CONNECTED BY CONSTRUCTION: LEARNING TRACTABLE NEAR-TOUR MARGINALS FOR TRAVELING SALESMAN PROBLEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning-based methods for the traveling salesman problem (TSP) are often evaluated through the tours produced after decoding or search, but the learned object itself frequently lives in a surrogate space such as heatmaps, assignments, construction policies, or search-guidance scores. This obscures a basic question: which Hamiltonian structures can a model represent and learn tractably, and how does learning them directly lead to better tours? In this study, we directly answer this question by learning a structurally meaningful latent object for TSP, rather than leaving most of the Hamiltonian structure to the final decoding stage. Based on a connected-by-construction rooted 1-tree Gibbs family, we propose an end-to-end unsupervised learning pipeline called *C2TSP*. The pipeline learns residual edge perturbations from the penalty-free TSP cost through implicit differentiation. For structural correction, a smoothed Held–Karp layer restores expected degree balance, while certificate-guided sharpening further pushes the connected distribution toward more tour-like structures. Experiments show that C2TSP learns interpretable, tour-like structure, and ablations verify that edge perturbation and certificate-guided sharpening jointly improve both this structure and the tour cost. C2TSP also yields strong decoding performance, transfers zero-shot to metric non-Euclidean instances, and remains robust as a candidate source for the LKH heuristic.

1 INTRODUCTION

Machine learning methods have been increasingly used for combinatorial optimization because they can provide fast approximate solutions while maintaining competitive empirical performance. As a canonical combinatorial optimization problem, the traveling salesman problem (TSP) has received particular attention. A feasible TSP solution must satisfy both local and global structural constraints: every node has degree two, and the selected edges must form one connected cycle. However, many learning-based TSP methods do not represent this global connected edge structure directly in the learned object. Autoregressive construction models generate tours sequentially and enforce feasibility procedurally through masking and decoding rules (Vinyals et al., 2015; Bello et al., 2016; Kool et al., 2019; Kwon et al., 2020). Heatmap-based methods predict edge scores and then rely on repair, sampling, local search, or tree search to obtain a valid tour (Fu et al., 2021; Qiu et al., 2022; Min et al., 2023; Sun & Yang, 2023; Li et al., 2024). Assignment-space relaxations, such as Sinkhorn-based methods, provide smooth permutation-like objects but do not directly model the edge-adjacency structure that determines tour cost (Mena et al., 2018; Paulus et al., 2020; Min & Gomes, 2026). Hybrid methods such as NeuroLKH use learned signals to guide strong handcrafted heuristics (Xin et al., 2021). Although these methods have achieved strong empirical results with affordable inference time, two limitations remain: they do not provide a tractable global distribution over connected combinatorial objects, and the contribution of their learned representation can be difficult to attribute separately from downstream decoding or search (Xia et al., 2024; Böther et al., 2022). In contrast, we build a method whose contribution can be attributed at the level of the learned near-tour marginals: the representation can be evaluated directly through marginal structural metrics and compared under matched decoders, even though decoding is still required to produce feasible tours.

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.

We first propose a connected-by-construction representation based on the rooted 1-tree family. After fixing a root node, a rooted 1-tree consists of a spanning tree on the non-root nodes together with exactly two edges incident to the root. Therefore, every latent configuration is globally connected by construction and satisfies the root degree constraint exactly. The remaining structural defect is the non-root degree mismatch: a rooted 1-tree becomes a Hamiltonian cycle when every non-root node also has degree two. Unlike a Gibbs distribution over Hamiltonian cycles, whose partition function is intractable, the rooted 1-tree Gibbs family admits an exact factorization, which enables exact marginal computation and expected-cost training. This connects our method to the classical Held–Karp relaxation, where node penalties modify edge costs and minimum 1-trees provide strong TSP lower bounds (Held & Karp, 1970; 1971). Prior work has explored tree-based or Held–Karp-related structures for TSP from theoretical, heuristic, or learning perspectives (Parjadis et al., 2023; Gharan et al., 2011; Genova & Williamson, 2017). Our goal is to turn this connected structure into an end-to-end differentiable latent representation for learning near-tour marginals.

We then build an end-to-end unsupervised learning pipeline from this connected-by-construction rooted 1-tree latent family, called *C2TSP*. The model first predicts residual edge perturbations that tilt the rooted 1-tree Gibbs distribution toward low-cost structures guided by the expected TSP cost. To correct the remaining non-root degree defect, we introduce a smoothed Held–Karp equilibration layer built on the classical Held–Karp dual (Held & Karp, 1970). This layer solves for node-additive dual variables so that the expected degree of every non-root node equals two under the dual-modified rooted 1-tree distribution, and it is trained end-to-end by differentiating through the equilibrium map, following the general paradigm of implicit optimization layers (Amos & Kolter, 2017; Agrawal et al., 2019; Bai et al., 2019; Blondel et al., 2022). After equilibration, the residual error is no longer a connectivity failure, but the non-tour mass inside a connected, degree-balanced latent family. We control this residual defect using a certificate that upper-bounds non-tour mass by the total variance of the non-root degrees. A certificate-guided residual refinement then reduces this bound, followed by another re-equilibration step. Tours are finally decoded from the learned edge marginals.

The contributions of this paper are as follows.

- We introduce a tractable rooted 1-tree representation for TSP that ensures global connectivity by construction while admitting exact marginal computation for expected-cost training.
- We develop C2TSP, an end-to-end unsupervised learning pipeline in which a graph neural network (GNN) predicts residual edge perturbations, a smoothed Held–Karp equilibration layer restores expected degree balance, and a non-tour mass certificate guides residual refinement.
- We empirically verify that the learned marginals are tour-like, and that they remain effective under matched decoders, in zero-shot transfer, and as candidate sources for LKH.

Although the rooted 1-tree construction and its guarantees are specific to TSP, C2TSP suggests a broader template: enforce a key feasibility property with a tractable structural family, then correct the residual constraints. Demonstrating such transfer to other problems is left for future work.

2 METHODOLOGY

2.1 PROBLEM DEFINITION

Symmetric TSP and inference over tours. Let $G = (V, \mathcal{E})$ be a complete undirected graph on $V = \{1, \dots, n\}$ with edge costs $D \in \mathbb{R}_+^{|\mathcal{E}|}$, and let $\mathcal{H}$ be its Hamiltonian cycles. For any edge set S, write $x_S \in \{0,1\}^{|\mathcal{E}|}$ for its incidence vector. TSP minimizes $\langle D, x_H \rangle$ over $H \in \mathcal{H}$. The same optimum is obtained by minimizing expected cost over distributions on $\mathcal{H}$, and every minimizing distribution is supported on optimal tours. To obtain a distribution that is differentiable in the edge costs, we add an entropy regularizer, whose unique minimizer is the Gibbs distribution over tours, $q_\eta^{\mathcal{H}}(H) \propto \exp(-\langle \eta, x_H \rangle / \tau)$, where edge energies η are predicted from D and $\tau > 0$ is a temperature. Its expected tour cost would provide a label-free objective for concentrating probability on optimal tours. Evaluating that objective and its gradient, however, requires inference over exponentially many Hamiltonian cycles and is therefore intractable.

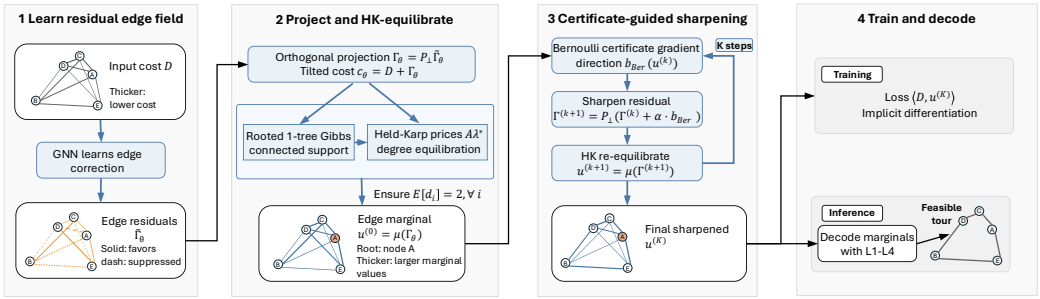

Figure 1: C2TSP workflow. A GNN maps costs to a projected residual edge field; exact rooted 1-tree inference and smoothed Held–Karp equilibration enforce connected support and expected degree balance; certificate-guided refinement sharpens and re-equilibrates the marginal before unsupervised cost training or test-time decoding.

Connected by construction: the rooted 1-tree surrogate. To make tour inference tractable, we seek a surrogate family that contains every Hamiltonian cycle and admits exact marginals and expected-cost gradients, together with a differentiable certificate bounding the resulting distribution's mass on non-tours. Note that a tour is precisely a connected spanning edge set with degree two at every node. Our construction therefore combines a rooted 1-tree representation that guarantees connectivity with the optimality conditions of a smoothed Held–Karp (HK) dual that enforce degree two in expectation. The analysis below establishes these requirements: exact inference (Lemma 2.1), differentiation through the HK layer (Theorem 2.2 and Appendix B.2), and certification (Theorem 2.3 and Proposition 2.5).

Fix a root $r \in V$ and write $\bar{V} := V \setminus \{r\}$. Let $\mathcal{U}_r$ denote the family of rooted 1-trees, namely spanning trees on $\bar{V}$ together with exactly two edges incident to r. For $U \in \mathcal{U}_r$, let $d_i(U)$ be the degree of node i in the subgraph U. Every $U \in \mathcal{U}_r$ is connected by construction and satisfies $d_r(U) = 2$. Moreover, $\mathcal{H} \subseteq \mathcal{U}_r$. For any edge parameter $\eta \in \mathbb{R}^{|\mathcal{E}|}$, we define the rooted 1-tree Gibbs law

$$q_\eta(U) = \frac{1}{Z_r(\eta)} \exp\left(-\frac{1}{\tau}\langle \eta, x_U \rangle\right), \qquad U \in \mathcal{U}_r, \tag{1}$$

and its edge marginal $\mu(\eta) := \mathbb{E}_{q_\eta}[x_U]$. Lemma 2.1 shows that the rooted 1-tree family admits an exactly computable partition function and marginal. The remaining task is to formalize the edge parameters η that preserve the degree-two structure and reduce the non-tour mass inside $\mathcal{U}_r$. The choice of root is not restrictive: since $\mathcal{H} \subseteq \mathcal{U}_r$ holds for every r, the root only selects the node whose degree is fixed exactly, and experiments show robustness to this choice (Appendix C.5).

