REMAP: LOSSLESS RESIDUAL NEURAL LIFTING FOR SCALABLE MAP INFERENCE IN ARBITRARY-ORDER MARKOV RANDOM FIELDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Maximum a posteriori inference in a finite Markov random field is generally NP-hard. We study an instance-wise optimizer that minimizes the exact multilinear energy under product marginals. The canonical parameterization adds a GNN output to free per-node logits. This residual bypass is a global change of coordinates, so the lifting preserves the infimum, critical points, local extrema (modulo the GNN-parameter fiber), and Hessian inertia of the free-logit problem. Joint gradient flow then induces the preconditioner $I + JJ^\top$: the GNN cannot enlarge the feasible set, but it can add dissipation $\|J^\top \nabla \ell\|_2^2$. For a two-layer branch this extra coupling is four-hop local. We also show that every finite critical point of a nonconstant multilinear objective is a saddle, and that a unique MAP gap Γ makes conditional-expectation rounding exact once the relaxed gap falls below Γ; residual gradient flow and joint gradient descent then obey explicit $O(1/t)$ and $O(1/k)$ rates in a small-gap region, with an improved bound under residual alignment. Experiments on synthetic MRFs, UAI 2022 benchmarks, and Physical Cell Identity (PCI) problems evaluate this residual geometry at scale. The code of our model can be accessed at `https://anonymous.4open.science/r/REMAP_ICLR2027-2F33`

.

1 INTRODUCTION

Let $x = (x_1, \ldots, x_n)$ be a collection of finite-valued variables and let $E(x)$ be a sum of local factor energies. MAP inference asks for an assignment with minimum energy. This elementary statement covers pairwise MRFs, arbitrary-order factor graphs, weighted constraint networks, and many application-specific discrete objectives in computer vision, language processing, communications, and network optimization (Wang et al., 2013; Su et al., 2021; Almutiri & Nadeem, 2022; Ammar et al., 2014; Lin et al., 2020; Wu et al., 2020; Yunfei Ma & Razavi, 2022). It is also NP-hard in general (Koller & Friedman, 2009). Approximate message-passing methods such as LBP and TRBP are efficient but often degrade on hard dense or high-order problems (Ihler et al., 2005; Pitkow et al., 2011; Wainwright et al., 2005b;a). Exact branch-and-bound solvers can certify optimality, and Toulbar2 (De Givry, 2023) is among the strongest general-purpose MAP solvers, having won all MPE and MMAP categories in the UAI 2022 Inference Competition. Even so, finding high-quality incumbents within a practical time budget remains challenging on large or high-order instances.

Neural approaches have shown promise for combinatorial optimization and probabilistic inference, but their use in MRF MAP inference has mostly been limited to small pairwise graphs, supervised imitation of classical solvers, or amortized inference across training distributions (Garcia Satorras & Welling, 2021; Kuck et al., 2020; Yoon et al., 2019; Cui et al., 2022). These settings do not fully address large arbitrary-order MRFs with heterogeneous label cardinalities. Related instance-wise or self-supervised neural methods, including PI-GNN (Schuetz et al., 2022), SSMP (Arya et al., 2024a), and SINE (Arya et al., 2025), likewise do not make "no labels" itself a standalone contribution. The central question is whether a graph-structured lifting supplies a useful optimization geometry without sacrificing the free-logit problem.

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.

REMAP explores this trade-off. For one input instance, a GNN and a free per-node residual produce categorical logits. The expected MRF energy under the resulting product distribution is the standard multilinear relaxation (Lê-Huu & Paragios, 2018); it is differentiable, so all variables can be optimized on a GPU. A discrete incumbent is extracted during the run and all neural parameters are discarded afterward. Thus, "training" is simply the numerical solution of one lifted optimization problem. The starting observation is concrete: if a GNN branch is added residually to free logits, $z = u + \rho_v(G)$, the neural map becomes a global coordinate lifting of the original problem. This gives landscape equivalence, modulo the GNN-parameter fiber, and identifies the only possible optimization benefit—an adaptive positive-semidefinite preconditioner induced by the GNN Jacobian. It also makes the necessary experimental comparison unambiguous: residual REMAP versus the identical free-logit optimizer and versus the GNN branch alone.

The theoretical analysis specializes locality statements to a two-layer message-passing branch. The lossless-lifting equivalence itself holds for any continuously differentiable GNN branch. In the experiments we retain the original multi-layer settings: a 5-layer GNN on synthetic and PCI instances and an 8-layer GNN on UAI instances.

This paper makes four contributions.

- We introduce a residual GNN lifting $z = u + \rho_v(G)$ whose parameter-to-logit map is globally equivalent to a change of coordinates. It is lossless in value and preserves the critical-point and local-extremum structure of free logits, with additional flat directions along the GNN-parameter fiber.
- We show that joint gradient flow induces the adaptive metric $I + JJ^\top$. We quantify its exact additional dissipation, derive an $O(1/T)$ logit-stationarity guarantee, and show that the node-embedding metric of a two-layer GNN is four-hop sparse. Under residual alignment $\|J_v^\top q\|_2^2 \geq \beta\|q\|_2^2$, the stationarity bound improves by the factor $(1+\beta)^{-1}$.
- We sharpen the underlying MAP theory: a nonconstant multilinear objective has no finite interior local extrema; a unique MAP gap gives exact conditional-rounding and argmax recovery thresholds; and small-gap gradient flow and joint gradient descent converge in objective and product-distribution distance at explicit $O(1/t)$ and $O(1/k)$ rates, with the constant c_0 replaced by $(1+\beta)c_0$ under residual alignment.
- We evaluate residual REMAP on synthetic pairwise and high-order MRFs, UAI 2022 inference benchmarks, and real-world Physical Cell Identity (PCI) problems, and isolate the residual, GNN-only, and free-logit parameterizations on the five real PCI cases.

Related work. A detailed comparison with Le-Huu–Paragios, PI-GNN, SSMP, SINE, and learned message-passing methods is given in Appendix A.

2 PRELIMINARIES

Markov Random Field. An MRF is defined over an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{C})$, where $\mathcal{V}$ represents the index set of random variables and $\mathcal{C} \subseteq 2^{\mathcal{V}}$ is the clique set representing the (high-order) dependencies among random variables. Throughout this paper, we associate each node index i with a random variable $x_i \in \mathcal{X}$, where $\mathcal{X}$ is a finite alphabet. Thus, given graph $\mathcal{G}$, the joint probability of a configuration of $X = \{x_i\}_{i\in\mathcal{V}}$ can be expressed as Eq. 1,

$$\mathbb{P}(X) = \frac{1}{Z}\exp(-E(X)) = \frac{1}{Z}\exp\left(-\sum_{i\in\mathcal{V}}\theta_i(x_i) - \sum_{C_k\in\mathcal{C}}\theta_{C_k}(\{x_l|\forall x_l \in C_k\})\right) \tag{1}$$

where Z is the partition function, $\theta_i(\cdot)$ denotes the unary energy functions, $\theta_C(\cdot)$ represent the clique energy functions. In this sense, MRF provides a compact representation of probability by introducing conditional dependencies:

$$\mathbb{P}(x_i|X\backslash\{x_i\}) = \mathbb{P}(x_i|\{x_j\} \text{ for } i,j \in C_k \text{ for } C_k \in \mathcal{C}). \tag{2}$$

In this paper, we consider the MAP estimate of Eq. 1. MAP inference asks for an assignment $x^\star \in \arg\min_x E(x)$, with optimal value $E^\star = \min_x E(x)$. One can consult (Koller & Friedman, 2009) for more details.

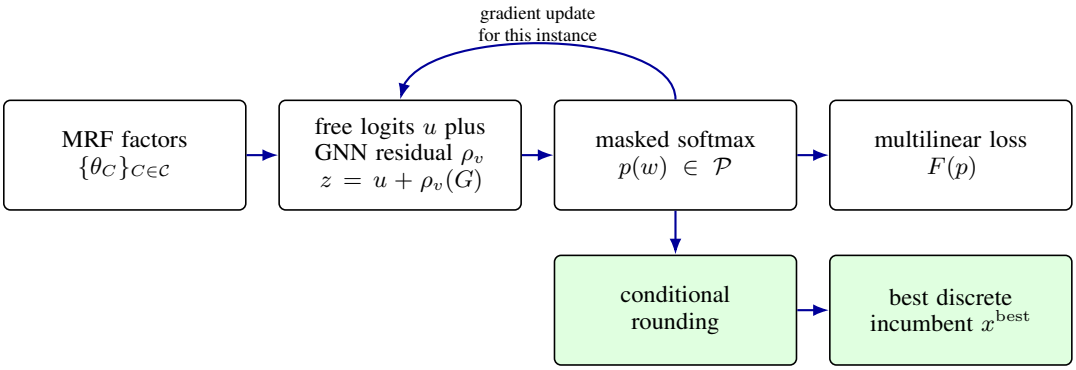


Figure 1: Residual REMAP is an optimizer for one instance. The factor energy supplies the loss; the free-logit bypass preserves every search direction; the GNN supplies an adaptive graph-structured geometry. Conditional-expectation rounding converts every relaxed iterate into a no-worse discrete incumbent.

Graph Neural Networks. GNNs represent a distinct class of neural network architectures specifically engineered to process graph-structured data (Kipf & Welling, 2017; Hamilton et al., 2017; Xu et al., 2019; Veličković et al., 2018). In general, when addressing a problem involving a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{E}$ is the edge set, GNNs utilize both the graph $\mathcal{G}$ and the initial node representations $\{h_i^{(0)} \in \mathbb{R}^d | \forall i \in \mathcal{V}\}$ as inputs, where d is the dimension of initial features. Assuming the total number of GNN layers to be K, at the k-th layer the graph convolutions typically read:

$$h_i^{(k)} = \sigma\left(W_k \cdot \text{AGGREGATE}^{(k)}\left(\left\{h_j^{(k-1)} : j \in \mathcal{N}(i) \cup \{i\}\right\}\right)\right) \tag{3}$$

where $\text{AGGREGATE}^{(k)}$ is defined by the specific model, W_k is a trainable weight matrix, $\mathcal{N}(i)$ is the neighborhood of node i, and σ is a non-linear activation function, e.g., ReLU.

3 METHODOLOGY

An overview of residual REMAP is in Fig. 1. Given one MRF instance, we expand each factor into pairwise GNN edges, optimize free per-node logits together with a GNN residual, evaluate the exact multilinear energy under the resulting product distribution, and extract a discrete incumbent by conditional-expectation rounding. All parameters are reset for the next instance. The factor tables—not the clique expansion—define the loss. The residual bypass $z = u + \rho_v(G)$ is what makes the lifting globally lossless: the GNN supplies an adaptive graph-structured geometry, while free logits preserve every search direction of the original continuous problem.

3.1 PREPROCESSING

Topology construction for GNNs. In an MRF instance, the high-order graph structure consists of nodes and cliques, diverging from typical GNNs allowing only pairwise edges (2nd-order). To facilitate the power of GNNs, we need to convert the high-order factor structure into a pairwise graph.

By the very definition of a clique, any two nodes that appear within the same clique are directly related. Thus, for any two nodes $i, j \in C_k$ in a clique C_k, we add a pairwise edge (i, j) to its GNN-oriented graph. An edge may appear in multiple cliques; we add each edge only once. This expanded pairwise graph is the support of the GNN branch $\rho_v(G)$ in Fig. 1; the factor tables, not the extra edges, define $F(p)$.

