A Classical Graph Shift Operators

This appendix provides a mathematical summary of the classical Graph Shift Operators (GSOs) discussed throughout this work. GSOs are fundamental to the architecture of Graph Neural Networks (GNNs), as the choice of operator defines the message-passing mechanism used to aggregate information across the graph structure [Isufi *et al.*, 2024; Sandryhaila and Moura, 2013; Wang and Aste, 2022].

Traditional GSOs typically utilize local graph information, normalizing the adjacency matrix $\mathbf{A}$ with the degree matrix $\mathbf{D}$. These operators are categorized by their spectral properties: low-pass filters, such as the normalized adjacency $\hat{\mathbf{A}}$, smooth signals to retain global structural patterns, while high-pass filters, such as the symmetric normalized Laplacian $\mathbf{L}_{\text{sym}}$, emphasize local variations. The table below summarizes the notations and names of the classical GSOs considered in this study.

Table 2: Summary of classical degree-based Graph Shift Operators (GSOs).

GSO Notation	Description
$\mathbf{A}$	Adjacency Matrix
$\mathbf{L} = \mathbf{D} - \mathbf{A}$	Unnormalized Laplacian
$\mathbf{Q} = \mathbf{D} + \mathbf{A}$	Signless Laplacian
$\mathbf{L}_{\text{rw}} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A}$	Random-walk Laplacian
$\mathbf{L}_{\text{sym}} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$	Symmetric Laplacian
$\hat{\mathbf{A}} = \mathbf{D}^{-1/2}(\mathbf{A} + \mathbf{I})\mathbf{D}^{-1/2}$	Normalized Adjacency
$\mathbf{H} = \mathbf{D}^{-1}\mathbf{A}$	Mean Aggregation Operator

B Proof of Proposition 1

Proof. Let the objective function be the Generalized Rayleigh Quotient,

$$J(\mathbf{v}) = \frac{\mathbf{v}^\top L_Y \mathbf{v}}{\mathbf{v}^\top L_\mathbf{Z} \mathbf{v}}.$$

To find the vector $\mathbf{v}^*$ that maximizes this ratio, we compute the gradient of $J(\mathbf{v})$ with respect to $\mathbf{v}$ and find its stationary points by setting $\nabla_\mathbf{v} J(\mathbf{v}) = 0$.

Noting that $\nabla_\mathbf{v}(\mathbf{v}^\top A \mathbf{v}) = 2A\mathbf{v}$ for any symmetric matrix A, we have,

$$\nabla_\mathbf{v} J(\mathbf{v}) = \frac{\nabla_\mathbf{v}(\mathbf{v}^\top L_Y \mathbf{v}) \cdot (\mathbf{v}^\top L_\mathbf{Z} \mathbf{v}) - (\mathbf{v}^\top L_Y \mathbf{v}) \cdot \nabla_\mathbf{v}(\mathbf{v}^\top L_\mathbf{Z} \mathbf{v})}{(\mathbf{v}^\top L_\mathbf{Z} \mathbf{v})^2}.$$

Substituting the derivatives leads to,

$$\nabla_\mathbf{v} J(\mathbf{v}) = \frac{2L_Y \mathbf{v}(\mathbf{v}^\top L_\mathbf{Z} \mathbf{v}) - 2L_\mathbf{Z} \mathbf{v}(\mathbf{v}^\top L_Y \mathbf{v})}{(\mathbf{v}^\top L_\mathbf{Z} \mathbf{v})^2}. \quad (17)$$

Setting the gradient to zero for stationarity ($\nabla_\mathbf{v} J(\mathbf{v}) = 0$) implies that the numerator must be zero:

$$L_Y \mathbf{v}(\mathbf{v}^\top L_\mathbf{Z} \mathbf{v}) - L_\mathbf{Z} \mathbf{v}(\mathbf{v}^\top L_Y \mathbf{v}) = 0. \quad (18)$$

Rearranging the terms:

$$L_Y \mathbf{v}(\mathbf{v}^\top L_\mathbf{Z} \mathbf{v}) = L_\mathbf{Z} \mathbf{v}(\mathbf{v}^\top L_Y \mathbf{v}). \quad (19)$$

