Random Probing for Structural Self-Interactions in Graph Neural Networks

Saku Peltonen[1], H. Çağrı Bilgi[2], and Kubilay Atasu[2]

[1] ETH Zurich, Switzerland speltonen@ethz.ch
[2] Delft University of Technology, Netherlands

Abstract. Graph Neural Networks (GNNs) fundamentally lack the ability to capture structural properties defined by node self-interactions, such as cycle counts, subgraph centrality, and heat kernel signatures. Existing solutions often rely on computationally expensive higher-order architectures or hand-crafted features. In this work, we introduce DIAL, a message-passing layer that gives nodes access to graph structure through the diagonal of graph-derived operators. Our approach uses randomized probing to provide nodes with learnable, permutation equivariant access to diagonal entries. We prove that our approach can express important structural features, including cycle counts, subgraph centrality, and heat kernel signatures. The proposed method can be integrated directly into message passing layers. Empirical evaluations demonstrate that this approach improves the structural expressivity of GNNs while being computationally efficient compared to higher-order baselines. Submission type: novel research paper.

Keywords: GNNs · Structural embeddings · Spectral graph filters.

1 Introduction

Graph Neural Networks (GNNs) have been successful in many applications. However, they are known to have limited expressive power, often failing to distinguish basic structures or count fundamental motifs like cycles [28]. Higher-order GNN architectures can partially address these limitations, but often at a significant computational cost [21]. In practice, one often resorts to hand-crafted structural features, such as cycle or motif counts [3]. While effective, this approach is brittle and requires choosing the relevant structures in advance.

Many structural node properties that message passing fails to capture can be described in terms of *self-interactions* of nodes. Intuitively, these interactions quantify how information originating from a node flows through the graph and returns to it. Standard GNNs cannot capture such patterns—for example closed walk counts—as they require isolating information that originates from a node and later returns to it. This concept naturally generalizes to subgraph centrality, which characterizes a node's global importance by weighing its closed walk participation. In the spectral domain, Laplacian-based self-interactions cover graph

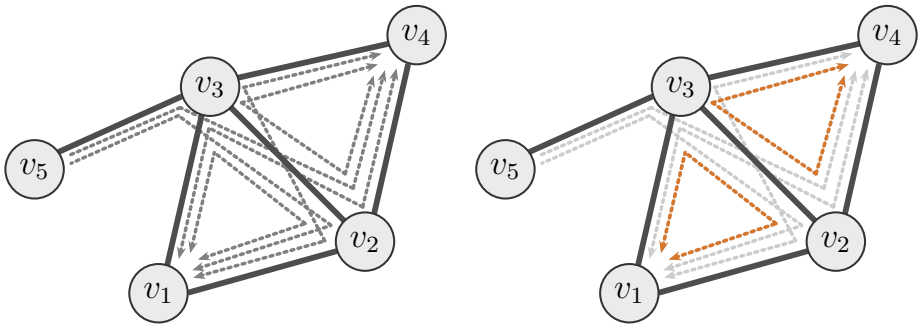

Fig. 1. Message-passing view of Hutchinson's diagonal estimator. We illustrate triangle counting for all nodes simultaneously using random probes and ordinary message passing. **Left:** The arrowed paths show a subset of length-3 messages received at v_1 and v_4. After propagation, node v observes $(A^3 z)_v = \sum_u (A^3)_{vu} z_u$, which mixes self-returning messages with messages originating elsewhere. Without identifying information, standard message passing cannot isolate the diagonal term. **Right:** Hutchinson's estimator correlates the propagated value with the original probe z_v. Since $\mathbb{E}[z_v z_u] = 0$ for $u \neq v$ and $\mathbb{E}[z_v^2] = 1$, the non-diagonal terms vanish in expectation, yielding $\mathbb{E}[z_v (A^3 z)_v] = (A^3)_{vv}$. For an adjacency matrix, this diagonal entry counts length-3 closed walks at v, i.e., twice the number of triangles incident to v.

angles (Figure 2), as well as diffusion dynamics like the heat kernel signature, which measures heat return rates over time.

Mathematically, these notions can all be expressed as functions of a Graph Shift Operator (GSO) S, such as the adjacency matrix A or Laplacian L. Crucially, all such node-level self-interactions correspond to the *diagonal entries* of these GSO-derived operators $M = f(S) \in \mathbb{R}^{n \times n}$. In practice, f is often approximated by low-degree polynomials of the GSO, $M = \sum_{k=0}^{K-1} \alpha_k S^k$. This naturally aligns with message passing architectures, which implicitly compute GSO polynomials through neighborhood aggregation. However, standard message passing GNNs cannot express any of the self-interaction based structural properties above. The aggregation mechanism mixes information from different nodes, failing to isolate self-interactions.

Our key observation is that message passing algorithms can access diagonal entries in a permutation equivariant way using a randomized probing mechanism. With a stochastic approach based on the Hutchinson diagonal estimator [13], nodes can efficiently approximate the diagonal of GSO powers, and hence the diagonal of any graph-derived operator, through simple message passing steps. This mechanism allows nodes to capture structural properties, such as cycle counts, within a fully learnable framework, avoiding rigid hand-crafted features and expensive higher-order architectures. See Figure 1 for an example with counting triangles.

The resulting architecture, which we call DIAL, can be integrated into any standard GNN and is permutation equivariant in expectation. Adjusting the number of random probes allows a trade-off between computational cost and estimation accuracy. We prove DIAL can express a variety of important structural properties, including cycle counts, graph angles, subgraph centrality, and heat kernel signatures.

Empirically, we evaluate DIAL on molecular property prediction on ZINC, and on synthetic node-level pattern counting tasks. The results show that DIAL is competitive with spectral and random-probe positional encoding methods on ZINC, and improves over other random-probe methods on cycle counting while remaining substantially cheaper than specialized subgraph and higher-order architectures.

2 Preliminaries

A graph is denoted $G = (V, E)$ with $|V| = n$. A graph shift operator (GSO) is a matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$. Common choices include the adjacency matrix $\mathbf{A}$, normalized adjacency $\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, the Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D}$ is the degree matrix, or the normalized Laplacian. We consider message-passing layers of the form

$$\mathbf{X}^{(\ell)} = \sigma\Big(\sum_{k=0}^{K-1} \mathbf{S}^k \mathbf{X}^{(\ell-1)} \mathbf{H}_k^{(\ell)} \Big), \tag{1}$$

where σ is a nonlinearity and $\mathbf{H}_k^{(\ell)}$ are learnable weights. The parameter K controls the maximum hop distance used for aggregation (specifically $K - 1$), with $K = 2$ for typical GNNs such as GCN.

For the following we assume the graph is undirected, making the GSO symmetric. Let $\mathbf{S} = \Phi \Lambda \Phi^{-1}$ be the eigendecomposition of the GSO $\mathbf{S}$, where $\Lambda = \mathsf{diag}(\lambda_1, \ldots, \lambda_n)$ is the diagonal matrix of eigenvalues and $\Phi = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_n]$ is the matrix of eigenvectors. For symmetric GSOs (undirected graphs), Φ is orthonormal, i.e. $\Phi^{-1} = \Phi^T$.

Each eigenvalue μ defines an eigenspace E_μ spanned by the eigenvectors with eigenvalue μ. Let $I_\mu = \{i : \lambda_i = \mu\}$ denote the corresponding index set and let $\Phi_\mu \in \mathbb{R}^{n \times |I_\mu|}$ be the matrix whose columns are these eigenvectors. The projector onto E_μ is $P_\mu = \Phi_\mu \Phi_\mu^T$. These projections allow the GSO to be expressed as a sum over eigenspaces: $S = \sum_\mu \mu P_\mu$, where the sum is over distinct eigenvalues.

