{H_{t-l}', ..., H_t'\}$ is combined with temporal embeddings T_t and fed into the Transformer encoder. This generates the final spatio-temporal embeddings $\{H_{t-l}'', ..., H_t''\}$, which retain both the network topology and evolutionary trends.

(2) Network Generation.

This module transforms the learned node embeddings into a weighted adjacency matrix A_{t+1} by deriving the structure through Edge Scoring and Probabilistic Allocation.

Edge Scoring. First, an MLP scores the likelihood of a link between any lender i and borrower j based on their fused representations (h_i'', h_j'') and the weight:

$$s_{ij} = \text{MLP}_{score}([h_i'' \parallel h_j'' \parallel T_{t+1} \parallel w_{ij}^t]) \tag{7}$$

The *Softmax* to convert these scores into a probability distribution p_{ij}, representing the proportion of capital bank i is willing to lend to bank j:

$$p_{ij} = \frac{\exp(s_{ij})}{\sum_{k\neq i}\exp(s_{ik})} \tag{8}$$

*Probabilistic Allocation.*The weights are allocated according to interbank assets in proportion to p_{ij}, as shown in Eq. 9, here $S_i^{IB,t+1}$ represents interbank assets of bank i.

$$\tilde{w}_{ij} = p_{ij} \cdot S_i^{IB,t+1} \tag{9}$$

(3) SAEDC.

SAEDC selects the final edge set using a global quantile rule based on a learnable density budget. Let S_{ij} denote the selection score for an edge (i, j); Given a learnable network density parameter g(Target Edges per Node), the target edge count is $M_{\text{target}} = \lfloor N \cdot g \rfloor$. The module computes a threshold τ as the $\left(1 - \frac{M_{\text{target}}}{m}\right)$ quantile of scores over all m candidate pairs. It keeps edges where $S_{ij} > \tau$ and resolves any ties via global ranking to produce exactly M_{target} edges. This procedure functions similarly to the "Top-k Selection". Finally, the adjacency matrix A_{t+1} derived through de-normalization and combined with the weighted network selected by SAEDC to constitute the final output$\tilde{G}_{t+1}$.

V. EXPERIMENTS

A. Loss

To ensure stable training and avoid issues like mode collapse, our framework is optimized using the Wasserstein distance. The loss functions for the Critic and Generator are defined as follows.

a) Critic Loss (L_C).

The Critic is trained to maximize the difference in scores between real graphs G and generated graphs $\tilde{G}$. Its loss function, which we aim to minimize, is composed of the estimated Wasserstein distance and a gradient penalty term.

$$L_C = \underbrace{\mathbb{E}_{\tilde{G}\sim P_g}[C(\tilde{G})] - \mathbb{E}_{G\sim P_r}[C(G)]}_{\text{Wasserstein Distance Estimation}} + \underbrace{\lambda \mathbb{E}_{\hat{G}\sim P_{\hat{G}}}\left[(\|\nabla_{\hat{G}} C(\hat{G})\|_2 - 1)^2\right]}_{\text{Gradient Penalty}} \tag{10}$$

Where $C(\cdot)$ is the score output by the Critic for a given graph. The first term encourages the Critic to give higher scores to real graphs and lower scores to generated ones. The Gradient Penalty term, which is central to the WGAN-GP method [26]. It enforces the 1-Lipschitz constraint on the Critic by penalizing gradients whose norms deviate from 1.

b) Generator Loss (L_G).

The Generator is trained to produce graphs that the Critic scores as highly as possible, effectively fooling the Critic into believing the generated graphs are real.

$$L_G = -\mathbb{E}_{\tilde{G}\sim P_g}[C(\tilde{G})] \tag{11}$$

By minimizing L_G, the Generator learns to produce outputs $\tilde{G}$ that receive high scores $C(\tilde{G})$ from the Critic, thus improving the quality of the generated networks.

B. Experimental Settings

(1) Baselines.