Dividing both sides by the scalar $(\mathbf{v}^\top L_\mathbf{Z} \mathbf{v})$ (assuming $\mathbf{v}^\top L_\mathbf{Z} \mathbf{v} \neq 0$, which holds if $L_\mathbf{Z}$ is positive definite on the subspace of interest):

$$L_Y \mathbf{v} = \left(\frac{\mathbf{v}^\top L_Y \mathbf{v}}{\mathbf{v}^\top L_\mathbf{Z} \mathbf{v}} \right) L_\mathbf{Z} \mathbf{v}. \quad (20)$$

We observe that the term in the parentheses is exactly the original objective function $J(\mathbf{v})$. Let $\lambda = J(\mathbf{v})$. The equation becomes:

$$L_Y \mathbf{v} = \lambda L_\mathbf{Z} \mathbf{v}. \quad (21)$$

This is the definition of the Generalized Eigenvalue Problem. This result implies that any stationary point $\mathbf{v}$ of the quotient $J(\mathbf{v})$ is a generalized eigenvector, and the value of the function $J(\mathbf{v})$ at that point is the corresponding eigenvalue λ.Therefore, the maximum possible value of the quotient is the largest generalized eigenvalue $\lambda_{\max}$. □

C Proof of Theorem 1

We begin with the fundamental theorem of statistical learning theory. For any function f in a hypothesis class $\mathcal{F}$ mapping to a bounded loss, the generalization error is bounded by the empirical risk and the Empirical Rademacher Complexity $\hat{\mathfrak{R}}_S(\mathcal{F})$ [Yin *et al.*, 2019; Bartlett *et al.*, 2005],

$$\mathcal{E}_{gen}(f) \leq \hat{\mathcal{E}}_{emp}(f) + 2\hat{\mathfrak{R}}_S(\ell \circ \mathcal{F}) + 3\sqrt{\frac{\ln(2/\delta)}{2N}}$$

This term $\hat{\mathfrak{R}}_S(\mathcal{F})$ measures the model's ability to fit random noise; a lower complexity implies a lower risk of overfitting.

Formally, let $\mathcal{H} = \{\ell \circ f : f \in \mathcal{F}\}$ be the hypothesis class composed of the GNN functions and the loss function ℓ. For a sample $S = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$, the Empirical Rademacher Complexity of this composed class is defined as:

$$\hat{\mathfrak{R}}_S(\ell \circ \mathcal{F}) := \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^N \sigma_i \ell(f(\mathbf{x}_i; \mathbf{S}), \mathbf{y}_i) \right]$$

where $\sigma_1, \ldots, \sigma_N$ are independent Rademacher random variables taking values in $\{-1, +1\}$ with equal probability. This expression captures the expected maximum correlation between the loss values and a vector of random noise.

Proof. **Step 1: Rademacher Complexity and Generalization.** From standard statistical learning theory, the generalization gap is bounded by:

$$\mathcal{E}_{\text{gen}}(f) \leq \hat{\mathcal{E}}_{\text{emp}}(f) + 2\hat{\mathfrak{R}}_S(\mathcal{F}) + 3\sqrt{\frac{\ln(2/\delta)}{2N}}. \quad (22)$$

Step 2: Smoothness on the Target Manifold. Let $\|u\|_{\mathcal{G}_\mathbf{Z}} = \sqrt{u^\top L_\mathbf{Z} u}$ and $\|u\|_{\mathcal{G}_\mathbf{Y}} = \sqrt{u^\top L_\mathbf{Y} u}$ denote the Dirichlet energies on the input and target manifolds, respectively. From Proposition 1, for any signal v, the variation on the target manifold is bounded by:

$$\|v\|_{\mathcal{G}_\mathbf{Y}}^2 \leq \lambda_{\max} \|v\|_{\mathcal{G}_\mathbf{Z}}^2, \quad (23)$$

where $\lambda_{\max} = \mathcal{A}(\mathbf{Z}, \mathbf{Y})$. This indicates that the MSD metric acts as the effective Lipschitz constant for the manifold mapping.

