# HYBRID DIFFUSION–TRANSPORT GNNS ON DIRECTED GRAPHS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Directionality is intrinsic to many graph-structured systems, yet the resulting asymmetric operators complicate the spectral foundations of classical graph neural networks. Hermitian Laplacians address this limitation by encoding edge direction through complex-valued phase information while retaining real eigenvalues and a unitary eigenbasis. We reinterpret this construction through a diffusion–transport decomposition, in which the real component captures symmetric diffusion and the imaginary component represents directional transport. Building on this interpretation, we introduce ArcShift, a first-order approximation of the magnetic phase that yields an explicit diffusion–transport propagation operator. We further propose ArcShift-CVNN, a complex-valued extension that jointly transforms the real and imaginary feature representations. Experiments on multiple directed graph benchmarks evaluate the proposed models against the exact magnetic formulation and representative graph-learning baselines in terms of node-classification performance and computational behavior. Overall, ArcShift provides an efficiency- and interpretability-oriented directional propagation operator while maintaining competitive predictive performance.

## 1 INTRODUCTION

Graph Neural Networks (GNNs) have emerged as a powerful framework for learning from graph-structured data, with applications including node classification, link prediction, graph classification, recommendation systems, and scientific domains (Zhou et al., 2020; Gao et al., 2023; Ponzi & Napoli, 2025). Many GNNs build upon graph Laplacian operators, which provide principled notions of graph frequency, smoothness, and convolution (Shuman et al., 2013; Bruna et al., 2013; Defferrard et al., 2016; Kipf & Welling, 2016).

Classical spectral graph signal processing is primarily formulated for undirected graphs, whose Laplacians are symmetric positive semidefinite and therefore admit orthonormal eigenbases (Shuman et al., 2013). However, many real-world systems are inherently directional, including citation, traffic, communication, and transaction networks (Sun et al., 2026). Ignoring edge orientation can therefore discard structural information important for learning on directed graphs. For directed graphs, asymmetric operators generally lose these properties, making spectral filtering more challenging. Hermitian Laplacian-based approaches address this problem by encoding edge direction through complex-valued phase factors while retaining a Hermitian operator with real eigenvalues and an orthonormal eigenbasis (Furutani et al., 2019; Zhang et al., 2021). However, the respective roles of symmetric propagation and directional transport remain implicit within the magnetic phase.

In this paper, we reinterpret Hermitian directed-graph propagation through an explicit diffusion–transport decomposition. Motivated by the magnetic phase, we introduce ArcShift, a first-order approximation based on

$$e^{i2\pi q\Phi} \approx 1 + i2\pi q\Phi,$$

where $\Phi$ denotes the directional imbalance matrix. The resulting operator couples a real symmetric diffusion component with an imaginary antisymmetric transport component, making the role of edge direction explicit.

ArcShift is designed as an efficiency- and interpretability-oriented directional propagation operator rather than primarily as an accuracy-oriented method. Its first-order formulation replaces the non-



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

linear magnetic phase representation with explicit diffusion and transport components, avoiding explicit trigonometric phase evaluation during operator construction. Because practical computational benefits depend on implementation and graph structure, we evaluate runtime empirically rather than infer efficiency from the approximation alone.

We further introduce *ArcShift-CVNN*, a complex-valued neural extension that jointly learns from the real and imaginary feature representations, following established principles of complex-valued neural networks (Hirose, 2011; Lee et al., 2022). The explicit diffusion–transport formulation further enables an adaptive variant in which the transport strength is learned jointly with the network parameters. We evaluate the proposed models on directed node-classification benchmarks against representative graph-learning baselines, examining predictive performance, computational behavior, transport sensitivity, and adaptive transport scaling.

We make three main contributions: (1) **Diffusion–transport formulation and ArcShift.** We reinterpret Hermitian directed-graph propagation through an explicit decomposition into symmetric diffusion and antisymmetric directional transport. Building on this interpretation, we introduce ArcShift, a first-order approximation of the magnetic phase that yields a Hermitian diffusion–transport operator with directly controllable transport strength. (2) **Complex-valued and adaptive ArcShift.** We introduce ArcShift-CVNN, which jointly transforms the real and imaginary representations using complex-valued learnable parameters. The explicit transport formulation further enables an adaptive variant in which the transport strength is learned jointly with the network parameters rather than selected as a fixed hyperparameter. (3) **Empirical and computational evaluation.** We evaluate the proposed models against representative graph-learning baselines on directed node classification, examining predictive performance, computational behavior, transport sensitivity, and the model-selection cost associated with transport-parameter search.

## 2 Related Work

Spectral learning on directed graphs has increasingly employed Hermitian or magnetic Laplacian-based operators to encode edge direction while retaining a well-defined spectral representation. MagNet (Zhang et al., 2021) uses the magnetic Laplacian to incorporate direction through complex-valued phase factors, enabling spectral graph convolution on directed graphs. MSGNN (He et al., 2022) extends this framework to signed directed graphs through the Magnetic Signed Laplacian, while SigMaNet (Fiorini et al., 2023) introduces a parameter-free Sign-Magnetic Laplacian that accommodates directed edge weights unrestricted in sign and magnitude. Framelet-MagNet (Lin & Gao, 2023) further develops magnetic spectral learning through a multiscale framelet decomposition.

Recent work has also explored alternative spectral representations for directed graphs. Multi-q MagPE (Huang et al., 2025) incorporates multiple magnetic phases for directed positional encoding, while GIST (Rigotti et al., 2026) and Xie & Wang (2026) develop spectral representations that avoid explicit eigendecomposition. More closely related to our diffusion–transport interpretation, Chan et al. (2026) establish a correspondence between the directed graph Laplacian and diffusion–advection dynamics and develop separate diffusive and advective kernels for directed graph signal processing.

Beyond magnetic Laplacian-based methods, DLGNet (Fiorini et al., 2024) constructs a Hermitian positive-semidefinite Directed Line Graph Laplacian for directed hypergraphs, while HoloNet (Koke & Cremers, 2024) extends spectral graph convolution to general directed operators through the holomorphic functional calculus. A2DUG (Maekawa et al., 2023) instead combines multiple directed and undirected graph representations for scalable node classification.

In contrast to these approaches, our work focuses specifically on simplifying the magnetic propagation mechanism through a first-order approximation of the complex magnetic phase. ArcShift makes the resulting diffusion–transport structure explicit while retaining a Hermitian direction-aware propagation operator.

# 3 HYBRID DIFFUSION–TRANSPORT FRAMEWORK

## 3.1 SPECTRAL FOUNDATION

We begin with the classical spectral framework for undirected graphs (Shuman et al., 2013) and extend it to directed graphs through the Hermitian Laplacian (Furutani et al., 2019). Let $G = (V, E)$ be a directed graph with weighted adjacency matrix $A = [w_{ij}]$. For an undirected graph, the Laplacian $L = D - A$ is symmetric and admits the eigendecomposition $L = U\Lambda U^*$. The corresponding graph Fourier transform is $\hat{f} = U^* f$, and spectral convolution can be written as

$$f_{\text{out}} = U\hat{S}U^* f_{\text{in}},$$

where $\hat{S}$ denotes a spectral filter (Shuman et al., 2013; Bruna et al., 2013).

For directed graphs, $A \neq A^\top$, and the corresponding Laplacian generally loses symmetry and an orthonormal spectral basis. Magnetic graph operators address this issue by encoding edge direction through complex-valued phase factors (Furutani et al., 2019; Zhang et al., 2021).

**Definition 1** (Hermitian Laplacian)**.** *Let*

$$A^{(s)} = \frac{1}{2}(A + A^\top), \qquad D_{ii} = \sum_j A^{(s)}_{ij}, \qquad \Phi = A - A^\top, \qquad \phi_{ij} = w_{ij} - w_{ji},$$

*where $A^{(s)}$ is the symmetric adjacency matrix, $D$ is its degree matrix, $\Phi$ is the directional imbalance matrix, and $\phi_{ij}$ denotes its $(i, j)$-th entry. The Hermitian Laplacian is*

$$\mathcal{L}_q = D - A^{(s)} \odot e^{i2\pi q\Phi}, \tag{1}$$

where $q$ is the magnetic phase parameter and $\odot$ denotes the Hadamard product. The term $e^{i2\pi q\phi_{ij}} \in U(1)$ is a complex unit-modulus phase factor, where $U(1) = \{e^{i\theta} : \theta \in \mathbb{R}\}$ denotes the unitary group of degree one. Its phase angle $2\pi q\phi_{ij}$ encodes the directional imbalance of edge $(i, j)$, while its unit magnitude preserves the edge magnitude represented by $A^{(s)}_{ij}$. Because $\phi_{ji} = -\phi_{ij}$, the reverse direction carries the complex-conjugate phase,

$$e^{i2\pi q\phi_{ji}} = e^{-i2\pi q\phi_{ij}} = \overline{e^{i2\pi q\phi_{ij}}},$$

which ensures the Hermitian structure of the resulting operator.