Initial features for GNNs. As there are no initial features associated with MRF instances, we initialize feature vectors to GNNs *randomly* with a predefined dimension d. Detailed information on how we will handle these artificial features to ensure they effectively capture the underlying information of the problem will be provided in Section 3.2.

Vectorizing the energy function. The transformed energy function $E(X)$ will serve as the loss function guiding the training of the neural networks. In Section 3.3, we will detail the transformation

process and discuss how to effectively utilize it. Note the values of these functions can be pre-evaluated and repeatedly used during the training process. Therefore, we employ a look-up table to memorize all function values with discrete inputs. For unary energies, we denote the vectorized unary energy of variable x_i as $\phi(x_i)$, where the n-th element corresponds to $\theta_i(x_i = n)$. Similarly, we represent the clique energy for clique C_k using the tensor $\psi(\{x_l|\forall x_l \in C_k\})$. This tensor can be derived using the same conceptual framework; for instance, the element $\psi(x_i, x_j, x_k)$ at position $(0, 2, 4)$ corresponds to the value of $\theta_{\{i,j,k\}}(x_i = 0, x_j = 2, x_k = 4)$.

Padding node embeddings & energy terms and Masking. GNNs typically require all node embeddings to be of the same dimension, meaning that the embeddings $h^{(K)}$ at the K-th layer must share the same size. However, in general MRFs, the variables often exhibit different numbers of states. While traditional belief-propagation-based methods can easily manage such variability, adapting GNNs to handle these discrepancies is less straightforward.

To address this mismatch, we pad node embeddings and unary/pairwise (or clique) energy tables to a common cardinality $|\mathcal{X}|$ so that batched GNN tensors have a uniform last dimension. Concretely, we introduce dummy states for nodes with $s_i < |\mathcal{X}|$ and fill the extra table slices with an arbitrary placeholder (implementation: the original table maximum). The schematic diagram is in Appendix B. Padding is only a layout device: invalid logits are set to $-\infty$ before softmax, so dummy labels have probability exactly zero. Once this mask is applied, the placeholder values stored in padded energy regions cannot affect $F(p)$ or the decoder.

Remark 3.1. If invalid logits were left finite and padded energies were filled with large constants, those dummy labels could still leak into the multilinear contraction. Masking with $-\infty$ before softmax removes this leakage: padded table entries are multiplied by zero probability and are irrelevant to both the loss and the decoder.

3.2 Residual GNN lifting

Let $D = \sum_i s_i$ be the number of valid logits and write $z = (z_i)_i \in \mathbb{R}^D$. The canonical parameterization uses free logits $u \in \mathbb{R}^D$ and a GNN branch $\rho_v(G) \in \mathbb{R}^D$:

$$z(u, v) = u + \rho_v(G), \qquad p_i(z; \tau) = \mathrm{softmax}(z_i/\tau), \qquad L_\tau(u, v) = \ell_\tau(z(u, v)) = F(p(z; \tau)). \tag{4}$$

All parameters (u, v) are reset and optimized for one instance. A fixed $\tau = 1$ is canonical; annealing is optional and lies outside the fixed-temperature theorems. Adding a constant to all logits of one node leaves p_i unchanged; we retain these harmless gauge directions.

The graph branch maps every factor to a clique, starts from trainable node embeddings $h_i^{(0)} \in \mathbb{R}^{d_l}$, applies message-passing layers, and decodes node states into the valid coordinates of $\rho_v(G)$. The intuition is the same as in loopy belief propagation: incoming messages aggregate information from subgraphs that a node cannot access directly. Node i updates its beliefs by combining unary information with neighborhood messages,

$$p^{\textbf{posterior}}(x_i|X\backslash\{x_i\}) = \exp(-\phi(x_i)) \prod_{(i,j)\in\mathcal{E}} \sum_{x_j} M_{ji}, \tag{5}$$

which is structurally analogous to GNN message passing.

Graph convolutions should treat adjacent nodes equally, consistent with MRF neighborhood aggregation. Typical GNNs are summarized in Table 11 in Appendix I. We use the GraphSAGE aggregator (Hamilton et al., 2017) as the backbone; GCN (Kipf & Welling, 2017) and GAT (Veličković et al., 2018) are compared in Fig. 5. We further use Jumping Knowledge (Xu et al., 2018) so that local and longer-range features can be combined. For Hessian statements we assume a twice continuously differentiable activation; first-order results require differentiability only along the trajectory.

The theoretical locality result in Section 4 is stated for a two-layer branch: if $h_i^{(0)} = e_i$ and $k \in \{1, 2\}$, a GraphSAGE-style update is given by

$$m_i^{(k)} = \frac{1}{|\mathcal{N}(i)|} \sum_{j\in\mathcal{N}(i)} h_j^{(k-1)}, \qquad h_i^{(k)} = \sigma\big(A_k h_i^{(k-1)} + B_k m_i^{(k)} + b_k\big) \tag{6}$$

For isolated nodes, the neighbor aggregate is taken to be zero. This update yields a four-hop node-embedding metric. A Jumping-Knowledge decoder then maps $[h_i^{(0)}, h_i^{(1)}, h_i^{(2)}]$ to $\rho_{v,i}(G)$. In

experiments we keep the original deeper stacks ($K = 5$ on synthetic and PCI instances, $K = 8$ on UAI instances). The lossless-lifting equivalence of Theorem 4.5 does not depend on width, depth, or expressivity; only the four-hop sparsity of Proposition 4.8 is two-layer specific.

The free logits, embeddings, shared matrices, and decoder are optimized jointly. Invalid logits induced by label-cardinality padding are set to $-\infty$ before softmax, so their probabilities are exactly zero and padded energy entries cannot affect $F(p)$.

3.3 Energy minimization and incumbent extraction

Let $\Delta_i = \{p_i \in \mathbb{R}_+^{s_i} : \mathbf{1}^\top p_i = 1\}$ and $\mathcal{P} = \prod_i \Delta_i$. The multilinear objective (Lê-Huu & Paragios, 2018) is

$$F(p) = \sum_{C \in \mathcal{C}} \sum_{x_C} \theta_C(x_C) \prod_{i \in C} p_i(x_i). \tag{7}$$

Equivalently, writing v_i for a one-hot encoding of x_i and $P_k = \otimes_{i \in C_k} p_i$,

$$E(\{v_i | i \in \mathcal{V}\}) = \underbrace{\sum_{i \in \mathcal{V}} \langle v_i, \phi(x_i) \rangle}_{\text{Unary Term}} + \underbrace{\sum_{C_k \in \mathcal{C}} \langle \psi(C_k), V_k \rangle}_{\text{Clique Term}} \tag{8}$$

$$F(p) = \underbrace{\sum_{i \in \mathcal{V}} \langle p_i, \phi(x_i) \rangle}_{\text{Unary Term}} + \underbrace{\sum_{C_k \in \mathcal{C}} \langle \psi(C_k), P_k \rangle}_{\text{Clique Term}}. \tag{9}$$

This is the expected energy under the product distribution $\prod_i p_i$; it is affine in each block p_i but not jointly convex. Residual ReMAP optimizes $L_\tau(u, v) = F(p(z(u, v); \tau))$ from Eq. 4. Exactness belongs to F on $\mathcal{P}$, not to the GNN.

At every checkpoint we decode with conditional-expectation rounding and retain the lowest-energy incumbent. This decoder returns a discrete assignment x_{CE} with $E(x_{\mathrm{CE}}) \le F(p)$ for every $p \in \mathcal{P}$. Node-wise argmax is a cheaper optional decoder and receives a separate confidence bound in Section 4; all reported experiments use conditional-expectation rounding. We use Adam (Kingma & Ba, 2015) as a practical optimizer; the dynamics below analyze exact gradient flow and gradient descent. Annealing, when used, is a heuristic outside the fixed-temperature theorems.

4 Theory: Lossless Lifting and Graph-Structured Optimization

Proofs of all formal statements are deferred to Appendix C. Write $E_\star = \min_{x \in \mathcal{X}} E(x)$ and, for a variable i, the conditional cost

$$g_i(a; p_{-i}) = \mathbb{E}_{X_{-i} \sim \prod_{j \neq i} p_j}[E(a, X_{-i})], \qquad F(p) = \langle p_i, g_i(p_{-i}) \rangle. \tag{10}$$

Theorem 4.1. *[Expectation identity, exact value, and minimizers] Let $X_i \sim p_i$ independently. For every $p \in \mathcal{P}$,*

$$F(p) = \mathbb{E}_{X \sim \prod_i p_i}[E(X)] \ge E_\star. \tag{11}$$

Moreover, $\min_{p \in \mathcal{P}} F(p) = E_\star$. Equality $F(p) = E_\star$ holds if and only if every assignment in $\mathrm{supp}(\prod_i p_i)$ is globally optimal. In particular, if the MAP assignment is unique, the unique relaxed minimizer is its one-hot encoding.

The theorem separates exactness of the objective on $\mathcal{P}$ from expressivity of its parameterization.

Corollary 4.2. *[Conditional-expectation rounding] For every $p \in \mathcal{P}$, a deterministic assignment x_{CE} can be constructed sequentially such that $E(x_{\mathrm{CE}}) \le F(p)$. At step i, set a label minimizing the expected energy when that label and all earlier choices are fixed while unrounded variables retain their current marginals. The guarantee holds for every variable order and arbitrary finite factors.*

This is the method of conditional expectations and is closely related to block rounding in Lê-Huu and Paragios (Lê-Huu & Paragios, 2018). For cases where argmax is retained for speed, define $\mathrm{osc}(\theta_C) = \max_{x_C} \theta_C(x_C) - \min_{x_C} \theta_C(x_C)$.

Proposition 4.3. *[Argmax-rounding deviation] Let $\hat{x}_i \in \arg\max_a p_i(a)$. Then*

$$\big|E(\hat{x}) - F(p)\big| \le \sum_{C\in\mathcal{C}} \operatorname{osc}(\theta_C)\Big(1 - \prod_{i\in C} p_i(\hat{x}_i)\Big) \le \sum_{C\in\mathcal{C}} \operatorname{osc}(\theta_C) \sum_{i\in C}\big(1 - p_i(\hat{x}_i)\big). \tag{12}$$

Thus the argmax gap is controlled by both distributional concentration and factor ranges.

What finite softmax can represent. Finite logits parameterize the relative interior of each simplex and cannot produce a one-hot vector. The closed product of simplices attains $\min_{p\in\mathcal{P}} F(p) = E_\star$ at vertices (Theorem 4.1); finite softmax generally only attains an infimum, approached as $\|z\| \to \infty$.

Theorem 4.4. *[Finite-softmax infimum and a quantitative approximation] Let $z = (z_i)_i$ be free finite logits and $p_i = \operatorname{softmax}(z_i/\tau)$ for fixed $\tau > 0$. Then $\inf_z F(p(z)) = E_\star$. The infimum is attained at finite z if and only if every assignment is globally optimal. More quantitatively, let $x^\star$ be optimal, let $R_E = \max_x E(x) - E_\star$, and set $z_i(x_i^\star) = m$, $z_i(a) = 0$ for $a \neq x_i^\star$. Then*

$$0 \le F(p(z)) - E_\star \le R_E \sum_{i=1}^{n} (s_i - 1) e^{-m/\tau}. \tag{13}$$

The result explains why a convergence theorem for the parameter iterates is the wrong default target: even in the free-logit model, a loss-minimizing sequence may require $\|z\| \to \infty$.