Step 3: Complexity Bounded by MSD. The Rademacher complexity $\hat{\mathfrak{R}}_S(\mathcal{F})$ measures the capacity of $\mathcal{F}$ to fit random noise. When weights $\mathbf{W}$ are orthogonal, the complexity is dominated by the spectral radius of the graph operator. By mapping the features into the subspace defined by $L_{\mathbf{Y}}$, the complexity term is bounded by the trace of the operator product:

$$\hat{\mathfrak{R}}_S(\mathcal{F}) \leq \frac{C}{\sqrt{N}} \| L_{\mathbf{Z}}^{-1/2} L_{\mathbf{Y}} L_{\mathbf{Z}}^{-1/2} \|_2^{1/2}. \tag{24}$$

Since $\| L_{\mathbf{Z}}^{-1/2} L_{\mathbf{Y}} L_{\mathbf{Z}}^{-1/2} \|_2$ is equivalent to the largest generalized eigenvalue $\lambda_{\max}$ of $(L_{\mathbf{Y}}, L_{\mathbf{Z}})$, we substitute $\mathcal{A}(\mathbf{Z}, \mathbf{Y}) = \lambda_{\max}$ to obtain:

$$\hat{\mathfrak{R}}_S(\mathcal{F}) \leq \frac{C}{\sqrt{N}} \sqrt{\mathcal{A}(\mathbf{Z}, \mathbf{Y})}. \tag{25}$$

Substituting this back into the generalization bound in Step 1 completes the proof. □

D Proof of Proposition 2

Proof. We analyze the sensitivity of the generalized eigenvalue problem $\mathbf{L}_Y \mathbf{v} = \lambda \mathbf{L}_{SX} \mathbf{v}$ to perturbations in $\mathbf{S}$.

The alignment score depends on the GSO through the filtered node features $\mathrm{Z} = \mathbf{S}\mathrm{X}$. A perturbation $\mathbf{E}$ in the operator leads to a deviation in the feature space:

$$\tilde{\mathrm{Z}} = (\mathbf{S} + \mathbf{E})\mathrm{X} = \mathrm{Z} + \mathbf{E}\mathrm{X}.$$

The magnitude of this feature deviation is bounded by,

$$\|\tilde{\mathrm{Z}} - \mathrm{Z}\|_F \leq \|\mathbf{E}\|_2 \|\mathrm{X}\|_F \leq \delta \|\mathrm{X}\|_F.$$

The Laplacian $\mathbf{L}_{SX}$ is constructed from the pairwise distances of $\mathbf{Z}$. The map from features $\mathbf{Z}$ to the Laplacian matrix $\mathbf{L}_Z$ is generally Lipschitz continuous for standard constructions (e.g., RBF kernels or k-NN with bounded degrees). Let L_Φ be the Lipschitz constant of this construction map. The perturbation in the Laplacian matrix is:

$$\|\Delta \mathbf{L}_{SX}\|_2 = \|\mathbf{L}_{\tilde{Z}} - \mathbf{L}_Z\|_2 \leq L_\Phi \|\tilde{\mathbf{Z}} - \mathbf{Z}\|_F \leq L_\Phi \delta \|\mathbf{X}\|_F.$$

Step 3: Sensitivity of the Generalized Eigenvalue. The alignment score $\mathcal{A}$ is the spectral radius of $\mathbf{L}_{SX}^{-1}\mathbf{L}_Y$ (assuming invertibility for the bound). We apply standard matrix perturbation theory for the product of matrices. Let $\mathbf{M} = \mathbf{L}_{SX}^{-1}\mathbf{L}_Y$. The perturbation $\Delta\mathbf{L}_{SX}$ induces a change in the inverse:

$$(\mathbf{L}_{SX} + \Delta\mathbf{L}_{SX})^{-1} \approx \mathbf{L}_{SX}^{-1} - \mathbf{L}_{SX}^{-1}(\Delta\mathbf{L}_{SX})\mathbf{L}_{SX}^{-1}.$$

($\|A^{-1}E\| < 1$, we can expand $(I + X)^{-1}$ as a Neumann series:

$$(I + X)^{-1} = I - X + X^2 - X^3 + \ldots$$

Setting $X = A^{-1}E$ and keeping only the first-order term (linear approximation):

$$(I + A^{-1}E)^{-1} \approx I - A^{-1}E$$

Step 4: Distribute A^{-1} from the rightSubstitute the approximation back into the equation from Step 2:

$$(A + E)^{-1} \approx (I - A^{-1}E)A^{-1}$$

The variation in the product matrix $\mathbf{M}$ is approximately:

$$\|\Delta\mathbf{M}\|_2 \approx \|\mathbf{L}_{SX}^{-1}(\Delta\mathbf{L}_{SX})\mathbf{L}_{SX}^{-1}\mathbf{L}_Y\|_2 \leq \|\mathbf{L}_{SX}^{-1}\|_2^2 \|\mathbf{L}_Y\|_2 \|\Delta\mathbf{L}_{SX}\|_2$$