Since $\mathcal{L}_q = \mathcal{L}_q^*$, it admits a unitary eigendecomposition

$$\mathcal{L}_q = U_q\Lambda_q U_q^*,$$

with real eigenvalues.

Figure 1 illustrates the directional encoding induced by the magnetic phase. When $w_{ij} = w_{ji}$, the directional imbalance is zero and the phase factor reduces to 1. When $w_{ij} \neq w_{ji}$, the sign of $w_{ij} - w_{ji}$ determines the orientation of the phase rotation in the complex plane, while $q$ controls its angular scaling.

## 3.2 DIFFUSION–TRANSPORT DECOMPOSITION

The Hermitian Laplacian can be interpreted as a direction-aware spectral operator. We derive a diffusion–transport decomposition directly from its magnetic phase representation by applying Euler's identity to Equation 1.

$$e^{i2\pi q\Phi} = \cos(2\pi q\Phi) + i\sin(2\pi q\Phi), \tag{2}$$

$$\mathcal{L}_q = \underbrace{D - A^{(s)} \odot \cos(2\pi q\Phi)}_{L_D} - i\underbrace{\left(A^{(s)} \odot \sin(2\pi q\Phi)\right)}_{L_T}. \tag{3}$$

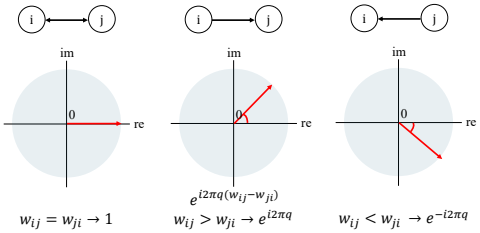


Figure 1: Illustration of the magnetic phase factor $e^{i2\pi q(w_{ij}-w_{ji})}$. Balanced interactions yield a zero phase angle, whereas opposite directional imbalances induce phase rotations with opposite orientations in the complex plane.

**Theorem 1** (Diffusion–Transport Decomposition)**.** *The Hermitian Laplacian admits the decomposition*

$$\mathcal{L}_q = L_D - iL_T, \tag{4}$$

*where $L_D$ is the symmetric component and $L_T$ captures the directional contribution induced by the magnetic phase.*

The proof is provided in Appendix 6.2.1. We interpret $L_D$ and $L_T$ as diffusion and directional transport components, respectively. This interpretation has a structural analogy to the continuum magnetic Laplacian (Möller, 2025). For a scalar field $f$ and vector potential $\mathcal{A}$,

$$(\nabla - i\mathcal{A})^2 f = \Delta f - 2i\mathcal{A} \cdot \nabla f - i(\nabla \cdot \mathcal{A})f - |\mathcal{A}|^2 f,$$

where $\Delta f$ is the second-order Laplacian term, while $\mathcal{A} \cdot \nabla f$ has a first-order directional transport structure. This analogy is used only as an interpretation. In contrast to diffusion–advection formulations derived from directed graph Laplacians (Chan et al., 2026), our diffusion–transport decomposition follows algebraically from the magnetic phase of the Hermitian Laplacian and motivates the first-order ArcShift approximation developed next.

### 3.3 ArcShift Approximation

Motivated by the diffusion–transport decomposition, we simplify the magnetic phase using a first-order Taylor approximation. For sufficiently small phase angles $2\pi q\Phi$,

$$e^{i2\pi q\Phi} \approx 1 + i2\pi q\Phi. \tag{5}$$

Substituting Equation 5 into Equation 1 gives

$$\mathcal{L}_q \approx D - A^{(s)} - i\,2\pi q \left(A^{(s)} \odot \Phi\right). \tag{6}$$

For graph propagation, we define the normalized symmetric adjacency, $A^n$, and directional transport, $T^n$, operators as

$$A^n = D^{-1/2} A^{(s)} D^{-1/2}, \qquad T^n = A^n \odot \Phi. \tag{7}$$

We employ GCN-style renormalization (Kipf & Welling, 2016) for the diffusion component,

$$\hat{A}^n = \tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2}, \qquad \tilde{A} = A^{(s)} + I, \qquad \tilde{D}_{ii} = \sum_j \tilde{A}_{ij}. \tag{8}$$

Equation 6 shows that the leading directional correction is linear in the imbalance $\Phi$, with $2\pi q$ arising from the magnetic phase scaling. ArcShift retains this first-order directional structure but separates its strength from the magnetic phase parameterization. The unscaled operator is obtained by using $T^n = A^n \odot \Phi$.

**Theorem 2** (ArcShift Operator)**.** *The unscaled ArcShift propagation operator is*

$$\text{ArcShift}^n = \hat{A}^n - iT^n. \tag{9}$$

The proof is provided in Appendix 6.2.2. Given node features $X$, a single ArcShift graph convolution is

$$Z = \mathrm{ArcShift}^n X\Theta. \tag{10}$$

The real component performs symmetric graph propagation, while the imaginary component introduces directional transport through edge imbalance. ArcShift therefore provides a simplified first-order representation of magnetic directed-graph propagation with an explicit transport component.

### 3.4 Approximation Error

The ArcShift formulation is motivated by truncating the magnetic phase expansion after its first-order term. The resulting approximation error is characterized as follows.

**Theorem 3** (First-Order Approximation Error)**.** *For each entry $(i, j)$, the first-order approximation of the magnetic phase has second-order truncation error:*

$$e^{i2\pi q\phi_{ij}} - (1 + i2\pi q\phi_{ij}) = \mathcal{O}\big((2\pi q\phi_{ij})^2\big)\,. \tag{11}$$

The proof is provided in Appendix 6.2.3. Thus, the first-order approximation retains the zeroth-order symmetric term and first-order directional term while discarding higher-order phase corrections. Its truncation error therefore decreases quadratically as the entrywise phase angles $2\pi q\phi_{ij}$ approach zero. This operator-level approximation error does not directly determine predictive performance because the resulting representation is embedded within a trainable nonlinear architecture. MagNet retains the exact magnetic phase, whereas ArcShift uses the first-order directional structure derived from it. We therefore evaluate the empirical effect of this approximation on node classification in Section 4.

### 3.5 Spectral and Spatial Properties

The Hermitian Laplacian provides the spectral foundation of the proposed framework. Since $\mathcal{L}_q$ is Hermitian and positive semidefinite, it admits a unitary eigendecomposition with real non-negative eigenvalues. ArcShift replaces the full magnetic phase with an explicit first-order directional transport component while preserving Hermitian structure.

**Lemma 1** (Hermitian Transport)**.** *The transport component $-iT^n$ is Hermitian.*

The proof is provided in Appendix 6.2.4. Consequently, ArcShift combines a real symmetric diffusion operator with a Hermitian directional transport component. To control their relative contributions, we introduce

$$\mathrm{ArcShift}^n_\beta = \hat{A}^n - i\beta T^n, \qquad \beta \geq 0, \tag{12}$$

where $\beta$ controls the strength of directional transport relative to diffusion. When $\beta = 0$, the operator reduces to symmetric GCN-style propagation, while $\beta = 1$ recovers the unscaled ArcShift formulation.

Since both $\hat{A}^n$ and $-i\beta T^n$ are Hermitian for real $\beta$, $\mathrm{ArcShift}^n_\beta$ remains Hermitian. Unlike the Hermitian Laplacian, however, positive semidefiniteness is not guaranteed. As $\beta$ decreases, the contribution of directional transport is reduced, and $\mathrm{ArcShift}^n_\beta$ continuously approaches the underlying symmetric propagation operator.

From a graph-learning perspective, the two components of ArcShift play complementary spectral and spatial roles. The diffusion component $\hat{A}^n$ retains the spectral foundation of symmetric Laplacian-based graph propagation and represents graph smoothing through the underlying symmetric structure. In contrast, the transport component $T^n = A^n \odot \Phi$ has a spatial interpretation: it acts directly over local graph neighborhoods and modulates propagation according to the directional imbalance of individual edges. Thus, the diffusion component is spectral in foundation, whereas the directional transport component provides a spatial mechanism for localized direction-aware message passing.

ArcShift therefore unifies a spectrally founded diffusion mechanism with spatial directional transport within a single Hermitian graph operator, providing a bridge between Hermitian spectral graph theory and spatial message-passing implementations.

### 3.6 Complex-Valued Neural Network

Because ArcShift produces complex-valued representations, complex-valued neural networks (CVNNs) provide a natural framework for jointly learning from their real and imaginary components (Hirose, 2011; Lee et al., 2022; Barrachina et al., 2023; Hammad, 2024). Complex-valued representations enable coupled transformations of these components within a unified neural architecture (Smith, 2023; Trabelsi et al., 2017). We therefore introduce ArcShift-CVNN, which employs complex-valued learnable transformations.

