GENERALIZED ROTARY POSITION ENCODING FOR GRAPHS VIA ARBITRARY PAIRWISE DISTANCES

Anonymous authors
Paper under double-blind review

ABSTRACT

Rotary Position Embedding (RoPE) has become a key building block of modern large language models. In this work, we investigate how RoPE can be generalized from sequential offsets to arbitrary pairwise relative distances. Because the original formulation of RoPE is intrinsically 1D, and its usual multi-axis extensions apply this 1D scheme to different slices of the hidden space, it cannot naturally capture relations beyond a small number of independent axes. We introduce δ-RoPE, a parameter-free generalized position encoding (PE) that directly conditions attention on an arbitrary pairwise function, making it naturally applicable to non-Euclidean structures. We explore this generalization on graphs, where the pairwise relation can naturally be instantiated as the shortest-path distance between nodes. This yields a unified formulation of position encoding in which sequential RoPE is recovered as a special case. We first evaluate δ-RoPE in a standalone transformer setting, where it consistently improves over standard attention. We then evaluate it within strong graph-learning settings, including the widely used General Powerful Scalable Graph Transformer (GraphGPS) framework, using standard graph-learning benchmarks. δ-RoPE yields substantial gains on established long-range graph benchmarks, where modeling non-local dependencies is particularly important. Code is available at https://anonymous.4open.science/r/delta_rope-FDFF/.

1 INTRODUCTION

Progress in machine learning does not always transfer well across domains. Core optimization machinery, such as optimizers and learning rate schedulers, has proven remarkably domain-agnostic (Kingma & Ba, 2015), transferring with little modification from computer vision (Dosovitskiy et al., 2021) to natural language processing (NLP) (Devlin et al., 2019) to graph learning (Rampášek et al., 2022). Other components, however, do not transfer so cleanly: their design is tied to the specific structure of each domain, and adapting them across domains with fundamentally different data structures is less obvious.

This is for instance the case for position encoding (PE). Given that attention is permutation equivariant, it relies on position encoding to inject structural information into its internal representations, yet the graph and NLP communities have developed their position encoding methods largely independently of one another.

NLP mostly relies on a single one, Rotary Position Embedding (RoPE) (Su et al., 2021), now used by many recent Large Language Models (LLMs) (DeepSeek-AI, 2024; Qwen Team, 2025). Graph learning, by contrast, offers a whole variety: encoding graph properties, such as the Laplacian, into node or edge features (Dwivedi & Bresson, 2020; Dwivedi et al., 2022a), learning an attention bias from pairwise quantities, such as the shortest-path distance (Ying et al., 2021), or combining attention with a message-passing branch (Rampášek et al., 2022).

In this work, we investigate whether RoPE is a suitable candidate for graphs. Given its wide adoption in LLMs, a variant compatible with graph structure is an appealing option. However, the original RoPE is designed for sequences and needs to be adapted for graph structures. Lifting this restriction has a further benefit: since RoPE already covers other domains such as text and images, the same mechanism would then serve several modalities, a reason for this work to use a non-parametric approach and to follow the original RoPE formulation as closely as possible.

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

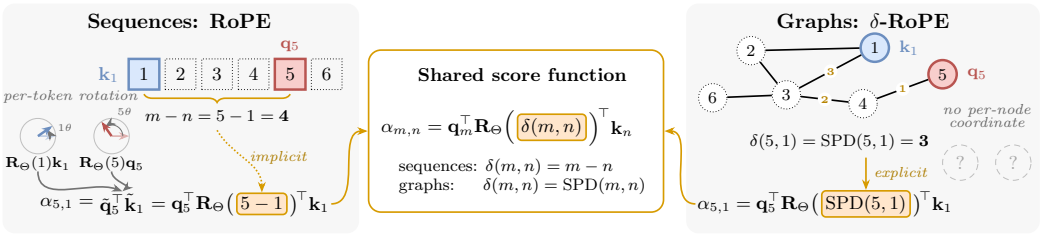


Figure 1: RoPE and δ-RoPE share the same score function (center), in which a pairwise relation $\delta(m, n)$ indexes the rotation between query and key. On sequences (left), RoPE rotates each token by its own position, and the offset $\delta(m, n) = m - n$ arises implicitly when the rotated query and key meet. On graphs (right), nodes have no canonical coordinate to rotate by; δ-RoPE instead provides the relation explicitly, as the shortest-path distance between the two nodes.

Concretely, our contributions are the following:

- **We introduce δ-RoPE**, a parameter-free generalization of RoPE that conditions attention on an arbitrary pairwise function $\delta(m, n)$ rather than on a sequential offset (Eq. 5, Figure 1). Standard RoPE is recovered exactly as a special case, while graph structure can be incorporated by instantiating δ as the shortest-path distance between nodes. This provides a step toward a unified position-encoding framework across data structures.
- **We show how to deploy δ-RoPE**, both as a standalone attention mechanism and within an established graph-transformer framework. For standalone evaluation, we introduce an edge-to-node graph augmentation that allows edge features to enter the attention in Section 3.2. We then integrate δ-RoPE into GraphGPS by replacing its global attention branch in Section 3.3.
- **We evaluate δ-RoPE** on established graph-learning benchmarks. In the standalone setting, it consistently improves over standard attention. Within GraphGPS, it yields substantial gains on tasks requiring long-range interactions and is competitive with state-of-the-art graph models. We further report its computational overhead and study how the choice of RoPE's frequency-scale hyperparameter, the base b, affects performance.

2 Related Work

Position encoding has followed largely independent paths in NLP and graph learning, shaped by the different structures (sequences versus graphs) each community targets. We review both threads below, focusing on attention-based architectures while briefly addressing complementary approaches, such as MPNN, in graph learning. Throughout, we assume a node-level tokenization of graphs, i.e., each node is treated as one token fed to the attention layer, mirroring how each word or sub-word is treated as one token in NLP.

Attention and the need for positional encoding. Attention operates on a set of tokens and is permutation equivariant: permuting the input tokens permutes the output identically. Attention-based architectures are therefore systematically paired with a position encoding (PE) mechanism to inject such structural information, originally in the form of a fixed, absolute sinusoidal encoding added to each token embedding (Vaswani et al., 2017).

Position encoding in NLP. More recently, methods have converged towards a relative approach, most notably Rotary Position Embedding (RoPE) (Su et al., 2021), now widely adopted by recent models (DeepSeek-AI, 2024; Qwen Team, 2025). Rather than adding a position vector to each token's input embedding, RoPE injects position inside the attention operation itself, by rotating queries and keys by an angle proportional to their own position; we detail this construction, and our generalization of it, in Section 3.1. This original formulation is intrinsically tied to sequences, encoding position solely as a scalar offset along a single axis, or along a small number of independent

axes (Heo et al., 2024), leaving open how a rotary-style PE could be defined on more complex structures such as graphs.

