

Graph Neural Diffusion via Generalized Opinion Dynamics

Asela Hevapathige
 asela.hevapathige@anu.edu.au
 School of Computing
 Australian National University
 Canberra, Australia

Asiri Wijesinghe
 asiriwijesinghe.wijesinghe@data61.csiro.au
 Data61
 CSIRO
 Canberra, Australia

Ahad N. Zehmakan
 ahadn.zehmakan@anu.edu.au
 School of Computing
 Australian National University
 Canberra, Australia

Abstract

There has been a growing interest in developing diffusion-based Graph Neural Networks (GNNs), building on the connections between message passing mechanisms in GNNs and physical diffusion processes. However, existing methods suffer from three critical limitations: (1) they rely on homogeneous diffusion with static dynamics, limiting adaptability to diverse graph structures; (2) their depth is constrained by computational overhead and diminishing interpretability; and (3) theoretical understanding of their convergence behavior remains limited. To address these challenges, we propose GODNF, a Generalized Opinion Dynamics Neural Framework, which unifies multiple opinion dynamics models into a principled, trainable diffusion mechanism. Our framework captures heterogeneous diffusion patterns and temporal dynamics via node-specific behavior modeling and dynamic neighborhood influence, while ensuring efficient and interpretable message propagation even at deep layers. We provide a rigorous theoretical analysis demonstrating GODNF's ability to model diverse convergence configurations. Extensive empirical evaluations of node classification and influence estimation tasks confirm GODNF's superiority over state-of-the-art GNNs.

CCS Concepts

- Computing methodologies → Machine learning

Keywords

Graph Neural Networks, Neural Diffusion, Opinion Dynamics, Node Classification, Influence Estimation

ACM Reference Format:

Asela Hevapathige, Asiri Wijesinghe, and Ahad N. Zehmakan. 2025. Graph Neural Diffusion via Generalized Opinion Dynamics. In . ACM, Boise, Idaho, USA , 14 pages. <https://doi.org/XXXXXX.XXXXXXX>

1 Introduction

Graph Neural Networks (GNNs) have become a principal technique for graph learning tasks due to their ability to integrate node features and structural information into meaningful representations. They have been successfully applied to a broad range of problems in various domains, including social network analysis [29, 42, 47],

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference'17, Washington, DC, USA

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM. ACM ISBN 978-1-4503-XXXX-X/2018/06
<https://doi.org/XXXXXX.XXXXXXX>

recommendation systems [6, 18, 63], molecular property prediction [4, 62, 65], and fraud detection [28, 31, 41]. Many GNNs are built on a message passing mechanism, where nodes iteratively exchange information with neighbors to refine their representations, allowing them to model the relational structure of graphs effectively [64].

Recently, researchers have increasingly focused on formulating the message passing process in GNNs as a diffusion phenomenon to better understand and improve the propagation of information in graph data [8, 20, 70]. These models are inspired by physical diffusion processes [23], offering distinct advantages over traditional GNNs as they explicitly model the dynamics of information flow across networks. Although traditional GNNs rely mainly on aggregation operations in the local neighborhood [34, 59], diffusion-based GNNs leverage principles from random walks, heat diffusion, and other physical processes, enabling them to capture local and global structural dependencies in graphs in a more effective manner [20, 36].

Despite their strengths, diffusion GNNs face several notable limitations:

- **Homogeneous Diffusion with Static Dynamics:** Traditional diffusion GNNs assume uniform information propagation across all nodes and fixed dynamics over time [8, 20]. This prevents them from capturing node-specific behaviors and temporal variations in information flow. Therefore, these models struggle to adapt to dynamic, node-dependent diffusion patterns, limiting their effectiveness in complex real-world tasks.
- **Computational Complexity and Interpretability Issues:** Deeper GNNs are often employed to enhance expressiveness, but this comes at the cost of increased model parameters, particularly due to unshared weights, which significantly raises computational overhead [38, 45]. Additionally, deeper architectures obscure the flow of information, making it more difficult to interpret how predictions are formed.
- **Limited Understanding of Convergence Properties:** While existing diffusion-based GNNs offer some insights into basic convergence dynamics, a rigorous theoretical characterization of their representational capacity remains lacking. In particular, the boundaries of the convergence configurations these models can capture are not well defined. Furthermore, the role of diffusion parameters in shaping convergence and representational behavior has not been systematically analyzed, hindering the adaptability of these models to diverse and complex tasks.

To address the aforementioned limitations, we propose incorporating principles from opinion dynamics into diffusion GNNs. Opinion dynamic models, such as the Friedkin-Johnsen [17], are robust mathematical frameworks that are designed to capture how

opinions propagate within a group of agents (i.e., nodes) in a social network [13, 24]. These models trace dynamic behaviors of the underlying propagation, depicting how individual opinions evolve over time and eventually converge. We observe that these models exhibit unique characteristics that can significantly strengthen diffusion GNNs:

- **Heterogeneous Diffusion with Temporal Evolution:** Opinion dynamics usually model node-specific (i.e., agent-specific) behaviors through parameters such as stubbornness (i.e., attachment to initial opinion) and capture temporal evolution through dynamic influence modeling between neighbors, enabling more complex propagations.
- **Efficiency and Interpretability:** Opinion dynamics provide interpretable, sequential information flow that can introduce a strong inductive bias. Unlike traditional GNNs that rely on heavy learnable transformations at each step, opinion dynamics achieve effective propagation through principled update rules with minimal parameterization, remaining simple and coherent even as model depth increases.
- **Theoretically Established Convergence Properties:** Opinion dynamics builds upon well-established mathematical foundations with provable convergence configurations [14, 17, 52], offering stronger theoretical insights into information propagation under diverse conditions. They also provide proper guidance on the impact of their propagation parameters, enabling adaptive parameter tuning based on task requirements.

However, integrating opinion dynamics into GNNs is a challenging task since it requires bridging several technical and conceptual gaps between computational social science and GNNs. This integration must effectively accommodate temporal modeling without incurring excessive computational costs and preserve convergence properties within the context of neural network training. Additionally, incorporating heterogeneous influence mechanisms requires adjustments to traditional message-passing schemes, which introduces new interpretability trade-offs that need careful consideration. Since different opinion dynamics models have distinct advantages, a unified framework is necessary to effectively integrate these mechanisms and fully leverage their potential to enhance GNN performance across various graph learning tasks. To address these challenges, we propose a novel Generalized Opinion Dynamics Neural Framework, *GODNF*, which is a more robust and theoretically grounded alternative to conventional diffusion GNNs.

Contributions. Our main contributions are summarized as follows. (1) We introduce GODNF, a generalized framework grounded in multiple opinion dynamic models, which provides a principled foundation for designing diffusion-based GNNs and enables the efficient modeling of diverse information propagation behaviors. (2) We advance diffusion GNNs by designing a message passing framework that captures evolving temporal dynamics with node-specific responses. (3) We show that our framework can accommodate deep layers while maintaining interpretability and computational efficiency. (4) We provide theoretical insights by proving that our framework guarantees convergence and can exhibit diverse convergence configurations, offering greater flexibility in modeling various diffusion behaviors. (5) Our extensive experiments on both

static and temporal prediction tasks demonstrate that GODNF outperforms leading spatial and diffusion-based GNN models.

2 Related Work

Diffusion GNNs. Diffusion GNNs are inspired by dynamical systems, where the propagation of information through the network layers is analogous to the time evolution of a dynamic process [23]. These GNNs are based on two primary diffusion paradigms: discrete and continuous. Discrete diffusion GNNs propagate information through iterative steps with explicit layer updates [11, 19, 20, 70]. They are often computationally efficient and straightforward to implement, but struggle to capture fine-grained diffusion dynamics, as the use of fixed, uniform update rules limits their ability to model heterogeneous node behaviors and evolving temporal patterns. In contrast, continuous models frame diffusion as dynamical systems governed by differential equations [7, 8, 12, 16]. The adaptive nature of continuous diffusion modeling allows these GNNs to capture intricate diffusion patterns more effectively than their discrete counterparts. Early continuous GNNs, such as GRAND [8], closely mimic heat diffusion but are prone to issues like oversmoothing and fail to capture long-range interactions in graphs effectively [57]. Recent continuous diffusion GNNs have advanced beyond this by employing techniques such as additional damping/source terms [12, 16, 56, 57], external forces [61, 71], and geometric tools [3, 9] to mitigate these problems. Nonetheless, these GNNs require passing node features into differential equation solvers, which introduces high computational overhead and makes it difficult for them to scale to deep layers due to numerical stability issues.

Our framework combines the strengths of both discrete and continuous diffusion GNNs. Unlike standard discrete diffusion GNNs, which rely on uniform update rules, our opinion dynamic-based update rule captures intricate diffusion patterns through dynamic influence modeling and node-specific behavior adaptation, all while maintaining minimal learnable parameters. This provides our model with the adaptability of continuous models without their computational overhead and numerical stability issues. Furthermore, our method builds on well-established mathematical foundations with provable convergence properties, offering better real-world adaptability and interpretability compared to existing diffusion GNNs. As a unified framework, it captures a broader range of diffusion dynamics, resulting enhanced expressive power.

Opinion Dynamics and GNNs. Concepts from opinion dynamic models have influenced the design of GNN architectures. GNNs with residual connections [19, 46, 67, 69] exhibit similarities to the Friedkin–Johnsen model [17], where the residual connections to the initial feature states mimic the concept of stubbornness in the model. Sheaf theory-based GNNs [3] adopt an opinion dynamics-inspired approach, where each node has both public and private opinions, and information propagation is modeled as the interplay between these two types of opinions. Zhou et al. [72], and Vargas-Pérez et al. [58] introduced message passing mechanisms based on the bounded confidence opinion model [52], designed to enhance information propagation in GNNs. Wang et al. [60] proposed a message-passing approach inspired by behavior opinion dynamics [37] to address the oversmoothing issue in GNNs. Compared to these works, our method provides a single unified framework that

encapsulates diverse opinion dynamics behaviours. Recently, Li et al. [39] proposed UniGO, a GNN framework for temporal opinion evolution prediction. While UniGO offers a unified formulation of opinion dynamics, it employs this formulation solely to synthesise training datasets. The actual GNN model instead relies on a separate coarsen-refine mechanism with graph pooling. In contrast, our framework integrates unified opinion dynamics directly into the neural diffusion mechanism, yielding a principled mathematical foundation for message passing. This integration not only provides theoretical convergence guarantees but also ensures broader applicability beyond domain-specific prediction tasks.

3 Background

Let $G = (V, E)$ be an undirected graph, where V denotes the set of nodes, E the set of edges, $|V| = n$, and $|E| = m$. The nodes are represented by the feature matrix $H \in \mathbb{R}^{n \times f}$, where f denotes the number of features per node. Further, $N(i)$ represents the set of nodes adjacent to node i .

3.1 Diffusion Graph Neural Networks

Diffusion GNNs extend message-passing by modeling information propagation as a diffusion process. Unlike local aggregation, they incorporate global structure through operators like the graph Laplacian or transition matrices [20].

