Enhancing Graph Representations with Neighborhood-Contextualized Message-Passing

Anonymous authors
Paper under double-blind review

Abstract

Graph neural networks (GNNs) have become an indispensable tool for analyzing relational data. Classical GNNs are broadly classified into three variants: convolutional, attentional, and message-passing. While the standard message-passing variant is expressive, its typical *pair-wise* messages only consider the features of the center node and each neighboring node *individually*. This design fails to incorporate contextual information contained within the *broader* local neighborhood, potentially hindering its ability to learn meaningful relationships within the *entire* set of neighboring nodes. To address this limitation, this work first formalizes the concept of neighborhood-contextualization, rooted in a key property of the attentional variant. This then serves as the foundation for generalizing the message-passing variant to the proposed neighborhood-contextualized message-passing (NCMP) framework. To demonstrate its utility, a simple, practical, and efficient method to parametrize and operationalize NCMP is presented, leading to the development of the proposed Soft-Isomorphic Neighborhood-Contextualized Graph Convolution Network (SINC-GCN). Across a diverse set of synthetic and benchmark datasets, SINC-GCN strikes an optimal balance between expressivity and efficiency. Notably, while more complex models incur significant computational overhead, SINC-GCN delivers substantial, statistically significant performance gains over baseline GNN models in graph property prediction tasks while maintaining the highly efficient asymptotic complexity, further underscoring the distinctive utility of neighborhood-contextualization. Overall, the paper lays the foundation for the NCMP framework as a practical path toward enhancing the graph representational power of classical GNNs.

1 Introduction

In the modern age of big data, graphs have become an indispensable tool for modeling complex relationships. Many real-world systems may be naturally represented as graphs, where nodes represent entities and edges represent interactions. For instance, financial systems may be viewed as graphs of users connected via transactions; social networking sites may correspond to graphs of people connected through friendships; and molecules may be represented as graphs of atoms connected by chemical bonds. Furthermore, centuries of research in the field of graph theory have provided a rich set of mathematical tools to study and analyze these structures.

With the growing interest in the field of machine learning from both academia and industry, graph neural networks (GNNs) have emerged as a special subclass of deep learning architectures specifically designed to process graph-structured data. In contrast to traditional architectures, GNNs consider both the graph structure via edge connections and the information contained within the nodes, making them well-suited for various graph tasks. For example, they may be used for node property prediction (*e.g.*, detecting fraudulent users in financial systems), edge prediction (*e.g.*, suggesting friends in social networking sites), and graph property prediction (*e.g.*, predicting chemical properties of molecules).

In the literature, one-hop localized GNN architectures, which are the primary focus of this paper, may be broadly classified into three variants or *flavors*: convolutional, attentional, and message-passing. Foundational works in the field, rooted in spectral graph theory, mainly fall under the convolutional variant,

whereby each node aggregates information or messages from its neighboring nodes by simply considering each neighborhood feature individually. With the introduction of the Transformer, various works have adopted the attention mechanism into GNNs, whereby each node aggregates messages from its neighboring nodes, similarly considering each neighborhood feature individually, with a dynamic weighting scheme based on their relative importance. More recently, with the developments in hardware, many works have studied message-passing variants to push the limits of GNNs, whereby each node aggregates messages from its neighboring nodes by considering both its own features and the features of each neighbor. Within this paradigm, researchers agree that the attentional variant is more expressive than the convolutional variant in terms of graph representational power, as the latter may be expressed as a particular instance of the former. Moreover, the message-passing variant is largely agreed to be the most expressive GNN variant, as it can be thought of as a generalization of the other two variants.

Despite the success and wide adoption of the classic message-passing variant, it has a key architectural limitation: the *pair-wise* messages are traditionally calculated using only the features of the center node and each *individual* neighboring node. Crucially, this design overlooks the rich contextual information embedded in the *broader* context of the local neighborhood, specifically with the relationships among the *entire* set of neighboring nodes. In line with this key insight, this work:

1. Formalizes the concept of **neighborhood-contextualization**, rooted in an implicit yet crucial property of the attentional variant;

2. Proposes the **neighborhood-contextualized message-passing (NCMP) framework** as a novel generalization of the message-passing variant, featuring both contextualized messages, as defined in Lim et al. (2025a), and neighborhood-contextualization; and

3. Presents a theoretical discussion on one simple, practical, and efficient method for its parametrization and operationalization, leading to the development of the **Soft-Isomorphic Neighborhood-Contextualized Graph Convolution Network (SINC-GCN)**.

Through extensive evaluation in both synthetic and benchmark datasets across node and graph property prediction tasks, SINC-GCN strikes a balance between performance and efficiency, achieving consistent and statistically significant gains against baseline GNN models. Overall, the NCMP framework and its SINC-GCN instantiation offer a scalable, theoretically-grounded, and practical path toward enhancing the representational capability of classical GNNs by integrating local structural semantics in a computationally efficient manner.

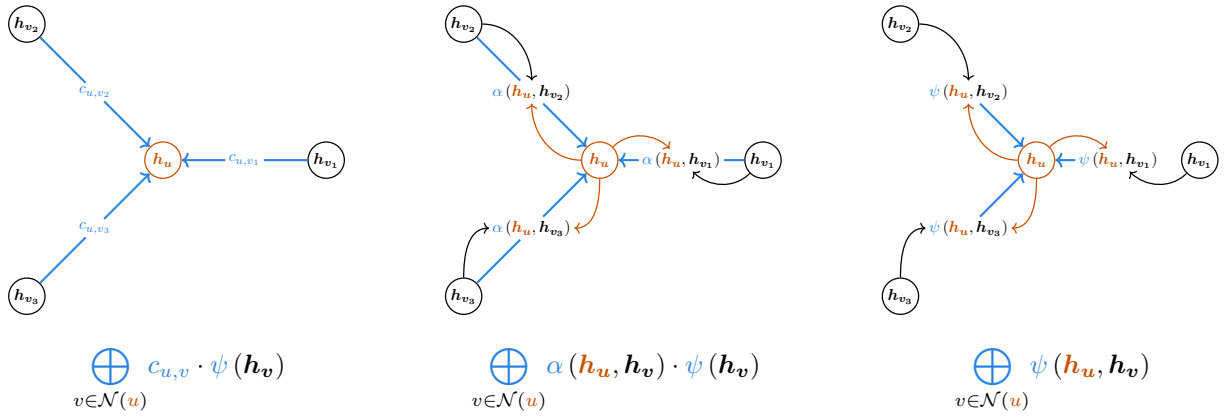
The paper is organized as follows. Section 2 first presents an overview of the classical GNNs. Section 3 subsequently motivates the proposed NCMP framework and presents a theoretical discussion for developing the simple SINC-GCN instance. Section 4 then highlights their practical utility and expressivity with experiments on synthetic and benchmark datasets. Section 5 finally concludes with a summary of the contributions and recommendations for future work.

2 Graph Neural Networks

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph, with $\mathcal{N}(u) \subseteq \mathcal{V}$ denoting the set of nodes adjacent to node $u \in \mathcal{V}$ and $\boldsymbol{h_u}$ denoting the features of node u. In the literature, the development of classical one-hop localized GNNs generally follows the chronology of convolutional, attentional, and message-passing variants.

