Benchmarking Positional Encodings for GNNs and Graph Transformers

Florian Grötschla
fgroetschla@ethz.ch
ETH Zurich
Zurich, Switzerland

Jiaqing Xie
jiaxie@ethz.ch
ETH Zurich
Zurich, Switzerland

Roger Wattenhofer
wattenhofer@ethz.ch
ETH Zurich
Zurich, Switzerland

Abstract

Positional Encodings (PEs) are essential for injecting structural information into Graph Neural Networks (GNNs) and Graph Transformers, yet their empirical impact remains poorly understood. We introduce a unified benchmarking framework that decouples PEs from architectural choices, enabling a fair comparison across 8 GNN and Transformer models, 9 PEs, and 10 synthetic and real-world datasets. Across more than 500 model–PE–dataset configurations, we find that commonly used expressiveness proxies, including Weisfeiler–Lehman distinguishability, do not reliably predict downstream performance. In particular, highly expressive PEs often fail to improve, and can even degrade performance on real-world tasks. At the same time, we identify several simple and previously overlooked model–PE combinations that match or outperform recent state-of-the-art methods. Our results highlight the strong task-dependence of PEs and underscore the need for empirical validation beyond theoretical expressiveness. To support reproducible research, we release an open-source benchmarking framework for evaluating PE for graph learning tasks.

CCS Concepts

• **Computing methodologies → Neural networks**; • **General and reference → Evaluation**; *Experimentation.*

Keywords

Graph Neural Networks, Graph Transformers, Positional Encodings

ACM Reference Format:
Florian Grötschla, Jiaqing Xie, and Roger Wattenhofer. 2026. Benchmarking Positional Encodings for GNNs and Graph Transformers. *ACM/IMS J. Data Sci.* 37, 4, Article 111 (August 2026), 12 pages. https://doi.org/10.1145/3770854.3785701

1 Introduction

Graphs are fundamental structures for modeling complex relationships in diverse domains, from social networks and molecular biology to recommendation systems. Graph Neural Networks (GNNs), particularly message-passing neural networks (MPNNs), have transformed graph learning through their powerful ability to aggregate information from local neighborhoods. This local aggregation paradigm has led to substantial progress in node classification, link prediction, and graph regression tasks [25, 45]. However, a significant limitation of MPNNs is their difficulty in capturing long-range dependencies, which are crucial in applications such as molecular interaction modeling and hierarchical social network analysis. To address these shortcomings, Graph Transformers (GTs) extend the self-attention framework to graphs, enabling global information exchange between all nodes [9]. Unlike sequences in natural language processing, graphs lack a natural positional ordering, making positional encodings (PEs) necessary for embedding structural information [1, 38]. These encodings provide geometric and topological information to otherwise position-agnostic neural architectures. Yet, incorporating effective PEs into graphs is considerably more difficult than in sequences because of the complex and non-linear structure of graph topology. Although PEs play a central role in graph learning, their impact is often conflated with architectural innovations. Existing evaluations typically assess PEs within specific models, making it difficult to isolate their individual contributions. For example, positional encodings such as RWSE (Random Walk Structural Encoding) and LapPE (Laplacian Positional Encoding) have been evaluated primarily alongside specific architectures such as GraphGPS [32] or Exphormer [35], rather than independently assessed across various models. Moreover, existing evaluations frequently rely on synthetic data or theoretical expressiveness metrics, such as Weisfeiler-Lehman (WL) distinguishability [34, 46], which often fail to correlate with practical performance on real-world tasks. This raises important questions regarding the utility of different PEs in various graph learning scenarios.

In this paper, we address these gaps through a comprehensive and systematic reassessment of positional encodings for GNNs. We propose a unified benchmarking framework explicitly designed to decouple the evaluation of PEs from architectural innovations. Our evaluation spans 8 graph architectures combined with 9 positional encodings across 10 synthetic and real-world datasets. In total, we analyze more than 500 model–PE–dataset configurations. To the best of our knowledge, this constitutes one of the broadest empirical evaluations of PEs across both MPNNs and GTs under a unified protocol. Our results reveal notable discrepancies between theoretical expressiveness and empirical performance. Positional encodings such as RWSE exhibit strong theoretical capabilities on synthetic benchmarks yet frequently fail to translate these strengths into consistent performance improvements on real-world datasets. In contrast, spectral PEs demonstrate a robust balance between theoretical and empirical effectiveness and provide more reliable results across varied graph datasets. Further, we uncover several combinations of established architectures and PEs that match or outperform recent state-of-the-art methods on multiple benchmarks. These

Authors' Contact Information: Florian Grötschla, fgroetschla@ethz.ch, ETH Zurich, Zurich, Switzerland; Jiaqing Xie, jiaxie@ethz.ch, ETH Zurich, Zurich, Switzerland; Roger Wattenhofer, wattenhofer@ethz.ch, ETH Zurich, Zurich, Switzerland.

This work is licensed under a Creative Commons Attribution 4.0 International License.
© 2026 Copyright held by the owner/author(s).
ACM 2831-3194/2026/8-ART111
https://doi.org/10.1145/3770854.3785701

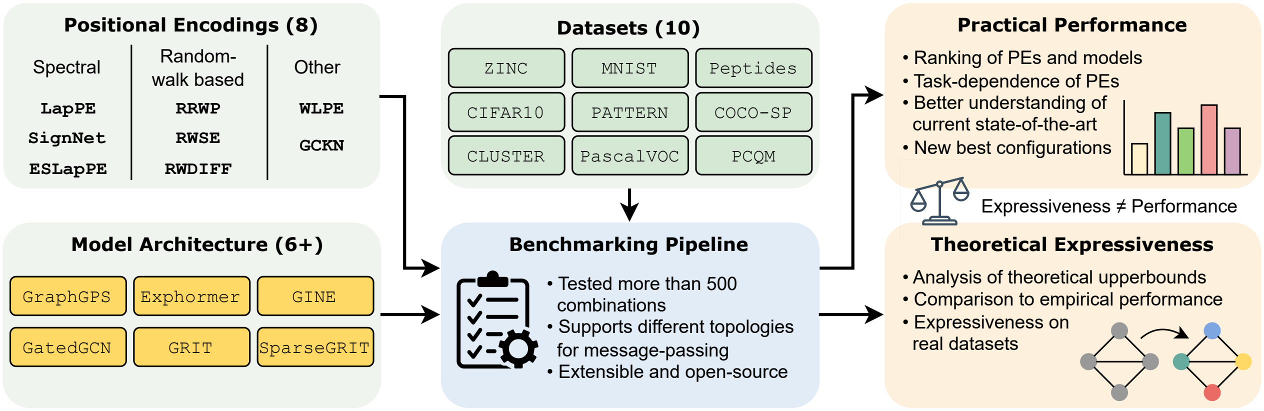
Positional Encodings (8)
Spectral
Random-walk based
Other
LapPE
SignNet
ESLapPE
RRWP
RWSE
RWDIFF
WLPE
GCKN
Datasets (10)
ZINC
MNIST
Peptides
CIFAR10
PATTERN
COCO-SP
CLUSTER
PascalVOC
PCQM
Practical Performance
• Ranking of PEs and models
• Task-dependence of PEs
• Better understanding of current state-of-the-art
• New best configurations
Expressiveness ≠ Performance
Model Architecture (6+)
GraphGPS
Exphormer
GINE
GatedGCN
GRIT
SparseGRIT
Benchmarking Pipeline
• Tested more than 500 combinations
• Supports different topologies for message-passing
• Extensible and open-source
Theoretical Expressiveness
• Analysis of theoretical upperbounds
• Comparison to empirical performance
• Expressiveness on real datasets

Figure 1: Conceptual overview of our benchmarking framework for PEs in GNNs and Graph Transformers. Our empirical results show that practical performance does not always align with theoretical expressiveness, challenging conventional assumptions in the literature. We identify new best-performing configurations by systematically exploring combinations across real and synthetic benchmarks on a wide variety of PEs, models and datasets.

findings challenge the prevailing assumption that newer architectures or encodings inherently outperform established methods. Our benchmarking thus not only highlights overlooked configurations but also provides empirical insights that can help practitioners shortlist PEs for a given task and compute budget, while emphasizing the need for task-specific validation. To facilitate reproducibility and continued evaluation in the community, we publicly release our benchmarking framework, including implementations of all PEs and model architectures tested in this study. In summary, our main contributions are as follows.

- We introduce a systematic benchmarking framework that explicitly isolates the evaluation of PEs from architectural innovations in GNNs and Graph Transformers.
- We empirically demonstrate that higher theoretical expressiveness does not reliably correlate with improved downstream performance across tasks and datasets.
- We identify multiple simple and previously underexplored combinations of established architectures and PEs that match or outperform recent state-of-the-art methods.
- We publicly release a reproducible and extensible open-source benchmark to support principled evaluation of PEs in future work[1].

2 Related Work

2.1 Graph Neural Network Architectures

Message-Passing GNNs. Early architectures like GCN [25], GraphSAGE [18], GAT [40], and GIN [46] established the message-passing framework, where nodes iteratively aggregate features from their neighbors. These models are bounded by the expressiveness of the Weisfeiler-Lehman (1-WL) test. Later variants improved this design: GatedGCN [3] adds gated residual connections, GINE [20] incorporates edge features, and PNA [7] leverages multi-statistic aggregators. Others extended expressiveness using subgraph or motif counts [48] or operating on higher-order structures such as cell complexes [2]. Still, most MPNNs struggle to capture long-range dependencies due to their inherently local nature.