For a complex representation $X = X_{\mathrm{re}} + iX_{\mathrm{im}}$ and weight $W = W_{\mathrm{re}} + iW_{\mathrm{im}}$, complex multiplication couples the real and imaginary components as (Hirose, 2011; Trabelsi et al., 2017; Barrachina et al., 2023)

$$Z = WX = (W_{\mathrm{re}}X_{\mathrm{re}} - W_{\mathrm{im}}X_{\mathrm{im}}) + i\,(W_{\mathrm{im}}X_{\mathrm{re}} + W_{\mathrm{re}}X_{\mathrm{im}})\,,$$

where $Z = Z_{\mathrm{re}} + iZ_{\mathrm{im}}$ denotes the complex-valued pre-activation representation.

ArcShift-CVNN applies a split activation independently to the real and imaginary components (Hammad, 2024),

$$H_{\mathrm{re}} = \tanh(Z_{\mathrm{re}}), \qquad H_{\mathrm{im}} = \tanh(Z_{\mathrm{im}}),$$

yielding $H = H_{\mathrm{re}} + iH_{\mathrm{im}}$.

For node classification, the hidden representation is normalized row-wise using the L1 norm before the final classification layer. For each node $i$,

$$\bar{H}_i = \frac{H_i}{\|H_i\|_1},$$

with zero rows left unchanged. The normalized representation is then mapped to complex class logits

$$O = \bar{H}W_{\mathrm{out}} + b_{\mathrm{out}} = O_{\mathrm{re}} + iO_{\mathrm{im}},$$

where $W_{\mathrm{out}}$ and $b_{\mathrm{out}}$ are complex-valued learnable parameters. The real and imaginary logits are aggregated into real-valued class probabilities as

$$P = \frac{1}{2}\left[\mathrm{Softmax}(O_{\mathrm{re}}) + \mathrm{Softmax}(O_{\mathrm{im}})\right], \qquad \hat{y}_i = \arg\max_j P_{ij}.$$

Thus, ArcShift-CVNN jointly transforms the real and imaginary components throughout the network while producing standard real-valued class probabilities for node classification.

The explicit diffusion–transport decomposition further allows ArcShift-CVNN to learn the contribution of directional transport. We therefore introduce

$$\mathrm{ArcShift}^n_\gamma = \hat{A}^n - i\gamma T^n, \tag{13}$$

where $\gamma = \sigma(\tilde{\gamma}) \in (0, 1)$ is a learnable transport coefficient parameterized by the sigmoid function. Unlike the fixed coefficient $\beta$ in Equation 12, $\gamma$ is optimized jointly with the network parameters. Since $\gamma$ is real, this scaling preserves the Hermitian structure of the ArcShift operator.

## 4 Experimental Evaluation

We evaluate ArcShift on the downstream task of semi-supervised directed node classification. The experiments examine (i) how the first-order ArcShift operator compares with the exact magnetic operator under a controlled architecture, (ii) how the proposed models perform when evaluation is restricted to the native directed graph, (iii) how directional transport affects predictive performance and whether its strength can be adaptively learned, and (iv) the computational cost associated with directional-parameter selection. Complete experimental configurations, hyperparameter settings, random seeds, and other reproducibility details are provided in Appendix 6.5.

### 4.1 Baselines and Datasets

We compare ArcShift with representative graph-learning baselines, including ChebNet (Defferrard et al., 2016), GCN (Kipf & Welling, 2016), APPNP (Gasteiger et al., 2018), GraphSAGE (Hamilton

Table 1: Test accuracy (%) for the controlled single-layer operator comparison. Values are the mean ± sample standard deviation across ten predefined data splits. Bold indicates the higher mean accuracy for each dataset.

| Model | Cornell | Texas | Wisconsin | Cora-ML | CiteSeer | Telegram |
|---|---|---|---|---|---|---|
| MagNet-Core | $51.9 \pm 7.9$ | $\mathbf{67.6 \pm 10.9}$ | $58.8 \pm 7.9$ | $73.3 \pm 2.4$ | $58.0 \pm 1.9$ | $\mathbf{51.6 \pm 6.1}$ |
| ArcShift-Core | $\mathbf{63.0 \pm 6.7}$ | $65.9 \pm 6.0$ | $\mathbf{62.9 \pm 4.9}$ | $\mathbf{79.9 \pm 2.0}$ | $\mathbf{66.0 \pm 2.0}$ | $46.2 \pm 5.9$ |

et al., 2017), GIN (Xu et al., 2018), GAT (Veličković et al., 2017), DGCN (Tong et al., 2020b), DiGraph (Tong et al., 2020a), and MagNet (Zhang et al., 2021), together with SigMaNet (Fiorini et al., 2023) and HoloNet (Koke & Cremers, 2024) for broader comparison. MagNet serves as the primary directional spectral baseline because it employs the exact magnetic Laplacian from which ArcShift is derived.

Experiments use the MagNet node-classification benchmark (Zhang et al., 2021), consisting of Cornell, Texas, Wisconsin, Cora-ML, CiteSeer, and Telegram. We use the graph representations and ten predefined train/validation/test splits provided by the benchmark and report the mean and sample standard deviation across splits.

### 4.2 Implementation Details

For direct comparison with MagNet, ArcShift and ArcShift-CVNN use a first-order Chebyshev basis ($K = 1$) and two graph-convolution layers, except for the controlled single-layer comparison below. For standard ArcShift, the basis is

$$B_0 = I, \qquad B_1 = \mathrm{ArcShift}_{\beta}^{n} = \hat{A}^n - i\beta T^n,$$

whereas the adaptive ArcShift-CVNN variant uses

$$B_0 = I, \qquad B_1 = \mathrm{ArcShift}_{\gamma}^{n} = \hat{A}^n - i\gamma T^n.$$

Given the hidden representation $H^{(\ell)}$, each graph-convolution layer computes

$$Z^{(\ell+1)} = B_0 H^{(\ell)} W_0^{(\ell)} + B_1 H^{(\ell)} W_1^{(\ell)} + b^{(\ell)}, \qquad H^{(\ell+1)} = \sigma\Big(Z^{(\ell+1)}\Big).$$

Standard ArcShift uses real-valued learnable transformations with complex-valued graph propagation, whereas ArcShift-CVNN uses the complex-valued transformations described in Section 3.6.

### 4.3 Controlled Comparison of Directional Operators

Before evaluating the complete ArcShift architecture, we compare ArcShift with the magnetic operator under a controlled network configuration. This experiment isolates the effect of the directional propagation operator while minimizing differences arising from the surrounding neural-network architecture.

Both operators are evaluated using the same single graph-convolution layer with a first-order basis, together with identical node features, data splits, training procedures, and classifiers. The primary difference is therefore the directed graph operator: MagNet-Core employs the exact magnetic operator, whereas ArcShift-Core employs the proposed first-order diffusion–transport representation.

Table 1 reports the controlled comparison, with hyperparameters selected exclusively by mean validation accuracy across the ten predefined splits. ArcShift-Core achieves a higher reported mean accuracy on four of the six datasets, while MagNet-Core achieves a higher mean on Texas and Telegram. These results indicate that replacing the exact magnetic phase with the first-order diffusion–transport representation can preserve useful predictive information under the controlled architecture.

