RETHINKING POSITIONAL ENCODING FOR NEURAL VEHICLE ROUTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Transformer-based models have become the dominant paradigm for neural combinatorial optimization (NCO) of vehicle routing problems (VRPs), yet the role of positional encoding (PE) in these architectures remains largely unexplored. Unlike natural language, where tokens are uniformly spaced on a line, routing solutions exhibit several properties that render standard NLP positional encodings inadequate. In this work, we formalize three such structural properties that a routing-aware PE should respect, namely anisometric node distances, cyclic and direction-aware topology, and hierarchical depot-anchored global multi-route structure, combining them with a unifying design principle of geometric grounding. Guided by these criteria, we analyze and compare PE methods spanning NLP, graph-transformer, and routing-specific families, and propose a hierarchical anisometric PE that combines a distance-indexed, circularly consistent in-route encoding with a depot-anchored angular cross-route encoding. Controlled experiments on four VRP variants and three improvement-based backbones show that geometry-grounded PE outperforms index-based and graph-transformer alternatives; at large scale the margins are small but, in most settings, resolved by paired per-instance tests, and they persist under the distribution shifts we test. Our source code will be made publicly available.

1 INTRODUCTION

Neural combinatorial optimization (NCO) has emerged as a promising paradigm for vehicle routing problems (VRPs), with learning-based solvers steadily narrowing the gap to classical operations-research methods (Kool et al., 2019; Kwon et al., 2020; Ma et al., 2021; Drakulic et al., 2023; Luo et al., 2023; Hottung et al., 2025b). A defining challenge in this setting is that routing inputs are unordered sets of nodes, while routing outputs are highly structured sequences, including tours, multi-route solutions, and ordering chains. Bridging this gap requires the model to reason explicitly about *position*: which node comes next, how far it is from its neighbors in the current solution, and which route it belongs to. Transformer-based architectures, equipped with *positional encoding* (PE) as the standard mechanism for injecting such positional information, have become the dominant backbone (Ma et al., 2021; 2022; Hottung et al., 2025b).

Despite this rapid progress, the role of PE has received remarkably little scrutiny. Most construction-based models omit PE entirely (Kool et al., 2019; Kwon et al., 2020) or directly apply standard natural language processing (NLP) encodings without adaptation. Improvement-based methods have begun to incorporate positional signals, such as a Gray-code cyclic PE in DACT (Ma et al., 2021) and an index-based PE in NDS (Hottung et al., 2025b), but these designs are ad hoc, neither justified by the structure of routing solutions nor compared against alternatives. In contrast, PE design in NLP and vision has received sustained attention and has been shown to meaningfully affect model performance (Vaswani et al., 2017; Shaw et al., 2018; Su et al., 2024).

This neglect matters because, while NLP positions are uniformly spaced on a line, routing positions exhibit structure at three distinct levels. At the metric level, routes are *anisometric*: consecutive nodes are separated by varying physical distance rather than equal index gaps, so a sinusoidal encoding that maps index pairs to identical embedding gaps ignores the geometry of the underlying tour. At the topological level, routes are *cyclic*, since each closed tour departs from and returns to the depot, making the first and last positions topologically identical, a property no linear PE can express. Routes are also *direction-aware* in a problem-dependent way: symmetric VRPs admit reversal symmetry,

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.

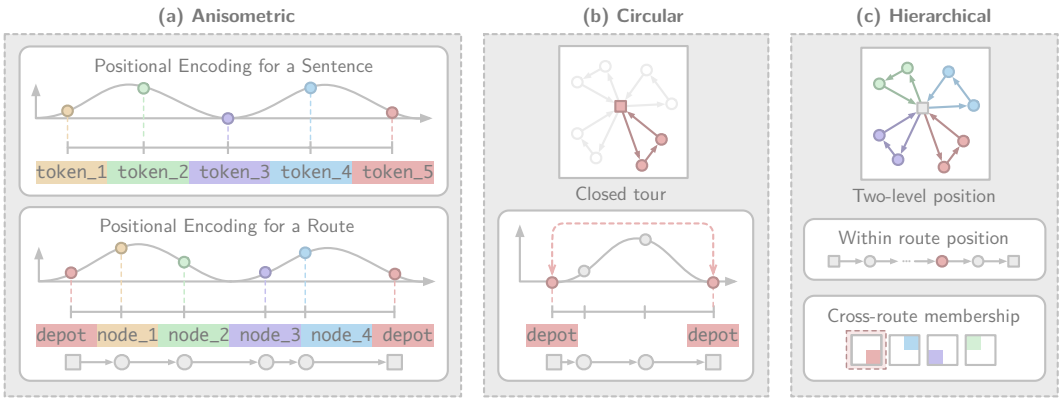


Figure 1: Properties of positional encoding for routes.

whereas PDTSP and VRPTW carry precedence and time-window semantics that break it. At the global level, VRP solutions are *hierarchical* and *depot-anchored*: a node's identity depends on its intra-route position, its route, and how that route relates to others through the depot. Since routes admit no canonical ordering, such a PE must be permutation-invariant across routes yet sequential within them.

Building on these observations, this paper presents a systematic study of positional encoding for neural vehicle routing, framed as a question of input design and inductive bias: how the geometry and structure of the current solution should enter a routing Transformer. Our contributions are as follows.

- We identify the structural properties a positional encoding should respect for vehicle routing, and use them to organize existing encodings into a unified taxonomy.
- We propose a simple hierarchical positional encoding tailored to the geometry of routing solutions, and isolate the roles of its distance signal, Fourier mapping, and symmetry choice.
- We show through probing and controlled comparison that existing encodings miss these properties and that this gap propagates to solution quality.
- We show gains over index-based and graph-transformer encodings in state-of-the-art improvement-based solvers, across the problem variants, backbones, and distribution shifts we test.

2 Background and Related Work

Vehicle routing problems. A VRP is defined on a graph $G = (V, E)$, where $V = \{0, 1, \ldots, N\}$ contains a depot x_0 and N customers, and each edge carries a travel cost (typically Euclidean distance). A feasible solution partitions customers into routes, each originating from and returning to the depot, subject to variant-specific constraints such as vehicle capacity (CVRP), time windows (VRPTW), or optional service with prizes (PCVRP). The objective is to minimize total travel cost (plus the prizes of unvisited customers in PCVRP). We defer full mathematical formulations to Section A.

Transformer-based NCO. Neural VRP solvers fall into two families with fundamentally different relationships to PE. *Construction methods* (Kool et al., 2019; Kwon et al., 2020; Drakulic et al., 2023; Luo et al., 2023) decode one node at a time from a permutation-equivariant encoder over an unordered customer set, so PE is not needed at the encoding stage; re-encoding variants such as LEHD and BQ re-encode an ordered partial solution, whose positional structure is unexplored. *Improvement methods* (Ma et al., 2021; 2022; Hottung et al., 2025b) operate on a complete current solution and refine it through learned k-opt moves, neighborhood search, or large-neighborhood deconstruction, so every forward pass consumes an ordered sequence of routed nodes and PE is an explicit, unavoidable design choice. Our analysis therefore concentrates on improvement methods and on re-encoding-style construction methods, where PE materially shapes the representation.

Positional encoding in routing models. Among improvement models, DACT (Ma et al., 2021) introduces a cyclic PE (CPE) based on Gray codes to represent the circular ordering of nodes within a tour. NDS (Hottung et al., 2025b) uses a simple index-based PE to encode the sequential position of each node in its route. CycleFormer (Yook et al., 2024) proposes a circular PE specifically designed for TSP. A complementary line injects geometric or structural context through features and architecture rather than through PE; e.g., DGL (Xiao et al., 2025) aggregates dynamic local and global information to generalize across scales (see Section U for a comparison). However, none of these works systematically compare their PE choice against alternatives, nor do they articulate what properties a PE should have for routing. This paper fills that gap.

3 Routing-Aware Positional Encoding Taxonomy

This section examines the structural properties of routing solutions and existing PE designs. Section 3.1 formalizes the geometric properties of routes that we will reason about and states desiderata that follow from them. Section 3.2 surveys existing PE methods, and discusses at the conceptual level the extent to which each satisfies the desiderata.

For clarity of exposition, the analysis throughout this section assumes *symmetric* distances and *closed-tour* VRP variants, i.e., every route vehicle returns to the depot. The treatment of asymmetric and open-tour variants is presented in Section 6.

3.1 Preliminaries and Desiderata

A solution of a VRP variant is a collection of routes $\mathcal{S} = \{r^{(1)}, \dots, r^{(K)}\}$, where each route $r^{(k)} = (v_1^{(k)}, \dots, v_{L_k}^{(k)})$ visits a subset of customers, with $v_1^{(k)} = v_{L_k}^{(k)} = x_0$, and x_0 being the depot. We highlight three structural properties of such solutions and motivate the corresponding desiderata.

Anisometry. The Euclidean distances between consecutive nodes $\|x_{v_{i+1}} - x_{v_i}\|_2$ generally vary with i within a route, so the cumulative distance $d_i = \sum_{j=2}^{i} \|x_{v_j} - x_{v_{j-1}}\|_2$ is a nonlinear function of the discrete index. Equal increments in index do not correspond to equal increments in physical distance. Standard positional encodings that map uniform indices to embedding space implicitly assume isometric spacing and discard this geometric information. Table 1 confirms that this anisometry is pronounced in near-optimal solutions across VRP variants. For each route in 1,000 instances per problem solved with LKH/HGS (Helsgaun, 2017; Vidal, 2022), we report the coefficient of variation (CV) of consecutive edge lengths, the max-to-min edge-length ratio (max/min), and the residual median absolute deviation (MAD) of a linear fit of d_i against i in units of the average edge length. The CV is large in every setting and grows with problem size on CVRP. Edge-length ratios reach the tens to hundreds, and the residual MAD shows that a linear index-to-distance fit deviates by roughly a third to a half of an average edge length on the multi-route variants.

Table 1: Anisometry measurement of near-optimal routes across VRP variants.

Problem	CV	max/min	MAD
CVRP-100	0.81	26.6	0.37
CVRP-200	0.93	42.1	0.44
CVRP-500	1.12	72.3	0.53
VRPTW-100	0.78	24.0	0.33
PDTSP-51	0.78	68.9	1.24

Circularity. Under the closed-tour assumption, the first and last positions of every route are the same physical node, yet a PE built from a linear index does not identify them. For variants with directional constraints such as VRPTW, traversing a route in reverse changes feasibility, so the encoding must preserve direction; without such constraints, both directions have the same cost, so a direction-invariant encoding generalizes more readily across equivalent solution representations. A routing-aware PE should respect both regimes.

Hierarchy. A multi-route solution is a two-level object: a node's identity is determined jointly by its within-route position and its route membership. A one-dimensional positional index conflates these two levels, so the "second node of route A" and the "second node of route B" receive identical encodings. We quantify the resulting ambiguity by collecting the set of customers occupying position k across the routes of an instance and measuring the entropy of their depot-relative polar-angle distribution. Across CVRP-{100, 200, 500} and VRPTW-100, the normalized entropy averaged over

$k = 2, \ldots, 15$ is 0.992, with a minimum of 0.953 over the entire (problem, k) grid: conditioning on the within-route index k reduces the binned angular entropy by at most 4.7% of its maximum (Section Q). A PE that maps the abstract index k into a node embedding therefore cannot well distinguish nodes that share k but lie in different parts of the instance.

These properties yield three desiderata for a routing PE.

Distance-awareness (D1): the PE of a node should be a function of its cumulative travel distance along the route rather than of its discrete index, so that, for nodes v_i, v_j on the same route, $\|\mathrm{PE}(v_i) - \mathrm{PE}(v_j)\|$ tracks their along-route separation $|d_i - d_j|$ rather than $|i - j|$ (measured cyclically for closed tours, see D2). D1 concerns the input signal and the rank agreement it induces, not isometry.

Circularity (D2): for closed-tour variants, the PE should satisfy $\mathrm{PE}(v_1) = \mathrm{PE}(v_L)$ (topological clause), and its behavior under route reversal should match the variant's symmetry: invariant for reversal-symmetric variants, sensitive otherwise (directional clause).

Hierarchical structure (D3): the PE should encode both within-route position and an across-route cue that disambiguates nodes sharing a within-route index but belonging to different routes, without depending on an arbitrary route numbering. We distinguish an *instance-level* geometric cue, which a per-node PE can supply, from exact *solution-level* route membership; D3 as used here targets the former as a proxy for the latter (Section Q).

3.2 A Taxonomy of Existing PE Methods

We organize existing PE methods into three categories by their design origin and discuss, for each method, which of desiderata are satisfied at the conceptual level.

Table 2: Conceptual satisfaction of the routing PE desiderata by the standard form of existing methods. ✓: guaranteed by construction; ∼: partially satisfied; ×: not satisfied (criteria in Section E).

