# On Oversquashing in Graph Neural Networks Through the Lens of Dynamical Systems

**Alessio Gravina**[*1], **Moshe Eliasof**[*2], **Claudio Gallicchio**[1], **Davide Bacciu**[1], **Carola-Bibiane Schönlieb**[2]
[1]Department of Computer Science, University of Pisa, Pisa, Italy
[2]Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge, United Kingdom
alessio.gravina@di.unipi.it, me532@cam.ac.uk, {claudio.gallicchio, davide.bacciu}@unipi.it, cbs31@cam.ac.uk

**Abstract**

A common problem in Message-Passing Neural Networks is oversquashing – the limited ability to facilitate effective information flow between distant nodes. Oversquashing is attributed to the exponential decay in information transmission as node distances increase. This paper introduces a novel perspective to address oversquashing, leveraging dynamical systems properties of global and local non-dissipativity, that enable the maintenance of a constant information flow rate. We present SWAN, a uniquely parameterized GNN model with antisymmetry both in space and weight domains, as a means to obtain non-dissipativity. Our theoretical analysis asserts that by implementing these properties, SWAN offers an enhanced ability to transmit information over extended distances. Empirical evaluations on synthetic and real-world benchmarks that emphasize long-range interactions validate the theoretical understanding of SWAN, and its ability to mitigate oversquashing.

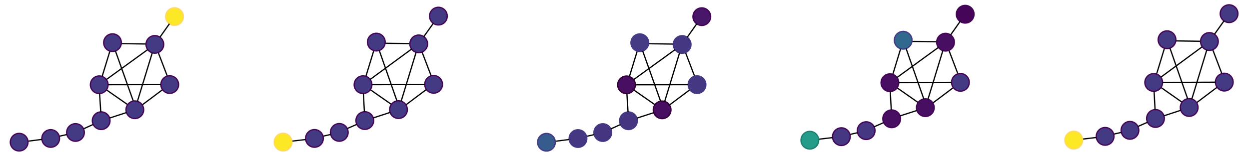
## 1 Introduction

A critical issue that limits Message-Passing Neural Networks (MPNNs) (Gilmer et al. 2017), a class of GNNs, is the oversquashing problem (Alon and Yahav 2021; Di Giovanni et al. 2023). In the oversquashing scenario, the capacity of MPNNs to transmit information between nodes exponentially decreases as their distance increases, which imposes challenges on modeling long-range interactions, which are often necessary for real-world tasks (Dwivedi et al. 2022). At the same time, it has been shown in (Haber et al. 2018; Chen et al. 2018) that neural networks can be interpreted as the discretization of ordinary differential equations (ODEs). Building on these observations and understandings, similar concepts were utilized to forge the field of differential-equations inspired GNNs (DE-GNNs), as shown in (Poli et al. 2019; Chamberlain et al. 2021b) and subsequent works. Through this view, it is possible to design GNNs with strong inductive biases, such as smoothness (Chamberlain et al. 2021b), energy preservation (Eliasof, Haber, and Treister 2021; Rusch et al. 2022), node-wise non-dissipativity (Gravina, Bacciu, and Gallicchio 2023), and more.

In this paper, we are interested in addressing the oversquashing problem in a principled manner, accompanied by theoretical understanding through the prism of DE-GNNs. Current literature offers several approaches to mitigate oversquashing, such as adding a virtual global node (Gilmer et al. 2017; Cai et al. 2023), graph rewiring (Gasteiger, Weißenberger, and Günnemann 2019; Topping et al. 2022; Karhadkar, Banerjee, and Montufar 2023), as well as using graph transformers (Dwivedi and Bresson 2021; Rampášek et al. 2022). Some of the methods above focus on providing ad-hoc mechanisms for the network to reduce oversquashing. Instead, in this paper, we are interested in theoretically understanding and improving the information propagation capacity of the network, thereby mitigating oversquashing. Specifically, we will show that it is possible to design GNNs with a constant flow of information, unlike typical diffusion-based methods. To this end, we take inspiration from the recent ADGN (Gravina, Bacciu, and Gallicchio 2023), a non-dissipative GNN. In ADGN, it was shown that by incorporating an antisymmetric transformation to the learned channel-mixing weights, it is possible to obtain a *locally*, i.e., node-wise, non-dissipative behavior. Here, we propose **SWAN** (**S**pace-**W**eight **AN**tisymmetry), a novel GNN model that is both *globally* (i.e., graph-wise) and *locally* (i.e., node-wise) non-dissipative, achieved by space and weight antisymmetric parameterization. To understand the behavior of SWAN and its effectiveness in mitigating oversquashing, we propose a *global*, i.e., graph-wise, analysis, and show that compared to ADGN (Gravina, Bacciu, and Gallicchio 2023), our SWAN is both globally and locally non-dissipative. The immediate implication of this property is that SWAN is guaranteed to have a constant information flow rate, thereby mitigating oversquashing. Such a property is visualized in Figure 1, where SWAN shows improved capacities of propagating information across the graph, with respect to diffusion and local non-dissipative approaches. To complement our theoretical analysis, we experiment with several synthetic and real-world long-range benchmarks.

**Main Contributions.** We present the following contributions. (i) A novel graph perspective theoretical analysis of the stability and non-dissipativity of antisymmetric DE-GNNs, providing a general design principle for introducing non-dissipativity as an inductive bias in any DE-GNN model. (ii) We propose SWAN, a space and weight antisymmetric DE-

[*]These authors contributed equally.



(a) Source (b) Target (c) Diffusion (d) Local Non-Dissipativity (e) Local and Global Non-Dissipativity

Figure 1: An illustration of the ability of Global and Local Non-Dissipativity in SWAN (e) to propagate information to distant nodes, from source (a) to the target (b). Other dynamics, such as diffusion (c) cannot achieve this behavior, while Local Non-Dissipativity (d) offers a limited effect.

GNN with a constant information flow rate, and an increased distant node interaction sensitivity. (iii) We experimentally verify our theoretical understanding of SWAN, on both synthetic and real-world datasets. Our experiments show score improvements of up to 117% in synthetic datasets, as well as competitive results on real-world benchmarks compared to existing methods, from MPNNs to other DE-GNNs, and graph transformers – highlighting the importance of global and local non-dissipative behavior offered by SWAN.

## 2 Preliminaries

We provide an overview of oversquashing problem in GNNs, followed by a brief discussion of DE-GNNs and antisymmetric weights in DE-GNNs. We then provide a mathematical background, which is thoroughly used in this paper to analyze and understand the properties of SWAN.

### 2.1 Oversquashing

*Oversquashing* (Alon and Yahav 2021) refers to the shortcoming of a GNN when transferring information between distant nodes, and is a common problem in Message-Passing Neural Networks (MPNNs). Because standard MPNNs update node states by aggregating neighborhood information, oversqashing is amplified as node distances increase, hampering the ability of MPNNs to model complex behaviors that require long-range interactions. Namely, to allow a node to receive information from $k$-hops distant nodes, an MPNN must employ at least $k$ layers, otherwise, it will suffer from *under-reaching*, because the two nodes are too far to interact with each other. However, the stacking of multiple message-passing layers also causes each node to receive an exponentially growing amount of information, as multiple hops are considered. This exponential growth of information, combined with the finite and fixed number of channels (features), can lead to loss of information and reduced long-range effectiveness.