Graph angles describe how nodes relate to these eigenspaces. For a node v, let $\mathbf{e}_v$ denote its indicator vector. The graph angle of v with respect to E_μ is

$$\angle_\mu(v) = (P_\mu)_{vv} = \|P_\mu \mathbf{e}_v\|^2 = \sum_{i \in I_\mu} (\boldsymbol{\phi}_i(v))^2.$$

The term *graph angle* comes from the geometric interpretation

$$(P_\mu)_{vv} = \cos^2(\theta_{v,\mu}), \qquad \theta_{v,\mu} = \angle(\mathbf{e}_v, E_\mu).$$

Unlike individual eigenvector coordinates, graph angles are invariant to sign changes and changes of basis inside eigenspaces.

Graph filters are operators of the form $p(S)$, where $p : \mathbb{R} \to \mathbb{R}$ is applied to the GSO S. For symmetric GSOs, spectral filters can be written as

$$p(S) = \Phi p(\Lambda)\Phi^T = \sum_{\mu} p(\mu)P_{\mu}.$$

A polynomial filter is the special case where $p(\mu) = \sum_{k=0}^{K-1} a_k \mu^k$, equivalently

$$p(S) = \sum_{k=0}^{K-1} a_k S^k.$$

The linear propagation part of Equation (1) naturally implements polynomial filters of the GSO.

3 Related Work

Expressivity of GNNs. Standard message passing architectures are limited in their expressiveness. In particular, they are at most as powerful as the Weisfeiler-Leman test [28] in distinguishing non-isomorphic graphs. This limitation surfaces in the inability of message-passing GNNs to count many basic substructures [5]. For example, standard GNNs cannot count triangles [21] or longer cycles, despite the practical relevance of this information in several applications, such as molecular property prediction [27]. From the perspective of this work, this limitation stems from the fact that message passing aggregates information arriving from neighboring nodes, but does not keep track of whom that information is originally coming from. Consequently, message passing cannot isolate self-interactions such as cycles.

There have been many works proposing more expressive GNN architectures [21,18,5,22,26]. Higher-order GNNs operate on tuples of nodes, making richer structural interactions available to the model [21]. However, this comes at a cost growing polynomially with the tuple order. Other approaches increase expressivity by explicitly augmenting node representations with counts of selected substructures [3,5]. While effective, these methods require choosing the relevant motifs in advance. In contrast to these approaches, our method exposes self-interaction information through randomized diagonal estimation. This allows a standard message passing backbone to access structural information without higher-order tuples or preselecting a fixed set of motifs.

Random Features in GNNs. Random node features have been studied as a way to overcome expressivity limitations in message passing [7,1,23]. These works show that random initialization can break node symmetries and increase expressive power, with universality results in some settings [24]. However, this relies on the model learning to use randomness as node identifiers, and the theoretical power

does not directly provide stable, task-relevant structural quantities. Hence, more direct uses of random features have been proposed [14,15,10].

R-PEARL [15] introduces a randomized variant of a positional encoding framework. While effective for PEs, its expressivity guarantees are limited.[3]

Eliasof et al. [10] use message passing with random features to learn positional encodings. They show that, combined with appropriate normalization, random feature propagation simulates power iteration and subspace iteration. This allows approximating the leading eigenvectors of the graph shift operator. They also note that random features can be used to estimate cycle counts using Hutchinson's trace estimator.[4] However, the focus of this work is on positional encodings, and their analysis is based on what can be achieved in the limit. In contrast, we focus on structural self-interactions and explicitly correlate propagated signals with the input randomness. This correlation is essential: it yields an unbiased estimator of diagonal entries in expectation and provides a stable structural signal that can be learned from finite random probes.

Spectral Graph Theory and GNNs. Spectral approaches to graph learning are rooted in spectral graph theory and graph signal processing [6]. Graph signal processing interprets node features as signals on the graph and defines filters as functions of a graph shift operator [25]. Classical spectral filters are typically fixed or analytically chosen.

Spectral GNNs learn such filters from data. Early spectral GNNs implemented filters directly in the Fourier basis, requiring an eigendecomposition [4]. ChebNet [8] introduced Chebyshev polynomial approximations. GCN can be viewed as a first-order approximation of this spectral filtering perspective [16]. These methods apply graph-derived operators to node features, but the resulting node representations do not explicitly expose the diagonal entries of those operators. Our method complements spectral GNNs by showing how randomized probing can make these diagonal entries accessible to message passing.

4 Main Tool: Diagonal Estimation via Message Passing

This section presents our main technical tool: a randomized technique to estimate the diagonal entries of a matrix via message passing. It is based on Hutchinson's diagonal estimator [13], which is formalized in the next lemma:

Lemma 1. *Let $M \in \mathbb{R}^{n\times n}$. Let $z \in \mathbb{R}^n$ be a random vector with i.i.d. Rademacher entries (i.e., each entry is independently $+1$ or -1 with probability $1/2$). Then*

$$\mathbb{E}[z \odot (Mz)] = \mathsf{diag}(M)$$

where $\odot$ is the elementwise product. The variance of the ith entry is $\sum_{j\neq i} M_{ij}^2$.

[3] The expressivity guarantees come mainly from [14], which uses random features for substructure counting. We defer a more detailed discussion to Appendix D.

[4] We omit quantitative results for Random Feature Propagation because no public implementation was available.

Proof. The ith entry is $\mathbb{E}[z_i(Mz)_i] = \mathbb{E}[\sum_j M_{ij} z_i z_j]$. Here, $\mathbb{E}[z_i z_j] = 0$ for $i \neq j$ since $\mathbb{E}[z_i] = 0$ and z_i, z_j are independent, making all cross terms vanish. Thus, only the $j = i$ term remains, giving $\mathbb{E}[z_i^2] M_{ii} = M_{ii}$. The variance calculation is deferred to the appendix. □

In the context of GNNs, the matrix M can be the GSO or a matrix derived from it, such as powers or spectral filters. Mz corresponds to a message passing operation on the graph with input features z.

4.1 Warm-up: Cycle Counting via Diagonal Estimation

Consider a message passing model of the form

$$X^{(l)} = \sigma(AX^{(l-1)}W) \tag{2}$$

Ignoring nonlinearities and weights, after k layers, we have $X^{(k)} = A^k X^{(0)}$. We illustrate how diagonal estimation can be used to compute cycle counts of the graph:

Lemma 2. *With a GNN in Equation 2 and uniformly random initial features $X^{(0)} \in \{-1, 1\}^n$, each node v can compute the number of closed walks of length k starting and ending at v in expectation.*

Proof. Initialize node features as independent Rademacher random variables, $z \in \{-1, 1\}^n$. By Lemma 1, $\mathbb{E}[z \odot (A^k z)] = \mathsf{diag}(A^k)$. It is well known that $(A^k)_{vv}$ counts the number of closed walks of length k starting and ending at node v (see Lemma 6). □

4.2 DIAL: Diagonal Access Layer

Building on Lemma 1, we propose the Diagonal Access Layer (DIAL), a stochastic message-passing module that gives nodes access to structural quantities through diagonal entries of graph-derived operators. DIAL is based on propagating random features, and correlating the propagated probes with the original random values. The technique is general and can be applied on directed and undirected graphs for any GNN architecture.

Random probes For initial features, we sample a probe vector $\mathbf{z} \in \{-1, 1\}^n$ with i.i.d. Rademacher entries.