A diffusion GNN is described by the evolution of node features over time through a temporal operator \mathcal{D} :

$$\mathcal{D}[X(t)] = \Phi\left(X(0), X(t), G; \Theta\right)$$

where $X(t) \in \mathbb{R}^{n \times d}$ denotes the node features at time t , with d being the feature dimensionality. The initial node features in $X(0)$ are typically derived from the raw node features in H through a learnable transformation. Φ is a learnable diffusion operator parameterized by Θ , which governs how information propagates across G .

Discrete-Time Diffusion. In the discrete case, the operator takes the form $\mathcal{D}[X(t)] = X(t+1) - X(t)$, yielding the iterative update rule:

$$X(t+1) = X(t) + \Phi\left(X(0), X(t), G; \Theta\right)$$

This formulation includes architectures such as APPNP [19] and GDC [20].

Continuous-Time Diffusion. Alternatively, in continuous-time models, the evolution is governed by an ordinary differential equation (ODE):

$$\frac{dX(t)}{dt} = \Phi\left(X(0), X(t), G; \Theta\right)$$

Notable examples include GRAND [8] and PDE-GCN [15].

3.2 Opinion Dynamics Models

Opinion dynamics models describe how the states or opinions of nodes in a network evolve through local interactions, capturing different mechanisms of information propagation.

French-DeGroot (FD) Model [14]. In the FD model, each node updates its opinion as a weighted average of its neighbors' opinions:

$$x_i(t+1) = \sum_{j=1}^n W_{ij} x_j(t)$$

where $x_i(t)$ denotes the opinion of node i at time t . The weight matrix $W \in \mathbb{R}^{n \times n}$ is row-stochastic, with $W_{ij} > 0$ if node i is influenced by node j . When the graph is connected, the opinions of all nodes converge to the same value (i.e., single consensus) [14].

Friedkin-Johnsen (FJ) Model [17]. The FJ model extends FD by allowing each node to retain partial attachment to its initial opinion:

$$x_i(t+1) = \lambda_i x_i(0) + (1 - \lambda_i) \sum_{j=1}^n W_{ij} x_j(t)$$

where $x_i(0)$ is the initial opinion of node i , and $\lambda_i \in [0, 1]$ representing node i 's level of attachment to its initial opinion. This opinion dynamics converges to an equilibrium state that reflects a balance between the influence of neighboring nodes and each node's adherence to its initial opinion, resulting in persistent diversity of opinions across the network [2, 68].

Hegselmann-Krause (HK) Model [52]. In the HK model, each node updates its opinion by averaging those of other nodes whose opinions lie within a specified confidence interval:

$$x_i(t+1) = \frac{1}{|\Gamma(i, t)|} \sum_{j \in \Gamma(i, t)} x_j(t),$$

$$\Gamma(i, t) = \{j \in N(i) \mid |x_i(t) - x_j(t)| \leq \epsilon\}$$

where $\epsilon > 0$ is the confidence bound. Unlike the FD and FJ models, the HK model depicts time-varying neighborhood influence as each node's influential neighbors change dynamically based on opinion proximity at each time step. This opinion dynamics typically leads to multiple opinion clusters instead of a global consensus [10].

4 A Generalized Opinion Dynamics Framework for Graph Neural Diffusion

In this section, we introduce a generalized discrete-time framework that leverages opinion dynamics for diffusion GNNs. We demonstrate how node features in diffusion GNNs evolve similarly to opinions in social networks, facilitating the transfer of insights across domains to develop a more powerful learning method for capturing complex diffusion patterns in graph-structured data.

4.1 Diffusion Mechanism in GODNF

Let $h_i \in H$ denote the feature vector of node i , and f_θ be a neural network parameterized by θ . The update rule of GODNF is defined as follows:

$$x_i(0) = f_\theta(h_i),$$

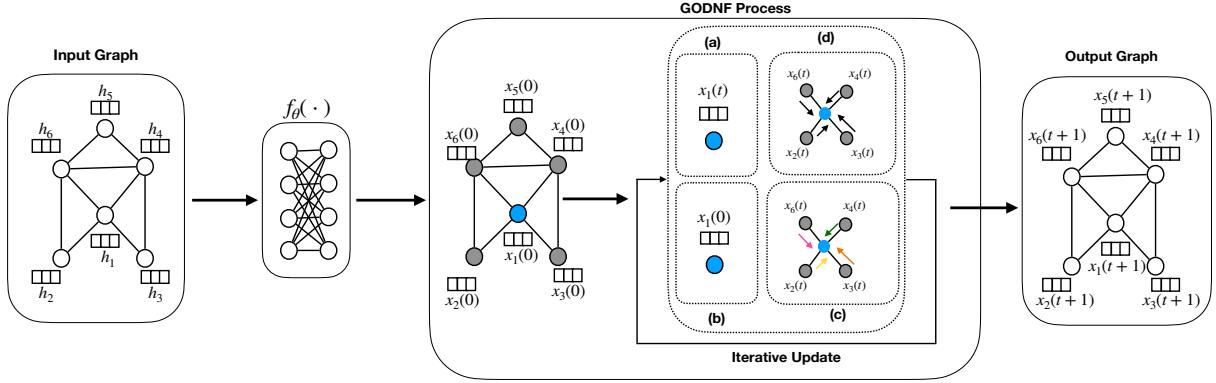


Figure 1: High-Level Architecture of GODNF: Raw features are first transformed to retrieve initial features. These initial features are iteratively updated based on four aspects: (a) Current feature retention, (b) Initial feature attachment, (c) Neighborhood influence, and (d) Structural regularization. The model is trained in an end-to-end manner.

$$\begin{aligned}
 x_i(t+1) = & \underbrace{\alpha x_i(t)}_{(a)} + (1 - \alpha) \left[\lambda_i \underbrace{x_i(0)}_{(b)} + (1 - \lambda_i) \underbrace{\left(\sum_{j \in N(i)} w_{ij}(t) x_j(t) \right)}_{(c)} \right. \\
 & \left. - \mu L_g x_i(t) \right] \quad (1)
 \end{aligned}$$

The update dynamics for a node's feature vector combine current feature retention, initial feature attachment, neighborhood influence, and structural regularization:

- (a) **Current Feature Retention:** $x_i(t)$ represents the node's current feature vector. Controlled by a retention parameter α ($0 \leq \alpha < 1$), this term preserves the node's recent state.
- (b) **Initial Feature Attachment:** $x_i(0)$ represents the node's initial feature after neural transformation, and the learnable parameter λ_i represents the node's stubbornness. A higher λ_i means node i is more attached to its initial feature, showing greater resistance to external influence.
- (c) **Neighborhood Influence:** This term aggregates features from node i 's neighbors. Each neighbor feature $x_j(t)$ is weighted by a time-dependent learnable parameter $w_{ij}(t)$ where $\sum_{j \in N(i)} w_{ij}(t) = 1$. These adaptive weights allow the model to adjust the influence of each neighbor on node i over time, thereby capturing temporal evolutions.
- (d) **Structural Regularization:** This term promotes feature smoothness across the nodes using a Laplacian operator L_g . The learnable coefficient $\mu \in \mathbb{R}_{\geq 0}$ controls the regularization strength, encouraging similar features among connected nodes.

Together, these terms allow GODNF to dynamically balance individual stubbornness with network-level adaptation, enabling nuanced diffusion dynamics over time.

Matrix Form. Let $X(t) \in \mathbb{R}^{n \times d}$ denote the feature matrix at time t , with $X(0)$ as the initial feature matrix after neural transformation. Stubbornness is modeled by a diagonal matrix $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$. Neighbor influence is captured by the time-dependent

row-stochastic influence matrix $W(t) \in \mathbb{R}^{n \times n}$, where $W_{ij}(t)$ denotes the influence of node j on node i at time step t . Structural regularization is enforced via the normalized Laplacian operator $L_g \in \mathbb{R}^{n \times n}$. With these notations, the update rule can be written in matrix form as:

$$\begin{aligned}
 X(0) &= f_\theta(H) \\
 X(t+1) &= \alpha X(t) + (1 - \alpha) [\Lambda X(0) + (I - \Lambda) (W(t) - \mu L_g) X(t)].
 \end{aligned}$$

A Simplified Variant: GODNF_{Static}. We also introduce a simplified variant of our model where the influence matrix $W(t)$ is constant over time ($W(t) = W$ for all t), meaning the influence between nodes remains fixed throughout propagation. The update rule then simplifies to:

$$X(t+1) = \alpha X(t) + (1 - \alpha) [\Lambda X(0) + (I - \Lambda) (W - \mu L_g) X(t)].$$

For brevity, we refer to our original model with the time-dependent influence matrix as **GODNF_{Dynamic}**. A detailed mapping of GODNF components to existing opinion dynamic models is provided in the Appendix B.

4.2 Bounded Weight Evolution

We bound the time-dependent evolution of influence weights in the GODNF_{Dynamic} model to prevent oscillation of $W(t)$, thereby ensuring stability in model training. To achieve this, we employ a decaying update rule:

$$W(t+1) = W(t) + \eta(t) \Delta W(t)$$

where $\eta(t)$ is a time-dependent learning rate that decreases over time with $\lim_{t \rightarrow \infty} \eta(t) = 0$. $\Delta W(t)$ is the weight adjustment computed from the gradient of the objective function w.r.t. $W(t)$. This ensures progressively smaller updates and eventual stabilization of $W(t)$ to a fixed matrix W^* as t increases.

Remarks: $\lim_{t \rightarrow \infty} \eta(t) = 0$ implies the convergence of $W(t)$ into W^* under following conditions: (1) $\|\Delta W(t)\| \leq k$ for all t and some constant $k > 0$; (2) $\sum_{t=0}^{\infty} \eta(t) = \infty$ and

$\sum_{t=0}^{\infty} \eta(t)^2 < \infty$ [53]. Condition (1) is naturally satisfied since $\Delta W(t)$ is computed on gradient weights under a well-behaved training loss function. To satisfy condition (2), we select a simple and efficient function $\eta(t) = \frac{1}{1+t}$.

4.3 Convergence Conditions of GODNF

We present the following theorem regarding the convergence of GODNF. It demonstrates that, under given conditions, the node features will converge to a unique fixed point. The proof is provided in the Appendix.

THEOREM 1 (CONVERGENCE OF GODNF FEATURE UPDATES). *Let $X(t) \in \mathbb{R}^{n \times d}$ evolve according to the update rule in Eq. (1), and define the combined influence matrix $M(t) = (I - \Lambda)(W(t) - \mu L_g)$. When the parameters Λ and μ are chosen such that the operator norm satisfies $\|M(t)\|_{\text{op}} < 1$ for all t , and $W(t) \rightarrow W^*$, the $X(t)$ is guaranteed to converge to a unique fixed point X^* as $t \rightarrow \infty$.*

Remarks: The condition $\|M(t)\|_{\text{op}} < 1$ is realistic and not overly restrictive as $W(t)$ (row-stochastic with eigenvalues in $[-1, 1]$) and L_g (normalized Laplacian with eigenvalues in $[0, 2]$) can be appropriately scaled by learnable parameters Λ and μ to satisfy this bound regardless of graph structure.