### 2.2 GNNs Inspired by Differential-Equations

**Notation.** We consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ as a system of interacting entities, referred to as nodes, where $\mathcal{V}$ is a set of $n$ nodes, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of $m$ edges. We define the neighborhood of the $u$-th node as the set of nodes directly attached to it, i.e., $\mathcal{N}_u = \{v | (v, u) \in \mathcal{E}\}$. The $u$-th node is associated with a (possibly time-dependent) hidden feature vector $\mathbf{x}_u(t) \in \mathbb{R}^d$ with $d$ features, which provides a representation of the node at time $t$ in the system. The term $\mathbf{X}(t) = [\mathbf{x}_0(t), \dots, \mathbf{x}_{n-1}(t)]^\top$ is an $n \times d$ matrix that represents the nodes state (features) at time $t$.

**DE-GNNs.** The core idea of DE-GNNs is to view GNNs as the discretization of the following ODE:

$$\frac{\partial \mathbf{x}_u(t)}{\partial t} = s\left(\{\mathbf{x}_v(t)\}_{v \in \mathcal{N}_u}; \mathcal{G}\right), \quad (1)$$

where $\mathbf{x}_u(t = 0) = \mathbf{x}^{(0)}$ and $s\left(\{\mathbf{x}_v(t)\}_{v \in \mathcal{N}_u}; \mathcal{G}\right)$ is a spatial aggregation function that depends on the graph $\mathcal{G}$ and the node features $\mathbf{x}_u(t)$. Specifically, it is common to implement $s\left(\{\mathbf{x}_v(t)\}_{v \in \mathcal{N}_u}; \mathcal{G}\right)$ with graph diffusion, combined with a channel mixing operator implemented by a multilayer perceptron (MLP). Some examples of such methods were proposed in (Chamberlain et al. 2021b; Eliasof, Haber, and Treister 2021; Choi et al. 2023), and others.

**Antisymmetric Weights in DE-GNNs.** Learned antisymmetric *weights* were studied in ADGN (Gravina, Bacciu, and Gallicchio 2023), summarized below for completeness:

$$\frac{\partial \mathbf{x}_u(t)}{\partial t} = \sigma\Big((\mathbf{W} - \mathbf{W}^\top)\mathbf{x}_u(t) + \Phi(\{\mathbf{x}_v\}_{v \in \mathcal{N}_u}, \mathbf{V})\Big), \quad (2)$$

where $\mathbf{W}, \mathbf{V} \in \mathbb{R}^{d \times d}$ are learnable weights, $\Phi(\{\mathbf{x}_v\}_{v \in \mathcal{N}_u}, \mathbf{V})$ is any permutation invariant neighborhood aggregation function, and $\sigma$ is an activation function. The main theoretical property of ADGN (Gravina, Bacciu, and Gallicchio 2023) is that it allows a *stable* and *non-dissipative* propagation from a *local*, node-wise perspective.

### 2.3 Mathematical Background

While Alon and Yahav (2021) did not mathematically define oversquashing, recent works (Topping et al. 2022; Di Giovanni et al. 2023) associate it with exponentially declining node embedding sensitivity. Building on this perspective, we connect oversquashing to non-dissipative dynamical systems, leading to our SWAN. Specifically, we are interested in studying the *stability* and *non-dissipativity* propagation of information in DE-GNNs. Therefore, we follow the analysis techniques presented in (Chang et al. 2019; Gravina, Bacciu, and Gallicchio 2023) and focus on analyzing the sensitivity of an ODE solution with respect to its initial condition, i.e.,

$$\frac{d}{dt}\left(\frac{\partial \mathbf{x}(t)}{\partial \mathbf{x}(0)}\right) = \mathbf{J}(t)\frac{\partial \mathbf{x}(t)}{\partial \mathbf{x}(0)}. \quad (3)$$

We now present an overview of the various outcomes that can arise from this analysis. We follow the results and assumptions from (Chang et al. 2019; Gravina, Bacciu, and Gallicchio 2023), and consider the Jacobian, $\mathbf{J}(t)$ in Equation (3) to not change significantly over time. We assume this condition in Theorems 3.1 and 3.2. In Appendix I, we provide a discussion of the justification as well as numerical verification of our assumption. Given that, we can apply results from autonomous differential equations (Ascher and Petzold 1998) and solve Equation (3) analytically as follows:

$$\frac{\partial \mathbf{x}(t)}{\partial \mathbf{x}(0)} = e^{t\mathbf{J}} = \mathbf{T}e^{t\mathbf{\Lambda}}\mathbf{T}^{-1} = \mathbf{T}\big(\sum_{k=0}^{\infty}\frac{(t\mathbf{\Lambda})^k}{k!}\big)\mathbf{T}^{-1}, \quad (4)$$

where $\mathbf{\Lambda}$ is the diagonal matrix whose non-zero entries contain the eigenvalues of $\mathbf{J}$ (i.e., $\lambda_i$), and $\mathbf{T}$ has the eigenvectors of $\mathbf{J}$ as columns. We observe that the qualitative behavior of $\partial \mathbf{x}(t)/\partial \mathbf{x}(0)$ is determined by the real parts of the eigenvalues of $\mathbf{J}$, leading to three different behaviors: (i) instability, (ii) dissipativity (i.e., information loss), (iii) non-dissipativity (i.e., information preservation).

**Instability** is observed when $\max_{i=1,...,d} Re(\lambda_i(\mathbf{J})) > 0$, i.e., a small perturbation of the initial condition would cause an exponential divergence in node representations.

**Dissipativity** occurs when $\max_{i=1,...,d} Re(\lambda_i(\mathbf{J})) < 0$. Only local neighborhood information is preserved by the system since the term $\partial \mathbf{x}(t)/\partial \mathbf{x}(0)$ would vanish exponentially fast over time, thereby making the nodes' representation insensitive to differences in the input graph.

**Non-dissipativity** is obtained when $Re(\lambda_i(\mathbf{J})) = 0$. Here, the system is stable, and the magnitude of $\partial \mathbf{x}(t)/\partial \mathbf{x}(0)$ is constant over time, and the input graph information is effectively propagated through the successive transformations into the final nodes' representations, addressing oversquashing.

# 3 SWAN: Space-Weight Antisymmetric GNN

We now turn to present **SWAN**, **s**pace and **w**eight **an**tisymmetric GNN. We analyze its theoretical behavior and show that it is both *global* (i.e., graph-wise) and *local* (i.e., node-wise) *non-dissipative*. As a consequence, one of the key features of SWAN is that it has a constant *global* information flow rate, unlike common diffusion GNNs. Therefore, SWAN should theoretically be able to propagate information between any nodes with a viable path in the graph, allowing to mitigate oversquashing. Figure 2 exemplifies the differences between dissipative, local non-dissipative, and global and local non-dissipative systems.