Method	D1	D2	D3
APE	×	×	×
Sinusoidal (SIN)	×	×	×
RoPE	×	∼	×
RPE	×	∼	×
ALiBi	×	×	×
Laplacian PE	×	×	∼
RWSE	×	×	∼
SPD	∼	×	×
DACT CPE	×	✓	×
CycleFormer PE	×	∼	×

NLP-Originated PE. These encodings were designed for sequential text data and assume uniform token spacing. *Absolute PE (APE)* maps each position i to a trainable vector $\mathbf{e}_i \in \mathbb{R}^D$ (Gehring et al., 2017), and *Sinusoidal PE (SIN)* to fixed sinusoidal functions of i (Vaswani et al., 2017); both assume uniform spacing, are not circular, have no hierarchical component, and thus satisfy none of D1–D3. *Rotary PE (RoPE)* rotates query–key pairs by position-dependent angles (Su et al., 2024). Because its position-dependent logit term depends only on $i - j$, RoPE is invariant to translations (not cyclic shifts) of the index, which we count as partial D2, but it does not identify the first and last positions and does not satisfy D1 or D3. *Relative PE (RPE)* adds learnable bias terms to attention logits based on the relative offset $i - j$ (Shaw et al., 2018). Like RoPE, RPE captures relative ordering but assumes uniform spacing and lacks both full circularity and hierarchy. *ALiBi* subtracts a head-specific linear penalty $m \cdot |i - j|$ from attention logits (Press et al., 2021), largest for the first and last positions, and satisfies none of D1–D3. All of these encodings are parameterized by *abstract index sequences*, which carry no information about the geometry of the solution.

Graph Transformer PE. These encodings exploit graph-structural information and are used in graph transformers: *Laplacian PE (Lap. PE)* uses the k smallest non-trivial eigenvectors of the graph Laplacian as node features (Dwivedi & Bresson, 2020). *Random Walk SE (RWSE)* uses the diagonals of random-walk matrix powers $\{RW^k\}_{k=1}^K$ (Dwivedi et al., 2021). Both capture solution-level graph structure and so partially satisfy D3, but encode neither route-level distance (D1) nor circularity (D2). *Shortest-Path Distance (SPD)* uses the shortest-path distance between node pairs as a bias in attention (Ying et al., 2021). SPD partially satisfies D1 (distance-awareness through graph distances) but does not address D2 or D3 in the routing context.

Routing-Specific PE. These encodings are designed with routing structure in mind. *DACT CPE* (Ma et al., 2021) assigns cyclic codes to tour positions, satisfying D2, but encodes neither travel

distance (D1) nor cross-route information (D3). *CycleFormer Circular PE* is designed specifically for TSP and constructs a circular PE that wraps position indices around the tour (Yook et al., 2024). Its cyclic phase term closes over the tour, but the added index-sinusoidal bands do not, so it only partially satisfies D2; like DACT CPE, it does not address D1 or D3.

In summary, Table 2 shows that no existing PE method satisfies all three desiderata: NLP-originated encodings address none, or at most partially address D2, graph transformer encodings partially address D3 but ignore the routing-specific properties, and existing routing-specific PEs target circularity but neither distance-awareness nor hierarchy.

4 Hierarchical Positional Encoding

This section introduces a positional encoding constructed to satisfy the desiderata D1–D3 of Section 3.1. The encoding consists of two complementary components defined per node: an *in-route positional encoding* (IPE) capturing the distance-indexed, circular ordering of a node within its route, and a *cross-route positional encoding* (XPE) capturing the depot-anchored angular position of the node.

4.1 In-Route Component: IPE

Distance-indexed sinusoidal encoding. For a route $r = (v_1, \dots, v_L)$ and $v_1 = v_L = x_0$, define the cumulative travel distance along the route,

$$d_i = \sum_{j=2}^{i} \left\| x_{v_j} - x_{v_{j-1}} \right\|_2, \qquad d_L \;=\; \text{total route length} > 0.$$

We rescale this quantity to the base angular interval, $\hat{d}_i \;=\; 2\pi\, d_i / d_L \;\in\; [0, 2\pi]$, and compose a multi-frequency encoding using integer harmonics of this period, $\omega_k = k+1$ for $k = 0, \dots, D/2-1$. Restricting ω_k to integers is essential: it guarantees that every frequency band completes a whole number of cycles over one tour, so that $\sin(\omega_k \hat{d})$ and $\cos(\omega_k \hat{d})$ are 2π-periodic in $\hat{d}$ for all k (a geometric, non-integer schedule as in Vaswani et al. (2017) would break this closure). The construction admits two variants, selected to match the symmetry of the VRP variant being solved.

Direction-aware variant. For variants whose feasibility is sensitive to traversal direction (VRPTW, PDTSP, and any problem with precedence or time-window constraints), we use the standard sine–cosine pair,

$$\mathrm{IPE}^{\rightarrow}(v_i) \;=\; \left[\sin(\omega_k \hat{d}_i),\, \cos(\omega_k \hat{d}_i)\right]_{k=0}^{D/2-1}.$$

Reversing the route maps $\hat{d}_i \mapsto 2\pi - \hat{d}_i$; since ω_k is an integer, $\sin(\omega_k(2\pi - \hat{d})) = -\sin(\omega_k \hat{d})$ and $\cos(\omega_k(2\pi - \hat{d})) = \cos(\omega_k \hat{d})$, so reversal flips the sign of every sine component and leaves the cosine components unchanged, and the encoding distinguishes a route from its reversal.

Direction-invariant variant. For reversal-symmetric variants (CVRP, TSP), we retain only the cosine components,

$$\mathrm{IPE}^{\leftrightarrow}(v_i) \;=\; \left[\cos(\omega_k \hat{d}_i)\right]_{k=0}^{D-1},$$

with $\omega_k = k+1$ for $k = 0, \dots, D-1$, i.e., D integer harmonics (twice as many as in the direction-aware variant) to keep the embedding dimension fixed at D. By the identity above, this encoding is invariant to route reversal by construction, and the directional clause of D2 is satisfied without architectural symmetrization (unlike, e.g., DACT, which augments training with reversed tours).

Relation to D1 and D2. The encoding is a function of cumulative travel distance rather than discrete index, so the embedding gap between two nodes on the same route is governed by their along-route separation rather than their index offset, as D1 requires; the multi-frequency map is not a global isometry of this separation, and we measure the resulting rank agreement empirically (Section 5.2). The rescaling maps both endpoints of a closed tour to the same phase, so $\mathrm{IPE}(v_1) = \mathrm{IPE}(v_L)$ for both variants, satisfying the topological clause of D2 exactly. Two trade-offs are deliberate: the cosine-only variant identifies phases $\hat{d}$ and $2\pi - \hat{d}$ in exchange for reversal invariance, and normalizing by d_L removes absolute route scale (ablated in Section P.1).

4.2 Cross-Route Component: XPE

Per-node depot-anchored polar encoding. For each customer v with coordinate $x_v \in \mathbb{R}^2$, define the depot-anchored polar angle

$$\theta_v \;=\; \operatorname{atan2}\big(x_v^{(y)} - x_0^{(y)},\; x_v^{(x)} - x_0^{(x)}\big) \;\in\; [-\pi, \pi),$$

and encode it via a multi-frequency sinusoidal map of the same form as IPE,

$$\mathrm{XPE}(v) \;=\; \big[\sin(\omega'_k \theta_v),\, \cos(\omega'_k \theta_v)\big]_{k=0}^{K-1}, \qquad \omega'_k = 2^k,$$

where the number of frequency bands K is chosen so that $2K \le D$; the remaining dimensions of XPE are zero-padded to match the IPE width.

Choice of polar over Cartesian. We encode θ_v rather than the Cartesian offset $x_v - x_0$ because the polar angle is bounded and periodic, matching the sinusoidal form of IPE, and because near-optimal routes are organized around the depot by direction rather than by signed displacement (Section 3); Table 11 compares both.

Why per node rather than per route. One could instead summarize each route by a single angular phase (e.g., a mean or median of $\{\theta_v\}_{v \in r}$) and broadcast it to the route's nodes. No such aggregation is canonical, and given the route assignment the multiset $\{\theta_v\}_{v \in r}$ recovers any of them; a per-route aggregate does, however, inject the assignment itself, so we compare against both aggregates and explicit membership signals (Sections P.2 and Q).

Relation to D3. XPE is static: θ_v depends on the instance, not on the current assignment, so XPE alone does not encode route membership, and a node that changes route keeps its XPE. It supplies the instance-level cue of D3: near-optimal routes occupy largely separated angular sectors around the depot, so two nodes sharing a within-route index but lying on different routes typically differ in θ; their IPE need not coincide either, since IPE depends on fractional along-route distance rather than on the index. The near-maximal conditional entropy of Section 3.1 shows that the index carries little angular information, so the two signals are complementary, but not that θ identifies routes. This cue weakens when routes overlap angularly, e.g., under tight capacity or early in the search; Section Q measures this overlap, relates the gain of XPE to it, and compares XPE with explicit membership signals.

4.3 Integration into the Encoder

The two positional components are concatenated per node and projected jointly with the node features by a shared feed-forward layer,

$$h_v^{(0)} \;=\; \mathrm{FF}\big([\,x_v \,\|\, \mathrm{IPE}(v) \,\|\, \mathrm{XPE}(v)\,]\big),$$

where x_v denotes the node features of the backbone (coordinates together with demand, time windows, prizes, or pickup–delivery pairing as applicable). For backbones with a dedicated positional stream (DACT, N2S), $[\mathrm{IPE} \,\|\, \mathrm{XPE}]$ instead replaces the input of that stream (Section G). Constraint states such as load, arrival times, and feasibility masks remain those of the backbone. IPE is recomputed by cumulative sums whenever the solution changes ($O(N)$ per step), and XPE once per instance and augmentation. We adopt concatenation rather than additive PE because the node features and the sinusoidal signals live on different numerical scales, and we avoid any specialized cross-route mechanism so that comparisons against PE baselines isolate the effect of the encoding.

5 Experiments

The experiments are organized along three layers. Layer 1 asks *"how much does the PE itself matter"* and isolates the encoding under a fixed lightweight backbone at small instance sizes, where training ten-plus encodings is feasible. Layer 2 asks *"can our PE push the strongest current method further"* and plugs the encoding into NDS (Hottung et al., 2025b), comparing against the strongest operations-research, neural, and alternative-PE baselines. Layer 3 asks *"does the encoding transfer across architectures and distributions"*, plugging it into N2S (Ma et al., 2022) on pickup-and-delivery problems and evaluating NDS checkpoints zero-shot under distribution shifts.

5.1 SETUP

Problems and datasets. We evaluate on four VRP variants. CVRP is the capacitated VRP; we use the uniform random instances of Kool et al. (2019) and the clustered instances of Queiroga et al. (2022). VRPTW adds time windows following Solomon (1987) to the uniform instances at $N = 100$; at larger sizes, VRPTW and PCVRP use the clustered generator of NDS (Hottung et al., 2025b), with time windows or with prizes that increase with demand. PDTSP is the pickup-and-delivery TSP, used to validate cross-architecture transfer. Sizes are $N = 100$ (CVRP, VRPTW) and $N = 51$ (PDTSP) at Layer 1, $N \in \{500, 1000, 2000\}$ (CVRP, VRPTW, PCVRP) at Layer 2, and $N \in \{21, 51, 101\}$ (PDTSP) at Layer 3.

Backbones. We pair each layer with the backbone best suited to its cost and purpose: DACT (Ma et al., 2021) for Layer 1 (small and cheap, enabling end-to-end retraining of more than ten encodings), NDS for Layer 2 (state-of-the-art neural deconstruction at scale), and N2S for Layer 3 (neural neighborhood search for pickup-and-delivery). All three admit a drop-in replacement of the positional component with all other architectural and training choices fixed (Section G).

Training and evaluation protocol. Within each layer, all encodings share the same training schedule and search procedure; only the PE module differs, and each (PE, problem) pair is trained from scratch following the published recipe of its backbone. Test sets contain 1,000 instances per (problem, size) configuration and are fixed across all PE methods within a layer; reported numbers are means over three independent training seeds, with seed SDs and paired per-instance differences with bootstrap confidence intervals in Section R. Full training and search hyperparameters appear in Section K; the Layer 1 budget and a convergence check are in Section L.

5.2 RESULTS AND ANALYSIS

We ask two questions at the encoding level: do existing PEs actually encode the desiderata D1–D3 of Section 3.1, and does this translate into downstream solution quality when only the encoding is varied?

Table 3: Spearman correlation between PE-induced pairwise distances and the three desiderata D1–D3 on CVRP-100; values in $[-1, 1]$, averaged over instances. †: bias/rotation encodings, probed via their position-only logit contribution (Section F). Best in **bold**, second best in underline.

Method	D1 ↑	D2 ↑	D3 ↑
NoPE	-	-	-
APE	0.002	- 0.001	0.003
SIN	- 0.310	- 0.319	0.329
RoPE†	- 0.290	- 0.125	0.072
RPE†	0.007	0.010	0.001
ALiBi†	- 0.310	- 0.319	0.327
Laplacian PE	0.715	0.721	0.359
RWSE	0.090	0.086	0.138
SPD†	0.824	0.830	0.332
DACT CPE	0.096	0.096	0.017
CycleFormer PE	0.126	0.085	0.218
IPE	**0.934**	0.866	0.229
XPE	0.815	0.806	**0.396**
IPE + XPE	0.877	**0.923**	0.384

Probing the encodings. For each PE we compute the Spearman rank correlation between embedding-space pairwise distances $\|\mathrm{PE}(v_i) - \mathrm{PE}(v_j)\|_2$ and a structural target aligned with each desideratum (D1: in-route cumulative travel-distance gap; D2: cyclic arc distance; D3: whether the pair lies on different routes), using 10^6 node pairs over 1,000 near-optimal HGS solutions on CVRP-100; correlations are computed per instance and averaged. The probe fits no parameters; learnable encodings are read from trained Layer 1 checkpoints, and bias- or rotation-type encodings are probed through their position-only logit contribution, so values across the two families are only indicative (Section F). Table 3 shows that the index-sinusoid family (SIN, RoPE, ALiBi) and APE/RPE carry no positive D1/D2 signal, the cyclic encodings (DACT CPE, CycleFormer PE) score similarly low, and the graph-transformer encodings recover D1/D2 only partially, with SPD strongest among prior methods. IPE+XPE is the only encoding strong on all three targets, consistent with the taxonomy of Table 2.