A lossless residual lifting. Let $v \in \mathcal{V} \subseteq \mathbb{R}^m$, with $\mathcal{V}$ open, parameterize any continuously differentiable GNN output $\rho_v(G) \in \mathbb{R}^D$. The next result holds regardless of width, depth, activation, factor order, or expressivity; its Hessian statement additionally assumes ρ is twice continuously differentiable.

Theorem 4.5. *[Global lossless-lifting equivalence] Define $T(u, v) = (u + \rho_v(G), v)$. Then T is a global diffeomorphism with inverse $T^{-1}(z, v) = (z - \rho_v(G), v)$, and $L_\tau \circ T^{-1}(z, v) = \ell_\tau(z)$. Consequently: (i) $\inf_{u,v} L_\tau(u, v) = \inf_z \ell_\tau(z) = E_\star$; (ii) (u, v) is critical, a local minimum, or a local maximum of L_τ if and only if $z = u + \rho_v(G)$ has the corresponding property for ℓ_τ; (iii) every sublevel set of L_τ is diffeomorphic to the corresponding free-logit sublevel set times $\mathcal{V}$; (iv) at a critical point, the Hessian of L_τ has the same positive and negative inertia as $\nabla^2\ell_\tau(z)$, with at least m additional zero directions.*

In the transformed coordinates (z, v), the lifted objective equals $\ell_\tau(z)$ and is constant along the v-fiber. Hence (u, v) is a critical point or local extremum of L_τ if and only if $z = u + \rho_v(G)$ is a critical point or local extremum of ℓ_τ; the additional zero Hessian directions in (iv) are these fiber directions. Theorem 4.5 is a general landscape-equivalence statement. In the nonconstant MAP case, Theorem 4.9 further implies that finite stationary points are saddles rather than local extrema. Any optimization benefit must come from the metric, not from a larger set of logits.

Identity-plus-GNN preconditioning. Write $J_v = D_v\rho_v(G)$ and $q = \nabla\ell_\tau(z)$. Residual alignment of strength $\beta \ge 0$ means

$$\|J_v^\top q\|_2^2 \ge \beta\|q\|_2^2. \tag{14}$$

When $q \neq 0$, the diagnostic ratio is $\beta_t = \|J_v^\top q\|_2^2/\|q\|_2^2$. This is a squared-norm ratio: the joint gradient satisfies

$$\|\nabla_{u,v} L_\tau\|_2^2 = \|q\|_2^2 + \|J_v^\top q\|_2^2 = q^\top(I + J_v J_v^\top)q \ge (1 + \beta)\|q\|_2^2, \tag{15}$$

so any stationarity or descent bound written in terms of $\|q\|_2^2$ improves by the factor $(1 + \beta)^{-1}$. Alignment is instance-, parameter-, and trajectory-dependent; the architecture does not force $\beta > 0$, and $\beta = 0$ recovers the free-logit bound.

Theorem 4.6. *[Induced logit dynamics and dissipation] Along $\dot{u} = -\nabla_u L_\tau$ and $\dot{v} = -\nabla_v L_\tau$, the logits obey $\dot{z} = -M_v q$ with $M_v = I_D + J_v J_v^\top \succeq I_D$, and*

$$\frac{d}{dt} L_\tau = -q^\top M_v q = -\|q\|_2^2 - \|J_v^\top q\|_2^2. \tag{16}$$

For every $T > 0$, $\min_{0\le t\le T} \|q(t)\|_2^2 \le (L_\tau(0) - E_\star)/T$. If $\|J_v^\top q\|_2^2 \ge \beta\|q\|_2^2$ throughout the interval, the right-hand side improves by the factor $(1 + \beta)^{-1}$.

Equation 16 is a pointwise comparison at the same logit state z, not a claim that two different trajectories reach the global solution in the same time. The extra graph contribution is the squared alignment ratio β_t in Eq. 14. It is not identified by comparing terminal energies of residual, GNN-only, and free parameterizations, which follow different trajectories.

Corollary 4.7. *[Discrete-time stationarity] Suppose L_τ has H-Lipschitz gradient on a convex region containing each segment between K consecutive joint gradient-descent iterates, and use $0 < \eta \le H^{-1}$. Then $\min_{0\le k<K} \|\nabla \ell_\tau(z_k)\|_2^2 \le 2(L_\tau(u_0, v_0) - E_\star)/(\eta K)$. If $\|J_{v_k}^\top q_k\|_2^2 \ge \beta \|q_k\|_2^2$ at all iterates, the right-hand side improves by $(1+\beta)^{-1}$.*

Proposition 4.8. *[Four-hop locality of a two-layer branch] Partition $v = (e, \omega)$ into node embeddings and shared GNN weights. If two message-passing layers are used, then $M_v = I_D + J_e J_e^\top + J_\omega J_\omega^\top$, and the node block $(J_e J_e^\top)_{ik}$ is zero whenever $\mathrm{dist}_G(i,k) > 4$. The embedding metric is therefore graph local; the shared-weight term can add a global structured coupling.*

Finite critical points are saddles.

Theorem 4.9. *[No finite-logit local extrema] If E is not constant on $\mathcal{X}$, neither ℓ_τ nor residual L_τ has a finite local minimum or maximum. A finite logit z is critical exactly when every $g_i(\cdot; p_{-i})$ is constant. Every such point is a topological saddle: every neighborhood contains both lower- and higher-valued points. If its Hessian is nonzero, the Hessian has at least one positive and one negative eigenvalue. Residual lifting preserves these signs and adds flat fiber directions.*

The absence of finite local minima does not imply global convergence: the infimum normally lies at diverging logits, saddle manifolds may be degenerate, and discrete coordinate-wise minima reappear on the boundary of $\mathcal{P}$.

MAP recovery and local convergence. Assume in this subsection that $x^\star$ is the unique MAP assignment. Define

$$\Gamma = \min_{x \ne x^\star} E(x) - E_\star, \qquad R_E = \max_x E(x) - E_\star, \qquad \kappa = \min_i \min_{a \ne x_i^\star} E(a, x_{-i}^\star) - E_\star. \tag{17}$$

Theorem 4.10. *[Gap thresholds for exact decoding] Let $\Delta(p) = F(p) - E_\star$. Then $\Pr_{X\sim\prod_i p_i}(X \ne x^\star) \le \Delta(p)/\Gamma$. If $\Delta(p) < \Gamma$, conditional-expectation rounding returns $x^\star$ for every variable order. If $\Delta(p) < \Gamma/2$, node-wise argmax also returns $x^\star$ uniquely.*

The first threshold is useful even though every finite-softmax marginal remains strictly fractional: once the relaxed objective falls below the second-best discrete energy, the safe decoder is already exact.

Theorem 4.11. *[Small-gap convergence of residual gradient flow] Let $\varepsilon_0 = \min\{\Gamma/2, \kappa\Gamma/(4nR_E)\}$ and $c_0 = \kappa^2/(16\tau^2 n R_E^2)$. If $\Delta(0) \le \varepsilon_0$, then residual gradient flow remains in this region and $\Delta(t) \le \Delta(0)/(1 + c_0\Delta(0)t)$. If $\|J_v^\top q\|_2^2 \ge \beta\|q\|_2^2$ along the trajectory, replace c_0 by $(1+\beta)c_0$. Conditional-expectation decoding equals $x^\star$ throughout this region, and the induced product $Q_t = \prod_i p_i(t)$ satisfies $\|Q_t - \delta_{x^\star}\|_{\mathrm{TV}} \le \Delta(0)/(\Gamma(1 + c_0\Delta(0)t))$.*

The rate is local and has conservative constants. It does not assume convexity or bounded logits.

Corollary 4.12. *[Small-gap rate for joint gradient descent] Under the assumptions of Theorem 4.11, let joint gradient descent update (u, v) with step size $0 < \eta \le H^{-1}$. Suppose L_τ has H-Lipschitz gradient on every iterate segment. If $\Delta_0 \le \varepsilon_0$, then all iterates remain in the same small-gap region and $\Delta_k \le \Delta_0/(1 + (\eta c_0/2)\Delta_0 k)$. If $\|J_{v_k}^\top q_k\|_2^2 \ge \beta\|q_k\|_2^2$ at every nonstationary iterate, replace c_0 by $(1+\beta)c_0$. The total-variation and exact conditional-decoding conclusions of Theorem 4.11 hold with the discrete rate in place of the continuous-time bound.*

A posteriori discrete certificates. If a complete solver, LP relaxation, or dual method supplies any lower bound $B \le E_\star$, every decoded assignment x has the certificate $0 \le E(x) - E_\star \le E(x) - B$. Combining neural incumbents with classical lower bounds is therefore more informative than comparing final energies alone.

For explicit dense factors, one objective/gradient evaluation scales with the total table size $\sum_C q_C$, where $q_C = \prod_{i\in C} s_i$; the input alone requires $\Theta(\sum_C q_C)$ memory. Sparse, low-rank, Potts, cardinality, and logical factors can be cheaper when contracted by specialized kernels. Full derivations appear in Appendix E.

Table 1: Representative pairwise synthetic instances. Discrete energy; lower is better. REMAP is the best of five seeds. Full best-of-five and mean ± std tables are Table 6 and Table 7 in Appendix F. Best in bold.

Graph	#Nodes/#Edges	LBP	TRBP	Toulbar2	REMAP
P_potts_1	1k/7591	-22215.700	-21365.800	$\mathbf{-22646.529}$	-21451.025
P_potts_5	50k/249624	25668.500	35389.000	12468.172	$\mathbf{11466.507}$
P_random_1	1k/7540	$\mathbf{-4901.100}$	-4505.020	-4900.759	-4564.763
P_random_5	50k/249374	12478.300	13337.000	12835.994	$\mathbf{11750.969}$

Table 2: Synthetic high-order MRFs. Energy values; lower is better. "NA": no solution within the time limit. Best in bold.

Graph	#Nodes/#Cliques	Toulbar2	SRMP	REMAP	
				Energy	Loss
H_Instances_1	500/41253	NA	-5785.093	$\mathbf{-7866.214 \pm 389.207}$	-7859.68 ± 393.719
H_Instances_2	500/57934	NA	-18504.788	$\mathbf{-20260.289 \pm 143.276}$	-20286.571 ± 143.624
H_Instances_3	1000/36993	NA	-5903.131	$\mathbf{-7232.648 \pm 337.393}$	-7229.483 ± 336.218

5 EXPERIMENT

Throughout this section, REMAP denotes the residual parameterization $z = u + \rho_v(G)$ of Eq. 4. Energy and Loss report the best discrete incumbent and best relaxed objective over the run, respectively. rwise baselines are LBP, TRBP (Wainwright et al., 2005b), and Toulbar2 (De Givry, 2023); high-order comparisons use Toulbar2 and SRMP (Kolmogorov, 2015). UAI files are unnormalized positive potentials converted by $\theta = -\log \phi$; the file layout is in Appendix L.

5.1 SYNTHETIC PROBLEMS

Synthetic graphs are Erdős–Rényi (Erdös & Rényi, 1959). Pairwise and high-order instances use either Potts factors

$$\theta_{ij} = \alpha \mathbb{I}(x_i = x_j) + \beta \tag{18}$$

or unary/pairwise values sampled uniformly from $[0.2, 3.0]$. Nodes have 2–6 labels; for Potts models, $\alpha, \beta \in [10^{-5}, 10^{3}]$. LBP/TRBP use 60 iterations and damping 0.1; Toulbar2 uses a 18000s limit. Residual REMAP uses a 5-layer GraphSAGE branch with $d_l = 1024$, learning rate 10^{-4}, at most 150 iterations, early stopping (absolute tolerance 10^{-4}, patience 10), and five random seeds. The two-layer theory in Section 4 is used only for the locality statement.