### 4.4 Directed-Graph Comparison

We next compare the representative graph-learning baselines on the original directed graphs provided by the benchmark datasets. ArcShift and ArcShift-CVNN are evaluated using their complete

Table 2: Directed-graph node-classification test accuracy (%). All methods use the native directed graph. Values are the mean ± sample standard deviation across ten predefined splits. Bold denotes the three highest means, ranked before rounding.

| Type | Model | Cornell | Texas | Wisconsin | Cora-ML | CiteSeer | Telegram |
|---|---|---|---|---|---|---|---|
| Spectral | GCN | $57.0 \pm 6.9$ | $58.6 \pm 5.6$ | $51.2 \pm 5.7$ | $72.1 \pm 2.4$ | $58.8 \pm 2.7$ | $61.2 \pm 5.2$ |
| Spatial | APPNP | $57.6 \pm 3.4$ | $58.1 \pm 4.1$ | $48.4 \pm 6.0$ | $68.0 \pm 3.8$ | $57.6 \pm 2.3$ | $64.4 \pm 6.5$ |
| | GraphSAGE | $80.5 \pm 7.2$ | $78.6 \pm 4.5$ | $78.4 \pm 7.2$ | $71.5 \pm 2.7$ | $60.1 \pm 3.7$ | $59.2 \pm 3.2$ |
| | GIN | $54.3 \pm 5.5$ | $58.4 \pm 6.0$ | $46.9 \pm 5.8$ | $66.3 \pm 2.8$ | $58.1 \pm 2.9$ | $67.4 \pm 5.1$ |
| | GAT | $58.4 \pm 5.1$ | $58.6 \pm 5.4$ | $51.2 \pm 5.9$ | $73.8 \pm 2.2$ | $62.6 \pm 1.8$ | $65.8 \pm 4.8$ |
| Directed | DGCN | $61.9 \pm 7.8$ | $69.5 \pm 5.1$ | $62.0 \pm 5.8$ | $\mathbf{79.8 \pm 2.3}$ | $\mathbf{65.1 \pm 1.8}$ | $\mathbf{87.8 \pm 4.5}$ |
| | DiGraph | $67.0 \pm 6.2$ | $62.7 \pm 8.4$ | $58.2 \pm 4.5$ | $78.8 \pm 1.6$ | $62.3 \pm 2.4$ | $81.4 \pm 5.5$ |
| | DiGraphIB | $67.3 \pm 9.0$ | $62.4 \pm 8.9$ | $67.3 \pm 12.6$ | $\mathbf{79.3 \pm 1.9}$ | $61.1 \pm 2.1$ | $67.2 \pm 4.4$ |
| Directed spectral | MagNet | $\mathbf{83.5 \pm 6.4}$ | $\mathbf{83.8 \pm 5.8}$ | $\mathbf{85.3 \pm 3.9}$ | $79.1 \pm 1.3$ | $\mathbf{66.8 \pm 1.6}$ | $87.2 \pm 3.6$ |
| | HoloNet | $53.2 \pm 4.2$ | $71.1 \pm 7.6$ | $65.9 \pm 9.7$ | $75.7 \pm 2.0$ | $56.9 \pm 2.8$ | $\mathbf{93.0 \pm 1.9}$ |
| | SigMaNet | $\mathbf{84.1 \pm 5.9}$ | $\mathbf{84.6 \pm 6.7}$ | $82.0 \pm 3.9$ | $67.1 \pm 2.0$ | $56.8 \pm 2.7$ | $81.2 \pm 5.4$ |
| This paper | ArcShift | $76.5 \pm 6.4$ | $82.7 \pm 5.9$ | $\mathbf{83.3 \pm 4.9}$ | $\mathbf{80.3 \pm 1.9}$ | $64.1 \pm 2.7$ | $\mathbf{88.6 \pm 4.0}$ |
| | ArcShift-CVNN | $\mathbf{81.6 \pm 6.8}$ | $\mathbf{85.9 \pm 5.9}$ | $\mathbf{86.1 \pm 4.5}$ | $77.5 \pm 2.0$ | $\mathbf{64.6 \pm 2.3}$ | $83.0 \pm 4.6$ |

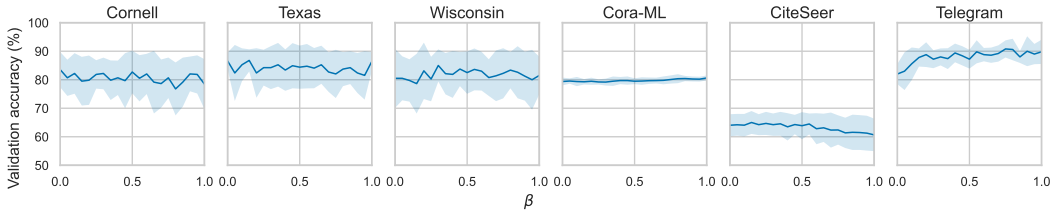


Figure 2: Validation accuracy of ArcShift as the directed-graph transport coefficient $\beta$ varies from 0 to 1 in increments of 0.05. Lines show mean validation accuracy and shaded regions show one sample standard deviation across ten predefined splits.

two-layer architectures. By preserving the original edge directions, this setting directly evaluates the ability of each model to exploit directional information during graph propagation. Table 2 shows that no single model achieves the highest reported mean across all six datasets. Among the proposed models, ArcShift achieves the highest reported mean on Cora-ML and the second-highest mean on Telegram, while ArcShift-CVNN achieves the highest reported mean on Texas and Wisconsin. On Cornell and CiteSeer, the highest reported means are obtained by SigMaNet and MagNet, respectively.

### 4.5 Transport Sensitivity and Adaptive Scaling

Using the fixed transport-strength parameter $\beta$ in Equation 12, we examine the contribution of directional transport over $\beta \in [0, 1]$, where $\beta = 0$ corresponds to diffusion-only propagation. Figure 2 shows that performance is relatively insensitive to the precise transport strength on most datasets, with changes in mean validation accuracy generally small relative to the variability across data splits. Selecting $\beta$ nevertheless introduces an additional hyperparameter-search dimension. ArcShift-CVNN instead learns $\gamma$ jointly with the network parameters, allowing the transport strength to adapt during training. As shown in Figure 3, the learned coefficients move away from the common initialization $\gamma_0 = 0.5$ toward dataset-dependent values.

The explicit diffusion–transport decomposition makes this adaptive parameterization particularly simple: $\gamma$ directly scales the fixed transport operator $T^n$. In contrast, MagNet's magnetic parameter $q$ appears inside the nonlinear complex phase $e^{i2\pi q\Phi}$. Although $q$ is differentiable and could in principle be learned, changing $q$ modifies the phase-dependent magnetic operator during optimization. ArcShift instead exposes transport strength as the linear coefficient $\gamma$ of the fixed operator $T^n$, allowing it to be optimized directly with the network parameters without reconstructing the transport operator.

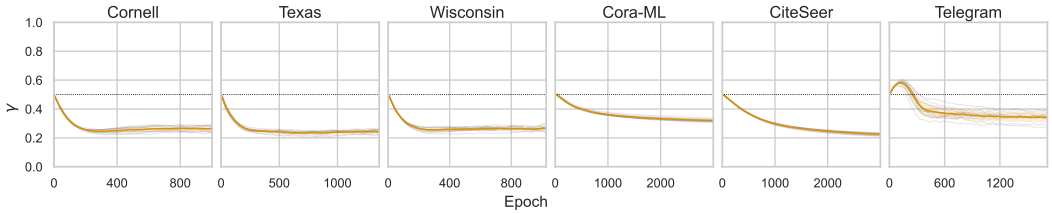


Figure 3: Learned $\gamma$ trajectories of ArcShift-CVNN on the six directed datasets. Thin gray lines show individual splits, the orange line shows their mean, and the shaded region shows one sample standard deviation. The dotted line marks $\gamma_0 = 0.5$. Epochs are shown through the longest prefix shared by all ten splits.

### 4.6 Computational Timing

Considering graph propagation, training, prediction, and directional-parameter selection, we analyze the computational behavior of the Hermitian-Laplacian-based models. For a graph with $|E|$ edges and feature dimension $d$, ArcShift constructs its normalized diffusion and transport operators once before training. With a sparse graph representation, this preprocessing requires $\mathcal{O}(|E|)$ operations. During training and inference, applying the resulting operators to the node features has graph-propagation complexity $\mathcal{O}(|E|d)$. Since the diffusion and transport terms share the same graph support, their explicit decomposition does not change the asymptotic order of graph propagation. Detailed timing results are reported in Appendix 6.4.

## 5 Conclusion and Future Work

ArcShift provides an explicit diffusion–transport formulation of magnetic directed-graph propagation through a first-order approximation while preserving the Hermitian operator structure. Across the directed node-classification benchmarks, ArcShift achieves competitive predictive performance relative to the exact magnetic formulation and representative baselines. The experiments further show that the effect of transport strength is dataset-dependent, while ArcShift-CVNN removes the need for an external search over the transport parameter.

The explicit diffusion–transport decomposition is particularly relevant to weighted directed systems, where interaction magnitude and direction can encode distinct information. For example, in flow networks, edge magnitude may represent flow volume, while directional imbalance determines the associated phase. In related Hermitian-Laplacian formulations, MagNet couples magnitude and direction through the complex magnetic phase, while SigMaNet uses a sign-based directional encoding rather than a continuously varying phase. Consequently, these formulations do not expose magnitude and directional transport as separately controllable components. ArcShift makes this separation explicit, allowing the directional transport contribution to be directly scaled independently of the diffusion component.

Our future work will evaluate ArcShift on larger-scale and dynamic directed graphs to further investigate its scalability. In dynamic settings, graph states and directional interactions may evolve over time, including node-hopping or recurrent frameworks in which predicted transitions update the graph. Independently evolving diffusion and transport could further support sequential prediction on transportation, supply-chain, communication, and dynamic-routing networks.