Attention and graphs. PEs for graph attention likewise splits into absolute and relative formulations. Absolute encodings augment every node with a fixed vector derived from graph structure, most commonly the k smallest non-trivial Laplacian eigenvectors (Dwivedi & Bresson, 2020), later refined to fix their sign and basis ambiguity (Kreuzer et al., 2021; Lim et al., 2023), or a random-walk-based encoding (Dwivedi et al., 2022a). Relative encodings instead insert the relation between two nodes directly into the attention computation: Graphormer (Ying et al., 2021) biases the attention scores with a graph-theoretic quantity, most commonly learnable parameters over the shortest-path distance, while Exphormer (Shirzad et al., 2023) restricts attention to a sparse pairwise mask, remaining competitive with full attention.

Closest to our work, Wave-Induced Rotary Encodings (WIRE) (Reid et al., 2025) extends RoPE itself to graphs with a learnable rotation: queries and keys are rotated by angles obtained from a learned projection of the graph Laplacian eigenvectors, recovering standard RoPE on regular grids while, on general graphs, yielding attention scores that depend on the effective resistance between nodes (Ellens et al., 2011). WIRE thus remains parametric and is tied to the Laplacian spectrum for its structural signal; part of our motivation is to encode pairwise graph distance directly within the rotation, without requiring an eigendecomposition nor extra parameters. Being non-parametric, and recovering standard RoPE exactly when the distance is the sequential offset, our formulation keeps a single mechanism shared with sequences: it can replace RoPE in an existing, possibly pre-trained, Transformer without introducing new weights, and techniques developed for RoPE in LLMs remain directly applicable as discussed in Section 5.

Message passing and hybrid architectures. Message-passing neural networks (MPNNs) (Gilmer et al., 2017), such as GatedGCN (Bresson & Laurent, 2018) or GIN (Xu et al., 2019), update each node at every layer by aggregating messages from its neighbors. Graph structure is thus built in, but information travels one hop per layer and is compressed into fixed-size node vectors, which limits long-range interactions (Alon & Yahav, 2021). Attention has the opposite profile: it connects all nodes in a single layer but relies on PE to see the graph. Given these complementary strengths, the two are often combined into hybrid architectures. GraphGPS (Rampášek et al., 2022) exemplifies this recipe: within each GPS layer, a local MPNN layer and a global attention layer are applied and combined, optionally alongside absolute PE (Dwivedi & Bresson, 2020; Dwivedi et al., 2022a).

3 METHOD

Rotary Position Embedding (RoPE) (Su et al., 2021) injects position inside the attention operation itself, by rotating queries and keys by an angle proportional to their own position. Our goal is to generalize this construction from sequences to graphs. We first recall the original RoPE formulation and its two equivalent readings, as per-token rotations and as a function of the scalar offset between two positions. Generalizing the second reading lets us replace that offset with an arbitrary pairwise function and define δ-RoPE, an attention mechanism conditioned on shortest-path distance. We then describe how to deploy δ-RoPE as a standalone transformer-only model that also incorporates edge features, and finally how we integrate it into GraphGPS (Rampášek et al., 2022), a widely used hybrid framework that combines a local message-passing branch with global attention.

3.1 FROM SEQUENTIAL OFFSET TO ARBITRARY PAIRWISE DISTANCE

Given token representations $\boldsymbol{h}_n^\ell$ at layer ℓ, with queries $\boldsymbol{q}_n = \boldsymbol{W}_Q \boldsymbol{h}_n^\ell$, keys $\boldsymbol{k}_n = \boldsymbol{W}_K \boldsymbol{h}_n^\ell$ and values $\boldsymbol{v}_n = \boldsymbol{W}_V \boldsymbol{h}_n^\ell$ in $\mathbb{R}^d$, where $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V$ are learnable projection matrices, and assuming no position encoding, attention computes the output $\boldsymbol{h}_m^{\ell+1}$ of token m as

$$\boldsymbol{h}_m^{\ell+1} = \sum_n \mathrm{softmax}_n(\alpha_{m,n}/\sqrt{d})\, \boldsymbol{v}_n, \qquad \alpha_{m,n} = \langle \boldsymbol{q}_m, \boldsymbol{k}_n \rangle, \tag{1}$$

where $\alpha_{m,n}$ is the attention score between tokens m and n.

RoPE on sequences. Consider first the sequential case, with tokens at positions m and n. Write $P := d/2$, and group the d coordinates of $\boldsymbol{q}_m$ into pairs $\boldsymbol{q}_m^{(l)} := (q_{m,l}, q_{m,l+P}) \in \mathbb{R}^2$, $l \in \{1, \dots, P\}$, defining $\boldsymbol{k}_n^{(l)}$ analogously. RoPE (Su et al., 2021) rotates each such pair by an angle proportional to the token's own position[1], with frequencies $\theta_l = b^{-2(l-1)/d}$ and base $b = 10000$ in the original formulation:

$$\tilde{\boldsymbol{q}}_m^{(l)} = \boldsymbol{R}(\theta_l m)\, \boldsymbol{q}_m^{(l)}, \qquad \tilde{\boldsymbol{k}}_n^{(l)} = \boldsymbol{R}(\theta_l n)\, \boldsymbol{k}_n^{(l)}, \qquad \boldsymbol{R}(\phi) := \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix}. \tag{2}$$

Writing $\boldsymbol{R}_\Theta(\phi)$ for the rotation of $\mathbb{R}^d$ acting on each pair l as $\boldsymbol{R}(\theta_l \phi)$, the RoPE score is the inner product of the rotated query and key which, since $\boldsymbol{R}(a)^\top \boldsymbol{R}(b) = \boldsymbol{R}(a-b)^\top$, reduces to

$$\alpha_{m,n} = \langle \boldsymbol{R}_\Theta(m)\, \boldsymbol{q}_m,\, \boldsymbol{R}_\Theta(n)\, \boldsymbol{k}_n \rangle = \boldsymbol{q}_m^\top \boldsymbol{R}_\Theta(m)^\top \boldsymbol{R}_\Theta(n)\, \boldsymbol{k}_n = \boldsymbol{q}_m^\top \boldsymbol{R}_\Theta(m-n)^\top \boldsymbol{k}_n. \tag{3}$$

This relative structure is intentional: Su et al. (2021) construct the per-token rotations precisely so that the offset $m - n$ is recovered once query and key meet in the inner product. Expanding Eq. 3 pair by pair gives:

$$\langle \tilde{\boldsymbol{q}}_m^{(l)}, \tilde{\boldsymbol{k}}_n^{(l)} \rangle = \langle \boldsymbol{q}_m^{(l)}, \boldsymbol{k}_n^{(l)} \rangle \cos(\theta_l (m-n)) + \det(\boldsymbol{q}_m^{(l)}, \boldsymbol{k}_n^{(l)}) \sin(\theta_l (m-n)), \tag{4}$$

where $\det(\boldsymbol{q}_m^{(l)}, \boldsymbol{k}_n^{(l)}) := q_{m,l} k_{n,l+P} - q_{m,l+P} k_{n,l}$ is the determinant of the two pairs.