Pairwise instances. Table 1 shows one small and one large graph for each energy model, using the best REMAP energy over five seeds. The full best-of-five table is Table 6 in Appendix F; mean ± std over the same trials is Table 7 in Appendix F. On 1k-node graphs the classical solvers are stronger or comparable; on the 50k-node instances REMAP attains the lowest energy on both models. The gap appears as size and density grow: residual lifting is most useful on the large, hard pairwise graphs where LBP, TRBP, and Toulbar2 degrade.

Higher-order instances. Table 2 reports three generated Potts high-order graphs. Clique counts include pairwise edges. REMAP obtains lower energy than Toulbar2, which returns no solution within the time limit, and SRMP.

5.2 UAI 2022 INFERENCE COMPETITION DATASETS

We evaluate UAI 2022 MPE instances under the competition budgets of 1200s and 3600s. REMAP uses an 8-layer residual GNN, at most 100 iterations, and embedding widths $\{64, 512, 1024, 4096, 8192\}$. Pairwise numbers are in Table 8 in Appendix G; the width sweep is Table 10 in Appendix G. On easy Segmentation instances Toulbar2 reports optima. On harder ProteinFolding and Grids instances REMAP improves over LBP/TRBP but does not overtake Toulbar2. High-order numbers are in Table 9 in Appendix G: unlike Table 2, Toulbar2 attains lower energy than REMAP on all five reported UAI high-order instances at both budgets.

Table 3: Real PCI instances. Energy values; lower is better. Best in bold.

Graph	#Nodes/#Edges	LBP	TRBP	Toulbar2	REMAP	
					Energy	Loss
PCI_1	30/165	20.344	20.455	**18.134**	18.372 ± 0.161	18.373 ± 0.160
PCI_2	40/311	**98.364**	98.762	**98.364**	98.555 ± 0.109	98.555 ± 0.109
PCI_3	80/1522	**1003.640**	**1003.640**	**1003.640**	$\mathbf{1003.640 \pm 0.0}$	1003.639 ± 0.0
PCI_4	286/10714	585.977	585.977	426.806	$\mathbf{410.945 \pm 2.009}$	410.996 ± 2.014
PCI_5	929/29009	1591.590	1591.590	1118.097	$\mathbf{1074.617 \pm 5.501}$	1074.676 ± 5.503

Table 4: Parameterization ablation on the five real PCI instances. Discrete energy (mean $\pm$ std over five seeds); lower is better.

Instance	#Nodes/#Edges	Residual $u + \rho_v$	GNN-only ρ_v	Free logits u
PCI_1	30/165	$\mathbf{18.372 \pm 0.161}$	18.399 ± 0.151	18.773 ± 0.000
PCI_2	40/311	$\mathbf{98.555 \pm 0.109}$	98.733 ± 0.159	98.874 ± 0.000
PCI_3	80/1522	$\mathbf{1003.640 \pm 0.000}$	1004.789 ± 1.241	$\mathbf{1003.640 \pm 0.000}$
PCI_4	286/10714	$\mathbf{410.945 \pm 2.009}$	413.458 ± 5.166	448.274 ± 0.000
PCI_5	929/29009	$\mathbf{1074.617 \pm 5.501}$	1092.796 ± 13.663	1286.412 ± 0.000

5.3 REAL-WORLD CHALLENGE: PHYSICAL CELL IDENTITY

PCI assignment in LTE/5G is converted to a pairwise MRF in Appendix M. Table 3 reports five real instances from a Chinese city, using the synthetic protocol with 100 iterations and a 3600s Toulbar2 limit. Toulbar2 is exact on the small graphs; REMAP is strongest on PCI_4 and PCI_5. Five larger synthetic PCI instances are in Table 17 in Appendix M.

5.4 ANALYSIS AND ABLATION STUDY

Table 4 compares residual $z = u + \rho_v(G)$, GNN-only $z = \rho_v(G)$, and free logits $z = u$ on the five real PCI instances, using $F(p)$, the same rounding, and—when a GNN is present—a 5-layer GraphSAGE branch, 100 iterations, and learning rate 10^{-4}. Residual improves over GNN-only, which improves over free logits; the gap is concentrated on PCI_4 and PCI_5. This compares terminal discrete energies on three trajectories, not residual alignment β_t in Theorem 4.6.

Anytime PCI comparison with Toulbar2 is Table 12 in Appendix J. Backbone and optimizer controls are Fig. 5 and Fig. 6 in Appendix J. Grid-graph wall-clock scaling is Fig. 8 in Appendix J. Quality-versus-iteration curves are shown in Fig. 3 in Appendix H; GNN-parameter landscapes are Fig. 4 in Appendix J.

6 CONCLUSION

Residual REMAP composes an exact multilinear MAP objective with a graph-structured optimization geometry. The free-logit bypass gives a global normal form: neural lifting changes neither feasible logits nor the landscape, but joint optimization induces the adaptive metric $I + JJ^\top$. This yields exact dissipation and stationarity guarantees, four-hop locality for a two-layer branch, a saddle characterization of every finite critical point, and margin-dependent exact decoding and local convergence. Empirically, REMAP is especially useful on larger, denser, and generated high-order instances, while Toulbar2 remains stronger on many small or standard UAI cases. On the five real PCI instances, a matched-protocol ablation of terminal discrete energy finds residual REMAP improving over GNN-only and free logits, with the gap concentrated on the two larger cases. That comparison does not identify the squared alignment ratio $\beta_t = \|J_v^\top q\|_2^2 / \|q\|_2^2$ of Theorem 4.6, which measures extra dissipation at a shared logit state rather than the endpoints of three different trajectories.

AI USE DISCLOSURE

In this work, we used generative AI tools for assisting with the verification of mathematical proofs. We have not used generative AI tools for any other tasks with required disclosure, and the remaining required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for editing the paper to improve readability and for editing software code. We have reviewed all AI-assisted work. In particular, the AI-assisted code modifications were verified and tested for correctness by 2 authors. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Talal Almutiri and Farrukh Nadeem. Markov models applications in natural language processing: A survey. *International Journal of Information Technology and Computer Science*, 14:1–16, 04 2022.

Waleed Ammar, Chris Dyer, and Noah A Smith. Conditional random field autoencoders for unsupervised structured prediction. *Advances in Neural Information Processing Systems*, 27, 2014.

Shivvrat Arya, Tahrima Rahman, and Vibhav Gogate. Neural network approximators for marginal map in probabilistic circuits. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38 (10):10918–10926, Mar. 2024a. doi: 10.1609/aaai.v38i10.28966.

Shivvrat Arya, Tahrima Rahman, and Vibhav Gogate. Learning to solve the constrained most probable explanation task in probabilistic graphical models. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li (eds.), *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 2791–2799. PMLR, 02–04 May 2024b.

Shivvrat Arya, Tahrima Rahman, and Vibhav Giridhar Gogate. Sine: Scalable mpe inference for probabilistic graphical models using advanced neural embeddings. In Yingzhen Li, Stephan Mandt, Shipra Agrawal, and Emtiyaz Khan (eds.), *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pp. 4465–4473. PMLR, 03–05 May 2025. URL https://proceedings.mlr.press/v258/arya25a.html.

Egon Balas, Sebastián Ceria, and Gérard Cornuéjols. A lift-and-project cutting plane algorithm for mixed 0-1 programs. *Math. Program.*, 58:295–324, 05 1993. doi: 10.1007/BF01581273.

Quentin Cappart, Didier Chételat, Elias Khalil, Andrea Lodi, Christopher Morris, and Petar Veličković. Combinatorial optimization and reasoning with graph neural networks. *Journal of Machine Learning Research*, 24(130):1–61, 2023.

Zijun Cui, Hanjing Wang, Tian Gao, Kartik Talamadupula, and Qiang Ji. Variational message passing neural network for maximum-a-posteriori (map) inference. In James Cussens and Kun Zhang (eds.), *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*, pp. 464–474. PMLR, 01–05 Aug 2022.

Simon De Givry. toulbar2, an exact cost function network solver. In *24ème édition du congrès annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision ROADEF 2023*, 2023.

P Erdös and A Rényi. On random graphs i. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.

Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient belief propagation for early vision. *International Journal of Computer Vision*, 70:41–54, 2004.

Brendan Frey and David Mackay. A revolution: Belief propagation in graphs with cycles. 08 2002.

Elīza Gaile, Andis Draguns, Emils Ozolins, and Karlis Freivalds. Unsupervised training for neural tsp solver, 07 2022.

Víctor Garcia Satorras and Max Welling. Neural enhanced belief propagation on factor graphs. In Arindam Banerjee and Kenji Fukumizu (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 685–693. PMLR, 13–15 Apr 2021.

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

Alexander T. Ihler, John W. Fisher III, and Alan S. Willsky. Loopy belief propagation: Convergence and effects of message errors. *Journal of Machine Learning Research*, 6(31):905–936, 2005. URL http://jmlr.org/papers/v6/ihler05a.html.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference for Learning Representations (ICLR)*, 2015.

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

Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

V. Kolmogorov. Convergent Tree-Reweighted Message Passing for Energy Minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1568–1583, October 2006.

Vladimir Kolmogorov. A new look at reweighted message passing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(5):919–930, 2015. doi: 10.1109/TPAMI.2014.2363465.

Jonathan Kuck, Shuvam Chakraborty, Hao Tang, Rachel Luo, Jiaming Song, Ashish Sabharwal, and Stefano Ermon. Belief propagation neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 667–678. Curran Associates, Inc., 2020.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Neural Information Processing Systems*, 2018.

Jerry Chun-Wei Lin, Yinan Shao, Ji Zhang, and Unil Yun. Enhanced sequence labeling based on latent variable conditional random fields. *Neurocomputing*, 403:431–440, 2020.

D. Khuê Lê-Huu and Nikos Paragios. Continuous relaxation of map inference: A nonconvex perspective. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Yimeng Min, Yiwei Bai, and Carla P Gomes. Unsupervised learning for solving the travelling salesman problem. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 47264–47278. Curran Associates, Inc., 2023.

Vinod Nair, Sergey Bartunov, Felix Gimeno, Ingrid von Glehn, Pawel Lichocki, Ivan Lobov, Brendan O'Donoghue, Nicolas Sonnerat, Christian Tjandraatmadja, Pengming Wang, Ravichandra Addanki, Tharindi Hapuarachchi, Thomas Keck, James Keeling, Pushmeet Kohli, Ira Ktena, Yujia Li, Oriol Vinyals, and Yori Zwols. Solving mixed integer programs using neural networks, 2021.

Yun Peng, Byron Choi, and Jianliang Xu. Graph learning for combinatorial optimization: A survey of state-of-the-art. *Data Science and Engineering*, 6, 06 2021. doi: 10.1007/s41019-021-00155-3.

Zachary Pitkow, Yashar Ahmadian, and Ken Miller. Learning unbelievable probabilities. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper_files/paper/2011/file/cee631121c2ec9232f3a2f028ad5c89b-Paper.pdf.

Meng Qu, Yoshua Bengio, and Jian Tang. GMNN: Graph Markov neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5241–5250. PMLR, 09–15 Jun 2019.