2.1 Convolutional Variant

Early works in graph machine learning attempted to define the convolution operation on graphs by building upon spectral graph theory, often using a graph Fourier transform on the graph Laplacian. However, the computational complexity of calculating the full spectrum led to the development of more efficient polynomial approximations. Among these works was the Graph Convolution Network (GCN), introduced as a learnable,



(a) Convolutional. (b) Attentional. (c) Message-Passing.

Figure 1: Graph Neural Network Architecture Variants.

first-order approximation of the graph convolution localized to the one-hop neighborhood, defined as

$$\boldsymbol{h}_{\boldsymbol{u}}^{*} = \sum_{v \in \mathcal{N}(u)} \frac{1}{\sqrt{|\mathcal{N}(u)|}\sqrt{|\mathcal{N}(v)|}} \boldsymbol{W} \boldsymbol{h}_{\boldsymbol{v}}, \tag{1}$$

where $\boldsymbol{W}$ is a learnable linear transformation and $\boldsymbol{h}_{\boldsymbol{u}}^{*}$ is the updated features for node u after the convolution operation (Kipf & Welling, 2017). GCN was shown to outperform existing methods in transductive semi-supervised tasks. Contemporaneously, the Graph Sample and Aggregate (GraphSAGE) was also introduced for inductive representation learning, defined as

$$\boldsymbol{h}_{\boldsymbol{u}}^{*} = \max_{v \in \mathcal{N}(u)} \boldsymbol{W} \boldsymbol{h}_{\boldsymbol{v}} + \boldsymbol{b}, \tag{2}$$

where $\boldsymbol{W}$ and $\boldsymbol{b}$ are a learnable linear transformation and bias term, respectively. GraphSAGE demonstrated strong performance on tasks requiring generalization to new and unseen graphs during evaluation. More recently, the Graph Isomorphism Network (GIN) was introduced as a maximally expressive GNN architecture for detecting graph isomorphism, rooted in the Weisfeiler-Lehman (WL) test (Weisfeiler & Leman, 1968), defined as

$$\boldsymbol{h}_{\boldsymbol{u}}^{*} = \mathrm{MLP}\left((1+\varepsilon) \cdot \boldsymbol{h}_{\boldsymbol{u}} + \sum_{v \in \mathcal{N}(u)} \boldsymbol{h}_{\boldsymbol{v}}\right), \tag{3}$$

where MLP is a learnable multi-layer perceptron (MLP) and ε is a learnable scalar parameter (Xu et al., 2019). GIN was shown to outperform other models in tasks where determining graph isomorphism becomes critical (Sato et al., 2021). Due to their simplicity and computational efficiency, GCN, GraphSAGE, and GIN became widely adopted across various applications (Li et al., 2020; Liu et al., 2022; Kim & Ye, 2020; Dwivedi et al., 2023; Hu et al., 2020). Notably, they may be classified as convolutional variants of GNN, as shown in Fig. 1(a), which may be expressed as

$$\bigoplus_{v \in \mathcal{N}(u)} c_{u,v} \cdot \psi\left(\boldsymbol{h}_{\boldsymbol{v}}\right), \tag{4}$$

for some neighborhood aggregator $\bigoplus$ (*e.g.*, sum, mean, symmetric mean, and max). With this variant, messages from a neighboring node $v \in \mathcal{N}(u)$ to node u are simply a function of the features of the neighboring node $\psi\left(\boldsymbol{h}_{\boldsymbol{v}}\right)$ multiplied by a scalar factor $c_{u,v}$ based on the local graph structure.

2.2 Attentional Variant

Following the introduction of the Transformer, researchers considered incorporating the attention mechanism into the graph convolution operation to boost model performance. One of the earliest works was the Graph

Attention Network (GAT), defined as

$$\boldsymbol{h}_{\boldsymbol{u}}^{*} = \sum_{v \in \mathcal{N}(u)} \alpha_{u,v} \cdot \boldsymbol{W} \boldsymbol{h}_{\boldsymbol{v}}, \tag{5}$$

$$\alpha_{u,v} = \mathrm{Softmax}\left(e_{u,v}\right), \tag{6}$$

$$e_{u,v} = \boldsymbol{a}^{\top} \ \textsc{LeakyReLU}\left(\boldsymbol{W}_{\boldsymbol{Q}} \boldsymbol{h}_{\boldsymbol{u}} + \boldsymbol{W}_{\boldsymbol{K}} \boldsymbol{h}_{\boldsymbol{v}}\right), \tag{7}$$

where $\boldsymbol{W}$, $\boldsymbol{a}$, $\boldsymbol{W}_{\boldsymbol{Q}}$, $\boldsymbol{W}_{\boldsymbol{K}}$ are learnable linear transformations (Veličković et al., 2018). Other works, such as GATv2 (Brody et al., 2022), build upon GAT by proposing different methods for computing the attention scores $e_{u,v}$ for various applications (Wang et al., 2021; Hsu et al., 2023; Jiang et al., 2023). These GNN architectures may then be aptly classified as attentional GNN variants, as shown in Fig. 1(b), conventionally expressed as

$$\bigoplus_{v \in \mathcal{N}(u)} \alpha\left(\boldsymbol{h}_{\boldsymbol{u}}, \boldsymbol{h}_{\boldsymbol{v}}\right) \cdot \psi\left(\boldsymbol{h}_{\boldsymbol{v}}\right). \tag{8}$$

In this variant, messages from a neighboring node $v \in \mathcal{N}(u)$ to node u are still a function of the features of the neighboring node $\psi\left(\boldsymbol{h}_{\boldsymbol{v}}\right)$. However, the scalar factor now becomes a function of both node features $\alpha\left(\boldsymbol{h}_{\boldsymbol{u}}, \boldsymbol{h}_{\boldsymbol{v}}\right)$, allowing it to dynamically adjust the contribution of each message based on the relative importance of the neighboring node.

2.3 Message-Passing Variant

The message-passing variant provides a more general framework for the graph convolution operation in GNNs, leading to many architectures tailored for specific applications (Lim et al., 2025b;c). A prominent example is the Message-Passing Neural Network (MPNN), which was shown to perform well in approximating quantum mechanical simulations, even achieving orders of magnitude decrease in computational time (Gilmer et al., 2017). More recently, the Soft-Isomorphic Relational Graph Convolution Network (SIR-GCN) was introduced as a simple and computationally efficient architecture with maximal graph representational power, defined as

$$\boldsymbol{h}_{\boldsymbol{u}}^{*} = \sum_{v \in \mathcal{N}(u)} \boldsymbol{W}_{\boldsymbol{R}} \, \sigma\left(\boldsymbol{W}_{\boldsymbol{Q}} \boldsymbol{h}_{\boldsymbol{u}} + \boldsymbol{W}_{\boldsymbol{K}} \boldsymbol{h}_{\boldsymbol{v}}\right), \tag{9}$$

where σ is a non-linear activation function and $\boldsymbol{W}_{\boldsymbol{R}}$, $\boldsymbol{W}_{\boldsymbol{Q}}$, $\boldsymbol{W}_{\boldsymbol{K}}$ are learnable linear transformations (Lim et al., 2025a). Owing to its message-passing *flavor*, SIR-GCN was even shown to mathematically generalize GCN, GraphSAGE, GIN, and GAT, among others. GNN architectures following these designs may be classified as message-passing variants, as shown in Fig. 1(c), expressed as

$$\bigoplus_{v \in \mathcal{N}(u)} \psi\left(\boldsymbol{h}_{\boldsymbol{u}}, \boldsymbol{h}_{\boldsymbol{v}}\right). \tag{10}$$

Crucially, messages from a neighboring node $v \in \mathcal{N}(u)$ to node u in this variant are now a (potentially non-linear) function of both node features $\psi\left(\boldsymbol{h}_{\boldsymbol{u}}, \boldsymbol{h}_{\boldsymbol{v}}\right)$. This design makes it highly expressive, as it may be able to learn complex relationships between neighboring nodes beyond simple scalars.