Probe propagation Let $\mathbf{p}^{(0)} = \mathbf{z}$. DIAL's main hyperparameter is K, the number of layers from which diagonal estimates are formed. For $k = 1, \ldots, K-1$, DIAL propagates the probe features by

$$\mathbf{p}^{(k)} = \mathcal{T}_{\mathbf{S}}^{(k)}(\mathbf{p}^{(k-1)}),$$

where $\mathcal{T}_{\mathbf{S}}^{(k)}$ is a local propagation operator built from the GSO $\mathbf{S}$. For example, choosing $\mathcal{T}_{\mathbf{S}}^{(k)}(\mathbf{p}) = \mathbf{S}\mathbf{p}$ gives $\mathbf{p}^{(k)} = \mathbf{S}^k \mathbf{z}$. More generally, the stack of probe-propagation layers may implement powers of the GSO or polynomial filters,

such as those arising from the linear propagation part of Equation (1). Thus each propagated probe state can be written as $\mathbf{p}^{(k)} = M_k\mathbf{z}$ for some graph-derived operator M_k.

Diagonal readout For each state $k = 0, \ldots, K-1$, DIAL forms a local correlation with the original probe,

$$\mathbf{d}^{(k)} = \mathbf{z} \odot \mathbf{p}^{(k)},$$

where $\odot$ denotes the elementwise product. For every k, Lemma 1 gives

$$\mathbb{E}\Big[\mathbf{z} \odot \mathbf{p}^{(k)}\Big] = \mathsf{diag}(M_k).$$

Thus the K DIAL readouts expose structural signals corresponding to the diagonals of $M_0, \ldots, M_{K-1}$. For instance, if $M_k = \mathbf{A}^k$, these signals give closed-walk counts up to length $K-1$.

Aggregation In practice, we reduce variance by using r independent probes $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(r)}$. Equivalently, stacking them as $\mathbf{Z} = [\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(r)}] \in \mathbb{R}^{n\times r}$ introduces a sample dimension, and the same probe propagation is applied column-wise to produce estimates $\mathbf{D}^{(k)} \in \mathbb{R}^{n\times r}$. The state-wise estimates are then aggregated across propagation depths and independent probes using a shared readout:

$$\mathbf{h}_i^{\mathrm{DIAL}} = \frac{1}{r}\sum_{s=1}^{r} \mathrm{MLP}\Big(\|_{k=0}^{K-1}\mathbf{D}_{i,s}^{(k)}\Big) \in \mathbb{R}^{d_{\mathrm{dial}}}.$$

Averaging over r probes reduces the estimator variance at $1/r$ rate, while the distribution of the probes makes the construction permutation equivariant in expectation.

The resulting DIAL layer can be stacked with a standard message-passing GNN by using $\mathbf{h}_i^{\mathrm{DIAL}}$ directly as a structural node representation or concatenating it with ordinary node features,

$$\mathbf{H}_i^{(0)} = \mathbf{x}_i \,\|\, \mathbf{h}_i^{\mathrm{DIAL}}.$$

5 Structural Embeddings via Diagonal Estimation

Many classical node-level structural features, such as closed-walk counts and centrality measures, can be expressed as node-wise statistics derived from linear operators on the graph. In this section, we first show that DIAL gives nodes access to diagonal entries of graph filters. This can be used to recover several classical node quantities relevant for structural embeddings, including closed-walk counts, subgraph centrality, heat kernel signatures and graph angles. We also extend the diagonal estimation technique to off-diagonals for neighboring nodes, which is useful for counting simple cycles. Proofs for this section are in Appendix B.

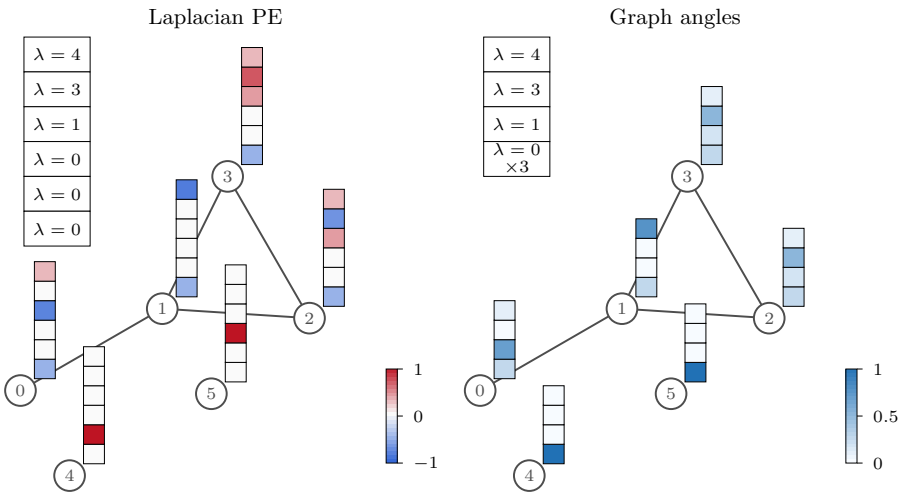

Fig. 2. Graph-angle structural encodings compared to Laplacian positional encodings. **Left**: Laplacian PE represents node i by $\Phi_{i,:}$, the row of eigenvector coordinates (in practice only a subset). This captures relative position in the graph: nodes 0 and 1 are close to each other so their encodings are similar, whereas nodes 4 and 5 are in different components so their encodings can be very different. **Right**: graph angles represent node i by $((P_{\mu_j})_{ii})_{j=1}^{q}$, where $\mu_1, \ldots, \mu_q$ are the distinct (Laplacian) eigenvalues. Graph angles are an example of a structural encoding: unlike eigenvectors, graph angles assign the same representation to automorphic nodes (such as 4 and 5). They are also invariant to basis and sign choices.

5.1 Diagonals of Graph Filters

Recall that a polynomial filter of degree $K-1$ can be expressed as

$$p(S) = \sum_{k=0}^{K-1} a_k S^k$$

where S is a graph shift operator. With diagonalization, this can also be written as $p(S) = \Phi p(\Lambda)\Phi^T$, where $p(\Lambda) = \mathsf{diag}(p(\lambda_1), \ldots, p(\lambda_n))$. In particular, spectral filters act independently on each eigenspace of S.

The linear propagation part of Equation (1) allows such filters to be applied in message passing effectively. However, many relevant structural properties, such as cycle counts, depend on *self-interactions*. Intuitively, this is encoded in the diagonal entries of $p(S)$. The diagonal is not accessible via standard message passing, which aggregates information row-wise and cannot isolate self-interactions. The following theorem shows that diagonal entries can be accessed in expectation using DIAL:

Theorem 1. *DIAL can compute node-wise statistics of the form*

$$v \mapsto (p(S))_{vv}$$

in expectation, where p is a polynomial filter of degree at most $K-1$. Equivalently, DIAL can compute expressions of the form

$$v \mapsto \sum_{i=1}^{n} g(\lambda_i)(\boldsymbol{\phi}_i(v))^2$$

where λ_i and $\boldsymbol{\phi}_i$ are the eigenvalues and eigenvectors of the GSO, and g is any polynomial function of degree at most $K-1$. Moreover, multiple polynomial filters can be expressed in parallel by increasing the (output) feature dimension.

5.2 Structural Quantities

The diagonal-filter view gives a compact way to recover several standard structural embeddings. We first focus on the quantities that can be expressed as polynomials of the GSO.

Example: graph angles. Figure 2 illustrates the distinction between positional and structural spectral encodings. Graph angles, defined in the preliminaries, are a canonical example of diagonal spectral features. They describe how strongly a node participates in an eigenspace through $\angle_\mu(v) = (P_\mu)_{vv}$. This differs from eigenvector positional encodings, which assign nodes coordinates in selected eigenvectors. While eigenvector coordinates are useful positional features, they depend on sign choices and, for repeated eigenvalues, on an arbitrary basis for the eigenspace. Graph angles instead summarize the whole eigenspace through the projector P_μ, making them intrinsic structural features.