**Space and Weight Antisymmetry.** We define SWAN by including a new term that introduces antisymmetry in the aggregation function, i.e.,

$$\frac{\partial \mathbf{x}_u(t)}{\partial t} = \sigma\Big((\mathbf{W}-\mathbf{W}^\top)\mathbf{x}_u(t) + \Phi(\{\mathbf{x}_j\}_{j\in\mathcal{N}_u}, \mathbf{V}) + \beta\Psi(\{\mathbf{x}_j\}_{j\in\mathcal{N}_u}, \mathbf{Z})\Big). \quad (5)$$

where $\mathbf{W}$, $\mathbf{V}$, and $\mathbf{Z}$ are learnable weight matrices. $\Phi$ and $\Psi$ are permutation invariant neighborhood aggregation functions, where $\Psi$ performs antisymmetric aggregation. While $\Psi$ can assume various forms of antisymmetric aggregation functions with imaginary eigenvalues, and $\Phi$ can be any aggregation function, in this paper we explore the following family of parameterizations:

$$\Phi = \sum_{v\in\mathcal{N}_u}(\hat{\mathbf{A}}_{uv} + \hat{\mathbf{A}}_{vu})(\mathbf{V}-\mathbf{V}^\top)\mathbf{x}_v(t), \quad (6)$$

$$\Psi = \sum_{v\in\mathcal{N}_u}(\tilde{\mathbf{A}}_{uv} - \tilde{\mathbf{A}}_{vu})(\mathbf{Z}+\mathbf{Z}^\top)\mathbf{x}_v(t), \quad (7)$$

where $\tilde{\mathbf{A}}, \hat{\mathbf{A}} \in \mathbb{R}^{n\times n}$ are neighborhood aggregation matrices that can be pre-defined or learned. In our experiments, we consider two instances of Equations (6) and (7). The first, where $\tilde{\mathbf{A}}, \hat{\mathbf{A}}$ are pre-defined by the random walk and symmetrically normalized adjacency matrices, respectively. The second, learns $\tilde{\mathbf{A}}, \hat{\mathbf{A}}$, as described in Appendix A. Notably, as shown below, the parametrization of $\Psi$ and $\Phi$, described in Equations (6) and (7), ensures that SWAN's node- and graph-wise Jacobians have purely imaginary eigenvalues (Sections 3.1 and 3.2), which, together with the background in Section 2.3, shows SWAN's ability to be non-dissipative both locally and globally, leading to a *globally* constant information flow regardless of time, i.e., the model's depth. Lastly, we note that the general formulation of $\Phi$ and $\Psi$ provide a general design principle for introducing non-dissipativity as an inductive bias in any DE-GNN (see Appendix A.4).

## 3.1 Node-wise Analysis of SWAN

We reformulate Equation (5) to consider the formulation of $\Phi$ and $\Psi$ as in Equations (6) and (7), reading:

$$\frac{\partial \mathbf{x}_u(t)}{\partial t} = \sigma\Big((\mathbf{W}-\mathbf{W}^\top)\mathbf{x}_u(t) + \sum_{v\in\mathcal{N}_u}(\hat{\mathbf{A}}_{uv} + \hat{\mathbf{A}}_{vu})(\mathbf{V}-\mathbf{V}^\top)\mathbf{x}_v(t) + \beta\sum_{v\in\mathcal{N}_u}((\tilde{\mathbf{A}}_{uv} - \tilde{\mathbf{A}}_{vu})(\mathbf{Z}+\mathbf{Z}^\top)\mathbf{x}_v(t)\Big). \quad (8)$$

**SWAN is locally non-dissipative.** Using the sensitivity analysis in Section 2.3, we show that SWAN is stable and non-dissipative from a node perspective, i.e., it is *locally* non-dissipative. In this case, the Jacobian $\mathbf{J}(t) = \mathbf{M}_1\mathbf{M}_2$ of Equation (8) is composed of :

$$\mathbf{M}_1 = \mathrm{diag}\Big[\sigma'\Big((\mathbf{W}-\mathbf{W}^\top)\mathbf{x}_u(t) + \sum_{v\in\mathcal{N}_u}(\hat{\mathbf{A}}_{uv} + \hat{\mathbf{A}}_{vu})(\mathbf{V}-\mathbf{V}^\top)\mathbf{x}_v(t) + \beta\sum_{v\in\mathcal{N}_u}(\tilde{\mathbf{A}}_{uv} - \tilde{\mathbf{A}}_{vu})(\mathbf{Z}+\mathbf{Z}^\top)\mathbf{x}_v(t)\Big)\Big], \quad (9)$$

$$\mathbf{M}_2 = (\mathbf{W}-\mathbf{W}^\top) + (\hat{\mathbf{A}}_{uv} + \hat{\mathbf{A}}_{vu})(\mathbf{V}-\mathbf{V}^\top). \quad (10)$$

Following results from (Chang et al. 2019; Gravina, Bacciu, and Gallicchio 2023), only the eigenvalues of $\mathbf{M}_2$ determine the *local* stability and non-dissipativity of the system in Equation (5) for the final behavior of the model. Specifically, if the real part of all the eigenvalues of $\mathbf{M}_2$ is zero, then stability and non-dissipativity are achieved. We note that this is indeed the case in our system, since the real part of the eigenvalues of antisymmetric matrices is zero, and $\mathbf{M}_2$ is composed of a summation of two antisymmetric matrices.

## 3.2 Graph-wise Analysis of SWAN

While the node-perspective analysis is important because it shows the long-term memory capacity of individual nodes, as illustrated in Figure 2b, it overlooks *pairwise* node interactions, which are described by the properties of Equation (5) with respect to the graph, illustrated in Figure 2a. As we show below, our SWAN is globally non-dissipative. Hence,

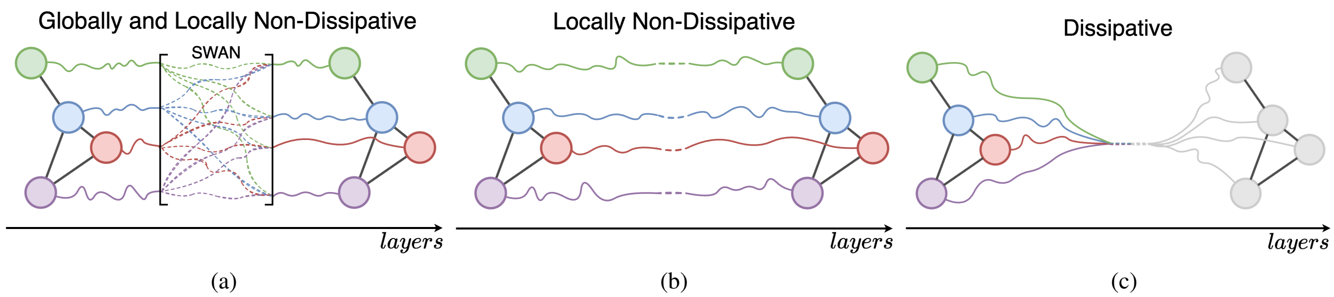



Figure 2: The difference between non-dissipative and dissipative behaviors. With global (i.e., graph-wise) and local (i.e., node-wise) non-dissipative behavior (a), information is propagated between any pair of nodes with a viable path in the graph. Therefore, such a behavior increases the long-range effectiveness of the model. A model exhibiting local non-dissipative behavior (b) enhances only the long-term memory capacity of individual nodes. A model demonstrating dissipative behavior (c) exhibits a convergence of node features toward non-informative values.

it allows the constant rate of information flow and node interactions, independently of time $t$, i.e., the network's depth. Therefore, we deem that SWAN's non-dissipativity behavior is beneficial in addressing oversquashing in MPNNs.