2.4 Beyond One-Hop Localization

One-hop localized GNN architectures are widely adopted in both literature and practice. Nevertheless, several studies attempt to increase the representational power of GNNs by considering various extensions. For instance, one line of work investigates using multiple aggregators, scalers, and basis weights within the standard graph convolution operation, such as with the Principal Neighborhood Aggregation (PNA) (Corso et al., 2020), Efficient Graph Convolution - Single (EGC-S) (Tailor et al., 2022), and Efficient Graph Convolution - Multiple (EGC-M) (Tailor et al., 2022). Going beyond these additional tricks, another line of work considers higher-order neighborhoods, such as with the k-dimensional GNNs (k-GNNs) (Morris et al., 2019), Folklore Graph Neural Networks (FGNN) (Azizian & Lelarge, 2021), and Cellular Weisfeiler-Lehman Networks (CWNs) (Bodnar et al., 2021), to capture higher-order topological properties in the

localized subgraphs to go beyond the 1-WL test in terms of graph isomorphism representational power. More recently, some studies also look into Graph Transformers incorporating various graph encodings into the standard Transformer model (Ying et al., 2021), graph diffusion networks capturing multi-hop neighborhood information (Sun et al., 2022), and subgraph isomorphism counting considering topologically-aware message-passing (Bouritsas et al., 2023), among others (Mao et al., 2024; Sun et al., 2024; Lachi et al., 2024).

While these advancements generally outperform the classical GNNs, their performance improvements often come with greater computational cost, making them infeasible for large graphs. Hence, this work primarily focuses on one-hop localized GNNs due to their simplicity and computational efficiency. Specifically, it examines how the current design of the message-passing variant may be further improved to create more powerful GNN architectures.

3 A Framework for Neighborhood-Contextualized Message-Passing

To motivate the development of a new GNN framework, Table 1 first compares the three existing variants. In particular, previous work has shown how contextualized messages—messages that are sufficiently expressive functions, *i.e.*, universal function approximators, of the features of both the center node $\boldsymbol{h_u}$ and neighboring node $\boldsymbol{h_v}$—are crucial in boosting graph representational power (Lim et al., 2025a). Notably, both the convolutional and attentional variants do not possess this property as their core message ψ solely considers the features of the neighboring node $\boldsymbol{h_v}$. Meanwhile, the message-passing variant *may* possess this property provided the message ψ has universal function approximation capabilities.

Table 1: Comparison of Graph Neural Network Variants.

GNN Variant	Contextualized Messages	Neighborhood-Contextualized
Convolutional	✗	✗
Attentional	✗	✓
Message-Passing	✓	✗

In addition to this dimension, this work also highlights an implicit yet notable property of the attentional variant. Crucially, while it is typically expressed as Eq. (8), it is mathematically more accurate to express it as

$$\bigoplus_{v \in \mathcal{N}(u)} \alpha\left(\boldsymbol{h_u}, \boldsymbol{h_v}, \{\boldsymbol{h_w} : w \in \mathcal{N}(u)\}\right) \cdot \psi\left(\boldsymbol{h_v}\right), \tag{11}$$

to explicitly capture the dependency of the scalar attention weight α on the entire set of neighborhood features for normalization. Rooted in this key insight, this work first formalizes the concept of ***neighborhood-contextualization*** in GNNs as the functional dependence of the convolution operation on the *entire* set of neighborhood features $\{\boldsymbol{h_w} : w \in \mathcal{N}(u)\}$ as additional context of the *broader* local neighborhood of the center node u.

Interestingly, only the attentional variant implicitly possesses neighborhood-contextualization. However, this simply serves as a scalar softmax normalization factor for the attention weights, hindering its ability to learn meaningful relationships among the one-hop neighboring nodes, as noted in Lim et al. (2025a). Meanwhile, both the convolutional and message-passing variants remain indifferent or agnostic to the broader context of the local neighborhood. Critically, the key architectural limitation of the message-passing variant lies with its *pair-wise* messages $\psi\left(\boldsymbol{h_u}, \boldsymbol{h_v}\right)$ only considering the features of the center node $\boldsymbol{h_u}$ and each neighboring node $\boldsymbol{h_v}$ for $v \in \mathcal{N}(u)$ *individually*. This design makes ψ neighborhood-agnostic, limiting its ability to perform more complex reasoning on the relationship among the entire set of one-hop neighboring nodes $\mathcal{N}(u)$ and potentially limiting its expressivity.

To address the limitations of both the message-passing and attentional variants, this work integrates both contextualized messages and neighborhood-contextualization within GNNs to propose the ***neighborhood-***

contextualized message-passing (NCMP) framework, as shown in Fig. 2, expressed as

$$\bigoplus_{v \in \mathcal{N}(u)} \psi\left(\boldsymbol{h_u}, \boldsymbol{h_v}, \{\boldsymbol{h_w} : w \in \mathcal{N}(u)\}\right). \tag{12}$$

Notably, unlike the attentional variant, where the neighborhood-contextualization is solely in the scalar α, NCMP extends this to the *multi-dimensional* ψ, adapting the *vector* of messages themselves based on the *entire* set of one-hop neighborhood features $\{\boldsymbol{h_w} : w \in \mathcal{N}(u)\}$ thereby equipping it with the ability to *learn more meaningful relationships within the local neighborhood*, which was not previously possible with existing GNN variants. Intuitively, rather than asking "given the context of my neighbors, *how much information* I should send?" as with the attentional variant, the proposed framework asks "given the context of my neighbors, *what is the appropriate information* I should send?". Furthermore, it is also easy to see that NCMP generalizes the message-passing variant. Hence, the proposed framework is functionally more expressive than classical neighborhood-agnostic message-passing GNNs, as the former encapsulates a broader class of graph-based models. Specifically, rather than restricting to rigid, localized pairs, it allows for feature transformations to adapt based on the broader structural and semantic context.

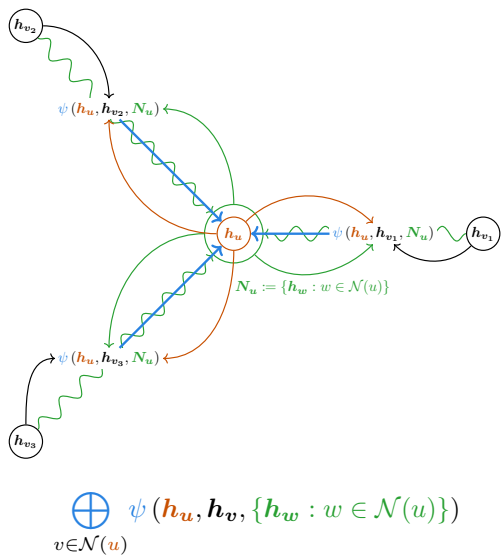


Figure 2: Neighborhood-Contextualized Message-Passing.