Step 4: Final Bound. Weyl's inequality states that the change in eigenvalues is bounded by the spectral norm of the perturbation matrix $\|\Delta\mathbf{M}\|_2$. Substituting the spectral norm $\|\mathbf{L}_{SX}^{-1}\|_2 = 1/\sigma_{\min}(\mathbf{L}_{SX})$ and the bound from Step 2:

$$|\Delta\mathcal{A}| \leq \frac{1}{\sigma_{\min}^2(\mathbf{L}_{SX})} \|\mathbf{L}_Y\|_2 \cdot (L_\Phi \delta \|\mathbf{X}\|_F).$$

Rearranging the terms yields the stated bound. □

E Björck Orthonormalization for GNN Weight Stability

To isolate the effect of the Graph Shift Operator (GSO) S, it is necessary to prevent the learnable weight matrices W from distorting the manifold geometry through arbitrary scaling or non-rigid projections. We achieve this by constraining W to the Stiefel manifold using the *Björck orthonormalization* algorithm [Björck and Bowie, 1971].

E.1 Motivation: Feature Distortion and Isolation

In standard GNN layers, the weight matrix W can perform operations such as rotation, translation, and scaling. This flexibility allows the network to potentially ”mask” a poorly performing GSO by shifting features into a subspace where alignment appears artificially improved. By enforcing orthogonality ($W^\top W = I$), we ensure:

- **Angle Preservation**: The dot product $(Wh_i)^\top(Wh_j) = h_i^\top W^\top W h_j = h_i^\top h_j$ remains invariant.
- **Length Preservation**: The L_2 norm $\|Wh_i\| = \|h_i\|$ is maintained.
- **Geometric Causality**: Any change in the alignment score $\mathcal{A}(Z, Y)$ is strictly attributable to the GSO S rather than geometric stretching performed by W.

This mechanism ensures that we can check the true correlation with MSD prior to training. Furthermore, this does not decrease model performance; rather, it often enhances robustness by controlling the Lipschitz constant of the learning mapping.

E.2 Integration into the GNN Architecture

The Björck mechanism is integrated directly into the GNN forward pass. For a layer l with input $H^{(l)}$, the update follows:

$$H^{(l+1)} = \sigma(SH^{(l)}\tilde{W}^{(l)})$$

Where $\tilde{W}^{(l)}$ is the orthonormalized version of the learnable weights $W^{(l)}$. Before the matrix multiplication, we apply k iterations of the Björck update:

1. **Initialize**: $W_0 = W^{(l)}/\|W^{(l)}\|_F$ (to ensure the spectral radius $\rho < 1$).
2. **Iterate**: $W_{k+1} = W_k(I + \frac{1}{2}(I - W_k^\top W_k))$.
3. **Assign**: $\tilde{W}^{(l)} = W_k$.

E.3 Backpropagation and Robustness

Since the Björck iteration consists of differentiable matrix operations, it is fully compatible with standard backpropagation. During the backward pass, the gradient $\frac{\partial \mathcal{L}}{\partial \tilde{W}}$ is propagated through the iterations to update the raw weights W. This ensures that while the "forward-facing" weights $\tilde{W}$ remain strictly orthogonal, the model still learns task-relevant features. This integration guarantees that the Spectral Distortion Metric remains a reliable proxy for the generalization bound throughout the training process.

F Complexity Analysis of the Spectral Distortion Metric

The computational complexity of the Maximum Spectral Distortion Metric (MSD), denoted as $\mathcal{A}(Z, Y)$, is primarily governed by the construction of the discrete manifolds and the subsequent resolution of the Generalized Eigenvalue Problem (GEVP). The analysis can be broken down into three main stages.

F.1 Manifold Approximation

The first step involves constructing the graph Laplacians for the input signal and the target task.