**SWAN is globally non-dissipative.** We start by reformulating Equation (8) from a node-wise formulation to a graph-perspective formulation, as follows:

$$\frac{\partial \mathbf{X}(t)}{\partial t} = \sigma\Big(\mathbf{X}(t)(\mathbf{W}-\mathbf{W}^\top) + (\hat{\mathbf{A}}+\hat{\mathbf{A}}^\top)\mathbf{X}(t)(\mathbf{V}-\mathbf{V}^\top) + \beta(\tilde{\mathbf{A}}-\tilde{\mathbf{A}}^\top)\mathbf{X}(t)(\mathbf{Z}+\mathbf{Z}^\top)\Big). \tag{11}$$

Following the sensitivity analysis introduced in Section 2.3, and applying the vectorization operator, the Jacobian of Equation (11), $\mathbf{J}(t) = \mathbf{M}_1\mathbf{M}_2$, writes as the multiplication of:

$$\mathbf{M}_1 = \mathrm{diag}\Big[\mathrm{vec}\Big(\sigma'\Big(\mathbf{IX}(\mathbf{W}-\mathbf{W}^\top) + (\hat{\mathbf{A}}+\hat{\mathbf{A}}^\top)\mathbf{X}(t)(\mathbf{V}-\mathbf{V}^\top) + \beta(\tilde{\mathbf{A}}-\tilde{\mathbf{A}}^\top)\mathbf{X}(t)(\mathbf{Z}+\mathbf{Z}^\top)\Big)\Big)\Big] \tag{12}$$

$$\mathbf{M}_2 = (\mathbf{W}-\mathbf{W}^\top)^\top \otimes \mathbf{I} + (\mathbf{V}-\mathbf{V}^\top)^\top \otimes (\hat{\mathbf{A}}+\hat{\mathbf{A}}^\top) + \beta(\mathbf{Z}+\mathbf{Z}^\top)^\top \otimes (\tilde{\mathbf{A}}-\tilde{\mathbf{A}}^\top), \tag{13}$$

where $\mathbf{I} \in \mathbb{R}^{n\times n}$ is the identity matrix, vec is the vectorization operator, and $\otimes$ is the Kronecker product (see Appendix H for more details). As in our node-wise analysis in Section 3.1, $\mathbf{M}_1$ in Equation (12) is a diagonal matrix. Thus, stability and non-dissipativity demand that $\mathbf{M}_2$ from Equation (13) has eigenvalues with real part equal to zero. We see that $\mathbf{M}_2$ satisfies this condition as it is composed of a summation of three antisymmetric matrices whose eigenvalues have a real part of zero. We conclude that SWAN (Equation (11)) is stable, and globally and locally non-dissipative. We now show that the properties of stability and global non-dissipativity allow the design of GNNs that can mitigate oversquashing.

**Theorem 3.1** (SWAN has a constant global information propagation rate)**.** *The information propagation rate among the graph nodes $\mathcal{V}$ is constant, $c$, independently of time $t$:*

$$\left\|\frac{\partial \mathrm{vec}(\mathbf{X}(\mathrm{t}))}{\partial \mathrm{vec}(\mathbf{X}(0))}\right\| = c, \tag{14}$$

*Proof.* Let us consider the following equation:

$$\frac{d}{dt}\left(\frac{\partial \mathbf{X}(t)}{\partial \mathbf{X}(0)}\right) = \frac{d}{dt}\left(\frac{\partial \mathrm{vec}(\mathbf{X}(\mathrm{t}))}{\partial \mathrm{vec}(\mathbf{X}(0))}\right) = \mathbf{J}(t)\frac{\partial \mathrm{vec}(\mathbf{X}(\mathrm{t}))}{\partial \mathrm{vec}(\mathbf{X}(0))}. \tag{15}$$

We follow the assumption in (Chang et al. 2019; Gravina, Bacciu, and Gallicchio 2023) that the Jacobian, $\mathbf{J}(t)$, does not change significantly over time, then we can apply results from autonomous differential equations and solve Equation (15):

$$\frac{\partial \mathrm{vec}(\mathbf{X}(\mathrm{t}))}{\partial \mathrm{vec}(\mathbf{X}(0))} = e^{t\mathbf{J}} = \mathbf{T}e^{t\mathbf{\Lambda}}\mathbf{T}^{-1} = \mathbf{T}\Big(\sum_{k=0}^{\infty}\frac{(t\mathbf{\Lambda})^k}{k!}\Big)\mathbf{T}^{-1}, \tag{16}$$

where $\mathbf{\Lambda}$ is the diagonal matrix whose non-zero entries contain the eigenvalues of $\mathbf{J}$, and $\mathbf{T}$ has the eigenvectors of $\mathbf{J}$ as columns, as in Section 2.3. As previously shown, it holds that $Re(\lambda_i(\mathbf{J}(t))) = 0$ for $i = 1, ..., d$, since the Jacobian is the result of the multiplication between a diagonal matrix and an antisymmetric matrix. Thus, the magnitude of $\partial\mathbf{X}(t)/\partial\mathbf{X}(0)$ is constant over time, allowing input features to propagate through the layers into the final node features. □

Theorem 3.1 states that regardless of time $t$ (equivalent to $\ell = t/\epsilon$ layers of SWAN, where $\epsilon$ is the step size), the information between nodes continues to propagate at a constant rate, unlike diffusion GNNs that exhibit an exponential decay in the propagation rate with respect to time, as shown below.

**Theorem 3.2** (Time Decaying Propagation in Diffusion GNNs)**.** *A diffusion GNN with Jacobian eigenvalues with magnitude $\mathbf{K}_{ii} = |\mathbf{\Lambda}_{ii}|$ , $i \in \{0,\ldots,n-1\}$ has an exponentially decaying information propagation rate, as follows:*

$$\left\|\frac{\partial \mathrm{vec}(\mathbf{X}(\mathrm{t}))}{\partial \mathrm{vec}(\mathbf{X}(0))}\right\| = \|e^{-t\mathbf{K}}\|, \tag{17}$$

See proof in Appendix B. This result means that for diffusion methods, as $t$ grows, deeper layers are not able to share new information between nodes as effectively as earlier layers of the network. On the contrary, our SWAN maintains the same effectiveness, independently of time, meaning it retains its ability to share information across nodes with the same effectiveness in each layer of the network, regardless of its depth, as illustrated in Figure 1 and Figure 2.

### 3.3 The Benefit of Spatial Antisymmetry

While oversquashig was not mathematically defined in (Alon and Yahav 2021), it was recently proposed in (Topping et al. 2022; Di Giovanni et al. 2023) to quantify the level, or lack of oversquashing, by measuring the sensitivity of node embedding after $\ell$ layers with respect to the input of another node $\|\partial \mathbf{x}_v(\ell)/\partial \mathbf{x}_u(0)\|$, which can be bounded as follows:

$$\left\| \frac{\partial \mathbf{x}_v(\ell)}{\partial \mathbf{x}_u(0)} \right\| \leq \underbrace{(c_\sigma wp)^\ell}_{model} \underbrace{(\mathbf{O}^\ell)_{vu}}_{topology}, \tag{18}$$

where $c_\sigma$ is the Lipschitz constant of the activation $\sigma$, $w$ is the maximal entry-value over all weight matrices, and $p$ is the embedding dimension. The term $\mathbf{O} = c_r\mathbf{I} + c_a\mathbf{A} \in \mathbb{R}^{n\times n}$ is the message passing matrix adopted by the MPNN, with $c_r$ and $c_a$ are the contributions of the residual and aggregation term. Oversquashing occurs if the right-hand side of Equation (18) is too small (Di Giovanni et al. 2023).

**The sensitivity of SWAN.** We present the sensitivity bound of SWAN, with its proof in Appendix C.

**Theorem 3.3** (SWAN sensitivity upper bound)**.** *Consider SWAN (Equation* (5)*), with $\ell$ layers, and $u, v \in \mathcal{V}$ two connected nodes of the graph. The sensitivity of $v$'s embedding after $\ell$ layers with respect to the input of node $u$ is*

$$\left\| \frac{\partial \mathbf{x}_v(\ell)}{\partial \mathbf{x}_u(0)} \right\| \leq \underbrace{(c_\sigma wp)^\ell}_{model} \underbrace{((c_r\mathbf{I} + c_a\mathbf{A} + \beta c_b \mathbf{S})^\ell)_{vu}}_{topology} \tag{19}$$

*with $c_\sigma$ the Lipschitz constant of non-linearity $\sigma$, $w$ is the maximal entry-value of all weight matrices, $p$ the embedding dimension, $\mathbf{A}$ the graph shift operator, $\mathbf{S} = (\tilde{\mathbf{A}} - \tilde{\mathbf{A}}^\top)$ the antisymmetric graph operator, and $c_r$ and $c_a$ the weighted contribution of the residual term and aggregation term.*

The result of Theorem 3.3 indicates that the added antisymmetric term $\Psi$ contributes to an increase in the measured upper bound. This result, together with the constant rate of information flow obtained from Theorem 3.1, holds the potential to theoretically mitigate oversquashing using SWAN.