Martin J. A. Schuetz, J. Kyle Brubaker, and Helmut G. Katzgraber. Combinatorial optimization with physics-inspired graph neural networks. *Nature Machine Intelligence*, 4(4):367–377, April 2022.

Wenkang Su, Jiangqun Ni, Xianglei Hu, and Jessica Fridrich. Image steganography with symmetric embedding using gaussian markov random field model. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(3):1001–1015, 2021.

Shixiang Tang, Dapeng Chen, Lei Bai, Kaijian Liu, Yixiao Ge, and Wanli Ouyang. Mutual crf-gnn for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2329–2339, June 2021.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adrian Romero, P. Lio, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018.

Martin J. Wainwright, Tommi S. Jaakkola, and Alan S. Willsky. Map estimation via agreement on (hyper)trees: Message-passing and linear programming, 2005a. URL https://arxiv.org/abs/cs/0508070.

M.J. Wainwright, T.S. Jaakkola, and A.S. Willsky. Map estimation via agreement on trees: message-passing and linear programming. *IEEE Transactions on Information Theory*, 51(11):3697–3717, 2005b.

Chaohui Wang, Nikos Komodakis, and Nikos Paragios. Markov random field modeling, inference & learning in computer vision & image understanding: A survey. *Computer Vision and Image Understanding*, 117(11):1610–1627, 2013.

Y. Weiss and W.T. Freeman. On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs. *IEEE Transactions on Information Theory*, 47(2):736–744, 2001. doi: 10.1109/18.910585.

Xuan Wu, Di Wang, Lijie Wen, Yubin Xiao, Chunguo Wu, Yuesong Wu, Chaoyu Yu, Douglas L. Maskell, and You Zhou. Neural combinatorial optimization algorithms for solving vehicle routing problems: A comprehensive survey with perspectives, 2024.

Yongji Wu, Defu Lian, Yiheng Xu, Le Wu, and Enhong Chen. Graph convolutional networks with markov random field reasoning for social spammer detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):1054–1061, Apr. 2020.

Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *International conference on machine learning*, pp. 5453–5462, 2018.

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

Chaolong Ying, Xinjian Zhao, and Tianshu Yu. Boosting graph pooling with persistent homology. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

KiJung Yoon, Renjie Liao, Yuwen Xiong, Lisa Zhang, Ethan Fetaya, Raquel Urtasun, Richard Zemel, and Xaq Pitkow. Inference in probabilistic graphical models by graph neural networks, 2019.

Tianshu Yu, Runzhong Wang, Junchi Yan, and Baoxin Li. Learning deep graph matching with channel-independent embedding and hungarian attention. In *International conference on learning representations*, 2019.

Elkafi Hassini Yunfei Ma, Amir Amiri and Saiedeh Razavi. Transportation data visualization with a focus on freight: a literature review. *Transportation Planning and Technology*, 45(4):358–401, 2022.

Zhen Zhang, Fan Wu, and Wee Sun Lee. Factor graph neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 8577–8587. Curran Associates, Inc., 2020.

Table 5: Methodological positioning. "Per instance" means that optimization variables are fitted for the test instance.

Method	Use	Variables/factors	Search parameterization	Decoding
Nonconvex MAP relaxation (Lê-Huu & Paragios, 2018)	Per instance	Categorical; arbitrary order	Free product-simplex variables; multilinear	Guaranteed non-increasing block rounding
PI-GNN (Schuetz et al., 2022)	Per instance	Primarily binary QUBO / polynomial	GNN optimizing a relaxed Hamiltonian	Projection to a discrete assignment
SSMP (Arya et al., 2024a)	Amortized + refinement	General PGM / MPE queries	Neural query predictor with self-supervised refinement	Threshold/argmax-style assignment
SINE (Arya et al., 2025)	Amortized inference	MPE queries	Structure- and parameter-aware neural embeddings	Oracle-assisted or neighborhood decoding
Residual REMAP (this work)	Per instance	Heterogeneous categorical; explicit arbitrary-order factors	Free logits plus a GNN residual on the multilinear energy	Conditional-expectation rounding

A Related work

Classical MAP inference. MAP inference in MRFs is NP-hard in general (Koller & Friedman, 2009). Practical methods include variants of belief propagation (Weiss & Freeman, 2001; Felzenszwalb & Huttenlocher, 2004; Frey & Mackay, 2002; Ihler et al., 2005), tree-reweighted message passing (Wainwright et al., 2005b; Kolmogorov, 2006; Wainwright et al., 2005a), SRMP for high-order MRFs (Kolmogorov, 2015), and complete solvers such as Toulbar2 (De Givry, 2023). Lê-Huu and Paragios (Lê-Huu & Paragios, 2018) already established tightness of the same multilinear relaxation and proposed block-coordinate and ADMM algorithms. Exactness of $F(p)$ is therefore not a novelty of REMAP; the contribution is the residual neural parameterization and its landscape/preconditioning analysis.

Unsupervised GNNs for combinatorial optimization. Graph Neural Networks have demonstrated their power in optimization (Yu et al., 2019; Ying et al., 2024), with recent unsupervised GNN methods capturing combinatorial structure (Peng et al., 2021). This line includes TSP (Gaile et al., 2022; Min et al., 2023), VRP (Wu et al., 2024), SAT (Cappart et al., 2023), and Max Independent Set / Max Cut (Schuetz et al., 2022). PI-GNN (Schuetz et al., 2022) optimizes a relaxed combinatorial objective through a GNN on each instance, primarily for binary QUBO and polynomial extensions. Residual REMAP instead targets heterogeneous categorical variables with explicit arbitrary-order factors, and adds a free-logit residual so that the lifting is a global coordinate change rather than a GNN-only parameterization.

Learning for MRFs. Supervised neural inference remains common. GNNs were used for MRF inference by (Yoon et al., 2019) on graphs of up to 16 nodes. Neural Enhanced BP (Garcia Satorras & Welling, 2021), BPNN (Kuck et al., 2020), a neural max-product method (Zhang et al., 2020), and variational message-passing networks (Cui et al., 2022) learn reusable inference operators or message updates across a distribution of instances. CRF-style combinations with GNNs have also been studied (Qu et al., 2019; Tang et al., 2021). These methods answer a different systems question from fitting a fresh optimizer to one instance.

Self-supervised alternatives include learning-to-optimize for MRFs (Nair et al., 2021), ALM-based CMPE models (Arya et al., 2024b), SSMP (Arya et al., 2024a), and SINE (Arya et al., 2025). SSMP combines a model/query predictor with inference-time refinement and teacher–student training; SINE improves structural embeddings and decoding for MPE. Consequently, neither "a neural network optimizes the graphical-model objective" nor "no MAP labels are needed" is a standalone novelty claim. Whether graph message passing, rather than the multilinear objective or node-specific parameters, improves optimization is tested by the residual / GNN-only / free-logit controls in Section 5.4.

Table 5 summarizes the methodological positioning.

B Padding procedure

The schematic diagram of the padding procedure is in Fig. 2. Padding equalizes tensor ranks so that a batched GNN can share one last dimension $|\mathcal{X}|$; it is not a substitute for masking. In this example, $|\mathcal{X}| = 5$ and the unary energy of x_i is $\phi(x_i) = \{1, 1, 3\}$. The implementation fills dummy slices with the original table maximum (here 3), shown in orange. Clique tables are padded analogously: a $3 \times 3 \times 4$ factor becomes a $5 \times 5 \times 5$ tensor whose extra slices are filled with the original maximum (here 4). After invalid logits are masked to $-\infty$, those placeholder entries are multiplied by zero probability and do not enter $F(p)$.

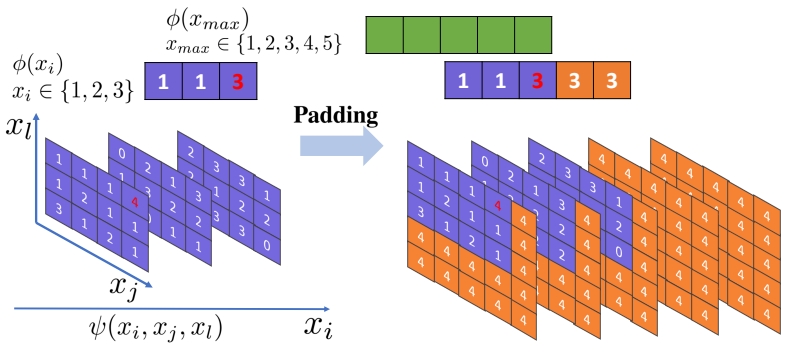


Figure 2: This illustrates the padding procedure for unary loss terms $\phi(x)$ and clique loss terms $\psi(x_i, x_j, x_k)$, with $|\mathcal{X}| = 5$. x_{max} denotes the variable that has the maximum value range. The elements shown in purple represent the energy values in the original ϕ and ψ. After padding, the dimension of vector ϕ, as well as each dimension of the energy tensor $\psi(x_i, x_j, x_k)$, will be 5. Orange entries are a layout placeholder (implementation: $\max(\phi)$ or $\max(\psi)$); after $-\infty$ masking they do not enter $F(p)$.

C Proofs of Section 4

We use the Euclidean topology on the finite-dimensional polytope $\mathcal{P}$. For a label $a \in \mathcal{X}_i$, let $e_{i,a} \in \Delta_i$ denote the corresponding one-hot vector.

C.1 Proof of Lemma C.1

Lemma C.1 (Block-affine decomposition)**.** *For every $i \in \mathcal{V}$, every $p \in \mathcal{P}$, and every $a \in \mathcal{X}_i$, $F(e_{i,a}, p_{-i}) = g_i(a; p_{-i})$ and $F(p) = \sum_a p_i(a) g_i(a; p_{-i})$. In particular, $\min_a F(e_{i,a}, p_{-i}) \leq F(p)$.*

Proof. Fix i and hold p_{-i} fixed. Splitting the expectation according to the value of X_i gives

$$\mathbb{E}[E(X)] = \sum_a p_i(a)\, \mathbb{E}_{X_{-i} \sim \prod_{j \neq i} p_j}[E(a, X_{-i})] = \sum_a p_i(a) g_i(a; p_{-i}).$$

Replacing p_i by $e_{i,a}$ leaves p_{-i} unchanged and forces $X_i = a$. The second identity is a convex combination of the vertex values. □

C.2 Proof of Theorem 4.1

Theorem 4.1. *[Expectation identity, exact value, and minimizers] Let $X_i \sim p_i$ independently. For every $p \in \mathcal{P}$,*

$$F(p) = \mathbb{E}_{X \sim \prod_i p_i}[E(X)] \geq E_\star. \tag{11}$$

Moreover, $\min_{p \in \mathcal{P}} F(p) = E_\star$. Equality $F(p) = E_\star$ holds if and only if every assignment in $\operatorname{supp}(\prod_i p_i)$ is globally optimal. In particular, if the MAP assignment is unique, the unique relaxed minimizer is its one-hot encoding.