Graph Transformers. Inspired by the transformer architecture [38], Graph Transformers capture long-range dependencies via global self-attention mechanisms. Fully-connected variants can be powerful [24], but typically require graph-specific inductive biases [21]. Models like Graphormer [47] introduce shortest-path and centrality encodings directly into attention, while GraphGPS [32] combines local MPNNs with global attention. Sparse attention models such as Exphormer [35] improve scalability, and GRIT [30] uses a more expressive attention mechanism with learnable edge updates. Transformers can integrate structural bias through learned masks [30] and match MPNN expressiveness when equipped with strong positional encodings [1] or virtual nodes [13]. In contrast, MPNNs can emulate Transformers under certain conditions [23].

2.2 Positional Encodings for Graphs

Unlike grids or sequences, graphs lack a natural order or coordinate system. PEs inject structural information into node representations and are indispensable for Graph Transformers, which would otherwise treat nodes as a permutation-invariant set [9, 10]. Even message-passing GNNs can benefit from PEs by receiving signals beyond the local neighborhood, thus improving their ability to capture long-range dependencies [11]. In general, PEs can be categorized into **spectral**, **random-walk**, and **other** types. Although this is not the only way to categorize PEs, it will prove useful in our evaluation as we generally see PEs of a group behaving similarly.

Spectral Encodings. Spectral PEs use eigenvectors of graph matrices (typically the Laplacian) as node coordinates. The Laplacian Positional Encoding (LapPE) [10] embeds the nodes along the principal directions of the graph. SignNet [28] addresses the sign and basis ambiguities of raw eigenvectors by learning invariant transformations. ESLapPE [41] enhances stability under graph isomorphisms. GCKN [31] approximates the spectral geometry through truncated kernel expansions.

[1] https://anonymous.4open.science/r/benchmarking-pes-2BD6

Table 1: Summary statistics of the datasets used in our benchmark, including graph size, connectivity, prediction level, task type, and evaluation metric. It spans both small synthetic graphs and larger real-world benchmarks from different domains.

Dataset	# Graphs	Avg. $\|\mathcal{N}\|$	Avg. $\|\mathcal{E}\|$	Directed	Prediction level	Prediction task	Metric
ZINC	12,000	23.2	24.9	No	graph	regression	Mean Abs. Error
MNIST	70,000	70.0	564.5	Yes	graph	10-class classif.	Accuracy
CIFAR10	60,000	117.6	941.1	Yes	graph	10-class classif.	Accuracy
PATTERN	14,000	118.9	2,359.2	No	inductive node	binary classif.	Accuracy
CLUSTER	12,000	117.2	1,510.9	No	inductive node	6-class classif.	Accuracy
PascalVOC-SP	11,355	479.4	2,710.5	No	inductive node	21-class classif.	F1 score
COCO-SP	123,286	476.4	2,693.7	No	inductive node	81-class classif.	F1 score
PCQM-Contact	529,434	30.1	69.1	No	inductive link	link ranking	MRR (Fil.)
Peptides-func	15,535	150.9	307.3	No	graph	10-task classif.	Avg. Precision
Peptides-struct	15,535	150.9	307.3	No	graph	11-task regression	Mean Abs. Error

Random-Walk and Diffusion Encodings. RWSE [10] encodes the probability of returning to the same node over multiple random walk lengths. Relative Random Walk PEs (RRWP), as used in GRIT [30], compute stationary distributions under personalized PageRank (PPR)-like schemes [15]. Related diffusion methods, such as heat kernels and resistance distances, have also been explored [8]. These encodings capture local and global connectivity, but can be computationally expensive for large graphs [36].

Others. Distance Encoding (DE) [27] enhances GNN expressiveness by adding distances to reference nodes. Graphormer [47] applies the shortest-path distances and edge connectivity as relative PE in the attention matrices. Node degrees, centrality, and WL-labels (WLPE) [9] also encode useful structural signals. Subgraph-based encodings [49] extend PEs with local structure indicators beyond the 1-hop neighborhood. These features can be used in either absolute or relative form, and are especially effective in Transformer-based models.

2.3 Benchmarking of PEs

Recent work has shown that improvements in graph models often stem from either the architecture or the PE, or both [1, 23]. For example, performance gains attributed to architectural changes in Transformers have sometimes been primarily due to the choice of PE. To fairly assess their respective contributions, it is crucial to decouple these components. Black et al. [1] provide a theoretical comparison of Graph Transformers with different PEs, demonstrating that certain encodings can render models equivalent in expressive power. Keriven and Vaiter [23] further analyze how the functional capacity of GNNs depends on the structure of their PEs. These studies suggest that sufficiently powerful PEs can enable even 1-WL-limited GNNs to solve more complex tasks, provided the model can make use of them.

Despite these insights, a practical benchmark for evaluating diverse combinations of PEs and architectures was lacking. Our work fills this gap by systematically evaluating a wide range of PEs (spectral, random-walk, structural, learnable) across both message-passing and Transformer-based models. By controlling for confounding factors, our benchmark isolates the effect of PEs across tasks and architectures. The results confirm that while PE performance may vary by task, some encodings consistently benefit many models. This supports the view that architectures and PEs can be designed independently, allowing researchers to combine strong components from both domains. Thus, our benchmark provides guidance for selecting or designing PEs in practice.

3 Topology and Attention

The effectiveness of PEs depends not only on their formulation, but also on how they interact with the underlying model architecture. In particular, the topology over which information is exchanged, whether local neighborhoods in message-passing GNNs or fully connected attention in Graph Transformers, can influence the extent to which a PE contributes to performance. For GTs, full self-attention allows each node to attend to all others and naturally allows for global information exchange. Although this design can compensate for the absence of long-range structural features in the input, it also reduces the reliance on PEs to encode this information. However, full attention comes with substantial computational overhead, particularly on large, sparse graphs. This raises a practical question: *To what extent is full attention necessary for competitive performance and does the answer depend on the presence of positional encodings?*

To study this, we introduce a sparsified variant of the GRIT architecture [30], referred to as *Sparse GRIT*. It retains GRIT's attention mechanism and edge update scheme, but restricts attention to a node's original neighbors, rather than using a fully connected topology. This effectively transforms the Graph Transformer into a sparse, local message-passing network, allowing us to isolate the impact of the update mechanism. To complement the analysis, we also evaluate message-passing convolutions on fully-connected graphs to test whether an attention mechanism is necessary to perform well on fully-connected graphs. This perspective allows us to treat all architectures as variants of MPNNs. In the case of (fully-connected) GTs, we only need to change the underlying message-passing topology to a fully-connected graph [39].

Sparse GRIT Convolution. Sparse GRIT applies the same edge update rules as GRIT [30], using updated edge encodings $\hat{\mathbf{e}}_{i,j}$, but only on the edges present in the input graph. This distinguishes it from fully connected attention mechanisms and also from local attention mechanisms like GAT, which do not update edge representations. The node update rule is given by:

$$\hat{\mathbf{x}}_i = \sum_{j \in \mathcal{N}(i)} \frac{e^{w_j \cdot \hat{\mathbf{e}}_{i,j}}}{\sum_{k \in \mathcal{N}(i)} e^{w_k \cdot \hat{\mathbf{e}}_{i,k}}} \cdot \left(\mathbf{W}_V \mathbf{x}_j + \mathbf{W}_{E_V} \hat{\mathbf{e}}_{i,j} \right),$$

where w_j denotes the attention weight, and $\mathbf{W}_V, \mathbf{W}_{E_V}$ are learnable projection matrices. The key distinction from GRIT is that attention is computed only over local neighborhoods $\mathcal{N}(i)$, using a sparse softmax normalization.

This design allows Sparse GRIT to scale more efficiently while preserving the core inductive biases of GRIT. Importantly, it enables us to evaluate how positional encodings perform under different connectivity regimes. As we show in Section 5, Sparse GRIT often matches the performance of GRIT, despite using significantly fewer edges. This suggests that full attention may not be required in settings where local structure is predictive and that PEs become increasingly important in sparser architectures where less global information is available through the model itself.

4 Benchmarking Framework

We perform a benchmarking of state-of-the-art models combined with commonly used PEs to identify optimal configurations. This analysis addresses a common gap in the literature, where new PEs are introduced alongside novel architectures but are rarely evaluated independently of existing models. By decoupling architectures from PEs, our approach enables a comprehensive exploration of possible combinations. To enable the evaluation of models and future research for measuring the impact of positional encodings, we provide a unified codebase that includes the implementation of all tested models and the respective positional encodings. We base the code on GraphGPS [32] and integrate all missing implementations. This makes for reproducible results and easy extensibility for new datasets, models, or positional encodings. Our codebase also provides readily available implementations for NodeFormer [43], Difformer [42], GOAT [26], GraphTrans [44], GraphiT [31], and SAT [4] that are based on the respective original codebases.

In our experiments, we used five different random seeds for the BenchmarkingGNNs datasets [10] and four for the others. All experiments can be executed on a single Nvidia RTX 3090 (24GB) or a single RTX A6000 (40GB). To avoid out-of-memory (OOM) issues on LRGB datasets, we reserve up to 100GB of host memory for preprocessing positional encodings. For configurations that exceeded this compute/memory envelope (e.g., due to PE preprocessing or attention quadratic costs), we mark them as infeasible and exclude them consistently across models.

4.1 Datasets and Model Configurations

We begin by describing the datasets and model configurations used in our benchmark, which define the fixed experimental setting under which all positional encodings are evaluated.