## 4 Experiments

**Objectives.** We evaluate our SWAN and compare it with various methods. Specifically, we seek to address the following questions: (i) Can SWAN effectively propagate information to distant nodes? (ii) Can SWAN accurately predict graph properties related to long-range interactions? (iii) How does SWAN perform on real-world long-range benchmarks?

**Baselines.** We consider (i) MPNNs with linear complexity (similar to the complexity of our SWAN), i.e., GCN (Kipf and Welling 2017), GraphSAGE (Hamilton, Ying, and Leskovec 2017), GAT (Veličković et al. 2018), GatedGCN (Bresson and Laurent 2018), GIN (Xu et al. 2019), GINE (Hu et al. 2020), and GCNII (Chen et al. 2020). (ii) DE-GNNs such as DGC (Wang et al. 2021), GRAND (Chamberlain et al. 2021b), GraphCON (Rusch et al. 2022), and ADGN (Gravina, Bacciu, and Gallicchio 2023). (iii) Oversquashing designated methods such as Graph Transformers and Higher-Order GNNs, i.e., Transformer (Vaswani et al. 2017; Dwivedi and Bresson 2021), DIGL (Gasteiger, Weißenberger, and Günnemann 2019), MixHop (Abu-El-Haija et al. 2019), SAN (Kreuzer et al. 2021), GraphGPS (Rampášek et al. 2022), and DRew (Gutteridge et al. 2023).

**Experimental Details.** We discuss the datasets in our experiments in Appendix D, and the complexity of SWAN and provide runtimes in Appendix F. Our implementation uses PyTorch, and is available at https://github.com/gravins/SWAN.

**SWAN Variants.** In the following experiments, we leverage the general formulation of our method, discretized by forward Euler (see Appendix A.1), and explore two main variants of SWAN, each distinguished by the implementation of the aggregation terms $\hat{\mathbf{A}}$, $\tilde{\mathbf{A}}$ in the functions $\Psi$ and $\Phi$, as shown in Equations (6) and (7). Specifically, we consider (i) SWAN, which implements the aggregation terms using pre-defined operators, which are the symmetric normalized and random walk adjacency matrices, as described in Appendix A.2, and, (ii) SWAN-LEARN which utilizes the learned aggregation terms described in Appendix A.2. Both variants follow the form of Equations (6) and (7), in line with the theoretical analysis in Section 3, and in particular Theorem 3.1. We refer the reader to Appendix E.1 for a detailed description.

### 4.1 Graph Transfer

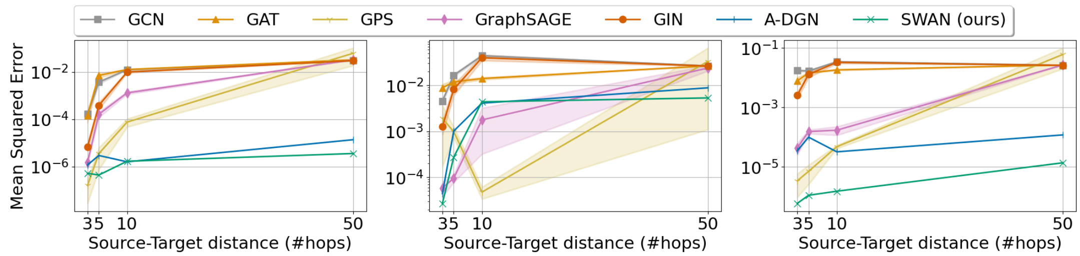
**Setup.** We consider a graph transfer task, where the goal is to transfer a label from a source to a target node, with a distance of $k$ hops. We note that this task can be effectively solved only by non-dissipative methods that preserve source information. We initialize nodes with a random valued feature and we assign values "1" and "0" to source and target nodes, respectively. We consider three graph distributions, i.e., line, ring, crossed-ring, as illustrated in Figure 4, with four different distances $k = \{3, 5, 10, 50\}$. Increasing $k$ increases the task complexity. In Appendix D.1 and E.2, we provide additional details about the dataset and the task.

**Results.** Figure 3 reports the results on graph transfer tasks. Overall, baseline methods struggle to accurately transfer the information through the graph, especially when the distance is high, i.e., $\#hops \geq 10$. Differently, non-dissipative methods, such as ADGN and SWAN, achieve low errors across all distances. Moreover, SWAN consistently outperforms ADGN, empirically supporting our theoretical findings that SWAN can better propagate information among distant nodes.

### 4.2 Graph Property Prediction

**Setup.** We consider the prediction of three graph properties - Diameter, Single-Source Shortest Paths (SSSP), and node Eccentricity on synthetic graphs (Corso et al. 2020), and follow the setup outlined in Gravina, Bacciu, and Gallicchio (2023), further discussed in Appendix D.2 and E.3. These three tasks rely on long-range interactions for shortest-path calculations, as in Bellman-Ford and Dijkstra's algorithms.

**Results.** In Table 1, we compare SWAN and SWAN-LEARN with MPNNs and DE-GNNs, indicating that SWAN consistently improves performance, where SWAN-LEARN yields the best results, with an improvement of up to 117% of the runner-up model. In Appendix G we also provide results for SWAN with layer-dependent weights showing improved performance. We note that similarly to the transfer task in




(a) Line (b) Ring (c) Crossed-Ring

Figure 3: Information transfer performance on (a) Line, (b) Ring, and (c) Crossed-Ring graphs. Non-dissipative methods like ADGN and SWAN allow for the accurate transfer of information.

| Model | Diameter | SSSP | Eccentricity |
|---|---|---|---|
| **MPNNs** | | | |
| GCN | $0.7424_{\pm 0.0466}$ | $0.9499_{\pm 9.18\cdot 10^{-5}}$ | $0.8468_{\pm 0.0028}$ |
| GAT | $0.8221_{\pm 0.0752}$ | $0.6951_{\pm 0.1499}$ | $0.7909_{\pm 0.0222}$ |
| GraphSAGE | $0.8645_{\pm 0.0401}$ | $0.2863_{\pm 0.1843}$ | $0.7863_{\pm 0.0207}$ |
| GIN | $0.6131_{\pm 0.0990}$ | $-0.5408_{\pm 0.4193}$ | $0.9504_{\pm 0.0007}$ |
| GCNII | $0.5287_{\pm 0.0570}$ | $-1.1329_{\pm 0.0135}$ | $0.7640_{\pm 0.0355}$ |
| **DE-GNNs** | | | |
| DGC | $0.6028_{\pm 0.0050}$ | $-0.1483_{\pm 0.0231}$ | $0.8261_{\pm 0.0032}$ |
| GRAND | $0.6715_{\pm 0.0490}$ | $-0.0942_{\pm 0.3897}$ | **$0.6602$**$_{\pm 0.1393}$ |
| GraphCON | **$0.0964$**$_{\pm 0.0620}$ | **$-1.3836$**$_{\pm 0.0092}$ | $0.6833_{\pm 0.0074}$ |
| ADGN | **$-0.5188$**$_{\pm 0.1812}$ | **$-3.2417$**$_{\pm 0.0751}$ | **$0.4296$**$_{\pm 0.1003}$ |
| **Ours** | | | |
| SWAN | **$-0.5249$**$_{\pm 0.0155}$ | **$-3.2370$**$_{\pm 0.0834}$ | **$0.4094$**$_{\pm 0.0764}$ |
| SWAN-LEARN | **$-0.5981$**$_{\pm 0.1145}$ | **$-3.5425$**$_{\pm 0.0830}$ | **$-0.0739$**$_{\pm 0.2190}$ |