To evaluate HFAN-ITF, we compare against several baselines categorized by their core approach. To ensure a rigorous and fair comparison, we construct our baselines using the same foundational architectural components (e.g., GCN, GAT, and GAN frameworks) that underpin current widely-adopted techniques. We note that directly re-implementing certain specialized models from related work is often suboptimal, as their performance is frequently tied to domain-specific design constraints inconsistent with our novel financial task. Consequently, all models in this study are unified under the same training protocol and evaluated on the identical dataset.

TABLE I
CORE HYPERPARAMETERS FOR HFAN-ITF.

Hyperparameter	**Value**
Hidden Dimension	64
GCN / GAT Layers	2
GAT Heads	4
Transformer Layers / Heads	1/8
Initial value of g	3.5
Critic:Generator Update Ratio	5:1
Optimizer	Adam (lr=10^{-3})

(2) Implementation Details.

Our model is implemented in PyTorch and trained on a single NVIDIA V100 GPU. We use the Adam optimizer with a learning rate of 1×10^{-3} for 2000 steps. For the WGAN training, the gradient penalty weight λ_{GP} is set to 10, and the Critic is updated 5 times for every Generator update. Core model hyperparameters are detailed in Table I.

(3) Evaluation Metrics.

To evaluate the predictions, we proposed four metrics for assessing weights, links, and attributes.

- This paper uses Vertex Mean Squared Error ($V\text{-}MSE$) to measure the deviation between predicted and real attributes: $V\text{-}MSE(\mathbf{V}_{t+1}, \tilde{\mathbf{V}}_{t+1}) = \frac{\|\mathbf{V}_{t+1} - \tilde{\mathbf{V}}_{t+1}\|_F^2}{N \cdot N}$, where $\mathbf{V}_{t+1}$ and $\tilde{\mathbf{V}}_{t+1}$ are the true and predicted node attributes at time $t+1$ and N is the number of nodes. Similarly, we employ Vertex Mean Absolute Error ($V\text{-}MAE$) to provide a more robust measure of attribute prediction accuracy, which is less sensitive to outliers than $V\text{-}MSE$: $V\text{-}MAE(\mathbf{V}_{t+1}, \tilde{\mathbf{V}}_{t+1}) = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} |v_{i,j} - \tilde{v}_{i,j}|}{N \cdot N}$, Where $v_{i,j}$ and $\tilde{v}_{i,j}$ denote the individual elements of the true and predicted attribute matrices, respectively.
- **Weight Similarity (Ws):** This paper uses Ws to evaluate similarity between predicted and true weight distributions:

$$W_s(\mathbf{W}_{t+1}, \tilde{\mathbf{W}}_{t+1}) = 1 - (\alpha \delta_{\mu_{t+1}} + \beta \delta_{\sigma_{t+1}})$$

  Where $\delta(x, \tilde{x}) = \frac{|x - \tilde{x}|}{x}$ represents the relative error operator, μ and σ denote mean and standard deviation, and $\alpha = 0.5$, $\beta = 0.5$ are weighting parameters.
- **Density Similarity (Ds):** It quantifies the discrepancy between the generated network and the real-world network in terms of the non-zero edge ratio:

$$\text{Ds}(D_{t+1}, \tilde{D}_{t+1}) = 1 - |D_{t+1} - \tilde{D}_{t+1}|$$

  Where $D_{t+1} = \frac{E_{t+1}}{N(N-1)}$, $\tilde{D}_{t+1} = \frac{\tilde{E}_{t+1}}{N(N-1)}$. Here, E_{t+1} and $\tilde{E}_{t+1}$ denote the real and predicted number of edges at time $t+1$, respectively, and N represents the number of nodes.