Lemma 2.1 (Tractable rooted 1-tree marginal)**.** *The partition function $Z_r(\eta)$ in equation 1 factorizes into a weighted spanning-tree partition function on $\bar{V}$ and a two-edge root-selection normalizer. Consequently, $Z_r(\eta)$, the edge marginal $\mu(\eta)$, and the first and second degree moments are computable exactly in polynomial time; on a dense graph, their evaluation is $O(n^3)$.*

2.2 Held–Karp equilibration and structural-gap certificate

Given the 1-tree Gibbs family, the remaining Hamiltonian condition is to enforce degree two at the non-root nodes. Since the Gibbs law is parameterized by edge costs, the dual prices for the non-root degree constraints need to be represented as an edge-additive field. For $\lambda \in \mathbb{R}^{n-1}$ indexed by $\bar{V}$, define the node-to-edge lift $A : \mathbb{R}^{n-1} \to \mathbb{R}^{|\mathcal{E}|}$ by

$$(A\lambda)_{ij} = \begin{cases} \lambda_i + \lambda_j, & i, j \in \bar{V}, \\ \lambda_k, & \{i, j\} = \{r, k\},\ k \in \bar{V}. \end{cases} \tag{2}$$

Then, for every $U \in \mathcal{U}_r$,

$$\langle A\lambda, x_U \rangle = \sum_{i \in \bar{V}} \lambda_i d_i(U). \tag{3}$$

Thus $A\lambda$ is exactly the edge-additive representation of the non-root degree prices.

For a generic edge cost field $c \in \mathbb{R}^{|\mathcal{E}|}$, consider the entropy-regularized degree-correction problem for the 1-tree Gibbs family:

$$\min_{p \in \Delta(\mathcal{U}_r)} \quad \sum_{U \in \mathcal{U}_r} p(U)\langle c, x_U \rangle + \tau \sum_{U \in \mathcal{U}_r} p(U) \log p(U) \tag{4}$$
$$\text{s.t.} \quad \sum_{U \in \mathcal{U}_r} p(U) d_i(U) = 2, \qquad i \in \bar{V},$$

where $\Delta(\mathcal{U}_r)$ denotes the probability simplex over rooted 1-trees. This problem asks for the lowest free-energy distribution over connected rooted 1-trees whose expected non-root degrees match the Hamiltonian target. Substituting the lift identity from equation 3 into the Lagrangian collapses the degree multipliers into a tilted edge cost $c + A\lambda$, so that the minimizer stays in the rooted 1-tree Gibbs family, namely $q_{c+A\lambda}$, as stated below.

Theorem 2.2 (Smoothed Held–Karp equilibrium)**.** *The Lagrange dual of equation 4 is*

$$\max_{\lambda \in \mathbb{R}^{n-1}} \Phi_\tau(c, \lambda), \qquad \Phi_\tau(c, \lambda) := -\tau \log Z_r(c + A\lambda) - 2\mathbf{1}^\top \lambda. \tag{5}$$

Moreover,

$$\frac{\partial \Phi_\tau(c, \lambda)}{\partial \lambda_i} = \mathbb{E}_{q_{c+A\lambda}}[d_i(U)] - 2, \qquad i \in \bar{V}, \tag{6}$$
$$\frac{\partial^2 \Phi_\tau(c, \lambda)}{\partial \lambda_i \partial \lambda_j} = -\frac{1}{\tau} \operatorname{Cov}_{q_{c+A\lambda}}\big(d_i(U), d_j(U)\big), \qquad i, j \in \bar{V}. \tag{7}$$

Hence $\Phi_\tau(c, \lambda)$ is concave in λ, and any maximizer $\lambda^\star(c)$ satisfies $\mathbb{E}_{q_{c+A\lambda^\star(c)}}[d_i(U)] = 2$ for all $i \in \bar{V}$.

The proof is given in Appendix A.2. Theorem 2.2 shows that the degree multipliers enter each 1-tree energy through $\langle A\lambda, x_U \rangle - 2\mathbf{1}^\top \lambda$, and that optimizing over $p \in \Delta(\mathcal{U}_r)$ recovers the Gibbs law $q_{c+A\lambda}$. The HK layer is therefore the implicit map $c \mapsto \lambda^\star(c)$ defined by $\nabla_\lambda \Phi_\tau(c, \lambda^\star(c)) = 0$. Since $\sum_{i \in \bar{V}} d_i(U) = 2(n-1)$ on $\mathcal{U}_r$, Φ_τ is invariant to $\lambda \mapsto \lambda + t\mathbf{1}$, so $\lambda^\star(c)$ is unique only up to this shift; we fix the gauge on a complementary subspace (Appendix B.2).

This equilibration removes the degree defect of the connected surrogate in expectation. The remaining structural gap is the residual probability mass that the degree-balanced distribution assigns to rooted 1-trees outside Hamiltonian-cycle support. The next result certifies this residual gap.

Structural-gap certificate. Let $\mathcal{N} := \mathcal{U}_r \setminus \mathcal{H}$, and $\gamma(q) := q(\mathcal{N})$ denote the set of non-tour 1-trees and their mass, respectively. HK equilibration enforces degree balance only in expectation: $\mathbb{E}_q[d_i(U)] = 2$ for all $i \in \bar{V}$. But this does not force q to be supported on $\mathcal{H}$, as non-root degree may still fluctuate across rooted 1-trees but still having mean degree 2. The residual gap is nevertheless bounded by the degree variance, which can serve as a certificate that we can formally establish using same second-order statistic governing the HK Hessian in equation 7, as stated in the theorem below.

Theorem 2.3 (Certificate for residual non-tour mass)**.** *For any $q \in \Delta(\mathcal{U}_r)$ with $\mathbb{E}_q[d_i(U)] = 2$ for all $i \in \bar{V}$, the non-tour mass satisfies*

$$\gamma(q) \le \frac{1}{2} \sum_{i \in \bar{V}} \operatorname{Var}_q(d_i(U)). \tag{8}$$

For the HK-equilibrated Gibbs law $q^\star = q_{c+A\lambda^\star(c)}$, this becomes

$$\gamma(q^\star) \le \frac{1}{2} \sum_{i \in \bar{V}} \operatorname{Var}_{q^\star}(d_i(U)) = \frac{1}{2} \operatorname{tr} \operatorname{Cov}_{q^\star}(d(U), d(U)) = -\frac{\tau}{2} \operatorname{tr} \nabla^2_{\lambda\lambda} \Phi_\tau(c, \lambda^\star(c)). \tag{9}$$

The proof is given in Appendix A.3. Thus the same covariance that defines the HK Hessian in Theorem 2.2 also certifies the residual structural gap left after HK equilibration. As a consequence, the 1-tree Gibbs family provides not only tractable marginals but also the analytical structure needed to push them toward Hamiltonian cycles.

2.3 ORACLE RESIDUAL EDGE FIELD AND LEARNED RE-EQUILIBRATION

The certificate in Theorem 2.3 measures the structural gap left after HK equilibration. This gap cannot be removed by another node-price update. To see this, following equation 5, for any $\beta \in \mathbb{R}^{n-1}$,

$$\Phi_\tau(c + A\beta, \lambda) = \Phi_\tau(c, \lambda + \beta) + 2\mathbf{1}^\top \beta. \tag{10}$$

Thus a node-additive edge shift only reparametrizes the HK price and leaves the equilibrated Gibbs law and its edge marginals unchanged. Hence the residual edge correction is in general not unique and can be absorbed into the HK price. We need to select the identifiable representative by removing this node-price component and retaining the part orthogonal to all node-additive edge fields $A\lambda$.

To identify the oracle form of this residual, suppose the omitted tour-valid structure inside the rooted 1-tree family is represented at the marginal level by inequalities $Q\mu \le h$, where Q and h are exponential in size in general, due to the combinatorial structure of the problem. Now extending equation 4 with these constraints gives the entropy-regularized oracle recovery problem

$$\begin{aligned} \min_{p \in \Delta(\mathcal{U}_r)} \quad & \sum_{U \in \mathcal{U}_r} p(U)\langle D, x_U\rangle + \tau \sum_{U \in \mathcal{U}_r} p(U)\log p(U) \\ \text{s.t.} \quad & \mathbb{E}_p[d_i(U)] = 2, \qquad i \in \bar{V}, \qquad Q\,\mathbb{E}_p[x_U] \le h. \end{aligned} \tag{11}$$

Proposition 2.4 (Oracle orthogonal residual edge field)**.** *Let $(p^\star, \bar{\lambda}^\star, \nu^\star)$ satisfy the KKT conditions of equation 11, where $\bar{\lambda}^\star$ are the degree multipliers and $\nu^\star \ge 0$ are the multipliers of $Q\mu \le h$. Define the raw oracle edge field $\widetilde{\Gamma}^\star := Q^\top \nu^\star$. Let $\widetilde{\Gamma}^\star = A\beta^\star + \Gamma^\star$ with $\Gamma^\star \in \mathrm{Im}(A)^\perp$ be its decomposition into a node-additive component and an orthogonal residual component, and define the absorbed HK price $\lambda^\star := \bar{\lambda}^\star + \beta^\star$. Then $\lambda^\star$ is an HK price for the shifted cost field $D + \Gamma^\star$: $\lambda^\star \in \arg\max_\lambda \Phi_\tau(D + \Gamma^\star, \lambda)$.*

Consequently, the oracle re-equilibrated edge cost is $\eta^\star := D + \Gamma^\star + A\lambda^\star$.

The proof is given in Appendix A.4. Proposition 2.4 shows that the raw dual effect $Q^\top \nu^\star$ separates into a node-additive component $A\beta^\star$, which is absorbed into the HK price, and an orthogonal residual component $\Gamma^\star \in \mathrm{Im}(A)^\perp$, which is the identifiable oracle edge field. The corresponding rooted 1-tree Gibbs law $q_{\eta^\star}$ then recovers the oracle solution. Since equation 11 constrains only the marginal, $q_{\eta^\star}$ need not be supported on $\mathcal{H}$. The proposition thus identifies the form of the learnable object, not its value. Since the omitted constraints are intractable, we seek to learn a residual edge field Γ_θ and re-equilibrate: $\lambda^\star_\theta := \lambda^\star(D + \Gamma_\theta)$ and $\eta^\star_\theta := D + \Gamma_\theta + A\lambda^\star_\theta$. Thus in this study, we use learning to supply the residual edge geometry, while rooted 1-tree inference and HK equilibration constrain how this residual is shaped and learned. The learned law is degree-balanced after re-equilibration, and Theorem 2.3 certifies its remaining structural gap.

2.4 UNSUPERVISED LEARNING WITH CERTIFICATE-DIRECTED RE-EQUILIBRATION

With the learnable object identified as an orthogonal residual edge field, we now present the training pipeline, guided by the penalty-free TSP cost over training instances $\{D^m\}_{m=1}^M$:

$$\mathcal{L}_{\mathrm{TSP}}(\theta) = \frac{1}{M}\sum_{m=1}^{M} \langle D^m, u_\theta(D^m)\rangle, \tag{12}$$

where $u_\theta(D)$ is the edge marginal of the 1-tree Gibbs family produced by the forward map discussed below. This yields an unsupervised framework that *learns from the penalty-free TSP cost*, because the structural constraints are enforced in the forward map rather than as loss penalties. The forward process supplies the structure constraints, where HK recovery leads to expected degree balance, and certificate-directed sharpening further reduces a certified upper bound on residual non-tour mass. The backward propagation can be computed via implicit differentiation (Appendix B.2).