3.1 Soft-Isomorphic Neighborhood-Contextualized Graph Convolution Network: A Conceptual Proof

While Eq. (12) provides a novel theoretical paradigm for designing more powerful GNN architectures, it nevertheless still needs careful design choices. Critically, any operation on $\{\boldsymbol{h_w} : w \in \mathcal{N}(u)\}$ in an NCMP instance must be permutation-invariant, *i.e.*, order-independent while being flexible to arbitrary neighborhood size. One simple, practical, and efficient method for operationalizing NCMP is presented below.

By construction, since the message ψ in Eq. (12) is a function of $\{\boldsymbol{h_w} : w \in \mathcal{N}(u)\}$, it is neighborhood-contextualized. Moreover, following the theoretical development of SIR-GCN (Lim et al., 2025a), ψ may be modeled as a two-layer MLP, guaranteeing contextualized messages. Using block matrix operations on the concatenated inputs $\boldsymbol{h_u}$, $\boldsymbol{h_v}$, and $\boldsymbol{h_w}$'s, one may then *initially* consider the equivalent parametrization

$$\boldsymbol{h}_u^* = \bigoplus_{v \in \mathcal{N}(u)} \boldsymbol{W_R}\, \sigma\left(\boldsymbol{W_Q}\boldsymbol{h_u} + \boldsymbol{W_K}\boldsymbol{h_v} + \boldsymbol{N_u}\right), \tag{13}$$

where

$$\boldsymbol{N_u} := \sum_{w \in \mathcal{N}(u)} \boldsymbol{W}_{\boldsymbol{N}}^{(\boldsymbol{w})} \boldsymbol{h_w}, \tag{14}$$

$\bigoplus$ is some permutation-invariant aggregator, σ is a non-linear activation function, and $\boldsymbol{W_R}$, $\boldsymbol{W_Q}$, $\boldsymbol{W_K}$, $\boldsymbol{W_N^{(w)}}$'s are learnable linear transformations. In this formulation, $\boldsymbol{N_u}$ may be interpreted as a compressed *vector* representation for the one-hop neighborhood features $\{\boldsymbol{h_w} : w \in \mathcal{N}(u)\}$, generalizing its analogous scalar normalization factor in softmax attention. Crucially, however, this naive approach requires learning a distinct $\boldsymbol{W_N^{(w)}}$ for every node $w \in \mathcal{V}$. This is not merely *parameter inefficient* and *infeasible for inductive learning tasks* that require generalizing to unseen graphs; it also fundamentally *violates the core symmetry constraints of GNNs*. Specifically, learning distinct $\boldsymbol{W_N^{(w)}}$ would tightly couple the model to an arbitrary node ordering, thereby breaking permutation equivariance.

To address these critical limitations, consider instead a constant $\boldsymbol{W_N}$ shared across all nodes $w \in \mathcal{V}$. This formulation not only promotes *parameter efficiency*, but is also a strict theoretical necessity to preserve both *permutation equivariance* and *generalizability*. Furthermore, other GNN aggregators may also be used in place of the sum aggregator in Eq. (14), as it is noted to exhibit difficulty generalizing to unseen graphs (Veličković et al., 2020). This approach further promotes flexibility while still respecting the permutation-invariance on $\{\boldsymbol{h_w} : w \in \mathcal{N}(u)\}$. Combining these features then results in the proposed ***Soft-Isomorphic Neighborhood-Contextualized Graph Convolution Network (SINC-GCN)***[1] instantiation of the NCMP framework, expressed as

$$\boldsymbol{h_u^*} = \bigoplus_{v \in \mathcal{N}(u)} \boldsymbol{W_R}\, \sigma\left(\boldsymbol{W_Q h_u} + \boldsymbol{W_K h_v} + \bigotimes_{w \in \mathcal{N}(u)} \boldsymbol{W_N h_w}\right), \tag{15}$$

where $\bigoplus$ and $\bigotimes$ are some, potentially distinct, permutation-invariant aggregators (*e.g.*, sum, mean, symmetric mean, and max), σ is a non-linear activation function, $\boldsymbol{W_R} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{hidden}}}$, and $\boldsymbol{W_Q}, \boldsymbol{W_K}, \boldsymbol{W_N} \in \mathbb{R}^{d_{\text{hidden}} \times d_{\text{in}}}$. Moreover, for commutative aggregators $\bigoplus$ (*e.g.*, sum, mean, and symmetric mean), SINC-GCN has a computational complexity of

$$\mathcal{O}\left(|\mathcal{V}| \times d_{\text{hidden}} \times d_{\text{in}} + |\mathcal{E}| \times d_{\text{hidden}} + |\mathcal{V}| \times d_{\text{out}} \times d_{\text{hidden}}\right), \tag{16}$$

by leveraging linearity in Eq. (15), applying only an activation function along edges, and performing a two-step convolution constrained to the one-hop neighborhood receptive field. Specifically, extracting the neighborhood context representation only incurs an additional $\mathcal{O}\left(|\mathcal{V}| \times d_{\text{hidden}} \times d_{\text{in}}\right)$ overhead, making SINC-GCN comparable to classical one-hop localized GNNs in terms of asymptotic runtime complexity (Lim et al., 2025a), thereby underscoring its efficiency. Furthermore, SIR-GCN may also be viewed as an instance of SINC-GCN where $\boldsymbol{W_N} = \boldsymbol{0}$, positioning SIR-GCN as an *ablated model* of SINC-GCN.

Overall, SINC-GCN is a simple yet flexible *conceptual proof* of the proposed NCMP framework, grounded in established theoretical results for designing GNNs. By integrating both contextualized messages and neighborhood-contextualization, the proposed GNN architecture extends and generalizes classical one-hop localized GNNs while maintaining their computational efficiency.

3.1.1 Theoretical Limitations

As SIR-GCN was shown to be comparable to a modified 1-WL test (Lim et al., 2025a), it follows that SINC-GCN, as a generalization restricted to the one-hop neighborhood, retains the theoretical upper bounds of the 1-WL test in distinguishing graph isomorphism. Consequently, without additional positional or structural encodings, SINC-GCN cannot inherently distinguish complex higher-order topological substructures. Nonetheless, by integrating the aggregated neighborhood context representation into the pair-wise message computation, SINC-GCN can capture localized structural semantics—such as local feature distributions and relative neighborhood densities—that standard message-passing fundamentally ignore.

4 Results

To demonstrate the practical utility of the proposed NCMP framework and the expressivity of SINC-GCN, this section provides an extensive analysis of its performance across both synthetic and benchmark datasets

[1] Lim et al. (2025a) defines the term soft-isomorphic.

in node and graph property prediction tasks. Crucially, as the primary objective of this work is to lay the foundations for the NCMP framework, the proposed SINC-GCN simply serves as an *illustrative instance* and is *not* explicitly designed to achieve state-of-the-art performance. Hence, only one-hop localized GNN architectures are used as baselines, ensuring a fair performance evaluation.