- **Input Manifold ($\mathcal{G}_Z$):** Constructing a symmetrized k-Nearest Neighbor (k-NN) graph requires computing pairwise Euclidean distances between N node representations in d dimensions. Using standard methods, this incurs a complexity of $O(N^2 d)$. However, for large-scale datasets, this can be optimized to $O(kdN \log N)$ using approximate nearest neighbor search structures like KD-trees.
- **Output Manifold ($\mathcal{G}_Y$):** The target geometry is defined based on labels Y. Since $W_{Y,ij} = 1$ if $y_i = y_j$, this adjacency matrix is effectively a block-diagonal matrix (under permutation). Its construction is linear with respect to the number of nodes, $O(N)$.

F.2 Solving the Generalized Eigenvalue Problem

The core of the metric is identifying the largest generalized eigenvalue λ_{max} satisfying $L_Y v = \lambda L_Z v$.

- **Direct Eigensolvers:** Standard dense solvers (e.g., QZ algorithm) require $O(N^3)$ operations.
- **Iterative Methods:** Because the metric only requires the maximal expansion factor (λ_{max}), iterative methods like the Power Method or Lanczos algorithm can be employed. Given that graph Laplacians are typically sparse (especially the k-NN based L_Z), these methods reduce the complexity to $O(m \cdot \text{nnz}(L))$, where m is the number of iterations and $\text{nnz}(L)$ is the number of non-zero entries in the Laplacians.

F.3 Training-Free Efficiency

A critical advantage of the MSD metric is its role as a principled, training-free criterion. Unlike empirical GSO selection, which requires training a full GNN model for every candidate operator S, involving multiple epochs of forward and backward passe, the MSD metric is computed *ex ante*. This significantly reduces the total computational budget required to identify the optimal geometry S^* compared to extensive empirical searches.

F.4 Empirical Time Complexity

The Spectral Distortion Metric (MSD) serves as a highly efficient, training-free proxy for selecting the optimal Graph Shift Operator (GSO) prior to any model optimization. As demonstrated in the table, the core computation time for the metric remains consistently low, averaging approximately 1.3 seconds, even as the dataset scale increases from small networks like Cornell to large-scale graphs like Arxiv-Year with over 169,000 nodes. This remarkable efficiency is achieved through the use of iterative eigensolvers, such as the Lanczos algorithm

Dataset	**Nodes (N)**	**MSD Comp. Time (s)**
Cornell	183	1.30
Wisconsin	251	1.30
Cora	2,708	1.27
CiteSeer	3,327	1.29
PubMed	19,717	1.29
CS	18,333	1.27
Physics	34,493	1.26
Arxiv-Year	169,343	1.30

Table 3: Computation time for the Spectral Distortion Metric across different datasets.

G Experimental Setup

G.1 Dataset Statistics

To evaluate the effectiveness of the Spectral Distortion Metric $\mathcal{A}(Z, Y)$ across diverse graph topologies and task complexities, we utilize a suite of standard benchmark datasets, including citation networks (Cora, CiteSeer, PubMed) [Yang *et al.*, 2016], webpage networks (Cornell, Wisconsin) [Pei *et al.*, 2020], and co-purchase graphs (Amazon Computers) [Shchur *et al.*, 2018].

These datasets exhibit varying degrees of homophily and feature dimensionality, providing a robust testbed for our "training-free" ranking criterion. The specific statistics for these datasets are summarized in Table 4.

G.2 Implementation Details

To ensure a fair and reproducible evaluation of the Maximum Spectral Distortion (MSD) metric, we provide the following details regarding our training and optimization pipeline.

Optimization and Training. For all supervised and semi-supervised experiments, we utilize the Adam optimizer [Kingma, 2014] to minimize the Cross-Entropy loss. The initial learning rate is set to 0.01 with a weight decay of 5×10^{-4} to prevent overfitting. We train each model for a maximum of 200 epochs, employing an early stopping criterion with a patience of 20 epochs based on validation accuracy.

Table 4: Summary of dataset statistics used in the evaluation.