Theorem 2. *DIAL can compute the following node-level structural quantities in expectation:*

- ***Closed-walk counts.*** *For every $1 \le k \le K-1$,*

  $$v \mapsto (A^k)_{vv},$$

  gives the number of closed walks of length k starting and ending at v.
- ***Graph angles.*** *For a Laplacian eigenspace E_μ with projector P_μ,*

  $$v \mapsto \angle_\mu(v) = (P_\mu)_{vv},$$

  can be computed, provided the eigenspace projector P_μ is represented by a polynomial in L of degree at most $K-1$. In particular, this holds if L has at most K distinct eigenvalues.

Spectral quantities defined by exponential graph filters can be approximated with polynomials of the GSO:

Theorem 3. *DIAL can approximate the following node-level structural quantities in expectation:*

- ***Subgraph centrality.*** *For* $SC(v) = (e^A)_{vv}$*, if* A *is symmetric and* $\|A\|_2 \leq \alpha$*, the degree-*$(K-1)$ *Taylor approximation yields an estimate* $\widehat{SC}_K(v)$ *satisfying*

$$|\mathbb{E}[\widehat{SC}_K(v)] - SC(v)| \leq \frac{e^\alpha \alpha^K}{K!}.$$

- ***Heat kernel signature.*** *For* $\mathrm{HKS}_t(v) = (e^{-tL})_{vv}$*, if* L *is symmetric and* $\|L\|_2 \leq \beta$*, the degree-*$(K-1)$ *Taylor approximation yields an estimate* $\widehat{\mathrm{HKS}}_{t,K}(v)$ *satisfying*

$$|\mathbb{E}[\widehat{\mathrm{HKS}}_{t,K}(v)] - \mathrm{HKS}_t(v)| \leq \frac{(t\beta)^K}{K!}.$$

5.3 Neighbor Off-Diagonal Readout

Some structural quantities require off-diagonal entries of graph-derived operators. For example, the number of common neighbors of adjacent nodes u and v is $(A^2)_{uv}$, which is useful for link prediction, and also appears in formulas for counting simple cycles. The same random probing idea can estimate such entries: nodes first share their original probe values with their neighbors, and after ℓ layers of message propagation, each endpoint correlates the last layer state with the original probe values of their neighbors.

Theorem 4. *For each node* v*, DIAL can compute the multiset of off-diagonal elements of* $p(S)$ *over neighbors* $w \in N(v)$*:*

$$\{\{(p(S))_{vw} : w \in N(v)\}\}$$

in expectation, where p *is a polynomial filter of degree at most* $K-1$*.*

The proof is in Appendix B.

Combined with closed-walk counts from Theorem 2, the off-diagonal estimation from Theorem 4 can be used to express counts of simple cycles:

Theorem 5. *A DIAL-augmented message-passing model can compute the number of simple cycles of length 3,4 and 5 at each node.*

The proof of Theorem 5 is given in Section C.

6 Experiments

We present initial empirical results in three settings: molecular property prediction on ZINC, synthetic node-level cycle counting, and directed structural pattern detection on synthetic multigraphs. These experiments test the usefulness of DIAL as a lightweight structural encoding, and its ability to recover explicitly structural information in controlled settings. Our comparisons include standard message-passing GNNs, hand-crafted cycle features, spectral positional encodings, PEARL and other random-probe baselines, as well as subgraph and higher-order architectures for cycle counting. Random Feature Propagation [10] is a closely related baseline, but we omit it in the experiments because no public implementation was available. Experimental details can be found in Appendix E.

Table 1. PE-focused comparison of test MAE for logP prediction on ZINC. Lower is better. The #PEs column reports the number of positional encodings used by PE-based baselines; Full denotes the full spectral basis, and N/A denotes methods without a fixed PE count.

Method	**#PEs**	**Test MAE**
GCN [16]	N/A	0.469 ± 0.002
GIN [28]	N/A	0.209 ± 0.018
GSN with cycles [3]	10	0.115 ± 0.012
GIN + rand id [28]	1	0.279 ± 0.023
SignNet-8S [17]	8	0.1034 ± 0.0056
SignNet [17]	Full	0.0853 ± 0.0026
BasisNet-8S [17]	8	0.1554 ± 0.0048
BasisNet [17]	Full	0.1555 ± 0.0124
SPE-8S [11]	8	0.0736 ± 0.0007
SPE [11]	Full	0.0693 ± 0.0040
R-PEARL [15]	N/A	0.0699 ± 0.002
B-PEARL [15]	N/A	0.0644 ± 0.001
DIAL ($K = 1$)	N/A	0.0703 ± 0.0012
DIAL ($K = 4$)	N/A	0.0658 ± 0.0020
DIAL ($K = 8$)	N/A	0.0733 ± 0.0026
DIAL ($K = 12$)	N/A	0.0752 ± 0.0074

6.1 Results

Table 1 reports a PE-focused comparison on ZINC, with logP prediction evaluated by test MAE. The non-DIAL entries follow the baseline set reported by [15]. Hence, the table should be read as a comparison to positional-encoding methods rather than as a comprehensive ZINC leaderboard. DIAL performs comparably to strong spectral and random-probe positional encoding methods.

Table 2 compares DIAL on node-level cycle counting. Methods designed specifically to expose cycle information, such as subgraph and higher-order architectures, achieve the lowest errors, but require substantially higher space or time complexity. Among methods with comparable random-probe complexity, DIAL improves over R-PEARL across all cycle lengths while retaining the same asymptotic cost.

We further evaluate DIAL on a directed synthetic pattern-detection task of Egressy et al. [9]. See Table 3 for the results. The task is node-level multi-label classification: nodes are labeled by whether they participate in directed cycles of length 2–6, scatter-gather motifs (S-G), or bicliques (B-C), following Altman et al. [2].

References

1. Abboud, R., Ceylan, I.I., Grohe, M., Lukasiewicz, T.: The surprising power of graph neural networks with random node initialization. arXiv preprint arXiv:2010.01179

Table 2. Normalized MAE results of counting cycles at node level on a synthetic dataset. Complexities ignore fixed message-passing depth and hidden width. Here N is the number of nodes, $\bar{d}$ is the average degree, s is the maximum extracted subgraph size, and r is the number of random probes. For random-probe methods, space is reported for parallel probe processing.

Method	Complexity		Synthetic (norm. MAE)			
	Space	Time	3-Cycle	4-Cycle	5-Cycle	6-Cycle
Base GNN	$O(N)$	$O(N\bar{d})$	0.3515	0.2742	0.2088	0.1555
ID-GNN [29]	$O(Ns)$	$O(Ns\bar{d})$	0.0006	0.0022	0.0490	0.0495
NGNN [30]	$O(Ns)$	$O(Ns\bar{d})$	0.0003	0.0013	0.0402	0.0439
GNN-AK+ [31]	$O(Ns)$	$O(Ns\bar{d})$	0.0004	0.0041	0.0133	0.0238
PPGN [18]	$O(N^2)$	$O(N^3)$	0.0003	0.0009	0.0036	0.0071
I^2-GNN [12]	$O(Ns\bar{d})$	$O(Ns\bar{d}^2)$	0.0003	0.0016	0.0028	0.0082
R-PEARL [15]	$O(rN)$	$O(rN\bar{d})$	0.0118	0.0360	0.0561	0.0622
DIAL (Ours)	$O(rN)$	$O(rN\bar{d})$	0.0041	0.0185	0.0271	0.0348

Table 3. Pattern detection on synthetic directed graphs. We report minority-class F1 scores in percent for node-level binary labels; higher is better. C2–C6 denote directed cycles of length 2–6, S-G denotes scatter-gather motifs, and B-C denotes bicliques. Both methods use $K = 4$ and six learned probe-aggregation layers.