BenchmarkingGNNs includes *MNIST*, *CIFAR10*, *CLUSTER*, *PATTERN*, and *ZINC*, following the protocols established in *GraphGPS* [32], *Exphormer* [35], and *GRIT* [30]. These datasets have traditionally been used to benchmark Graph Neural Networks (GNNs) [10], excluding graph transformers. We adhere to established settings from the relevant literature for each model. Specifically, for GatedGCN and GraphGPS, we follow the configurations detailed for GraphGPS [32]. For Exphormer, we utilize the settings from the original paper [35]. For GINE, Sparse GRIT, and global GRIT models, we adopt the configurations from GRIT [30].

Long-Range Graph Benchmark (LRGB) [12] includes *Peptides-func*, *Peptides-struct*, *PascalVOC-SP*, *PCQM-Contact*, and *COCO*. The tasks have been developed to necessitate long range interactions. We consider four models: GatedGCN, GraphGPS, Exphormer, and Sparse GRIT. For GatedGCN and GraphGPS, we mainly follow the fine-tuned configurations as described by Tönshoff et al. [36]. For sparse GRIT, we adopt the hyperparameters used for the Peptides-func and Peptides-struct datasets and transfer these settings to COCO-SP, Pascal-VOC, and PCQM-Contact, as detailed by Dwivedi et al. [12]. For Exphormer, we follow the configurations proposed by Shirzad et al. [35].

Table 2 shows that, while absolute performance varies under equal-budget tuning, the relative ordering of positional encodings is stable across the depth and dropout settings tested. This ablation is not exhaustive: a full per-configuration hyperparameter search over > 500 combinations would be computationally infeasible and is not standard practice for these datasets [12, 32]. Instead, these controlled checks provide evidence that our main ranking trends are robust to non-trivial hyperparameter variation.

Scalability to Larger Graphs. We additionally evaluated positional encoding preprocessing on large-scale graphs such as ogbn-products and ogbn-mag. In these settings, only no-PE and lightweight encodings such as WLPE are feasible. Spectral and random-walk-based encodings (e.g., LapPE, RWSE) exceed available memory due to dense matrix operations. As our goal is a controlled comparison of the effectiveness of PE, such datasets are outside the feasible scope of our benchmark, at least with the PEs we evaluate. Implementations and configuration files for large-scale graphs are available in the codebase. Statistics and prediction tasks for the datasets used are listed in Table 1.

Table 2: Robustness of positional encoding rankings under controlled equal-budget tuning. We vary model depth (L) and dropout (D) for representative settings. CIFAR10 is evaluated using accuracy (↑), while ZINC is evaluated using mean absolute error (MAE, ↓). Absolute performance changes, but relative PE rankings remain stable.

Dataset / Model	Var.	LapPE	RWSE	SignNet
CIFAR10 / GRIT	$L = 3$	73.33±0.51	73.65±0.62	72.81±0.48
	$L = 5$	74.45±0.28	75.28±0.43	73.97±0.05
	$L = 7$	76.03±0.31	76.45±0.16	75.86±0.54
	$D = 0.1$	74.03±1.17	74.05±0.11	73.34±0.38
	$D = 0.3$	73.48±1.04	73.30±0.79	73.26±0.19
	$D = 0.5$	73.33±0.51	73.65±0.62	72.81±0.48
ZINC / GatedGCN	$L = 4$	0.172±0.002	0.102±0.003	0.106±0.002
	$L = 6$	0.155±0.004	0.103±0.007	0.106±0.002
	$L = 8$	0.167±0.006	0.116±0.006	0.110±0.002
	$D = 0.0$	0.172±0.002	0.102±0.003	0.106±0.002
	$D = 0.3$	0.192±0.010	0.135±0.050	0.147±0.000
	$D = 0.5$	0.260±0.005	0.205±0.009	0.218±0.021

Table 3: Results for the best-performing models and the PE they use for the BenchmarkingGNNs datasets. All runs except those for EGT and TIGT were done by us. SparseGRIT performs on par with GRIT on most datasets, indicating that, on the datasets we evaluate, full attention is not always necessary to achieve competitive performance. We color the best, second best, and third best models.

Model	CIFAR10 ↑		CLUSTER ↑		MNIST ↑		PATTERN ↑		ZINC ↓	
EGT [22]	68.70 ± 0.41		79.23 ± 0.35		98.17 ± 0.09		86.82 ± 0.02		0.108 ± 0.009	
TIGT [6]	73.96 ± 0.36		78.03 ± 0.22		98.23 ± 0.13		86.68 ± 0.06		0.057 ± 0.002	
GINE	66.14 ± 0.31	(ESLapPE)	59.66 ± 0.63	(SignNet)	97.75 ± 0.10	(RWDIFF)	86.69 ± 0.08	(RWSE)	0.075 ± 0.006	(RWDIFF)
GatedGCN	69.57 ± 0.79	(RRWP)	75.29 ± 0.05	(SignNet)	97.91 ± 0.08	(RRWP)	86.83 ± 0.03	(RWSE)	0.102 ± 0.003	(RWSE)
SparseGRIT	74.95 ± 0.26	(RRWP)	79.87 ± 0.08	(RRWP)	98.12 ± 0.05	(RWSE)	87.17 ± 0.04	(RRWP)	0.065 ± 0.003	(RRWP)
Exphormer	75.21 ± 0.10	(LapPE)	78.28 ± 0.21	(SignNet)	98.42 ± 0.18	(RRWP)	86.82 ± 0.04	(RWSE)	0.092 ± 0.007	(SignNet)
GRIT	75.66 ± 0.41	(RRWP)	79.81 ± 0.11	(RRWP)	98.12 ± 0.14	(RRWP)	87.22 ± 0.03	(RRWP)	0.059 ± 0.001	(RRWP)
GatedGCN (FC)	71.08 ± 0.60	(RRWP)	74.78 ± 0.46	(SignNet)	98.20 ± 0.15	(GCKN)	86.85 ± 0.02	(RWSE)	0.114 ± 0.003	(RWSE)
GraphGPS	72.31 ± 0.20	(noPE)	78.31 ± 0.11	(SignNet)	98.18 ± 0.12	(ESLapPE)	86.87 ± 0.01	(RWSE)	0.074 ± 0.006	(RWSE)

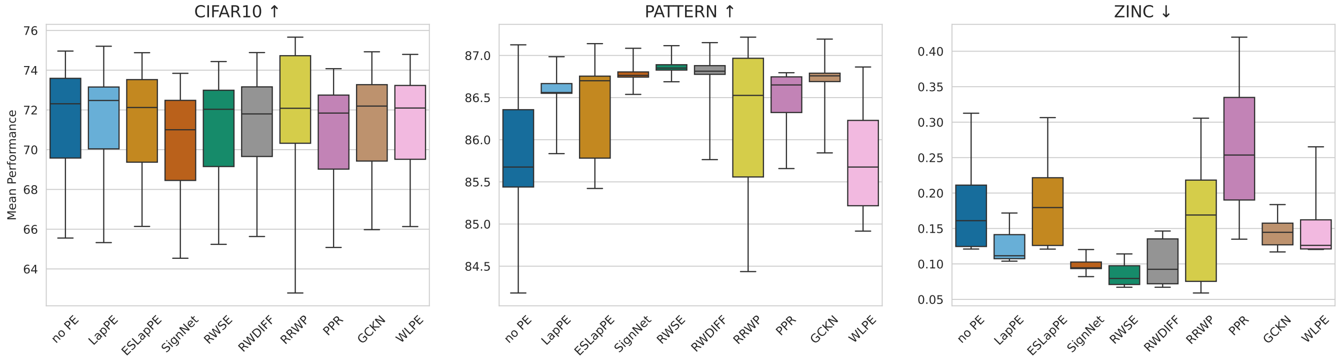

Figure 2: Performance comparison of target metrics across selected datasets from BenchmarkingGNNs. The boxplots illustrate the performance range for all models included in the study, with whiskers representing the minimum and maximum performance observed. Notably, RRWP consistently achieves the best results, whereas certain PEs, such as SignNet on CIFAR10, can sometimes decrease performance relative to the baseline without PEs.

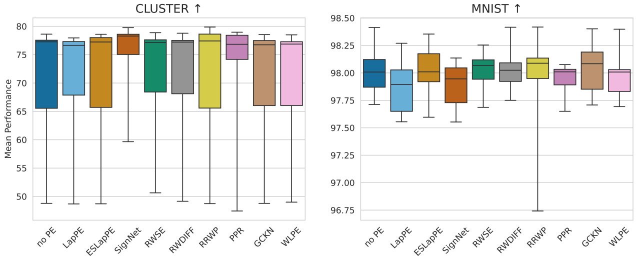

Figure 3: Mean performance of different positional encodings on more datasets from the BenchmarkingGNNs.

4.2 WL Distinguishability

Several parts of our analysis rely on *WL distinguishability* as a proxy for the structural expressiveness induced by a PE. We therefore formally define this notion here.

Given a graph $G = (V, E)$ with initial node features (including positional encoding), we run r rounds of the 1-dimensional Weisfeiler–Lehman (1-WL) algorithm. Let p denote the number of distinct node color classes obtained after r refinement rounds. We define the WL distinguishability score as

$$\text{WL-dist}(G) = \frac{p}{|V|} \in (0, 1]. \tag{1}$$

A value of 1 indicates that all nodes are structurally distinguishable under 1-WL, while smaller values indicate that nodes cannot be uniquely identified. For example, on a cycle graph with identical initial node features, 1-WL assigns all nodes the same color in every round, so $p = 1$ and the score equals $1/|V|$ (minimal distinguishability).