Table 1: Mean test set $log_{10}(\text{MSE})$ and std averaged over 4 random weight initializations on the Graph Property Prediction tasks. The lower, the better. **First**, **second**, **third** and **fourth** best results for each task are color-coded.

| Model | Peptides-func AP ↑ | Peptides-struct MAE ↓ | Pascal VOC-sp F1 ↑ |
|---|---|---|---|
| **MPNNs** | | | |
| GCN | $59.30_{\pm 0.23}$ | $0.3496_{\pm 0.0013}$ | $12.68_{\pm 0.60}$ |
| GINE | $54.98_{\pm 0.79}$ | $0.3547_{\pm 0.0045}$ | $12.65_{\pm 0.76}$ |
| GCNII | $55.43_{\pm 0.78}$ | $0.3471_{\pm 0.0010}$ | $16.98_{\pm 0.80}$ |
| GatedGCN | $58.64_{\pm 0.77}$ | $0.3420_{\pm 0.0013}$ | $28.73_{\pm 2.19}$ |
| **Transformers** | | | |
| Transformer+LapPE | $63.26_{\pm 1.26}$ | **$0.2529$**$_{\pm 0.0016}$ | $26.94_{\pm 0.98}$ |
| SAN+LapPE | **$63.84$**$_{\pm 1.21}$ | $0.2683_{\pm 0.0043}$ | **$32.30$**$_{\pm 0.39}$ |
| GraphGPS+LapPE | **$65.35$**$_{\pm 0.41}$ | **$0.2500$**$_{\pm 0.0005}$ | **$37.48$**$_{\pm 1.09}$ |
| **DE-GNNs** | | | |
| GRAND | $57.89_{\pm 0.62}$ | $0.3418_{\pm 0.0015}$ | $19.18_{\pm 0.97}$ |
| GraphCON | $60.22_{\pm 0.68}$ | $0.2778_{\pm 0.0018}$ | $21.08_{\pm 0.91}$ |
| ADGN | $59.75_{\pm 0.44}$ | $0.2874_{\pm 0.0021}$ | $23.49_{\pm 0.54}$ |
| **Ours** | | | |
| SWAN | $63.13_{\pm 0.46}$ | $0.2571_{\pm 0.0018}$ | $27.96_{\pm 0.48}$ |
| SWAN-LEARN | **$67.51$**$_{\pm 0.39}$ | **$0.2485$**$_{\pm 0.0009}$ | **$31.92$**$_{\pm 2.50}$ |

Table 2: Performance of standard MPNNs, graph Transformers, DE-GNNs, and our SWAN across three LRGB tasks. Results are averaged over 3 weight initializations. The **first**, **second**, and **third** best results for each task are color-coded.

Section 4.1, solving Graph Property Prediction tasks necessitates capturing long-term dependencies. Hence, successful prediction requires the mitigation of oversquashing. For instance, in the eccentricity task, the goal is to calculate the maximal shortest path between node $u$ and all other nodes – a task requiring propagation from distant nodes.

## 4.3 Long-Range Graph Benchmark

**Setup.** We follow the settings from Dwivedi et al. (2022) on the Peptides-func, Peptides-struct, and PascalVOC-sp datasets, on which we elaborate in Appendix D.3 and E.4. As metrics, we consider the average precision (AP) for the Peptides-func, mean-absolute-error (MAE) for Peptides-struct, and the macro-weighted F1 score for PascalVOC-sp.
**Results.** Table 2 compares SWAN with various MPNNs, DE-GNNs, and Graph Transformers. In Appendix G, we also provide a comparison with multi-hop methods. Our results in Table 2 suggest the following: (i) SWAN achieves significantly better results than standard MPNNs such as GCN, GINE, or GCNII. For example, on Peptides-func, SWAN-LEARN achieves an average precision of 66.54, while GCN achieves a score of 59.30. (ii) Compared with Transformers, which are of complexity $\mathcal{O}(|V|^2)$, our SWAN achieves better performance while remaining with a linear complexity of $\mathcal{O}(|\mathcal{V}| + |\mathcal{E}|)$. (iii) Among its class of DE-GNNs, our SWAN offers overall better performance.

## 4.4 Ablation Study

We now study the contribution of global (graph-wise) and local (node-wise) non-dissipativity, and spatial antisymmetry.
**The importance of global and local non-dissipativity.** To verify the contribution of the global and local non-dissipativity in SWAN, we evaluate the performance of several variants that can deviate from being globally and locally non-dissipative, although in a bounded manner, as discussed in Appendix E.1. Specifically, we consider the non-enforced (NE) variants of SWAN and SWAN-LEARN, which use an unconstrained weight matrix $\mathbf{V}$, rather than forcing it to be antisymmetric as in SWAN. These two additional variants are called SWAN-NE and SWAN-LEARN-NE, respectively.

| Dataset ↓ / Model → | SWAN$_{\beta=0}$ | SWAN-NE | SWAN-NE-LEARN | SWAN | SWAN-LEARN |
|---|---|---|---|---|---|
| **Diam.** ($log_{10}$(MSE)↓) | -0.3882$_{\pm 0.0610}$ | **-0.5497**$_{\pm 0.0766}$ | **-0.5631**$_{\pm 0.0694}$ | -0.5249$_{\pm 0.0155}$ | **-0.5981**$_{\pm 0.1145}$ |
| **SSSP** ($log_{10}$(MSE)↓) | -3.2061$_{\pm 0.0416}$ | -3.1913$_{\pm 0.0762}$ | **-3.5296**$_{\pm 0.0831}$ | **-3.2370**$_{\pm 0.0834}$ | **-3.5425**$_{\pm 0.0830}$ |
| **Ecc.** ($log_{10}$(MSE)↓) | 0.5573$_{\pm 0.0247}$ | **0.3792**$_{\pm 0.1514}$ | **0.1317**$_{\pm 0.1253}$ | 0.4094$_{\pm 0.0764}$ | **-0.0739**$_{\pm 0.219}$ |
| **Peptides-func** (AP ↑) | 61.95$_{\pm 00.67}$ | 61.19$_{\pm 0.37}$ | **62.49**$_{\pm 0.51}$ | **63.13**$_{\pm 0.46}$ | **67.51**$_{\pm 0.39}$ |
| **Peptides-struct** (MAE ↓) | 0.2703$_{\pm 0.0023}$ | 0.2672$_{\pm 0.0012}$ | **0.2606**$_{\pm 0.0007}$ | **0.2571**$_{\pm 0.0018}$ | **0.2485**$_{\pm 0.0009}$ |

Table 3: Performance of different versions of SWAN on Graph Property Prediction and LRGB tasks. Results are averaged over 4 random weight initializations on the Graph Property Prediction, while over 3 on the LRGB. The **first**, **second**, and **third** best results for each task are color-coded. Global and Local Non-Dissipative variants achieve highest performance.

Table 3 shows the performance of the SWAN on the Property Prediction and LRGB tasks. The highest performance on synthetic and real-world problems was achieved with SWAN-LEARN, which is globally and locally non-dissipative.