4.4 Convergence-Preserving Regularization

According to Theorem 1, convergence of GODNF requires $\|M(t)\|_{\text{op}} < 1$ at every time step. Therefore, we design a regularization term enforcing this condition. However, computing $\|M(t)\|_{\text{op}}$ can be expensive in practice. To address this, we employ an efficient approximation based on a well-established matrix norm inequality. The operator norm of a matrix is bounded above by the geometric mean of its 1-norm and infinity-norm [26],

$$\|M(t)\|_{\text{op}} \leq \sqrt{\|M(t)\|_1 \cdot \|M(t)\|_{\infty}},$$

Unlike the op-norm, the infinity-norm and 1-norm can be computed in linear time.

Regularization Term. We design a regularization term to enforce the condition $\sqrt{\|M(t)\|_1 \cdot \|M(t)\|_{\infty}} < 1$, penalizing violations when this condition is not satisfied. The regularization term is:

$$\mathcal{L}_{\text{reg}} = \sum_{t=0}^T \max \left(0, \sqrt{\|M(t)\|_1 \cdot \|M(t)\|_{\infty}} - 1 \right).$$

In GODNF_{static} variant, since $W(t) = W$ is constant, which makes $M(t) = M$ time invariant, the regularization term can be simplified to:

$$\mathcal{L}_{\text{reg}} = \max \left(0, \sqrt{\|M\|_1 \cdot \|M\|_{\infty}} - 1 \right).$$

This eliminates the need to sum over time steps, reducing computational complexity and making GODNF_{static} more efficient than its dynamic counterpart.

Training Loss. We define training loss of GODNF as a combination of downstream task loss and the regularization term as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \mathcal{L}_{\text{reg}}.$$

5 Theoretical Analysis

5.1 Complexity Analysis

The time complexity of GODNF comprises the initialization and iterative update phases. Initial feature transformation costs $O(n \cdot p_f)$, where p_f is the upper bound on the cost of computing $f_{\theta}(h_i)$. The complexity of each iteration incurred by GODNF components is: initial feature attachment $O(n \cdot d)$, neighborhood influence and structural regularization both $O(m \cdot d)$ through sparse matrix multiplication, and norm regularization term $O(m)$, which is efficient due to the sparsity of $M(t)$. Here, d is the feature dimension. The total time complexity over T iterations is $O(n \cdot p_f + T \cdot (m \cdot d + n \cdot d + m)) = O(n \cdot p_f + T \cdot m \cdot d)$, using the graph connectivity assumption. In practical scenarios where p_f , d , and T are usually small, this complexity is bounded by $O(m)$, which is asymptotically optimal.

The space complexity of GODNF includes storing feature matrices $X(0)$ and $X(t)$ with $O(n \cdot d)$, the sparse matrices $W(t)$ and L_g with $O(m)$, and the diagonal matrix Λ with $O(n)$. Overall, the space complexity is $O(n \cdot d + m)$.

Overall, GODNF achieves comparable asymptotic complexity to traditional naive GNNs such as GCN [34]. The absence of intermediate feature transformations in GODNF can lead to computational efficiency gains, especially for deeper layers. An empirical analysis of the scalability of our approach is provided in Section 6.3.3.

5.2 Representation of Diverse Convergence Configurations

In this section, we demonstrate that GODNF can capture multiple convergence configurations, emphasizing its adaptability to diverse learning scenarios. We start by defining these convergence configurations and show that GODNF can theoretically converge to these diverse configurations by adjusting its components. For the following theorems, we consider the following conditions of GODNF to hold: (1) $W^t \rightarrow W^*$, (2) $\|M(t)\|_{\text{op}} < 1$ for all t where $M(t) = (I - \Lambda)(W(t) - \mu L_g)$. All the proofs are provided in the Appendix.

5.2.1 Single Consensus.

Definition 1. A diffusion model is said to reach a *single consensus* if there exists a vector $v \in \mathbb{R}^d$ such that, for all nodes $i \in V$,

$$\lim_{t \rightarrow \infty} x_i(t) = v,$$

We now establish conditions under which GODNF converges to a single consensus.

THEOREM 2 (SINGLE CONSENSUS UNDER FULLY DIFFUSIVE DYNAMICS). *GODNF reaches a single consensus if $\Lambda = 0$, $\mu = 0$, $\alpha = 0$, and W^* is a row-stochastic matrix.*

Single consensus is valuable for modeling linear diffusion processes on networks, such as heat diffusion and basic opinion dynamics, as it captures the convergence of these processes.

	Texas	Cornell	Wisconsin	Film	Amazon-rating	Cora Full	CiteSeer	Cora-ML	PubMed	DBLP
GCN	60.00 \pm 6.45	55.14 \pm 8.46	61.60 \pm 7.00	32.16 \pm 1.34	37.99 \pm 0.61	68.06 \pm 0.98	77.72 \pm 1.18	87.07 \pm 1.21	86.74 \pm 0.47	83.93 \pm 0.84
GAT	61.21 \pm 8.17	53.64 \pm 11.10	60.00 \pm 11.00	32.63 \pm 1.61	42.52 \pm 1.22	67.55 \pm 1.23	76.79 \pm 1.00	84.12 \pm 0.55	87.24 \pm 0.55	80.61 \pm 1.21
GraphSAGE	82.46 \pm 5.19	75.32 \pm 5.96	80.00 \pm 4.81	36.00 \pm 1.20	41.32 \pm 0.63	69.77 \pm 0.46	77.42 \pm 0.83	86.52 \pm 1.32	84.50 \pm 0.39	86.16 \pm 0.50
AirGNN	65.57 \pm 11.19	53.19 \pm 10.43	56.25 \pm 5.03	36.34 \pm 2.73	42.29 \pm 0.88	65.27 \pm 0.72	78.20 \pm 1.17	88.09 \pm 1.25	83.51 \pm 0.87	84.69 \pm 0.30
DirGNN	76.25 \pm 6.31	76.51 \pm 6.14	80.50 \pm 5.50	35.76 \pm 1.68	46.66 \pm 0.61	67.80 \pm 0.53	77.71 \pm 0.78	85.66 \pm 0.31	86.94 \pm 0.55	81.22 \pm 0.54
BEC-GCN	68.85 \pm 11.02	65.96 \pm 9.03	66.25 \pm 4.55	34.34 \pm 2.08	42.17 \pm 0.58	71.34 \pm 0.44	78.14 \pm 1.31	88.65 \pm 1.32	87.00 \pm 0.28	85.20 \pm 0.48
APPNP	62.30 \pm 6.65	58.75 \pm 5.97	61.70 \pm 6.94	36.44 \pm 1.78	43.48 \pm 0.68	70.63 \pm 0.51	77.89 \pm 1.07	87.41 \pm 1.54	83.37 \pm 0.42	85.36 \pm 0.40
GDC	72.13 \pm 5.48	53.19 \pm 12.19	62.50 \pm 5.13	30.65 \pm 1.20	40.06 \pm 2.50	62.23 \pm 0.59	78.17 \pm 1.11	84.41 \pm 1.59	80.61 \pm 0.47	84.87 \pm 0.40
ODNet	85.25 \pm 8.36	86.49 \pm 4.06	88.75 \pm 2.15	37.57 \pm 0.94	43.33 \pm 0.91	70.23 \pm 0.43	74.76 \pm 1.15	76.26 \pm 1.03	86.74 \pm 0.37	83.45 \pm 0.47
GRAND	81.70 \pm 8.42	81.76 \pm 13.90	84.00 \pm 7.50	27.93 \pm 1.25	37.53 \pm 0.36	67.66 \pm 1.01	76.55 \pm 1.69	88.49 \pm 0.81	86.79 \pm 0.57	84.60 \pm 0.99
GRAND++	79.34 \pm 7.22	81.34 \pm 7.12	81.50 \pm 6.00	36.32 \pm 0.50	38.01 \pm 0.50	67.53 \pm 0.74	78.51 \pm 1.58	88.44 \pm 0.53	87.21 \pm 0.33	85.21 \pm 0.24
ACMP	87.65 \pm 3.54	85.66 \pm 5.10	86.50 \pm 5.00	34.44 \pm 1.36	37.32 \pm 0.64	71.76 \pm 0.03	75.73 \pm 1.38	76.11 \pm 2.12	88.01 \pm 1.44	82.31 \pm 0.44
HiD-Net	77.11 \pm 6.91	83.33 \pm 7.10	81.30 \pm 6.60	28.86 \pm 1.00	41.19 \pm 1.03	68.11 \pm 0.64	77.85 \pm 0.65	89.00 \pm 0.51	88.60 \pm 0.45	84.92 \pm 0.31
GNRF	87.39 \pm 4.13	87.28 \pm 3.12	88.00 \pm 2.00	34.22 \pm 1.40	46.89 \pm 1.08	72.12 \pm 0.50	75.79 \pm 0.94	89.18 \pm 0.19	90.37 \pm 0.69	85.73 \pm 0.76
GODNF _{Static}	92.95 \pm 2.32	87.23 \pm 4.66	91.62 \pm 3.49	40.21 \pm 1.85	49.17 \pm 1.06	72.60 \pm 0.58	78.83 \pm 1.22	88.85 \pm 1.52	90.66 \pm 0.47	88.24 \pm 2.01
GODNF _{Dynamic}	93.93 \pm 2.65	87.87 \pm 4.57	92.50 \pm 2.56	39.96 \pm 2.21	49.40 \pm 0.44	72.51 \pm 0.53	78.85 \pm 0.99	89.26 \pm 1.13	90.78 \pm 0.68	88.62 \pm 1.92

Table 1: Node classification accuracy \pm standard deviation (%). The best and second-best results are highlighted in blue, and orange, respectively. The baseline results are sourced from Chen et al. [9].

5.2.2 Multi Consensus.

Definition 2. A diffusion model is said to reach a *multi consensus* if there exist distinct vectors $v_1, v_2, \dots, v_k \in \mathbb{R}^d$ with $k < n$ such that, for all nodes $i \in V$,

$$\lim_{t \rightarrow \infty} x_i(t) = v_j, \quad j \in \{1, 2, \dots, k\}.$$

We now present the conditions under which GODNF converges to a multi consensus.

THEOREM 3 (MULTI-CONSENSUS UNDER BLOCK-STRUCTURED DYNAMICS). GODNF reaches multi consensus if the matrix $M^* = (1 - \Lambda)(W^* - \mu L_g)$ has a block diagonal structure corresponding to multiple disconnected components, with initial conditions and parameters differ for at least two components.

The multi consensus is beneficial for multi-label classification as it can avoid the oversmoothing problem in traditional GNNs. GODNF can preserve multiple clusters rather than converging to a single value, effectively aligning node features with class boundaries and handling both homophilic and heterophilic scenarios.

5.2.3 Individualized Consensus.

Definition 3. A diffusion model is said to reach an *individualized consensus* if there exist distinct vectors $v_1, v_2, \dots, v_n \in \mathbb{R}^d$ such that, for all nodes $i \in V$,

$$\lim_{t \rightarrow \infty} x_i(t) = v_i, \quad i \in \{1, 2, \dots, n\}, \text{ with } v_i \neq v_j \text{ for } i \neq j.$$

We next look into the conditions that lead GODNF to reach individualized consensus, where each node converges to a unique value.