Interpretation and Limitations. WL distinguishability provides an upper bound on the structural discrimination capacity that a message-passing GNN could theoretically exploit when equipped with a given PE. However, it is agnostic to task semantics, noise, and feature distributions. As a result, higher WL distinguishability does not necessarily translate into improved downstream performance. In some cases, highly expressive PEs may introduce inductive biases

Table 4: Best-performing models and PEs for the LRGB datasets under the LRGB evaluation protocol. On PCQM-Contact, Exphormer+LapPE achieves the best performance among the methods evaluated in our benchmark.

Model	COCO-SP ↑		PCQM-Contact ↑		PascalVOC-SP ↑		Peptides-func ↑		Peptides-struct ↓	
GCN [36]	13.38 ± 0.07		45.26 ± 0.06		0.78 ± 0.31		68.60 ± 0.50		24.60 ± 0.07	
GINE [36]	21.25 ± 0.09		46.17 ± 0.05		27.18 ± 0.54		66.21 ± 0.67		24.73 ± 0.17	
GatedGCN [36]	29.22 ± 0.18		46.70 ± 0.04		38.80 ± 0.40		67.65 ± 0.47		24.77 ± 0.09	
CRaWl [37]	-		-		45.88 ± 0.79		70.74 ± 0.32		25.06 ± 0.22	
S^{2}GCN [16]	-		-		-		73.11 ± 0.66		24.47 ± 0.32	
DRew [17]	-		34.42 ± 0.06		33.14 ± 0.24		71.50 ± 0.44		25.36 ± 0.15	
Graph ViT [19]	-		-		-		68.76 ± 0.59		24.55 ± 0.27	
GatedGCN-VN [33]	32.44 ± 0.25		-		44.77 ± 1.37		68.23 ± 0.69		24.75 ± 0.18	
Exphormer	34.85 ± 0.11	(ESLapPE)	47.37 ± 0.24	(LapPE)	42.42 ± 0.44	(LapPE)	64.24 ± 0.63	(LapPE)	24.96 ± 0.13	(LapPE)
GraphGPS	38.91 ± 0.33	(RWSE)	46.96 ± 0.17	(LapPE)	45.38 ± 0.83	(ESLapPE)	66.20 ± 0.73	(LapPE)	24.97 ± 0.24	(LapPE)
SparseGRIT	19.76 ± 0.38	(noPE)	45.85 ± 0.11	(LapPE)	35.19 ± 0.40	(GCKN)	67.02 ± 0.80	(RRWP)	24.87 ± 0.14	(LapPE)
GRIT	21.28 ± 0.08	(RWDIFF)	46.08 ± 0.07	(SignNet)	35.56 ± 0.19	(noPE)	68.65 ± 0.50	(RRWP)	24.54 ± 0.10	(RRWP)

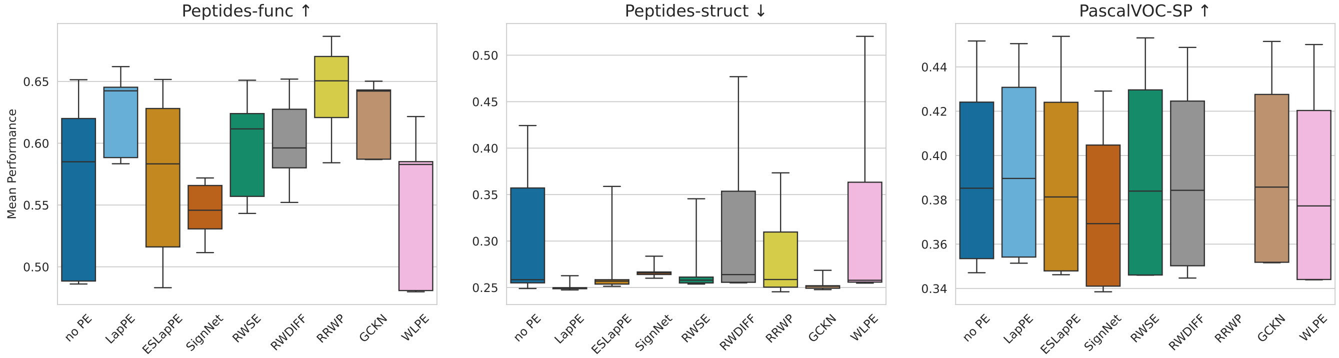

Figure 4: Performance comparison of target metrics across selected datasets from the Long-Range Graph Benchmark. The boxplots illustrate the performance range of all models included in the study, with whiskers indicating the minimum and maximum performance observed. Plots for the remaining datasets are provided in Figure 5.

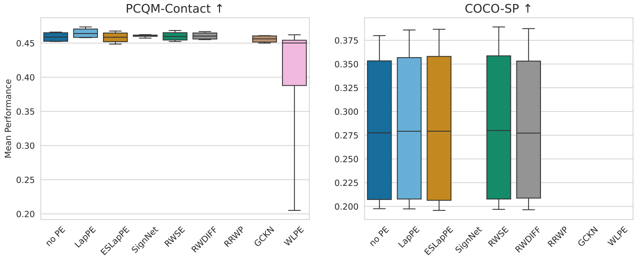

Figure 5: Mean performance of different positional encodings on more datasets from the Long Range Graph Benchmark.

that are misaligned with the task, leading to overfitting or degraded performance.

5 Performance Comparison

Based on the framework we established in Section 4, we benchmark the performance of different PEs on the BenchmarkingGNNs [10] and LRGB [12] datasets. Throughout this section, we emphasize relative ranking trends rather than absolute performance values, as our goal is to compare positional encodings under standardized settings rather than to optimize individual configurations.

Benchmarking GNNs Datasets. We first conduct a dataset-centric analysis where we assess the impact of various PEs on model performance. Figure 2 presents the range of target metric values achieved across different PEs, aggregated over all models. Unaggregated results are provided in the Appendix. Although most of the prior results were reproducible, we consistently observed slightly lower values for GRIT, even when using its official codebase and configurations. Our findings reveal that PEs can significantly influence model performance, with the best choice of PE varying depending on the dataset and task. However, PEs can also negatively impact performance in some cases. For example, while RRWP performs best on the CIFAR10 dataset and ZINC, there are not always clear winners. Sometimes, good performance can be achieved even without any positional encoding (e.g., for PATTERN). This is also evident when examining the best-performing configurations for each model and PE. The full set of runs (all model–PE–dataset configurations), including configuration files and raw logs, is available in our released benchmark codebase. We summarize the best-performing configurations for the BenchmarkingGNNs datasets in Table 3, where we

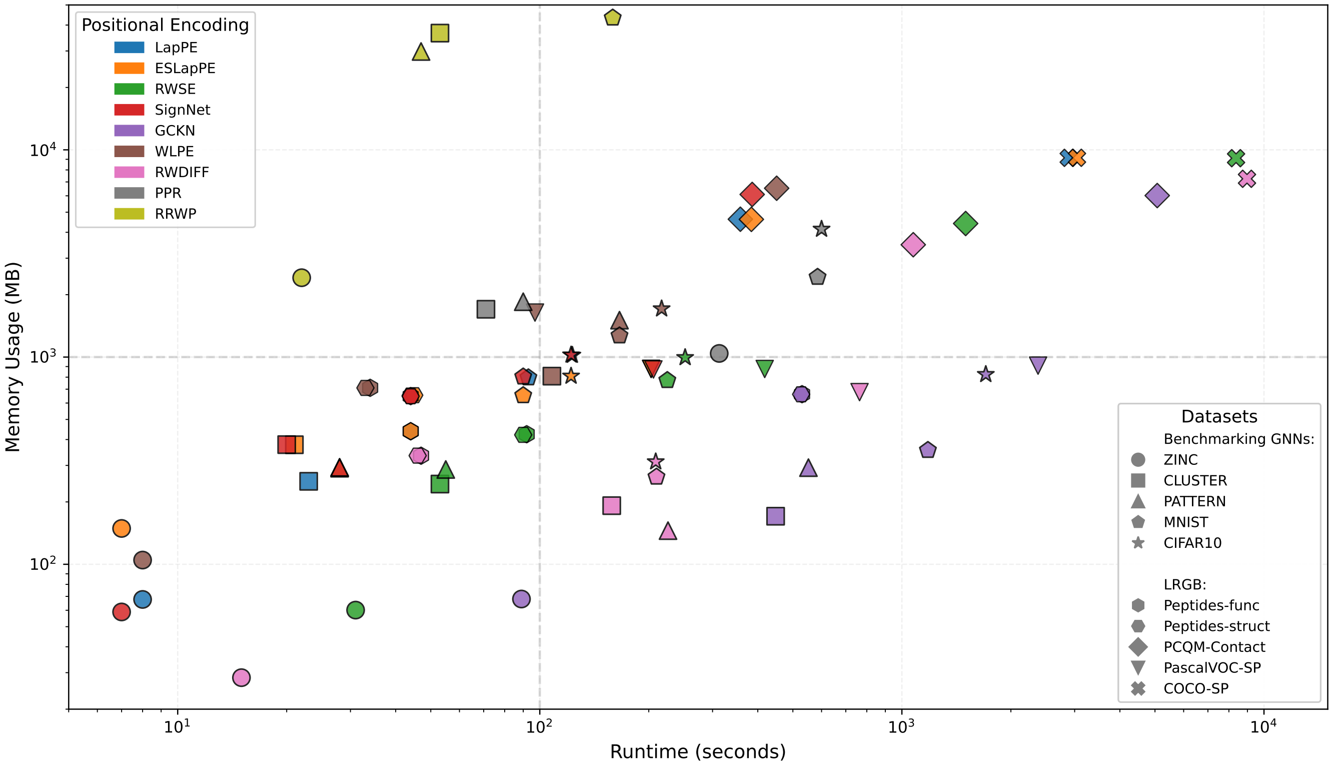

Figure 6: Computational requirements of positional encodings across benchmark datasets. We compare runtime versus memory usage. LapPE and ESLapPE are amongst the most efficient. In contrast, RRWP exhibits the highest resource consumption, with memory usage exceeding 40,000 MB on several datasets and failing to compute (OOM) on larger instances. SignNet, GCKN, and WLPE also encounter memory limitations on the COCO-SP dataset.

can observe which PE led to the best performance for each model and dataset. This enables a fair comparison of all architectures and helps to determine which PE works best.