**The benefit of spatial antisymmetry.** Table 3 highlights the advantages of the spatial antisymmetry in Equation (5). By setting $\beta = 0$, we obtain a model with antisymmetry solely in the weight space, exhibiting only a local non-dissipative behavior. Our results indicate a noteworthy performance improvement when spatial antisymmetry is employed. This is further supported by the improved performance of -NE versions of SWAN, which do not guarantee both global and local non-dissipative behavior, compared to SWAN$_{\beta=0}$.

# 5 Related Work

**Graph Neural Networks based on Differential Equations.** Adopting the interpretation of convolutional neural networks (CNNs) as discretization of ODEs and PDEs (Ruthotto and Haber 2020; Chen et al. 2018) to GNNs, works like CGNN (Xhonneux, Qu, and Tang 2020), GCDE (Poli et al. 2019), GODE (Zhuang et al. 2020), GRAND (Chamberlain et al. 2021b), PDE-GCN$_D$ (Eliasof, Haber, and Treister 2021), DGC (Wang et al. 2021), and others, propose interpreting GNN layers as discretization steps of the heat equation. This allows controlling the diffusion (smoothing) in the network and understanding the problem of oversmoothing (Nt and Maehara 2019; Oono and Suzuki 2020; Cai and Wang 2020) in GNNs. Differently, Choromanski et al. (2022) propose an architecture with an attention mechanism based on the heat diffusion kernel. Other architectures like PDE-GCN$_M$ (Eliasof, Haber, and Treister 2021) and GraphCON (Rusch et al. 2022) propose to mix diffusion and oscillatory processes as a feature energy preservation mechanism. Other recent works proposed mechanisms such as anti-symmetry (Gravina, Bacciu, and Gallicchio 2023), reaction-diffusion-based dynamics (Wang et al. 2022; Choi et al. 2023), and advection-reaction-diffusion (Eliasof, Haber, and Treister 2023). However, most of the aforementioned works are diffusion-based DE-GNNs, limited in modeling long-range interactions, while our SWAN is a DE-GNN with a constant information flow, addressing oversquashing in graphs. While the aforementioned works focus on the spatial aggregation term of DE-GNNs, the temporal domain of DE-GNNs has also been studied in (Eliasof et al. 2024; Gravina et al. 2024a; Kang et al. 2024; Gravina et al. 2024b). A review of these methods is given in Han et al. (2023).

**Oversquashing in MPNNs.** Oversquashing in MPNNs, which hampers information transfer across distant nodes (Alon and Yahav 2021), has prompted various mitigation strategies. *Graph rewiring* methods like SDRF (Topping et al. 2022) densify graphs as a preprocessing step, while approaches such as GRAND (Chamberlain et al. 2021b), BLEND (Chamberlain et al. 2021a), and DRew (Gutteridge et al. 2023) dynamically adjust connectivity based on node features. Transformer-based models (Dwivedi and Bresson 2021; Rampášek et al. 2022) bypass oversquashing with all-to-all message passing. Another direction uses *non-local dynamics* to enable dense communication, as in FLODE (Maskey et al. 2023), which leverages fractional graph shifts, QDC (Markovich 2023) with quantum diffusion kernels, and G2TN (Toth et al. 2022), which captures diffusion paths. While effective, these methods often increase computational complexity due to dense propagation operators. For further discussion, see (Shi et al. 2023). We note that, long-range interactions have also been explored in sequential models (Hochreiter and Schmidhuber 1997; Gu, Goel, and Re 2022).

**Non-dissipative Systems.** Non-dissipative Systems are characterized by the absence of energy dissipation, thus playing a crucial role in various domains such as physics (Goldstein, Poole, and Safko 2002), engineering (Ogata 2010), and machine learning, where such systems were shown to effectively model information flow. For example, the effectiveness of a non-dissipativity was demonstrated to enhance the power of reservoir computing (Gallicchio 2024) and recurrent neural networks (Chang et al. 2019). In the context of GNNs, it was shown in (Gravina, Bacciu, and Gallicchio 2023) that *local* non-dissipativity is beneficial for long interaction modeling.

# 6 Summary

In this work, we have presented SWAN (Space-Weight ANtisymmetry), a novel Differential Equation GNN (DE-GNN) designed to address the oversquashing problem. SWAN incorporates both global (i.e., graph-wise) and local (i.e., node-wise) non-dissipative properties through space and weight antisymmetric parameterization, and provides a general design principle for introducing non-dissipativity as an inductive bias in any DE-GNN. Our theoretical and experimental results emphasize the significance of global and local non-dissipativity, achieved by SWAN. For such reasons, we believe SWAN represents a significant step forward in addressing oversquashing in GNNs.

## Acknowledgments

The work has been partially supported by EU-EIC EMERGE (Grant No. 101070918). ME is funded by the Blavatnik-Cambridge fellowship, the Accelerate Programme for Scientific Discovery, and the Maths4DL EPSRC Programme.

## References

Abu-El-Haija, S.; Perozzi, B.; Kapoor, A.; Alipourfard, N.; Lerman, K.; Harutyunyan, H.; Ver Steeg, G.; and Galstyan, A. 2019. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *international conference on machine learning*, 21–29. PMLR.

Alon, U.; and Yahav, E. 2021. On the Bottleneck of Graph Neural Networks and its Practical Implications. In *International Conference on Learning Representations*.

Ascher, U. M.; and Petzold, L. R. 1998. *Computer methods for ordinary differential equations and differential-algebraic equations*, volume 61. Siam.

Bresson, X.; and Laurent, T. 2018. Residual Gated Graph ConvNets. arXiv:1711.07553.

Cai, C.; Hy, T. S.; Yu, R.; and Wang, Y. 2023. On the connection between mpnn and graph transformer. *arXiv preprint arXiv:2301.11956*.

Cai, C.; and Wang, Y. 2020. A note on over-smoothing for graph neural networks. *arXiv preprint arXiv:2006.13318*.

Chamberlain, B.; Rowbottom, J.; Eynard, D.; Di Giovanni, F.; Dong, X.; and Bronstein, M. 2021a. Beltrami Flow and Neural Diffusion on Graphs. In *Advances in Neural Information Processing Systems*, volume 34.

Chamberlain, B. P.; Rowbottom, J.; Gorinova, M.; Webb, S.; Rossi, E.; and Bronstein, M. M. 2021b. GRAND: Graph neural diffusion. In *International Conference on Machine Learning*.

Chang, B.; Chen, M.; Haber, E.; and Chi, E. H. 2019. AntisymmetricRNN: A Dynamical System View on Recurrent Neural Networks. In *International Conference on Learning Representations*.

Chen, M.; Wei, Z.; Huang, Z.; Ding, B.; and Li, Y. 2020. Simple and Deep Graph Convolutional Networks. In *Proceedings of the 37th International Conference on Machine Learning*.

Chen, T. Q.; Rubanova, Y.; Bettencourt, J.; and Duvenaud, D. K. 2018. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*.

Choi, J.; Hong, S.; Park, N.; and Cho, S.-B. 2023. GREAD: Graph Neural Reaction-Diffusion Networks. In *ICML*.

Choromanski, K.; Lin, H.; Chen, H.; Zhang, T.; Sehanobish, A.; Likhosherstov, V.; Parker-Holder, J.; Sarlos, T.; Weller, A.; and Weingarten, T. 2022. From block-Toeplitz matrices to differential equations on graphs: towards a general theory for scalable masked Transformers. In *Proceedings of the 39th International Conference on Machine Learning*.