Proof. Define the product mass $q_p(x) = \prod_i p_i(x_i)$. Expanding E factor by factor and using independence yields $F(p) = \mathbb{E}[E(X)]$. Since $E(x) \geq E_\star$ for every x, averaging gives $F(p) \geq E_\star$. Conversely, the one-hot encoding of any MAP assignment $x^\star$ is a point mass, so $F(p) = E(x^\star) = E_\star$. The gap decomposition $F(p) - E_\star = \sum_x q_p(x)(E(x) - E_\star)$ has nonnegative summands, hence vanishes iff every x in $\operatorname{supp}(q_p)$ is optimal. If $x^\star$ is unique, the support must be the singleton $\{x^\star\}$, so every marginal is the corresponding one-hot vector. □

C.3 Proof of Corollary 4.2

Corollary 4.2. *[Conditional-expectation rounding] For every $p \in \mathcal{P}$, a deterministic assignment x_{CE} can be constructed sequentially such that $E(x_{\mathrm{CE}}) \le F(p)$. At step i, set a label minimizing the expected energy when that label and all earlier choices are fixed while unrounded variables retain their current marginals. The guarantee holds for every variable order and arbitrary finite factors.*

Proof. Fix an arbitrary variable order $(i_1, \dots, i_n)$. Set $p^{(0)} = p$. After $k-1$ blocks have been made one-hot, choose $a_k \in \arg\min_a F(e_{i_k,a}, p^{(k-1)}_{-i_k})$ and replace block i_k by e_{i_k,a_k}. By Lemma C.1 the selected vertex value is no larger than the current convex combination, so $F(p^{(k)}) \le F(p^{(k-1)})$. After n steps, $p^{(n)}$ is the point mass at x_{CE} and $E(x_{\mathrm{CE}}) = F(p^{(n)}) \le F(p)$. □

C.4 Proof of Proposition 4.3

Proposition 4.3. *[Argmax-rounding deviation] Let $\hat{x}_i \in \arg\max_a p_i(a)$. Then*

$$\big|E(\hat{x}) - F(p)\big| \le \sum_{C \in \mathcal{C}} \mathrm{osc}(\theta_C)\Big(1 - \prod_{i \in C} p_i(\hat{x}_i)\Big) \le \sum_{C \in \mathcal{C}} \mathrm{osc}(\theta_C) \sum_{i \in C} \big(1 - p_i(\hat{x}_i)\big). \tag{12}$$

Thus the argmax gap is controlled by both distributional concentration and factor ranges.

Proof. If Y is finite-valued and h is real, $|h(y) - \mathbb{E}[h(Y)]| \le \mathrm{osc}(h) \Pr(Y \ne y)$. Apply this to $Y = X_C$, $y = \hat{x}_C$, and $h = \theta_C$, then sum over factors. Independence gives $\Pr(X_C = \hat{x}_C) = \prod_{i \in C} p_i(\hat{x}_i)$; the union bound yields the second inequality. □

C.5 Proof of Theorem 4.4

Theorem 4.4. *[Finite-softmax infimum and a quantitative approximation] Let $z = (z_i)_i$ be free finite logits and $p_i = \mathrm{softmax}(z_i/\tau)$ for fixed $\tau > 0$. Then $\inf_z F(p(z)) = E_\star$. The infimum is attained at finite z if and only if every assignment is globally optimal. More quantitatively, let $x^\star$ be optimal, let $R_E = \max_x E(x) - E_\star$, and set $z_i(x_i^\star) = m$, $z_i(a) = 0$ for $a \ne x_i^\star$. Then*

$$0 \le F(p(z)) - E_\star \le R_E \sum_{i=1}^{n} (s_i - 1) e^{-m/\tau}. \tag{13}$$

Proof. Finite softmax maps onto the relative interior of each simplex via $z_i(a) = \tau \log p_i(a)$, and cannot reach the boundary vertices. Interior product distributions can approach a MAP one-hot encoding, so $\inf_z F(p(z)) = E_\star$ by continuity and Theorem 4.1; the minimum of F over the closed polytope $\mathcal{P}$ is attained at vertices, but that minimum is generally not attained at finite z. Every finite-logit product has full support, so $F(p) = E_\star$ is attained at finite logits iff every assignment is optimal. For the quantitative bound, the stated logits satisfy $1 - p_i(x_i^\star) \le (s_i - 1)e^{-m/\tau}$; a union bound and $0 \le E(x) - E_\star \le R_E$ give the claim. □

C.6 Proof of Lemma C.2

Lemma C.2 (Logit-gradient identity)**.** *For finite z, node i, and label $b \in \mathcal{X}_i$, $\partial \ell_\tau / \partial z_i(b) = (p_i(b)/\tau)\,(g_i(b; p_{-i}) - F(p))$. Consequently, $\nabla \ell_\tau(z) = 0$ iff every $g_i(\cdot; p_{-i})$ is constant.*

Proof. Use $\partial p_i(a)/\partial z_i(b) = \tau^{-1} p_i(a)(\mathbf{1}\{a = b\} - p_i(b))$ and $F = \sum_a p_i(a) g_i(a; p_{-i})$. Every finite-softmax probability is positive, so the block gradient vanishes exactly when all conditional costs equal their mean $F(p)$. □

C.7 Proof of Theorem 4.5

Theorem 4.5. *[Global lossless-lifting equivalence] Define $T(u, v) = (u + \rho_v(G), v)$. Then T is a global diffeomorphism with inverse $T^{-1}(z, v) = (z - \rho_v(G), v)$, and $L_\tau \circ T^{-1}(z, v) = \ell_\tau(z)$. Consequently: (i) $\inf_{u,v} L_\tau(u, v) = \inf_z \ell_\tau(z) = E_\star$; (ii) (u, v) is critical, a local minimum, or a*

local maximum of L_τ if and only if $z = u + \rho_v(G)$ has the corresponding property for ℓ_τ; (iii) every sublevel set of L_τ is diffeomorphic to the corresponding free-logit sublevel set times $\mathcal{V}$; (iv) at a critical point, the Hessian of L_τ has the same positive and negative inertia as $\nabla^2 \ell_\tau(z)$, with at least m additional zero directions.

Proof. The stated inverse of T is immediate, and

$$DT = \begin{pmatrix} I & J_v \\ 0 & I_m \end{pmatrix}, \qquad DT^{-1} = \begin{pmatrix} I & -J_v \\ 0 & I_m \end{pmatrix}.$$

Thus T is a global diffeomorphism and substitution gives $L_\tau \circ T^{-1}(z, v) = \ell_\tau(z)$. In these coordinates the objective is independent of v. For fixed v, $u \mapsto u + \rho_v(G)$ is onto $\mathbb{R}^D$, proving equality of infima. Diffeomorphisms preserve neighborhoods, criticality, and local extrema, while $\{L_\tau \le c\} = T^{-1}(\{\ell_\tau \le c\} \times \mathcal{V})$. At a critical point, $\nabla \ell_\tau = 0$, so second derivatives of the coordinate map disappear and $\nabla^2 L_\tau$ is congruent to $\mathrm{diag}(\nabla^2 \ell_\tau, 0)$; Sylvester's law of inertia completes the proof. □

C.8 Proof of Theorem 4.6

Theorem 4.6. *[Induced logit dynamics and dissipation] Along $\dot{u} = -\nabla_u L_\tau$ and $\dot{v} = -\nabla_v L_\tau$, the logits obey $\dot{z} = -M_v q$ with $M_v = I_D + J_v J_v^\top \succeq I_D$, and*

$$\frac{d}{dt} L_\tau = -q^\top M_v q = -\|q\|_2^2 - \|J_v^\top q\|_2^2. \tag{16}$$

For every $T > 0$, $\min_{0 \le t \le T} \|q(t)\|_2^2 \le (L_\tau(0) - E_\star)/T$. If $\|J_v^\top q\|_2^2 \ge \beta \|q\|_2^2$ throughout the interval, the right-hand side improves by the factor $(1+\beta)^{-1}$.

Proof. The chain rule gives $\nabla_u L_\tau = q$ and $\nabla_v L_\tau = J_v^\top q$. Hence

$$\|\nabla_{u,v} L_\tau\|_2^2 = \|q\|_2^2 + \|J_v^\top q\|_2^2 = q^\top (I + J_v J_v^\top) q$$

and $\dot{z} = \dot{u} + J_v \dot{v} = -(I + J_v J_v^\top) q$. Taking the inner product with q proves Eq. 16. Integration yields $\int_0^T \|q(t)\|_2^2 \, dt \le L_\tau(0) - E_\star$; the minimum is no larger than the time average. Under Eq. 14, the integrand is at least $(1+\beta)\|q\|_2^2$, so the bound improves by $(1+\beta)^{-1}$. □

C.9 Proof of Corollary 4.7

Corollary 4.7. *[Discrete-time stationarity] Suppose L_τ has H-Lipschitz gradient on a convex region containing each segment between K consecutive joint gradient-descent iterates, and use $0 < \eta \le H^{-1}$. Then $\min_{0 \le k < K} \|\nabla \ell_\tau(z_k)\|_2^2 \le 2(L_\tau(u_0, v_0) - E_\star)/(\eta K)$. If $\|J_{v_k}^\top q_k\|_2^2 \ge \beta \|q_k\|_2^2$ at all iterates, the right-hand side improves by $(1+\beta)^{-1}$.*

Proof. The descent lemma and $\eta H \le 1$ give $L_\tau(w_{k+1}) \le L_\tau(w_k) - (\eta/2)\|\nabla L_\tau(w_k)\|_2^2$. By the chain rule, $\|\nabla L_\tau(w_k)\|_2^2 = \|q_k\|_2^2 + \|J_{v_k}^\top q_k\|_2^2$. Sum over k, use $L_\tau(w_K) \ge E_\star$, and bound the minimum by the average. Under Eq. 14 the summand is at least $(1+\beta)\|q_k\|_2^2$, which improves the bound by $(1+\beta)^{-1}$. □

C.10 Proof of Proposition 4.8

Proposition 4.8. *[Four-hop locality of a two-layer branch] Partition $v = (e, \omega)$ into node embeddings and shared GNN weights. If two message-passing layers are used, then $M_v = I_D + J_e J_e^\top + J_\omega J_\omega^\top$, and the node block $(J_e J_e^\top)_{ik}$ is zero whenever $\mathrm{dist}_G(i,k) > 4$. The embedding metric is therefore graph local; the shared-weight term can add a global structured coupling.*

Proof. The output at node i depends only on embeddings in its two-hop neighborhood $B_2(i)$, so $\partial \rho_i / \partial e_\ell = 0$ for $\ell \notin B_2(i)$. Therefore $(J_e J_e^\top)_{ik} = \sum_\ell (\partial \rho_i / \partial e_\ell)(\partial \rho_k / \partial e_\ell)^\top$ vanishes when $B_2(i) \cap B_2(k) = \emptyset$, which holds if $\mathrm{dist}_G(i,k) > 4$. Partitioning Jacobian columns by $v = (e, \omega)$ gives the metric decomposition. □

C.11 Proof of Theorem 4.9

Theorem 4.9. *[No finite-logit local extrema] If E is not constant on $\mathcal{X}$, neither ℓ_τ nor residual L_τ has a finite local minimum or maximum. A finite logit z is critical exactly when every $g_i(\cdot; p_{-i})$ is constant. Every such point is a topological saddle: every neighborhood contains both lower- and higher-valued points. If its Hessian is nonzero, the Hessian has at least one positive and one negative eigenvalue. Residual lifting preserves these signs and adds flat fiber directions.*