In our comparison, we observe that the sparse GRIT convolution emerges as the best graph convolution for sparse topologies. It competes effectively with full GRIT attention in most datasets. This suggests that these datasets do not require extensive long-range information exchange and can achieve strong performance with sparse message-passing. The GatedGCN convolution on the fully-connected graph does perform better than the original overall, but generally lags behind attention-based layers. Regarding the effectiveness of different PEs, random-walk-based encodings such as RRWP and RWSE consistently perform well in all tested models. The only notable exception is the CLUSTER dataset, where SignNet performs competitively for some architectures, although the best results are still achieved with RRWP.

Long-Range Graph Benchmark. We extend our evaluation to the LRGB datasets and use hyperparameter configurations based on those of Tönshoff et al. [36], with results presented in Table 4. In these datasets, Laplacian-based encodings generally outperform others (except for the Peptides variations), likely due to their ability to capture more global structure in the slightly larger graphs. This might also be reflected in the fact that transformer-based architectures or models that facilitate global information exchange consistently perform better. Our findings largely align with previous rankings, except for PCQM-Contact, where Exphormer achieves the best performance among the methods evaluated in our benchmark, which underscores the importance of thorough benchmarking of existing models.

Figure 4 further analyzes the performance of the employed PEs. It should be noted that RRWP could not be utilized for larger datasets due to its significant memory footprint and computational complexity, similar to models employing full attention mechanisms. The results align with our previous analysis and show that on datasets like Peptides-func, the PE has a consistent impact on the performance, even when the values are aggregated over different architectures. This impact can also be negative compared to the baseline that does not use any PE. On other datasets (for example, PascalVOC-SP), the PE seems to play a lesser role, and good results can be achieved without any PE.

5.1 Preprocessing Cost of Positional Encodings

Although much of the existing literature focuses on the expressiveness and accuracy impact of PEs, practical adoption also depends on their computational overhead. Some PEs require solving large

Table 5: WL distinguishability scores for positional encodings under sparse and fully-connected topologies on synthetic datasets. RWSE and LapPE consistently achieve perfect or near-perfect scores. Higher expressiveness under idealized conditions does not necessarily translate to better downstream performance.

PE	Topology	LIMITS 1	LIMITS 2	SKIP-CIRCLES	TRIANGLES	4-CYCLES
no PE	sparse	0.5	0.5	0.1	0.65	0.5
	fully-connected	0.5	0.5	0.1	0.65	0.69
GCKN	sparse	1.0	1.0	0.35	0.65	0.94
	fully-connected	1.0	1.0	0.7	0.82	1.0
GPSE	sparse	1.0	1.0	0.25	0.95	0.52
	fully-connected	1.0	1.0	0.1	1.0	0.48
LapPE	sparse	1.0	1.0	0.75	0.65	1.0
	fully-connected	1.0	1.0	1.0	0.83	1.0
RWSE	sparse	1.0	1.0	1.0	0.97	1.0
	fully-connected	1.0	1.0	1.0	1.0	1.0
RRWP	sparse	1.0	1.0	0.5	0.65	1.0
	fully-connected	1.0	1.0	1.0	0.99	1.0
SignNet	sparse	0.75	1.0	1.0	0.89	1.0
	fully-connected	1.0	1.0	1.0	0.99	1.0

linear systems, computing high-order spectral decompositions, or simulating random walks, all of which can become prohibitive on large-scale graphs. To quantify this overhead, we benchmark the preprocessing cost of each PE in terms of both runtime and peak memory usage. We report measurements on four representative datasets that differ in size and structure: ZINC, CLUSTER, PCQM-Contact, and COCO-SP. This subset includes both molecule-level graphs and large image-derived superpixel graphs. Figure 6 summarizes the results. Among the evaluated PEs, LapPE, ESLapPE, and SignNet show consistently low preprocessing time and memory usage. In contrast, GCKN, RWSE, and RWDIFF incur significantly higher costs, particularly on larger graphs. For example, on COCO-SP, both SignNet and GCKN exceeded GPU memory limits during preprocessing and could not be run. This analysis highlights an important trade-off: while more expressive PEs may offer potential performance benefits, their computational cost can limit practical applicability in real-world systems, particularly on high-resolution graphs or large datasets. As such, empirical performance should be evaluated not only in terms of accuracy, but also in light of resource constraints and scalability.

6 The Impact of Expressiveness

Prior work has primarily evaluated PEs through theoretical lenses such as WL distinguishability [1, 23]. While such metrics capture structural expressiveness, their practical relevance remains uncertain. We therefore investigate whether higher theoretical expressiveness correlates with downstream performance, using both synthetic expressiveness datasets and real-world benchmarks.

We first assess PEs on synthetic datasets designed to test specific structural distinctions. These include LIMITS 1 and LIMITS 2 [14], SKIP-CIRCLES [5], TRIANGLES [34], and 4-CYCLES [29]. We evaluate both sparse and fully connected topologies using GIN and GRIT and report the results in table 5. RWSE consistently achieves perfect scores across tasks and topologies, followed by LapPE and SignNet. GCKN and GPSE show more variable results and perform poorly on structure-sensitive datasets such as SKIP-CIRCLES and 4-CYCLES. These results confirm that many PEs are capable of encoding complex structural patterns under idealized conditions. However, this synthetic expressiveness does not consistently translate to improved performance on real-world datasets. For example, while RWSE achieves perfect accuracy in synthetic tasks, it fails to outperform simpler PEs on benchmarks such as ZINC and Peptides-func. LapPE, in contrast, provides a more balanced profile, performing well in both controlled and practical settings. These observations highlight the limitations of synthetic evaluations: although useful for isolating specific capabilities, they cannot predict the effectiveness of PEs under more realistic conditions, where noise, feature distributions, and task semantics come into play.

6.1 Expressiveness ≠ Performance

To further examine the practical value of theoretical expressiveness, we analyze WL distinguishability induced by different PEs on real-world datasets (formally defined in Section 4.2). Although this metric offers an upper bound on the discriminative capacity that a GNN might achieve with a given PE, it does not imply that full distinguishability is necessary or even beneficial for a given task. In fact, we find that empirical performance often does not align with this score. For image-derived datasets such as CIFAR10, MNIST, PascalVOC-SP, and COCO-SP, node features (e.g., pixel coordinates) are inherently unique, which renders additional expressiveness from PEs unnecessary. This aligns with our experimental evaluation, where PEs yield marginal improvements in these settings. More interesting patterns emerge in datasets such as ZINC, Peptides, and PCQM-Contact, which we show in Figure 7, 8 and 9. For ZINC, Laplacian-based PEs achieve nearly perfect WL scores but do not translate to the strongest empirical results, which contradicts common expectations. For the Peptides datasets, SignNet exhibits high WL distinguishability but underperforms even the baseline without PE on Peptides-func (fig. 4). In the PCQM-Contact dataset, we observe a divergence between WL-based expressiveness and empirical performance. Spectral encodings such

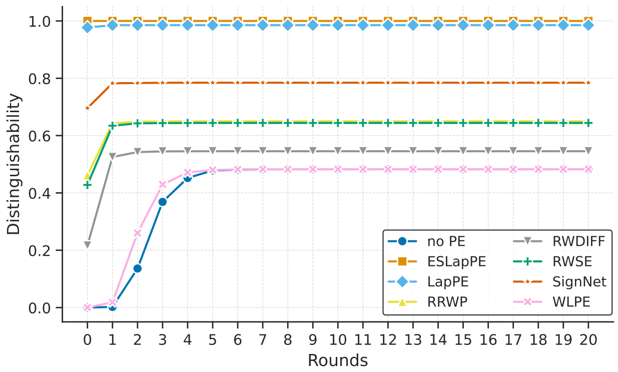

Figure 7: WL distinguishability scores on the ZINC dataset. Spectral PEs such as LapPE and ESLapPE achieve near-perfect node distinguishability. However, this does not consistently align with empirical performance.

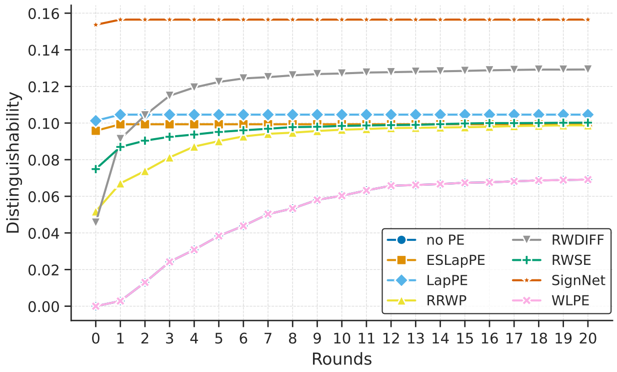

Figure 8: WL distinguishability for Peptides. None of the evaluated configurations achieves full distinguishability and the one with the highest expressiveness (SignNet) is among the weakest performers on this benchmark.