Corso, G.; Cavalleri, L.; Beaini, D.; Liò, P.; and Veličković, P. 2020. Principal Neighbourhood Aggregation for Graph Nets. In *Advances in Neural Information Processing Systems*.

Di Giovanni, F.; Giusti, L.; Barbero, F.; Luise, G.; Liò, P.; and Bronstein, M. 2023. On over-squashing in message passing neural networks: the impact of width, depth, and topology. In *International Conference on Machine Learning*.

Dwivedi, V. P.; and Bresson, X. 2021. A Generalization of Transformer Networks to Graphs. *AAAI Workshop on Deep Learning on Graphs: Methods and Applications*.

Dwivedi, V. P.; Rampášek, L.; Galkin, M.; Parviz, A.; Wolf, G.; Luu, A. T.; and Beaini, D. 2022. Long Range Graph Benchmark. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Eliasof, M.; Haber, E.; and Treister, E. 2021. PDE-GCN: Novel architectures for graph neural networks motivated by partial differential equations. *Advances in Neural Information Processing Systems*, 34: 3836–3849.

Eliasof, M.; Haber, E.; and Treister, E. 2023. ADR-GNN: Advection-diffusion-reaction graph neural networks. *arXiv preprint arXiv:2307.16092*.

Eliasof, M.; Haber, E.; Treister, E.; and Schönlieb, C.-B. 2024. On The Temporal Domain of Differential Equation Inspired Graph Neural Networks. *arXiv preprint arXiv:2401.11074*.

Gallicchio, C. 2024. Euler State Networks: Non-dissipative Reservoir Computing. *Neurocomputing*, 579: 127411.

Gasteiger, J.; Weißenberger, S.; and Günnemann, S. 2019. Diffusion Improves Graph Learning. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; and Dahl, G. E. 2017. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning*.

Goldstein, H.; Poole, C.; and Safko, J. 2002. Classical mechanics.

Gravina, A.; Bacciu, D.; and Gallicchio, C. 2023. Anti-Symmetric DGN: a stable architecture for Deep Graph Networks. In *The Eleventh International Conference on Learning Representations*.

Gravina, A.; Lovisotto, G.; Gallicchio, C.; Bacciu, D.; and Grohnfeldt, C. 2024a. Long Range Propagation on Continuous-Time Dynamic Graphs. In Salakhutdinov, R.; Kolter, Z.; Heller, K.; Weller, A.; Oliver, N.; Scarlett, J.; and Berkenkamp, F., eds., *Proceedings of the 41st International Conference on Machine Learning*.

Gravina, A.; Zambon, D.; Bacciu, D.; and Alippi, C. 2024b. Temporal Graph ODEs for Irregularly-Sampled Time Series. In Larson, K., ed., *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*.

Gu, A.; Goel, K.; and Re, C. 2022. Efficiently Modeling Long Sequences with Structured State Spaces. In *International Conference on Learning Representations*.

Gutteridge, B.; Dong, X.; Bronstein, M. M.; and Di Giovanni, F. 2023. Drew: Dynamically rewired message passing with delay. In *International Conference on Machine Learning*.

Haber, E.; Ruthotto, L.; Holtham, E.; and Jun, S.-H. 2018. Learning Across Scales - Multiscale Methods for Convolution Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Hamilton, W. L.; Ying, R.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. In *NIPS*.

Han, A.; Shi, D.; Lin, L.; and Gao, J. 2023. From continuous dynamics to graph neural networks: Neural diffusion and beyond. *arXiv preprint arXiv:2310.10121*.

Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Computation*, 9(8): 1735–1780.

Hu, W.; Liu, B.; Gome, J.; Zitnik, M.; Liang, P.; Pande, V.; and Leskovec, J. 2020. Strategies for Pre-training Graph Neural Networks. In *International Conference on Learning Representations*.

Kang, Q.; Zhao, K.; Ding, Q.; Ji, F.; Li, X.; Liang, W.; Song, Y.; and Tay, W. P. 2024. Unleashing the Potential of Fractional Calculus in Graph Neural Networks with FROND. In *The Twelfth International Conference on Learning Representations*.

Karhadkar, K.; Banerjee, P. K.; and Montufar, G. 2023. FoSR: First-order spectral rewiring for addressing oversquashing in GNNs. In *The Eleventh International Conference on Learning Representations*.

Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations (ICLR)*.

Kreuzer, D.; Beaini, D.; Hamilton, W.; Létourneau, V.; and Tossou, P. 2021. Rethinking graph transformers with spectral attention. *Advances in Neural Information Processing Systems*, 34.

Markovich, T. 2023. QDC: Quantum Diffusion Convolution Kernels on Graphs. arXiv:2307.11234.

Maskey, S.; Paolino, R.; Bacho, A.; and Kutyniok, G. 2023. A Fractional Graph Laplacian Approach to Oversmoothing. In *Conference on Neural Information Processing Systems*.

Nt, H.; and Maehara, T. 2019. Revisiting graph neural networks: All we have is low-pass filters. *arXiv preprint arXiv:1905.09550*.

Ogata, K. 2010. *Modern control engineering fifth edition*.

Oono, K.; and Suzuki, T. 2020. Graph Neural Networks Exponentially Lose Expressive Power for Node Classification. In *International Conference on Learning Representations*.

Poli, M.; Massaroli, S.; Park, J.; Yamashita, A.; Asama, H.; and Park, J. 2019. Graph neural ordinary differential equations. *arXiv preprint arXiv:1911.07532*.

Rampášek, L.; Galkin, M.; Dwivedi, V. P.; Luu, A. T.; Wolf, G.; and Beaini, D. 2022. Recipe for a General, Powerful, Scalable Graph Transformer. *Advances in Neural Information Processing Systems*, 35.

Rusch, T. K.; Chamberlain, B.; Rowbottom, J.; Mishra, S.; and Bronstein, M. 2022. Graph-coupled oscillator networks. In *International Conference on Machine Learning*.

Ruthotto, L.; and Haber, E. 2020. Deep neural networks motivated by partial differential equations. *Journal of Mathematical Imaging and Vision*, 62: 352–364.

Shi, D.; Han, A.; Lin, L.; Guo, Y.; and Gao, J. 2023. Exposition on over-squashing problem on GNNs: Current Methods, Benchmarks and Challenges. *arXiv preprint arXiv:2311.07073*.

Topping, J.; Giovanni, F. D.; Chamberlain, B. P.; Dong, X.; and Bronstein, M. M. 2022. Understanding over-squashing and bottlenecks on graphs via curvature. In *International Conference on Learning Representations*.

Toth, C.; Lee, D.; Hacker, C.; and Oberhauser, H. 2022. Capturing Graphs with Hypo-Elliptic Diffusions. In *Advances in Neural Information Processing Systems*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. *International Conference on Learning Representations*.

Wang, Y.; Wang, Y.; Yang, J.; and Lin, Z. 2021. Dissecting the Diffusion Process in Linear Graph Convolutional Networks. In *Proceedings of the 35th Conference on Neural Information Processing Systems*.

Wang, Y.; Yi, K.; Liu, X.; Wang, Y. G.; and Jin, S. 2022. ACMP: Allen-Cahn Message Passing for Graph Neural Networks with Particle Phase Transition. *arXiv preprint arXiv:2206.05437*.

Xhonneux, L.-P.; Qu, M.; and Tang, J. 2020. Continuous Graph Neural Networks. In *Proceedings of the 37th International Conference on Machine Learning*.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How Powerful are Graph Neural Networks? In *International Conference on Learning Representations*.

Zhuang, J.; Dvornek, N.; Li, X.; and Duncan, J. S. 2020. Ordinary differential equations on graph networks.