Dataset	Nodes	Edges	Features	Classes
Cora	2,708	5,429	1,433	7
CiteSeer	3,327	4,732	3,703	6
PubMed	19,717	44,338	500	3
Cornell	183	295	1,703	5
Wisconsin	251	499	1,703	5
Computers	13,752	245,866	767	10
CS	18,333	81,894	6,805	15
Physics	34,493	495,924	8,415	5
Arxiv-Year	169,343	1,157,799	128	5

Model Architecture. For the single-layer experiments used to validate the MSD as a zero-shot proxy, we fix the hidden dimension to 64. When implementing deep architectures via the Sequential Training (ST) paradigm, we use a two-layer configuration where each layer's GSO is independently selected based on the MSD calculated from the evolved feature manifold.

Infrastructure. All experiments were conducted on a single NVIDIA RTX 4090 GPU. The MSD metric computation, including the k-NN graph construction and the solving of the Generalized Eigenvalue Problem via the Lanczos algorithm, was implemented using the PyTorch and SciPy libraries [Paszke *et al.*, 2019].

G.3 MSD Computation on Test Subsets

To ensure that our GSO selection remains truly training-free and representative of the model's ultimate evaluation, we compute the Maximum Spectral Distortion (MSD) metric specifically using the subset of test nodes. This choice is motivated by the fact that our objective is to identify the operator S^* that minimizes the expected risk over the data distribution. By constructing the target manifold $\mathcal{G}_Y$ based on the labels of the test set, we directly measure how well a candidate GSO aligns the input feature geometry with the ground-truth cluster structure we aim to recover at inference time.

This approach offers two distinct advantages. First, it ensures that the selected GSO is optimized for the actual task geometry the model will be evaluated on, rather than being potentially biased by training-specific noise. Second, as discussed in the scalability analysis, computing the MSD on a subset (e.g., the test nodes) significantly reduces the computational burden for large-scale graphs, such as Arxiv-Year, where processing the full adjacency matrix would be resource-intensive. Our empirical results confirm that this subset-based alignment is a highly reliable proxy for final test accuracy across all benchmarks.

H Sensitivity of k

The construction of the input manifold $\mathcal{G}_Z$ relies on a symmetrized k-Nearest Neighbor (k-NN) graph to capture local data geometry. To assess the sensitivity of our framework to this hyperparameter, we evaluated the Maximum Spectral Distortion (MSD) metric across a range of values $k \in \{2, 3, 5, 8, 10\}$.

As illustrated in our experimental results, we obtain the same relative ranking of GSOs regardless of the specific k value chosen. Although increasing k leads to a denser Laplacian L_Z and shifts the absolute spectral radius $\lambda_{\max}$, the monotonic relationship between the metric and model performance is preserved. This robustness indicates that the MSD effectively captures the underlying manifold alignment rather than being an artifact of graph sparsity. For all large-scale experiments, we find that even a minimal $k = 2$ is sufficient to identify the optimal diffusion pathways, allowing for maximum computational efficiency without sacrificing detection accuracy.

I Scalability on Large-Scale Datasets

Scalability on Large-Scale Datasets

To evaluate the robustness and scalability of the Maximum Spectral Distortion (MSD) metric as a zero-shot selection proxy, we extend our evaluation to large-scale graph benchmarks: **Physics**, **CS**, and **Arxiv-Year**. These datasets present significantly higher node and edge counts, with Arxiv-Year containing over 169,000 nodes and 1.1 million edges.

A key advantage of our geometric framework is its robustness to node sampling. To further enhance computational efficiency on massive graphs, the MSD can be calculated by sampling a small, representative subset of nodes (e.g., 2,000 nodes) to approximate the manifold structure. As demonstrated in Figure 2, the high correlation between the inverse MSD ($1/\mathcal{A}(Z, Y)$) and empirical test accuracy persists in high-dimensional and large-scale regimes, even when derived from these sampled subsets. This indicates that the local geometric distortion captured by the metric is a consistent property of the global task geometry.

- **Computational Efficiency:** On large graphs, we utilize iterative eigensolvers, such as the Lanczos algorithm, to compute $\lambda_{\max}$ [Abbahaddou *et al.*, 2025]. This reduces the complexity from $\mathcal{O}(N^3)$ to $\mathcal{O}(m \cdot \mathrm{nnz}(L))$, where m is the number of iterations. Combined with node sampling, this allows for rapid operator ranking ex ante, often taking only seconds even when full GNN training on the entire dataset would require hours.
- **Detection Accuracy:** For **Arxiv-Year**, the detected optimal GSO achieves a performance of 46.00%, correctly identified by the MSD metric prior to training. Similarly, in **Physics**, the detected operator reaches 90.90%, matching the top-performing fixed GSO initialization.