Controlled comparison across encodings. We retrain all thirteen PE methods plus the no-PE baseline with the DACT backbone on CVRP-100, VRPTW-100, and PDTSP-51. Table 4 confirms the probing trend at the solution level: distance-aware encodings outperform index-based ones, circularity alone does not match a distance-aware design, and the in-route and cross-route components are

Table 4: Controlled comparison of positional encoding methods under the DACT backbone (Layer 1, Section 5.1). All methods share the encoder, decoder, optimization schedule, and evaluation protocol; only the positional encoding differs. Obj.: objective (↓), Gap: relative gap to the operations-research reference (PyVRP-HGS for CVRP, VRPTW, LKH for PDTSP) (↓). Seed SDs and paired differences are in Table 15.

	PE Method	CVRP-100		VRPTW-100		PDTSP-51	
		Obj.↓	Gap↓	Obj.↓	Gap↓	Obj.↓	Gap↓
	HGS/LKH	15.58	-	25.42	-	6.87	-
	No PE	17.42	11.81%	29.99	17.96%	7.95	15.86%
NLP	APE	16.34	4.88%	27.05	6.40%	7.20	4.93%
NLP	Sinusoidal (SIN)	16.45	5.58%	27.34	7.54%	7.27	5.95%
NLP	RoPE	16.42	5.39%	27.22	7.07%	7.24	5.51%
NLP	RPE	16.30	4.62%	27.00	6.20%	7.18	4.63%
NLP	ALiBi	16.40	5.26%	27.30	7.39%	7.26	5.80%
Graph	Laplacian PE	15.91	2.12%	26.55	4.43%	7.07	3.06%
Graph	RWSE	16.06	3.08%	26.79	5.38%	7.13	3.94%
Graph	SPD	15.86	1.80%	26.41	3.88%	7.05	2.74%
Rout.	DACT CPE	15.95	2.37%	26.58	4.55%	7.05	2.84%
Rout.	CycleFormer PE	15.94	2.31%	26.55	4.43%	7.05	2.74%
Ours	IPE	15.81	1.48%	26.32	3.53%	7.01	2.16%
Ours	XPE	15.84	1.67%	26.27	3.33%	7.02	2.30%
Ours	**IPE + XPE**	**15.76**	**1.16%**	**26.18**	**2.97%**	**6.99**	**1.87%**

Table 5: Main results on CVRP, VRPTW, and PCVRP. Gap is relative to the reference solver (HGS for CVRP, PyVRP-HGS for VRPTW and PCVRP); Time is per-instance wall-clock. Negative gaps mean lower cost than the reference solver under the same time budget. Seed SDs and paired differences are in Table 15.

Methods	CVRP-500			CVRP-1000			CVRP-2000		
	Obj.↓	Gap↓	Time↓	Obj.↓	Gap↓	Time↓	Obj.↓	Gap↓	Time↓
HGS	36.66	-	60s	41.51	-	121s	57.38	-	241s
NDS	36.57	-0.20%	60s	41.11	-0.90%	120s	56.00	-2.34%	240s
IPE+XPE	**36.54**	**-0.30%**	60s	**41.05**	**-1.11%**	120s	**55.98**	**-2.45%**	240s

Methods	VRPTW-500			VRPTW-1000			VRPTW-2000		
	Obj.↓	Gap↓	Time↓	Obj.↓	Gap↓	Time↓	Obj.↓	Gap↓	Time↓
PyVRP-HGS	49.01	-	60s	90.35	-	120s	173.46	-	240s
NDS	47.94	-2.17%	60s	87.54	-3.14%	120s	167.48	-3.50%	240s
IPE+XPE	**47.90**	**-2.27%**	60s	**87.45**	**-3.21%**	120s	**166.80**	**-3.84%**	240s

Methods	PCVRP-500			PCVRP-1000			PCVRP-2000		
	Obj.↓	Gap↓	Time↓	Obj.↓	Gap↓	Time↓	Obj.↓	Gap↓	Time↓
PyVRP-HGS	44.97	-	60s	84.91	-	120s	165.56	-	240s
NDS	**43.12**	**-4.12%**	60s	80.99	-4.71%	120s	158.09	-4.60%	240s
IPE+XPE	43.13	-4.09%	60s	**80.77**	**-4.87%**	120s	**157.37**	**-4.94%**	240s

individually useful, with their combination performing best in every setting. The downstream ranking broadly tracks the probing ranking, consistent with the backbone exploiting what the encoding records. Matching the IPE variant to the problem's symmetry matters: the direction-invariant variant raises the gap from 2.97% to 4.04% on VRPTW-100 and from 1.87% to 3.71% on PDTSP-51, but slightly lowers it on CVRP-100 (1.16% vs. 1.28%; Table 13). Linearly projecting the same distance recovers only part of the gain of IPE (Table 11), so both the distance and its Fourier map matter.

Table 6: Model-agnostic validation on PDTSP using the N2S backbone. HA-N2S replaces only the PE with our hierarchical encoding. Obj: objective (↓), Gap: gap to LKH (↓).

Methods	PDTSP-21			PDTSP-51			PDTSP-101		
	Obj.↓	Gap↓	Time↓	Obj.↓	Gap↓	Time↓	Obj.↓	Gap↓	Time↓
LKH (10k)	4.563	0.00%	5m	6.862	0.00%	19m	9.428	0.00%	98m
DACT (3k)	4.564	0.03%	1m	7.057	2.83%	1.5m	10.195	8.13%	2.5m
N2S (3k)	4.565	0.05%	1m	7.027	2.40%	1.5m	9.846	4.44%	3m
HA-N2S (3k)	**4.564**	**0.03%**	1m	**6.997**	**1.96%**	1.5m	**9.701**	**2.89%**	3m

Main Results across VRP variants. We plug IPE+XPE into NDS without other modifications and evaluate against operations-research and neural baselines on CVRP, VRPTW, and PCVRP. Table 5 shows that the resulting model improves on the original NDS in eight of the nine (problem, size) cells by 0.07–0.34 percentage points, with PCVRP-500 the only exception (within 0.03). Since these margins are small, we test them per instance: the 95% bootstrap interval of the paired difference to NDS excludes zero in the same eight cells, and replacing the native PE of NDS by SIN, SPD, or Laplacian PE under the same protocol yields smaller gains (Sections R and S). Following the symmetry rule of Section 4.1, VRPTW uses the direction-aware IPE, and CVRP and PCVRP, whose objectives and constraints are reversal-symmetric, the direction-invariant one.

Cross-architecture transfer and generalization. We plug the same IPE+XPE module into N2S, replacing only its positional component, and train and evaluate on PDTSP at $N \in \{21, 51, 101\}$. Table 6 shows that the resulting HA-N2S improves on the original N2S at every instance size, with the improvement growing with N; since each size is trained separately, this is a cross-architecture rather than a cross-size generalization result. For zero-shot generalization, the CVRP-500 NDS checkpoints are evaluated under tighter capacity and clustered customer layouts (Section N) and on 84 CVRPLib X instances (Uchoa et al., 2017) (Section O); IPE+XPE attains the lowest gap among NDS, SIN, SPD, and Laplacian PE in all three, with small margins on the synthetic shifts.

6 Discussion

When to use which PE. For improvement methods that observe the full current solution, IPE+XPE, with the direction variant chosen by the problem's reversal symmetry, is the only tested design that targets all of D1–D3 and performed best in each controlled comparison we ran. For single-route problems no cross-route disambiguation is needed, yet XPE can still add a global angular cue (PDTSP-51: 1.87% with XPE vs. 2.16% without), and the distance-indexed IPE remains beneficial over standard sinusoidal encoding. For construction methods we make no empirical claim, since a partial solution has no final route length d_L (Section T).

Limitations and future work. The evidence is anchored to three improvement/search backbones; validation on *construction* backbones remains open, and at large scale the gains over NDS are small (below 0.35 percentage points). XPE is a static, depot-anchored proxy for route membership whose benefit shrinks with angular overlap (Section Q) and which needs planar coordinates and a single depot (Section T); multi-depot and non-Euclidean settings remain open.

7 Conclusion

We presented a systematic study of positional encoding for neural vehicle routing: we formalized three structural desiderata, organized prior PE methods into a taxonomy in which none of the surveyed encodings targets all three, and proposed a hierarchical PE combining a distance-indexed, circularly consistent in-route encoding with a depot-anchored angular cross-route encoding. Experiments show gains over index-based and graph-transformer encodings across the variants, backbones, and shifts we tested, with margins at scale that are small but resolved by paired tests in eight of nine settings. Within the improvement-based setting studied here, respecting the anisometric, cyclic, and hierarchical nature of routes in the PE yields measurably better results.

AI USE STATEMENT

We use LLMs as writing assistants for improving clarity, style, and grammar, providing feedback and as coding assistants. Notably, the core research contributions including the design of the framework and validation of results were conceived and verified by the authors. All outputs from LLMs were critically assessed, refined, and integrated to ensure correctness and adherence to academic standards.

REFERENCES

Jan Christiaens and Greet Vanden Berghe. Slack induction by string removals for vehicle routing problems. *Transportation Science*, 54(2):417–433, 2020.

Darko Drakulic, Sofia Michel, Florian Mai, Arnaud Sors, and Jean-Marc Andreoli. BQ-NCO: Bisimulation Quotienting for Efficient Neural Combinatorial Optimization. *Advances in Neural Information Processing Systems*, 36, 2023.

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. *arXiv preprint arXiv:2110.07875*, 2021.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *International conference on machine learning*, pp. 1243–1252. PMLR, 2017.

Nathan Grinsztajn, Daniel Furelos-Blanco, Shikha Surana, Clément Bonnet, and Tom Barrett. Winner Takes It All: Training Performant RL Populations for Combinatorial Optimization. *Advances in Neural Information Processing Systems*, 36, 2023.

Keld Helsgaun. An extension of the Lin-Kernighan-Helsgaun TSP solver for constrained traveling salesman and vehicle routing problems. Technical report, Roskilde University, 2017.

André Hottung, Mridul Mahajan, and Kevin Tierney. PolyNet: Learning Diverse Solution Strategies for Neural Combinatorial Optimization. *International Conference on Learning Representations*, 2025a.

André Hottung, Paula Wong-Chung, and Kevin Tierney. Neural Deconstruction Search for Vehicle Routing Problems. *Transactions on Machine Learning Research*, 2025b.

Wouter Kool, Herke Van Hoof, and Max Welling. Attention, learn to solve routing problems! *International Conference on Learning Representations*, 2019.

Yeong-Dae Kwon, Jinho Choo, Byoungjip Kim, Iljoo Yoon, Youngjune Gwon, and Seungjai Min. POMO: Policy optimization with multiple optima for reinforcement learning. *Advances in Neural Information Processing Systems*, 33:21188–21198, 2020.

Fu Luo, Xi Lin, Fei Liu, Qingfu Zhang, and Zhenkun Wang. Neural combinatorial optimization with heavy decoder: Toward large scale generalization. *Advances in Neural Information Processing Systems*, 36:8845–8864, 2023.

Yining Ma, Jingwen Li, Zhiguang Cao, Wen Song, Le Zhang, Zhenghua Chen, and Jing Tang. Learning to iteratively solve routing problems with dual-aspect collaborative transformer. *Advances in Neural Information Processing Systems*, 34:11096–11107, 2021.

Yining Ma, Jingwen Li, Zhiguang Cao, Wen Song, Hongliang Guo, Yuejiao Gong, and Yeow Meng Chee. Efficient Neural Neighborhood Search for Pickup and Delivery Problems. *International Joint Conference on Artificial Intelligence*, 2022.

NVIDIA Corporation. NVIDIA cuOpt: GPU-Accelerated Decision Optimization. https://github.com/NVIDIA/cuopt, 2025. Accessed: 2025-09-25.

Ofir Press, Noah A Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*, 2021.

Eduardo Queiroga, Ruslan Sadykov, Eduardo Uchoa, and Thibaut Vidal. 10,000 optimal cvrp solutions for testing machine learning based heuristics. In *AAAI-22 Workshop on Machine Learning for Operations Research (ML4OR)*, 2022.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 464–468, 2018.

Marius M Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research*, 35(2):254–265, 1987.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Eduardo Uchoa, Diego Pecin, Artur Pessoa, Marcus Poggi, Thibaut Vidal, and Anand Subramanian. New benchmark instances for the Capacitated Vehicle Routing Problem. *European Journal of Operational Research*, 257(3):845–858, 2017. doi: 10.1016/j.ejor.2016.08.012.