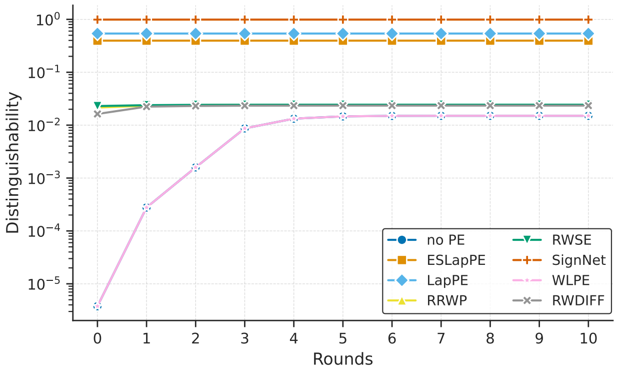

Figure 9: WL distinguishability on PCQM-Contact. Laplacian-based PEs yield much higher expressiveness than random-walk based ones.

as LapPE and ESLapPE achieve high WL distinguishability, which indicates a strong theoretical capacity to differentiate node structures. However, similar to our findings on ZINC and Peptides-func, this increased expressiveness does not consistently translate into better predictive performance. In fact, some PEs with lower WL scores, such as RRWP, achieve competitive or even superior results in practice.

These findings reveal a critical disconnect: higher WL distinguishability does not imply better downstream performance. One plausible explanation for this disconnect is that highly expressive positional encodings capture structural distinctions that are weakly aligned, or even misaligned, with task-relevant signals. In contrast, moderately expressive encodings may provide inductive biases that better match the semantics of real-world prediction tasks, leading to stronger generalization despite lower theoretical expressiveness. In general, our results show that theoretical expressiveness is not a reliable predictor of practical utility. WL distinguishability, while informative about structural capacity, fails to account for dataset-specific factors such as noise, feature informativeness, and task complexity. Consequently, empirical validation remains essential for PE selection. These insights emphasize the need for rigorous benchmarking, as outlined in section 5, to assess the practical effectiveness of PEs. Expressiveness alone does not guarantee better performance, and in some cases may even be detrimental.

Limitations and Scope. Our analysis is primarily empirical and descriptive in nature. While we observe systematic mismatches between WL distinguishability and downstream performance (e.g., SignNet on Peptides or LapPE on ZINC), providing a causal explanation is challenging due to complex interactions between PEs, model architectures, optimization dynamics, and dataset characteristics. Consequently, our study aims to expose patterns and counterexamples to commonly held assumptions, rather than to make causal claims about the mechanisms underlying PE effectiveness. From a practical perspective, our results suggest that PEs should be selected with care and empirically validated. Encodings that achieve high WL distinguishability do not consistently yield better performance and in some cases may even be detrimental. In contrast, PEs that offer a more moderate trade-off between expressiveness and computational cost often perform competitively across datasets. These observations should be interpreted as empirical regularities within our benchmark, rather than as prescriptive recommendations.

7 Conclusions

This paper presents a comprehensive investigation into the role of Positional Encodings in Graph Neural Networks and Graph Transformers through a systematic and reproducible benchmarking framework. By decoupling PE evaluation from architectural innovations, we assess the empirical utility and theoretical expressiveness of PEs across a broad range of datasets and models. Our findings reveal that the effectiveness of PEs is highly dependent on the task and the dataset. In particular, theoretical expressiveness, as measured by WL distinguishability, does not reliably predict downstream performance, which underscores the need for empirical validation. While random-walk-based PEs such as RRWP and RWSE frequently perform well, they incur significant preprocessing cost and memory overhead. In our benchmark, spectral PEs (e.g., LapPE, ESLapPE) often provide a favorable trade-off between accuracy and preprocessing cost. In contrast, image-derived graphs with unique node features (e.g., CIFAR10) sometimes benefit little from PEs, suggesting that their utility depends strongly on input feature distributions. Our benchmark also provides insights into architectural design. For instance, we show that sparsified attention mechanisms, as implemented in SparseGRIT, can match the performance of fully connected GTs while significantly reducing computational cost. These results challenge the assumption that full attention is always necessary and demonstrate that careful design of local message-passing topologies, combined with suitable PEs, can yield highly competitive models. To support future research, we publicly release our benchmarking codebase, including standardized implementations of all models and PEs. We hope this resource facilitates more principled evaluation and design of positional encodings in graph representation learning.

References

[1] Mitchell Black, Zhengchao Wan, Gal Mishne, Amir Nayyeri, and Yusu Wang. 2024. Comparing Graph Transformers via Positional Encodings. *arXiv preprint arXiv:2402.14202* (2024).

[2] Cristian Bodnar, Fabrizio Frasca, Nina Otter, Yuguang Wang, Pietro Lio, Guido F Montufar, and Michael Bronstein. 2021. Weisfeiler and lehman go cellular: Cw networks. *Advances in neural information processing systems* 34 (2021), 2625–2640.

[3] Xavier Bresson and Thomas Laurent. 2017. Residual gated graph convnets. *arXiv preprint arXiv:1711.07553* (2017).

[4] Dexiong Chen, Leslie O'Bray, and Karsten Borgwardt. 2022. Structure-aware transformer for graph representation learning. In *International Conference on Machine Learning*. PMLR, 3469–3489.

[5] Zhengdao Chen, Soledad Villar, Lei Chen, and Joan Bruna. 2019. On the equivalence between graph isomorphism testing and function approximation with gnns. *Advances in neural information processing systems* 32 (2019).

[6] Yun Young Choi, Sun Woo Park, Minho Lee, and Youngho Woo. 2024. Topology-Informed Graph Transformer. *arXiv preprint arXiv:2402.02005* (2024).

[7] Gabriele Corso, Luca Cavalleri, Dominique Beaini, Pietro Liò, and Petar Veličković. 2020. Principal neighbourhood aggregation for graph nets. *Advances in Neural Information Processing Systems* 33 (2020), 13260–13271.

[8] Xiaowen Dong, Dorina Thanou, Michael Rabbat, and Pascal Frossard. 2019. Learning graphs from data: A signal representation perspective. *IEEE Signal Processing Magazine* 36, 3 (2019), 44–63.

[9] Vijay Prakash Dwivedi and Xavier Bresson. 2020. A generalization of transformer networks to graphs. *arXiv preprint arXiv:2012.09699* (2020).

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

[11] Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. 2021. Graph neural networks with learnable structural and positional representations. *arXiv preprint arXiv:2110.07875* (2021).

[12] Vijay Prakash Dwivedi, Ladislav Rampášek, Michael Galkin, Ali Parviz, Guy Wolf, Anh Tuan Luu, and Dominique Beaini. 2022. Long range graph benchmark. *Advances in Neural Information Processing Systems* 35 (2022), 22326–22340.

[13] Dongqi Fu, Zhigang Hua, Yan Xie, Jin Fang, Si Zhang, Kaan Sancak, Hao Wu, Andrey Malevich, Jingrui He, and Bo Long. 2024. VCR-Graphormer: A Mini-batch Graph Transformer via Virtual Connections. *arXiv preprint arXiv:2403.16030* (2024).

[14] Vikas Garg, Stefanie Jegelka, and Tommi Jaakkola. 2020. Generalization and representational limits of graph neural networks. In *International Conference on Machine Learning*. PMLR, 3419–3430.

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

[16] Simon Geisler, Arthur Kosmala, Daniel Herbst, and Stephan Günnemann. 2024. Spatio-Spectral Graph Neural Networks. *arXiv preprint arXiv:2405.19121* (2024).

[17] Benjamin Gutteridge, Xiaowen Dong, Michael M Bronstein, and Francesco Di Giovanni. 2023. Drew: Dynamically rewired message passing with delay. In *International Conference on Machine Learning*. PMLR, 12252–12267.

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

[19] Xiaoxin He, Bryan Hooi, Thomas Laurent, Adam Perold, Yann LeCun, and Xavier Bresson. 2023. A generalization of vit/mlp-mixer to graphs. In *International Conference on Machine Learning*. PMLR, 12724–12745.

[20] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. 2019. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265* (2019).

[21] Md Shamim Hussain, Mohammed J Zaki, and Dharmashankar Subramanian. 2021. Edge-augmented graph transformers: Global self-attention is enough for graphs. *arXiv preprint arXiv:2108.03348* 3 (2021).

[22] Md Shamim Hussain, Mohammed J Zaki, and Dharmashankar Subramanian. 2022. Global self-attention as a replacement for graph convolution. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 655–665.

[23] Nicolas Keriven and Samuel Vaiter. 2024. What functions can Graph Neural Networks compute on random graphs? The role of Positional Encoding. *Advances in Neural Information Processing Systems* 36 (2024).

[24] Jinwoo Kim, Dat Nguyen, Seonwoo Min, Sungjun Cho, Moontae Lee, Honglak Lee, and Seunghoon Hong. 2022. Pure transformers are powerful graph learners. *Advances in Neural Information Processing Systems* 35 (2022), 14582–14595.

[25] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[26] Kezhi Kong, Jiuhai Chen, John Kirchenbauer, Renkun Ni, C Bayan Bruss, and Tom Goldstein. 2023. GOAT: A global transformer on large-scale graphs. In *International Conference on Machine Learning*. PMLR, 17375–17390.

[27] Pan Li, Yanbang Wang, Hongwei Wang, and Jure Leskovec. 2020. Distance encoding: Design provably more powerful neural networks for graph representation learning. *Advances in Neural Information Processing Systems* 33 (2020), 4465–4478.

[28] Derek Lim, Joshua Robinson, Lingxiao Zhao, Tess Smidt, Suvrit Sra, Haggai Maron, and Stefanie Jegelka. 2022. Sign and basis invariant networks for spectral graph representation learning. *arXiv preprint arXiv:2202.13013* (2022).