4.1 Synthetic Dataset

UniqueSignature. This original synthetic dataset consists of randomly generated graphs, each having 30 to 70 nodes with an edge creation probability p_{edge} following the Erdős-Rényi model. Each node $u \in \mathcal{V}$ is also assigned an integer weight w_u, $-W \leq w_u \leq W$. The task is then to identify *catalyst* nodes—nodes u with a neighboring node $v \in \mathcal{N}(u)$ whose weight matches the total weight of all neighboring nodes of u, *i.e.*, $w_v = \sum_{w \in \mathcal{N}(u)} w_w$. As the graphs are generated randomly, the number of catalyst nodes naturally varies based on the generation configurations, as indicated by $\%_{\text{pos}}$. Motivated by previous works (Brody et al., 2022; Lim et al., 2025a), this dataset is *not* intended to serve as a comprehensive measure of general expressivity, but rather as a highly targeted *diagnostic probe* intentionally designed to highlight a specific limitation—neighborhood-agnosticism—inherent in existing message-passing architectures. By requiring the model to reason about the aggregate state of the local neighborhood, this *diagnostic* binary node classification problem empirically illustrates the necessity and utility of neighborhood-contextualization.

Table 2: Test Balanced Accuracy on UniqueSignature.

Model	**Dataset Configuration**								
	$W=1$ $p_{\text{edge}}=0.3$ $\%_{\text{pos}}=0.37$	$W=1$ $p_{\text{edge}}=0.5$ $\%_{\text{pos}}=0.29$	$W=1$ $p_{\text{edge}}=0.7$ $\%_{\text{pos}}=0.25$	$W=2$ $p_{\text{edge}}=0.3$ $\%_{\text{pos}}=0.35$	$W=2$ $p_{\text{edge}}=0.5$ $\%_{\text{pos}}=0.28$	$W=2$ $p_{\text{edge}}=0.7$ $\%_{\text{pos}}=0.24$	$W=3$ $p_{\text{edge}}=0.3$ $\%_{\text{pos}}=0.32$	$W=3$ $p_{\text{edge}}=0.5$ $\%_{\text{pos}}=0.27$	$W=3$ $p_{\text{edge}}=0.7$ $\%_{\text{pos}}=0.23$
GCN	0.50 ± 0.00	0.54 ± 0.13	0.59 ± 0.18	0.54 ± 0.12	0.63 ± 0.19	0.59 ± 0.17	0.54 ± 0.11	0.67 ± 0.20	0.67 ± 0.21
GraphSAGE	0.50 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	0.50 ± 0.00
GATv2	0.84 ± 0.17	0.81 ± 0.20	0.73 ± 0.23	0.62 ± 0.19	0.67 ± 0.21	0.54 ± 0.13	0.66 ± 0.19	0.64 ± 0.19	0.50 ± 0.00
GIN	0.84 ± 0.00	0.85 ± 0.00	0.86 ± 0.00	0.82 ± 0.00	0.84 ± 0.00	0.85 ± 0.00	0.80 ± 0.00	0.84 ± 0.00	0.84 ± 0.00
SIR-GCN	0.50 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	0.50 ± 0.00
PNA	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.99 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.98 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
EGC-S	0.50 ± 0.00	0.54 ± 0.11	0.50 ± 0.00	0.58 ± 0.16	0.54 ± 0.13	0.50 ± 0.00	0.50 ± 0.01	0.54 ± 0.12	0.50 ± 0.00
EGC-M	1.00 ± 0.00	1.00 ± 0.00	0.98 ± 0.05	0.97 ± 0.06	0.96 ± 0.08	0.96 ± 0.08	0.94 ± 0.08	0.93 ± 0.10	0.96 ± 0.09
SINC-GCN	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.01	1.00 ± 0.00	1.00 ± 0.00

Note: blue: best model.

Table 2 presents the mean and standard deviation of the test balanced accuracy for SINC-GCN and baseline models—GCN, GraphSAGE, GATv2, GIN, and SIR-GCN—across different dataset configurations W and p_{edge} with varying percentage of positive class $\%_{\text{pos}}$. The performance for more advanced models—PNA, EGC-S, and EGC-M—is also presented as additional baselines. Notably, SINC-GCN consistently achieves perfect accuracy, attributed to its contextualized messages and neighborhood-contextualization. This design allows it to correctly identify *catalyst* nodes using the features of each neighboring node, contextualized on the entire set of neighborhood features. In fact, it may even be shown that with the appropriate choice of parameters $\bigoplus = \sum$, $\bigotimes = \sum$, $\sigma = \text{ReLU}$, $\boldsymbol{W_R} = [-1, -1]$, $\boldsymbol{W_Q} = \boldsymbol{0}$, $\boldsymbol{W_K} = [1, -1]^\top$, and $\boldsymbol{W_N} = [-1, 1]^\top$, SINC-GCN will mathematically always produce the correct classifications. Meanwhile, GCN, GraphSAGE, SIR-GCN, and EGC-S exhibit near-random performance, since their architectural design does not explicitly allow them to learn the appropriate relationship needed for this simple task. Likewise, GATv2 and GIN perform better than random on simpler dataset configurations, but fail to generalize well as problem complexity increases. In contrast, the performance of PNA and EGC-M is substantially better than random, as their use of the more *exotic* standard deviation aggregator *implicitly* involves the mean of the neighborhood features as standardization. Nevertheless, their performance comes with greater computational costs, as presented in Table 5 on Appendix B. Overall, the results illustrate the limitations of classical message-passing GNNs, the utility of both contextualized messages and neighborhood-contextualization, and the expressivity and efficiency of SINC-GCN.

4.2 Benchmark Datasets

Benchmarking GNNs (Dwivedi et al., 2023). This collection of benchmark datasets features a variety of mathematical and real-world graphs for various GNN tasks. Specifically, the WikiCS, PATTERN, and CLUSTER datasets are tailored for node property prediction tasks, while the MNIST, CIFAR10, and ZINC datasets are designed for graph property prediction tasks. Additionally, the mean absolute error (MAE) is the performance metric for ZINC, while accuracy is the primary metric for the remaining datasets. Collectively, these six datasets cover a diverse range of GNN applications, facilitating a comprehensive evaluation of model performance. Dwivedi et al. (2023) provides detailed information on the individual datasets.

Open Graph Benchmark (Hu et al., 2020). This collection of datasets offers realistic, extensive, and varied benchmarks suitable for GNNs. Specifically, the ogbn-arxiv dataset is used for node property prediction tasks. Meanwhile, the ogbg-molhiv dataset is designated for graph property prediction tasks. Accuracy serves as the performance metric for ogbn-arxiv, while the area under the receiver operating characteristic curve (ROC-AUC) is the primary metric for ogbg-molhiv. Hu et al. (2020) provides more details regarding the specific datasets.

Table 3: Test Performance on Benchmark Datasets.