Two readings of RoPE. RoPE thus admits two equivalent readings: per-token (Eq. 2), where each token is rotated according to its own coordinate, and pairwise (Eqs. 3–4), where the score is a fixed function of the query, the key and a single scalar relation between the two tokens. On sequences they coincide, and RoPE uses the per-token form because it avoids any pairwise computation. Beyond sequences, they generalize differently. Existing extensions generalize the per-token reading, with one coordinate per axis (Heo et al., 2024) or coordinates derived from the graph Laplacian spectrum (Reid et al., 2025), which requires node coordinates that graphs do not provide canonically. The pairwise reading only requires a relation between two nodes, which graphs provide natively, e.g., the shortest-path distance.

δ-RoPE. We therefore ask whether RoPE's relative score function, which Su et al. (2021) use only with the sequential offset, remains a suitable position encoding for a graph relation. Keeping the pairing of coordinates and the frequencies unchanged, we replace $m - n$ in Eq. 4 with a general pairwise relation $\delta(m, n)$:

$$\alpha_{m,n} := \sum_{l=1}^{P} \langle \boldsymbol{q}_m^{(l)}, \boldsymbol{k}_n^{(l)} \rangle \cos(\theta_l \delta(m,n)) + \det(\boldsymbol{q}_m^{(l)}, \boldsymbol{k}_n^{(l)}) \sin(\theta_l \delta(m,n)). \tag{5}$$

We call this formulation δ-RoPE. In matrix form (Figure 1, center), $\alpha_{m,n} = \boldsymbol{q}_m^\top \boldsymbol{R}_\Theta(\delta(m,n))^\top \boldsymbol{k}_n$: RoPE's relative rotation, indexed directly by a pairwise function $\delta : \mathbb{N}^2 \to \mathbb{R}$ rather than arising from two per-token rotations. Setting $\delta(m,n) = m - n$ recovers standard RoPE attention exactly (Su et al., 2021); in this sequential case, the sign of $m - n$ distinguishes tokens to the left of the query from those to its right.

In this work, we instantiate δ-RoPE on undirected graphs by taking $\delta(m,n) = \mathrm{SPD}(m,n)$, the shortest-path distance between nodes m and n. This provides a graph-relative notion of position within the same attention formulation. We first evaluate this construction in a standalone transformer-only model in Section 3.2 and then use it as the global attention mechanism within GraphGPS (Rampášek et al., 2022) in Section 3.3.

3.2 δ-RoPE IN A STANDALONE TRANSFORMER

We first evaluate δ-RoPE in a transformer-only setting in order to isolate its effect from other graph-specific components, in particular message passing, which is widely used in state-of-the-art graph neural networks. Under the node-level tokenization considered throughout this work, each graph node becomes one token and its node features can be encoded directly. Graphs, however, may also

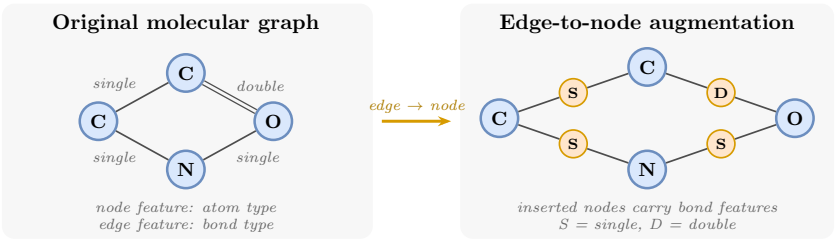


Figure 2: Edge-to-node augmentation used in the standalone transformer setting, illustrated with molecular node and edge features. Original node tokens are shown in blue, while edge-derived tokens are shown in orange. Atoms are represented by node tokens, and each bond is converted into an inserted token connected to its two endpoint atoms. The graph is illustrative rather than a specific benchmark sample.

carry informative features on their edges. In the transformer-only baseline considered here, node tokens alone provide no direct channel through which such edge features can enter the model.

For this standalone setting, we therefore use a simple edge-to-node augmentation so that edge features can enter the transformer directly (Figure 2). Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we construct an augmented graph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ by replacing every edge $e = (i, j) \in \mathcal{E}$ with a new node v_e carrying the corresponding edge features. The original edge is removed, and v_e is connected to its two endpoints:

$$\mathcal{V}' = \mathcal{V} \cup \{v_e : e \in \mathcal{E}\}, \qquad \mathcal{E}' = \{(i, v_e), (v_e, j) : e = (i, j) \in \mathcal{E}\}.$$

Original nodes and edge-derived nodes use separate encoders. Attention and the shortest-path distances are computed over the augmented graph $\mathcal{G}'$.

This construction is deliberately simple. Its purpose is to expose node and edge information to the same transformer-only architecture while keeping the comparison between standard attention and δ-RoPE controlled; we do not claim it as an optimal tokenization of graphs.

3.3 δ-RoPE in a hybrid graph transformer

The standalone setting of Section 3.2 evaluates δ-RoPE in a generic transformer architecture. We therefore also evaluate δ-RoPE within GraphGPS (Rampášek et al., 2022), a well-established hybrid graph transformer framework that combines graph-specific message passing with global attention. This provides a complementary evaluation setting in which δ-RoPE is assessed inside a graph-native architecture rather than in a transformer adapted to graph inputs. In GraphGPS, the local MPNN branch aggregates information from neighboring nodes and edges, while the global attention branch allows interactions between arbitrary pairs of nodes. We integrate δ-RoPE by modifying the query-key compatibility score in this global attention branch, while leaving the remainder of the GraphGPS architecture unchanged.

A GraphGPS model first encodes node and edge features, optionally together with an absolute position encoding, and then applies a sequence of L GPS layers. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, let $\boldsymbol{h}_m^\ell$ denote the representation of node $m \in \mathcal{V}$ entering layer ℓ, let $\boldsymbol{e}_{m,n}^\ell$ denote the representation of edge $(m, n) \in \mathcal{E}$, and let $\boldsymbol{h}^\ell$ and $\boldsymbol{e}^\ell$ denote the corresponding collections. Each GPS layer applies a local MPNN branch and a global attention branch in parallel, then combines their outputs[2]:

$$\begin{aligned} \boldsymbol{h}_M^{\ell+1}, \boldsymbol{e}^{\ell+1} &= \mathrm{MPNN}^\ell(\boldsymbol{h}^\ell, \boldsymbol{e}^\ell, \mathcal{E}), && \text{(local message passing)} \\ \boldsymbol{h}_{T,m}^{\ell+1} &= \textstyle\sum_{n \in \mathcal{V}} \mathrm{softmax}_n(\alpha_{m,n}/\sqrt{d})\, \boldsymbol{v}_n, && \text{(global attention)} \\ \boldsymbol{h}_m^{\ell+1} &= \mathrm{MLP}^\ell(\boldsymbol{h}_{M,m}^{\ell+1} + \boldsymbol{h}_{T,m}^{\ell+1}), && \text{(branch fusion)} \end{aligned} \tag{6}$$

where queries, keys and values are computed from $\boldsymbol{h}^\ell$ as in Eq. 1, with layer-specific projections, and standard attention uses the attention score $\alpha_{m,n} = \langle \boldsymbol{q}_m, \boldsymbol{k}_n \rangle$.