[29] Andreas Loukas. 2019. What graph neural networks cannot learn: depth vs width. *arXiv preprint arXiv:1907.03199* (2019).

[30] Liheng Ma, Chen Lin, Derek Lim, Adriana Romero-Soriano, Puneet K Dokania, Mark Coates, Philip Torr, and Ser-Nam Lim. 2023. Graph inductive biases in transformers without message passing. In *International Conference on Machine Learning*. PMLR, 23321–23337.

[31] Grégoire Mialon, Dexiong Chen, Margot Selosse, and Julien Mairal. 2021. GraphiT: Encoding Graph Structure in Transformers. arXiv:2106.05667 [cs.LG]

[32] Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. 2022. Recipe for a general, powerful, scalable graph transformer. *Advances in Neural Information Processing Systems* 35 (2022), 14501–14515.

[33] Eran Rosenbluth, Jan Tönshoff, Martin Ritzert, Berke Kisin, and Martin Grohe. 2024. Distinguished In Uniform: Self Attention Vs. Virtual Nodes. *arXiv preprint arXiv:2405.11951* (2024).

[34] Ryoma Sato, Makoto Yamada, and Hisashi Kashima. 2021. Random features strengthen graph neural networks. In *Proceedings of the 2021 SIAM international conference on data mining (SDM)*. SIAM, 333–341.

[35] Hamed Shirzad, Ameya Velingker, Balaji Venkatachalam, Danica J Sutherland, and Ali Kemal Sinop. 2023. Exphormer: Sparse transformers for graphs. In *International Conference on Machine Learning*. PMLR, 31613–31632.

[36] Jan Tönshoff, Martin Ritzert, Eran Rosenbluth, and Martin Grohe. 2023. Where did the gap go? reassessing the long-range graph benchmark. *arXiv preprint arXiv:2309.00367* (2023).

[37] Jan Tönshoff, Martin Ritzert, Hinrikus Wolf, and Martin Grohe. 2021. Walking out of the weisfeiler leman hierarchy: Graph learning beyond message passing. *arXiv preprint arXiv:2102.08786* (2021).

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[39] Petar Veličković. 2023. Everything is connected: Graph neural networks. *Current Opinion in Structural Biology* 79 (2023), 102538.

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

[41] Haorui Wang, Haoteng Yin, Muhan Zhang, and Pan Li. 2022. Equivariant and stable positional encoding for more powerful graph neural networks. *arXiv preprint arXiv:2203.00199* (2022).

[42] Qitian Wu, Chenxiao Yang, Wentao Zhao, Yixuan He, David Wipf, and Junchi Yan. 2023. Difformer: Scalable (graph) transformers induced by energy constrained diffusion. *arXiv preprint arXiv:2301.09474* (2023).

[43] Qitian Wu, Wentao Zhao, Zenan Li, David P Wipf, and Junchi Yan. 2022. Nodeformer: A scalable graph structure learning transformer for node classification. *Advances in Neural Information Processing Systems* 35 (2022), 27387–27401.

[44] Zhanghao Wu, Paras Jain, Matthew Wright, Azalia Mirhoseini, Joseph E Gonzalez, and Ion Stoica. 2021. Representing long-range context for graph neural networks with global attention. *Advances in Neural Information Processing Systems* 34 (2021), 13266–13279.

[45] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems* 32, 1 (2020), 4–24.

[46] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How Powerful are Graph Neural Networks?. In *International Conference on Learning Representations*.

[47] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. 2021. Do transformers really perform badly for graph representation? *Advances in neural information processing systems* 34 (2021), 28877–28888.

[48] Muhan Zhang and Pan Li. 2021. Nested graph neural networks. *Advances in Neural Information Processing Systems* 34 (2021), 15734–15747.

[49] Lingxiao Zhao, Wei Jin, Leman Akoglu, and Neil Shah. 2021. From stars to subgraphs: Uplifting any GNN with local structure awareness. *arXiv preprint arXiv:2110.03753* (2021).

Table 6: Results for BenchmarkingGNNs including ZINC, MNIST, CIFAR10, PATTERN, and CLUSTER.

Sparse Graph	**MNIST** ↑	**CIFAR10** ↑	**PATTERN** ↑	**CLUSTER** ↑	**ZINC** ↓
GatedGCN + noPE	97.800±0.138	69.303±0.318	85.397±0.040	61.695±0.261	0.2398±0.0094
GatedGCN + ESLapPE	97.870±0.090	69.438±0.297	85.422±0.161	61.953±0.082	0.2409±0.0131
GatedGCN + LapPE	97.575±0.025	69.285±0.205	86.700±0.000	65.130±0.405	0.1718±0.0024
GatedGCN + RWSE	97.840±0.171	69.038±0.152	86.833±0.030	65.675±0.296	0.1016±0.0030
GatedGCN + SignNet	97.553±0.167	68.570±0.240	86.763±0.027	75.293±0.047	0.1060±0.0021
GatedGCN + PPR	97.797±0.045	69.224±0.546	86.522±0.093	74.175±0.122	0.3678±0.0198
GatedGCN + GCKN	97.745±0.069	69.408±0.222	86.758±0.049	62.478±0.156	0.1446±0.0048
GatedGCN + WLPE	97.693±0.235	69.418±0.165	84.980±0.160	62.738±0.291	0.1779±0.0059
GatedGCN + RWDIFF	97.823±0.119	69.528±0.494	86.760±0.043	65.653±0.470	0.1346±0.0074
GatedGCN + RRWP	97.908±0.076	69.572±0.787	85.465±0.148	61.728±0.174	0.2451±0.0131
GINE + noPE	97.712±0.120	65.554±0.225	85.482±0.272	48.783±0.060	0.1210±0.0107
GINE + ESLapPE	97.596±0.071	66.140±0.310	85.546±0.114	48.708±0.061	0.1209±0.0066
GINE + LapPE	97.555±0.045	65.325±0.195	85.835±0.195	48.685±0.035	0.1144±0.0028
GINE + RWSE	97.686±0.073	65.238±0.283	86.688±0.084	50.642±0.694	0.0795±0.0034
GINE + SignNet	97.692±0.165	64.538±0.314	86.538±0.044	59.660±0.630	0.0993±0.0069
GINE + PPR	97.650±0.088	65.082±0.434	85.658±0.048	47.440±2.290	0.3019±0.0122
GINE + GCKN	97.708±0.105	65.976±0.308	85.844±0.157	48.780±0.149	0.1169±0.0029
GINE + WLPE	97.716±0.118	66.132±0.225	85.676±0.084	48.997±0.068	0.1205±0.0062
GINE + RWDIFF	97.750±0.097	65.632±0.553	85.764±0.209	49.148±0.168	0.0750±0.0058
GINE + RRWP	96.742±0.277	62.790±1.501	86.526±0.036	48.736±0.108	0.0857±0.0009
Exphormer + noPE	98.414±0.047	74.962±0.631	85.676±0.049	77.500±0.151	0.1825±0.0209
Exphormer + ESLapPE	98.354±0.108	74.880±0.322	86.734±0.024	78.218±0.267	0.2023±0.0140
Exphormer + LapPE	98.270±0.070	75.205±0.095	86.565±0.075	77.175±0.165	0.1503±0.0117
Exphormer + RWSE	98.254±0.084	74.434±0.205	86.820±0.040	77.690±0.147	0.0933±0.0050
Exphormer + SignNet	98.136±0.094	73.842±0.317	86.752±0.088	78.280±0.211	0.0924±0.0072
Exphormer + PPR	98.076±0.126	74.076±0.104	86.712±0.047	78.098±0.211	0.2414±0.0123
Exphormer + GCKN	98.402±0.067	74.926±0.288	86.730±0.040	77.470±0.067	0.1690±0.0056
Exphormer + WLPE	98.398±0.162	74.794±0.358	85.454±0.033	77.402$_{\pm 0.120}$	0.1465±0.0095
Exphormer + RWDIFF	98.416$_{\pm 0.055}$	74.886±0.810	86.792±0.023	77.550$_{\pm 0.057}$	0.1360±0.0082
Exphormer + RRWP	98.418±0.179	74.504±0.369	85.652±0.001	77.434±0.056	0.1914±0.0153
GraphGPS + noPE	98.136±0.085	72.310±0.198	84.182±0.276	77.590±0.158	0.1610±0.0045
GraphGPS + ESLapPE	98.180±0.117	72.122±0.511	86.700±0.055	77.800±0.107	0.1795±0.0110
GraphGPS + LapPE	98.065±0.075	72.310±0.530	86.550±0.150	77.355±0.115	0.1086±0.0062
GraphGPS + RWSE	98.116±0.102	72.034±0.756	86.866±0.010	77.550±0.195	0.0744±0.0060
GraphGPS + SignNet	98.012±0.091	72.152±0.323	86.734±0.069	78.308±0.111	0.0945±0.0019
GraphGPS + PPR	98.010±0.097	71.842±0.325	86.124±0.214	76.828±0.250	0.1349±0.0054
GraphGPS + GCKN	98.180±0.117	72.194±0.515	86.786±0.043	77.514±0.182	0.1460±0.0078
GraphGPS + WLPE	98.038±0.134	72.258±0.661	84.916±0.195	76.866±0.171	0.1204±0.0055
GraphGPS + RWDIFF	98.026±0.101	71.800±0.363	86.820±0.063	77.478±0.150	0.0924±0.0212
GraphGPS + RRWP	98.146±0.105	72.084±0.466	84.436±0.224	77.420±0.080	0.1690±0.0084
SparseGRIT + noPE	97.940±0.071	72.778±0.627	85.948±0.148	77.274±0.170	0.1255±0.0062
SparseGRIT + ESLapPE	97.970±0.110	72.494±0.501	86.018±0.319	77.238±0.066	0.1280±0.0077
SparseGRIT + LapPE	97.915±0.065	72.640±0.040	86.555±0.025	76.100±0.085	0.1070±0.0017
SparseGRIT + RWSE	98.122±0.054	72.330±0.600	86.914±0.031	77.148±0.174	0.0676±0.0060
SparseGRIT + SignNet	97.946±0.122	71.003±0.301	86.794±0.055	78.882±0.146	0.0821±0.0043
SparseGRIT + PPR	98.020±0.194	71.926±0.833	86.650±0.033	78.732±0.202	0.2536±0.0193
SparseGRIT + GCKN	97.958±0.127	72.598±0.535	86.650±0.033	76.746±0.187	0.1233±0.0071
SparseGRIT + WLPE	97.946±0.125	72.096±0.835	85.712±0.027	77.170±0.143	0.1262±0.0059
SparseGRIT + RWDIFF	98.022±0.083	72.366±0.388	86.938±0.045	77.214±0.065	0.0690±0.0039
SparseGRIT + RRWP	98.088±0.048	74.954±0.256	87.168±0.041	79.872±0.079	0.0651±0.0027
GRIT + noPE	98.108±0.190	74.402±0.135	87.126±0.033	78.616±0.178	0.1237±0.0057
GRIT + ESLapPE	98.010±0.141	74.558±0.682	87.140±0.064	78.588±0.111	0.1241±0.0031
GRIT + LapPE	97.875±0.001	73.325±0.505	86.985±0.015	77.960±0.310	0.1039±0.0035
GRIT + RWSE	98.068±0.182	73.652±0.623	87.116±0.046	78.880±0.057	0.0671±0.0037
GRIT + SignNet	97.766±0.220	72.812±0.482	87.085±0.064	79.770±0.150	0.0945±0.0098
GRIT + PPR	97.986±0.082	73.568±0.451	86.780±0.001	78.958±0.175	0.1390±0.0076
GRIT + GCKN	98.084±0.139	73.946±0.910	87.194±0.044	78.542±0.149	0.1306±0.0141
GRIT + WLPE	98.022±0.173	74.206±0.684	86.863±0.033	78.500±0.091	0.1218±0.0035
GRIT + RWDIFF	98.024±0.148	73.956±0.202	87.152±0.045	78.778±0.090	0.0671±0.0060
GRIT + RRWP	98.124±0.141	75.662±0.410	87.217±0.034	79.812±0.109	0.0590±0.0010