Canonical HK forward map. By Theorem 2.2, every edge input c induces an HK price $\lambda^\star(c)$, an equilibrated edge cost $c + A\lambda^\star(c)$, and a rooted 1-tree marginal $\mu(c + A\lambda^\star(c))$. As a result, for each instance D, the canonical forward map starts with a raw edge field $\widetilde{\Gamma}_\theta(D)$, followed by removing the node-price component as in Prop. 2.4:

$$\Gamma^{(0)}_\theta(D) = P_\perp \widetilde{\Gamma}_\theta(D), \qquad P_\perp := P_{\mathrm{Im}(A)^\perp}.$$

The projected residual then enters the HK layer: with $c_\theta^{(0)}(D) = D + \Gamma_\theta^{(0)}(D)$, we run T HK iterations to obtain $\lambda_{\theta,T}^{(0)}(D) \approx \lambda^\star(c_\theta^{(0)}(D))$ and set $\eta_\theta^{(0)}(D) = c_\theta^{(0)}(D) + A\lambda_{\theta,T}^{(0)}(D)$ and $u_\theta^{(0)}(D) = \mu(\eta_\theta^{(0)}(D))$, the marginal evaluated in the loss equation 12.

Bernoulli certificate and marginal-sharpening forward map. The certificate in Theorem 2.3 bounds the residual non-tour mass after HK equilibration, but directly differentiating the full covariance certificate would require higher-order covariance derivatives. In practice, one can instead use a Bernoulli upper certificate that depends only on the recovered edge marginal u at HK equilibrium:

$$\mathcal{C}_{\mathrm{Ber}}(u) := \sum_{e\in\mathcal{E}} w_e u_e(1-u_e), \qquad w_e := \frac{1}{2}\sum_{i\in\bar{V}} B_{ie},$$

where $B : \mathbb{R}^{|\mathcal{E}|} \to \mathbb{R}^{|\bar{V}|}$ is the non-root degree operator, and $Bx_U = d_{\bar{V}}(U) := (d_i(U))_{i\in\bar{V}}$. With the lift A defined in equation 2, we have $A = B^\top$. Thus $w_e = 1$ for non-root edges and $w_e = \frac{1}{2}$ for root edges. Its marginal derivative is

$$b_{\mathrm{Ber}}(u) := \nabla_u \mathcal{C}_{\mathrm{Ber}}(u) = w \odot (1-2u). \tag{13}$$

Since η is an edge-cost parameter, a positive step in this direction sharpens the marginal by penalizing low-probability edges and rewarding high-probability edges. We step along $g_{\mathrm{Ber}} := P_\perp b_{\mathrm{Ber}}$, which stays in $\mathrm{Im}(A)^\perp$ and, by equation 10, gives the same re-equilibrated law.

Proposition 2.5 (Bernoulli certificate descent)**.** *Let q_η be an HK-equilibrated rooted 1-tree Gibbs law and let $u = \mu(\eta)$. Then $\mathcal{C}_{\mathrm{Ber}}(u)$ is a valid non-tour certificate $\gamma(q_\eta) \le \mathcal{C}_{\mathrm{Ber}}(u)$. Moreover, for any edge input c at which the HK map is differentiable, let $\eta(c) = c + A\lambda^\star(c)$ and $u(c) = \mu(\eta(c))$. For $c_\alpha = c + \alpha g_{\mathrm{Ber}}(u(c))$ following equation 13 and $u_\alpha = \mu(c_\alpha + A\lambda^\star(c_\alpha))$, we have, for all sufficiently small $\alpha > 0$, $\mathcal{C}_{\mathrm{Ber}}(u_\alpha) < \mathcal{C}_{\mathrm{Ber}}(u(c))$, whenever the re-equilibrated sharpening direction is nonzero.*

Proposition 2.5 shows that the projected marginal derivative in equation 13 is a descent direction for the certified upper bound $\mathcal{C}_{\mathrm{Ber}}$ on the non-tour mass. This directly motivates a deterministic sharpening map applied after the canonical HK forward pass. Starting from $(\Gamma_\theta^{(0)}, u_\theta^{(0)})$, for $k = 0, \dots, K-1$, with a fixed step $\alpha > 0$,

$$\Gamma_\theta^{(k+1)} = \Gamma_\theta^{(k)} + \alpha g_{\mathrm{Ber}}(u_\theta^{(k)}), \quad \eta_\theta^{(k+1)} = c_\theta^{(k+1)} + A\lambda^\star(c_\theta^{(k+1)}), \quad u_\theta^{(k+1)} = \mu(\eta_\theta^{(k+1)}),$$

where $c_\theta^{(k+1)} = D + \Gamma_\theta^{(k+1)}$ is re-equilibrated at every step, which supplies the final marginal $u_\theta(D) := u_\theta^{(K)}(D)$ to equation 12. When $K = 0$, this reduces to the canonical HK forward map. The full workflow of our method is shown in Figure 1.

3 Experiments

Our experiments have three parts. We first examine the structure of the learned latent representation. We then compare decoding performance against representative neural baselines, DIFUSCO (Sun & Yang, 2023), DIMES (Qiu et al., 2022), Fast-T2T (Li et al., 2024), and UTSP (Min et al., 2023), on Euclidean and metric non-Euclidean instances. Finally, we connect each method to the LKH solver, with the hybrid NeuroLKH (Xin et al., 2021) as an additional baseline, to assess the robustness of C2TSP.

The main benchmark consists of 2-D symmetric Euclidean TSP instances with node coordinates sampled uniformly from the unit square, following the standard protocol of prior neural TSP work (Fu et al., 2021); we denote the dataset with n nodes by TSP-n. Reference optimal tours are computed with Concorde (Applegate et al., 2011) for $n \le 1000$, and LKH solutions serve as references for $n = 2000$ in Table 4. For the structural analysis, C2TSP is trained on TSP-50; for the performance comparison, all models are trained on TSP-100 and evaluated zero-shot at all other sizes. All models are trained under an equal training budget, with the same number of training instances and epochs. Training details are given in Appendix B. The metric non-Euclidean dataset used to test generalization is described in Appendix C. The decoding and LKH-integration settings are as follows.

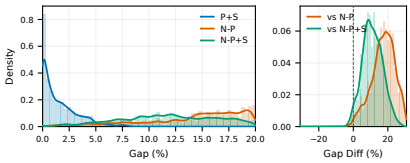

Figure 2: TSP-50 ablation. Left: gap distributions of P+S (the full pipeline), N-P+S, and N-P. Right: instance-wise gap differences to P+S.

Table 1: TSP-50 ablation of structural and decoding metrics. Top2: share of each node's incident marginal mass on its two largest edges; Cov. (Cost Cov.): unweighted (cost-weighted) recovery of Concorde-tour edges.

Method	Gap (%)↓	Top2↑	Cov.↑	Cost Cov.↑
P + S	**1.55**	**0.9000**	**0.8751**	**0.8357**
P	2.21	0.7438	0.8442	0.8006
N-P	19.25	0.1507	0.5845	0.5105
N-P + S	12.74	0.1923	0.6847	0.6046

Decoding. The trained network produces symmetric edge marginals from which a tour is recovered without an external solver. **L1** greedily grows edges in descending marginal order while avoiding subtours. **L2**$\times k$ applies k 2-opt improvement sweeps to L1. **L3** returns the best greedy tour obtained from a fixed budget of Gumbel perturbations on the final edge cost matrix. **L4** repeatedly extracts a maximum a posteriori (MAP) rooted 1-tree from perturbed model costs, repairs its residual degree defects, and returns the lowest-cost tour; Appendix D gives the algorithm.

Connecting to LKH. Each learning method is used as a candidate-and-initial-tour oracle for LKH-3, the current extension of the Lin–Kernighan–Helsgaun heuristic (Helsgaun, 2000; 2017). There are five integration levels with progressively more learned signal: **H0** is vanilla LKH-3; **H1** supplies the L1 tour as an initial tour; **H2** supplies the top-5 candidates per node ranked by the final edge marginal; **H3** supplies the score LKH uses to initialize its candidate ranking, which only C2TSP and NeuroLKH can provide; for C2TSP, this score is a 1:1 fusion of the final edge marginal and the final equilibrated cost; and **H4** combines H1 with H2 (H3 for C2TSP).

3.1 Structure of the learned marginals

We write P for edge perturbation, S for sharpening, and N-P for no perturbation. Figure 2 shows that removing edge perturbation substantially degrades decoded performance: the average gap increases from 1.55% to 12.74%, with many long-tail instances exceeding 20%. Its instance-wise panel further shows that the full pipeline improves over the no-perturbation variants on almost all test instances. Table 1 explains this gap from a structural perspective. Without learned perturbation, the smoothed rooted 1-tree distribution has weak edge concentration and low recovery of optimal tour edges. Sharpening alone improves these metrics, but remains far from the learned variants. In contrast, perturbation-only training already produces substantially more tour-like marginals and much better decoded tours. Figure 3 (left) shows a complementary training view. Without sharpening, the relaxed rooted 1-tree objective converges quickly but the decoded tour quality remains limited, suggesting that low relaxed cost alone is not sufficient. Sharpening helps translate the learned connected structure into better decoded tours. A representative TSP-50 instance (Figure 4, Appendix C.9) further shows that edge perturbation moves low-cost mass toward optimal-tour edges and sharpening suppresses the remaining nonoptimal ambiguity, shaping the 1-tree toward the optimal tour.

Figure 3 (middle and right) provides further evidence. Tours read directly from the C2TSP marginal by `argmax` recover more optimal-tour edges than those of every baseline, as its coverage CDF lies to the right of all others. Under 2-opt, C2TSP starts from its MAP tour at a lower gap and converges to a lower plateau than every baseline. Both panels indicate that the learned marginal already carries tour-like structure before search.

3.2 Decoding performance and non-Euclidean transfer

Table 2 reports optimality gap and per-instance wall time at every decode level. C2TSP achieves the lowest optimality gap in 20 of the 25 (size, decoder-level) cells, and is uniformly best at the heaviest local-search budget (L2$\times$100). For $n \leq 200$, the advantage is largest precisely where the decoder is weakest, indicating the rooted 1-tree distribution already encodes tour-like structure before any

Table 2: Decoder results on Euclidean TSP. Optimality gap (%) and per-instance wall time in seconds (in parentheses).