THEOREM 4 (INDIVIDUALIZED CONSENSUS UNDER STRONG FEATURE RETENTION). GODNF reaches individualized consensus if the initial features $x_i(0)$ are distinct for all nodes $i \in V$, and $\lambda_i \in (0, 1]$ are sufficiently large (i.e., close to 1).

Individualized consensus is beneficial for node-level prediction tasks like regression. Unlike standard GNNs that oversmooth features, this consensus maintains distinct node features while still

capturing network structure, offering adaptability to diverse graph settings where preserving node identity is essential.

Remarks: In addition to the three fundamental convergence configurations above, GODNF can adapt to more complex convergence patterns due to its unification of diverse opinion dynamics mechanisms. This flexibility enables GODNF to align with sophisticated convergence configurations that capture the nuances present in the ground truth.

6 Experiments

We evaluate GODNF on two types of tasks: node classification and influence estimation. The goal of the node classification task is to predict the label of individual nodes in a graph. The influence estimation is a node regression task that focuses on determining a node's susceptibility to activation given a specific diffusion model and an initial set of activated nodes [66]. Although node classification operates as a static downstream task, influence estimation requires capturing the temporal dynamics of the diffusion process to learn the underlying propagation characteristics accurately.

6.1 Experimental Setup

6.1.1 Datasets. For node classification, we use ten benchmark datasets that include both homophily and heterophily label patterns. For homophily datasets, we use the Cora-ML, CiteSeer, PubMed, DBLP, and Cora Full datasets from the CitationFull benchmark [5]. For heterophily datasets, we include Texas, Cornell, Wisconsin, and Film datasets from WebKB [50], and the Amazon-rating dataset from the Heterophilous Graph benchmark [51]. Further, we employ the OGB benchmark's ogbn-arxiv dataset [27] for our scalability analysis. For influence estimation, we use four real-world datasets: Jazz, Network Science, Power Grid [55], and Cora-ML [5].

	Jazz			Cora-ML			Network Science			Power Grid		
	IC	LT	SIS									
GCN	0.233 \pm 0.010	0.199 \pm 0.006	0.344 \pm 0.023	0.277 \pm 0.007	0.255 \pm 0.008	0.365 \pm 0.065	0.270 \pm 0.019	0.190 \pm 0.012	0.180 \pm 0.007	0.313 \pm 0.024	0.335 \pm 0.023	0.207 \pm 0.001
GAT	0.342 \pm 0.005	0.156 \pm 0.100	0.288 \pm 0.017	0.352 \pm 0.004	0.192 \pm 0.010	0.208 \pm 0.008	0.274 \pm 0.002	0.114 \pm 0.008	0.123 \pm 0.013	0.331 \pm 0.002	0.280 \pm 0.015	0.133 \pm 0.001
GraphSAGE	0.201 \pm 0.028	0.120 \pm 0.004	0.301 \pm 0.018	0.255 \pm 0.010	0.203 \pm 0.019	0.222 \pm 0.051	0.241 \pm 0.010	0.112 \pm 0.005	0.102 \pm 0.005	0.313 \pm 0.024	0.341 \pm 0.018	0.133 \pm 0.001
DirGNN	0.205 \pm 0.016	0.124 \pm 0.002	0.379 \pm 0.032	0.255 \pm 0.006	0.193 \pm 0.012	0.229 \pm 0.025	0.236 \pm 0.011	0.084 \pm 0.017	0.142 \pm 0.019	0.287 \pm 0.010	0.257 \pm 0.062	0.132 \pm 0.001
SGNN	0.183 \pm 0.004	0.164 \pm 0.014	0.330 \pm 0.007	0.210 \pm 0.003	0.134 \pm 0.004	0.211 \pm 0.006	0.213 \pm 0.003	0.049 \pm 0.004	0.127 \pm 0.00	0.257 \pm 0.002	0.192 \pm 0.001	0.175 \pm 0.004
UniGO	0.192 \pm 0.013	0.159 \pm 0.020	0.335 \pm 0.002	0.255 \pm 0.001	0.155 \pm 0.019	0.212 \pm 0.001	0.201 \pm 0.001	0.110 \pm 0.049	0.108 \pm 0.023	0.231 \pm 0.002	0.235 \pm 0.005	0.206 \pm 0.023
DeepIS	0.151 \pm 0.003	0.219 \pm 0.002	0.434 \pm 0.003	0.203 \pm 0.001	0.301 \pm 0.005	0.304 \pm 0.001	0.223 \pm 0.001	0.306 \pm 0.001	0.256 \pm 0.001	0.206 \pm 0.001	0.374 \pm 0.001	0.251 \pm 0.001
DeepIM	0.178 \pm 0.002	0.134 \pm 0.014	0.383 \pm 0.010	0.210 \pm 0.002	0.271 \pm 0.010	0.270 \pm 0.006	0.216 \pm 0.003	0.118 \pm 0.000	0.135 \pm 0.004	0.258 \pm 0.003	0.331 \pm 0.002	0.205 \pm 0.005
GLIE	0.136 \pm 0.003	0.055 \pm 0.028	0.454 \pm 0.062	0.199 \pm 0.041	0.286 \pm 0.016	0.205 \pm 0.029	0.163 \pm 0.031	0.160 \pm 0.063	0.103 \pm 0.023	0.183 \pm 0.009	0.384 \pm 0.020	0.132 \pm 0.026
APPNP	0.200 \pm 0.006	0.124 \pm 0.003	0.357 \pm 0.059	0.265 \pm 0.022	0.220 \pm 0.031	0.321 \pm 0.042	0.248 \pm 0.006	0.084 \pm 0.000	0.100 \pm 0.012	0.290 \pm 0.006	0.189 \pm 0.011	0.132 \pm 0.001
ODNet	0.180 \pm 0.003	0.053 \pm 0.003	0.322 \pm 0.006	0.232 \pm 0.001	0.210 \pm 0.004	0.196 \pm 0.002	0.255 \pm 0.002	0.104 \pm 0.015	0.106 \pm 0.005	0.274 \pm 0.001	0.296 \pm 0.002	0.145 \pm 0.001
HIDNet	0.216 \pm 0.003	0.180 \pm 0.003	0.404 \pm 0.005	0.258 \pm 0.001	0.273 \pm 0.006	0.360 \pm 0.003	0.242 \pm 0.002	0.166 \pm 0.005	0.168 \pm 0.001	0.282 \pm 0.001	0.295 \pm 0.007	0.224 \pm 0.001
GNRF	0.195 \pm 0.006	0.150 \pm 0.020	0.444 \pm 0.024	0.255 \pm 0.005	0.312 \pm 0.029	0.280 \pm 0.027	0.294 \pm 0.005	0.293 \pm 0.001	0.186 \pm 0.006	0.280 \pm 0.003	0.459 \pm 0.019	0.196 \pm 0.018
GODNF _{Static}	0.177 \pm 0.005	0.051 \pm 0.002	0.318 \pm 0.004	0.150 \pm 0.001	0.104 \pm 0.005	0.197 \pm 0.001	0.205 \pm 0.002	0.036 \pm 0.002	0.098 \pm 0.002	0.223 \pm 0.002	0.116 \pm 0.003	0.120 \pm 0.002
GODNF _{Dynamic}	0.175 \pm 0.003	0.050 \pm 0.001	0.309 \pm 0.006	0.148 \pm 0.001	0.094 \pm 0.005	0.192 \pm 0.002	0.197 \pm 0.002	0.032 \pm 0.003	0.096 \pm 0.001	0.221 \pm 0.001	0.108 \pm 0.004	0.119 \pm 0.002

Table 2: Influence estimation mean absolute error \pm standard deviation. Lower value indicates better performance. The best and second-best results are highlighted in blue, and orange, respectively.

6.1.2 Baselines. For node classification, we compare GODNF against both spatial and diffusion-based GNNs. Our spatial GNN baselines consist of three classical architectures: GCN [34], GAT [59], and GraphSAGE [22], as well as three advanced models: AirGNN [46], DirGNN [54], and BEC-GCN [25]. Diffusion GNNs include both discrete and continuous variants, spanning classical and state-of-the-art models such as APPNP [19], GDC [20], ODNET [72], GRAND [8], GRAND++ [57], ACMP [61], HiD-Net [43], and GNRF [9]. AirGNN and APPNP have residual connections in their architectures that are conceptually similar to the FJ model, while ODNet is designed based on the HK model. For influence estimation, we use a comprehensive set of aforementioned general-purpose baselines that span spatial and diffusion GNNs and GNNs specifically designed for the influence estimation task. These specialized models include SGNN [35], DeepIM [44], DeepIS [66], and GLIE [49]. Additionally, we tailor UniGO [39] as a baseline for influence estimation.

6.1.3 Evaluation settings. For node classification, we use a 60:20:20 random split strategy for the training, validation, and test sets and report the mean and standard deviation of accuracy over 10 random initializations, following the experimental setup of Chen et al. [9]. We evaluate influence estimation performance under three diffusion models: Linear Threshold (LT) [30], Independent Cascade (IC) [21], and Susceptible-Infected-Susceptible (SIS) [32]. LT and IC are progressive diffusion models (i.e., nodes activate once and remain activated permanently), whereas SIS is non-progressive (i.e., nodes can undergo multiple activation cycles) [40]. For data splits, we follow the experimental setup of Ling et al. [44], with an initial activation node set consisting of 10% of the total nodes in the graph. For evaluation, we employ 10-fold cross-validation and report the mean and standard deviation of Mean Absolute Error (MAE). For baseline results, we report the results from previous papers that used the same experimental setup. When such results are unavailable, we generate baseline results by adopting the hyperparameter configurations specified in the original papers.

Additional experimental details, including model hyperparameters, implementation details, and benchmark dataset statistics, are provided in the Appendix C.

6.2 Main Results

6.2.1 Node Classification. We present node classification results in Table 1. GODNF consistently outperforms baseline models in both heterophily and homophily datasets. In particular, GODNF demonstrates significant superiority over spatial and discrete diffusion GNNs, especially within heterophily datasets. Additionally, GODNF comfortably surpasses continuous diffusion GNNs by exceeding their expressive power. We credit GODNF’s success to its capacity to model complex diffusion dynamics in graphs, which is enhanced by its representation power derived from multiple opinion dynamic components. The performance of models such as ODNET, APPNP, and AirGNN further supports this point. Although these models incorporate elements related to opinion dynamics, they do not achieve the same level of performance as GODNF, primarily because their components lack the representation capabilities found in GODNF. Further, we observe that the dynamic variant of GODNF performs better than the static variant because it captures temporal dependencies between diffusion time steps.

6.2.2 Influence Estimation. We report influence estimation results in Table 2. GODNF outperforms or provides comparable performance to both general-purpose spatial and diffusion GNNs, as well as GNNs specifically designed for influence estimation. While DeepIS and GLIE excel in the IC model due to their tailored update rules, they exhibit suboptimal performance within other diffusion models. In contrast, GODNF shows robust generalizability, consistently maintaining better performance across all diffusion models. Additionally, GODNF_{dynamic} consistently outperforms GODNF_{static} because the temporal dynamics captured by the dynamic variant are beneficial for tasks like influence estimation, which rely on temporal characteristics.