Proof. Modulo one additive gauge per node, finite softmax is a diffeomorphism onto $\mathrm{relint}(\mathcal{P})$, so it is equivalent to study F at an interior p. By Lemma C.2, criticality makes all first-order feasible derivatives vanish. For $U_i = \{h_i : \mathbf{1}^\top h_i = 0\}$, multi-affinity gives the exact expansion

$$F(p+h) - F(p) = \sum_{\emptyset \neq S \subseteq \mathcal{V}} A_S((h_i)_{i \in S}),$$

with A_S multilinear and all $|S| = 1$ terms zero. If every remaining term vanished, the polynomial would be locally, hence globally, constant, making all vertex values $E(x)$ equal. Choose a nonzero term of smallest order $k \geq 2$ and tangent vectors of value $c \neq 0$. Perturbing only those k blocks, lower-order terms vanish by minimality and higher-order terms require inactive blocks, so the change is exactly ct^k. Flipping one tangent vector changes the sign, and both higher and lower values occur arbitrarily near p. If the Hessian is nonzero, $k = 2$ and the quadratic form takes both signs. Apply Theorem 4.5 for the residual conclusions. □

C.12 Proof of Theorem 4.10

Theorem 4.10. *[Gap thresholds for exact decoding] Let $\Delta(p) = F(p) - E_\star$. Then $\Pr_{X \sim \prod_i p_i}(X \neq x^\star) \leq \Delta(p)/\Gamma$. If $\Delta(p) < \Gamma$, conditional-expectation rounding returns $x^\star$ for every variable order. If $\Delta(p) < \Gamma/2$, node-wise argmax also returns $x^\star$ uniquely.*

Proof. The gap decomposition gives $\Delta(p) \geq \Gamma \Pr(X \neq x^\star)$. If $\Delta < \Gamma$, conditional rounding has energy below $E_\star + \Gamma$ and must equal $x^\star$. If $\Delta < \Gamma/2$, then $q_p(x^\star) > 1/2$, so each marginal factor exceeds $1/2$ and $x_i^\star$ is the unique argmax. □

C.13 Proof of Theorem 4.11

Theorem 4.11. *[Small-gap convergence of residual gradient flow] Let $\varepsilon_0 = \min\{\Gamma/2, \kappa\Gamma/(4nR_E)\}$ and $c_0 = \kappa^2/(16\tau^2 n R_E^2)$. If $\Delta(0) \leq \varepsilon_0$, then residual gradient flow remains in this region and $\Delta(t) \leq \Delta(0)/(1 + c_0\Delta(0)t)$. If $\|J_v^\top q\|_2^2 \geq \beta\|q\|_2^2$ along the trajectory, replace c_0 by $(1+\beta)c_0$. Conditional-expectation decoding equals $x^\star$ throughout this region, and the induced product $Q_t = \prod_i p_i(t)$ satisfies $\|Q_t - \delta_{x^\star}\|_{\mathrm{TV}} \leq \Delta(0)/(\Gamma(1 + c_0\Delta(0)t))$.*

Proof. Let $\delta_i = 1 - p_i(x_i^\star)$ and $S = \sum_i \delta_i$. Theorem 4.10 gives $\delta_i \leq \Delta/\Gamma$, $S \leq n\Delta/\Gamma$, and $\Delta \leq R_E S$. For $a \neq x_i^\star$, let $A_i = \{X_{-i} \neq x_{-i}^\star\}$. Every complete-assignment energy lies in $[E_\star, E_\star + R_E]$, so the range of $x_{-i} \mapsto E(a, x_{-i})$ is at most R_E and

$$g_i(a; p_{-i}) - E(a, x_{-i}^\star) \leq R_E \Pr(A_i) \leq R_E \sum_{j \neq i} \Pr(X_j \neq x_j^\star) \leq R_E S.$$

The same bound holds for $a = x_i^\star$. Since κ gives $E(a, x_{-i}^\star) - E(x_i^\star, x_{-i}^\star) \geq \kappa$,

$$g_i(a; p_{-i}) - g_i(x_i^\star; p_{-i}) \geq \kappa - 2R_E S.$$

Inside the region $\Delta \leq \varepsilon_0$, this is at least $\kappa/2$ and $p_i(x_i^\star) \geq 1/2$. Lemma C.2 at the optimal label and $F - g_i(x_i^\star; p_{-i}) \geq (\kappa/2)\delta_i$ then yield

$$\Big|\frac{\partial \ell_\tau}{\partial z_i(x_i^\star)}\Big| = \frac{p_i(x_i^\star)}{\tau}\big(F(p) - g_i(x_i^\star; p_{-i})\big) \geq \frac{\kappa}{4\tau}\delta_i.$$

Consequently

$$\|q\|_2^2 \geq \frac{\kappa^2}{16\tau^2}\sum_i \delta_i^2 \geq \frac{\kappa^2 S^2}{16\tau^2 n} \geq \frac{\kappa^2}{16\tau^2 n R_E^2}\Delta^2 = c_0\Delta^2.$$

Dissipation keeps the trajectory in the region and gives $\dot{\Delta} \le -c_0\Delta^2$. Under Eq. 14, $\|\nabla_{u,v} L_\tau\|_2^2 \ge (1+\beta)\|q\|_2^2$, so c_0 is replaced by $(1+\beta)c_0$. Integration proves the $O(1/t)$ rate. Exact conditional decoding follows from Theorem 4.10; total variation from a point mass equals the mass outside that point. □

C.14 Proof of Corollary 4.12

Corollary 4.12. *[Small-gap rate for joint gradient descent] Under the assumptions of Theorem 4.11, let joint gradient descent update (u, v) with step size $0 < \eta \le H^{-1}$. Suppose L_τ has H-Lipschitz gradient on every iterate segment. If $\Delta_0 \le \varepsilon_0$, then all iterates remain in the same small-gap region and $\Delta_k \le \Delta_0/(1 + (\eta c_0/2)\Delta_0 k)$. If $\|J_{v_k}^\top q_k\|_2^2 \ge \beta\|q_k\|_2^2$ at every nonstationary iterate, replace c_0 by $(1+\beta)c_0$. The total-variation and exact conditional-decoding conclusions of Theorem 4.11 hold with the discrete rate in place of the continuous-time bound.*

Proof. The descent lemma, $\eta H \le 1$, and the chain rule give

$$\Delta_{k+1} \le \Delta_k - \frac{\eta}{2}\big(\|q_k\|_2^2 + \|J_{v_k}^\top q_k\|_2^2\big) \le \Delta_k - \frac{\eta c_0}{2}\Delta_k^2.$$

Thus the objective cannot leave the small-gap sublevel set. If an iterate attains zero gap, the conclusion is immediate. Otherwise, taking reciprocals yields

$$\frac{1}{\Delta_{k+1}} - \frac{1}{\Delta_k} = \frac{\Delta_k - \Delta_{k+1}}{\Delta_k \Delta_{k+1}} \ge \frac{\eta c_0}{2}.$$

Telescoping proves the $O(1/k)$ rate. Under Eq. 14 the decrement is multiplied by $1+\beta$, so c_0 is replaced by $(1+\beta)c_0$. Theorem 4.10 gives the decoding and total-variation conclusions. □

C.15 Derivation of the a posteriori certificate

Let B be any valid lower bound, so $B \le E_\star \le E(x)$ for every decoded assignment x. Rearrangement gives $0 \le E(x) - E_\star \le E(x) - B$. If $x = x_{\mathrm{CE}}$, Corollary 4.2 additionally gives $E(x_{\mathrm{CE}}) \le F(p)$, which is an upper-bound comparison and does not replace an independently valid lower bound B.

D Relation to Lifting

Our framework aligns with the classical concept of lifting (Balas et al., 1993) by expanding the parameter space while leaving the logit space unchanged. The residual map $T(u, v) = (u + \rho_v(G), v)$ is a global diffeomorphism, so the lifted objective is exactly the free-logit objective times a parameter fiber (Theorem 4.5). The GNN does not create a larger set of logits or a low-energy tunnel in the discrete space; it changes the optimization metric to $I + JJ^\top$.

E Complexity Analysis

For a dense explicit factor C, evaluating its contribution to Eq. 7 enumerates $q_C = \prod_{i \in C} s_i$ table entries. Summing over factors gives the forward arithmetic bound $T_{\mathrm{factor}} = O(\sum_C q_C)$. Reverse-mode differentiation has the same asymptotic arithmetic order. Since the dense tables themselves contain exactly $\sum_C q_C$ scalars, their memory lower bound is $\Theta(\sum_C q_C)$. If $s_i \le s_{\max}$ and $|C| \le r$, then $q_C \le s_{\max}^r$ and the cost is $O(|\mathcal{C}| s_{\max}^r)$; it is not $O(r|\mathcal{C}| s_{\max})$.

Representation structure can reduce this cost. A sparse table with a default value plus m_C exceptional tuples is $O(m_C|C|)$ after the default expectation. A CP factorization $\theta_C(x_C) = \sum_{h=1}^{R_C} \prod_{i \in C} a_{Cih}(x_i)$ contracts in $O(R_C \sum_{i \in C} s_i)$ plus product operations. Potts, cardinality, and logical factors likewise admit specialized kernels.

Let d_h be hidden width, K the number of layers, and E_G the clique-expanded edge set. One dense GraphSAGE layer costs approximately $O(|\mathcal{V}| d_h^2 + |E_G| d_h)$, and storing activations across K layers costs $O(K|\mathcal{V}| d_h)$ before checkpointing. Conditional-expectation rounding need not recompute the full energy for every candidate label: caching incident-factor contractions gives a worst-case sweep cost $O(\sum_C |C| q_C)$ for dense factors.

Table 6: Full pairwise synthetic results. Discrete energy; lower is better. REMAP is the best energy over five seeds. The four-row excerpt in the main text is Table 1. Best in bold.

Graph	#Nodes/#Edges	LBP	TRBP	Toulbar2	REMAP	Graph	#Nodes/#Edges	LBP	TRBP	Toulbar2	REMAP
P_potts_1	1k/7591	-22215.700	-21365.800	**-22646.529**	-21451.025	P_random_1	1k/7540	**-4901.100**	-4505.020	-4900.759	-4564.763
P_potts_2	5k/37439	**-111319.000**	-105848.000	-110022.248	-105952.531	P_random_2	5k/37488	-24059.900	-22934.000	**-24139.194**	-21834.693
P_potts_3	10k/75098	**-221567.000**	-210570.000	-218311.424	-209925.269	P_random_3	10k/74518	-47873.200	-47002.000	**-48107.172**	-42120.325
P_potts_4	50k/248695	12411.200	13454.600	12955.129	**11679.429**	P_random_4	50k/249554	12881.500	14342.300	12233.890	**11769.934**
P_potts_5	50k/249624	25668.500	35389.000	12468.172	**11466.507**	P_random_5	50k/249374	12478.300	13337.000	12835.994	**11750.969**
P_potts_6	50k/300181	17609.800	17362.600	17635.791	**16756.999**	P_random_6	50k/299601	16723.600	16754.500	18031.964	**16700.674**
P_potts_7	50k/299735	**16962.500**	**16962.500**	19532.817	17002.578	P_random_7	50k/299538	**16689.200**	16701.600	18179.548	**16689.252**
P_potts_8	50k/374169	**24552.400**	24596.800	25446.235	**24552.413**	P_random_8	50k/374203	**24556.000**	**24556.000**	25549.594	**24555.995**
P_potts_9	50k/375603	25099.800	25095.600	25502.495	**25050.522**	P_random_9	50k/374959	**24635.600**	24689.500	25908.500	**24640.039**

Table 7: Pairwise synthetic instances, same five trials as Table 6. REMAP Energy and Loss are mean $\pm$ std. Energy is the decoded discrete energy; Loss is the relaxed multilinear objective. Lower is better.