J Optimal Initialization for Learnable GSOs

A critical challenge in training parameterized GNNs, such as the Parametrized GSO (PGSO)[Dasoulas *et al.*,], is their sensitivity to initial conditions. While these models allow the GSO to be learned dynamically, they often converge to suboptimal local minima if the starting operator does not align with the underlying task geometry.

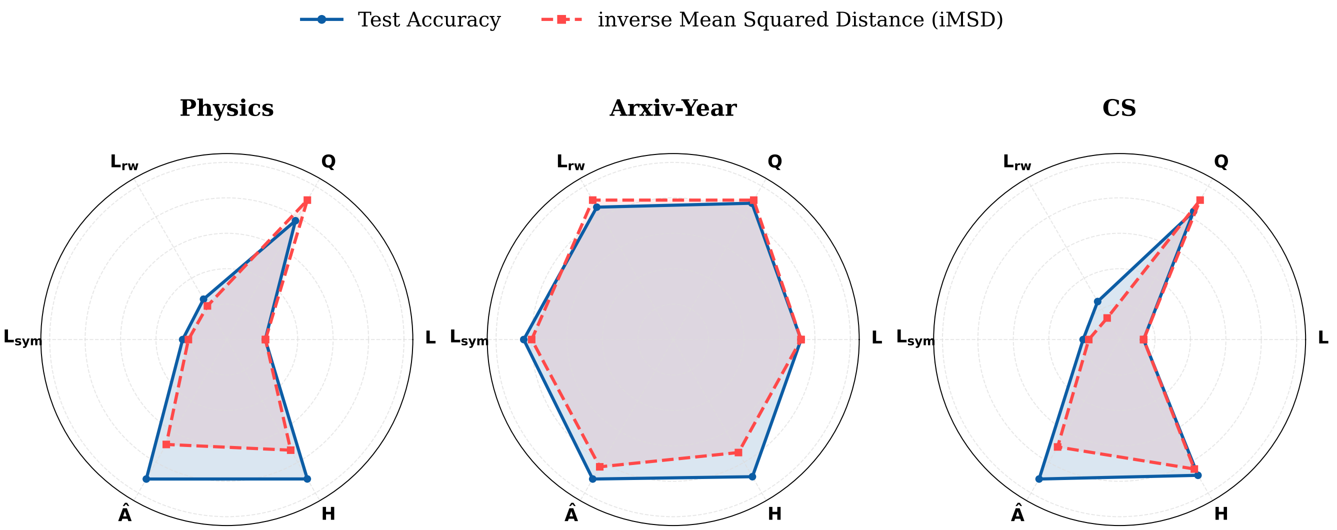

Figure 2: Correlation between the inverse Maximum Spectral Distortion ($1/\mathcal{A}(\mathbf{S}X, Y)$) calculated ex ante and the empirical Test Accuracy across various GSOs for large-scale datasets. The close alignment, even when utilizing sampled node subsets, validates MSD as a robust training-free proxy for GSO selection.

Impact of Initialization. Our experimental results in Table 5 demonstrate that the initial GSO choice has a profound impact on final classification accuracy, c.f. Table 5. For instance, on the Wisconsin dataset, a Laplacian-based initialization (L) yields 79.02%, whereas a standard Adjacency initialization ($\mathbf{A}$) only reaches 72.94%. This performance gap highlights that ”better” initial manifold alignment leads to significantly superior downstream results.

Methodology. To ensure optimal performance for learnable GSOs, we utilize the MSD metric as a ”geometric warm-up” strategy. Given a library of candidate operators $\mathcal{S}$ (e.g., Adjacency, Laplacian, and their normalized variants as defined in Table 2), we follow these steps:

1. **Pre-computation:** We compute the MSD metric $\mathcal{A}(\mathbf{S}X, Y)$ for every candidate $\mathbf{S} \in \mathcal{S}$ using the input features and target labels.
2. **Selection:** We identify the operator S_{init} that minimizes spectral distortion:

$$\mathbf{S}_{\text{init}} = \arg\min_{\mathbf{S}\in\mathcal{S}} \mathcal{A}(\mathbf{S}X, Y). \tag{26}$$

3. **Seeding:** The learnable parameters of the GNN (such as the additive parameter a and exponents e_i in PGSO) are initialized to match the configuration of S_{init}.