6.2.3 Robustness to adversarial attacks. We evaluate the robustness of GODNF under a diverse set of adversarial attack scenarios, benchmarking its resilience against four distinct attack strategies and comparing it with established baselines. In the feature corruption setting, Gaussian noise is injected into node features by sampling a noise matrix and adding it to the original features, following Li et al. [43]. The edge addition attack perturbs the topology by inserting

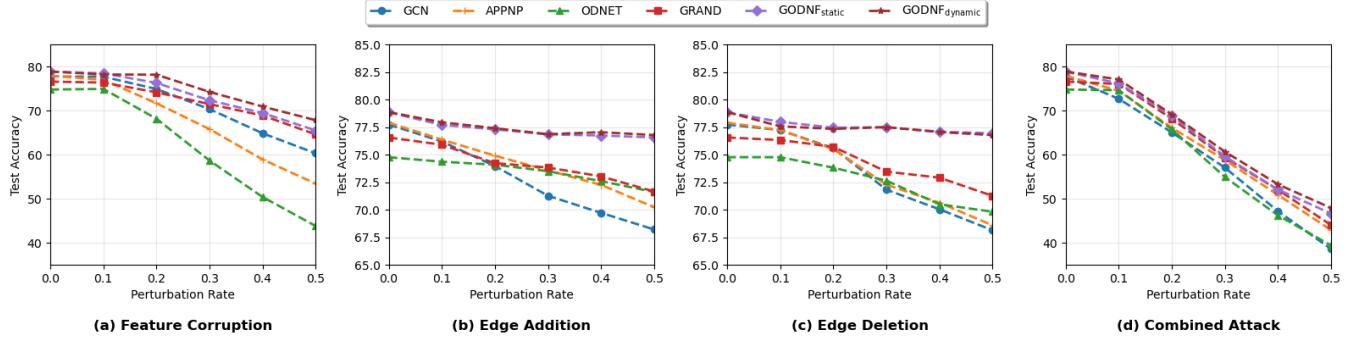


Figure 2: Node classification performance for Citeseer dataset under different adversarial attacks

random edges (excluding self-loops and duplicates) at a specified perturbation rate, while edge deletion removes randomly selected edges. The combined attack applies edge deletion, edge addition, and feature corruption sequentially at the same perturbation rate. Results for the Citeseer dataset are presented in Figure 2, with additional results for other datasets provided in the Appendix C.4.2.

GODNF exhibits consistent and superior robustness across all adversarial attack types. Under feature corruption, it shows only gradual performance degradation versus steep drops in baselines. For structural attacks such as edge addition or deletion, GODNF variants maintain near-constant performance while baselines degrade significantly. Combined attacks further accentuate GODNF's resilience. This robustness stems from GODNF's multi-component defence mechanism, creating redundant information pathways. The initial feature attachment acts as persistent memory, structural regularization provides graph-based smoothing, and bounded weight evolution ensures stability. When attacks compromise one component, the others compensate, thereby preventing the single-pathway failures common in standard GNNs.

6.2.4 Oversmoothing Analysis. We evaluate the performance of GODNF in a deep layer setting to understand its resilience to oversmoothing. The results are depicted in Figure 3. Unlike traditional GNNs like GCN and GRAND, which show substantial performance degradation when the layer depth is increased, both GODNF variants show robustness against oversmoothing by maintaining consistent performance across varying layer depths.

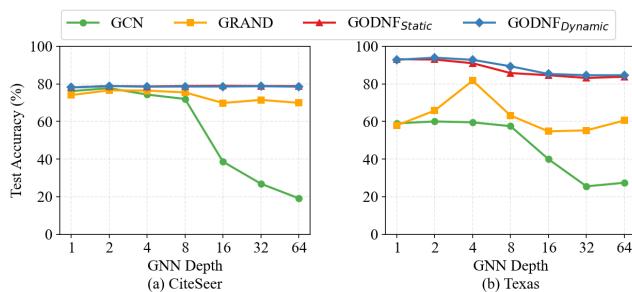


Figure 3: Deep layer comparison for node classification

6.3 Ablation Studies

6.3.1 Impact of Different Components. We conduct an ablation study to evaluate the importance of each component of GODNF on its performance. To demonstrate this, we derive variants of GODNF by omitting each component and assess their performance. The results are depicted in Figure 4.

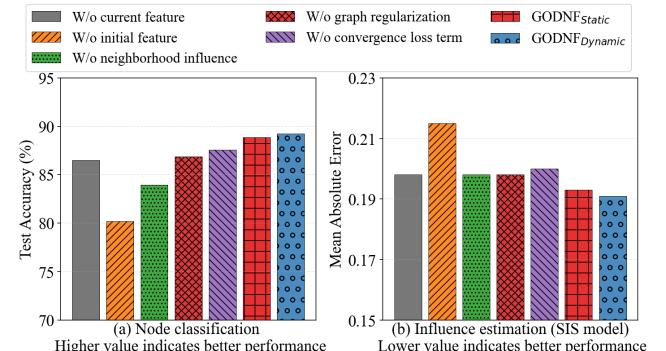
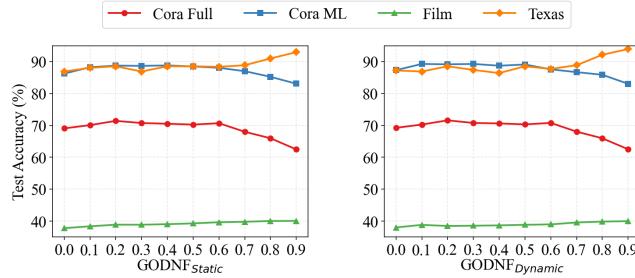


Figure 4: Performance comparison of GODNF component variants on Cora ML dataset

As anticipated, removing each component decreases performance, demonstrating their individual importance. Further, we observe that the initial feature attachment has more impact on performance than other components. We believe this is due to complex node-specific behaviors captured by that component.

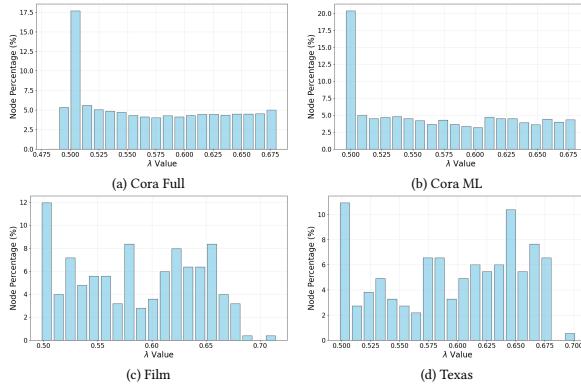
6.3.2 Parameter Analysis. We evaluate the sensitivity of GODNF to its hyperparameter α in Eq. (1), by varying its value from 0 to 0.9. In Figure 5, we observe that the performance of homophilic datasets (Cora ML and Coral Full) in node classification tends to decrease slowly as α increases, while the opposite behavior can be seen in heterophilic datasets (Film and Texas). This is because a decrease of α promotes neighborhood information aggregation, which helps homophily datasets where neighbors tend to share the same label, but adversely impacts heterophilic datasets, where neighbors tend to have different labels.

Then, we present the distribution of learned node stubbornness λ for datasets under node classification in Figure 6. The results reveal

Figure 5: Sensitivity analysis of α

few interesting observations. First, most nodes exhibit moderate stubbornness (0.4-0.7), highlighting the importance of balancing initial features with neighborhood information. Second, the distribution variability shows adaptive learning of heterogeneous aggregation requirements across nodes. Third, heterophily datasets tend to yield higher stubbornness values compared to homophily datasets, suggesting that when neighbors have dissimilar labels, nodes benefit from retaining their initial representations to mitigate information loss from inappropriate aggregation.

We further observe that the learned value for μ is in the range of 0.7-0.9 for most datasets. We believe this range offers essential structural regularisation for convergence stability.

Figure 6: Distribution of learned node stubbornness λ_i

6.3.3 Scalability Analysis. We compare the scalability of our approach with existing GNNs in Table 3. For this, we perform node classification task and employ two datasets: Cora Full and ogbn-arxiv. ogbn-arxiv is a large-scale dataset with 169,343 nodes and 1,166,243 edges. For a fair comparison, we fix the hidden layer size at 128 for all GNNs, and record average runtime per epoch, number of learnable parameters, and accuracy metrics.

As observed, GODNF is significantly more effective than traditional GNNs like GAT and diffusion GNNs like GRAND and ACMP. Furthermore, GODNF has a runtime and parameter complexity comparable to that of highly scalable traditional GNNs like GCN. This efficiency primarily comes from eliminating the learnable transformation layers found in conventional GNNs, resulting in less parameter count and memory usage, even at increasing depths.

Layers	Model	Cora Full			ogbn-arxiv		
		# Param (k)	Time (ms)	Acc. (%)	# Param (k)	Time (ms)	Acc. (%)
4	GCN	1157	20.6	68.44	55	154.2	69.53
	GAT	4631	65.3	59.87	206	OOM	OOM
	GRAND	716	111.1	69.27	81	2077.0	69.30
	ACMP	1158	1107.6	70.94	55	OOM	OOM
	GODNF _{Static}	1144	31.7	72.42	191	926.4	69.76
	GODNF _{Dynamic}	1144	35.5	72.58	191	936.9	70.43
8	GCN	1223	34.6	65.87	123	327.2	68.39
	GAT	4897	121.3	31.10	474	OOM	OOM
	GRAND	716	150.7	69.81	81	3196.5	67.24
	ACMP	1158	2231.5	71.76	55	OOM	OOM
	GODNF _{Static}	1144	51.1	72.42	191	2156.1	70.23
	GODNF _{Dynamic}	1144	63.4	72.49	191	2378.4	70.91

Table 3: Scalability comparison for node classification. Best results are highlighted.

6.3.4 Case Study. We empirically validate GODNF’s ability to depict multiple convergence configurations. Specifically, we create a stochastic block model (SBM) graph [1] (50 nodes and five communities) and generate node labels to be aligned with different convergence configurations, as described in Section 5.2, and train our model to learn these label patterns. In a multi-consensus scenario, we examine homophily and heterophily label patterns separately. The visualization of color-coded node labels learned by our model for each convergence configuration is depicted in Figure 7.

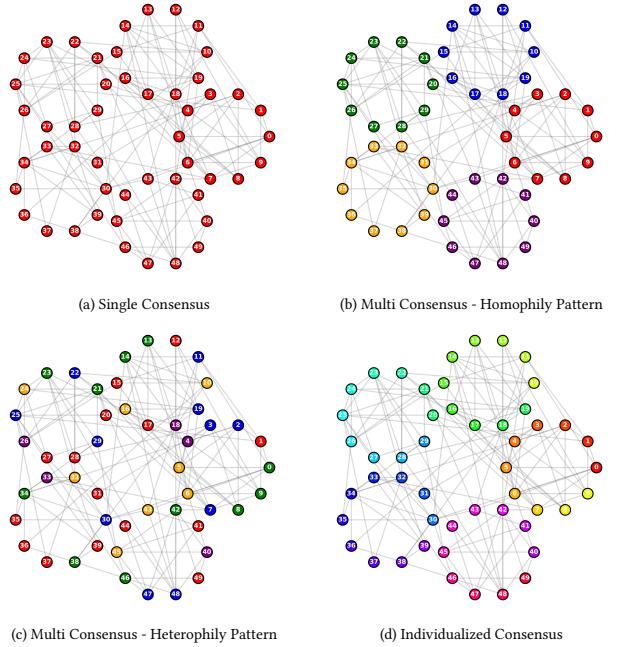


Figure 7: Visualisation of colour-coded node labels learned by GODNF for different convergence configurations.