Method	Level	TSP-50	TSP-100	TSP-200	TSP-500	TSP-1000
C2TSP	L1	**6.12** (0.14)	**9.44** (0.15)	**12.71** (0.20)	17.73 (0.97)	21.98 (7.34)
	L2 ×1	**3.73** (0.14)	**7.03** (0.15)	**10.32** (0.19)	**14.97** (0.93)	19.77 (7.19)
	L2 ×10	**1.44** (0.14)	**1.97** (0.15)	**3.25** (0.20)	**7.84** (0.94)	13.06 (7.19)
	L2 ×100	**1.43** (0.14)	**1.90** (0.15)	**2.21** (0.20)	**2.75** (0.94)	**3.83** (7.21)
	L3	**3.09** (0.14)	**5.11** (0.17)	**7.97**(0.29)	**13.54** (1.61)	20.35 (10.08)
	L4	2.36 (0.14)	4.83 (0.17)	8.74 (0.29)	15.38 (1.63)	21.96 (11.98)
DIFUSCO	L1	17.24 (0.07)	22.60 (0.35)	36.34 (0.96)	82.08 (5.97)	141.05 (24.18)
	L2 ×1	12.93 (0.07)	19.07 (0.35)	33.92 (0.96)	80.56 (5.94)	139.96 (24.06)
	L2 ×10	2.71 (0.07)	6.78 (0.35)	22.43 (0.96)	72.08 (5.94)	133.36 (24.06)
	L2 ×100	2.48 (0.07)	3.73 (0.35)	5.74 (0.98)	32.86 (5.96)	94.93 (24.09)
	L3	9.63 (0.07)	15.41 (0.37)	36.39 (1.05)	97.00 (6.50)	172.62 (26.40)
Fast-T2T	L1	17.63 (<.01)	14.31 (0.01)	21.36 (0.02)	41.20 (0.15)	74.62 (0.62)
	L2 ×1	12.55 (<.01)	11.04 (0.01)	18.60 (0.02)	39.23 (0.12)	73.20 (0.49)
	L2 ×10	2.60 (<.01)	2.80 (0.01)	9.01 (0.02)	31.84 (0.12)	67.50 (0.49)
	L2 ×100	2.46 (<.01)	2.44 (0.01)	4.39 (0.03)	10.47 (0.14)	42.72 (0.52)
	L3	8.46 (0.01)	7.77 (0.03)	24.95 (0.11)	74.42 (0.70)	136.48 (2.96)
DIMES	L1	13.72 (<.01)	16.08 (<.01)	16.18 (0.01)	**16.23** (0.04)	**15.84** (0.16)
	L2 ×1	10.05 (<.01)	13.29 (<.01)	14.28 (<.01)	**14.97** (0.01)	**14.96** (0.03)
	L2 ×10	2.66 (<.01)	4.28 (<.01)	6.61 (<.01)	9.67 (0.01)	**11.23** (0.04)
	L2 ×100	2.55 (<.01)	3.25 (<.01)	3.54 (0.01)	3.73 (0.02)	3.95 (0.07)
	L3	7.87 (0.01)	9.66 (0.02)	11.58 (0.09)	15.12 (0.57)	**19.38** (2.41)
UTSP	L1	16.56 (<.01)	20.59 (<.01)	24.43 (<.01)	31.90 (0.02)	39.21 (0.10)
	L2 ×1	12.54 (<.01)	17.62 (<.01)	22.45 (<.01)	30.80 (<.01)	38.54 (<.01)
	L2 ×10	4.28 (<.01)	8.40 (<.01)	14.57 (<.01)	25.35 (<.01)	34.77 (0.01)
	L2 ×100	4.02 (<.01)	5.66 (<.01)	7.19 (0.01)	9.98 (0.02)	18.88 (0.04)
	L3	60.95 (0.01)	125.21 (0.02)	220.61 (0.08)	414.43 (0.49)	633.80 (2.03)

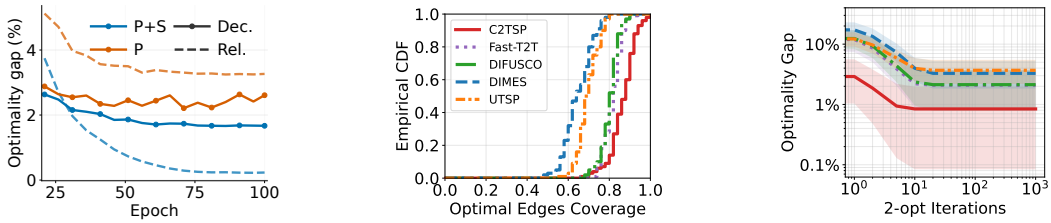

Figure 3: **Left**: training dynamics on TSP-50 of P and P+S; solid ("Dec.") and dashed ("Rel.") lines give the gap of the decoded tour and of the continuous relaxation. **Middle** and **Right**: models trained on TSP-100, evaluated on TSP-50; the middle panel shows the CDF of optimal-edge coverage of `argmax` tours over the edge scores (no search), and the right panel the gap versus 2-opt iterations, where baselines start from greedy heatmap tours and C2TSP from MAP decoding.

post-processing. The remaining best cells go to DIMES at $n \geq 500$ under low-search regimes, where its near-scale-invariant heatmap edges out our marginal-based decoders. Within C2TSP, the MAP decoder (L4) is competitive with L2×1 at small n but degrades at $n \geq 500$, indicating that the soft marginal carries mass beyond what the single MAP 1-tree retains.

Zero-shot metric non-Euclidean transfer. Dataset A follows random shortest-path metric constructions (Bringmann et al., 2015; Klootwijk et al., 2021). Its cost matrix D is metric but generally not Euclidean. All methods use their TSP-100 checkpoints without fine-tuning and the same L2×100 budget. C2TSP stays at 1.4–2.1% at every size, while every baseline exceeds 7% at $n = 100$ and 184% at $n = 1000$ (Table 3). This is consistent with the design: every C2TSP component operates directly on the edge costs D, whereas the baselines here receive only node coordinates, not D. More transfer results are given in Appendix C.

Table 3: Zero-shot transfer to non-Euclidean Dataset A (L2×100): gap (%, mean±std.) and time (s).

Method	$n = 100$	$n = 200$	$n = 500$	$n = 1000$
C2TSP	**1.41**±0.63 (0.0)	**1.52**±0.47 (0.1)	**1.93**±0.42 (0.3)	**2.05**±0.22 (2.0)
DIFUSCO	8.03±2.21 (0.3)	20.56±2.34 (0.7)	108.84±4.95 (5.7)	185.14±5.12 (15.9)
Fast-T2T	7.67±2.04 (0.0)	20.51±2.55 (0.0)	108.06±4.91 (0.2)	184.33±5.60 (0.5)
DIMES	7.14±2.29 (0.0)	20.39±2.21 (0.0)	107.67±5.72 (0.0)	184.31±5.90 (0.2)
UTSP	7.31±1.68 (0.0)	20.39±2.80 (0.0)	107.97±4.63 (0.0)	184.19±5.29 (0.1)

Table 4: LKH-integration results. Optimality gap in basis points (‰) and median LKH wall time per instance (s); OOM: out of memory.

Level	Method	TSP-100	TSP-200	TSP-500	TSP-1000	TSP-2000
H0	LKH-3	0.38 (0.11)	0.22 (0.49)	0.36 (2.32)	0.39 (7.49)	0.02 (31.23)
H1	C2TSP	**0.18** (0.07)	0.41 (0.29)	0.20 (2.49)	0.46 (5.89)	**-0.07** (26.73)
	DIFUSCO	0.29 (0.07)	**0.34** (0.28)	0.19 (2.54)	0.40 (7.98)	OOM
	Fast-T2T	0.32 (0.08)	0.64 (0.39)	0.23 (1.99)	0.39 (7.24)	0.06 (27.93)
	DIMES	0.44 (0.10)	0.73 (0.32)	0.19 (1.84)	**0.29** (6.39)	0.24 (21.87)
	UTSP	0.27 (0.11)	0.61 (0.27)	**0.16** (2.16)	0.41 (6.46)	0.12 (24.73)
H2	C2TSP	0.11 (0.07)	**0.13** (0.30)	**0.20** (2.82)	**0.53** (8.70)	**0.71** (37.77)
	DIFUSCO	0.88 (0.08)	4.91 (0.22)	50.14 (1.87)	130.10 (5.57)	OOM
	Fast-T2T	**0.02** (0.09)	0.76 (0.41)	10.99 (1.99)	36.46 (6.40)	80.65 (20.75)
	DIMES	3.24 (0.12)	6.18 (0.29)	8.27 (1.80)	9.21 (6.77)	6.56 (24.03)
	UTSP	145.13 (0.09)	211.09 (0.15)	217.26 (1.57)	185.35 (6.11)	142.49 (19.77)
H3	C2TSP	**0.11** (0.06)	**0.04** (0.29)	**0.18** (2.49)	**0.52** (9.20)	**0.56** (37.23)
	NeuroLKH	0.63 (0.06)	0.53 (0.21)	23.11 (1.01)	38.38 (4.82)	40.66 (27.04)
H4	C2TSP	0.11 (0.07)	**0.01** (0.30)	**0.20** (2.25)	**0.58** (8.41)	**0.64** (36.87)
	DIFUSCO	1.01 (0.08)	2.50 (0.24)	50.53 (1.76)	135.08 (5.50)	OOM
	Fast-T2T	**0.04** (0.09)	0.87 (0.44)	9.16 (2.07)	34.85 (6.59)	82.91 (21.11)
	DIMES	3.57 (0.12)	4.11 (0.32)	4.33 (1.95)	4.23 (7.66)	3.55 (27.26)
	UTSP	65.56 (0.10)	105.35 (0.15)	131.00 (1.57)	118.76 (6.41)	102.62 (21.77)

3.3 Robustness under LKH integration

Table 4 tests whether the learned marginal stays useful when it drives LKH, reporting gap and median LKH wall time. At H1, every method stays within $\sim 0.5\,‰$ of vanilla LKH-3 (H0): a learned initial tour is largely uninformative, because LKH's local search quickly escapes any mediocre start. The picture inverts at H2 and H4, where the heatmap drives LKH's candidate machinery, and DIFUSCO, DIMES, and Fast-T2T degrade by one to three orders of magnitude at $n \geq 500$. At $n \geq 500$, C2TSP is the only learned model that survives this deeper integration, remaining within $\sim 1\,‰$ of vanilla LKH at every size.

4 Conclusion

We introduced C2TSP, a differentiable pipeline built on rooted 1-trees: the support is connected by construction, the marginals are exactly computable, and a smoothed Held–Karp equilibration layer corrects the remaining degree defects. The model learns residual edge perturbations from the expected cost of this connected relaxation, while certificate-guided sharpening reduces an upper bound on residual non-tour mass. In this way, C2TSP shifts the learned object from unconstrained edge scores to a tractable near-tour marginal with explicit structural meaning. Empirically, under an equal training budget, the learned marginal yields strong pure-decoding performance and stays close to vanilla LKH when used as its candidate source, where several heatmap-based baselines degrade at larger sizes. The main limitation is the surrogate itself: connectivity is exact, but degree balance holds only in expectation, so residual degree fluctuations are reduced rather than eliminated. Tighter certificate-directed updates, richer tractable connected families, and scalable approximations to exact marginal computation are natural next steps.

AI USE STATEMENT

In this work, we used generative AI tools to assist in formulating mathematical claims, identifying ingredients useful for proving mathematical claims, and writing and refining proofs. Generative AI tools were also used to assist with software coding and debugging. Where applicable, AI-assisted code was used to implement components of the proposed methods and was subsequently reviewed and tested by the authors.

We did not use generative AI tools to generate synthetic datasets, develop the theoretical or conceptual framework of the work, propose or refine research hypotheses, design the research methodology or experiments, translate the manuscript, or clean or reformat datasets. Generative AI tools were not used to perform qualitative or thematic data analysis. The interpretation and conclusions drawn from the experimental results were made by the authors.

Additionally, we used generative AI tools to edit and polish the manuscript for grammar, clarity, and readability, including revisions to the textual presentation of experimental results. These edits were made based on analyses and interpretations provided by the authors and did not independently determine the interpretation of the results.