[1] $\boldsymbol{R}(\phi)$ follows the counterclockwise rotation convention.

[2] Multiple attention heads, residual connections, dropout, and normalization are omitted for clarity.

We replace this attention score $\alpha_{m,n}$ with δ-RoPE (Eq. 5), using $\delta(m,n) = \mathrm{SPD}(m,n)$, the shortest-path distance between nodes m and n in the original graph $\mathcal{G}$. Since edge features enter GraphGPS through the MPNN branch, the edge-to-node augmentation of Section 3.2 is not needed.

4 EXPERIMENTS

We evaluate δ-RoPE in two steps. We first isolate the gain of δ-RoPE over standard attention, using the standalone transformer-only model of Section 3.2. We then integrate δ-RoPE into GraphGPS to compare it against state-of-the-art architectures. We report results on the Long Range Graph Benchmarks (LRGB) (Dwivedi et al., 2022b) as well as on canonical graph-learning benchmarks (Dwivedi et al., 2023). We then study the effect of the RoPE base b on performance (Section 4.3), and the computational overhead of our attention in a practical setting in Section 4.4. Full model and training configurations for every dataset are given in Appendix A.1 (Tables 4–7).

4.1 δ-ROPE AS A STANDALONE TRANSFORMER-ONLY MODEL

We first evaluate our method as a standalone module (Section 3.2), in order to isolate the gain of δ-RoPE over standard attention. For that, we train two transformer-only models on the same edge-to-node transformed graphs: one with δ-RoPE, and one with plain attention (no distance conditioning). Rather than tuning either model, we rely on the benchmark of Grötschla et al. (2026), which evaluates GNNs and graph Transformers with a range of position encodings under a common training protocol: for each dataset, we take the configuration of an attention-only architecture from this benchmark, which is described in Appendix A.1. Both models share this configuration (number of layers, hidden width, number of heads and absolute position encoding, i.e., LapPE, or none on PascalVOC-SP and COCO-SP) and differ only in their attention block. The base of δ-RoPE is fixed a priori to $b = 20$ on every dataset, without tuning. Table 1 reports results on eight datasets, as the mean and standard deviation over 4 seeds.

Table 1: Results of δ-RoPE used standalone (no MPNN branch) against a transformer-only model using the same edge-to-node transformation but with plain attention (no distance conditioning), isolating the effect of δ-RoPE. Parentheses indicate the absolute position encoding used by each run. All metrics except MAE are reported $\times 100$.

Dataset	Model	Score
CIFAR10 (Acc.) ↑	Attention	62.58 ± 2.78 (LapPE)
	Ours	$\mathbf{68.21 \pm 1.27}$ (LapPE)
ZINC (MAE) ↓	Attention	0.468 ± 0.030 (LapPE)
	Ours	$\mathbf{0.119 \pm 0.012}$ (LapPE)
MNIST (Acc.) ↑	Attention	86.62 ± 3.95 (LapPE)
	Ours	$\mathbf{95.50 \pm 1.01}$ (LapPE)
COCO-SP (F1) ↑	Attention	24.56 ± 0.24 (noPE)
	Ours	$\mathbf{25.91 \pm 1.53}$ (noPE)
PCQM-Contact (MRR filt.) ↑	Attention	44.79 ± 1.36 (LapPE)
	Ours	$\mathbf{45.48 \pm 0.68}$ (LapPE)
PascalVOC-SP (F1) ↑	Attention	26.44 ± 0.77 (noPE)
	Ours	$\mathbf{30.38 \pm 1.18}$ (noPE)
Pept-func (AP) ↑	Attention	60.76 ± 1.14 (LapPE)
	Ours	$\mathbf{69.19 \pm 0.85}$ (LapPE)
Pept-struct (MAE) ↓	Attention	0.2543 ± 0.0013 (LapPE)
	Ours	$\mathbf{0.2470 \pm 0.0024}$ (LapPE)

As shown in Table 1, δ-RoPE improves over the plain Transformer baseline on every dataset, although within seed variance on COCO-SP and PCQM-Contact, indicating that the distance conditioning, rather than the edge-to-node transformation alone, accounts for most of the gains. We note that the edge-to-node transformation is deliberately kept simple, to compare attention mechanisms.

4.2 δ-RoPE INSIDE GRAPHGPS

Having isolated the gain of δ-RoPE over standard attention, we next assess how it compares to state-of-the-art models. Prior work has shown that hybrid architectures (combining an MPNN branch with attention) typically outperform pure attention models in many cases (Rampášek et al., 2022); we therefore test δ-RoPE as a replacement for GraphGPS's attention branch. For every dataset, we take the GraphGPS configuration from the benchmark of Grötschla et al. (2026) and only replace its Transformer attention block with our distance-conditioned attention, keeping the MPNN branch, encoders and absolute position encoding unchanged. As in the standalone setting, the base is fixed to $b = 20$ on every dataset. We focus the discussion on the LRGB benchmarks, as we observed δ-RoPE to be particularly effective at handling long-range interactions; results on the canonical benchmarks are provided in Appendix A.2. Table 2 reports results on the long-range benchmarks of LRGB (COCO-SP, PCQM-Contact, PascalVOC-SP, Peptides-func, and Peptides-struct), as the mean and standard deviation over 4 random seeds, with baseline numbers taken from Grötschla et al. (2026) and, for WIRE, from Reid et al. (2025). The full list of baselines, with references, is given in Appendix A.2.

Table 2: Results of GraphGPS with δ-RoPE as its attention branch, on the long-range benchmarks of LRGB, against baselines from Grötschla et al. (2026) and WIRE (Reid et al., 2025). Parentheses indicate the position/structural encoding used by each baseline; "–" marks a benchmark not reported for that model. All metrics except MAE are reported $\times 100$. Per column, green marks the best result, orange the second-best, and blue the third-best; WIRE, obtained under a different setup, is excluded from the ranking.