Model	**WikiCS (↑)**	**PATTERN (↑)**	**CLUSTER (↑)**	**MNIST (↑)**	**CIFAR10 (↑)**	**ZINC (↓)**	**ogbn-arxiv (↑)**	**ogbg-molhiv (↑)**
GCN	77.47 ± 0.85	85.50 ± 0.05	47.83 ± 1.51	90.12 ± 0.15	54.14 ± 0.39	0.416 ± 0.006	71.92 ± 0.21	76.14 ± 1.29
GraphSAGE	74.77 ± 0.95	50.52 ± 0.00	50.45 ± 0.15	97.31 ± 0.10	65.77 ± 0.31	0.468 ± 0.003	71.73 ± 0.26	75.97 ± 1.69
GATv2	-	-	-	-	67.48 ± 0.53	0.447 ± 0.015	71.87 ± 0.43	77.15 ± 1.55
GIN	75.86 ± 0.58	85.59 ± 0.01	58.38 ± 0.24	96.49 ± 0.25	55.26 ± 1.53	0.387 ± 0.015	67.33 ± 1.47	76.02 ± 1.35
SIR-GCN	78.06 ± 0.66	85.75 ± 0.03	63.35 ± 0.19	97.90 ± 0.08	71.98 ± 0.40	0.278 ± 0.024	72.52 ± 0.16	77.63 ± 0.84
PNA	-	-	-	97.19 ± 0.08	70.21 ± 0.15	0.320 ± 0.032	71.21 ± 0.30	79.05 ± 1.32
EGC-S	-	-	-	-	66.92 ± 0.37	0.364 ± 0.020	72.21 ± 0.17	77.44 ± 1.08
EGC-M	-	-	-	-	71.03 ± 0.42	0.281 ± 0.007	71.96 ± 0.23	78.18 ± 1.53
SINC-GCN	78.17 ± 0.68	**85.79 ± 0.02**	63.51 ± 0.15	**98.28 ± 0.05**	**73.37 ± 0.41**	**0.256 ± 0.006**	**72.66 ± 0.09**	78.50 ± 1.23

Notes: blue: best model; **bold**: statistically significant by Welch's t-test at $\alpha = 0.05$ vs. best baseline model; missing values: no publicly published results.

Table 3 presents the mean and standard deviation of the test performance for SINC-GCN and baseline models—GCN, GraphSAGE, GATv2, GIN, and SIR-GCN—across the eight benchmark datasets. Crucially, the reported results for SINC-GCN follow the experimental configuration of Dwivedi et al. (2023) as presented in Appendix A, ensuring differences in model performance are primarily attributed to the GNN architecture. Notably, SINC-GCN achieves competitive performance and statistically significant gains against baseline one-hop localized GNNs across all datasets, all while operating with a *smaller hidden representation* and incurring only *minimal asymptotic computational overhead*. While the absolute performance gains on node property prediction tasks are modest, SINC-GCN achieves these with notably tighter standard deviations, highlighting how neighborhood-contextualization not only improves performance, but also makes the model more robust against random initialization variance. Crucially, the performance gains are markedly more prominent for graph property prediction tasks. This disparity highlights a key advantage of the NCMP framework: neighborhood-contextualization is particularly vital for graph-level representations. Because graph-level tasks require aggregating relational information across the entire network, equipping local message-passing with the broader structural and semantic context of the neighborhood allows the final readout function to capture a much richer global representation than neighborhood-agnostic baselines. The results thus reveal the significance of neighborhood-contextualization and position SINC-GCN as a performant and efficient alternative to classical GNNs.

Furthermore, the test performance for more advanced models—PNA, EGC-S, and EGC-M—is also presented in Table 3 to provide a more comprehensive evaluation. While PNA achieves superior performance on ogbg-molhiv, where preserving injectivity is critical, across the majority of datasets, these computationally expensive advanced aggregation techniques fail to outperform the simpler and more efficient neighborhood-contextualization in SINC-GCN. Crucially, as demonstrated in Appendix B, architectures like PNA and EGC-M incur massive computational overhead. Hence, the significance and utility of SINC-GCN lie in its Pareto efficiency: it delivers consistent, statistically significant improvements over classical GNN baselines

while maintaining their highly efficient asymptotic complexity. Rather than relying on computationally expensive techniques, SINC-GCN demonstrates that effectively contextualizing the local neighborhood offers a highly scalable method for enhancing graph expressivity.

4.2.1 Ablation & Sensitivity Analysis

To systematically validate the core contribution of the proposed NCMP framework, Table 4 further presents a comprehensive ablation and sensitivity analysis across all benchmark datasets. The analyses evaluate SINC-GCN under various parameterizations of the neighborhood aggregator $\bigotimes$, including a direct ablation where $\bigotimes$ is entirely disabled (none).

Table 4: Ablation & Sensitivity Analysis on Benchmark Datasets.

Aggregator $\bigotimes$	WikiCS (↑)	PATTERN (↑)	CLUSTER (↑)	MNIST (↑)	CIFAR10 (↑)	ZINC (↓)	ogbn-arxiv (↑)	ogbg-molhiv (↑)
none	78.06 ± 0.66	85.75 ± 0.03	63.35 ± 0.19	97.90 ± 0.08	71.98 ± 0.40	0.278 ± 0.024	72.52 ± 0.16	77.63 ± 0.84
sum	FP Error	85.74 ± 0.02	59.46 ± 0.09	97.97 ± 0.10	70.08 ± 1.25	0.258 ± 0.008	70.05 ± 0.74	77.33 ± 0.93
max	76.05 ± 1.69	85.64 ± 0.06	62.25 ± 0.32	98.13 ± 0.10	72.19 ± 0.60	0.262 ± 0.009	71.55 ± 0.16	75.22 ± 1.15
mean	78.17 ± 0.68	85.79 ± 0.02	63.51 ± 0.15	98.28 ± 0.05	73.37 ± 0.41	0.256 ± 0.006	72.22 ± 0.09	76.43 ± 0.77
sym. mean	78.00 ± 0.75	85.79 ± 0.02	63.38 ± 0.16	98.18 ± 0.11	73.16 ± 0.26	0.262 ± 0.017	72.66 ± 0.09	78.50 ± 1.23

Ablating Neighborhood Context. Setting $\bigotimes$ to none effectively eliminates neighborhood-contextualization from the convolution operation, directly reducing SINC-GCN to the baseline SIR-GCN. The results demonstrate a clear, consistent performance drop across almost all datasets. The performance gap is particularly prominent in tasks requiring deeper structural awareness. For instance, the MAE increased from 0.256 (mean) to 0.278 (none) in ZINC, while the accuracy drops from 73.37% (mean) to 71.98% (none) in CIFAR10. This direct ablation explicitly isolates the proposed neighborhood-contextualization and underscores the representational value of making the message-passing operation structurally and semantically aware of the broader local neighborhood.

Sensitivity to Aggregator Choice. Beyond the core ablation, Table 4 also illustrates the sensitivity of SINC-GCN to the specific $\bigotimes$ used to extract the neighborhood context representation. The sum aggregator exhibits severe instability—yielding a floating-point (FP) error on WikiCS—and generally sub-optimal performance across datasets like CLUSTER and ogbn-arxiv. This behavior highlights a critical sensitivity: unnormalized sum aggregations over highly variable neighborhood degrees lead to large feature variances and optimization failures (Veličković et al., 2020). Similarly, the max aggregator discards significant structural nuance by only retaining extreme feature values, resulting in less expressive neighborhood context representations compared to the optimal configurations (Xu et al., 2019). Conversely, the mean and symmetric mean (sym. mean) aggregators deliver the most robust and performant results across the eight datasets. By naturally normalizing the contextual vector against the neighborhood size, these aggregators provide a stable, smooth representation of the local structural and semantic context. Ultimately, this sensitivity analysis validates the practical design of SINC-GCN: utilizing a simple, normalized representation of the neighborhood results in statistically significant performance gains over standard baselines, entirely avoiding the massive computational overhead of more advanced models.