All AI-assisted mathematical claims, proofs, code, and manuscript text were reviewed and verified by the authors. The authors take responsibility for the final content of this work, including all claims, proofs, code, analyses, and text produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Complete proofs of Lemma 2.1, Theorems 2.2 and 2.3, and Propositions 2.4 and 2.5 are given in Appendix A. The training procedure, implicit differentiation of the HK layer, and hardware and software environment are described in Appendix B (Algorithm 1); the structured MAP decoder is given in Appendix D. Data generation for random Euclidean instances and for the non-Euclidean benchmarks is described in Section 3 and Appendix C. Anonymized code is available at `https://anonymous.4open.science/r/C2TSP-EF65`.

REFERENCES

Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J. Zico Kolter. Differentiable convex optimization layers. In *Advances in Neural Information Processing Systems*, pp. 9558–9570, 2019.

Brandon Amos and J. Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 136–145, 2017.

David L Applegate, Robert E Bixby, Vašek Chvátal, and William J Cook. The traveling salesman problem: a computational study. In *The traveling salesman problem*. Princeton university press, 2011.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. Deep equilibrium models. In *Advances in Neural Information Processing Systems*, pp. 688–699, 2019.

Pouya Baniasadi, Vladimir Ejov, Michael Haythorpe, and Serguei Rossomakhine. A new benchmark set for traveling salesman problem and hamiltonian cycle problem. *arXiv preprint arXiv:1806.09285*, 2018.

Irwan Bello, Hieu Pham, Quoc V Le, Mohammad Norouzi, and Samy Bengio. Neural combinatorial optimization with reinforcement learning. *arXiv preprint arXiv:1611.09940*, 2016.

Mathieu Blondel, Quentin Berthet, Marco Cuturi, Roy Frostig, Stephan Hoyer, Felipe Llinares-López, Fabian Pedregosa, and Jean-Philippe Vert. Efficient and modular implicit differentiation. In *Advances in Neural Information Processing Systems*, 2022.

Julius Borcea, Petter Brändén, and Thomas Liggett. Negative dependence and the geometry of polynomials. *Journal of the American Mathematical Society*, 22(2):521–567, 2009.

Maximilian Böther, Otto Kißig, Martin Taraz, Sarel Cohen, Karen Seidel, and Tobias Friedrich. What's wrong with deep learning in tree search for combinatorial optimization. *arXiv preprint arXiv:2201.10494*, 2022.

Karl Bringmann, Christian Engels, Bodo Manthey, and BV Raghavendra Rao. Random shortest paths: Non-euclidean instances for metric optimization problems. *Algorithmica*, 73(1):42–62, 2015.

Jan De Leeuw. Applications of convex analysis to multidimensional scaling. 2005.

Tomás Feder and Milena Mihail. Balanced matroids. In *Proceedings of the twenty-fourth annual ACM symposium on Theory of computing*, pp. 26–38, 1992.

Zhang-Hua Fu, Kai-Bin Qiu, and Hongyuan Zha. Generalize a small pre-trained model to arbitrarily large tsp instances. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 7474–7482, 2021.

Kyle Genova and David P Williamson. An experimental evaluation of the best-of-many christofides' algorithm for the traveling salesman problem. *Algorithmica*, 78(4):1109–1130, 2017.

Shayan Oveis Gharan, Amin Saberi, and Mohit Singh. A randomized rounding approach to the traveling salesman problem. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pp. 550–559. IEEE, 2011.

John C Gower. Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika*, 53(3-4):325–338, 1966.

Michael Held and Richard M Karp. The traveling-salesman problem and minimum spanning trees. *Operations research*, 18(6):1138–1162, 1970.

Michael Held and Richard M Karp. The traveling-salesman problem and minimum spanning trees: Part ii. *Mathematical programming*, 1(1):6–25, 1971.

Keld Helsgaun. An effective implementation of the lin–kernighan traveling salesman heuristic. *European journal of operational research*, 126(1):106–130, 2000.

Keld Helsgaun. An extension of the lin-kernighan-helsgaun tsp solver for constrained traveling salesman and vehicle routing problems. *Roskilde: Roskilde University*, 12(966-980):1, 2017.

Stefan Klootwijk, Bodo Manthey, and Sander K Visser. Probabilistic analysis of optimization problems on generalized random shortest path metrics. *Theoretical computer science*, 866:107–122, 2021.

Wouter Kool, Herke van Hoof, and Max Welling. Attention, learn to solve routing problems! In *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. In *Advances in Neural Information Processing Systems*, 2020.

Yang Li, Jinpei Guo, Runzhong Wang, Hongyuan Zha, and Junchi Yan. Fast t2t: Optimization consistency speeds up diffusion-based training-to-testing solving for combinatorial optimization. *Advances in Neural Information Processing Systems*, 37:30179–30206, 2024.

Gonzalo E. Mena, David Belanger, Scott W. Linderman, and Jasper Snoek. Learning latent permutations with gumbel-sinkhorn networks. In *International Conference on Learning Representations*, 2018.

Yimeng Min and Carla P Gomes. Graph neural networks are heuristics. *arXiv preprint arXiv:2601.13465*, 2026.

Yimeng Min, Yiwei Bai, and Carla P. Gomes. Unsupervised learning for solving the travelling salesman problem. In *Advances in Neural Information Processing Systems*, 2023.

Augustin Parjadis, Quentin Cappart, Bistra Dilkina, Aaron Ferber, and Louis-Martin Rousseau. Learning lagrangian multipliers for the travelling salesman problem. *arXiv preprint arXiv:2312.14836*, 2023.

Max Paulus, Dami Choi, Daniel Tarlow, Andreas Krause, and Chris J Maddison. Gradient estimation with stochastic softmax tricks. *Advances in Neural Information Processing Systems*, 33:5691–5704, 2020.

Ruizhong Qiu, Zhiqing Sun, and Yiming Yang. Dimes: A differentiable meta solver for combinatorial optimization problems. *Advances in Neural Information Processing Systems*, 35:25531–25546, 2022.

Gerhard Reinelt. Tsplib—a traveling salesman problem library. *ORSA journal on computing*, 3(4): 376–384, 1991.

Zhiqing Sun and Yiming Yang. Difusco: Graph-based diffusion solvers for combinatorial optimization. *Advances in neural information processing systems*, 36:3706–3731, 2023.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Yifan Xia, Xianliang Yang, Zichuan Liu, Zhihao Liu, Lei Song, and Jiang Bian. Position: Rethinking post-hoc search-based neural approaches for solving large-scale traveling salesman problems. *arXiv preprint arXiv:2406.03503*, 2024.

Liang Xin, Wen Song, Zhiguang Cao, and Jie Zhang. Neurolkh: Combining deep learning model with lin-kernighan-helsgaun heuristic for solving the traveling salesman problem. In *Advances in Neural Information Processing Systems*, 2021.

A PROOFS

A.1 PROOF OF LEMMA 2.1

Proof. Let $r \in V$ be the fixed root and $\bar{V} = V \setminus \{r\}$. For non-root edges $\{i, j\} \subseteq \bar{V}$, define

$$w_{ij}^{\eta} = \exp(-\eta_{ij}/\tau)\,,$$

and for root edges define

$$a_i = \exp(-\eta_{ri}/\tau)\,, \qquad i \in \bar{V}.$$

Every rooted 1-tree $U \in \mathcal{U}_r$ decomposes uniquely into a spanning tree T on $\bar{V}$ and an unordered pair of distinct root neighbors $\{i, j\} \subseteq \bar{V}$. Therefore,

$$\begin{aligned} Z_r(\eta) &= \sum_T \sum_{i<j} \exp\left(-\frac{1}{\tau}\Big[\sum_{e\in T} \eta_e + \eta_{ri} + \eta_{rj}\Big]\right) \\ &= \left(\sum_T \prod_{e\in T} w_e^{\eta}\right)\left(\sum_{i<j,\ i,j\in\bar{V}} a_i a_j\right). \end{aligned} \tag{14}$$

By the matrix-tree theorem,

$$\sum_T \prod_{e\in T} w_e^{\eta} = \det L^{\sharp}(W),$$

where $W = (w_{ij}^{\eta})$ collects the non-root edge weights and $L^{\sharp}(W)$ is any principal cofactor (reduced Laplacian) of the weighted Laplacian on the non-root graph. Hence,

$$Z_r(\eta) = \det L^{\sharp}(W) \sum_{i<j,\ i,j\in\bar{V}} a_i a_j.$$

This proves the partition factorization.

The same factorization gives exact edge marginals. For non-root edges, the spanning-tree marginal identity gives

$$\mu_{ij}(\eta) = w_{ij}^{\eta} R_{ij}^{\mathrm{eff}}(W), \qquad \{i, j\} \subseteq \bar{V},$$

where $R_{ij}^{\mathrm{eff}}(W)$ is the effective resistance between i and j in the weighted non-root graph. For root edges, the marginal is the probability that i belongs to the sampled unordered pair of root neighbors:

$$\mu_{ri}(\eta) = \frac{a_i \sum_{j\in\bar{V},\ j\neq i} a_j}{\sum_{k<\ell,\ k,\ell\in\bar{V}} a_k a_\ell}, \qquad i \in \bar{V}.$$

The computational cost is dominated by one Laplacian factorization on the non-root graph, so the partition function, edge marginals, and degree moments are computable exactly. □

A.2 PROOF OF THEOREM 2.2

Proof. For a fixed edge cost field $c \in \mathbb{R}^{|\mathcal{E}|}$ and node prices $\lambda \in \mathbb{R}^{n-1}$, define

$$E_{c,\lambda}(U) := \langle c + A\lambda, x_U \rangle.$$

By the node-to-edge lift identity,

$$\langle A\lambda, x_U \rangle = \sum_{i\in\bar{V}} \lambda_i d_i(U).$$

Therefore the Lagrangian of equation 4, up to the simplex constraint $p \in \Delta(\mathcal{U}_r)$, is

$$\mathcal{L}(p, \lambda) = \sum_{U\in\mathcal{U}_r} p(U) E_{c,\lambda}(U) + \tau \sum_{U\in\mathcal{U}_r} p(U) \log p(U) - 2\mathbf{1}^{\top}\lambda.$$

For fixed λ, minimizing over $p \in \Delta(\mathcal{U}_r)$ gives the Gibbs distribution

$$p_\lambda(U) = \frac{1}{Z_r(c + A\lambda)} \exp\left(-\frac{1}{\tau} E_{c,\lambda}(U)\right) = q_{c+A\lambda}(U),$$

and the minimized value is

$$-\tau \log Z_r(c + A\lambda) - 2\mathbf{1}^\top \lambda.$$

Thus the Lagrange dual objective is

$$\Phi_\tau(c, \lambda) = -\tau \log Z_r(c + A\lambda) - 2\mathbf{1}^\top \lambda.$$

We next compute its derivatives. Since

$$\frac{\partial E_{c,\lambda}(U)}{\partial \lambda_i} = d_i(U),$$

differentiating the log-partition function gives

$$\frac{\partial \Phi_\tau(c, \lambda)}{\partial \lambda_i} = \mathbb{E}_{q_{c+A\lambda}}[d_i(U)] - 2.$$