- **KL-Divergence (KL):** It measures the information loss between the predicted distribution and the ground-truth distribution. Defined as:

$$\mathrm{KL}(P\|Q) = \sum_k P(k) \log \left(\frac{P(k)}{Q(k)} \right)$$

$P(k) = \frac{\mathrm{Count}(D=k)}{\sum_i \mathrm{Count}(D=i)} + \epsilon$ represents the probability that the real network has degree k, and ϵ is a small positive number to prevent zero probability, taken as 10^{-8} in our paper. Similarly, $Q(k) = \frac{\mathrm{Count}(D=k)}{\sum_i \mathrm{Count}(D=i)} + \epsilon$ represents the probability that the predicted network has degree k. Here, $\mathrm{Count}(D=k)$ denotes the number of nodes with degree k.

C. Training

Convergence Analysis. We compared the training loss curves of three models: the GAN, the HFAN-ITF with the SAEDC module, and the HFAN-ITF without the SAEDC module. As shown in Figure 2, the training process of the GAN is highly unstable, with the loss function exhibiting severe oscillations and failing to converge effectively. In sharp contrast, our proposed HFAN-ITF models demonstrate significantly improved training stability, with smaller fluctuations in the loss function and rapid convergence to lower loss values. This validates the effectiveness of the HFAN-ITF architecture in achieving stable training. Further analysis of the loss curves between the HFAN-ITF with SAEDC and the one without SAEDC reveals that the former converges faster, especially during the early stages of training, reaching lower loss values more quickly. This result strongly supports the role of the SAEDC module in accelerating the convergence of the HFAN-ITF model, indicating its importance in optimizing connectivity density control and enhancing training efficiency.

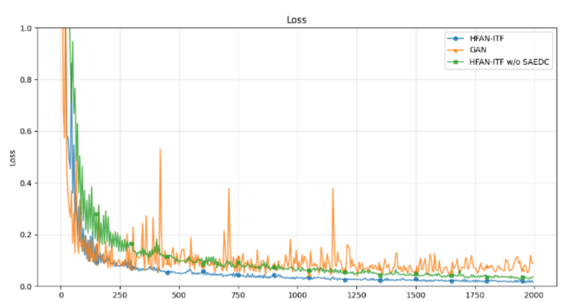

Fig. 2. Training-loss comparison. HFAN-ITF (blue) and its variant without SAEDC (green) are much more stable than a vanilla GAN (orange); with SAEDC the model converges faster in the early stage.

D. Experiments

(1) Network Prediction Comparison Experiment.

As summarized in Table II, HFAN-ITF demonstrates superior performance across all evaluation metrics. In terms of structural fidelity, it achieves a Ws of 0.9594 ± 0.0036, outperforming the most competitive baseline, WGAN+GCN (0.9515 ± 0.0060). Most notably, in distributional similarity, HFAN-ITF attains a KL divergence of 1.8429 ± 0.6654, representing a substantial 45.9% improvement over the second-best performing model, WGAN+LSTM (3.4102 ± 0.7305). These results, combined with a near-perfect Density Similarity ($Ds = 0.9997$), underscore the robustness of our integrated hierarchical fusion and adaptive density control mechanisms in capturing the intricate characteristics of interbank networks.

TABLE II
COMPARISON OF NETWORK STRUCTURE PREDICTION RESULTS (MEAN ± STD)

Model	**Ws (↑)**	**Ds (↑)**	**KL (↓)**
HFAN-ITF	**0.9594** ± 0.0036	**0.9997** ± 0.0003	**1.8429** ± 0.6654
WGAN+RNN	0.5376 ± 0.0085	0.9934 ± 0.0004	5.8979 ± 1.0199
WGAN	0.9452 ± 0.0030	0.9967 ± 0.0004	6.7398 ± 0.9101
WGAN+ Node2Vec	0.0277 ± 0.0626	0.8816 ± 0.0005	25.7610 ± 0.3545
WGAN+LSTM	0.5126 ± 0.0101	0.9916 ± 0.0004	3.4102 ± 0.7305
WGAN+GCN	0.9515 ± 0.0060	0.9967 ± 0.0005	6.8994 ± 0.9134
WGAN+GAT	0.9422 ± 0.0011	0.9967 ± 0.0004	4.6997 ± 1.5777
Diffusion	0.9431 ± 0.0010	0.7522 ± 0.0062	26.0050 ± 0.2153