## REFERENCES


Jose Agustin Barrachina, Chengfang Ren, Gilles Vieillard, Christele Morisseau, and Jean-Philippe Ovarlez. Theory and implementation of complex-valued neural networks. *arXiv preprint arXiv:2302.08286*, 2023.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

CHM Chan, A Cionca, V Škultéty, and D Van De Ville. Graph diffusion-advection operator for directed graph signal processing. *arXiv preprint arXiv:2606.16306*, 2026.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

Stefano Fiorini, Stefano Coniglio, Michele Ciavotta, and Enza Messina. Sigmanet: One laplacian to rule them all. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 7568–7576, 2023.

Stefano Fiorini, Giulia M Bovolenta, Stefano Coniglio, Michele Ciavotta, Pietro Morerio, Michele Parrinello, and Alessio Del Bue. Dlgnet: Hyperedge classification through directed line graphs for chemical reactions. *arXiv preprint arXiv:2410.06969*, 2024.

Satoshi Furutani, Toshiki Shibahara, Mitsuaki Akiyama, Kunio Hato, and Masaki Aida. Graph signal processing for directed graphs based on the hermitian laplacian. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 447–463. Springer, 2019.

Chen Gao, Yu Zheng, Nian Li, Yinfeng Li, Yingrong Qin, Jinghua Piao, Yuhan Quan, Jianxin Chang, Depeng Jin, Xiangnan He, et al. A survey of graph neural networks for recommender systems: Challenges, methods, and directions. *ACM Transactions on Recommender Systems*, 1(1):1–51, 2023.

Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997*, 2018.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

MM Hammad. Comprehensive survey of complex-valued neural networks: Insights into backpropagation and activation functions. *arXiv preprint arXiv:2407.19258*, 2024.

Yixuan He, Michael Perlmutter, Gesine Reinert, and Mihai Cucuringu. Msgnn: A spectral graph neural network based on a novel magnetic signed laplacian. In *Learning on graphs conference*, pp. 40–1. PMLR, 2022.

Akira Hirose. Complex-valued neural networks. *IEEJ Transactions on Electronics, Information and Systems*, 131(1):2–8, 2011.

Yinan Huang, Haoyu Wang, and Pan Li. What are good positional encodings for directed graphs? In *International Conference on Learning Representations*, volume 2025, pp. 36473–36497, 2025.

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

Christian Koke and Daniel Cremers. Holonets: Spectral convolutions do extend to directed graphs. In *International Conference on Learning Representations*, volume 2024, pp. 49121–49154, 2024.

ChiYan Lee, Hideyuki Hasegawa, and Shangce Gao. Complex-valued neural networks: A comprehensive survey. *IEEE/CAA Journal of Automatica Sinica*, 9(8):1406–1426, 2022.

Lequan Lin and Junbin Gao. A magnetic framelet-based convolutional neural network for directed graphs. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Seiji Maekawa, Yuya Sasaki, and Makoto Onizuka. A simple and scalable graph neural network for large directed graphs. *arXiv preprint arXiv:2306.08274*, 2023.

Jakob Möller. The pauli-poisson equation and its semiclassical limit. *Communications in Partial Differential Equations*, 50(1-2):130–161, 2025.

Valerio Ponzi and Christian Napoli. Graph neural networks: Architectures, applications, and future directions. *IEEE Access*, 13:62870–62891, 2025.

Mattia Rigotti, Nicholas Thumiger, and Thomas Frick. Gist: Gauge-invariant spectral transformers for scalable graph neural operators. *arXiv preprint arXiv:2603.16849*, 2026.

David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine*, 30(3):83–98, 2013.

Josiah W Smith. Complex-valued neural networks for data-driven signal processing and signal understanding. *arXiv preprint arXiv:2309.07948*, 2023.

Henan Sun, Xunkai Li, Daohan Su, Junyi Han, Rong-Hua Li, and Guoren Wang. Towards data-centric machine learning on directed graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 2026.

Zekun Tong, Yuxuan Liang, Changsheng Sun, Xinke Li, David Rosenblum, and Andrew Lim. Digraph inception convolutional networks. *Advances in neural information processing systems*, 33: 17907–17918, 2020a.

Zekun Tong, Yuxuan Liang, Changsheng Sun, David S Rosenblum, and Andrew Lim. Directed graph convolutional network. *arXiv preprint arXiv:2004.13970*, 2020b.

Chiheb Trabelsi, Olexa Bilaniuk, Ying Zhang, Dmitriy Serdyuk, Sandeep Subramanian, Joao Felipe Santos, Soroush Mehri, Negar Rostamzadeh, Yoshua Bengio, and Christopher J Pal. Deep complex networks. *arXiv preprint arXiv:1705.09792*, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Jiaqing Xie and Yuxin Wang. Eigenbasis-independent learnable spectral positional encodings for directed graphs via hermitian block krylov subspaces. *arXiv e-prints*, pp. arXiv–2607, 2026.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

Xitong Zhang, Yixuan He, Nathan Brugnone, Michael Perlmutter, and Matthew Hirn. Magnet: A neural network for directed graphs. *Advances in neural information processing systems*, 34: 27003–27015, 2021.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020.

# 6 APPENDIX

## 6.1 NOTATION

**Notation and Descriptions**

| | |
|---|---|
| $A^{(s)}$ | Symmetric adjacency matrix |
| $\Phi$ | Directional imbalance matrix |
| $\mathcal{L}_q$ | Hermitian (magnetic) Laplacian |
| $L_D$ | Symmetric diffusion component of $\mathcal{L}_q$ |
| $L_T$ | Directional transport component of $\mathcal{L}_q$ |
| $A^n$ | Normalized symmetric adjacency matrix |
| $\hat{A}^n$ | Renormalized diffusion operator |
| $T^n$ | Normalized directional transport operator |
| $q$ | Magnetic phase parameter |
| $\beta$ | Fixed transport-strength parameter |
| $\gamma$ | Learnable transport-strength parameter |
| $\mathrm{ArcShift}_\beta^n$ | ArcShift operator with fixed transport strength |
| $\mathrm{ArcShift}_\gamma^n$ | ArcShift operator with learnable transport strength |
| $B_k$ | Graph-convolution basis operator indexed by $k$ |
| $Z^{(\ell)}$ | Pre-activation representation at layer $\ell$ |
| $H^{(\ell)}$ | Hidden representation at layer $\ell$ |
| $(\cdot)_{\mathrm{re}}$ | Real component of a complex-valued quantity |
| $(\cdot)_{\mathrm{im}}$ | Imaginary component of a complex-valued quantity |

## 6.2 PROOFS

### 6.2.1 PROOF OF THEOREM 1

*Proof.* The result follows directly from Euler's identity in Equation 2 and substitution into Equation 1. □

### 6.2.2 PROOF OF THEOREM 2

*Proof.* The first-order expansion in Equation 6 separates the symmetric term from a directional term linear in $\Phi$. Replacing the symmetric component by the renormalized propagation operator $\hat{A}^n$ and defining the normalized directional component as $T^n = A^n \odot \Phi$ yields Equation 9. □

### 6.2.3 PROOF OF THEOREM 3

*Proof.* By the Taylor series expansion of the complex exponential,

$$e^{ix} = 1 + ix - \frac{x^2}{2} + \mathcal{O}(x^3).$$

For each entry $(i, j)$, setting $x = 2\pi q\phi_{ij}$ and subtracting $1 + ix$ yields

$$e^{i2\pi q\phi_{ij}} - (1 + i2\pi q\phi_{ij}) = -\frac{(2\pi q\phi_{ij})^2}{2} + \mathcal{O}\big((2\pi q\phi_{ij})^3\big),$$

which proves Equation 11. □

Table 3: Mixed directed/undirected model-level node-classification test accuracy (%). The experiment follows the original MagNet benchmark evaluation setup. Values are the mean ± sample standard deviation across ten predefined data splits. Bold denotes the three highest means, ranked before rounding.