Overall, these benchmark datasets underscore the viability and potential of the proposed NCMP framework in offering a simple and practical path toward designing more powerful GNN architectures.

5 Conclusion

In summary, the contribution of this work is threefold. It first formalizes the concept of ***neighborhood-contextualization*** in GNNs, motivated by the implicit property of the attentional variant. It then proposes a novel generalization of the message-passing variant called ***neighborhood-contextualized message passing (NCMP)***, which features both contextualized messages and neighborhood-contextualization. To illustrate its practical utility, a theoretically-grounded method for parametrizing NCMP is presented, leading to the development of the proposed ***Soft-Isomorphic Neighborhood-Contextualized Graph Convolution***

Network (SINC-GCN) as a simple, practical, and efficient *conceptual proof* of the proposed framework. A comprehensive evaluation, spanning both synthetic and benchmark datasets in node and graph property prediction tasks, demonstrates how SINC-GCN achieves consistent gains against baseline GNN architectures, striking an optimal balance between model expressivity and computational efficiency. Overall, the results underscore the potential of SINC-GCN for various GNN applications and the practical contribution of the proposed NCMP framework in enhancing the representational power of classical GNNs via local structural semantics without significant computational overhead. Future works may consider neighborhood-contextualization as a robust alternative to softmax attention across various applications, explore more expressive NCMP parametrizations, and apply SINC-GCN to tasks and domains where neighborhood context is paramount.

References

Waiss Azizian and Marc Lelarge. Expressive power of invariant and equivariant graph neural networks. In *International Conference on Learning Representations*, 2021.

Cristian Bodnar, Fabrizio Frasca, Nina Otter, Yuguang Wang, Pietro Liò, Guido F. Montufar, and Michael Bronstein. Weisfeiler and Lehman go cellular: CW networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 2625–2640, 2021.

Giorgos Bouritsas, Fabrizio Frasca, Stefanos Zafeiriou, and Michael M. Bronstein. Improving graph neural network expressivity via subgraph isomorphism counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):657–668, 2023.

Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *International Conference on Learning Representations*, 2022.

Gabriele Corso, Luca Cavalleri, Dominique Beaini, Pietro Liò, and Petar Veličković. Principal neighbourhood aggregation for graph nets. In *Advances in Neural Information Processing Systems*, volume 33, pp. 13260–13271, 2020.

Vijay Prakash Dwivedi, Chaitanya K. Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *Journal of Machine Learning Research*, 24(43):1–48, 2023.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1263–1272, 2017.

Yi-Ling Hsu, Yu-Che Tsai, and Cheng-Te Li. FinGAT: Financial graph attention networks for recommending top-k profitable stocks. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):469–481, 2023.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems*, volume 33, pp. 22118–22133, 2020.

Nan Jiang, Jie Wen, Jin Li, Ximeng Liu, and Di Jin. GATrust: A multi-aspect graph attention network model for trust assessment in OSNs. *IEEE Transactions on Knowledge and Data Engineering*, 35(6): 5865–5878, 2023.

Byung-Hoon Kim and Jong Chul Ye. Understanding graph isomorphism network for rs-fMRI functional connectivity analysis. *Frontiers in Neuroscience*, 14:630, 2020.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Veronica Lachi, Francesco Ferrini, Antonio Longa, Bruno Lepri, and Andrea Passerini. A simple and expressive graph neural network based method for structural link representation. In *Proceedings of the Geometry-grounded Representation Learning and Generative Modeling Workshop (GRaM)*, volume 251 of *Proceedings of Machine Learning Research*, pp. 187–201, 2024.

Guohao Li, Chenxin Xiong, Ali Thabet, and Bernard Ghanem. DeeperGCN: All you need to train deeper GCNs, 2020. arXiv:2006.07739.

Brian Godwin Lim, Galvin Brice Lim, Renzo Roel Tan, and Kazushi Ikeda. Contextualized messages boost graph representations. *Transactions on Machine Learning Research*, 2025a. URL `https://openreview.net/forum?id=sXr1fRjs1N`.

Brian Godwin Lim, Jiahong Liu, Hans Jarett Ong, Jan Adrian Chan, Renzo Roel Tan, Irwin King, and Kazushi Ikeda. FinSIR: Financial SIR-GCN for market-aware stock recommendation. In *2025 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2025b.

Galvin Brice S. Lim, Brian Godwin S. Lim, Argel A. Bandala, John Anthony C. Jose, Timothy Scott C. Chu, and Edwin Sybingco. AGTCNet: A graph-temporal approach for principled motor imagery EEG classification. *IEEE Access*, 13:187383–187409, 2025c.

Jielun Liu, Ghim Ping Ong, and Xiqun Chen. GraphSAGE-based traffic speed forecasting for segment network with sparse data. *IEEE Transactions on Intelligent Transportation Systems*, 23(3):1755–1766, 2022.

Chengsheng Mao, Liang Yao, and Yuan Luo. Towards expressive graph representations for graph neural networks. In *2024 IEEE International Conference on Data Mining (ICDM)*, pp. 797–802, 2024.

Christopher Morris, Martin Ritzert, Matthias Fey, William L. Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: higher-order graph neural networks. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19, pp. 4602–4609. AAAI Press, 2019.

Ryoma Sato, Makoto Yamada, and Hisashi Kashima. Random features strengthen graph neural networks. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pp. 333–341, 2021.

Chuxiong Sun, Jie Hu, Hongming Gu, Jinpeng Chen, and Mingchuan Yang. Adaptive graph diffusion networks, 2022. arXiv:2012.15024.

Junshu Sun, Chenxue Yang, Xiangyang Ji, Qingming Huang, and Shuhui Wang. Towards dynamic message passing on graphs. In *Advances in Neural Information Processing Systems*, volume 37, pp. 80936–80964, 2024.

Shyam A. Tailor, Felix Opolka, Pietro Lio, and Nicholas Donald Lane. Do we need anisotropic graph neural networks? In *International Conference on Learning Representations*, 2022.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Petar Veličković, Rex Ying, Matilde Padovano, Raia Hadsell, and Charles Blundell. Neural execution of graph algorithms. In *International Conference on Learning Representations*, 2020.

Ziming Wang, Jun Chen, and Haopeng Chen. EGAT: Edge-featured graph attention network. In *Artificial Neural Networks and Machine Learning – ICANN 2021*, pp. 253–264, Cham, 2021. Springer.

Boris Weisfeiler and Andrei Leman. The reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsia*, 2(9):12–16, 1968.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? In *Advances in Neural Information Processing Systems*, volume 34, pp. 28877–28888, 2021.

A Experimental Set-up

The reported results for the synthetic dataset are obtained from the models at the final epoch across 5 seed initializations, while results for the benchmark datasets are obtained from the models with the best validation loss across 5 seed initializations. All experiments are conducted on a single NVIDIA® A800 (40GB) GPU using Deep Graph Library (DGL) with PyTorch backend.