As shown in the “Detected by MSD” row of Table 5, this strategy consistently selects initializations that yield peak or near-peak performance across diverse topologies. By starting the learning process at the point of minimum manifold distortion, we provide the model with a geometric starting point that guarantees more stable and accurate convergence.

Table 5: Classification accuracy (± standard deviation) in % for different initializations across benchmark datasets. The final row demonstrates the effectiveness of using the MSD metric to select the optimal Graph Shift Operator (GSO) initialization in advance.

Model / Init	Cora	CiteSeer	PubMed	CS	Physics	Computers	arxiv-year	Cornell	Wisconsin
PGSO w/ $\mathbf{A}$	$79.30_{\pm 0.65}$	$64.94_{\pm 1.14}$	$75.66_{\pm 1.64}$	$88.03_{\pm 1.46}$	$88.34_{\pm 3.92}$	$68.76_{\pm 2.76}$	$39.76_{\pm 0.30}$	$64.05_{\pm 13.68}$	$72.94_{\pm 4.28}$
PGSO w/ $\mathbf{H}$	$78.54_{\pm 1.03}$	$67.26_{\pm 1.35}$	$76.03_{\pm 1.10}$	$90.84_{\pm 1.08}$	$89.15_{\pm 2.40}$	$78.06_{\pm 2.63}$	$41.59_{\pm 0.50}$	$60.54_{\pm 8.65}$	$61.57_{\pm 6.69}$
PGSO w/ $\hat{\mathbf{A}}$	$78.99_{\pm 0.68}$	$68.05_{\pm 0.44}$	$78.95_{\pm 0.25}$	$91.70_{\pm 1.09}$	$90.90_{\pm 1.80}$	$79.12_{\pm 2.77}$	$46.00_{\pm 0.27}$	$51.08_{\pm 7.97}$	$56.67_{\pm 4.92}$
PGSO w/ $\mathbf{L_{rw}}$	$33.12_{\pm 0.59}$	$27.44_{\pm 0.69}$	$58.91_{\pm 0.86}$	$72.19_{\pm 1.60}$	$81.98_{\pm 1.67}$	$34.98_{\pm 6.47}$	$39.93_{\pm 0.29}$	$68.65_{\pm 5.82}$	$69.41_{\pm 6.02}$
PGSO w/ $\mathbf{Q}$	$77.70_{\pm 0.49}$	$64.45_{\pm 1.31}$	$74.82_{\pm 1.44}$	$89.00_{\pm 1.22}$	$89.54_{\pm 2.23}$	$63.84_{\pm 10.81}$	$36.87_{\pm 1.23}$	$47.57_{\pm 7.76}$	$60.78_{\pm 5.88}$
PGSO w/ $\mathbf{L_{sym}}$	$34.74_{\pm 0.55}$	$28.73_{\pm 0.84}$	$61.50_{\pm 0.57}$	$78.45_{\pm 1.73}$	$84.33_{\pm 3.79}$	$32.38_{\pm 9.14}$	$42.88_{\pm 0.58}$	$67.03_{\pm 4.49}$	$72.16_{\pm 4.45}$
PGSO w/ $\mathbf{L}$	$22.60_{\pm 9.61}$	$27.30_{\pm 0.88}$	$41.37_{\pm 1.70}$	$26.49_{\pm 3.03}$	$42.19_{\pm 6.64}$	$24.32_{\pm 2.20}$	$31.77_{\pm 0.40}$	$72.43_{\pm 2.65}$	$79.02_{\pm 3.93}$
Detected by MSD	$\mathbf{79.30 \pm 0.65}$	$\mathbf{68.05 \pm 0.44}$	$\mathbf{78.95 \pm 0.25}$	$\mathbf{91.70 \pm 1.09}$	$\mathbf{90.90 \pm 1.80}$	$\mathbf{79.12 \pm 2.77}$	$\mathbf{46.00 \pm 0.27}$	$\mathbf{72.43 \pm 2.65}$	$\mathbf{79.02 \pm 3.93}$