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.

Thibaut Vidal. Hybrid genetic search for the CVRP: Open-source implementation and SWAP* neighborhood. *Computers & Operations Research*, 140:105643, 2022.

Niels A Wouda, Leon Lan, and Wouter Kool. PyVRP: A high-performance VRP solver package. *INFORMS Journal on Computing*, 36(4):943–955, 2024. doi: 10.1287/ijoc.2023.0055.

Yubin Xiao, Yuesong Wu, Rui Cao, Di Wang, Zhiguang Cao, Xuan Wu, Peng Zhao, Yuanshu Li, You Zhou, and Yuan Jiang. DGL: Dynamic global-local information aggregation for scalable VRP generalization with self-improvement learning. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 8669–8677, 2025.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do Transformers Really Perform Badly for Graph Representation? In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=OeWooOxFwDa.

Jieun Yook, Junpyo Seo, Joon Huh, Han Joon Byun, and Byung-ro Moon. Cycleformer: TSP solver based on language modeling, 2024. arXiv:2405.20042v1; withdrawn by the authors.

A Detailed VRP Definitions

We consider a complete directed graph $G = (V, E)$ with $V = \{0, 1, \ldots, N\}$, where x_0 is the depot and x_i for $i \geq 1$ are customers. Let $\mathbf{x}_i \in \mathbb{R}^2$ denote the coordinates of node x_i and define travel cost $c_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2$ for $(i, j) \in E$. A fleet $K = \{1, \ldots, M\}$ of identical vehicles of capacity C is available. We use binary routing variables $x_{ij}^k \in \{0, 1\}$ indicating whether vehicle k traverses arc (i, j) and service variables $y_i^k \in \{0, 1\}$ indicating whether vehicle k serves customer i. For convenience, we also write $y_i = \sum_{k \in K} y_i^k$.

Capacitated VRP (CVRP). Each customer $i \in \{1, \ldots, N\}$ has demand $d_i \geq 0$. The objective is to minimize total travel distance:

$$\min_{x,y} \quad \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} c_{ij}\, x_{ij}^k. \tag{1}$$

Subject to service, capacity, flow, and depot constraints:

$$\sum_{k \in K} y_i^k = 1 \qquad \forall\, i \in \{1, \ldots, N\}, \tag{CVRP-1a}$$

$$\sum_{i \in \{1,\ldots,N\}} \sum_{j \in V} d_i\, x_{ij}^k \leq C \qquad \forall\, k \in K, \tag{CVRP-1b}$$

$$\sum_{j \in V} x_{ij}^k = y_i^k, \quad \sum_{j \in V} x_{ji}^k = y_i^k \qquad \forall\, i \in \{1, \ldots, N\},\ \forall\, k \in K, \tag{CVRP-1c}$$

$$\sum_{j \in V} x_{0j}^k = 1, \quad \sum_{i \in V} x_{i0}^k = 1 \qquad \forall\, k \in K, \tag{CVRP-1d}$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij}^k \leq |S| - 1 \qquad \forall\, S \subseteq \{1, \ldots, N\},\ |S| \geq 2,\ \forall\, k \in K, \tag{CVRP-1e}$$

$$x_{ii}^k = 0 \qquad \forall\, i \in \{1, \ldots, N\},\ \forall\, k \in K. \tag{CVRP-1f}$$

Constraints (CVRP-1a) ensure each customer is served exactly once, (CVRP-1b) enforce per-vehicle capacity, (CVRP-1c) link service and flow, (CVRP-1d) ensure each vehicle departs from and returns to the depot, (CVRP-1e) eliminate subtours that do not contain the depot, and (CVRP-1f) forbids customer self-loops, the single-customer case not covered by (CVRP-1e). Unused vehicles are represented by the empty route $x_{00}^k = 1$.

VRP with Time Windows (VRPTW). Each customer i has a service duration $s_i \geq 0$ and an admissible time window $[a_i, b_i]$ within which service must start. Early arrival is allowed (vehicles may wait); tardiness is not. Let travel time $\tau_{ij} = c_{ij}$ and introduce service start times $t_i \in \mathbb{R}_{\geq 0}$. The objective is to minimize total travel time:

$$\min_{x,y,t} \quad \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} \tau_{ij}\, x_{ij}^k, \tag{2}$$

subject to CVRP constraints and time-feasibility:

$$a_i \leq t_i \leq b_i \qquad \forall\, i \in \{1, \ldots, N\}, \tag{VRPTW-2b}$$

$$t_j \geq t_i + s_i + \tau_{ij} - M_{\text{tw}}\big(1 - \textstyle\sum_{k \in K} x_{ij}^k\big) \qquad \forall\, i \in V,\ j \in \{1, \ldots, N\}, \tag{VRPTW-2c}$$

$$t_i + s_i + \tau_{i0} \leq b_0 + M_{\text{tw}}\big(1 - \textstyle\sum_{k \in K} x_{i0}^k\big) \qquad \forall\, i \in \{1, \ldots, N\}, \tag{VRPTW-2d}$$

where $t_0 = a_0 = 0$ is the departure time from the depot, $s_0 = 0$, $[a_0, b_0]$ is the depot horizon, and M_{tw} is a sufficiently large constant. Excluding the depot as the head j in (VRPTW-2c) avoids a contradictory time cycle through the depot; the return time is constrained separately by (VRPTW-2d).

Prize-Collecting VRP (PCVRP). Customers carry prizes $p_i \geq 0$ and need not all be served. The objective trades off travel cost and the prizes of unvisited customers:

$$\min_{x,y} \quad \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} c_{ij}\, x_{ij}^k + \sum_{i \in \{1,\ldots,N\}} p_i\,(1 - y_i). \tag{3}$$

This equals travel cost minus collected prizes plus the instance constant $\sum_i p_i$, so it has the same optimal solutions; we use this nonnegative form, as in PyVRP and NDS, for all reported PCVRP objectives and gaps. Capacity, flow, depot, and subtour-elimination constraints follow CVRP, except that (CVRP-1a) is relaxed to $\sum_{k\in K} y_i^k \le 1$, so that $y_i \in \{0,1\}$ indicates whether customer i is visited; (CVRP-1c) ties y_i^k to the vehicle flows.

Pickup and Delivery TSP (PDTSP). The node set is partitioned into pickup nodes $P = \{1,\dots,n\}$ and delivery nodes $D = \{n+1,\dots,2n\}$, with each request r pairing pickup r and delivery $r+n$. A single vehicle visits all $2n$ nodes exactly once while ensuring each pickup precedes its paired delivery. The objective is:

$$\min_{x,u} \quad \sum_{i\in V}\sum_{j\in V} c_{ij}\, x_{ij}, \tag{4}$$

subject to degree constraints, MTZ-style subtour elimination $u_j \ge u_i + 1 - M_{\text{sub}}(1 - x_{ij})$ for all $i, j \in V \setminus \{0\}$ with $u_i \in [1, 2n]$ (the depot, node 0, is excluded so that the tour can close), and precedence constraints $u_{r+n} \ge u_r + 1$ for $r \in P$.

B MDP Formulation

We describe the MDP of the Layer 2 backbone (NDS); DACT and N2S use the k-opt and pair-removal-reinsertion MDPs of their original papers (Ma et al., 2021; 2022), unchanged except for the PE. NDS models deconstruction as a Markov Decision Process (MDP) in which a policy selects customers to remove from a feasible solution, after which a repair procedure reinserts them to reduce total travel cost. The *state* $s_t = \Psi(l, \mathbf{s}_t)$ encodes instance features and the current solution, including positional signals (IPE, XPE) attached per node. The *action space* at step t selects one customer for removal, $x_t \in \{1,\dots,N\}$, with the depot never selectable and infeasible actions masked. Within a rollout, the *transition* $\mathcal{T}(\mathbf{s}_t, x_t)$ removes the selected customer and updates the mask; once the prescribed number of customers has been removed, the randomized greedy repair of Section H reinserts them, yielding a complete feasible solution $\mathbf{s}_{t+1}$, on which IPE is recomputed. The *reward* is the improvement over the incumbent best $\mathbf{s}_t^\star$:

$$r_t = \text{cost}(\mathbf{s}_t^\star) - \min\{\text{cost}(\mathbf{s}_{t+1}),\, \text{cost}(\mathbf{s}_t^\star)\},$$

which is positive iff a new incumbent is found. The *policy* $\pi_\theta(x_t \mid l, \mathbf{s}_t, v, x_{1:t-1})$ is conditioned on a rollout seed v for stochastic diversification, and actions are sampled during training and selected stochastically at test time.

C Formal Definition of Anisometry

For a route $r = (v_1,\dots,v_L)$ with node coordinates $x_{v_i} \in \mathbb{R}^2$, the Euclidean distances between consecutive nodes $\|x_{v_{i+1}} - x_{v_i}\|_2$ generally vary with i. The cumulative-distance mapping $d_i = \sum_{j=2}^{i} \|x_{v_j} - x_{v_{j-1}}\|_2$ is a nonlinear function of the discrete index i: increments in the index do not translate to uniform increments in physical space. We refer to such structures as *anisometric* to emphasize that VRP routes possess heterogeneous spatial separations that must be represented explicitly when designing positional encodings.

D Detailed PE Implementation Formulas

We summarize the precise form in which each baseline PE is plugged into the routing encoder. Throughout, i denotes the position of a node in its current route (with the depot indexed 0 and L the route length), D is the embedding dimension, and $h_v^{(0)}$ denotes the input embedding of node v before the first attention layer. All variants share the same encoder, decoder, and training schedule of their respective backbone; only the PE module differs. For methods that produce a per-node vector, we follow the integration in Section 4.3 and concatenate it with the node features before a shared feed-forward projection (for DACT and N2S, it is the input of the positional stream; Section G). For methods defined as an attention bias, we add the bias term to attention logits at every encoder layer.

Absolute PE (APE). A learnable lookup table $\mathbf{E} \in \mathbb{R}^{L_{\max} \times D}$ indexed by within-route position:

$$\mathrm{PE}^{\mathrm{APE}}(v_i) = \mathbf{E}_i, \qquad \mathbf{E}_i \in \mathbb{R}^D \text{ trainable},$$

with $L_{\max}$ chosen as the maximum route length observed during training.

Sinusoidal PE (SIN). Fixed sinusoidal functions of the integer index i (Vaswani et al., 2017):

$$\mathrm{PE}^{\mathrm{SIN}}_{2k}(v_i) = \sin\!\big(i/\lambda^{2k/D}\big), \quad \mathrm{PE}^{\mathrm{SIN}}_{2k+1}(v_i) = \cos\!\big(i/\lambda^{2k/D}\big),$$

with $\lambda = 10{,}000$ and $k = 0, \ldots, D/2 - 1$.

Rotary PE (RoPE). RoPE is realized as a query/key rotation rather than an additive embedding (Su et al., 2024). For each pair of channels $(2k, 2k+1)$ in the query $\mathbf{q}_i$ and key $\mathbf{k}_j$, we apply

$$\mathrm{R}_{\theta_i}\begin{pmatrix}\mathbf{q}_{i,2k}\\ \mathbf{q}_{i,2k+1}\end{pmatrix}, \quad \mathrm{R}_{\theta_j}\begin{pmatrix}\mathbf{k}_{j,2k}\\ \mathbf{k}_{j,2k+1}\end{pmatrix}, \qquad \theta_i = i \cdot \lambda^{-2k/D},$$

where R_θ is the 2×2 rotation matrix. The position-dependent part of the attention logit depends only on the index difference $i - j$; the full logit additionally depends on the content of $\mathbf{q}_i$ and $\mathbf{k}_j$.

Relative PE (RPE). A learnable bias indexed by signed index offset is added to each attention logit (Shaw et al., 2018):

$$\mathrm{logit}(i, j) \mathrel{+}= b_{\,\mathrm{clip}(i-j,\,-W,\,W)},$$

where $\{b_\Delta\}_{\Delta=-W}^{W} \in \mathbb{R}$ is shared across heads and clipped to a window of size W. This is the scalar-bias simplification of the relative representations of Shaw et al. (2018), which additionally use relative-position vectors in keys and values.

ALiBi. A fixed linear penalty subtracted from attention logits (Press et al., 2021):

$$\mathrm{logit}(i, j) \mathrel{-}= m_h \cdot |i - j|,$$

with head-specific slopes m_h following the geometric schedule of the original paper.

Laplacian PE (Lap. PE). For each current solution we form the *route graph*: an undirected, unweighted graph on V with an edge for every pair of consecutive nodes on a route, where all routes share the single depot node. Laplacian PE, RWSE, and SPD are all computed on this graph and recomputed whenever the solution changes. We compute the unnormalized Laplacian $\mathbf{L}$. The PE is the matrix of the K smallest non-trivial eigenvectors $\mathbf{U}_K \in \mathbb{R}^{|V| \times K}$ (Dwivedi & Bresson, 2020), with sign disambiguation by random flips at training time:

$$\mathrm{PE}^{\mathrm{Lap}}(v) = \mathbf{U}_K[v, :].$$

We pad to D dimensions with zeros and use $K = 8$.