Model	**COCO-SP** (F1) ↑	**PCQM-Contact** (MRR filt.) ↑	**PascalVOC-SP** (F1) ↑	**Pept-func** (AP) ↑	**Pept-struct** (MAE) ↓
GCN	13.38 ± 0.07	45.26 ± 0.06	0.78 ± 0.31	68.60 ± 0.50	0.2460 ± 0.0007
GINE	21.25 ± 0.09	46.17 ± 0.05	27.18 ± 0.54	66.21 ± 0.67	0.2473 ± 0.0017
GatedGCN	29.22 ± 0.18	46.70 ± 0.04	38.80 ± 0.40	67.65 ± 0.47	0.2477 ± 0.0009
CRaWl	–	–	45.88 ± 0.79	70.74 ± 0.32	0.2506 ± 0.0022
S^2GCN	–	–	–	73.11 ± 0.66	0.2447 ± 0.0032
DRew	–	34.42 ± 0.06	33.14 ± 0.24	71.50 ± 0.44	0.2536 ± 0.0015
Graph ViT	–	–	–	68.76 ± 0.59	0.2455 ± 0.0027
GatedGCN-VN	32.44 ± 0.25	–	44.77 ± 1.37	68.23 ± 0.69	0.2475 ± 0.0018
Exphormer	34.85 ± 0.11 (ESLapPE)	47.37 ± 0.24 (LapPE)	42.42 ± 0.44 (LapPE)	64.24 ± 0.63 (LapPE)	0.2496 ± 0.0013 (LapPE)
GraphGPS	38.91 ± 0.33 (RWSE)	46.96 ± 0.17 (LapPE)	45.38 ± 0.83 (ESLapPE)	66.20 ± 0.73 (LapPE)	0.2497 ± 0.0024 (LapPE)
SparseGRIT	19.76 ± 0.38 (noPE)	45.85 ± 0.11 (LapPE)	35.19 ± 0.40 (GCKN)	67.02 ± 0.80 (RRWP)	0.2487 ± 0.0014 (LapPE)
GRIT	21.28 ± 0.08 (RWDIFF)	46.08 ± 0.07 (SignNet)	35.56 ± 0.19 (noPE)	68.65 ± 0.50 (RRWP)	0.2454 ± 0.0010 (RRWP)
WIRE[3]	–	–	37.6	64.9	0.2566
Ours	39.14 ± 0.40 (RWSE)	47.64 ± 0.19 (LapPE)	45.17 ± 0.75 (ESLapPE)	70.60 ± 0.48 (LapPE)	0.2450 ± 0.0009 (LapPE)

The results in Table 2 show that δ-RoPE improves over GraphGPS beyond seed variance on Peptides-func ($+4.40$ AP), Peptides-struct (-0.0047 MAE) and PCQM-Contact ($+0.68$ MRR), obtaining the best result among all reported models on PCQM-Contact. On COCO-SP and PascalVOC-SP it remains on par with GraphGPS, although it reaches the best mean score of all reported models on COCO-SP. The two Peptides datasets, where the gains are largest, are also those with the largest diameters and the widest spread of shortest-path distances (Table 9, Figure 4). On the canonical benchmarks, δ-RoPE performs on par with GraphGPS, with all differences within seed variance, as shown in Appendix A.2.

4.3 EFFECT OF THE ROPE BASE

The RoPE frequencies $\theta_l = b^{-2(l-1)/d}$ are controlled by a single base b, set to 10000 in the original, sequence-based RoPE. Since graph diameters are typically far smaller than the sequence lengths RoPE was designed for, we study how b affects performance on CIFAR10 and MNIST (Figure 3).

As shown in Figure 3, on both benchmarks, performance peaks well below the standard sequential value of 10000, but the best base differs between datasets ($b = 20$ on CIFAR10, $b = 80$ on MNIST). This study is carried out a posteriori and is not used for model selection: all other experiments keep the value $b = 20$ fixed beforehand.

[3]WIRE results are taken from Reid et al. (2025) and obtained with a Performer-based (linear-attention) backbone under a different training configuration; the comparison is therefore not strictly fair and is included for reference only.

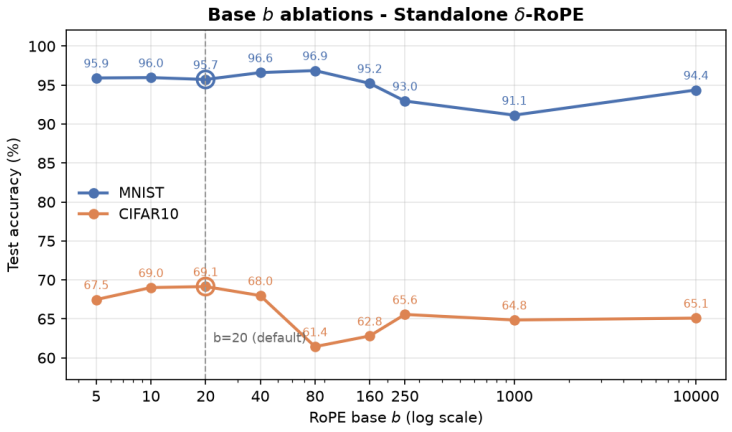


Figure 3: Effect of the RoPE base b on test accuracy of the standalone δ-RoPE model (single seed) for CIFAR10 and MNIST.

4.4 Computational cost

We implement δ-RoPE's distance-conditioned attention as a custom Triton kernel. When δ-RoPE is employed, the shortest-path distance (SPD) matrix is computed either online, at every training step, or offline, prior to training. Table 3 reports the mean training time per epoch (in seconds) for GraphGPS with standard attention and with δ-RoPE; timings for the standalone model are given in Appendix A.4. We omit datasets whose graphs are too large for their SPD matrices to be cached without substantial further engineering. Model configurations are otherwise unchanged from Appendix A.1 (Tables 4–5).

Table 3: Mean training time per epoch (in seconds) for GraphGPS with standard attention and with δ-RoPE, with the shortest-path distance computed either online or offline.

Dataset	**GraphGPS**	**+δ-RoPE online**	**+δ-RoPE offline**
Peptides-func	5.7	46.4	25.3
Peptides-struct	4.2	40.7	24.0
MNIST	90.6	132.1	123.1
ZINC	12.4	26.7	26.1
PCQM-Contact	97.8	99.3	108.1

As shown in Table 3, while adding δ-RoPE increases training time compared to the vanilla GraphGPS, computing SPD offline reduces overhead especially on graphs with large diameters. Note that batch size and model size differ across datasets, so training times should not be compared directly across rows; only within-row comparisons across attention variants are meaningful.

5 Conclusion and Future Work

We introduced a generalization of Rotary Position Embedding that replaces its scalar sequential offset $m - n$ with an arbitrary pairwise function $\delta(m, n)$, recovering standard RoPE and a graph shortest-path variant as two instances of the same attention formulation. Used as a standalone transformer-only model, via an edge-to-node transformation, δ-RoPE improves over standard attention on every benchmark we consider, while keeping parameters constant. Integrated into GraphGPS, it yields consistent gains on the long-range benchmarks, where it is competitive with state-of-the-art models.

Taken together, these results show δ-RoPE to be a practical position encoding for graph attention, and a new step toward unifying position encoding across modalities under a single formulation.

Several directions could extend this work. First, since δ-RoPE only requires a pairwise relation between two tokens, it is not limited to graphs. To handle data with several axes, standard RoPE splits the hidden dimensions into one group per axis, which restricts it to a small number of independent axes; δ-RoPE could instead encode such structures with a single distance function using all hidden dimensions. Second, since δ-RoPE has no learnable parameters and keeps the RoPE formulation, it could replace RoPE in any Transformer that already uses it, including pre-trained ones. Prior work has shown that the position encoding of a pre-trained model can be changed at the cost of only a short finetuning phase (Gelberg et al., 2025), suggesting that pre-trained models such as LLMs could be adapted to graph-structured inputs in this way. Finally, techniques developed around RoPE in LLMs could be transferred to graphs. For instance, RoPE scaling methods such as YaRN (Peng et al., 2024) adjust the rotation frequencies so that a model handles sequences longer than those seen during training, and could similarly help a model generalize to graphs of heterogeneous diameters. We leave these directions to future work.