In the single consensus (Figure 5a), GODNF learns to assign identical labels across all nodes. In the multi-consensus scenario with homophily labels (Figure 5b), our model correctly identifies the pattern with neighbors having similar labels. In contrast, the multi-consensus configuration with heterophily labels (Figure 5c)

yields more intricate patterns, with neighboring nodes exhibiting markedly different labels. Finally, the individualized consensus (Figure 5d) shows the highest color diversity, capturing the personalized nature of node consensus.

6.3.5 Additional Experiments. An experiment illustrating the convergence behaviour of GODNF variants, validating Theorem 1, is included in the Appendix C.4.1.

7 Conclusion and Future Work

We proposed a novel neural diffusion framework based on opinion dynamics. Our method generalizes multiple opinion dynamic models, providing an efficient and expressive solution that advances existing GNNs. We provided a comprehensive theoretical analysis of our framework's convergence properties. The experiments on node classification and influence estimation tasks demonstrated the effectiveness and efficiency of our solution.

In our existing framework, the current feature retention parameter α is designed as a global hyperparameter. This design choice offers parameter efficiency and ensures stable convergence properties. In our future work, we plan to investigate learning this parameter in a node-specific manner from the underlying data distribution. Additionally, we aim to extend our evaluation to more challenging learning settings, such as transfer learning and out-of-distribution generalization.

References

- [1] Emmanuel Abbe. 2018. Community detection and stochastic block models: recent developments. *Journal of Machine Learning Research* 18, 177 (2018), 1–86.
- [2] Elisabetta Biondi, Chiara Boldrini, Andrea Passarella, and Marco Conti. 2023. Dynamics of opinion polarization. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 53, 9 (2023), 5381–5392.
- [3] Cristian Bodnar, Francesco Di Giovanni, Benjamin Chamberlain, Pietro Lio, and Michael Bronstein. 2022. Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in gnn. *Advances in Neural Information Processing Systems* 35 (2022), 18527–18541.
- [4] Cristian Bodnar, Fabrizio Frasca, Nina Otter, Yuguang Wang, Pietro Lio, Guido F Montufar, and Michael Bronstein. 2021. Weisfeiler and lehman go cellular: Cw networks. *Advances in neural information processing systems* 34 (2021), 2625–2640.
- [5] Aleksandar Bojchevski and Stephan Günnemann. 2018. Deep Gaussian Embedding of Graphs: Unsupervised Inductive Learning via Ranking. In *International Conference on Learning Representations*.
- [6] Xuheng Cai, Lianghao Xia, Xubin Ren, and Chao Huang. 2023. How expressive are graph neural networks in recommendation?. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 173–182.
- [7] Benjamin Chamberlain, James Rowbottom, Davide Eynard, Francesco Di Giovanni, Xiaowen Dong, and Michael Bronstein. 2021. Beltrami flow and neural diffusion on graphs. *Advances in Neural Information Processing Systems* 34 (2021), 1594–1609.
- [8] Ben Chamberlain, James Rowbottom, Maria I Goranova, Michael Bronstein, Stefan Webb, and Emanuele Rossi. 2021. Grand: Graph neural diffusion. In *International conference on machine learning*. PMLR, 1407–1418.
- [9] Jialong Chen, Bowen Deng, Chuan Chen, Zibin Zheng, et al. 2025. Graph Neural Ricci Flow: Evolving Feature from a Curvature Perspective. In *The Thirteenth International Conference on Learning Representations*.
- [10] Xi Chen, Xiao Zhang, Yong Xie, and Wei Li. 2017. Opinion Dynamics of Social-Similarity-Based Hegselmann-Krause Model. *Complexity* 2017, 1 (2017), 182057.
- [11] Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. 2021. Adaptive Universal Generalized PageRank Graph Neural Network. In *International Conference on Learning Representations*.
- [12] Jeongwhan Choi, Seoyoung Hong, Noseong Park, and Sung-Bae Cho. 2023. Gread: Graph neural reaction-diffusion networks. In *International Conference on Machine Learning*. PMLR, 5722–5747.
- [13] Abhimanyu Das, Sreenivas Gollapudi, and Kamesh Munagala. 2014. Modeling opinion dynamics in social networks. In *Proceedings of the 7th ACM international conference on Web search and data mining*. 403–412.
- [14] Morris H DeGroot. 1974. Reaching a consensus. *Journal of the American Statistical association* 69, 345 (1974), 118–121.
- [15] Mosha Eliasof, Eldad Haber, and Eran Treister. 2021. Pde-gcn: Novel architectures for graph neural networks motivated by partial differential equations. *Advances in neural information processing systems* 34 (2021), 3836–3849.
- [16] Mosha Eliasof, Eldad Haber, and Eran Treister. 2024. Feature transportation improves graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 11874–11882.
- [17] Noah E Friedkin and Eugene C Johnsen. 1990. Social influence and opinions. *Journal of mathematical sociology* 15, 3-4 (1990), 193–206.
- [18] Chen Gao, Xiang Wang, Xiangnan He, and Yong Li. 2022. Graph neural networks for recommender system. In *Proceedings of the fifteenth ACM international conference on web search and data mining*. 1623–1625.
- [19] Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Predict then Propagate: Graph Neural Networks meet Personalized PageRank. In *International Conference on Learning Representations*.
- [20] Johannes Gasteiger, Stefan Weissenberger, and Stephan Günnemann. 2019. Diffusion improves graph learning. *Advances in neural information processing systems* 32 (2019).
- [21] Jacob Goldenberg, Barak Libai, and Eitan Muller. 2001. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing letters* 12 (2001), 211–223.
- [22] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).
- [23] Andi Han, Dai Shi, Lequan Lin, and Junbin Gao. 2023. From Continuous Dynamics to Graph Neural Networks: Neural Diffusion and Beyond. *Transactions on Machine Learning Research* (2023).
- [24] Hossein Hassani, Roozbeh Razavi-Far, Mehrdad Saif, Francisco Chiclana, Ondrej Krejcar, and Enrique Herrera-Viedma. 2022. Classical dynamic consensus and opinion dynamics models: A survey of recent trends and methodologies. *Information Fusion* 88 (2022), 22–40.
- [25] Asela Hevapathige, Ahad N Zehmakan, and Qing Wang. 2025. Depth-Adaptive Graph Neural Networks via Learnable Bakry-Emery Curvature. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining* V. 2. 944–955.
- [26] Roger A Horn and Charles R Johnson. 2012. *Matrix analysis*. Cambridge university press.
- [27] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems* 33 (2020), 22118–22133.
- [28] Woochang Hyun, Insoo Lee, and Bongwon Suh. 2024. LEX-GNN: Label-Exploring Graph Neural Network for Accurate Fraud Detection. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 3802–3806.
- [29] Lokesh Jain, Rahul Katarya, and Shelly Sachdeva. 2023. Opinion leaders for information diffusion using graph neural network in online social networks. *ACM Transactions on the Web* 17, 2 (2023), 1–37.
- [30] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. 137–146.
- [31] Hwan Kim, Byung Suk Lee, Won-Yong Shin, and Sungsu Lim. 2022. Graph anomaly detection with graph neural networks: Current status and challenges. *IEEE Access* 10 (2022), 111820–111829.
- [32] Masahiro Kimura, Kazumi Saito, and Hiroshi Motoda. 2009. Efficient Estimation of Influence Functions for SIS Model on Social Networks.. In *IJCAI*. 2046–2051.
- [33] Diederik Kingma. 2015. Adam: a method for stochastic optimization. In *International Conference on Learning Representations*.
- [34] Thomas N Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.
- [35] Sanjay Kumar, Abhishek Mallik, Anavi Khetarpal, and Bhawani Sankar Panda. 2022. Influence maximization in social networks using graph embedding and graph neural network. *Information Sciences* 607 (2022), 1617–1636.
- [36] Jong-whi Lee and Jinhong Jung. 2023. Time-aware random walk diffusion to improve dynamic graph learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 8473–8481.
- [37] Naomi Ehrlich Leonard, Anastasia Bizyaeva, and Alessio Franci. 2024. Fast and flexible multiagent decision-making. *Annual Review of Control, Robotics, and Autonomous Systems* 7 (2024).
- [38] Guohao Li, Matthias Müller, Bernard Ghanem, and Vladlen Koltun. 2021. Training graph neural networks with 1000 layers. In *International conference on machine learning*. PMLR, 6437–6449.
- [39] Hao Li, Hao Jiang, Yuke Zheng, Hao Sun, and Wenying Gong. 2025. UniGO: A Unified Graph Neural Network for Modeling Opinion Dynamics on Graphs. In *Proceedings of the ACM on Web Conference 2025*. 530–540.