Random Walk SE (RWSE). A node-wise vector built from the diagonal of random-walk matrix powers (Dwivedi et al., 2021):

$$\mathrm{PE}^{\mathrm{RWSE}}(v) = \big[(\mathbf{R}^k)_{vv}\big]_{k=1}^{K}, \qquad \mathbf{R} = \mathbf{D}^{-1}\mathbf{A},$$

on the same route graph used for Laplacian PE, with $K = 8$ and zero padding to D.

Shortest-Path Distance (SPD). A bias term added to attention logits, indexed by shortest-path distance on the route graph (Ying et al., 2021):

$$\mathrm{logit}(i, j) \mathrel{+}= b_{\mathrm{spd}(v_i, v_j)},$$

with a learnable scalar bias per integer hop distance, capped at the diameter of the route graph. Because the graph is unweighted, spd counts hops rather than travel cost.

DACT CPE. The cyclic PE introduced by DACT (Ma et al., 2021) maps the within-tour index to a real-valued embedding whose channels are periodic functions of the index, constructed after cyclic Gray codes so that the embedding is cyclic over the tour and neighboring positions receive similar codes:

$$\mathrm{PE}^{\mathrm{DACT}}(v_i) = \mathrm{CPE}_L(i),$$

where CPE_L is the authors' construction with the tour length replaced by the route length L for multi-route problems. We use the authors' implementation and feed it, as in DACT, to the positional stream of the encoder.

CycleFormer Circular PE. A circular variant of the sinusoidal map that adds a tour-level cyclic phase to every band (Yook et al., 2024):

$$\mathrm{PE}^{\mathrm{Cyc}}_{2k}(v_i) = \sin\big(i/\lambda^{2k/D} + 2\pi i/L\big), \quad \mathrm{PE}^{\mathrm{Cyc}}_{2k+1}(v_i) = \cos\big(i/\lambda^{2k/D} + 2\pi i/L\big),$$

following the original formulation (Eq. 9 of arXiv v1) with the number of tour nodes replaced by the route length L. The arXiv preprint has since been withdrawn by its authors; we retain only this PE, implemented from v1, as a representative circular variant of the sinusoidal map, and make no claim about the full CycleFormer solver. The cyclic phase $2\pi i/L$ closes over the route, whereas the index term $i/\lambda^{2k/D}$ does not, which is why Table 2 marks D2 as partial.

Our IPE+XPE. For completeness, the full method (Section 4) computes per node

$$\mathrm{IPE}(v_i) = \big[\sin(\omega_k \hat{d}_i),\, \cos(\omega_k \hat{d}_i)\big]_{k=0}^{D/2-1}, \quad \mathrm{XPE}(v) = \big[\sin(\omega'_k \theta_v),\, \cos(\omega'_k \theta_v)\big]_{k=0}^{K-1},$$

with integer harmonics $\omega_k = k + 1$ (so that every band is 2π-periodic in $\hat{d}_i$) and $\omega'_k = 2^k$, and concatenates the two with the node features before a shared feed-forward projection. For reversal-symmetric variants IPE keeps only cosine components, $\big[\cos(\omega_k \hat{d}_i)\big]_{k=0}^{D-1}$, so the output width stays D. The depot is assigned $\hat{d} = 0$ (equivalently 2π) on every route, and a route of zero length, which cannot occur for a route visiting at least one customer at a distinct location, would be assigned $\hat{d}_i = 0$ for all its nodes. In PCVRP, a customer that lies on no current route has no along-route phase; it receives the zero vector as IPE, which no routed node attains, and enters a route only through the backbone's repair operator, after which its IPE is recomputed. The depot, for which $\mathrm{atan2}(0, 0)$ is undefined, is assigned $\theta_0 = 0$.

Parameter and width matching. IPE and XPE are each of width D, so $[\mathrm{IPE} \,\|\, \mathrm{XPE}]$ has width $2D$; every other vector-valued PE, including IPE or XPE alone in the ablations, is zero-padded to the same width $2D$ and enters through the same projection, so the first-layer parameter count is identical across vector-valued PEs; bias-type encodings (RPE, ALiBi, SPD) add at most a few hundred scalar parameters. No PE-specific hyperparameters were tuned beyond the values stated above.

E Criteria for the Conceptual Taxonomy

The marks of Table 2 refer to the standard form of each encoding, not to a particular trained model. For D1, ✓ requires the encoding to be a function of along-route travel distance; $\sim$ means it depends on a graph or hop distance that correlates with travel distance only through the solution structure (SPD); $\times$ means it depends only on indices. For D2, ✓ requires $\mathrm{PE}(v_1) = \mathrm{PE}(v_L)$ by construction for every route length; $\sim$ means the encoding either closes only a subset of its bands (CycleFormer PE), or depends on positions only through the offset $i - j$ (invariance to translations of the index origin, not to cyclic shifts modulo L) without forcing the first and last positions to be the most separated pair (RoPE, RPE); $\times$ means neither, including ALiBi, whose penalty is monotone in $|i - j|$. The D2 marks score the topological clause only; the directional clause depends on the variant and is examined empirically (Table 13). For D3, $\sim$ means the encoding is computed on the whole multi-route solution and can therefore differ across routes, without an explicit across-route cue (Laplacian PE, RWSE); $\times$ means it is computed from the within-route index alone. These are properties of the encodings, not guarantees of downstream performance, which Section 5.2 measures.

F PROBING PROTOCOL

Pairs and targets. For each of the 1,000 HGS solutions on CVRP-100 we sample 1,000 customer pairs. D1 and D2 use same-route pairs, with targets $|d_i - d_j|$ and the cyclic arc distance $\min(|d_i - d_j|, d_L - |d_i - d_j|)$. D3 uses customer pairs sampled so that half lie on the same route and half on different routes, with the binary target $\mathbb{1}$[different routes], so its Spearman coefficient measures, on ranks, how well the PE distance separates different-route from same-route pairs. Correlations are computed per instance and then averaged over instances, so the statistical unit is the instance rather than the pair; the across-instance SD is below 0.05 for every entry of Table 3.

Pairwise quantity per encoding family. For vector-valued encodings (APE, SIN, Laplacian PE, RWSE, DACT CPE, CycleFormer PE, and ours) the pairwise quantity is $\|\mathrm{PE}(v_i) - \mathrm{PE}(v_j)\|_2$ of the PE output before the projection. Bias-type encodings (RPE, ALiBi, SPD) have no node vector; we use the negated bias $-b(i,j)$, averaged over heads, so that a larger penalty counts as a larger distance. For RoPE we use the negated position-dependent part of the logit, $-(\mathbf{q}_i^\top \mathrm{R}_{\theta_j - \theta_i}\mathbf{k}_j - \mathbf{q}_i^\top \mathbf{k}_j)$, averaged over heads and over the first-layer queries and keys of the trained model. Because bias- and rotation-type quantities are logit contributions rather than embedding distances, we mark them in Table 3 and do not rank them against vector-valued encodings in the text. A negative coefficient means that the ranking induced by the encoding is inverted with respect to the target, not merely absent.

Learnable encodings. The probe itself fits no parameters. Encodings with trainable components (APE, RPE, SPD) are read from the trained Layer 1 CVRP-100 checkpoints and averaged over the three training seeds; fixed encodings, including the cyclic code of DACT CPE before any projection, are evaluated in closed form.

Analytic checks. Independently of the probe, IPE satisfies $\mathrm{IPE}(v_1) = \mathrm{IPE}(v_L)$ exactly, reversal flips every sine channel and preserves every cosine channel of $\mathrm{IPE}^{\rightarrow}$, and $\mathrm{IPE}^{\leftrightarrow}$ is reversal-invariant; all encodings of Table 3 are invariant to renumbering the routes, since none of them uses a route index.

G BACKBONE INTEGRATION, NODE FEATURES, AND RUNTIME

Insertion points. Section 4.3 gives the generic form. DACT and N2S keep separate node-feature and positional streams; there, $[\mathrm{IPE} \,\|\, \mathrm{XPE}]$ replaces the cyclic PE as the input of the positional stream, and the node stream, the dual-aspect (DACT) or synthesis (N2S) attention, the decoders, and the training algorithms are unchanged. NDS adds an index-based PE to the node embeddings of its encoder; there, $[x_v \,\|\, \mathrm{IPE} \,\|\, \mathrm{XPE}]$ is projected by the input layer that replaces the original embedding plus PE, and all later layers are unchanged. Bias-type baselines are added to the attention logits of every encoder layer in all three backbones.

Node features and constraint state. x_v denotes the node features of the respective backbone: coordinates and demand for CVRP; additionally time windows and service times for VRPTW, prizes for PCVRP, and pickup–delivery pairing for PDTSP. Load, arrival times, slack, and precedence feasibility are handled by the backbone's masks and features exactly as in the original implementations and are identical across PE methods. IPE and XPE are not substitutes for these states.

Update frequency and cost. XPE depends only on the instance and is computed once per instance, and again after each geometric augmentation, which changes θ_v. IPE is recomputed after every change of the current solution by a cumulative sum along each route, which costs $O(N)$ and can be restricted to the modified routes. On CVRP-2000 with NDS, computing IPE and XPE takes less than 1% of the per-step time, and the number of search steps completed within the 240 s budget differs from native NDS by less than 1%; SPD and Laplacian PE are recomputed on the route graph after every step and reduce the number of completed steps by 9% and 6%, respectively (Section S).

H Decoder, Improvement Strategy, and Training

Decoder. We employ a PolyNet-style pointer network decoder (Hottung et al., 2025a). At step m, a GRU is conditioned on the last removals and produces a query q_m; keys K are linear projections of the encoder outputs $\mathbf{h} = \{\hat{h}_i\}$. The probability of selecting a node is:

$$\pi_\theta(x_m \mid l, \mathbf{s}, v, x_{1:m-1}) = \mathrm{softmax}\left(\frac{q_m K^\top}{\sqrt{d}} + \mathrm{mask}_m\right).$$

Improvement (ASA). At test time, the policy is embedded in augmented simulated annealing (ASA) (Hottung et al., 2025b). Each improvement step samples routes, deconstructs by removing chosen customers, repairs with a randomized greedy routine, and evaluates candidates. Acceptance follows:

$$\Pr[\text{accept } \mathbf{s}' \mid \mathbf{s}] = \min\left(1, \exp\left(-\frac{\mathrm{cost}(\mathbf{s}') - \mathrm{cost}(\mathbf{s})}{\sigma}\right)\right).$$

Training. We use winner-takes-all policy-gradient training (Grinsztajn et al., 2023; Hottung et al., 2025a). Among the M rollouts per instance, only the best one, $i^\star = \arg\max_i R_i$, contributes to the policy gradient:

$$\nabla_\theta J(\theta) \approx (R_{i^\star} - b)\, \nabla_\theta \log \pi_\theta(\mathbf{s}_{i^\star} \mid \mathcal{S}),$$

where b is the rollout baseline.

I Data Generation

Except for the Layer 1 VRPTW time windows, all generators follow the public NDS implementation (Hottung et al., 2025b).

CVRP instances. Uniform instances follow Kool et al. (2019): depot and customer coordinates are sampled uniformly in the unit square, demands are $d_i \sim \mathrm{Unif}\{1, \dots, 9\}$, and vehicles are identical with size-dependent capacity $C = 50, 100, 200, 300$ for $N = 100, 500, 1000, 2000$, as in NDS. Structured instances use the X generator of Uchoa et al. (2017) in the implementation of Queiroga et al. (2022), on a 1000×1000 grid rescaled to the unit square, with a centered depot. Under clustered customer positioning, $S \sim \mathrm{Unif}\{2, \dots, 6\}$ cluster seeds are placed uniformly, and further customers are drawn uniformly and accepted with probability proportional to $\sum_s 2^{-\|\mathbf{x}-\mathbf{x}_s\|_2/40}$ (grid units) until N customers exist. X-generator demands are $d_i \sim \mathrm{Unif}\{1, \dots, 10\}$, and the capacity is set per instance to $C = \max\big(\max_i d_i,\ \lceil r \sum_i d_i / N \rceil\big)$ with $r \sim \mathrm{Unif}[8, 12]$ (medium route size). Travel cost is Euclidean distance and speed is 1 so that time equals distance.

PCVRP instances. Locations, demands, and capacity come from the clustered X generator above, and each customer receives a prize that mixes a demand-proportional and a random term:

$$p_i = \alpha\left(p_{\min} + (p_{\max} - p_{\min})\tfrac{d_i - d_{\min}}{d_{\max} - d_{\min}}\right) + (1 - \alpha)\, u_i, \qquad u_i \sim \mathrm{Unif}[p_{\min}, p_{\max}],$$

where $d_{\min}$ and $d_{\max}$ are the smallest and largest demand in the instance, and $p_{\min} = 0.1$, $p_{\max} = 0.2$, $\alpha = 0.5$ as in NDS.