Method	**C2**	**C3**	**C4**	**C5**	**C6**	**S-G**	**B-C**	**Macro F1**
R-PEARL [15]	15.48 ± 7.59	41.21 ± 5.42	66.05 ± 0.52	55.13 ± 0.80	54.74 ± 0.60	70.54 ± 1.09	62.56 ± 2.49	52.24
DIAL (Ours)	91.07 ± 3.33	82.44 ± 4.48	79.72 ± 1.50	72.10 ± 2.21	74.83 ± 2.95	78.68 ± 2.79	66.07 ± 1.48	77.84

(2020)
2. Altman, E., Blanuša, J., Von Niederhäusern, L., Egressy, B., Anghel, A., Atasu, K.: Realistic synthetic financial transactions for anti-money laundering models. Advances in Neural Information Processing Systems **36**, 29851–29874 (2023)
3. Bouritsas, G., Frasca, F., Zafeiriou, S., Bronstein, M.M.: Improving graph neural network expressivity via subgraph isomorphism counting. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(1), 657–668 (2022)
4. Bruna, J., Zaremba, W., Szlam, A., LeCun, Y.: Spectral networks and locally connected networks on graphs. arXiv preprint arXiv:1312.6203 (2013)
5. Chen, Z., Chen, L., Villar, S., Bruna, J.: Can graph neural networks count substructures? Advances in neural information processing systems **33**, 10383–10395 (2020)
6. Chung, F.R.: Spectral graph theory, vol. 92. American Mathematical Soc. (1997)
7. Dasoulas, G., Santos, L.D., Scaman, K., Virmaux, A.: Coloring graph neural networks for node disambiguation. arXiv preprint arXiv:1912.06058 (2019)
8. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. Advances in neural information processing systems **29** (2016)
9. Egressy, B., Von Niederhäusern, L., Blanuša, J., Altman, E., Wattenhofer, R., Atasu, K.: Provably powerful graph neural networks for directed multigraphs. In:

Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 11838–11846 (2024)
10. Eliasof, M., Frasca, F., Bevilacqua, B., Treister, E., Chechik, G., Maron, H.: Graph positional encoding via random feature propagation. In: International conference on machine learning. pp. 9202–9223. PMLR (2023)
11. Huang, Y., Lu, W., Robinson, J., Yang, Y., Zhang, M., Jegelka, S., Li, P.: On the stability of expressive positional encodings for graphs. In: International Conference on Learning Representations. vol. 2024, pp. 39745–39774 (2024)
12. Huang, Y., Peng, X., Ma, J., Zhang, M.: Boosting the cycle counting power of graph neural networks with I^2-GNNs. arXiv preprint arXiv:2210.13978 (2022)
13. Hutchinson, M.: A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. Communications in Statistics - Simulation and Computation **19**(2), 433–450 (1990). https://doi.org/10.1080/03610919008812866, https://doi.org/10.1080/03610919008812866
14. Kanatsoulis, C., Ribeiro, A.: Counting graph substructures with graph neural networks. In: The twelfth international conference on learning representations (2024)
15. Kanatsoulis, C.I., Choi, E., Jegelka, S., Leskovec, J., Ribeiro, A.: Learning efficient positional encodings with graph neural networks. arXiv preprint arXiv:2502.01122 (2025)
16. Kipf, T.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
17. Lim, D., Robinson, J., Zhao, L., Smidt, T., Sra, S., Maron, H., Jegelka, S.: Sign and basis invariant networks for spectral graph representation learning. arXiv preprint arXiv:2202.13013 (2022)
18. Maron, H., Ben-Hamu, H., Serviansky, H., Lipman, Y.: Provably powerful graph networks. Advances in neural information processing systems **32** (2019)
19. Maron, H., Ben-Hamu, H., Shamir, N., Lipman, Y.: Invariant and equivariant graph networks. arXiv preprint arXiv:1812.09902 (2018)
20. Maron, H., Fetaya, E., Segol, N., Lipman, Y.: On the universality of invariant networks. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 4363–4371. PMLR (09–15 Jun 2019), https://proceedings.mlr.press/v97/maron19a.html
21. Morris, C., Ritzert, M., Fey, M., Hamilton, W.L., Lenssen, J.E., Rattan, G., Grohe, M.: Weisfeiler and leman go neural: Higher-order graph neural networks. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 4602–4609 (2019)
22. Papp, P.A., Martinkus, K., Faber, L., Wattenhofer, R.: Dropgnn: Random dropouts increase the expressiveness of graph neural networks. Advances in Neural Information Processing Systems **34**, 21997–22009 (2021)
23. Papp, P.A., Wattenhofer, R.: A theoretical comparison of graph neural network extensions. In: International Conference on Machine Learning. pp. 17323–17345. PMLR (2022)
24. Puny, O., Ben-Hamu, H., Lipman, Y.: Global attention improves graph networks generalization (2020), https://arxiv.org/abs/2006.07846
25. Shuman, D.I., Narang, S.K., Frossard, P., Ortega, A., Vandergheynst, P.: 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)

26. Vignac, C., Loukas, A., Frossard, P.: Building powerful and equivariant graph neural networks with structural message-passing. Advances in neural information processing systems **33**, 14143–14155 (2020)
27. Wollschläger, T., Kemper, N., Hetzel, L., Sommer, J., Günnemann, S.: Expressivity and generalization: Fragment-biases for molecular gnns. arXiv preprint arXiv:2406.08210 (2024)
28. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? arXiv preprint arXiv:1810.00826 (2018)
29. You, J., Gomes-Selman, J.M., Ying, R., Leskovec, J.: Identity-aware graph neural networks. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35, pp. 10737–10745 (2021)
30. Zhang, M., Li, P.: Nested graph neural networks. Advances in Neural Information Processing Systems **34**, 15734–15747 (2021)
31. Zhao, L., Jin, W., Akoglu, L., Shah, N.: From stars to subgraphs: Uplifting any gnn with local structure awareness. arXiv preprint arXiv:2110.03753 (2021)

A Hutchinson's Diagonal Estimator

Lemma 1. *Let $M \in \mathbb{R}^{n\times n}$. Let $z \in \mathbb{R}^n$ be a random vector with i.i.d. Rademacher entries (i.e., each entry is independently $+1$ or -1 with probability $1/2$). Then*

$$\mathbb{E}[z \odot (Mz)] = \mathsf{diag}(M)$$

where $\odot$ is the elementwise product. The variance of the ith entry is $\sum_{j\neq i} M_{ij}^2$.

Proof. Note that $\mathbb{E}[z_i] = 0$, $\mathbb{E}[z_i^2] = 1$, and $\mathbb{E}[z_i z_j] = \mathbb{E}[z_i]\mathbb{E}[z_j] = 0$ for $i \neq j$. Let $d_i = z_i(Mz)_i$. Then $\mathbb{E}[d_i] = \mathbb{E}\big[\sum_j M_{ij} z_i z_j\big] = M_{ii}$ where the cross terms vanish because $\mathbb{E}[z_i z_j] = 0$ for $i \neq j$.

The variance is computed from $\mathsf{Var}(d_i) = \mathbb{E}[d_i^2] - (\mathbb{E}[d_i])^2$. Here,

$$d_i^2 = (\sum_j M_{ij} z_i z_j)(\sum_j M_{ij} z_i z_j) = \sum_{j,k} M_{ij} M_{ik} z_i^2 z_j z_k = \sum_{j,k} M_{ij} M_{ik} z_j z_k$$

because $z_i^2 = 1$. Taking the expectation, $\mathbb{E}[d_i^2] = \mathbb{E}\Big[\sum_{j,k} M_{ij} M_{ik} z_j z_k\Big] = \sum_j M_{ij}^2$, since $\mathbb{E}[z_j z_k] = 0$ for $j \neq k$. Thus, $\mathsf{Var}(d_i) = \mathbb{E}[d_i^2] - (\mathbb{E}[d_i])^2 = \sum_j M_{ij}^2 - M_{ii}^2$ □

B Proofs for Section 5

Theorem 1. *DIAL can compute node-wise statistics of the form*

$$v \mapsto (p(S))_{vv}$$

in expectation, where p is a polynomial filter of degree at most $K-1$. Equivalently, DIAL can compute expressions of the form

$$v \mapsto \sum_{i=1}^{n} g(\lambda_i)(\boldsymbol{\phi}_i(v))^2$$

where λ_i and $\boldsymbol{\phi}_i$ are the eigenvalues and eigenvectors of the GSO, and g is any polynomial function of degree at most $K-1$. Moreover, multiple polynomial filters can be expressed in parallel by increasing the (output) feature dimension.