| Type | Model | Cornell | Texas | Wisconsin | Cora-ML | CiteSeer | Telegram |
|---|---|---|---|---|---|---|---|
| Spectral | ChebNet | $77.8 \pm 4.0$ | $80.5 \pm 7.0$ | $82.9 \pm 4.1$ | $79.9 \pm 1.5$ | $65.5 \pm 3.0$ | $83.2 \pm 2.5$ |
| | GCN | $58.6 \pm 5.6$ | $57.0 \pm 6.6$ | $52.2 \pm 5.6$ | $\mathbf{81.6 \pm 1.8}$ | $64.9 \pm 2.0$ | $83.8 \pm 4.4$ |
| Spatial | APPNP | $58.9 \pm 4.7$ | $58.4 \pm 4.1$ | $52.2 \pm 7.0$ | $\mathbf{82.6 \pm 1.7}$ | $\mathbf{66.2 \pm 2.0}$ | $69.2 \pm 3.2$ |
| | GraphSAGE | $78.6 \pm 5.0$ | $\mathbf{82.7 \pm 6.1}$ | $78.6 \pm 5.5$ | $\mathbf{81.6 \pm 1.5}$ | $64.6 \pm 2.7$ | $68.2 \pm 6.1$ |
| | GIN | $58.6 \pm 2.9$ | $64.6 \pm 5.6$ | $59.0 \pm 2.8$ | $77.9 \pm 1.6$ | $61.1 \pm 2.4$ | $\mathbf{88.4 \pm 2.6}$ |
| | GAT | $60.3 \pm 6.5$ | $62.2 \pm 5.3$ | $51.4 \pm 8.3$ | $81.4 \pm 1.3$ | $\mathbf{66.6 \pm 2.0}$ | $77.2 \pm 4.7$ |
| Directed | DGCN | $64.1 \pm 7.4$ | $71.6 \pm 5.4$ | $64.7 \pm 6.2$ | $80.7 \pm 1.8$ | $64.0 \pm 2.2$ | $87.2 \pm 4.2$ |
| | DiGraph | $63.8 \pm 6.5$ | $64.9 \pm 6.5$ | $58.0 \pm 6.9$ | $78.9 \pm 2.1$ | $60.4 \pm 3.4$ | $80.8 \pm 4.7$ |
| | DiGraphIB | $64.6 \pm 8.6$ | $61.6 \pm 11.1$ | $65.3 \pm 9.6$ | $79.7 \pm 1.4$ | $60.4 \pm 1.9$ | $65.2 \pm 3.7$ |
| Directed spectral | MagNet | $\mathbf{83.2 \pm 7.2}$ | $\mathbf{83.2 \pm 5.1}$ | $82.7 \pm 6.7$ | $79.5 \pm 1.0$ | $\mathbf{66.5 \pm 2.1}$ | $87.0 \pm 1.4$ |
| | HoloNet | $55.7 \pm 6.3$ | $71.1 \pm 6.2$ | $67.5 \pm 7.2$ | $79.0 \pm 1.7$ | $63.3 \pm 1.5$ | $\mathbf{94.8 \pm 2.5}$ |
| | SigMaNet | $\mathbf{81.4 \pm 8.7}$ | $82.2 \pm 6.0$ | $\mathbf{83.3 \pm 3.8}$ | $67.3 \pm 2.2$ | $57.1 \pm 1.8$ | $81.6 \pm 5.1$ |
| This paper | ArcShift | $76.8 \pm 6.0$ | $81.6 \pm 7.0$ | $\mathbf{86.1 \pm 7.5}$ | $80.0 \pm 1.5$ | $64.8 \pm 3.4$ | $\mathbf{88.2 \pm 5.0}$ |
| | ArcShift-CVNN | $\mathbf{81.4 \pm 6.2}$ | $\mathbf{86.2 \pm 4.7}$ | $\mathbf{85.1 \pm 2.6}$ | $78.7 \pm 1.5$ | $65.2 \pm 1.9$ | $81.4 \pm 6.0$ |

#### 6.2.4 Proof of Lemma 1

*Proof.* Since $A^n$ is symmetric and $\Phi$ is antisymmetric, $T^n = A^n \odot \Phi$ is real and antisymmetric. Hence

$$(T^n)^* = (T^n)^\top = -T^n.$$

Therefore,

$$(-iT^n)^* = i(T^n)^* = -iT^n,$$

which proves that $-iT^n$ is Hermitian. □

### 6.3 Mixed Directed/Undirected Model-Level Comparison

Using the representative graph-learning baselines described in Section 4, we evaluate ArcShift and ArcShift-CVNN using their complete two-layer architectures. This experiment follows the original model-level evaluation setup provided by the MagNet benchmark, in which both directed and symmetrized (undirected) graph representations may be considered according to the evaluated model and configuration. ArcShift uses real-valued learnable transformations with complex-valued graph propagation, whereas ArcShift-CVNN uses complex-valued learnable transformations. All models use the same benchmark datasets and predefined data splits, with configurations selected according to the common validation-based evaluation protocol.

Across all six datasets, no single model achieves the highest reported mean, as shown in Table 3. Among the proposed models, ArcShift achieves the highest reported mean on Wisconsin, while ArcShift-CVNN achieves the highest reported mean on Texas. Overall, the relative performance of ArcShift and ArcShift-CVNN varies across datasets, indicating that the benefit of complex-valued learnable transformations is dataset-dependent.

### 6.4 Computational Timing

Here, we examine the computational behavior of the Hermitian-Laplacian-based models. We report training, prediction, and directional-parameter-selection measurements from one sequential timing run on a single NVIDIA L40S GPU under matching software conditions. Training measurements are aggregated over six datasets and ten predefined splits per dataset. Table 4 summarizes training and prediction measurements together with the computational cost of directional-parameter selection. ArcShift has the lowest measured mean per-epoch time among the four Hermitian-Laplacian-based models (16.84 ms) and the lowest total training time (13.97 min). Its mean prediction time (2.43 ms) is also lower than those of MagNet and SigMaNet. ArcShift-CVNN has a higher total training cost

Table 4: Aggregate computational timing and directional-parameter search cost on one NVIDIA L40S GPU. Values are the mean ± sample standard deviation over 60 split runs unless reported as totals.

| Metric | MagNet | SigMaNet | ArcShift | ArcShift-CVNN |
|---|---|---|---|---|
| *Single-configuration timing* | | | | |
| Epoch time (ms) | $22.95 \pm 3.20$ | $23.29 \pm 9.35$ | $16.84 \pm 1.31$ | $21.39 \pm 2.07$ |
| Epochs run | $921.6 \pm 412.8$ | $884.6 \pm 341.7$ | $749.5 \pm 311.0$ | $1761.5 \pm 551.0$ |
| Total training time (min) | 21.84 | 21.51 | 13.97 | 42.96 |
| Prediction time (ms) | $4.79 \pm 2.46$ | $6.65 \pm 5.42$ | $2.43 \pm 1.65$ | $2.77 \pm 1.96$ |
| *Directional-parameter search* | | | | |
| Search configurations | 36 | 6 | 126 | 6 |
| Total search time (min) | 121.17 | 21.51 | 336.75 | 42.96 |

and runs for substantially more epochs, requiring $42.96$ min in total, although its mean prediction time remains $2.77$ ms.

The fixed transport coefficient $\beta$ introduces an additional hyperparameter-search dimension for ArcShift. ArcShift evaluates 21 values of $\beta$ from 0 to 1 in increments of 0.05, whereas MagNet evaluates the six positive values $q \in \{0.01, 0.05, 0.10, 0.15, 0.20, 0.25\}$ provided by the MagNet benchmark. Across the six datasets, these searches require $336.75$ min and $121.17$ min, respectively. ArcShift-CVNN instead learns $\gamma$ jointly with the network parameters and therefore requires only one directional configuration per dataset, with a total measured selection time of $42.96$ min. SigMaNet similarly requires no directional-parameter sweep because its Sign-Magnetic Laplacian is parameter-free, resulting in one directional configuration per dataset and $21.51$ min across the six datasets. Its directional encoding uses the sign of the weight imbalance rather than a continuously varying phase. For an imbalanced edge pair, the corresponding real component vanishes and the directional relation is represented through the imaginary component. Consequently, SigMaNet captures the orientation of the weight imbalance, but its directional term does not distinguish its precise magnitude through a continuously varying phase. Thus, ArcShift has the lowest single-configuration training cost in this comparison, while ArcShift-CVNN removes the external transport-parameter sweep by learning a continuous directional coefficient jointly with the network parameters.

Table 5 shows the details of training and inference time. ArcShift generally reduces the per-epoch computation time relative to MagNet and SigMaNet, with the exception of Cornell, where its epoch time is comparable to SigMaNet. The reduction is more pronounced on the larger Cora-ML and CiteSeer datasets. ArcShift also achieves the lowest prediction time across all six datasets. The total training time depends on both the per-epoch cost and the number of epochs required before stopping; consequently, the relative training-time advantage varies across datasets. In contrast, ArcShift-CVNN retains low prediction time but requires substantially longer total training time, primarily because it runs for more epochs. These results indicate that the simplified ArcShift propagation provides practical computational benefits for standard ArcShift, while ArcShift-CVNN has a higher overall training cost.

### 6.5 Reproducibility Settings

We provide the model definitions, mathematical formulation, and implementation details necessary to reproduce the proposed ArcShift and ArcShift-CVNN models in the main paper. Dataset splits, hyperparameter settings, evaluation protocols, and additional experimental details are provided. The experiments follow the MagNet benchmark protocol where applicable.