TABLE III
DETAILED ABLATION STUDY RESULTS

Algorithm	**Ws (↑)**	**Ds (↑)**	**KL (↓)**
HFAN-ITF	**0.9594**	**0.9997**	**1.8429**
Ablation Variants			
w/o GCN	0.9422	0.9967	4.6997
w/o GAT	0.5376	0.9934	5.8979
w/o Transformer	0.0277	0.8816	25.7610
w/o SAEDC	0.9431	0.7522	26.0050
w/o L_c	0.9515	0.9967	6.8994
w/o Node Task	0.9452	0.9967	6.7398

TABLE IV
PERFORMANCE COMPARISON ON ATTRIBUTES

Algorithm	**V-MSE (↓)**	**V-MAE (↓)**
LSTM	0.2313	0.3845
RNN	0.2399	0.3912
Transformer	0.1756	0.3120
HFAN-ITF	**0.1120**	**0.2450**

(2) Ablation Study.

As shown in Table III, the full *HFAN-ITF* model achieves the best performance across all metrics. The ablation study highlights the critical contribution of each component. Removing the *Transformer* causes the most catastrophic degradation ($Ws = 0.0277$), confirming that temporal dependency modeling is the primary pillar of our framework. Disabling *SAEDC* severely impacts structural density and distribution ($Ds = 0.7522, KL = 26.0050$), underscoring the necessity of constraint-aware sparsification. Within the *HFN*, the *GAT* branch is more critical for weight similarity than the

GCN branch (Ws drops to 0.5376 vs. 0.9422), though their combination yields the best results. Ablating L_c or the *Node Task*(attributes prediction) leads to a significant increase in KL divergence, validating that adversarial training and auxiliary supervision are essential for distributional alignment.

(3) Comparison Experiment on Node Attribute Prediction.

For the node attribute prediction task, we evaluate the performance of LSTM, RNN, Transformer, and HFAN-ITF models using both $V\text{-}MSE$ and $V\text{-}MAE$ metrics. As illustrated in Table IV, HFAN-ITF consistently achieves the lowest error rates across all dimensions. Specifically, it yields a $V\text{-}MSE$ of 0.1120, significantly outperforming the second-best model, Transformer (0.1756). Similarly, our model achieves the optimal $V\text{-}MAE$ (0.2450), which further confirms its robustness against outliers compared to traditional RNN and LSTM models (with $V\text{-}MAE$ values of 0.3912 and 0.3845, respectively). These results clearly demonstrate that in terms of node attribute forecasting accuracy, HFAN-ITF exhibits substantial performance advantages by effectively capturing complex spatio-temporal dependencies. Consequently, considering both structural and attribute-level fidelity, our proposed framework proves superior in executing comprehensive interbank network prediction tasks.

VI. Conclusion

In this paper, we proposed HFAN-ITF, a novel generative framework for temporal interbank network forecasting that integrates hierarchical feature capture with stability-oriented training. Extensive experiments demonstrate that HFAN-ITF consistently outperforms current widely-adopted methods. The proposed model can be generalized to similar tasks, such as supply chain networks, where purchase orders (weights) are directly influenced by credit ratings and inventory levels (attributes). Future work will incorporate multi-dimensional metrics and extended prediction windows to comprehensively evaluate the performance of HFAN-ITF.

References

[1] A. Mahdavi Ardekani, "Liquidity, interbank network topology and bank capital," *Review of Accounting and Finance*, vol. 23, no. 1, pp. 40–58, 2024.

[2] Z. Liu, J. Hou, X. Wang, X. Ye, J. Wang, H. Zhao, and X. Bai, "Lion: Linear group rnn for 3d object detection in point clouds," *Advances in Neural Information Processing Systems*, vol. 37, pp. 13 601–13 626, 2024.

[3] Y. Tang, P. Dong, Z. Tang, X. Chu, and J. Liang, "Vmrnn: Integrating vision mamba and lstm for efficient and accurate spatiotemporal forecasting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5663–5673.