Proof. Let $p(S) = \sum_{k=0}^{K-1} a_k S^k$ be the target polynomial filter. Since the linear propagation part of Equation (1) represents polynomial graph filters of degree $K-1$, the DIAL probe-propagation weights can learn the coefficients of p. Thus, for a Rademacher probe $\mathbf{z}$, the learned probe propagation can produce

$$\mathbf{p} = p(S)\mathbf{z}.$$

The DIAL readout then forms the node-wise correlation

$$\mathbf{z} \odot \mathbf{p} = \mathbf{z} \odot p(S)\mathbf{z}.$$

Applying Lemma 1 with $M = p(S)$ gives

$$\mathbb{E}[\mathbf{z} \odot p(S)\mathbf{z}] = \mathsf{diag}(p(S)).$$

Hence node v obtains $(p(S))_{vv}$ in expectation.

For the equivalent spectral expression, write $S = \Phi\Lambda\Phi^T$. Then $p(S) = \Phi p(\Lambda)\Phi^T$, and therefore

$$(p(S))_{vv} = \sum_{i=1}^{n} p(\lambda_i)(\boldsymbol{\phi}_i(v))^2.$$

Since p is an arbitrary polynomial of degree at most $K-1$, this gives the stated expression for any polynomial response g of that degree. Multiple polynomial filters can be handled in parallel by using separate output channels for $p_1, \ldots, p_d$ and applying the same argument channel-wise. □

Theorem 4. *For each node v, DIAL can compute the multiset of off-diagonal elements of $p(S)$ over neighbors $w \in N(v)$:*

$$\{\{(p(S))_{vw} : w \in N(v)\}\}$$

in expectation, where p is a polynomial filter of degree at most $K-1$.

Proof. Let $M = p(S)$. Since the linear propagation part of Equation (1) represents polynomial graph filters of degree $K-1$, the DIAL probe-propagation weights can learn the coefficients of p. Thus, after the corresponding probe-propagation layer, the probe feature is

$$\mathbf{p}^{(\ell)} = M\mathbf{z}.$$

For the read-out, each node first sends its original probe value to its neighbors (this is effectively done in the first layer). Thus node v receives the unordered

multiset $\{\{z_w : w \in N(v)\}\}$. Correlating each received probe with the propagated probe value $p_v^{(\ell)}$ gives the unordered multiset of estimates

$$R_v^{(\ell)} = \{\{z_w p_v^{(\ell)} : w \in N(v)\}\}.$$

For the analysis, label one incoming message by its sender w and write $r_{vw}^{(\ell)} = z_w p_v^{(\ell)}$. Since $p_v^{(\ell)} = \sum_j M_{vj} z_j$, we have

$$\mathbb{E}[r_{vw}^{(\ell)}] = \mathbb{E}\left[\sum_j M_{vj} z_w z_j\right] = \sum_j M_{vj}\mathbb{E}[z_w z_j] = M_{vw},$$

because $\mathbb{E}[z_w z_j] = 0$ for $j \neq w$ and $\mathbb{E}[z_w^2] = 1$. Therefore $R_v^{(\ell)}$ estimates the multiset $\{\{M_{vw} : w \in N(v)\}\}$ in expectation. Substituting $M = p(S)$ gives the claim. □

B.1 Structural Quantities

The next proofs use a Lagrange polynomial, which is a construction for polynomials of minimal degree interpolating a given set of points:

Lemma 3 (Lagrange interpolation polynomials). *Given a set of $k+1$ points $\{(x_1, y_1), \ldots, (x_{k+1}, y_{k+1})\}$, where $x_i \neq x_j$, there is a polynomial $L(x)$ of degree $\leq k$ such that $L(x_i) = y_i$ for all $1 \leq i \leq k+1$.*

Proof. For each i, construct a polynomial $\ell_i(x)$ such that $\ell_i(x_i) = 1$ and $\ell_i(x_j) = 0$ for all $j \neq i$. Such a polynomial is given by

$$\ell_i(x) = \prod_{j \neq i : 1 \leq j \leq k+1} \frac{x - x_j}{x_i - x_j}$$

since all factors are equal to one for $x = x_i$, and for $x = x_j, j \neq i$, $(x - x_j)/(x_i - x_j) = 0$. Note that the degree of $\ell_i(x)$ is k. The polynomial $L(x)$ is given by $\sum_{i=1}^{k+1} y_i \ell_i(x)$. □

Theorem 2. *DIAL can compute the following node-level structural quantities in expectation:*

- ***Closed-walk counts.*** *For every $1 \leq k \leq K-1$,*

  $$v \mapsto (A^k)_{vv},$$

  gives the number of closed walks of length k starting and ending at v.
- ***Graph angles.*** *For a Laplacian eigenspace E_μ with projector P_μ,*

  $$v \mapsto \angle_\mu(v) = (P_\mu)_{vv},$$

  can be computed, provided the eigenspace projector P_μ is represented by a polynomial in L of degree at most $K-1$. In particular, this holds if L has at most K distinct eigenvalues.

Proof. **Closed walks:** With the adjacency matrix A as the GSO, $(A^k)_{vv}$ gives the number of closed walks of length k starting and ending at v (Lemma 6). This is the diagonal of the polynomial filter $p(A) = A^k$, which has degree k. Theorem 1 gives access to $(A^k)_{vv}$ in expectation.

Graph angles: Use the Laplacian L as the GSO, and let $\eta_1, \ldots, \eta_m$ be the distinct eigenvalues of L. Fix an eigenvalue $\mu \in \{\eta_1, \ldots, \eta_m\}$. Let $P_\mu = \Phi_\mu \Phi_\mu^T$ be the projector onto the eigenspace E_μ, so that the graph angle is $\angle_\mu(v) = (P_\mu)_{vv}$. By Lemma 3, there is a polynomial p_μ of degree at most $m-1$ such that

$$p_\mu(\eta) = \begin{cases} 1 & \eta = \mu \\ 0 & \eta \in \{\eta_1, \ldots, \eta_m\} \setminus \{\mu\}. \end{cases}$$

With this choice of polynomial,

$$p_\mu(L) = \Phi p_\mu(\Lambda) \Phi^T = \sum_{i \in I_\mu} \boldsymbol{\phi}_i \boldsymbol{\phi}_i^T = P_\mu.$$

If the Laplacian has at most K distinct eigenvalues, then $\deg(p_\mu) \leq m-1 \leq K-1$. Theorem 1 then gives access to $\mathsf{diag}(P_\mu)$ in expectation, which is exactly the vector of graph angles for E_μ. □

Theorem 3. *DIAL can approximate the following node-level structural quantities in expectation:*

- ***Subgraph centrality.*** *For $SC(v) = (e^A)_{vv}$, if A is symmetric and $\|A\|_2 \leq \alpha$, the degree-$(K-1)$ Taylor approximation yields an estimate $\widehat{SC}_K(v)$ satisfying*
$$|\mathbb{E}[\widehat{SC}_K(v)] - SC(v)| \leq \frac{e^\alpha \alpha^K}{K!}.$$
- ***Heat kernel signature.*** *For $\mathrm{HKS}_t(v) = (e^{-tL})_{vv}$, if L is symmetric and $\|L\|_2 \leq \beta$, the degree-$(K-1)$ Taylor approximation yields an estimate $\widehat{\mathrm{HKS}}_{t,K}(v)$ satisfying*
$$|\mathbb{E}[\widehat{\mathrm{HKS}}_{t,K}(v)] - \mathrm{HKS}_t(v)| \leq \frac{(t\beta)^K}{K!}.$$