Differentiating once more,

$$\frac{\partial^2 \Phi_\tau(c, \lambda)}{\partial \lambda_i \partial \lambda_j} = -\frac{1}{\tau} \operatorname{Cov}_{q_{c+A\lambda}}\big(d_i(U), d_j(U)\big).$$

For any vector $a \in \mathbb{R}^{n-1}$,

$$a^\top \nabla^2_{\lambda\lambda} \Phi_\tau(c, \lambda) a = -\frac{1}{\tau} \operatorname{Var}_{q_{c+A\lambda}} \left(\sum_{i \in \bar{V}} a_i d_i(U) \right) \leq 0.$$

Hence $\Phi_\tau(c, \lambda)$ is concave in λ. Therefore any stationary maximizer $\lambda^\star(c)$ satisfies

$$0 = \frac{\partial \Phi_\tau(c, \lambda^\star(c))}{\partial \lambda_i} = \mathbb{E}_{q_{c+A\lambda^\star(c)}}[d_i(U)] - 2, \qquad i \in \bar{V}.$$

This proves the HK degree-balance condition. □

A.3 Proof of Theorem 2.3

Proof. For any $U \in \mathcal{U}_r$, define

$$S(U) := \sum_{i \in \bar{V}} (d_i(U) - 2)^2.$$

Every rooted 1-tree has root degree two and n edges, hence

$$\sum_{i \in \bar{V}} d_i(U) = 2(n-1), \qquad \sum_{i \in \bar{V}} (d_i(U) - 2) = 0.$$

If $U \notin \mathcal{H}$, the integral deviations $\{d_i(U) - 2\}_{i \in \bar{V}}$ are not all zero and sum to zero. Therefore at least one deviation is positive and at least one is negative, so

$$S(U) = \sum_{i \in \bar{V}} (d_i(U) - 2)^2 \geq 2.$$

Thus

$$\mathbf{1}\{U \in \mathcal{N}\} \leq \frac{1}{2} S(U).$$

Taking expectation under q gives

$$\gamma(q) = q(\mathcal{N}) \leq \frac{1}{2} \sum_{i \in \bar{V}} \mathbb{E}_q[(d_i(U) - 2)^2].$$

Under the degree-balance condition $\mathbb{E}_q[d_i(U)] = 2$, this becomes

$$\gamma(q) \leq \frac{1}{2} \sum_{i \in \bar{V}} \operatorname{Var}_q(d_i(U)).$$

For $q^\star = q_{c+A\lambda^\star(c)}$, the Hessian identity follows directly from equation 7:

$$\nabla^2_{\lambda\lambda} \Phi_\tau(c, \lambda^\star(c)) = -\frac{1}{\tau} \operatorname{Cov}_{q^\star}(d(U), d(U)).$$

Taking traces gives equation 9. □

A.4 Proof of Proposition 2.4

Proof. The Lagrangian of equation 11, up to constants independent of p, is

$$\sum_{U\in\mathcal{U}_r} p(U)\left\langle D + A\bar{\lambda} + Q^\top \nu, x_U\right\rangle + \tau \sum_{U\in\mathcal{U}_r} p(U)\log p(U).$$

Stationarity in p gives

$$p^\star(U) = q_{D+A\bar{\lambda}^\star + Q^\top\nu^\star}(U).$$

Decompose the raw field as

$$Q^\top\nu^\star = A\beta^\star + \Gamma^\star, \qquad \Gamma^\star \in \mathrm{Im}(A)^\perp,$$

and set $\lambda^\star := \bar{\lambda}^\star + \beta^\star$. Then

$$D + A\bar{\lambda}^\star + Q^\top\nu^\star = D + \Gamma^\star + A\lambda^\star,$$

so $p^\star = q_{\eta^\star}$ with

$$\eta^\star = D + \Gamma^\star + A\lambda^\star.$$

The degree KKT conditions give

$$\mathbb{E}_{p^\star}[d_i(U)] = 2, \qquad i \in \bar{V}.$$

By equation 6, this is equivalent to

$$\nabla_\lambda \Phi_\tau(D + \Gamma^\star, \lambda^\star) = 0.$$

Concavity of Φ_τ in λ gives

$$\lambda^\star \in \arg\max_\lambda \Phi_\tau(D + \Gamma^\star, \lambda).$$

Finally, the oracle marginal $u^\star := \mathbb{E}_{p^\star}[x_U]$ satisfies

$$u^\star = \mathbb{E}_{p^\star}[x_U] = \mathbb{E}_{q_{\eta^\star}}[x_U] = \mu(\eta^\star).$$

□

A.5 Proof of Proposition 2.5

For an edge parameter η, write

$$u(\eta) := \mu(\eta) = \mathbb{E}_{q_\eta}[x_U], \qquad \Sigma_x(\eta) := \mathrm{Cov}_{q_\eta}(x_U, x_U).$$

When the parameter is clear, we write simply u and Σ_x.

Recall

$$\mathcal{C}_{\mathrm{Ber}}(u) = \sum_{e\in\mathcal{E}} w_e u_e(1-u_e), \qquad w_e := \frac{1}{2}\sum_{i\in\bar{V}} B_{ie}.$$

Its marginal gradient is

$$b_{\mathrm{Ber}}(u) := \nabla_u \mathcal{C}_{\mathrm{Ber}}(u) = w \odot (1-2u).$$

The projected sharpening direction is

$$g_{\mathrm{Ber}}(u) := P_\perp b_{\mathrm{Ber}}(u), \qquad P_\perp := P_{\mathrm{Im}(A)^\perp}.$$

Lemma A.1 (Bernoulli upper certificate)**.** *Let q_η be HK-equilibrated and let $u = \mu(\eta)$. Then*

$$\gamma(q_\eta) \le \mathcal{C}_{\mathrm{Ber}}(u).$$

Proof. For each $i \in \bar{V}$,

$$d_i(U) = \sum_{e\in\mathcal{E}} B_{ie} x_{U,e}.$$

The rooted 1-tree Gibbs law factorizes into a weighted spanning-tree law on $\bar{V}$ and an independent weighted two-edge root-selection law. Edge indicators in both factors are negatively associated

(spanning-tree measures by Feder & Mihail (1992); the weighted two-subset root law is strongly Rayleigh and hence negatively associated by Borcea et al. (2009)); hence, for $e \neq f$,

$$\mathrm{Cov}_{q_\eta}(x_{U,e}, x_{U,f}) \leq 0.$$

Therefore,

$$\begin{aligned}\mathrm{Var}_{q_\eta}(d_i(U)) &= \mathrm{Var}_{q_\eta}\left(\sum_{e\in\mathcal{E}} B_{ie}x_{U,e}\right)\\ &= \sum_{e\in\mathcal{E}} B_{ie}^2\,\mathrm{Var}_{q_\eta}(x_{U,e}) + 2\sum_{e<f} B_{ie}B_{if}\,\mathrm{Cov}_{q_\eta}(x_{U,e}, x_{U,f})\\ &\leq \sum_{e\in\mathcal{E}} B_{ie}u_e(1-u_e),\end{aligned}$$

because $B_{ie} \in \{0,1\}$. Combining this with Theorem 2.3 gives

$$\begin{aligned}\gamma(q_\eta) &\leq \frac{1}{2}\sum_{i\in\bar{V}} \mathrm{Var}_{q_\eta}(d_i(U))\\ &\leq \frac{1}{2}\sum_{i\in\bar{V}}\sum_{e\in\mathcal{E}} B_{ie}u_e(1-u_e)\\ &= \sum_{e\in\mathcal{E}}\left(\frac{1}{2}\sum_{i\in\bar{V}} B_{ie}\right)u_e(1-u_e)\\ &= \mathcal{C}_{\mathrm{Ber}}(u).\end{aligned}$$

□

Lemma A.2 (Tangent identity of HK re-equilibration)**.** *Fix an edge input c at which the local HK map $c \mapsto \lambda^\star(c)$ is differentiable. Define*

$$\eta(c) := c + A\lambda^\star(c), \qquad T_c := D\eta(c).$$

Then

$$B\Sigma_x T_c = 0, \qquad T_c v - v \in \mathrm{Im}(A) \quad \textit{for every } v \in \mathbb{R}^{|\mathcal{E}|},$$

where $\Sigma_x = \Sigma_x(\eta(c))$.

Proof. The HK equilibrium condition is

$$B\mu(c + A\lambda^\star(c)) - 2\mathbf{1} = 0.$$

The rooted 1-tree Gibbs marginal satisfies

$$D\mu(\eta)[v] = -\frac{1}{\tau}\Sigma_x v.$$

Differentiating the HK equilibrium condition in direction v gives

$$B\Sigma_x T_c v = 0.$$

Since this holds for every v,

$$B\Sigma_x T_c = 0.$$

The second identity follows directly from

$$\eta(c) = c + A\lambda^\star(c).$$

Indeed, differentiating in direction v gives

$$T_c v = v + A\,D\lambda^\star(c)[v],$$

and hence

$$T_c v - v = A\,D\lambda^\star(c)[v] \in \mathrm{Im}(A).$$

□

Lemma A.3 (Projected Bernoulli descent identity)**.** *Define the reduced certificate*

$$\overline{\mathcal{C}}_{\mathrm{Ber}}(c) := \mathcal{C}_{\mathrm{Ber}}\left(\mu(c + A\lambda^\star(c))\right).$$

Let

$$u = u(c) := \mu(c + A\lambda^\star(c)), \qquad \Sigma_x = \Sigma_x(c + A\lambda^\star(c)).$$

Then

$$D\overline{\mathcal{C}}_{\mathrm{Ber}}(c)[g_{\mathrm{Ber}}(u)] = -\frac{1}{\tau}\left\|T_c g_{\mathrm{Ber}}(u)\right\|^2_{\Sigma_x} \le 0,$$

where

$$\|v\|^2_{\Sigma_x} := v^\top \Sigma_x v.$$

Proof. Let

$$b = b_{\mathrm{Ber}}(u), \qquad g = g_{\mathrm{Ber}}(u).$$

For any edge-input perturbation δc,

$$\begin{aligned} D\overline{\mathcal{C}}_{\mathrm{Ber}}(c)[\delta c] &= b^\top D\mu(c + A\lambda^\star(c))[T_c\delta c] \\ &= -\frac{1}{\tau} b^\top \Sigma_x T_c \delta c. \end{aligned}$$

Taking $\delta c = g$, it remains to simplify $b^\top \Sigma_x T_c g$.