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

[5] C. He, X. Zhou, C. Wang, L. Cao, J. Shao, and Z. Tari, "Influence-aware group recommendation for social media propagation," in *Proceedings of the 2024 IEEE International Conference on Data Mining (ICDM)*, 2024.

[6] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[7] J. Shang, Y. Jiao, C. Guo, M. Sun, Y. Gao, J. Liu, M. Momma, I. Taru, and Y. Sun, "Transitivity-encoded graph attention networks for complementary item recommendations," in *2024 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2024, pp. 430–439.

[8] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[9] W. Li, X. Su, S. You, F. Wang, C. Qian, and C. Xu, "Diffnas: Bootstrapping diffusion models by prompting for better architectures," in *2023 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2023, pp. 1121–1126.

[10] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[11] D. Li, Z. Liu, M. Hu, B. Song, and X. Shan, "Gadin: Generative adversarial denoise imputation network for incomplete data," in *2024 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2024, pp. 221–230.

[12] P. Laux, W. Qian, and H. Zhang, "Learning from lending in the interbank network," *Data Science in Science*, vol. 2, no. 1, p. 2151949, 2023.

[13] A. Petropoulos, V. Siakoulis, P. Lazaris, and S. Chatzis, "Re-constructing the interbank links using machine learning techniques. an application to the greek interbank market," *Intelligent Systems with Applications*, vol. 12, p. 200055, 2021.

[14] H. Zhang, "A deep learning approach to dynamic interbank network link prediction," *International Journal of Financial Studies*, vol. 10, no. 3, p. 54, 2022.

[15] S. Takahashi, K. Yamamoto, S. Kobayashi, R. Kondo, and R. Hisano, "Dynamic link and flow prediction in bank transfer networks," in *International Conference on Complex Networks and Their Applications*. Springer, 2024, pp. 125–136.

[16] A. Bojchevski, O. Shchur, D. Zügner, and S. Günnemann, "Netgan: Generating graphs via random walks," in *International conference on machine learning*. PMLR, 2018, pp. 610–619.

[17] H. Tang, D. Xu, Y. Yan, P. H. Torr, and N. Sebe, "Local class-specific and global image-level generative adversarial networks for semantic-guided scene generation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 7870–7879.

[18] H. Wang, J. Wang, J. Wang, M. Zhao, W. Zhang, F. Zhang, X. Xie, and M. Guo, "Graphgan: Graph representation learning with generative adversarial nets," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[19] K. Lei, M. Qin, B. Bai, G. Zhang, and M. Yang, "Gcn-gan: A non-linear temporal link prediction model for weighted dynamic networks," in *IEEE INFOCOM 2019-IEEE conference on computer communications*. IEEE, 2019, pp. 388–396.

[20] X. Guo, S. Wang, and L. Zhao, "Graph neural networks: Graph transformation," *Graph neural networks: Foundations, frontiers, and applications*, pp. 251–275, 2022.

[21] J. Jeon, J. Kim, H. Song, S. Cho, and N. Park, "Gt-gan: General purpose time series synthesis with generative adversarial networks," *Advances in Neural Information Processing Systems*, vol. 35, pp. 36 999–37 010, 2022.

[22] X. Guo and L. Zhao, "A systematic survey on deep generative models for graph generation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 5, pp. 5370–5390, 2022.

[23] J. Yan, H. Hua, W. Huang, X. Fang, W. Ge, J. Yang, and Y. Wang, "Exploring causal information bottleneck for adversarial defense," *IEEE Transactions on Information Forensics and Security*, 2025.

[24] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*. PMLR, 2017, pp. 214–223.

[25] S. A. Assefa, D. Dervovic, M. Mahfouz, R. E. Tillman, P. Reddy, and M. Veloso, "Generating synthetic data in finance: opportunities, challenges and pitfalls," in *Proceedings of the First ACM International Conference on AI in Finance*, 2020, pp. 1–8.

[26] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," *Advances in neural information processing systems*, vol. 30, 2017.