Proof. For subgraph centrality, use the degree-$(K-1)$ Taylor polynomial

$$p_{K-1}(x) = \sum_{r=0}^{K-1} \frac{x^r}{r!}.$$

By Theorem 1, DIAL can compute an estimate $\widehat{\mathrm{SC}}_K(v)$ such that

$$\mathbb{E}[\widehat{\mathrm{SC}}_K(v)] = (p_{K-1}(A))_{vv}.$$

Since A is symmetric, write $A = \Phi \Lambda \Phi^T$. Then

$$\|e^A - p_{K-1}(A)\|_2 = \max_i |e^{\lambda_i} - p_{K-1}(\lambda_i)|.$$

Taylor's theorem gives

$$|e^{\lambda} - p_{K-1}(\lambda)| \leq \frac{e^{\alpha}\alpha^{K}}{K!} \qquad \text{for all } |\lambda| \leq \alpha.$$

Therefore

$$|\mathbb{E}[\widehat{\mathrm{SC}}_K(v)] - \mathrm{SC}(v)| = |(p_{K-1}(A) - e^{A})_{vv}| \leq \|p_{K-1}(A) - e^{A}\|_2 \leq \frac{e^{\alpha}\alpha^{K}}{K!}.$$

For the heat kernel signature at time t, use

$$h_{t,K-1}(x) = \sum_{r=0}^{K-1} \frac{(-tx)^r}{r!}.$$

Theorem 1 gives an estimate $\widehat{\mathrm{HKS}}_{t,K}(v)$ satisfying

$$\mathbb{E}[\widehat{\mathrm{HKS}}_{t,K}(v)] = (h_{t,K-1}(L))_{vv}.$$

Since L is symmetric positive semidefinite, its eigenvalues satisfy $0 \leq \lambda_i \leq \beta$. Taylor's theorem applied to e^x on $x \in [-t\beta, 0]$ gives

$$|e^{-t\lambda} - h_{t,K-1}(\lambda)| \leq \frac{(t\beta)^K}{K!} \qquad \text{for all } 0 \leq \lambda \leq \beta.$$

As above,

$$|\mathbb{E}[\widehat{\mathrm{HKS}}_{t,K}(v)] - \mathrm{HKS}_t(v)| \leq \|h_{t,K-1}(L) - e^{-tL}\|_2 \leq \frac{(t\beta)^K}{K!}.$$

□

B.2 Simulating BasisNet

BasisNet [17] computes functions of the projection matrices $P_{\mu_i} = \Phi_{\mu_i}\Phi_{\mu_i}^T$. Permutation equivariance requires that any such function φ satisfy

$$\varphi(\Pi\Phi\Phi^T\Pi^T) = \Pi\varphi(\Phi\Phi^T)$$

for any permutation matrix Π, since a permutation of the nodes permutes both the rows and columns of the projection matrix. The class of functions with this property is characterized in [20], and BasisNet instantiates φ using the Invariant Graph Network (IGN) [19].

IGNs are a family of GNNs that operate on tensors of order k representing k-tuples of nodes. [19] characterize all linear equivariant layers between such tensors. In the setting of BasisNet, only a *mixed-order equivariant layer* is needed, mapping a matrix $X \in \mathbb{R}^{n\times n}$ to node features $Y \in \mathbb{R}^{n\times d}$. The following operations can be shown to span the space of all linear layers of this type.

- The diagonal operation: $Y_i = X_{ii}$
- Row-wise sum: $Y_i = \sum_{j=1}^{n} X_{ij}$
- Column-wise sum: $Y_i = \sum_{j=1}^{n} X_{ji}$
- Sum of all entries: $Y_i = \sum_{j=1}^{n} \sum_{k=1}^{n} X_{jk}$
- Sum of diagonal entries: $Y_i = \sum_{j=1}^{n} X_{jj}$

Lemma 4. *DIAL with a virtual node can simulate BasisNet.*

Proof. From Theorem 1, DIAL can compute $\mathsf{diag}(P_{\mu_i})$ for each eigenspace. This gives the diagonal operation. For a symmetric graph shift operator, the row-wise and column-wise sums can be computed with $z = \mathbf{1}$ as the initial features. The sum of all entries and sum of diagonal entries can be computed with the help of a virtual node, by summing over the node features obtained from the previous operations. □

The diagonal estimate is key to obtaining the diagonal operation, which is not possible with standard message passing GNNs. The diagonal operation is the most important one, as it is the only one not constant across nodes:

Lemma 5. *Let L be the Laplacian matrix and $P_\lambda = \Phi_\lambda \Phi_\lambda^T$ be the projection matrix to the eigenspace of eigenvalue λ. Then, the diagonal operation is the only non-constant operation among the five operations above when applied to P_λ.*

Proof. The vector $\mathbf{1}$ is an eigenvector with eigenvalue 0, since $L\mathbf{1} = 0$. The row-wise sums are zero for $\lambda \neq 0$: $P_\lambda \mathbf{1} = 0$ since $\mathbf{1}$ is orthogonal to these eigenspaces for any $\lambda \neq 0$. The same holds for the column sums by symmetry. For the case $\lambda = 0$, $P_0 \mathbf{1} = \mathbf{1}$ since $\mathbf{1}$ is an eigenvector. Thus, the row-wise and column-wise sums are constant across nodes in all cases.

The diagonal sum and total sum are constant across nodes by definition. Note that the diagonal sum $\sum_i (P_\lambda)_{ii}$ is equal to the multiplicity of λ, since $\mathsf{trace}(P_\lambda) = \mathsf{trace}(P_\lambda^T) = \mathsf{trace}(\Phi_\lambda^T \Phi_\lambda) = \mathrm{rank}(\Phi_\lambda)$. □

C Cycle Counting Proofs

The following lemma is a basic result in graph theory, which we include for completeness.

Lemma 6. *Let A be the adjacency matrix. $(A^k)_{vw}$ gives the number of walks of length exactly k from v to w.*

Proof. By induction on k. The base case $k = 1$ is true by definition of the adjacency matrix. Assume the claim holds for k. The number of walks of length $k+1$ from v to w can be partitioned according to the node u visited at step k, where u must be a neighbor of w. This is given by $A^k A$:

$$(A^{k+1})_{vw} = \sum_u (A^k)_{vu} A_{uw}$$

□

Let $C_k(v)$ denote the number of distinct simple undirected k-cycles containing vertex v. $C_k(v) = 0$ for $k \leq 2$. Let $C_k^{\rightarrow}(v)$ denote the number of distinct simple directed k-cycles containing vertex v. Note that $C_k^{\rightarrow}(v) = 2C_k(v)$ since each undirected cycle can be traversed in two directions.