**Run seeds.** Table 6 lists the recorded run seed for each paper-facing study. The directional-coefficient analysis performs no additional training and derives its evidence from the final model-level and directed-only comparisons.

Table 5: Computational timing on one NVIDIA L40S GPU. Values are the mean ± sample standard deviation across ten predefined data splits. Epoch time measures synchronized training and validation computation after warmup, training wall time covers one complete split run, and prediction time measures one synchronized forward pass.

| Dataset | Model | Epoch (ms) | Epochs run | Train/split (s) | Predict (ms) |
|---|---|---|---|---|---|
| Cornell | MagNet | $20.26 \pm 0.03$ | $865.7 \pm 293.3$ | $17.66 \pm 6.08$ | $3.04 \pm 0.01$ |
| Cornell | SigMaNet | $16.26 \pm 0.06$ | $889.5 \pm 230.2$ | $14.59 \pm 3.86$ | $2.84 \pm 0.02$ |
| Cornell | ArcShift | $16.37 \pm 0.28$ | $651.5 \pm 234.5$ | $12.09 \pm 4.29$ | $1.31 \pm 0.01$ |
| Cornell | ArcShift-CVNN | $19.80 \pm 0.13$ | $1822.0 \pm 348.0$ | $42.44 \pm 7.96$ | $1.47 \pm 0.03$ |
| Texas | MagNet | $22.07 \pm 0.08$ | $791.4 \pm 223.6$ | $17.59 \pm 4.98$ | $3.10 \pm 0.02$ |
| Texas | SigMaNet | $17.75 \pm 0.11$ | $883.9 \pm 307.5$ | $15.81 \pm 5.47$ | $2.85 \pm 0.01$ |
| Texas | ArcShift | $15.69 \pm 0.27$ | $863.5 \pm 341.2$ | $15.35 \pm 6.20$ | $1.32 \pm 0.01$ |
| Texas | ArcShift-CVNN | $20.81 \pm 0.06$ | $1999.5 \pm 462.1$ | $48.91 \pm 11.29$ | $1.46 \pm 0.01$ |
| Wisconsin | MagNet | $20.34 \pm 0.03$ | $848.2 \pm 328.1$ | $17.36 \pm 6.79$ | $3.06 \pm 0.01$ |
| Wisconsin | SigMaNet | $16.52 \pm 0.13$ | $792.0 \pm 251.9$ | $13.17 \pm 4.13$ | $2.91 \pm 0.01$ |
| Wisconsin | ArcShift | $15.96 \pm 0.07$ | $806.4 \pm 396.3$ | $14.56 \pm 7.20$ | $1.32 \pm 0.01$ |
| Wisconsin | ArcShift-CVNN | $19.94 \pm 0.12$ | $1985.3 \pm 568.0$ | $46.71 \pm 13.45$ | $1.47 \pm 0.00$ |
| Cora-ML | MagNet | $24.56 \pm 0.05$ | $1000.4 \pm 322.7$ | $24.70 \pm 7.96$ | $7.19 \pm 0.01$ |
| Cora-ML | SigMaNet | $34.73 \pm 0.03$ | $1067.2 \pm 484.9$ | $37.21 \pm 16.79$ | $13.60 \pm 0.01$ |
| Cora-ML | ArcShift | $17.09 \pm 0.03$ | $1033.0 \pm 270.7$ | $19.75 \pm 5.11$ | $3.88 \pm 0.00$ |
| Cora-ML | ArcShift-CVNN | $20.98 \pm 0.04$ | $1847.5 \pm 418.4$ | $45.47 \pm 10.24$ | $4.26 \pm 0.01$ |
| CiteSeer | MagNet | $29.27 \pm 0.16$ | $1302.6 \pm 690.1$ | $38.33 \pm 20.33$ | $9.12 \pm 0.01$ |
| CiteSeer | SigMaNet | $37.91 \pm 0.12$ | $948.8 \pm 356.3$ | $36.13 \pm 13.70$ | $14.85 \pm 0.02$ |
| CiteSeer | ArcShift | $19.53 \pm 0.01$ | $507.8 \pm 0.8$ | $10.68 \pm 0.17$ | $5.45 \pm 0.01$ |
| CiteSeer | ArcShift-CVNN | $25.86 \pm 0.06$ | $1072.8 \pm 216.9$ | $31.09 \pm 6.20$ | $6.46 \pm 0.01$ |
| Telegram | MagNet | $21.21 \pm 0.09$ | $721.4 \pm 228.0$ | $15.40 \pm 4.93$ | $3.26 \pm 0.01$ |
| Telegram | SigMaNet | $16.55 \pm 0.15$ | $725.9 \pm 336.2$ | $12.14 \pm 5.86$ | $2.84 \pm 0.01$ |
| Telegram | ArcShift | $16.41 \pm 0.53$ | $634.9 \pm 186.4$ | $11.42 \pm 3.55$ | $1.33 \pm 0.01$ |
| Telegram | ArcShift-CVNN | $20.94 \pm 0.16$ | $1842.0 \pm 667.7$ | $43.12 \pm 15.57$ | $1.49 \pm 0.01$ |

Table 6: Recorded random seeds for the experimental studies.

| Study | Seed |
|---|---|
| Mixed model-level comparison | 1196373155 |
| Directed-graph comparison | 983601116 |
| Computational timing | 1181655710 |
| Transport sensitivity | Derived from the model-level and directed-graph comparisons |
| Controlled operator comparison | 1970310374 |

**Hyperparameter grids.** Unless an exception is listed below, the model-level and directed-only comparisons use learning rates

$$\{0.001, 0.005, 0.01\}$$

and hidden widths

$$\{16, 32, 48\}.$$

Every configuration uses at most 3,000 epochs, dropout 0.5, and weight decay $5 \times 10^{-4}$. Configurations are selected using mean validation accuracy across the ten predefined splits.

**Model-specific settings.** Here, "native" denotes the original directed graph and `-tud` denotes the symmetrized branch. The remaining model-specific settings and exceptions are as follows.

- ChebNet uses $K = 2$, is evaluated only on the symmetric graph in the model-level comparison, and is excluded from the directed-only comparison.
- GCN, GraphSAGE, and GIN have no additional searched parameter. Their model-level runs include native and `-tud` branches; their directed-only runs use native graphs only.

Table 7: Validation-selected configurations for the controlled single-layer operator comparison. A dash denotes a parameter that does not apply to the corresponding model.

| Dataset | Model | $q$ | $\beta$ | Learning rate | Width | Validation (%) | Test (%) |
|---|---|---|---|---|---|---|---|
| Cornell | MagNet-Core | 0.25 | – | 0.01 | 16 | $52.9 \pm 5.0$ | $51.9 \pm 7.9$ |
| | ArcShift-Core | – | 0.85 | 0.005 | 48 | $65.9 \pm 7.0$ | $63.0 \pm 6.7$ |
| Texas | MagNet-Core | 0.15 | – | 0.005 | 48 | $69.2 \pm 6.2$ | $67.6 \pm 10.9$ |
| | ArcShift-Core | – | 1.00 | 0.01 | 16 | $68.5 \pm 4.1$ | $65.9 \pm 6.0$ |
| Wisconsin | MagNet-Core | 0.05 | – | 0.001 | 48 | $59.3 \pm 3.3$ | $58.8 \pm 7.9$ |
| | ArcShift-Core | – | 1.00 | 0.01 | 48 | $64.3 \pm 3.5$ | $62.9 \pm 4.9$ |
| Cora-ML | MagNet-Core | 0.00 | – | 0.01 | 16 | $72.2 \pm 2.4$ | $73.3 \pm 2.4$ |
| | ArcShift-Core | – | 1.00 | 0.01 | 32 | $80.1 \pm 1.2$ | $79.9 \pm 2.0$ |
| CiteSeer | MagNet-Core | 0.00 | – | 0.01 | 48 | $57.4 \pm 2.6$ | $58.0 \pm 1.9$ |
| | ArcShift-Core | – | 0.35 | 0.005 | 48 | $65.8 \pm 2.8$ | $66.0 \pm 2.0$ |
| Telegram | MagNet-Core | 0.15 | – | 0.005 | 16 | $50.8 \pm 4.3$ | $51.6 \pm 6.1$ |
| | ArcShift-Core | – | 0.95 | 0.005 | 32 | $46.4 \pm 6.0$ | $46.2 \pm 5.9$ |

- APPNP and DiGraph search $\alpha \in \{0.05, 0.10, 0.15, 0.20\}$. APPNP uses native and `-tud` branches in the model-level comparison and native graphs in the directed-only comparison; DiGraph is native-only.
- GAT searches the number of attention heads in $\{2, 4, 8\}$, with native and `-tud` model-level branches and a native-only directed comparison.
- DGCN uses dataset-generation seed 0, uses the experiment seed for training, and is native-only.
- DiGraphIB uses widths $\{6, 11, 21\}$, searches $\alpha \in \{0.05, 0.10, 0.15, 0.20\}$, and is native-only.
- MagNet uses $K = 1$ and native graphs. The model-level comparison uses the paper-reported dataset-specific $q$; the directed-only comparison searches $q \in \{0.01, 0.05, 0.10, 0.15, 0.20, 0.25\}$.
- HoloNet uses $\alpha = 0.5$, $k_+ = 3$, AdamW, and a validation-accuracy checkpoint. Its model-level runs include native and `-tud` branches; its directed-only runs use native graphs.
- SigMaNet uses $K = 1$ and native graphs only.
- ArcShift uses $K = 1$ and searches $\beta \in \{0.00, 0.05, \ldots, 1.00\}$. Its model-level comparison includes native and `-tud` branches, with the operator-equivalent value $\beta = 0$ used for `-tud`; its directed-only comparison is native-only.
- **ArcShift-CVNN** uses $K = 1$, with learnable $\gamma \in (0, 1)$ initialized at 0.5. The hidden complex L1 normalization described in Section 3.6 is retained. Its model-level comparison includes native and `-tud` branches; its directed-only comparison is native-only.