- [40] Hui Li, Susu Yang, Mengting Xu, Sourav S Bhowmick, and Jiangtao Cui. 2023. Influence maximization in social networks: A survey. *arXiv preprint arXiv:2309.04668* (2023).
- [41] Pengbo Li, Hang Yu, Xiangfeng Luo, and Jia Wu. 2023. LGM-GNN: A local and global aware memory-based graph neural network for fraud detection. *IEEE Transactions on Big Data* 9, 4 (2023), 1116–1127.
- [42] Xiao Li, Li Sun, Mengjie Ling, and Yan Peng. 2023. A survey of graph neural network based recommendation in social networks. *Neurocomputing* 549 (2023), 126441.
- [43] Yibo Li, Xiao Wang, Hongrui Liu, and Chuan Shi. 2024. A generalized neural diffusion framework on graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 8707–8715.
- [44] Chen Ling, Junji Jiang, Junxiang Wang, My T Thai, Renhao Xue, James Song, Meikang Qiu, and Liang Zhao. 2023. Deep graph representation learning and optimization for influence maximization. In *International conference on machine learning*. PMLR, 21350–21361.
- [45] Meng Liu, Hongyang Gao, and Shuiwang Ji. 2020. Towards deeper graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 338–348.
- [46] Xiaorui Liu, Jiayuan Ding, Wei Jin, Han Xu, Yao Ma, Zitao Liu, and Jiliang Tang. 2021. Graph neural networks with adaptive residual. *Advances in Neural Information Processing Systems* 34 (2021), 9720–9733.
- [47] Nuoyan Lyu, Bingbing Xu, Fangda Guo, and Huawei Shen. 2023. DCGNN: Dual-channel graph neural network for social bot detection. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 4155–4159.
- [48] Md Abdul Mannan, Md R Rahman, Halima Akter, Nazmun Nahar, and Samiran Mondal. 2021. A study of Banach fixed point theorem and its applications. *American Journal of Computational Mathematics* 11, 2 (2021), 157–174.
- [49] George Panagopoulos, Nikolaos Tziortziotis, Michalis Vaziriannis, and Fragkiskos Malliaros. 2023. Maximizing influence with graph neural networks. In *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining*. 237–244.
- [50] Hongbin Pei, Bingzhe Wei, Kevin Chen Chuan Chang, Yu Lei, and Bo Yang. 2020. GEOM-GCN: GEOMETRIC GRAPH CONVOLUTIONAL NETWORKS. In *8th International Conference on Learning Representations, ICLR*.
- [51] Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. 2023. A critical look at the evaluation of GNNs under heterophily: Are we really making progress?. In *The Eleventh International Conference on Learning Representations*.
- [52] Hegselmann Rainer and Ulrich Krause. 2002. Opinion dynamics and bounded confidence: models, analysis and simulation. (2002).
- [53] Herbert Robbins and Sutton Monro. 1951. A stochastic approximation method. *The annals of mathematical statistics* (1951), 400–407.
- [54] Emanuele Rossi, Bertrand Charpentier, Francesco Di Giovanni, Fabrizio Frasca, Stephan Günnemann, and Michael M Bronstein. 2024. Edge directionality improves learning on heterophilic graphs. In *Learning on Graphs Conference*. PMLR, 25–1.
- [55] Ryan Rossi and Nesreen Ahmed. 2015. The network data repository with interactive graph analytics and visualization. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 29.
- [56] T Konstantin Rusch, Ben Chamberlain, James Rowbottom, Siddhartha Mishra, and Michael Bronstein. 2022. Graph-coupled oscillator networks. In *International Conference on Machine Learning*. PMLR, 18888–18909.
- [57] Matthew Thorpe, Tan Minh Nguyen, Hedi Xia, Thomas Strohmer, Andrea Bertozi, Stanley Osher, and Bao Wang. 2022. GRAND++: Graph Neural Diffusion with A Source Term. In *International Conference on Learning Representations*.
- [58] Víctor A Vargas-Pérez, Jesús Giráldez-Cru, Pablo Mesejo, and Oscar Cordón. 2024. Unveiling Agents' Confidence in Opinion Dynamics Models via Graph Neural Networks. *IEEE Transactions on Computational Social Systems* (2024).
- [59] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.
- [60] Kegin Wang, Yulong Yang, Ishan Saha, and Christine Allen-Blanchette. 2025. Resolving Oversmoothing with Opinion Dissensus. *arXiv preprint arXiv:2501.19089* (2025).
- [61] Yuelin Wang, Kai Yi, Xinliang Liu, Yu Guang Wang, and Shi Jin. 2022. ACMP: Allen-cahn message passing with attractive and repulsive forces for graph neural networks. In *The Eleventh International Conference on Learning Representations*.
- [62] Oliver Wieder, Stefan Kohlbacher, Mélanie Kuennemann, Arthur Garon, Pierre Ducrot, Thomas Seidel, and Thierry Langer. 2020. A compact review of molecular property prediction with graph neural networks. *Drug Discovery Today: Technologies* 37 (2020), 1–12.
- [63] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. 2022. Graph neural networks in recommender systems: a survey. *Comput. Surveys* 55, 5 (2022), 1–37.
- [64] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems* 32, 1 (2020), 4–24.
- [65] Zhenxing Wu, Jike Wang, Hongyan Du, Dejun Jiang, Yu Kang, Dan Li, Peichen Pan, Yafeng Deng, Dongsheng Cao, Chang-Yu Hsieh, et al. 2023. Chemistry-intuitive explanation of graph neural networks for molecular property prediction with substructure masking. *Nature communications* 14, 1 (2023), 2585.
- [66] Wenwen Xia, Yuchen Li, Jun Wu, and Shenghong Li. 2021. Deepis: Susceptibility estimation on social networks. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 761–769.
- [67] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. 2018. Representation learning on graphs with jumping knowledge networks. In *International conference on machine learning*. PMLR, 5453–5462.
- [68] Wanyue Xu, Liwang Zhu, Jiale Guan, Zuobai Zhang, and Zhongzhi Zhang. 2022. Effects of stubbornness on opinion dynamics. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*. 2321–2330.
- [69] Lei Zhang, Xiaodong Yan, Jianshan He, Ruopeng Li, and Wei Chu. 2023. Drgcn: Dynamic evolving initial residual for deep graph convolutional networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 37. 11254–11261.
- [70] Jialin Zhao, Yuxiao Dong, Ming Ding, Evgeny Kharlamov, and Jie Tang. 2021. Adaptive diffusion in graph neural networks. *Advances in neural information processing systems* 34 (2021), 23321–23333.
- [71] Kai Zhao, Qiyu Kang, Yang Song, Rui She, Sijie Wang, and Wee Peng Tay. 2023. Graph neural convection-diffusion with heterophily. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*. 4656–4664.
- [72] Bingxin Zhou, Outongyi Lv, Jing Wang, Xiang Xiao, and Weishu Zhao. 2024. ODNet: Opinion Dynamics-Inspired Neural Message Passing for Graphs and Hypergraphs. *Transactions on Machine Learning Research* (2024).

Appendix

A Proofs

In this section, we present the proofs for the theorems discussed in the main content.

THEOREM 1 (CONVERGENCE OF GODNF FEATURE UPDATES). *Let $X(t) \in \mathbb{R}^{n \times d}$ evolve according to the update rule in Eq. (1), and define the combined influence matrix $M(t) = (I - \Lambda)(W(t) - \mu L_g)$. When the parameters Λ and μ are chosen such that the operator norm satisfies $\|M(t)\|_{\text{op}} < 1$ for all t , and $W(t) \rightarrow W^*$, the $X(t)$ is guaranteed to converge to a unique fixed point X^* as $t \rightarrow \infty$.*

PROOF. We work in the Banach space $(\mathbb{R}^{n \times d}, \|\cdot\|)$. $W(t) \rightarrow W^*$, which implies $M(t) \rightarrow M^*$, where M^* is a fixed matrix. Since $\|M(t)\|_{\text{op}} < 1$ for all t , we have $\|M^*\|_{\text{op}} < 1$. Let X^* satisfy:

$$X^* = \alpha X^* + (1 - \alpha)\Lambda X(0) + (1 - \alpha)M^* X^*$$

Define $A(t) = \alpha I + (1 - \alpha)M(t)$ and $A^* = \alpha I + (1 - \alpha)M^*$.

For any t , we have:

$$\|A(t)\|_{\text{op}} \leq \alpha + (1 - \alpha)\|M(t)\|_{\text{op}} < \alpha + (1 - \alpha) = 1$$

Let $\beta = \sup_t \|A(t)\|_{\text{op}} < 1$. For the error $E(t) = X(t) - X^*$, we have:

$$E(t+1) = A(t)X(t) - A^*X^* + (1 - \alpha)\Lambda X(0) - (1 - \alpha)M^* X^*$$

$$E(t+1) = A(t)E(t) + (A(t) - A^*)X^*.$$

Taking norms:

$$\|E(t+1)\| \leq \|A(t)\|_{\text{op}}\|E(t)\| + \|A(t) - A^*\|_{\text{op}}\|X^*\|$$

$$\|E(t+1)\| \leq \beta\|E(t)\| + (1 - \alpha)\|M(t) - M^*\|_{\text{op}}\|X^*\|$$

Since $M(t) \rightarrow M^*$, for any $\tau > 0$, there exists T such that for all $t \geq T$, $\|M(t) - M^*\|_{\text{op}} < \frac{\tau}{(1-\alpha)\|X^*\|}$ (assuming $X^* \neq 0$).

For $t \geq T$:

$$\|E(t+1)\| \leq \beta\|E(t)\| + \tau$$

This is a contraction mapping with a small perturbation, as $t \rightarrow \infty$ and $\tau \rightarrow 0$. Thus, according to the Banach fixed point theorem [48], $X(t)$ converges to X^* . \square

THEOREM 2 (SINGLE CONSENSUS UNDER FULLY DIFFUSIVE DYNAMICS). *GODNF reaches a single consensus if $\Lambda = 0$, $\mu = 0$, $\alpha = 0$, and W^* is a row-stochastic matrix.*

PROOF. With these parameter configurations, the update rule of GODNF becomes $X(t+1) = W^*X(t)$. Given that W^* is a row-stochastic matrix, this would be equivalent to the FD model. As proven in DeGroot et al. [14], FD model converges to single consensus. Therefore, GODNF will exhibit the same behavior. \square

THEOREM 3 (MULTI-CONSENSUS UNDER BLOCK-STRUCTURED DYNAMICS). *GODNF reaches multi consensus if the matrix $M^* = (1 - \Lambda)(W^* - \mu L_g)$ has a block diagonal structure corresponding to multiple disconnected components, with initial conditions and parameters differ for at least two components.*

PROOF. The fixed point of GODNF can be written as,

$X^* = (I - M^*)^{-1}\Lambda X(0)$. Since $\|M(t)\|_{\text{op}} < 1$ for all t , $\|M^*\|_{\text{op}} < 1$ and $(I - M^*)$ invertible. Therefore, the existence of the above fixed point is guaranteed. Let $B^* = (I - M^*)^{-1}$, and $X^* = B^*\Lambda X(0)$. Given M^* has block structure, we can write B^* as follows:

$$B^* = \begin{bmatrix} B_1 & 0 & \cdots & 0 \\ 0 & B_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & B_k \end{bmatrix}$$

This leads to set of independent components $X_1^* = B_1^*\Lambda_1 X_1(0)$, $X_2^* = B_2^*\Lambda_2 X_2(0)$, ..., $X_k^* = B_k^*\Lambda_k X_k(0)$. Each component reaches its own consensus independently, leading to multiple consensuses when initial conditions and parameters differ for at least two components. \square

THEOREM 4 (INDIVIDUALIZED CONSENSUS UNDER STRONG FEATURE RETENTION). *GODNF reaches individualized consensus if the initial features $x_i(0)$ are distinct for all nodes $i \in V$, and $\lambda_i \in (0, 1]$ are sufficiently large (i.e., close to 1).*

PROOF. The fixed point of GODNF is $X^* = (I - M^*)^{-1}\Lambda X(0)$ where $M^* = (1 - \Lambda)(W^* - \mu L_g)$. Since $\|M(t)\|_{\text{op}} < 1$ for all t , $\|M^*\|_{\text{op}} < 1$ and $(I - M^*)$ invertible. This ensures the existence of the above fixed point. When λ_i approaches 1 for all nodes, $M \approx 0$, giving:

$$X^* \approx (I - 0)^{-1}\Lambda X(0) = \Lambda X(0)$$

Since Λ has positive diagonal entries close to 1, each node's convergence value closely preserves its initial feature. With $X(0)$ containing distinct initial features, this naturally leads to individualized consensus. \square

B Mapping of GODNF Components to Opinion Dynamic Models

Model	Current Feature Retention	Initial Feature Attachment	Dynamic Neighborhood Influence	Structural Regularization
FD	✗	✗	✗	✗
FJ	✗	✓	✗	✗
HK	✓	✗	✓	✗
GODNF _{Static}	✓	✓	✗	✓
GODNF _{Dynamic}	✓	✓	✓	✓

Table 4: Comparison of Opinion Dynamics Models and GODNF Components. Note that for FD and FJ models, we consider formulations without self-loops, hence no current feature retention. While these models incorporate neighbourhood influence, they utilise static (time-invariant) connection weights and topologies, thus marked as lacking *Dynamic Neighbourhood Influence*. Further, we consider the formulation where HK includes agent's current opinion in bounded confidence averaging.