AI USE STATEMENT

We used generative AI tools (LLMs) to assist with implementation: the authors wrote the initial δ-RoPE attention code, and AI tools iterated on it and wrote parts of the remaining code, including integration into GraphGPS and GRIT, compatibility patches, Triton kernels, experiment scripts and cluster job submission/monitoring. All AI-assisted code was reviewed by the authors and verified by cross-checking results against baseline implementations and known experimental results. We did not use generative AI tools to develop theoretical models, formulate mathematical claims, assist with proofs, or design our methodology or experiments. Additionally, we used generative AI tools for writing and editing prose and paper structure, formatting, and reference management; all such output was reviewed and edited by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Our implementation builds on two public codebases: GraphGPS, in which we replace only the attention module, and the benchmarking repository of Grötschla et al. (2026), source of several baseline numbers and of some of our configuration files. We additionally implement the edge-to-node graph augmentation (Figure 2) underlying our standalone variant. Appendix A.1 lists, for every dataset, the full architecture and training configuration of each baseline/δ-RoPE pair (Tables 4–7). All datasets are public benchmarks (ZINC, MNIST, CIFAR10, and LRGB). An anonymized version of our code is available at https://anonymous.4open.science/r/delta_rope-FDFF/; the complete repository, including graphGPS integration, will be released upon acceptance.

ETHICS STATEMENT

This work is purely methodological, studying position encodings for graph attention. All experiments use standard public benchmarks. We do not see any ethical concerns with this work.

REFERENCES

Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. In *International Conference on Learning Representations*, 2021.

Xavier Bresson and Thomas Laurent. Residual gated graph convnets. In *International Conference on Learning Representations (Workshop Track)*, 2018.

Yun Young Choi, Sun Woo Park, Minho Lee, and Youngho Woo. Topology-informed graph transformer. In *ICML 2024 Workshop on Geometry-grounded Representation Learning and Generative Modeling (GRaM)*, volume 251 of *Proceedings of Machine Learning Research*, 2024.

DeepSeek-AI. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of*

the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *arXiv preprint arXiv:2012.09699*, 2020.

Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Graph neural networks with learnable structural and positional representations. In *International Conference on Learning Representations*, 2022a.

Vijay Prakash Dwivedi, Ladislav Rampášek, Mikhail Galkin, Ali Parviz, Guy Wolf, Anh Tuan Luu, and Dominique Beaini. Long range graph benchmark. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, volume 35, 2022b.

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

Wendy Ellens, Floske M. Spieksma, Piet Van Mieghem, Almerima Jamakovic, and Robert E. Kooij. Effective graph resistance. *Linear Algebra and its Applications*, 435(10):2491–2506, 2011.

Simon Geisler, Arthur Kosmala, Daniel Herbst, and Stephan Günnemann. Spatio-spectral graph neural networks. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Yoav Gelberg, Koshi Eguchi, Takuya Akiba, and Edoardo Cetin. Extending the context of pretrained llms by dropping their positional embeddings. *arXiv preprint arXiv:2512.12167*, 2025.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, pp. 1263–1272, 2017.

Florian Grötschla, Jiaqing Xie, and Roger Wattenhofer. Benchmarking positional encodings for GNNs and graph transformers. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2026.

Benjamin Gutteridge, Xiaowen Dong, Michael M. Bronstein, and Francesco Di Giovanni. DRew: Dynamically rewired message passing with delay. In *International Conference on Machine Learning*, 2023.

Xiaoxin He, Bryan Hooi, Thomas Laurent, Adam Perold, Yann LeCun, and Xavier Bresson. A generalization of ViT/MLP-Mixer to graphs. In *International Conference on Machine Learning*, 2023.

Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary position embedding for vision transformer. In *European Conference on Computer Vision*, 2024.

Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations*, 2020.

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

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Derek Lim, Joshua Robinson, Lingxiao Zhao, Tess Smidt, Suvrit Sra, Haggai Maron, and Stefanie Jegelka. Sign and basis invariant networks for spectral graph representation learning. In *International Conference on Learning Representations*, 2023.

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

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models. In *International Conference on Learning Representations*, 2024.

Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Ladislav Rampášek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Isaac Reid, Arijit Sehanobish, Cederik Höfs, Bruno Mlodozeniec, Leonhard Vulpius, Federico Barbero, Adrian Weller, Krzysztof Choromanski, Richard E. Turner, and Petar Veličković. Rotary position encodings for graphs. *arXiv preprint arXiv:2509.22259*, 2025.

Eran Rosenbluth, Jan Tönshoff, Martin Ritzert, Berke Kisin, and Martin Grohe. Distinguished in uniform: Self-attention vs. virtual nodes. In *International Conference on Learning Representations*, 2024.

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

Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021.

Jan Tönshoff, Martin Ritzert, Hinrikus Wolf, and Martin Grohe. Walking out of the Weisfeiler Leman hierarchy: Graph learning beyond message passing. *Transactions on Machine Learning Research*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform bad for graph representation? In *Advances in Neural Information Processing Systems*, volume 34, 2021.

A Appendix

A.1 Experimental setup

We build on the public GraphGPS codebase and change only the attention module. Each experiment pairs a *baseline* with a δ-*RoPE* variant that shares its entire configuration. Given that our experiments rely on tables from Grötschla et al. (2026), we also use configuration files from that repository.

- **GraphGPS / GraphGPS+δ-RoPE.** The δ-RoPE variant keeps the MPNN branch, node/edge encoders and absolute position encoding of the GraphGPS baseline, and only swaps the global Transformer attention for our distance-conditioned δ-RoPE attention. Width, depth, heads, dropout, optimizer and schedule are identical.
- **Attention / Standalone δ-RoPE (edge-to-node).** The standalone model has no MPNN: every edge is lifted to a node and the graph is processed by a distance-conditioned transformer alone. It adopts the GRIT (Ma et al., 2023) configurations of Grötschla et al. (2026). The plain-attention baseline of Table 1 shares the exact configuration of the standalone model, only replacing δ-RoPE with standard attention. Since δ-RoPE already acts as a relative position encoding, we do not stack GRIT's relative RRWP encoding on top of it, and instead use LapPE, or no absolute PE on PascalVOC-SP and COCO-SP. The only other deviations from the GRIT configurations are a lower learning rate on ZINC (5×10^{-4}), as training diverged with GRIT's learning rate, and a batch size of 16 on COCO-SP due to memory constraints; a two-seed check with gradient accumulation to an effective batch size of 32 gave comparable results.