VRPTW instances. In both settings below, the single-customer route $(0, i, 0)$ is feasible for every i, so a feasible solution always exists, and infeasible moves are masked at decode time using the standard arrival-plus-service feasibility check. At $N = 100$ (Layer 1), we reuse the uniform CVRP instances and synthesize time windows following Solomon (1987), with service times $s_i \sim \mathrm{Unif}[0.15, 0.18]$, window lengths $w_i \sim \mathrm{Unif}[0.18, 0.20]$, depot horizon $[a_0, b_0] = [0, 4.6]$, window starts $a_i \sim \mathrm{Unif}[\tau_{0i},\ b_0 - s_i - w_i - \tau_{i0}]$, and $b_i = a_i + w_i$. At $N \geq 500$ (Layer 2), locations, demands, and capacity come from the clustered X generator above with the NDS time-window parameters: $[a_0, b_0] = [0, 1.8]$, $s_i = 0.05$, window centers $c_i \sim \mathrm{Unif}[\tau_{0i},\ b_0 - s_i - \tau_{i0}]$ (on the grid), and windows $[c_i - 0.2,\ c_i + 0.2]$ clipped to that interval.

J OR BASELINES

HGS (Vidal, 2022). Hybrid Genetic Search combining population-based recombination with powerful local search (ejection chains, k-opt, route exchanges) and adaptive penalty management; we use the public implementation with author-recommended settings.

SISRs (Christiaens & Vanden Berghe, 2020). Slack Induction by String Removals, a ruin-and-recreate framework that removes contiguous strings and repairs with tailored insertion heuristics.

LKH3 (Helsgaun, 2017). Extended Lin-Kernighan-Helsgaun heuristic with sophisticated candidate sets and powerful move pruning; supports several VRP variants in addition to TSP.

PyVRP (Wouda et al., 2024). Open-source Python/C++ solver implementing a modern HGS variant with route-improvement neighborhoods. We use version 0.9.0 with default configuration.

NVIDIA cuOpt (NVIDIA Corporation, 2025). GPU-accelerated local-search engine for rich VRPs; we follow the vendor's recommended settings for our variants.

K TRAINING CONFIGURATION

We provide the training and evaluation configuration for the Layer 2 (NDS) backbone in Table 7. For Layer 3 (N2S), we use the published configuration of the original paper without modification (Ma et al., 2022); only the positional encoding module is replaced. This includes optimizer, learning rate schedule, batch size, training horizon, curriculum, and inference protocol. Layer 1 (DACT) uses the published architecture and training algorithm with a shorter schedule, detailed in Section L.

Compute resources. All training and evaluation runs use a single NVIDIA A100 (40GB) GPU with one CPU core. At test time, each instance is evaluated under a fixed wall-clock budget of 60, 120, or 240 seconds for sizes $N = 500$, 1000, and 2000, respectively, matching the protocol of NDS (Hottung et al., 2025b). Layer 3 (N2S) follows the compute budget specified in the original paper (Ma et al., 2022); the Layer 1 budget is given in Section L.

L LAYER 1 BUDGET AND CONVERGENCE

Configuration. Layer 1 uses the DACT architecture and PPO training algorithm of Ma et al. (2021) (encoder, dual-aspect attention, decoder, feasibility masks, and optimizer settings as released by the authors), with a shorter schedule so that fourteen encodings, three problems, and three seeds can be trained: 100 epochs of 10,240 instances with batch size 512 and $T_{\text{train}} = 200$ steps, and inference with $T = 1{,}000$ improvement steps and $8\times$ augmentation for CVRP and VRPTW. PDTSP follows the N2S-adapted DACT setting of Ma et al. (2022) with $T = 3{,}000$. For VRPTW, DACT is extended with the time-window feasibility mask of the same code base, and the action space is unchanged. With the published training schedule and $T = 10{,}000$, our DACT CPE reproduction reaches a CVRP-100 gap within 0.1 percentage points of the value reported by Ma et al. (2021). Absolute Layer 1 gaps are therefore higher than published DACT results and should be read as budget-matched comparisons rather than converged performance.

Convergence and budget sensitivity. Table 8 varies the training budget from $0.5\times$ to $2\times$ and the search budget by $5\times$ for the six encodings that matter most for our conclusions. Gaps keep decreasing with longer training, confirming that the $1\times$ models are not fully converged; however, the ranking of the six encodings is identical in every column, and the margin of IPE+XPE over the strongest baseline (SPD) changes only from 0.64 to 0.58 percentage points at $2\times$ training and to 0.37 at $5\times$ search. The advantage thus shrinks but persists as the budget grows, so part of it reflects faster learning and part a better final representation. Checkpoints are selected by validation gap on a separate set of 1,000 instances, with the same rule for every encoding.

Table 7: Training and evaluation hyperparameters for the Layer 2 (NDS) backbone. For Layer 3 (N2S), we use the published configuration without modification, replacing only the positional encoding module; the Layer 1 (DACT) budget is given in Section L.

Hyperparameter	Value
Optimization	
Optimizer	Adam
Initial learning rate	10^{-4}
Weight decay	10^{-6}
LR scheduler	MultiStepLR (milestones: 60, 85; $\gamma = 0.1$)
Warmup	Linear, 5,000 steps (start factor 0.01)
Mixed precision	Yes (AMP)
Gradient accumulation steps	100
Optimizer step interval	Every 100 inner updates
Neural Architecture	
Embedding dimension	128
Number of attention heads	8
QKV dimension	16
Message passing layers	1
Encoder layers	2
Feedforward dimension	512
Decoder	GRUCell($128 \to 128$) + attention
Polynomial head dim	256
Logit clipping	10
Latent vector (z) dimension	10
Inference rule (training)	Softmax sampling
Training Dynamics	
Total training epochs	2000
Iterations per epoch	150,000
Batch size	64
Rollout size (per instance)	128
Skipped iterations (search warmup)	10
Nodes removed per rollout	15
Destroy/repair recreate parameter	5
Reward type	Improvement (absolute)
Evaluation	
Validation episodes	1,000
Iterations per episode	50
Validation batch size	50
Rollouts per instance (validation)	200
Augmented rollouts	$8\times$ (8-fold geometric augmentation)

Table 8: Budget sensitivity of the Layer 1 comparison on CVRP-100 (gap, ↓, mean over three training seeds). Training budget is expressed relative to the Layer 1 schedule ($1\times$); the last column evaluates the $1\times$ checkpoints with $5\times$ more search steps.

	Training budget			Search budget
PE Method	$0.5\times$	$1\times$	$2\times$	$5\times$
No PE	13.95%	11.81%	11.02%	9.84%
SIN	6.87%	5.58%	5.01%	3.92%
DACT CPE	3.41%	2.37%	1.98%	1.21%
Laplacian PE	3.05%	2.12%	1.79%	1.10%
SPD	2.66%	1.80%	1.52%	0.95%
IPE + XPE	**1.83%**	**1.16%**	**0.94%**	**0.58%**

Table 9: Zero-shot generalization of CVRP-500 NDS checkpoints (medium capacity, uniform layout) to a low-capacity setting, a clustered customer layout, and 84 CVRPLib X instances, without retraining. PE rows replace only the positional component of NDS. Gap is relative to HGS on the synthetic shifts and to the BKS on CVRPLib (↓).

Methods	Low Capacity		Clustered		CVRPLib X
	Obj.↓	Gap↓	Obj.↓	Gap↓	Gap↓
HGS	91.73	–	44.53	–	–
SISRs	91.34	−0.38%	44.31	−0.49%	–
NDS	91.15	−0.59%	44.29	−0.54%	4.127%
SIN	91.08	−0.70%	44.25	−0.63%	3.951%
Laplacian PE	91.07	−0.72%	44.24	−0.65%	3.412%
SPD	91.06	−0.73%	44.23	−0.67%	3.284%
IPE + XPE	**91.04**	−0.76%	**44.22**	−0.70%	**2.842%**

M HA-N2S Configuration

For the HA-N2S experiment, we follow the original N2S setup (Ma et al., 2022) in all training, model, and inference configurations, replacing only the positional encoding with our proposed hierarchical anisometric embedding. The encoder and decoder structures, including the Synthesis Attention module and the two PDTSP-specific decoders, remain unchanged; only the PE component is substituted while keeping embedding dimensionality identical. Training uses the same PPO-based procedure with curriculum learning, batch size, training horizon, and optimization hyperparameters, separately for $|V| = 21, 51, 101$; the three sizes therefore test transfer across architectures, not zero-shot transfer across sizes. Inference uses the same augmentation-based search scheme and step counts (1k, 2k, 3k), with feasibility constraints, masking rules, and reinsertion operations kept fully consistent with the N2S framework.

N Distribution Shift

Using the NDS backbone of Layer 2, we evaluate the same checkpoints trained on uniform-distribution medium-capacity CVRP-500 instances, for every PE, on two distribution shifts without retraining or re-tuning: a low-capacity setting ($C = 50$ instead of 100) that produces more and shorter routes, and a clustered customer layout (depot and customers placed by the clustered X generator of Section I, with demands and capacity as in training), both at $N = 500$ with 1,000 test instances. Table 9 reports the gap relative to HGS on each shift, together with the CVRPLib average of Section O. IPE+XPE attains the best objective and gap in both synthetic settings, but its margin over the strongest alternative PE (SPD) is only 0.03 percentage points, and SIN, SPD, and Laplacian PE all improve on native NDS; on the synthetic shifts, most of the benefit over native NDS is thus shared by non-native encodings. The separation is larger on CVRPLib, where the shift combines size, layout, demand, and route-length changes, so we do not attribute it to a single factor. In the low-capacity setting, solutions contain more routes and their angular sectors overlap more (Section Q), which is the regime in which the static angular cue of XPE is least informative; under clustered layouts, intra-route edge lengths become highly nonuniform, which the distance-indexed in-route component represents where index-based encodings assume equal spacing.

O CVRPLib Detailed Results

As a further out-of-distribution test, we evaluate on 84 of the 100 instances of the X set of CVRPLib (Uchoa et al., 2017) using the NDS backbone of Layer 2. The X set is a benchmark designed to vary depot position, customer layout, demand distribution, and route length; it covers customer counts $N \in [100, 1000]$. The 16 instances from X-n298-k31 to X-n393-k38 are not part of the evaluated set; the set was fixed before results were inspected, and no evaluated instance failed or timed out. We directly apply the CVRP-500 checkpoints without any retraining or size-specific adaptation, with a 60-second time limit per instance. Each Obj. entry is the mean over the three training seeds of a

Table 10: Detailed evaluation on 84 instances of the X set of CVRPLib: best-known solutions (BKS), NDS and IPE+XPE costs, and their percentage gaps (the last row is the equal-weight average over the 84 listed instances). Models trained on CVRP-500 are applied without retraining.

		NDS		IPE+XPE				NDS		IPE+XPE	
Instance	BKS	Obj.	Gap%	Obj.	Gap%	Instance	BKS	Obj.	Gap%	Obj.	Gap%
X-n101-k25	27591	27696	0.381	27696	0.381	X-n401-k29	66154	68272	3.202	67508	2.047
X-n106-k14	26362	26711	1.324	26567	0.778	X-n411-k19	19712	20837	5.707	20490	3.947
X-n110-k13	14971	15165	1.296	15082	0.741	X-n420-k130	107798	112070	3.963	110825	2.808
X-n115-k10	12747	12765	0.141	12762	0.118	X-n429-k61	65449	68765	5.067	67692	3.427
X-n120-k6	13332	13595	1.973	13518	1.395	X-n439-k37	36391	37377	2.709	37085	1.907
X-n125-k30	55539	56943	2.528	56551	1.822	X-n449-k29	55233	57363	3.856	56606	2.486
X-n129-k18	28940	29326	1.334	29190	0.864	X-n459-k26	24139	25361	5.062	24966	3.426
X-n134-k13	10916	11110	1.777	11050	1.228	X-n469-k138	221824	246070	10.930	238130	7.351
X-n139-k10	13590	13699	0.802	13643	0.390	X-n480-k70	89449	93513	4.543	92099	2.963
X-n143-k7	15700	16160	2.930	16000	1.911	X-n491-k59	66483	69124	3.972	68373	2.843
X-n148-k46	43448	44351	2.078	44076	1.445	X-n502-k39	69226	70762	2.219	70297	1.547
X-n153-k22	21220	21609	1.833	21466	1.159	X-n513-k21	24201	25313	4.595	24952	3.103
X-n157-k13	16876	16957	0.480	16957	0.480	X-n524-k153	154593	161430	4.423	159215	2.990
X-n162-k11	14138	14262	0.877	14210	0.509	X-n536-k96	94846	98716	4.080	97547	2.848
X-n167-k10	20557	21012	2.213	20905	1.693	X-n548-k50	86700	93650	8.016	91431	5.457
X-n172-k51	45607	46201	1.302	46048	0.967	X-n561-k42	42717	44239	3.563	43732	2.376
X-n176-k26	47812	49478	3.484	48940	2.359	X-n573-k30	50673	52296	3.203	51740	2.106
X-n181-k23	25569	25966	1.553	25806	0.927	X-n586-k159	190316	206204	8.348	201364	5.805
X-n186-k15	24145	24364	0.907	24312	0.692	X-n599-k92	108451	115105	6.135	113179	4.360
X-n190-k8	16980	17536	3.274	17377	2.338	X-n613-k62	59535	62191	4.461	61294	2.955
X-n195-k51	44225	45025	1.809	44750	1.187	X-n627-k43	62164	66173	6.449	65024	4.601
X-n200-k36	58578	60847	3.873	60041	2.498	X-n641-k35	63684	66650	4.657	65782	3.294
X-n204-k19	19565	19924	1.835	19814	1.273	X-n655-k131	106780	110878	3.838	109468	2.517
X-n209-k16	30656	31059	1.315	30973	1.034	X-n670-k130	146332	160574	9.733	156357	6.851
X-n214-k11	10856	11236	3.500	11102	2.266	X-n685-k75	68205	70980	4.069	70186	2.904
X-n219-k73	117595	120258	2.265	119247	1.405	X-n701-k44	81923	85467	4.326	84368	2.985
X-n223-k34	40437	41043	1.499	40885	1.108	X-n716-k35	43373	45545	5.008	44907	3.537
X-n228-k23	25742	26685	3.663	26389	2.513	X-n733-k159	136187	143437	5.324	141241	3.711
X-n233-k16	19230	19726	2.579	19594	1.893	X-n749-k98	77269	81463	5.428	80249	3.857
X-n237-k14	27042	27971	3.435	27726	2.529	X-n766-k71	114417	122799	7.326	120327	5.165
X-n242-k48	82751	84082	1.608	83717	1.167	X-n783-k48	72386	76158	5.211	75081	3.723
X-n247-k50	37274	38867	4.274	38307	2.771	X-n801-k40	73311	79846	8.914	77895	6.253
X-n251-k28	38684	39733	2.712	39404	1.861	X-n819-k171	158121	169901	7.450	166419	5.248
X-n256-k16	18839	19233	2.091	19086	1.311	X-n837-k142	193737	212659	9.767	207069	6.881
X-n261-k13	26558	27434	3.298	27200	2.417	X-n856-k95	88965	93339	4.917	91904	3.304
X-n266-k58	75478	80431	6.562	78942	4.589	X-n876-k59	99299	103914	4.648	102414	3.137
X-n270-k35	35291	35918	1.777	35726	1.233	X-n895-k37	53860	57016	5.860	56042	4.051
X-n275-k28	21245	21850	2.848	21642	1.869	X-n916-k207	329179	362272	10.053	352372	7.046
X-n280-k17	33503	34693	3.552	34273	2.298	X-n936-k151	132715	146776	10.595	142537	7.401
X-n284-k15	20226	21570	6.645	21158	4.608	X-n957-k87	85465	92705	8.471	90414	5.791
X-n289-k60	95151	99201	4.256	97797	2.781	X-n979-k58	118976	128300	7.837	125484	5.470
X-n294-k50	47161	48074	1.936	47789	1.332	X-n1001-k43	72355	78832	8.952	76758	6.085
						Avg. Gap%	*	4.127%		**2.842%**	