### 6.5.1 Selected Experimental Configurations

Tables 7–9 report the validation-selected configurations for the controlled operator comparison, the mixed directed/undirected model-level comparison, and the directed-only comparison. Configurations were selected exclusively by mean validation accuracy across the ten predefined splits, with test accuracy reported only after configuration selection.

**Controlled operator comparison.** Table 7 reports the selected configurations for the controlled single-layer comparison. Validation and test results are the mean $\pm$ sample standard deviation across the ten predefined splits.

For the controlled operator comparison, MagNet-Core uses the dataset-specific magnetic phase values $q$ shown in Table 7, while ArcShift-Core searches $\beta \in \{0.00, 0.05, \ldots, 1.00\}$. Learning rate and hidden width are selected by mean validation accuracy for both models.

Table 8: Validation-selected ArcShift configurations for the mixed directed/undirected model-level comparison. Native denotes the original directed graph; `-tud` denotes the corresponding symmetrized graph branch. A dash denotes a parameter that does not apply.

| Dataset | Model | Graph mode | Selected $\beta$ | Learned $\gamma$ | Learning rate | Width | Validation (%) | Test (%) |
|---|---|---|---|---|---|---|---|---|
| Cornell | ArcShift | Native | 0.00 | – | 0.005 | 48 | $83.6 \pm 6.2$ | $76.8 \pm 6.0$ |
| | ArcShift-CVNN | `-tud` | – | – | 0.01 | 32 | $84.7 \pm 7.0$ | $81.4 \pm 6.2$ |
| Texas | ArcShift | Native | 0.15 | – | 0.01 | 48 | $86.8 \pm 3.9$ | $81.6 \pm 7.0$ |
| | ArcShift-CVNN | Native | – | $0.329 \pm 0.042$ | 0.01 | 16 | $89.2 \pm 4.4$ | $86.2 \pm 4.7$ |
| Wisconsin | ArcShift | Native | 0.30 | – | 0.01 | 48 | $85.0 \pm 5.7$ | $86.1 \pm 7.5$ |
| | ArcShift-CVNN | `-tud` | – | – | 0.01 | 32 | $84.7 \pm 3.2$ | $85.1 \pm 2.6$ |
| Cora-ML | ArcShift | Native | 1.00 | – | 0.005 | 48 | $80.6 \pm 0.9$ | $80.0 \pm 1.5$ |
| | ArcShift-CVNN | `-tud` | – | – | 0.001 | 48 | $78.6 \pm 1.8$ | $78.7 \pm 1.5$ |
| CiteSeer | ArcShift | `-tud` | – | – | 0.001 | 48 | $65.6 \pm 4.3$ | $64.8 \pm 3.4$ |
| | ArcShift-CVNN | `-tud` | – | – | 0.001 | 48 | $65.1 \pm 2.6$ | $65.2 \pm 1.9$ |
| Telegram | ArcShift | Native | 0.75 | – | 0.005 | 16 | $90.8 \pm 3.0$ | $88.2 \pm 5.0$ |
| | ArcShift-CVNN | `-tud` | – | – | 0.01 | 16 | $83.0 \pm 4.0$ | $81.4 \pm 6.0$ |

Table 9: Validation-selected ArcShift configurations for the directed-only comparison. All rows use the native directed graph. A dash denotes a parameter that does not apply.

| Dataset | Model | Selected $\beta$ | Learned $\gamma$ | Learning rate | Width | Validation (%) | Test (%) |
|---|---|---|---|---|---|---|---|
| Cornell | ArcShift | 1.00 | – | 0.005 | 48 | $84.9 \pm 4.7$ | $76.5 \pm 6.4$ |
| | ArcShift-CVNN | – | $0.268 \pm 0.021$ | 0.01 | 32 | $84.6 \pm 6.4$ | $81.6 \pm 6.8$ |
| Texas | ArcShift | 0.00 | – | 0.005 | 48 | $86.9 \pm 2.5$ | $82.7 \pm 5.9$ |
| | ArcShift-CVNN | – | $0.244 \pm 0.013$ | 0.01 | 48 | $89.0 \pm 4.2$ | $85.9 \pm 5.9$ |
| Wisconsin | ArcShift | 0.70 | – | 0.01 | 48 | $86.8 \pm 3.2$ | $83.3 \pm 4.9$ |
| | ArcShift-CVNN | – | $0.268 \pm 0.021$ | 0.01 | 32 | $85.1 \pm 3.0$ | $86.1 \pm 4.5$ |
| Cora-ML | ArcShift | 1.00 | – | 0.01 | 48 | $80.0 \pm 1.4$ | $80.3 \pm 1.9$ |
| | ArcShift-CVNN | – | $0.319 \pm 0.009$ | 0.001 | 48 | $77.4 \pm 1.7$ | $77.5 \pm 2.0$ |
| CiteSeer | ArcShift | 0.05 | – | 0.001 | 48 | $64.0 \pm 3.0$ | $64.1 \pm 2.7$ |
| | ArcShift-CVNN | – | $0.224 \pm 0.008$ | 0.001 | 48 | $63.7 \pm 2.7$ | $64.6 \pm 2.3$ |
| Telegram | ArcShift | 0.75 | – | 0.005 | 48 | $90.4 \pm 3.4$ | $88.6 \pm 4.0$ |
| | ArcShift-CVNN | – | $0.339 \pm 0.044$ | 0.005 | 48 | $82.6 \pm 4.4$ | $83.0 \pm 4.6$ |

**Mixed directed/undirected model-level comparison.** Table 8 reports the twelve validation-selected configurations for ArcShift and ArcShift-CVNN under the original MagNet benchmark setting, in which native directed and symmetrized (`-tud`) graph branches may be considered. ArcShift-CVNN initializes $\gamma_0 = 0.5$ in every run; the table instead reports learned $\gamma$ as the mean $\pm$ sample standard deviation across the ten split-specific checkpoints used for the reported results. For a `-tud` branch, the transport operator vanishes, so its directional coefficient is inoperative and is reported as a dash. The complete set of 84 selected model–dataset configurations is provided in the supplementary material.

**Directed-only comparison.** Table 9 reports the validation-selected ArcShift and ArcShift-CVNN configurations when evaluation is restricted to the native directed graph. ArcShift-CVNN uses $\gamma_0 = 0.5$; its learned $\gamma$ is reported as the mean $\pm$ sample standard deviation across the ten split-specific checkpoints used for the reported results.

**Directional-parameter search timing.** The timing experiment reported in Section 4.6 was conducted sequentially on one NVIDIA L40S GPU. Each model–dataset configuration used the native directed graph, ten predefined splits, two layers, $K = 1$, width 48, learning rate 0.01, dropout 0.5, weight decay $5 \times 10^{-4}$, and at most 3,000 epochs. ArcShift-CVNN used learned $\gamma$ initialized at 0.5.

Mean epoch time excludes the first five warm-up epochs. Prediction time is measured over 50 synchronized forward passes after 10 warm-up passes. For the fixed single-configuration timing comparison, ArcShift uses $\beta = 1.0$ and MagNet uses $q = 0.25$.

## REPRODUCIBILITY STATEMENT

Complete reproducibility settings and experimental details are provided in Appendix 6.5.

## AI USE STATEMENT

Large language models (LLMs) were used in this work solely for auxiliary purposes. Specifically, they assisted with manuscript editing and language refinement, terminology and presentation, literature exploration, and coding support. All research ideas, methodological developments, mathematical formulations, experimental design, analysis, and conclusions were independently developed and verified by the authors. The authors take full responsibility for the final content of the paper.