All scores are the mean $\pm$ standard deviation over seeds $0, 1, 2, 3$, taken as the test metric at the best-validation epoch. Pairwise distances are the (unweighted) shortest-path distances. When the per-head dimension is odd (e.g., 13 on MNIST and CIFAR10), the last query/key channel of each head has no rotation partner and does not contribute to the attention scores; values still use all channels. The RoPE base is $b = 20$ throughout, except in the ablation of Section 4.3. Tables 4–7 list the exact configurations; cells are given as `baseline / δ-RoPE`. Parameter counts are rounded to the nearest thousand; the small difference between GraphGPS and GraphGPS+δ-RoPE comes from our attention implementation omitting the bias terms of the query, key, value and output projections.

Table 4: GraphGPS vs. GraphGPS+δ-RoPE architecture (baseline / δ-RoPE). Part 1.

	ZINC	**MNIST**	**CIFAR10**	**PCQM-Contact**
Layers	10 / 10	3 / 3	3 / 3	6 / 6
Hidden dim	64 / 64	52 / 52	52 / 52	76 / 76
Heads	4 / 4	4 / 4	4 / 4	4 / 4
Dropout	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0
Attn dropout	0.5 / 0.5	0.5 / 0.5	0.5 / 0.5	0.5 / 0.5
Abs. PE	RWSE / RWSE	ESLapPE / LapPE	noPE / LapPE	LapPE / LapPE
Optimizer	AdamW / AdamW	AdamW / AdamW	AdamW / AdamW	AdamW / AdamW
Base LR	1e-3 / 1e-3	1e-3 / 1e-3	1e-3 / 1e-3	1e-3 / 1e-3
Weight decay	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0
Warmup epochs	50 / 50	5 / 5	5 / 5	5 / 5
Max epochs	2000 / 2000	100 / 100	100 / 100	150 / 150
Scheduler	cosine with warmup			
Batch size	32 / 32	16 / 16	16 / 16	500 / 500
# Params	424k / 421k	115k / 115k	113k / 112k	478k / 477k

Table 5: GraphGPS vs. GraphGPS+δ-RoPE architecture (baseline / δ-RoPE). Part 2.

	Peptides-func	**Peptides-struct**	**PascalVOC-SP**	**COCO-SP**
Layers	6 / 6	4 / 4	8 / 8	8 / 8
Hidden dim	76 / 76	96 / 96	68 / 68	68 / 68
Heads	4 / 4	4 / 4	4 / 4	4 / 4
Dropout	0.1 / 0.1	0.0 / 0.0	0.1 / 0.1	0.1 / 0.1
Attn dropout	0.5 / 0.5	0.5 / 0.5	0.5 / 0.5	0.5 / 0.5
Abs. PE	LapPE / LapPE	LapPE / LapPE	ESLapPE / ESLapPE	RWSE / RWSE
Optimizer	AdamW / AdamW	AdamW / AdamW	AdamW / AdamW	AdamW / AdamW
Base LR	1e-3 / 1e-3	3e-4 / 3e-4	1e-3 / 1e-3	1e-3 / 1e-3
Weight decay	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0
Warmup epochs	5 / 5	10 / 10	10 / 10	10 / 10
Max epochs	250 / 250	200 / 200	200 / 200	200 / 200
Scheduler	cosine with warmup			
Batch size	200 / 200	128 / 128	50 / 50	50 / 50
# Params	479k / 477k	505k / 503k	503k / 500k	500k / 498k

Table 6: Plain attention vs. δ-RoPE, both on the edge-to-node lifted graph (attention / δ-RoPE). Part 1.

	ZINC	**MNIST**	**CIFAR10**	**PCQM-Contact**
Layers	10 / 10	3 / 3	3 / 3	10 / 10
Hidden dim	64 / 64	52 / 52	52 / 52	64 / 64
Heads	8 / 8	4 / 4	4 / 4	8 / 8
Dropout	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0
Attn dropout	0.2 / 0.2	0.5 / 0.5	0.5 / 0.5	0.2 / 0.2
Pos. enc.	LapPE / LapPE	LapPE / LapPE	LapPE / LapPE	LapPE / LapPE
Optimizer	AdamW / AdamW	AdamW / AdamW	AdamW / AdamW	AdamW / AdamW
Base LR	1e-3 / 5e-4	1e-3 / 1e-3	1e-3 / 1e-3	1e-3 / 1e-3
Weight decay	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0
Warmup epochs	50 / 50	5 / 5	5 / 5	10 / 10
Max epochs	1500 / 1500	150 / 150	150 / 150	100 / 100
Scheduler	cosine with warmup			
Batch size	32 / 32	16 / 16	16 / 16	500 / 500
# Params	347k / 344k	71k / 70k	71k / 70k	354k / 351k

Table 7: Plain attention vs. δ-RoPE, both on the edge-to-node lifted graph (attention / δ-RoPE). Part 2.

	Peptides-func	**Peptides-struct**	**PascalVOC-SP**	**COCO-SP**
Layers	4 / 4	4 / 4	4 / 4	4 / 4
Hidden dim	96 / 96	96 / 96	64 / 64	64 / 64
Heads	4 / 4	8 / 8	4 / 4	8 / 8
Dropout	0.0 / 0.0	0.05 / 0.05	0.0 / 0.0	0.0 / 0.0
Attn dropout	0.5 / 0.5	0.2 / 0.2	0.2 / 0.2	0.2 / 0.2
Pos. enc.	LapPE / LapPE	LapPE / LapPE	noPE / noPE	noPE / noPE
Optimizer	AdamW / AdamW	AdamW / AdamW	AdamW / AdamW	AdamW / AdamW
Base LR	3e-4 / 3e-4	3e-4 / 3e-4	1e-3 / 1e-3	5e-4 / 5e-4
Weight decay	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0
Warmup epochs	10 / 10	10 / 10	10 / 10	10 / 10
Max epochs	200 / 200	200 / 200	250 / 250	200 / 200
Scheduler	cosine with warmup			
Batch size	16 / 16	32 / 32	16 / 16	16 / 16
# Params	317k / 315k	326k / 324k	141k / 140k	149k / 148k

A.2 GraphGPS results on canonical benchmarks

Baselines. Tables 2 and 8 compare against MPNNs (GCN (Kipf & Welling, 2017), GINE (Hu et al., 2020), GatedGCN (Bresson & Laurent, 2018)), architectures that go beyond local message passing (CRaWl (Tönshoff et al., 2023), S^2GCN (Geisler et al., 2024), DRew (Gutteridge et al., 2023), Graph ViT (He et al., 2023), GatedGCN-VN (Rosenbluth et al., 2024)), and graph transformers (Exphormer (Shirzad et al., 2023), GRIT (Ma et al., 2023), GraphGPS (Rampášek et al., 2022), WIRE (Reid et al., 2025), and, on the canonical benchmarks only, EGT (Hussain et al., 2022) and TIGT (Choi et al., 2024)), together with the SparseGRIT and fully connected GatedGCN (FC) variants of Grötschla et al. (2026). Baseline numbers are taken from Grötschla et al. (2026) and, for WIRE, from Reid et al. (2025).