In Table 4, we demonstrate the versatility of GODNF by showing how it generalizes various opinion dynamics models, aligning its components with the mechanisms of these models.

Our neural framework offers a unified approach to opinion dynamics. It encapsulates the diverse and nuanced diffusion dynamics inherent in these distinct models, enhancing expressiveness.

C Supplementary Experimental Details

C.1 Model Hyperparameters

In our experiments, we employ a Grid search to find optimal hyperparameters of GODNF in the following ranges: number of layers $\in \{2, 3, 4\}$, learning rate $\in \{5e-3, 1e-2\}$, weight decay $\in \{5e-4, 1e-1\}$, dropout $\in \{0.4, 0.5\}$, and the hidden dimension size $\in \{32, 64, 256\}$. GODNF model hyperparameter, α is chosen from $\{0.1, 0.3, 0.6, 0.9\}$. Note that Λ , and μ parameters in our model are learned from the ground truth. In node classification, we train models for up to 1000 epochs, while for influence estimation, we use 200 epochs. We utilize the Adam algorithm [33] as the optimizer.

C.2 Implementation Details

All experiments were conducted on a Linux server with an Intel Xeon W-2175 2.50 GHz processor, comprising 28 cores, an NVIDIA RTX A6000 GPU, and 512 GB of RAM. We use the following Python libraries for our implementation: PyTorch version 2.3.1, torchvision 0.18.1, torchaudio 2.3.1, torch-geometric 2.7.0, torch-cluster 1.6.3, torch-scatter 2.0.9, and torch_sparse 0.6.18.

C.3 Benchmark Dataset Statistics

Dataset statistics for node classification and node regression tasks are depicted in table 5 and table 6, respectively.

Type	Dataset	Homophily Level	# Nodes	# Edges	# Classes
Heterophily	Texas	0.06	183	309	5
Heterophily	Cornell	0.12	183	295	5
Heterophily	Wisconsin	0.18	251	499	5
Heterophily	Film	0.22	7,600	33,544	5
Heterophily	Amazon-rating	0.38	24,492	186,100	5
Homophily	Cora Full	0.57	19,793	126,842	70
Homophily	Citeseer	0.74	3,312	4,732	6
Homophily	Cora-ML	0.79	2,995	16,316	7
Homophily	PubMed	0.80	19,717	44,338	3
Homophily	DBLP	0.83	17,716	105,734	4

Table 5: Dataset statistics for node classification

Dataset	# Nodes	# Edges
Jazz	198	2,742
Cora-ML	2,810	7,981
Network Science	1,565	13,532
Power Grid	4,941	6,594

Table 6: Dataset statistics for influence estimation task

C.4 Additional Experiments

C.4.1 Convergence and Visualization Analysis. We conduct experiments to empirically validate the convergence properties of GODNF, performing node classification tasks on the Cora Full dataset. Figure 8 shows the convergence behaviour of GODNF variants through step-wise average embedding changes over evolution time. The exponential decay pattern demonstrates that both GODNF variants achieve stable convergence, with average embedding changes decreasing rapidly from initial large adjustments to minimal fluctuations by step 7-8, confirming the theoretical convergence guarantees presented in Theorem 1.

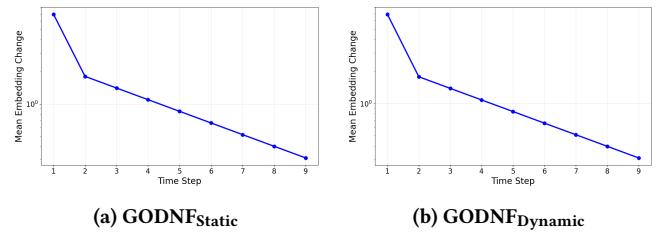


Figure 8: Step-wise embedding changes of GODNF over time steps for the Cora Full for node classification.

Next, we provide a visualisation of embedding convergence. Figure 9 shows the embedding evolution of GODNF variants in 2D principal component analysis (PCA) space across time steps. The embedding evolution shows that GODNF successfully learns better node representations over time. Initially, nodes are scattered randomly in the embedding space, but as iterations progress, nodes of the same label gradually move closer together to form distinct clusters. The decreasing embedding norm values (i.e., $\|h\|_2$) indicate that embeddings become more stable. At the same time, the improved visual separation between different colored clusters demonstrates that the model learns to distinguish between different node classes effectively. The static variant of GODNF exhibits a slightly smoother progression of embeddings and more consistent cluster boundaries. In contrast, the dynamic variant adjusts weights for each time step, resulting in slightly higher initial variability, but it achieves comparable final convergence quality.

C.4.2 Robustness to adversarial attacks. Figure 10 compares the node classification performance of GODNF against the baselines of the Film dataset under various adversarial attacks. This is in addition to the Citeseer dataset provided in the main content. Consistent with the Citeseer results, GODNF variants also outperform all baselines on the Film dataset under various adversarial attack scenarios.

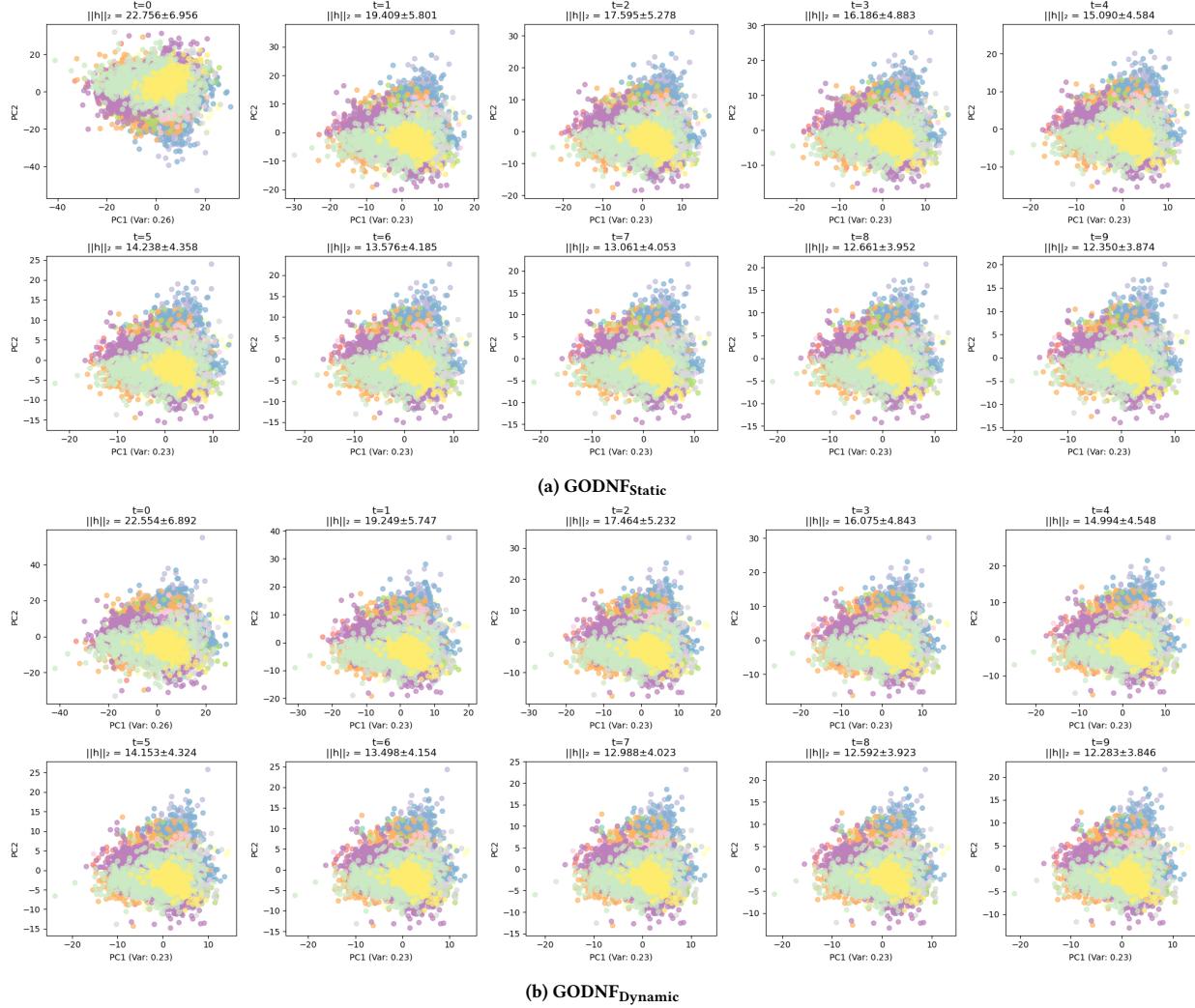


Figure 9: Embedding evolution for Cora Full dataset over multiple time steps. $\|h\|_2$ represents the average L2 norm across all individual node embeddings, indicating the mean embedding magnitude. PCA (Var) shows the percentage of original high-dimensional embedding variance captured in the 2D visualization.

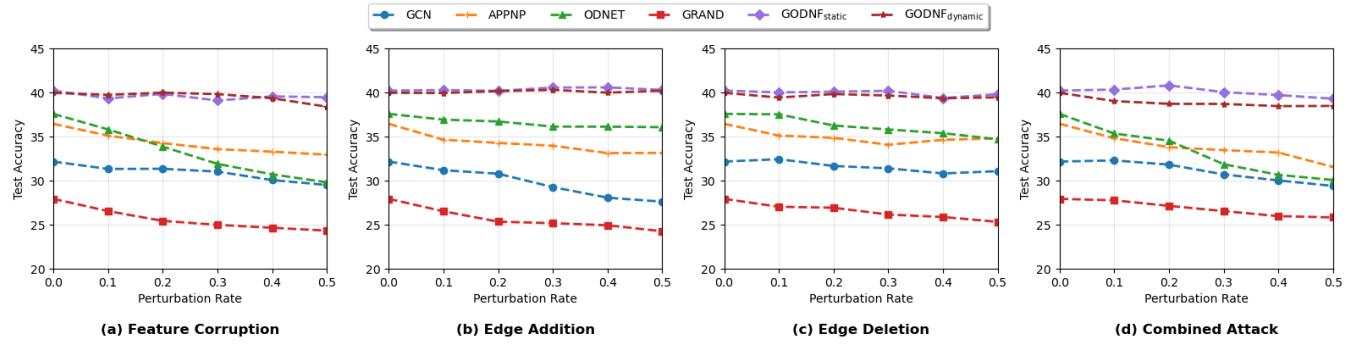


Figure 10: Node classification performance for Film dataset under different adversarial attacks.