Lemma 7 (Triangles). *For any vertex v, $C_3^{\rightarrow}(v) = (A^3)_{vv}$*

Proof. Since there are no self-loops, any closed walk of length 3 is a simple directed 3-cycle. □

Lemma 8 (4-cycles). $C_4^{\rightarrow}(v) = (A^4)_{vv} - d(v)^2 + d(v) - \sum_{w \in N(v)} d(w)$

Proof. We count the number of closed walks of length 4 that are not simple cycles, starting from v. Closed walks of length 4 which are not simple cycles must revisit some node. This node can be v, meaning the walk must be of the form $v - x - v - y - v$, which can be formed in $d(v)^2 = ((A^2)_{vv})^2$ ways. If the revisited node is not v, the walk must be of the form $v - x - y - x - v$, which can be formed in $\sum_{w \in N(v)} d(w)$ ways. These cases overlap for paths of the form $v - x - v - x - v$, of which there are $d(v) = (A^2)_{vv}$ in total. □

Lemma 9 (5-cycles).

$$C_5^{\rightarrow}(v) = (A^5)_{vv} - \Big(2d(v)(A^3)_{vv} + 2\sum_{u \in N(v)} d(u)(A^2)_{uv} + \sum_{u \in N(v)} (A^3)_{uu} - 5(A^3)_{vv}\Big).$$

Proof. The entry $(A^5)_{vv}$ counts all closed walks of length 5 starting and ending at v. We subtract those that are not simple 5-cycles.

Every non-simple closed walk of length 5 in a simple graph consists of a directed triangle together with a backtracking walk of the form $x - y - x$. We count such 5-walks in two cases.

First suppose that the triangle contains v. Let T_v be the set of undirected triangles containing v,

$$T_v = \{\{v, a, b\} : \{v, a\}, \{a, b\}, \{b, v\} \in E\}$$

Fix an undirected triangle $\{v, a, b\}$. There are two orientations. For one orientation, say

$$v \to a \to b \to v,$$

the backtracking walk can be inserted at one of the four positions v, a, b, v. Here, the two occurrences of v are distinct, since the 5-walk is rooted at v. Thus, before correcting overcounting, this gives

$$2\big(2d(v) + d(a) + d(b)\big)$$

walks for the triangle $\{v, a, b\}$.

However, if the edge on the backtracking walk is one of the triangle edges, the same walk is counted twice: once as a backtrack inserted at one endpoint of the edge and once as a backtrack inserted at the other endpoint. For each

orientation there are three such choices. Therefore the number of non-simple closed 5-walks whose underlying triangle contains v is

$$\sum_{\{v,a,b\}\in T_v} (4d(v) + 2d(a) + 2d(b) - 6)\,,$$

We now rewrite this expression in matrix form. First, since $d(v)$ is independent of the triangle,

$$\sum_{\{v,a,b\}\in T_v} 4d(v) = 4d(v)|T_v| = 2d(v)(A^3)_{vv}$$

Second, consider the term involving the degrees of the two non-v vertices:

$$\sum_{\{v,a,b\}\in T_v} \big(2d(a) + 2d(b)\big).$$

We rewrite this by summing over neighbors $u \in N(v)$. A neighbor u appears as one of the two non-v vertices in exactly as many triangles as there are common neighbors of u and v. This number is $(A^2)_{uv}$. Thus

$$\sum_{\{v,a,b\}\in T_v} \big(2d(a) + 2d(b)\big) = 2\sum_{u\in N(v)} d(u)(A^2)_{uv}.$$

Finally,

$$\sum_{\{v,a,b\}\in T_v} 6 = 6|T_v| = 3(A^3)_{vv}.$$

Combining the three identities gives

$$\sum_{\{v,a,b\}\in T_v} (4d(v) + 2d(a) + 2d(b) - 6) = 2d(v)(A^3)_{vv} + 2\sum_{u\in N(v)} d(u)(A^2)_{uv} - 3(A^3)_{vv}.$$

It remains to count the non-simple closed 5-walks whose triangle does not contain v. Such a walk must have the form

$$v \to u \to (\text{a directed triangle walk based at } u) \to v$$

for some $u \in N(v)$. The number of directed triangles through u is $(A^3)_{uu}$. Among these, the directed triangles that also contain v are exactly those using the edge $\{u, v\}$, and there are $2(A^2)_{uv}$ of them. Therefore this case contributes

$$\sum_{u\in N(v)} \big((A^3)_{uu} - 2(A^2)_{uv}\big)\,.$$

The expression can be simplified using the fact that $\sum_{u\in N(v)}(A^2)_{uv} = (A^3)_{vv}$. Hence, the second case contributes

$$-2(A^3)_{vv} + \sum_{u\in N(v)} (A^3)_{uu}.$$

Adding the two cases, the number of non-simple closed 5-walks from v to itself is

$$2d(v)(A^3)_{vv} + 2\sum_{u \in N(v)} d(u)(A^2)_{uv} + \sum_{u \in N(v)} (A^3)_{uu} - 5(A^3)_{vv}.$$

Subtracting these walks from all closed 5-walks gives the claimed formula. □

D Cycle Counting in [14] and [15]

Kanatsoulis and Ribeiro [14] study the substructure-counting power of GNNs under random inputs. However, a key assumption in their analysis does not appear to be satisfiable, which undermines the validity of the resulting guarantees. We highlight this issue because the same assumption is directly reused in PEARL [15], which is closely related to our method.

In more detail, the analysis in [14] uses higher-order moments of random variables. A core assumption is access to random variables with all moments equal to 1:

> We assume that they $(x_1, \ldots, x_N)$ are identically distributed and satisfy $\mathbb{E}[x_i] = 0, \mathbb{E}[x_i^p] = 1$ for $i = 1, \ldots, N$ and $p \geq 2 \in \mathbb{Z}$

While this assumption can hold for even moments (e.g. by taking $\Pr(x_i = \pm 1) = 1/2$), it cannot hold simultaneously for odd moments:

Lemma 10. *There does not exist a random variable X such that*

$$\mathbb{E}[X] = 0, \mathbb{E}[X^2] = 1, \mathbb{E}[X^3] = 1, \mathbb{E}[X^4] = 1$$

hold simultaneously.

Proof. Apply the Cauchy-Schwarz inequality (Theorem 6) with $A = X$ and $B = X^2$:

$$|\mathbb{E}[X^3]|^2 \leq \mathbb{E}[X^2]\mathbb{E}[X^4]$$

Under the stated assumptions, both sides are 1, so equality holds. Equality in Cauchy-Schwarz implies that $X = cX^2$ almost surely, for some constant c. This implies that $X(X - c) = 0$ almost surely, so X is either 0 or c. This contradicts $\mathbb{E}[X] = 0$ and $\mathbb{E}[X^2] = 1$. □

The Cauchy-Schwarz inequality is included below for convenience:

Theorem 6 (Cauchy-Schwarz inequality). *For any two random variables A, B, we have*

$$|\mathbb{E}[AB]|^2 \leq \mathbb{E}[A^2]\mathbb{E}[B^2]$$

where equality holds iff $A = cB$, for some constant c.

E Experimental Details

We use the PEARL codebase [15] as the starting point, modifying the positional-encoding module for DIAL. In the DIAL variants reported here, Hutchinson correlations are applied at two levels inside the structural encoding module: first after explicit graph-shift propagation of the random probes, and then again after a number of learned GIN layers. The outputs of each layer are concatenated and projected to the final node encoding.

For ZINC, the structural encoder uses $r = 120$ Rademacher probes, a 37-dimensional output encoding, and an eight-layer GIN with hidden dimension 40. The resulting encoding is added to the initial node embeddings and passed to the task GNN: four GINE layers with hidden dimension 128, additive graph pooling, and a three-layer prediction MLP.

For the cycle counting task, we use the same DIAL structural encoder and no additional base GNN. DIAL uses $r = 120$ Rademacher probes, a 37-dimensional output encoding, and a six-layer GIN with hidden dimension 40. This encoding is added to 64-dimensional node embeddings, and node-level predictions are made directly with a two-layer MLP.

For directed pattern detection, we use synthetic directed multigraphs with separate train, validation, and test graphs, each with 8192 nodes. Both R-PEARL and DIAL are trained as node-level multi-label classifiers using $r = 120$ random probes, $K = 4$, a 37-dimensional structural encoding, six learned GIN layers with hidden dimension 40, and no additional downstream base GNN. Predictions are made from the resulting node embeddings with a two-layer MLP.