A.1 Synthetic Dataset

UniqueSignature. The models are trained using a set of 4,000 graphs and evaluated against a separate set of 1,000 graphs. These graphs are generated using the Erdős-Rényi model, each having 30 to 70 nodes with an edge creation probability p_{edge}. All reported results use a single GNN layer with 16 hidden units. Moreover, a two-layer MLP is used for GIN, while both PNA and EGC-M use the sum, max, and standard deviation aggregators. The models are then trained using the AdamW optimizer for 500 epochs with a 1×10^{-3} learning rate and a batch size of 256. The learning rate is also scheduled to decay by a factor of 0.5 with a patience of 10 epochs based on the training loss.

A.2 Benchmark Datasets

Benchmarking GNNs (Dwivedi et al., 2023). Following the experimental set-up of Dwivedi et al. (2023); Corso et al. (2020); Tailor et al. (2022); Lim et al. (2025a), the reported results for SINC-GCN also use 4 GNN layers, employing batch normalization and residual connections, while constrained to a 100,000 parameter budget without extensive tuning. Consequently, SINC-GCN operates with a smaller hidden representation due to the additional parameters $\boldsymbol{W_N}$. To prevent overfitting, weight decays of rate 1×10^{-1} and dropouts with rates in $\{0.1, 0.2, 0.3\}$ are also employed. Additionally, $\bigoplus$ is chosen as either the mean, symmetric mean, or max aggregator, similar to SIR-GCN (Lim et al., 2025a), while $\bigotimes$ is simply chosen as the mean aggregator. The graph readout function is chosen as the sum aggregator for ZINC and the mean aggregator for MNIST and CIFAR10. The models are then trained using the AdamW optimizer for a maximum of 500 epochs with a 1×10^{-3} learning rate and a batch size of 128, when applicable. The learning rate is also scheduled to decay by a factor of 0.5 with a patience of 10 epochs based on the validation loss. The results for other models in Table 3 are obtained from previous works.

Open Graph Benchmark (Hu et al., 2020). Following the experimental set-up of Corso et al. (2020); Tailor et al. (2022); Lim et al. (2025a), the reported results for SINC-GCN also use 4 GNN layers, employing batch normalization and residual connections, while constrained to a 100,000 parameter budget without extensive tuning. Similarly, SINC-GCN operates with a smaller hidden representation due to the additional parameters $\boldsymbol{W_N}$. To prevent overfitting, weight decays with factors in $\{1 \times 10^{-3}, 1 \times 10^{-1}\}$ and dropouts with rates in $\{0.1, 0.2, 0.3, 0.4\}$ are also employed. Additionally, $\bigoplus$ is chosen as the mean aggregator, while $\bigotimes$ is simply chosen as the symmetric mean aggregator. The graph readout function is chosen as the mean aggregator for ogbg-molhiv. The models are then trained using the AdamW optimizer for a maximum of 1000 epochs with a learning rate in $\{1 \times 10^{-3}, 1 \times 10^{-2}\}$ and a batch size of 64 for ogbg-molhiv. The learning rate is also scheduled to decay by a factor of 0.5 with a patience of 10 or 50 epochs based on the validation loss. The results for other models in Table 3 are obtained from previous works.

B Runtime Analysis

The inference runtime for each model in UniqueSignature is presented in Table 5. These figures underscore how the proposed GNN architecture achieves a strong balance between model expressivity and computational efficiency. In particular, SINC-GCN has an inference runtime comparable to the baseline models—GCN, GraphSAGE, GATv2, GIN, and SIR-GCN—yet is functionally more expressive than these architectures. Moreover, when considered alongside the results in Table 2, they highlight PNA and EGC-M incurring significantly higher computational costs for their performance, in stark contrast to the significantly shorter runtime of SINC-GCN. Overall, these additional results demonstrate the practical utility of the proposed GNN architecture.

Table 5: UniqueSignature Inference Runtime.

Model	**Dataset Configuration**								
	$W = 1$ $p_{\text{edge}} = 0.3$ $\%_{\text{pos}} = 0.37$	$W = 1$ $p_{\text{edge}} = 0.5$ $\%_{\text{pos}} = 0.29$	$W = 1$ $p_{\text{edge}} = 0.7$ $\%_{\text{pos}} = 0.25$	$W = 2$ $p_{\text{edge}} = 0.3$ $\%_{\text{pos}} = 0.35$	$W = 2$ $p_{\text{edge}} = 0.5$ $\%_{\text{pos}} = 0.28$	$W = 2$ $p_{\text{edge}} = 0.7$ $\%_{\text{pos}} = 0.24$	$W = 3$ $p_{\text{edge}} = 0.3$ $\%_{\text{pos}} = 0.32$	$W = 3$ $p_{\text{edge}} = 0.5$ $\%_{\text{pos}} = 0.27$	$W = 3$ $p_{\text{edge}} = 0.7$ $\%_{\text{pos}} = 0.23$
GCN	0.32s ± 0.10s	0.37s ± 0.09s	0.41s ± 0.10s	0.34s ± 0.12s	0.38s ± 0.14s	0.41s ± 0.10s	0.34s ± 0.11s	0.37s ± 0.08s	0.40s ± 0.10s
GraphSAGE	0.32s ± 0.10s	0.37s ± 0.09s	0.41s ± 0.10s	0.33s ± 0.11s	0.39s ± 0.14s	0.41s ± 0.10s	0.34s ± 0.11s	0.39s ± 0.13s	0.40s ± 0.10s
GATv2	0.33s ± 0.11s	0.39s ± 0.13s	0.41s ± 0.10s	0.34s ± 0.11s	0.39s ± 0.13s	0.41s ± 0.10s	0.34s ± 0.12s	0.40s ± 0.13s	0.41s ± 0.10s
GIN	0.32s ± 0.10s	0.39s ± 0.14s	0.40s ± 0.10s	0.34s ± 0.11s	0.38s ± 0.13s	0.40s ± 0.10s	0.32s ± 0.10s	0.37s ± 0.08s	0.40s ± 0.10s
SIR-GCN	0.34s ± 0.12s	0.39s ± 0.13s	0.41s ± 0.10s	0.34s ± 0.12s	0.38s ± 0.13s	0.41s ± 0.10s	0.34s ± 0.11s	0.39s ± 0.14s	0.40s ± 0.10s
PNA	0.62s ± 0.10s	0.72s ± 0.14s	0.75s ± 0.10s	0.64s ± 0.11s	0.72s ± 0.14s	0.75s ± 0.10s	0.65s ± 0.12s	0.72s ± 0.13s	0.75s ± 0.09s
EGC-S	0.39s ± 0.10s	0.50s ± 0.15s	0.56s ± 0.10s	0.39s ± 0.11s	0.48s ± 0.09s	0.55s ± 0.10s	0.39s ± 0.11s	0.50s ± 0.15s	0.55s ± 0.10s
EGC-M	0.65s ± 0.10s	0.92s ± 0.15s	1.12s ± 0.09s	0.65s ± 0.10s	0.92s ± 0.15s	1.12s ± 0.09s	0.66s ± 0.10s	0.90s ± 0.08s	1.12s ± 0.09s
SINC-GCN	0.34s ± 0.11s	0.39s ± 0.13s	0.40s ± 0.09s	0.34s ± 0.11s	0.39s ± 0.13s	0.40s ± 0.10s	0.32s ± 0.11s	0.39s ± 0.13s	0.40s ± 0.09s