Because $g = P_\perp b$, the removed component lies in $\mathrm{Im}(A)$. Thus there exists β such that

$$b - g = A\beta.$$

Using $A = B^\top$ and Lemma A.2,

$$(A\beta)^\top \Sigma_x T_c g = \beta^\top B \Sigma_x T_c g = 0.$$

Therefore,

$$b^\top \Sigma_x T_c g = g^\top \Sigma_x T_c g.$$

By Lemma A.2,

$$T_c g - g \in \mathrm{Im}(A).$$

Hence there exists ζ such that

$$T_c g = g + A\zeta, \qquad \text{equivalently} \qquad g = T_c g - A\zeta.$$

Again using $A = B^\top$ and $B\Sigma_x T_c = 0$,

$$(A\zeta)^\top \Sigma_x T_c g = \zeta^\top B \Sigma_x T_c g = 0.$$

Thus

$$\begin{aligned} g^\top \Sigma_x T_c g &= (T_c g - A\zeta)^\top \Sigma_x T_c g \\ &= (T_c g)^\top \Sigma_x T_c g \\ &= \|T_c g\|^2_{\Sigma_x}. \end{aligned}$$

Substituting into the directional derivative gives

$$D\overline{\mathcal{C}}_{\mathrm{Ber}}(c)[g] = -\frac{1}{\tau}\|T_c g\|^2_{\Sigma_x} \le 0.$$

□

The above three lemmas together allow us to formally close the proof for Proposition 2.5, as summarized below.

Proof of Proposition 2.5. The certificate claim follows from Lemma A.1.

For the descent claim, fix an edge input c at which the local HK map is differentiable, and let

$$u(c) = \mu(c + A\lambda^\star(c)).$$

By Lemma A.3,

$$D\overline{\mathcal{C}}_{\mathrm{Ber}}(c)[g_{\mathrm{Ber}}(u(c))] = -\frac{1}{\tau}\left\|T_c g_{\mathrm{Ber}}(u(c))\right\|_{\Sigma_x}^2.$$

Therefore the derivative is strictly negative whenever the re-equilibrated sharpening direction is nonzero, namely whenever

$$\left\|T_c g_{\mathrm{Ber}}(u(c))\right\|_{\Sigma_x} > 0.$$

By differentiability of $\overline{\mathcal{C}}_{\mathrm{Ber}}$, there exists $\bar{\alpha} > 0$ such that for every $0 < \alpha < \bar{\alpha}$,

$$\overline{\mathcal{C}}_{\mathrm{Ber}}\left(c + \alpha g_{\mathrm{Ber}}(u(c))\right) < \overline{\mathcal{C}}_{\mathrm{Ber}}(c).$$

With

$$c_\alpha = c + \alpha g_{\mathrm{Ber}}(u(c)), \qquad u_\alpha = \mu(c_\alpha + A\lambda^\star(c_\alpha)),$$

this is exactly

$$\mathcal{C}_{\mathrm{Ber}}(u_\alpha) < \mathcal{C}_{\mathrm{Ber}}(u(c)).$$

The proposition follows. □

B Training details

B.1 Experimental environment

Experiments were conducted on a workstation equipped with an AMD Ryzen Threadripper PRO 7955WX v16 CPU and 1 NVIDIA RTX 5090, running Ubuntu `22.04` LTS. The models were implemented using Python `3.10` and PyTorch `2.7.0`, with additional libraries including torch-scatter `2.1.2` and torch-geometric `2.6.1`. For reproducibility, a random seed of `42` was set for all random number generators. The code for this experiment is available at `https://anonymous.4open.science/r/C2TSP-EF65`.

B.2 Training procedure and implicit differentiation

We summarize the training procedure of C2TSP and the implicit differentiation used to backpropagate through the smoothed HK equilibration layer. The node price λ has a degenerate additive direction, so we resolve this degeneration by mapping λ to a fixed nondegenerate subspace; for simplicity, we still write the resulting coordinate as λ.

Theorem B.1 (Implicit differentiation of the HK layer)**.** *Let*

$$F(\theta, \lambda) := \mathbb{E}_{q_{D+\Gamma_\theta + A\lambda}}[d_{\bar{V}}(U)] - 2\mathbf{1}.$$

Suppose that

$$F(\theta, \lambda^\star(\theta)) = 0$$

and that

$$H(\theta, \lambda^\star) := -\partial_\lambda F(\theta, \lambda^\star)$$

is positive definite on the chosen nondegenerate subspace. Then $\lambda^\star(\theta)$ is locally unique and differentiable, and

$$\frac{d\lambda^\star}{d\theta} = -\big(\partial_\lambda F(\theta, \lambda^\star)\big)^{-1}\partial_\theta F(\theta, \lambda^\star).$$

Hence, the backward pass through the HK layer reduces to a single linear solve in the dimension of the restricted λ subspace.

B.3 Implemented inputs and sign convention

The GNN receives both a 2-D coordinate/proxy array X and the explicit cost matrix D: X supplies centered radial and inner-product features, while D supplies normalized costs, cost ranks, and the candidate graph. The rooted 1-tree, HK, certificate, objective, and decoding layers use D. The main-text raw field corresponds to $\widetilde{\Gamma}_\theta(X, D) := -\omega f_\theta(X, D)$, as shown in Algorithm 1.

Algorithm 1 Training C2TSP with residual rooted 1-tree sharpening

Require: Training batch (X, D), root r, GNN f_θ, perturbation weight ω, temperature τ, sharpening weight α, sharpening steps K, learning rate ρ
Ensure: Updated parameters θ

$\widetilde{\Gamma} \leftarrow -\omega f_\theta(X, D)$ ▷ raw residual edge field
$\Gamma^{(0)} \leftarrow P_\perp \widetilde{\Gamma}$ ▷ remove node-additive component
$c^{(0)} \leftarrow D + \Gamma^{(0)}$
$\lambda^{(0)} \leftarrow \lambda^\star(c^{(0)})$ ▷ HK equilibration
$\eta^{(0)} \leftarrow c^{(0)} + A\lambda^{(0)}$
$u^{(0)} \leftarrow \mu(\eta^{(0)})$
for $k = 0, \dots, K-1$ **do**
  $g^{(k)} \leftarrow P_\perp(w \odot (1 - 2u^{(k)}))$ ▷ certificate-guided sharpening
  $c^{(k+1)} \leftarrow c^{(k)} + \alpha g^{(k)}$
  $\lambda^{(k+1)} \leftarrow \lambda^\star(c^{(k+1)})$ ▷ re-equilibration
  $\eta^{(k+1)} \leftarrow c^{(k+1)} + A\lambda^{(k+1)}$
  $u^{(k+1)} \leftarrow \mu(\eta^{(k+1)})$
end for
$\mathcal{L}_{\mathrm{TSP}}(\theta) \leftarrow \langle D, u^{(K)} \rangle$
$\theta \leftarrow \theta - \rho \nabla_\theta \mathcal{L}_{\mathrm{TSP}}(\theta)$
return θ

Table 5: Zero-shot TSPLIB gap (%, mean±std.; time in seconds in parentheses).

	Structured (50)		Non-Euclidean (27)	
Method	L1	L2×100	L1	L2×100
C2TSP	**10.61**±7.27 (3.55)	**2.46**±1.93 (3.55)	**7.36**±3.96 (2.90)	**1.81**±1.94 (2.90)
DIFUSCO	47.50±29.86 (2.35)	11.72±15.34 (2.35)	31.98±26.96 (1.84)	7.22±11.42 (1.84)
Fast-T2T	27.72±17.16 (0.06)	7.06±8.48 (0.07)	25.67±17.39 (0.05)	5.13±6.97 (0.06)
DIMES	16.74±5.80 (0.02)	3.42±1.76 (0.03)	19.13±10.29 (0.02)	3.95±3.66 (0.02)
UTSP	36.06±15.00 (0.02)	9.18±6.18 (0.02)	29.09±21.96 (0.01)	7.26±7.77 (0.01)

C Additional experiments and robustness analyses

C.1 Common protocol and metric inputs

Unless stated otherwise, all transfer experiments use the TSP-100 checkpoints from the main paper without fine-tuning, and gaps are $100(\widehat{c} - c_{\mathrm{ref}})/c_{\mathrm{ref}}$. C2TSP receives both coordinates/proxies X and the explicit cost matrix D: its GNN uses invariant features derived from both, whereas rooted 1-tree inference, HK equilibration, the training objective, and tour evaluation use D. Dataset A supplies its sampled coordinates. Dataset B has no native coordinates: the C2TSP run reported here uses an all-zero coordinate proxy (making coordinate-derived channels constant) together with the exact 1–2 matrix, while coordinate-only baselines use a two-dimensional metric-MDS embedding (De Leeuw, 2005).

Dataset A costs are stored as symmetric integers after common positive scaling and upward rounding; the model receives each matrix normalized by its maximum, while tour costs and reference gaps use the stored integers.

C.2 TSPLIB transfer

We evaluate every selected TSPLIB instance (Reinelt, 1991) below 1,000 nodes: 50 instances with structured node distributions and 27 instances with non-Euclidean cost conventions. Table 5 reports aggregate zero-shot performance against published optima.

Table 6: Dataset A characterization.

n	Instances	Sparsity	Mean ratio	Median ratio	Max ratio
100	1000	0.1859	1.5277	1.0824	3094.90
200	1000	0.1066	1.6192	1.1387	2849.24
500	1000	0.0498	1.7305	1.2091	6813.94
1000	1000	0.0277	1.8059	1.2576	17198.38

Table 7: Dataset B L2×100 gap (%, mean±std.; time in seconds in parentheses).

Method	3-regular (24)	Sparse (9)	Dense (4)
C2TSP	**3.56**±3.24 (6.3)	**4.20**±0.79 (3.9)	**0.60**±0.40 (0.6)
DIFUSCO	63.80±32.78 (17.1)	66.53±25.92 (8.6)	18.71±6.86 (2.1)
Fast-T2T	62.05±32.67 (0.5)	62.43±25.36 (0.3)	18.59±8.36 (0.1)
DIMES	59.46±32.56 (0.2)	59.36±24.57 (0.1)	18.82±7.24 (0.0)
UTSP	61.16±32.71 (0.1)	61.52±24.83 (0.1)	15.84±8.95 (0.0)

Table 8: Dataset B C2TSP L2×100 per-instance gaps.

Instance	n	Gap (%)	Instance	n	Gap (%)
COL_1000.hcp	1086	1.75	COL_1950.hcp	1950	2.10
FLS3_1054.hcp	1054	4.27	FLS3_408.hcp	408	3.92
FLS3_731.hcp	731	4.24	FLS_1014.hcp	1014	4.14
FLS_170.hcp	170	4.71	FLS_338.hcp	338	5.92
FLS_507.hcp	507	3.35	FLS_676.hcp	676	3.25
FLS_845.hcp	845	4.02	GP0m_1006.hcp	1006	0.60
GP0m_130.hcp	130	0.77	GP0m_250.hcp	250	2.40
GP0m_490.hcp	490	1.02	GP3_1002.hcp	1002	0.50
GP3_126.hcp	126	1.59	GP3_246.hcp	246	2.44
GP3_486.hcp	486	0.82	GPN_122.hcp	122	0.82
GPN_244.hcp	244	2.46	GPN_482.hcp	482	0.83
GPN_998.hcp	998	0.50	II_1002.hcp	1002	6.19
II_1992.hcp	1992	5.82	QN_1044.hcp	1044	6.61
QN_1968.hcp	1968	5.89	SH_125.hcp	125	0.80
SH_250.hcp	250	0.80	SH_500.hcp	500	0.80
SH_64.hcp	64	0.00	SNm_1004.hcp	1004	4.08
SNm_124.hcp	124	10.48	SNm_252.hcp	252	11.11
SNm_500.hcp	500	9.20	SSP_1011.hcp	1011	3.86
SSP_2007.hcp	2007	3.64			

Table 9: LKH candidate-source controls (gap in ‰; 100 instances/size).