single evaluation run, rounded to an integer; distances follow the CVRPLib convention of rounding each Euclidean edge length to the nearest integer, and BKS values are those listed on CVRPLib at the time of evaluation. Table 10 reports the gap to the BKS on each instance and the equal-weight average across the 84 listed instances; averages for SIN, SPD, and Laplacian PE under the same protocol are in Table 9. IPE+XPE attains a smaller gap than NDS on 82 of the 84 instances and ties on the remaining two, with the largest improvements on the largest instances. Because training on CVRP-500 and testing on X instances changes size and distribution simultaneously, this is a test of generalization to a heterogeneous benchmark rather than evidence of robustness in a specific deployment setting.

Table 11: Component and input-signal ablation on CVRP-100 (DACT backbone). In-route variants are used without XPE; "Fourier" denotes the integer-harmonic map of Section 4.1 with D output channels and "linear" a learned linear projection to the same width. The index control uses the same phase range as IPE, $2\pi(i-1)/(L-1)$, so that only the signal changes. Node coordinates are part of the input in every row.

Signal	Mapping	Obj.↓	Gap↓
No PE		17.42	11.81%
In-route signal (no XPE)			
Index $2\pi(i-1)/(L-1)$	Fourier	16.18	3.85%
Raw distance d_i	linear	16.02	2.83%
Normalized d_i/d_L	linear	15.97	2.50%
Raw distance d_i	Fourier	15.89	1.97%
Normalized d_i/d_L and d_L	Fourier	15.81	1.45%
Normalized d_i/d_L (IPE)	Fourier	15.81	1.48%
Cross-route signal			
XPE only	Fourier ($K=4$)	15.84	1.67%
IPE + depot offset $x_v - x_0$	linear	15.79	1.34%
IPE + $(\sin\theta_v, \cos\theta_v)$	Fourier ($K=1$)	15.78	1.29%
IPE + XPE (full)	Fourier ($K=4$)	**15.76**	**1.16%**

P Ablation Studies (Layer 1)

The ablation interrogates three design choices under the same DACT backbone and small-scale protocol used for the PE-level analysis (Section 5.2), which keeps the cost of each ablation cell within a single training cycle: component contributions (Section P.1), per-node versus per-route XPE design (Section P.2), and the direction-aware versus direction-invariant IPE switch (Section P.3). All variants share the encoder, decoder, optimization schedule, and evaluation protocol of the Layer 1 controlled comparison; only the PE module is altered.

P.1 Component Ablation

Table 11 separates the input signal from its mapping, using the same output width and the same projection in every row. *Signal.* With the Fourier map and phase range held fixed, replacing the normalized distance by the normalized index $2\pi(i-1)/(L-1)$ raises the gap from 1.48% to 3.85%, so along-route distance rather than ordering carries most of the in-route gain. *Mapping.* Feeding the same distance through a linear projection recovers only part of this gain (2.83% raw, 2.50% normalized), so the multi-frequency map adds a benefit beyond exposing the scalar distance. *Normalization.* Unnormalized distance under the Fourier map, with phase $2\pi d_i/\bar{d}$ for a global scale $\bar{d}$ (the mean training route length), is worse (1.97%), because a fixed set of harmonics cannot close routes of different lengths; adding the route length d_L as an extra input to the normalized encoding changes the gap only marginally (1.45% vs. 1.48%), so discarding absolute route scale costs little on CVRP-100. *Cross-route cue.* Since node coordinates are already part of every input, a depot-relative Cartesian offset adds little over IPE alone (1.34%), a single angular band helps more (1.29%), and the multi-band XPE is best (1.16%). Removing either component degrades the gap, and in this setting the combination performs best.

P.2 Per-node vs. Per-route XPE

A natural alternative to per-node XPE is to compute a single angular phase per route and broadcast the result to all of its nodes. We compare against five such aggregations: the arithmetic mean, the circular mean $\arg(\sum_v e^{i\theta_v})$, the demand-weighted mean, the median, and the depot-anchored max-min midpoint.

Table 12 shows that all five per-route variants underperform the per-node design, that no single aggregation is uniformly best among them, and that the circular mean, the only aggregate that respects the wrap-around at $\pm\pi$, is the strongest of them. The arithmetic mean and median are computed

Table 12: Per-node vs. per-route XPE under the DACT backbone. Per-route variants compute a single angular phase per route, encode it via the same sinusoidal map as XPE, and broadcast the result to all nodes of the route. The per-node design (used in our main method) preserves the full multiset $\{\theta_v\}_{v\in r}$ and is not tied to any particular aggregation. Reference values for CVRP-100 IPE+XPE and VRPTW-100 IPE+XPE follow Table 4.

XPE Variant	CVRP-100		VRPTW-100	
	Obj.↓	Gap↓	Obj.↓	Gap↓
Per-route, arithmetic mean	15.83	1.61%	26.30	3.46%
Per-route, circular mean	15.82	1.54%	26.27	3.34%
Per-route, demand-weighted mean	15.83	1.61%	26.31	3.50%
Per-route, median	15.84	1.67%	26.32	3.54%
Per-route, max-min midpoint	15.85	1.73%	26.34	3.62%
Per-node (ours)	**15.76**	**1.16%**	**26.18**	**2.97%**

Table 13: Direction-aware ($\mathrm{IPE}^{\rightarrow}$) vs. direction-invariant ($\mathrm{IPE}^{\leftrightarrow}$) variants of IPE within the full IPE+XPE model, on a reversal-symmetric variant (CVRP-100) and two direction-sensitive variants (VRPTW-100, PDTSP-51). The last row restricts $\mathrm{IPE}^{\leftrightarrow}$ to the same $D/2$ harmonics as $\mathrm{IPE}^{\rightarrow}$ (zero-padded to width D), separating the effect of direction from that of band coverage. The variant chosen for each problem in the main method (Table 4) is bold. Reference values follow Table 4.

Variant of IPE	CVRP-100		VRPTW-100		PDTSP-51	
	Obj.↓	Gap↓	Obj.↓	Gap↓	Obj.↓	Gap↓
$\mathrm{IPE}^{\rightarrow}$ (direction-aware)	15.78	1.28%	**26.18**	**2.97%**	**6.99**	**1.87%**
$\mathrm{IPE}^{\leftrightarrow}$ (direction-invariant)	**15.76**	**1.16%**	26.45	4.04%	7.12	3.71%
$\mathrm{IPE}^{\leftrightarrow}$, $D/2$ harmonics	15.77	1.22%	26.47	4.13%	7.13	3.78%

on $\theta_v \in [-\pi, \pi)$ and are therefore weak for routes that straddle the cut at $\pm\pi$; they are included as commonly used simple choices. These per-route variants share the angular source of XPE; the comparison against signals that encode route membership itself is in Section Q.

P.3 Direction-aware vs. Direction-invariant IPE

We compare $\mathrm{IPE}^{\rightarrow}$ and $\mathrm{IPE}^{\leftrightarrow}$ on a reversal-symmetric variant (CVRP-100) and two direction-sensitive variants (VRPTW-100 and PDTSP-51).

Table 13 reports the result. On CVRP, the direction-invariant variant marginally improves on the direction-aware variant, consistent with the prediction that building reversal symmetry into the encoding removes a learning burden the model would otherwise absorb. On VRPTW and PDTSP, the direction-aware variant is clearly better: the direction-invariant variant cannot distinguish a route from its reversal, which changes feasibility under time windows or precedence, and the gap widens accordingly. Because the two variants differ in band coverage ($D/2$ sine–cosine pairs vs. D cosines) as well as in direction sensitivity, the last row restricts the direction-invariant variant to the same $D/2$ harmonics; its gaps change by less than 0.1 percentage points, so the effect is attributable to direction sensitivity rather than to band coverage.

Q XPE, Route Membership, and Angular Overlap

What XPE does and does not encode. XPE is a function of the instance only. It does not change when the current solution changes, two nodes on different routes can have the same angle, and a node keeps its XPE when a move transfers it to another route. Route membership therefore enters the joint encoding only through IPE, whose phases are computed along the current routes, and through the attention of the backbone; XPE contributes an instance-level cue that correlates with membership to the extent that routes occupy separate angular sectors. We quantify this extent and compare XPE with signals that encode membership explicitly.

Table 14: Static angular cue vs. explicit route-membership signals under the DACT backbone (gap, ↓). The same-route bias adds a learnable per-head scalar to the attention logit of node pairs on the same route of the current solution; route pooling mean-pools node embeddings per route and broadcasts the result back to its nodes. Both are permutation-invariant over routes and are recomputed after every step.

Variant	CVRP-100	VRPTW-100
IPE only	1.48%	3.53%
IPE + route pooling (membership)	1.33%	3.24%
IPE + same-route bias (membership)	1.30%	3.18%
IPE + XPE (static angle, ours)	1.16%	2.97%
IPE + XPE + same-route bias	1.12%	2.92%

Angular mixing ρ of test instance (tercile)	Low	Mid	High
Mean ρ	0.11	0.18	0.27
Gap, IPE only	1.29%	1.47%	1.68%
Gap, IPE + XPE	0.88%	1.15%	1.45%
Gain of XPE (pp)	0.41	0.32	0.23

Entropy measurement of Section 3.1. Angles are discretized into $B = 16$ equal circular bins, and for each (problem, k) we compute $H(\theta \mid k)/\log B$ over the customers at within-route position k, pooled over the 1,000 instances; the marginal $H(\theta)/\log B$ is 0.998 on the uniform instances, so the mutual information between θ and k is at most $0.045 \log B$. This shows that the index says little about the angle; it does not show that the angle identifies the route, which is what the overlap statistic below measures.

Angular overlap. For a solution s we define the angular mixing rate $\rho(s) = \frac{1}{N}\sum_v \mathbb{1}[\mathrm{route}_s(\mathrm{nn}_\theta(v)) \neq \mathrm{route}_s(v)]$, where $\mathrm{nn}_\theta(v)$ is the customer with the smallest circular angular difference $|\theta_v - \theta_w|$ taken modulo 2π, so that the cut at $\pm\pi$ plays no role. $\rho = 0$ when every route occupies a contiguous angular sector, and ρ approaches $1 - 1/K$ for K angularly interleaved routes. On HGS solutions, ρ is 0.18 on CVRP-100, 0.24 on VRPTW-100, 0.31 on the low-capacity CVRP-500 shift, and 0.27 on the clustered shift. Along the Layer 1 search, ρ decreases from 0.62 on the initial solutions to 0.35 after 100 steps and 0.21 on the final solutions of IPE+XPE on CVRP-100, so XPE operates in a higher-overlap regime early in the search.