Table 7: Results for LRGB datasets which include Peptides_func, Peptides_struct, PCQM_Contact, PascalVOC-SuperPixels and COCO-SuperPixels. The hyperparameters for Peptides_func and Peptides_struct follow the original GraphGPS settings.

Sparse Graph	Peptides-func	Peptides-struct	PCQM-Contact	PascalVOC-SP	COCO-SP
GatedGCN + noPE	0.6523±0.0074	0.2470±0.0005	0.4730±0.0003	0.3923±0.0020	0.2619±0.0045
GatedGCN + LapPE	0.6581±0.0068	0.2472±0.0003	0.4764±0.0004	0.3920±0.0033	0.2671±0.0006
GatedGCN + ESLapPE	0.6484±0.0037	0.2490±0.0020	0.4736±0.0006	0.3930±0.0041	0.2628±0.0004
GatedGCN + RWSE	0.6696±0.0022	0.2485±0.0022	0.4749±0.0005	0.3882±0.0041	0.2657±0.0007
GatedGCN + SignNet	0.5327±0.0137	0.2688±0.0016	0.4672±0.0001	0.3814±0.0005	-
GatedGCN + GCKN	0.6544±0.0040	0.2483±0.0009	0.4687±0.0002	0.3933±0.0044	-
GatedGCN + WLPE	0.6562±0.0053	0.2473±0.0012	0.4671±0.0003	0.3805±0.0018	-
GatedGCN + RWDIFF	0.6527±0.0053	0.2474±0.0003	0.4740±0.0003	0.3919±0.0019	0.2674±0.0031
GatedGCN + RRWP	0.6516±0.0072	0.2514±0.0001	-	-	-
GraphGPS + noPE	0.6514±0.0123	0.4243±0.0305	0.4649±0.0025	0.4517±0.0112	0.3799±0.0056
GraphGPS + LapPE	0.6620±0.0073	0.2497±0.0024	0.4696±0.0017	0.4505±0.0062	0.3859±0.0016
GraphGPS + ESLapPE	0.6516±0.0062	0.2568$_{\pm 0.0013}$	0.4639±0.0031	0.4538±0.0083	0.3866±0.0017
GraphGPS + RWSE	0.6510±0.0071	0.2549$_{\pm 0.0033}$	0.4685±0.0009	0.4531±0.0073	0.3891±0.0033
GraphGPS + SignNet	0.5719±0.0055	0.2657$_{\pm 0.0021}$	0.4624±0.0020	0.4291±0.0056	-
GraphGPS + GCKN	0.6502±0.0101	0.2519$_{\pm 0.0005}$	0.4609±0.0007	0.4515±0.0053	-
GraphGPS + WLPE	0.5851±0.0441	0.5203$_{\pm 0.0504}$	0.4622±0.0012	0.4501±0.0057	-
GraphGPS + RWDIFF	0.6519±0.0077	0.4769$_{\pm 0.0360}$	0.4669±0.0006	0.4488±0.0097	0.3873±0.0024
GraphGPS + RRWP	0.6505±0.0058	0.3734±0.0157	-	-	-
Exphormer + noPE	0.6200±0.0052	0.2584±0.0019	0.4661±0.0021	0.4149±0.0047	0.3445±0.0052
Exphormer + LapPE	0.6424±0.0063	0.2496±0.0013	0.4737±0.0024	0.4242±0.0044	0.3471±0.0028
Exphormer + ESLapPE	0.6281±0.0085	0.2513±0.0022	0.4676±0.0018	0.4141±0.0054	0.3485±0.0011
Exphormer + RWSE	0.6240±0.0069	0.2579±0.0010	0.4642±0.0039	0.4218±0.0063	0.3485±0.0011
Exphormer + SignNet	0.5458±0.0097	0.2667±0.0037	0.4615±0.0066	0.3966±0.0020	-
Exphormer + GCKN	0.6422±0.0080	0.2514±0.0012	0.4604±0.0038	0.4196±0.0049	-
Exphormer + WLPE	0.6216±0.0069	0.2558±0.0011	0.2051±0.0080	0.4104±0.0071	-
Exphormer + RWDIFF	0.6275±0.0031	0.2556±0.0021	0.4642±0.0032	0.4165±0.0059	0.3417±0.0006
Exphormer + RRWP	0.6208±0.0074	0.2586±0.0014	-	-	-
SparseGRIT + noPE	0.4885±0.0036	0.2550±0.0006	0.4527±0.0006	0.3471±0.0030	0.1976±0.0038
SparseGRIT + LapPE	0.5884±0.0059	0.2487±0.0014	0.4585±0.0011	0.3514±0.0026	0.1974±0.0008
SparseGRIT + ESLapPE	0.5161±0.0069	0.2537±0.0005	0.4532±0.0005	0.3462±0.0035	0.1958±0.0001
SparseGRIT + RWSE	0.5570±0.0079	0.2537±0.0012	0.4553±0.0014	0.3460±0.0071	0.1969±0.0010
SparseGRIT + SignNet	0.5115±0.0064	0.2640±0.0018	0.4573±0.0003	0.3419±0.0074	-
SparseGRIT + GCKN	0.5871±0.0042	0.2492±0.0010	0.4500±0.0004	0.3519±0.0040	-
SparseGRIT + WLPE	0.4808±0.0016	0.2547±0.0005	0.4489±0.0012	0.3439±0.0027	-
SparseGRIT + RWDIFF	0.5521±0.0072	0.2550±0.0008	0.4551±0.0005	0.3447±0.0046	0.1965±0.0011
SparseGRIT + RRWP	0.6702±0.0080	0.2504±0.0025	-	-	-
GRIT + noPE	0.4861±0.0053	0.2489±0.0008	0.4525 ± 0.0001	0.3556 ± 0.0019	0.2105 ± 0.0004
GRIT + LapPE	0.5834±0.0105	0.2474±0.0005	0.4580 ± 0.0020	0.3551 ± 0.0032	0.2112 ± 0.0005
GRIT + ESLapPE	0.4831±0.0023	0.2584±0.0002	0.4486 ± 0.0014	0.3485 ± 0.0028	0.2100 ± 0.0008
GRIT + RWSE	0.5432±0.0034	0.2612±0.0008	0.4524 ± 0.0001	0.3461 ± 0.0058	0.2114 ± 0.0009
GRIT + SignNet	0.5307±0.0085	0.2600±0.0018	0.4608 ± 0.0007	0.3385 ± 0.0045	-
GRIT + GCKN	0.5868±0.0051	0.2477±0.0006	0.4521 ± 0.0002	0.3516 ± 0.0003	-
GRIT + WLPE	0.4798±0.0012	0.2578±0.0011	0.4515 ± 0.0004	0.3441 ± 0.0011	-
GRIT + RWDIFF	0.5801±0.0036	0.2639±0.0010	0.4563 ± 0.0003	0.3521 ± 0.0079	0.2128 ± 0.0008
GRIT + RRWP	0.6865±0.0050	0.2454±0.0010	-	-	-