Candidate source	TSP100	TSP200	TSP500
Random 5 of 20-NN	666.44	568.66	345.84
Distance 5-NN	7.30	7.93	6.86
Non-learned 1-tree (α-nearness)	6.04	6.23	5.93
Vanilla LKH-3	0.38	0.22	0.36
Distance 5-NN + C2TSP initial tour	2.49	2.67	4.35
C2TSP (H2)	**0.11**	**0.13**	**0.20**

C.3 Metric non-Euclidean benchmarks

Dataset A: random shortest-path metrics. Following the random shortest-path construction (Bringmann et al., 2015; Klootwijk et al., 2021), we sample points in $[0, 1]^2$, draw an Erdős–Rényi graph $\mathrm{ER}(n, p)$ with $p = 4\log(n)/(n-1)$ until connected, weight retained edges by Euclidean length, and take the all-pairs weighted shortest-path closure. We generate 1,000 instances for $n \in \{100, 200, 500, 1000\}$. Table 6 characterizes the source graphs and induced detours. Sparsity is $|E|/\binom{n}{2}$; the mean and median columns first compute the corresponding cost-to-direct-Euclidean ratio within each instance and then aggregate across instances, while "max ratio" is the maximum per-instance maximum.

The large maxima arise when two sampled points are extremely close in X but their direct source edge is absent, so their shortest path makes a finite detour while the Euclidean denominator approaches zero. The aggregate mean and median are therefore more representative of typical pairs.

Dataset B: Hamiltonian 1–2 metrics. We use the 37 Flinders Hamiltonian-cycle benchmark graphs (Baniasadi et al., 2018) with at most 3,000 nodes (the retained range is 64–2,007). Existing graph edges receive cost 1 and non-edges cost 2. This is a metric, and because each source graph is Hamiltonian, the exact TSP optimum is n. Dataset B has no native coordinates or coordinate-ratio statistic. Table 7 reports L2×100 results by graph family. C2TSP has an overall mean gap of 3.40% and median gap of 3.25% over all 37 instances; Table 8 gives its per-instance gaps. The remaining decoder means on Dataset B are 4.81% (L1), 4.53% (L2×1), 3.44% (L2×10), 6.44% (L3), and 7.09% (L4).

Non-Euclideanity quantification. We use a spectral non-Euclideanity score based on Gower's criterion (Gower, 1966): the score is zero for Euclidean distances, and larger values indicate stronger deviations from Euclidean geometry. Dataset A scores 0.27–0.40 and Dataset B scores 0.10–0.28. These clearly nonzero scores support their use as non-Euclidean benchmarks.

C.4 No-learned-signal LKH controls

Each control in Table 9 uses five candidates from the same 20-nearest-neighbor pool and otherwise matches H2; only the ranking signal changes. Structural compatibility alone is insufficient: non-learned 1-tree candidates remain well behind C2TSP's learned ranking.

C.5 Root sensitivity and averaging

On the TSP-50 test set, evaluating the same checkpoint once with each possible root gives mean gap 1.626%, standard deviation 0.046%, range $[1.528, 1.717]\%$, and median 1.634%. Table 10 compares root 0 with averaging model outputs over roots 0–9 before decoding. Averaging changes L1–L3 little and degrades the root-dependent L4 decoder, so we retain a single root.

Table 10: TSP-50 root averaging (mean gap, %; 1,000 instances).

Setting	L1	L2×1	L2×10	L2×100	L3	L4
Root 0	6.100	4.229	1.328	1.316	2.966	1.920
Average roots 0–9	6.176	4.274	1.324	1.302	2.988	3.973

Table 11: TSP-50 forward-equilibration ablation.

Method	Gap (%)	Mean $\|r\|_2$
Forward HK only	2.206	0.0341
Penalty only	20.811	0.6959
Forward HK + penalty	2.212	0.0337

Table 12: Model size and training wall time.

Method	Parameters	Train instances	Wall time
C2TSP	123,750	2,048	1 h 16 m 40 s
DIFUSCO	5,333,762	2,048	52 m 05 s
Fast-T2T	5,333,762	2,048	1 h 16 m 12 s
DIMES	64,321	2,048	15 m 25 s
UTSP	36,072	2,048	1 m 51 s

Table 13: C2TSP stochastic decoder gap (%, mean±std. across 10 seeds).

	TSP50	TSP100	TSP200	TSP500	TSP1000
L3	3.12±0.05	5.09±0.02	7.93±0.05	13.45±0.07	20.42±0.06
L4	2.30±0.03	4.61±0.04	8.75±0.05	15.51±0.02	21.66±0.03

C.6 Forward HK versus a loss penalty

We disable sharpening and compare forward equilibration (Table 11) with the training penalty $5\|r\|_2^2$, where r contains degree-two residuals. Forward HK yields substantially better tour quality and degree balance than penalty-only training; adding the penalty to forward HK has no measurable benefit.

C.7 Training cost

Table 12 reports 100-epoch wall time on the same RTX 5090 workstation and 2,048 TSP-100 training instances. The supervised DIFUSCO and Fast-T2T times exclude optimal-tour label generation; C2TSP is label-free.

C.8 Evaluation-seed stability

We repeated the core decoder and LKH evaluations with 10 random seeds on the same test instances. L1 and L2 are deterministic for C2TSP, DIMES, and UTSP; variability occurs only in stochastic sampling or solver components. No method ranking changes. The largest standard deviation across all cells is 0.28 percentage points (UTSP L3 on TSP-500), and every C2TSP cell is at most 0.07 points. For example, the stochastic C2TSP rows are shown in Table 13.

C.9 Representative refinement and decoding on TSP-50

Figure 4 presents an example of a TSP-50 instance that demonstrates the mechanisms of the proposed model across its refinement stages. Because nodes are reordered according to the optimal tour, low-

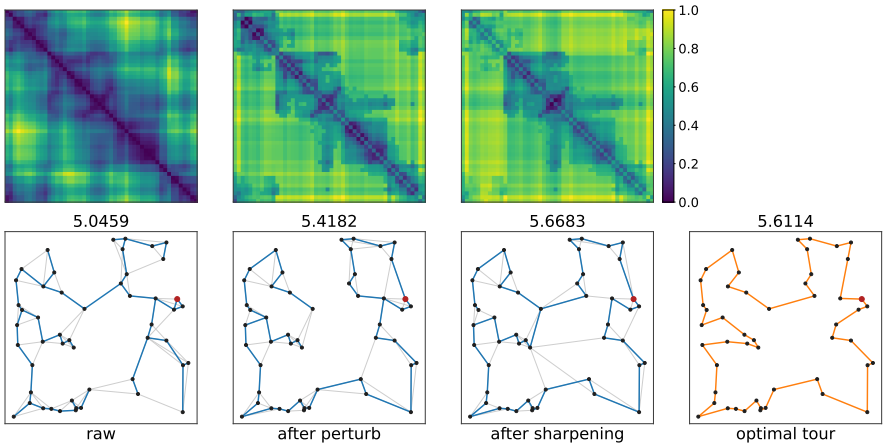

Figure 4: Refinement and decoding on a representative TSP-50 instance. **Top**: raw, perturbed, and sharpened cost matrices, with nodes ordered by the optimal tour so that optimal-tour edges lie near the super- and sub-diagonals. **Bottom**: the corresponding MAP rooted 1-trees and the optimal tour. Titles report rooted 1-tree or tour costs.

cost mass near the super- and sub-diagonals indicates alignment with optimal-tour edges, whereas isolated low-cost regions far from the diagonal may induce structurally poor rooted 1-trees. In the raw cost matrix, low-cost edges are dispersed across the edge space. Edge perturbation substantially reshapes this pattern by moving low-cost mass toward the tour-aligned diagonal band. Sharpening then makes a more localized adjustment: the heatmap before and after sharpening has a similar global pattern, but ambiguous off-diagonal low-cost regions are further suppressed. This is expected because sharpening acts as a fine-tuning step rather than a full re-learning step, which is also reflected in the MAP rooted 1-tree, where only a small number of edges are changed. The resulting MAP rooted 1-tree is therefore shaped toward a more tour-like structure, rather than merely assigned a lower cost.

D STRUCTURED MAP DECODING

The structured MAP decoder converts the final rooted 1-tree distribution into feasible tours for evaluation. Given the final sharpened cost $C^{(K)}$ and marginal $u^{(K)}$, the decoder repeatedly extracts MAP rooted 1-trees, repairs them into Hamiltonian tours, and returns the candidate with the lowest true metric cost. Unlike local-search-based post-processing, this decoder directly uses the connected rooted 1-tree structure and only applies a repair step to enforce the degree-two constraints (the MAP rooted 1-tree is already connected).

For each decode draw $s = 0, \ldots, S-1$, we form a perturbed cost

$$C^{(s)} = C^{(K)} + \sigma \Delta C^{(s)},$$

where $\Delta C^{(0)} = 0$ gives the deterministic candidate and the remaining draws use small graph-structured perturbations. The perturbation is used only to diversify the MAP rooted 1-tree candidates. Given $C^{(s)}$, the decoder computes

$$\mathcal{T}^{(s)} = \mathrm{MAPRootedOneTree}(C^{(s)}, r),$$

which consists of a minimum spanning tree on the non-root nodes together with the two cheapest root edges. Since $\mathcal{T}^{(s)}$ is connected but not necessarily a tour, it is repaired into a feasible Hamiltonian cycle:

$$\pi^{(s)} = \mathrm{Repair}(\mathcal{T}^{(s)}, u^{(K)}, C^{(s)}, D).$$

The repair step uses the learned marginals, the perturbed cost, and the true metric distance matrix. Each candidate is evaluated by its true metric tour cost, and the decoder returns the best one:

$$\widehat{\pi} = \arg\min_{\pi^{(s)}} \mathrm{TourCost}(\pi^{(s)}, D).$$

Algorithm 2 Structured MAP decoding

Require: Final cost $C^{(K)}$, final marginal $u^{(K)}$, distance matrix D, root r, decode draws S, perturbation scale σ
Ensure: Decoded tour $\widehat{\pi}$
1: **for** $s = 0, \ldots, S-1$ **do**
2: Generate a small perturbation $\Delta C^{(s)}$ with $\Delta C^{(0)} = 0$
3: $C^{(s)} \leftarrow C^{(K)} + \sigma \Delta C^{(s)}$
4: $\mathcal{T}^{(s)} \leftarrow \mathrm{MAPRootedOneTree}(C^{(s)}, r)$
5: $\pi^{(s)} \leftarrow \mathrm{Repair}(\mathcal{T}^{(s)}, u^{(K)}, C^{(s)}, D)$
6: $c^{(s)} \leftarrow \mathrm{TourCost}(\pi^{(s)}, D)$
7: **end for**
8: $\widehat{\pi} \leftarrow \arg\min_{\pi^{(s)}} c^{(s)}$
9: **return** $\widehat{\pi}$