Gain of XPE versus overlap. The bottom of Table 14 splits the CVRP-100 test instances into terciles of ρ computed on their HGS solutions (333 or 334 instances each). The gain of XPE over IPE alone decreases from 0.41 to 0.23 percentage points from the lowest to the highest tercile, but remains positive; overlap limits, without removing, the benefit of the static angular cue.

Explicit membership signals. The top of Table 14 compares XPE with two signals that do encode the current route assignment and are invariant to route renumbering: a learnable same-route attention bias and a route-pooled embedding broadcast to the nodes of each route. Both improve on IPE alone, confirming that membership is useful, but both are weaker than XPE, and adding the same-route bias to IPE+XPE improves the gap only slightly. We read this as evidence that the benefit of XPE is not only a proxy for membership but also comes from the global angular layout around the depot; explicit membership signals require recomputation at every step and change the attention computation, so we do not adopt them in the main method. A learned route-ID embedding would not be invariant to route renumbering, and we therefore do not use it as a comparison.

R Variance Across Training Seeds

All numbers in Table 4, Table 5, and Table 6 are means over three independent training seeds, each evaluated on the same fixed 1,000-instance test set per (problem, size) configuration. We separate three sources of variability.

Table 15: Mean ± sample SD across three independent training seeds ($n = 3$) for our methods and the main baselines, complementing Table 4 (top) and Table 5 (bottom). The per-seed value is the average over the fixed 1,000-instance test set. Δ rows: mean per-instance paired difference $\mathrm{gap}(\text{IPE+XPE}) - \mathrm{gap}(\text{NDS})$ in percentage points (negative favors IPE+XPE; per-instance gaps first averaged over seeds), with a 95% bootstrap interval over instances, Holm-adjusted across the nine cells.

PE Method	CVRP-100		VRPTW-100		PDTSP-51	
	Obj.↓	Gap↓	Obj.↓	Gap↓	Obj.↓	Gap↓
SIN	16.45 ± 0.04	$5.58 \pm 0.26\%$	27.34 ± 0.07	$7.54 \pm 0.28\%$	7.27 ± 0.02	$5.95 \pm 0.31\%$
Laplacian PE	15.91 ± 0.03	$2.12 \pm 0.20\%$	26.55 ± 0.05	$4.43 \pm 0.21\%$	7.07 ± 0.02	$3.06 \pm 0.27\%$
SPD	15.86 ± 0.03	$1.80 \pm 0.18\%$	26.41 ± 0.05	$3.88 \pm 0.19\%$	7.05 ± 0.02	$2.74 \pm 0.25\%$
DACT CPE	15.95 ± 0.03	$2.37 \pm 0.20\%$	26.58 ± 0.05	$4.55 \pm 0.22\%$	7.05 ± 0.02	$2.84 \pm 0.24\%$
IPE	15.81 ± 0.03	$1.48 \pm 0.19\%$	26.32 ± 0.05	$3.53 \pm 0.20\%$	7.01 ± 0.02	$2.16 \pm 0.24\%$
XPE	15.84 ± 0.03	$1.67 \pm 0.21\%$	26.27 ± 0.04	$3.33 \pm 0.17\%$	7.02 ± 0.02	$2.30 \pm 0.26\%$
IPE + XPE	15.76 ± 0.02	$1.16 \pm 0.14\%$	26.18 ± 0.04	$2.97 \pm 0.15\%$	6.99 ± 0.01	$1.87 \pm 0.19\%$

Problem / Method		$N = 500$		$N = 1000$		$N = 2000$	
		Obj.↓	Gap↓	Obj.↓	Gap↓	Obj.↓	Gap↓
CVRP	NDS	36.57 ± 0.03	$-0.20 \pm 0.08\%$	41.11 ± 0.04	$-0.90 \pm 0.10\%$	56.00 ± 0.07	$-2.34 \pm 0.12\%$
	IPE+XPE	36.54 ± 0.03	$-0.30 \pm 0.08\%$	41.05 ± 0.04	$-1.11 \pm 0.10\%$	55.98 ± 0.07	$-2.45 \pm 0.12\%$
	Δ (pp)	-0.10	$[-0.13, -0.07]$	-0.21	$[-0.24, -0.18]$	-0.11	$[-0.15, -0.07]$
VRPTW	NDS	47.94 ± 0.05	$-2.17 \pm 0.10\%$	87.54 ± 0.09	$-3.14 \pm 0.10\%$	167.48 ± 0.20	$-3.50 \pm 0.12\%$
	IPE+XPE	47.90 ± 0.05	$-2.27 \pm 0.10\%$	87.45 ± 0.09	$-3.21 \pm 0.10\%$	166.80 ± 0.21	$-3.84 \pm 0.12\%$
	Δ (pp)	-0.10	$[-0.13, -0.07]$	-0.07	$[-0.10, -0.04]$	-0.34	$[-0.38, -0.30]$
PCVRP	NDS	43.12 ± 0.04	$-4.12 \pm 0.09\%$	80.99 ± 0.08	$-4.71 \pm 0.10\%$	158.09 ± 0.18	$-4.60 \pm 0.11\%$
	IPE+XPE	43.13 ± 0.04	$-4.09 \pm 0.09\%$	80.77 ± 0.08	$-4.87 \pm 0.09\%$	157.37 ± 0.19	$-4.94 \pm 0.11\%$
	Δ (pp)	$+0.03$	$[-0.00, +0.06]$	-0.16	$[-0.19, -0.13]$	-0.34	$[-0.38, -0.30]$

Training randomness. For each seed we first average the objective (and gap) over the test set, and then report the mean and sample standard deviation across the three seed-level values (Table 15), so the SD reflects training randomness rather than instance-to-instance variability. With $n = 3$ the SD estimate is itself noisy, and we use it to indicate the scale of training variability, not as a significance test.

Search randomness. A fixed search budget does not by itself fix search randomness. Every evaluation run uses the same evaluation random seed per test instance for all methods and training seeds, so that sampling noise is common across compared methods; on CVRP-500 and VRPTW-500, re-evaluating each IPE+XPE and NDS checkpoint with five different evaluation seeds changes the test-set-mean gap by at most 0.02 percentage points, well below the Layer 2 differences in Table 15.

Paired differences across test instances. Because every method is evaluated on the same instances, we compare methods through per-instance paired differences $\Delta = \mathrm{gap}(\text{IPE+XPE}) - \mathrm{gap}(\text{baseline})$ rather than through overlapping SDs. We apply this test where the margins are comparable to the seed SD, namely IPE+XPE against native NDS at every Layer 2 cell; the Δ rows of Table 15 report its mean and a 95% percentile bootstrap interval over instances, Holm-adjusted across the nine cells. The intervals are conditional on the trained models: they quantify test-instance variability, while training variability is reflected in the SDs. The interval excludes zero in eight of the nine cells; the exception is PCVRP-500, where the difference is $+0.03$ percentage points with an interval that includes zero. The Layer 2 improvements are practically small (0.07–0.34 percentage points) but come at no measurable runtime cost (Section G). In Layers 1 and 3, the differences to the strongest baseline are several times the seed SD (e.g., 0.64–0.91 percentage points against SPD in Layer 1), with one exception: on PDTSP-21, both N2S and HA-N2S are within 0.05 percentage points of LKH, and we make no claim of a difference there.

Table 16: Alternative PEs inside the NDS backbone (Layer 2). Each row replaces only the positional component of NDS and is trained and evaluated with the protocol of Table 5 (three training seeds, same wall-clock budget). Entries are gaps to the reference solver (↓). SPD and Laplacian PE are recomputed on the route graph after every search step, and this cost is included in the time budget. Best per column in **bold**.

PE in NDS	CVRP			VRPTW			PCVRP		
	500	1000	2000	500	1000	2000	500	1000	2000
Native (index)	−0.20%	−0.90%	−2.34%	−2.17%	−3.14%	−3.50%	**−4.12%**	−4.71%	−4.60%
SIN	−0.21%	−0.93%	−2.36%	−2.18%	−3.13%	−3.55%	−4.10%	−4.72%	−4.63%
Laplacian PE	−0.24%	−0.99%	−2.36%	−2.20%	−3.16%	−3.61%	−4.11%	−4.76%	−4.72%
SPD	−0.26%	−1.02%	−2.33%	−2.22%	−3.17%	−3.66%	−4.10%	−4.78%	−4.75%
IPE+XPE	**−0.30%**	**−1.11%**	**−2.45%**	**−2.27%**	**−3.21%**	**−3.84%**	−4.09%	**−4.87%**	**−4.94%**

S Alternative PEs at Scale

To test whether the Layer 2 gains reflect our encoding specifically or any replacement of the native index PE of NDS, we plug SIN, SPD, and Laplacian PE, the three strongest or most common alternatives of Layer 1, into NDS and train and evaluate them with the protocol of Table 5. All encodings share the same training budget, output width, and projection (Section D); SPD and Laplacian PE are recomputed on the route graph of the current solution after every search step, and this cost is part of the fixed wall-clock budget. Table 16 shows that every alternative improves on native NDS in most cells, with SPD the strongest, but IPE+XPE attains the lowest gap in eight of the nine cells; on PCVRP-500 all five encodings lie within 0.03 percentage points. At $N = 2000$, SPD and Laplacian PE lose part of their representational advantage to recomputation overhead, which reduces the number of completed search steps by 9% and 6% (Section G); IPE+XPE has no such trade-off. The same checkpoints are used for the zero-shot evaluations of Table 9.

T Extra Discussion

Rotation reference frame. The angle θ_v used in XPE is defined in the absolute coordinate frame of the instance. A global rotation of the instance by α shifts every θ_v by α, which acts on each band of XPE as a fixed rotation of its $(\sin, \cos)$ pair; the pairwise inner products of XPE vectors are therefore invariant, but the downstream feed-forward projection is not, so the network as a whole is not rotation-invariant by construction. We adopt the standard NCO practice of rotation augmentation during training (Kwon et al., 2020), which encourages, but does not guarantee, robustness to global rotations; for variants that additionally exhibit reflection symmetry (CVRP, TSP), we extend the augmentation to random reflections across an axis through the depot. At test time, the $8\times$ geometric augmentation of NDS evaluates each instance under rotations and reflections and keeps the best solution.

Frequency schedule. The XPE frequency schedule is intentionally low compared to NLP usage: the angular range $[-\pi, \pi)$ is fixed, and the first band $(\sin\theta_v, \cos\theta_v)$ already identifies every angle modulo 2π. Higher bands $2^k\theta_v$ make the encoding more sensitive to small angular differences, which helps separate neighboring routes when a near-optimal solution at $N \approx 100$ contains on the order of ten routes; the number of bands is not a hard limit on the number of distinguishable sectors. We use $K = 4$ throughout the main experiments; a single band is weaker (Table 11).

Scope of applicability. IPE and XPE share the same multi-frequency sinusoidal form and are concatenated to form a drop-in replacement for index-based PE in any encoder that consumes a complete (possibly suboptimal) solution. Our experiments cover three improvement/search backbones (Ma et al., 2021; 2022; Hottung et al., 2025b). Re-encoding construction methods (Drakulic et al., 2023; Luo et al., 2023) consume an ordered partial solution, but it is neither closed nor of known final length, so $\hat{d}_i = 2\pi d_i / d_L$ is undefined without future information; using the final route length would leak the solution. A construction variant would need, e.g., distance normalized by a problem-level scale, and we leave its design and validation open; one-shot autoregressive constructors (Kool et al.,

2019; Kwon et al., 2020) have no solution at encoding time. While θ_v used by XPE is by itself an instance-level quantity, the joint encoding $[\mathrm{IPE} \,\|\, \mathrm{XPE}]$ is solution-conditioned through the route assignment that defines d_i and d_L.

Asymmetric, non-Euclidean, and multi-depot settings. IPE only needs along-route edge costs and can use road-network or asymmetric costs directly; with asymmetric costs, however, the reversed route has cumulative cost different from $d_L - d_i$, so the reversal identities of Section 4.1 no longer hold and only the direction-aware variant is meaningful. XPE requires a planar embedding with a distinguished depot and is undefined for purely graph-defined instances without coordinates; multi-depot problems would require a depot assignment and one angular reference per depot. We have not evaluated these settings.

U Relation to Geometric and Global-Local Feature Designs

Our contribution is an input design and inductive bias for the positional channel of a fixed backbone, rather than a new solver architecture. A related line improves generalization by changing what information a solver aggregates and how. DGL (Xiao et al., 2025) restricts node selection to a dynamically adjusted local neighborhood, fuses it with global context, and trains with a replacement-based self-improvement scheme, targeting scale generalization of a construction-style policy. The two approaches differ in three respects. (i) *Source of information*: DGL's local and global views are derived from the instance and the current decision state of a construction policy; IPE is derived from the current complete solution, and XPE from a static depot-anchored angle. (ii) *Intervention*: DGL changes the architecture and the training procedure; we change only the PE module and keep architecture and training fixed, which is what allows the controlled comparisons of Section 5. (iii) *Cost*: IPE and XPE add $O(N)$ work per step, whereas dynamic neighborhood selection adds per-step neighborhood computation. The approaches are therefore complementary rather than competing; the simplest geometric inputs that a DGL-style design might provide to our backbones, such as depot-relative offsets, a single angular band, and raw along-route distance, are the controls of Table 11. A full system-level comparison with DGL would require matching data, hardware, and budget across different solver families and is beyond the scope of this study.