Results. Table 8 reports the results of GraphGPS with δ-RoPE on the canonical benchmarks CIFAR10, MNIST and ZINC, against the baselines listed above. As in the main text, our runs keep the absolute position encoding of the GraphGPS baseline, with one exception: on MNIST and CIFAR10, the best reported GraphGPS baselines use ESLapPE and noPE respectively, whereas our runs use LapPE. δ-RoPE performs on par with GraphGPS on all three benchmarks, with all differences within seed variance.

Table 8: Results of GraphGPS with δ-RoPE as its attention branch, on canonical benchmarks, against baselines from Grötschla et al. (2026) and WIRE (Reid et al., 2025). Parentheses indicate the position/structural encoding used by each baseline. Per column, green marks the best result, orange the second-best, and blue the third-best; WIRE, obtained under a different setup, is excluded from the ranking.

Model	**CIFAR10** (Acc.) ↑	**ZINC** (MAE) ↓	**MNIST** (Acc.) ↑
EGT	68.70 ± 0.41	0.108 ± 0.009	98.17 ± 0.09
TIGT	73.96 ± 0.36	0.057 ± 0.002	98.23 ± 0.13
GINE	66.14 ± 0.31 (ESLapPE)	0.075 ± 0.006 (RWDIFF)	97.75 ± 0.10 (RWDIFF)
GatedGCN	69.57 ± 0.79 (RRWP)	0.102 ± 0.003 (RWSE)	97.91 ± 0.08 (RRWP)
SparseGRIT	74.95 ± 0.26 (RRWP)	0.065 ± 0.003 (RRWP)	98.12 ± 0.05 (RWSE)
Exphormer	75.21 ± 0.10 (LapPE)	0.092 ± 0.007 (SignNet)	98.42 ± 0.18 (RRWP)
GRIT	75.66 ± 0.41 (RRWP)	0.059 ± 0.001 (RRWP)	98.12 ± 0.14 (RRWP)
GatedGCN (FC)	71.08 ± 0.60 (RRWP)	0.114 ± 0.003 (RWSE)	98.20 ± 0.15 (GCKN)
GraphGPS	72.31 ± 0.20 (noPE)	0.074 ± 0.006 (RWSE)	98.18 ± 0.12 (ESLapPE)
WIRE	73.48	–	98.46
Ours	72.22 ± 0.67 (LapPE)	0.070 ± 0.001 (RWSE)	98.24 ± 0.05 (LapPE)

A.3 Dataset properties

Table 9 reports graph statistics for the eight benchmarks and Figure 4 the full distribution of shortest-path distances within each. For the shortest-path distribution, we sample 1000 graphs from each dataset and compute the shortest-path distance between all node pairs in each graph.

Table 9: Raw graph statistics per dataset. Size and degree are computed over all graphs; the shortest-path block (diameter, mean SPD) is over a fixed 1000-graph random sample (seed 0). Values are mean±std across graphs. Peptides-func and Peptides-struct share the same graphs.

Dataset	#graphs	#nodes	#edges	degree	diameter	SPD mean
PCQM-Contact	529434	30.1±6.4	30.5±6.6	2.0±1.2	9.8±1.8	4.6±0.6
COCO-SP	123286	476.9±15.8	1346.8±46.8	5.6±1.2	27.2±2.3	10.6±0.6
PascalVOC-SP	11355	479.4±16.5	1355.2±48.3	5.7±1.2	27.7±2.2	10.7±0.5
Peptides-func/struct	15535	150.9±84.2	153.7±86.1	2.0±0.8	56.0±28.9	20.5±9.9
ZINC	12000	23.2±4.5	24.9±5.3	2.2±0.7	12.4±2.5	5.1±0.8
MNIST	70000	70.6±6.8	282.3±27.4	8.0±0.0	6.0±0.5	3.0±0.2
CIFAR10	60000	117.6±4.3	470.5±17.1	8.0±0.0	8.5±0.5	4.0±0.1

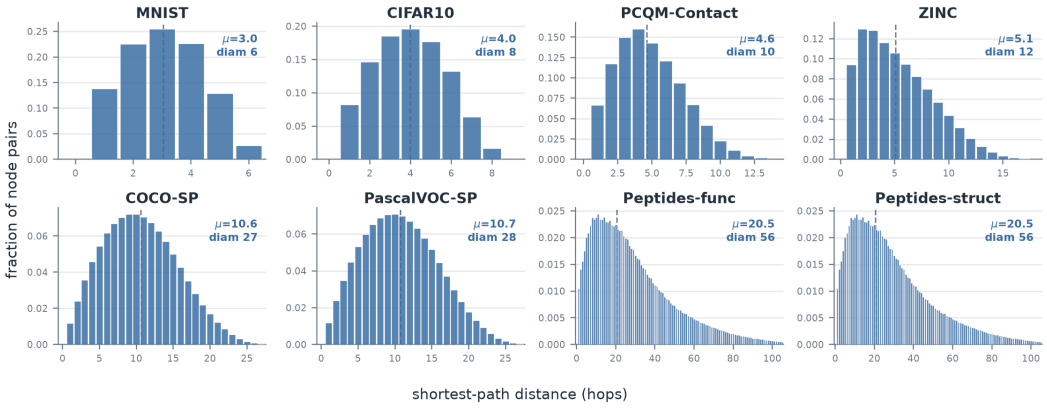


Figure 4: Distribution of shortest-path distances over node pairs, per benchmark, computed on a fixed 1000-graph random sample. Counts are normalized so that each panel sums to one. The spread of each distribution, rather than the number of nodes, determines how many distinct rotation angles δ-RoPE assigns.

A.4 COMPUTATIONAL COST OF THE STANDALONE MODEL

Table 10 reports the mean training time per epoch (in seconds) for the standalone model (Section 4.1), with plain attention and with δ-RoPE. As a runtime reference only, we also time GRIT (Ma et al., 2023), an attention-only graph transformer, from a single-seed run using the same configuration as the standalone model, including the number of layers; GRIT is not part of our accuracy comparisons. The standalone configurations are those of Tables 6–7. With the shortest-path distance computed offline, the training time of the standalone δ-RoPE model is comparable to that of GRIT. As in Table 3, only within-row comparisons are meaningful.

Table 10: Mean training time per epoch (in seconds) for the standalone model: our edge-to-node transformation (Section 4.1) with plain attention (plain TF) and with δ-RoPE (shortest-path distance computed online or offline), compared against GRIT (single seed, same configuration as the standalone model).

Dataset	**GRIT**	**standalone online**	**standalone offline**	**standalone plain TF**
Peptides-func	94.2	198.0	66.0	16.5
Peptides-struct	97.5	219.0	79.3	16.5
MNIST	138.9	138.4	154.5	65.5
ZINC	20.1	27.3	24.7	10.9
PCQM-Contact	386.1	341.6	437.7	361.1