Graph	#Nodes/#Edges	LBP	TRBP	Toulbar2	REMAP	
					Energy	Loss
P_potts_1	1k/7591	-22215.700	-21365.800	-22646.529	-21791.868 $\pm$ 218.106	-21799.268 $\pm$ 216.075
P_potts_2	5k/37439	-111319.000	-105848.000	-110022.248	-105762.092 $\pm$ 434.674	-106016.855 $\pm$ 168.861
P_potts_3	10k/75098	-221567.000	-210570.000	-218311.424	-211406.914 $\pm$ 1489.099	-210182.681 $\pm$ 275.164
P_potts_4	50k/248695	12411.200	13454.600	12955.129	12219.817 $\pm$ 538.969	11811.682 $\pm$ 123.877
P_potts_5	50k/249624	25668.500	35389.000	12468.172	12010.036 $\pm$ 266.610	11673.708 $\pm$ 146.513
P_potts_6	50k/300181	17609.800	17362.600	17635.791	18399.913 $\pm$ 1475.526	16988.347 $\pm$ 163.587
P_potts_7	50k/299735	16962.500	16962.500	19532.817	17480.701 $\pm$ 212.084	17265.434 $\pm$ 140.904
P_potts_8	50k/374169	24552.400	24596.800	25446.235	26115.840 $\pm$ 1677.627	24668.087 $\pm$ 163.587
P_potts_9	50k/375603	25099.800	25095.600	25502.495	26348.525 $\pm$ 1095.319	25189.789 $\pm$ 132.693
P_random_1	1k/7540	-4901.100	-4505.020	-4900.759	-4570.079 $\pm$ 31.228	-4574.664 $\pm$ 31.411
P_random_2	5k/37488	-24059.900	-22934.000	-24139.194	-21774.416 $\pm$ 52.910	-21798.702 $\pm$ 32.389
P_random_3	10k/74518	-47873.200	-47002.000	-48107.172	-41953.991 $\pm$ 237.577	-41972.379 $\pm$ 216.574
P_random_4	50k/249554	12881.500	14342.300	12233.890	12552.252 $\pm$ 30.311	11983.388$\pm$ 213.454
P_random_5	50k/249374	12478.300	13337.000	12835.994	12308.580 $\pm$ 14.045	11945.450 $\pm$ 194.481
P_random_6	50k/299601	16723.600	16754.500	18031.964	17705.219 $\pm$ 435.560	17207.997 $\pm$ 405.217
P_random_7	50k/299538	16689.200	16701.600	18179.548	18343.026 $\pm$ 1448.821	16971.435 $\pm$ 209.021
P_random_8	50k/374203	**24556.000**	**24556.000**	25549.594	25949.446 $\pm$ 995.956	24787.343 $\pm$ 163.587
P_random_9	50k/374959	**24635.600**	24689.500	25908.500	25871.264 $\pm$ 1087.915	24811.354 $\pm$ 171.315

F FULL RESULTS ON PAIRWISE SYNTHETIC INSTANCES

This appendix expands the pairwise synthetic comparison of Section 5.1. Instances are Erdős–Rényi graphs with prefix "P_potts_" for Potts pairwise energies (Eq. 18) and "P_random_" for unary and pairwise values sampled uniformly from $[0.2, 3.0]$. Node cardinalities range from 2 to 6. LBP and TRBP run for 60 iterations with damping 0.1; Toulbar2 uses a default mode with a 18000s limit. Residual REMAP uses a 5-layer GraphSAGE branch, $d_l = 1024$, learning rate 10^{-4}, at most 150 iterations, and early stopping (absolute tolerance 10^{-4}, patience 10).

Table 1 in the main text reports four representative graphs. Table 6 lists the best discrete energy among the same five REMAP seeds on all eighteen instances. Table 7 reports mean $\pm$ standard deviation of Energy and Loss over those five trials. Energy is the decoded discrete energy; Loss is the relaxed multilinear objective. Lower is better for both.

On 1k–10k-node graphs, LBP or Toulbar2 typically attains the lowest energy, and the best REMAP seed remains comparable on Potts instances. From 50k nodes onward, the best REMAP seed is lowest on most Potts and random instances in Table 6. The mean in Table 7 is more conservative: several dense 50k-node graphs have a large seed standard deviation, so a single best seed in Table 6 should not be read as a tight typical value.

G RESULTS OF UAI INFERENCE COMPETITION 2022 DATASET

This appendix reports the UAI 2022 MPE numbers summarized in Section 5. Pairwise and high-order instances use the competition budgets of 1200s and 3600s. LBP and TRBP run for 30 iterations with damping 0.1; Toulbar2 is limited to 1200s in the pairwise table. Residual REMAP uses an 8-layer GraphSAGE branch, at most 100 iterations, and the same UAI-file conversion as Appendix L. Unless noted, the embedding width is 1024; Table 10 varies that width.

Table 8 lists pairwise instances. On Segmentation graphs Toulbar2 reports certified optima (marked "opt"), and LBP/TRBP are already close; REMAP is comparable but not better. On ProteinFolding and Grids instances, which are larger or denser, REMAP improves over LBP and TRBP while remaining above Toulbar2. This is the intended reading of the main-text claim: residual lifting is more useful when message passing degrades, not a replacement for a complete solver on small, well-solved UAI cases.

Table 9 lists five large high-order competition instances. Unlike the generated high-order graphs in Table 2, Toulbar2 attains lower energy than REMAP at both 1200s and 3600s on every reported instance. The shift should be read together with factor order, table representation, and energy range rather than as a uniform high-order advantage.

Table 10 repeats the pairwise suite across embedding widths $\{64, 512, 1024, 4096, 8192\}$. Width changes the attained energy, but a moderate width is already competitive on most graphs; larger embeddings are not uniformly better and cost more memory. The main experiments therefore keep a single width per suite rather than selecting the best column after the fact.

Table 8: Results on the UAI inference competition 2022. Numbers correspond to the obtained energy values. Best in bold."opt" denotes it is the optimal solution.

Graph	#Nodes/#Edges	LBP	TRBP	Toulbar2	REMAP	
					Energy	Loss
ProteinFolding_11	400/7160	-3106.080	-3079.030	-4461.047	-3976.908 ± 52.047	-4018.784 ± 36.491
ProteinFolding_12	250/1848	3570.210	3604.240	3562.387(opt)	16137.682 ± 16.020	16090.801 ± 22.869
Grids_19	1600/3200	-2250.440	-2103.610	-2643.107	-2400.251 ± 20.061	-2398.078 ± 16.010
Grids_21	1600/3200	-13119.300	-12523.300	-18895.393	-16592.926 ± 94.368	-16605.564 ± 113.096
Grids_24	1600/3120	-13210.400	-13260.900	-18274.302	-16323.767 ± 171.950	-16222.104 ± 222.593
Grids_25	1600/3120	-2170.890	-2171.050	-2620.268	-2361.900 ± 10.231	-2361.055 ± 12.678
Grids_26	400/800	-2063.350	-1903.910	-3010.719	-2595.041 ± 43.306	-2577.378 ± 39.370
Grids_27	1600/3120	-9024.640	-9019.470	-12284.284	-10898.595 ± 160.435	-10771.257 ± 170.329
Grids_30	400/760	-2142.890	-2154.910	-2984.248	-2651.035 ± 35.508	-2676.246 ± 19.886
Segmentation_11	228/624	329.950	339.762	312.760 (opt)	432.291 ± 34.208	391.971 ± 40.099
Segmentation_12	231/625	75.867	77.898	51.151 (opt)	90.248 ± 22.655	105.639 ± 21.165
Segmentation_13	225/607	75.299	88.554	49.859 (opt)	80.156 ± 6.462	78.685 ± 18.546
Segmentation_14	231/632	95.619	98.691	92.334 (opt)	102.263 ± 7.169	101.268 ± 5.467
Segmentation_15	229/622	412.990	418.853	380.393 (opt)	417.276 ± 22.357	408.214 ± 27.037
Segmentation_16	228/610	100.853	101.670	95.000 (opt)	102.687 ± 4.571	101.687 ± 7.358
Segmentation_17	225/612	421.888	432.012	407.065 (opt)	445.843 ± 24.459	478.881 ± 32.824
Segmentation_18	235/647	100.389	98.411	82.669 (opt)	104.721 ± 6.489	96.315 ± 6.124
Segmentation_19	228/624	86.589	86.692	58.704 (opt)	96.173 ± 4.731	84.882 ± 10.063
Segmentation_20	232/635	289.435	291.527	262.216 (opt)	335.245 ± 36.163	315.482 ± 24.268

Table 9: Results on high-order cases of the UAI inference competition 2022. Numbers correspond to the obtained energy values. Best in bold.

Graph	#Nodes/#Cliques	Toulbar2 (1200s)	Toulbar2 (3600s)	REMAP	
				Energy	Loss
Maxsat_gss-25-s100	31931/96111	**-145969.060**	-145969.060	-139904.266 ± 1717.483	-139914.341 ± 1710.139
BN-nd-250-5-10	250/250	155.129	**154.610**	189.395 ± 5.424	187.729 ± 4.966
Maxsat_mod4block_2vars_10gates_u2_autoenc	479/123509	-186103.111	-186103.111	-146166.620 ± 41250.035	-146166.797 ± 41249.859
Maxsat_mod2c-rand3bip-sat-240-3.shuffled-as.sat05-2520	339/2416	-3734.627	**-3737.076**	-3511.994 ± 220.654	-3511.822 ± 220.471
Maxsat_mod2c-rand3bip-sat-250-3.shuffled-as.sat05-2535	352/2492	**-3863.259**	-3863.259	-3686.567 ± 166.998	-3686.085 ± 166.499

H MORE ANALYSIS

This section supplements the ablation paragraph in Section 5.4. The parameterization table itself remains in the main text (Table 4); here we only discuss iteration-wise quality and, in Appendix J, landscapes, backbones, optimizers, anytime PCI traces, and grid scaling. **Efficiency vs Solution Quality.** We evaluate the performance of the REMAP using the same network size and a consistent learning rate of 1e-4 on the Segmentation_14 dataset from the UAI 2022 inference competition, along with two of our generated Potts instances: P_potts_6 and P_potts_8. This setup allows us to observe the trends associated with changes in graph size and sparsity. The results are presented in Fig. 3. It is seen that the model converges rapidly when the graph is small and sparse, within approximately 20 iterations on the Segmentation_14 dataset. Comparing P_potts_6 and P_potts_8, we observe that

Table 10: Full results on the UAI inference competition 2022 of REMAP with different feature dimensions. Numbers correspond to the obtained energy values.

Graph	#Nodes/#Edges	dim=64	dim=512	dim=1024	dim=4096	dim=8192
ProteinFolding_11	400/7160	-3892.949	-3886.701	-3946.168	-4065.294	-4003.323
ProteinFolding_12	250/1848	16064.795	16068.406	16051.798	16088.073	16071.324
Grids_19	1600/3200	-2355.159	-2404.975	-2337.281	-2341.2746	-2373.618
Grids_21	1600/3200	-16478.466	-16169.0320	-16446.410	-16209.017	-16278.668
Grids_24	1600/3120	-16008.008	-15900.249	-15841.799	-15608.162	-15948.219
Grids_25	1600/3120	-2343.547	-2353.223	-2319.899	-2306.686	-2288.182
Grids_26	400/800	-2532.837	-2608.395	-2553.781	-2559.572	-2535.464
Grids_27	1600/3120	-10748.024	-10704.057	-10514.857	-10389.031	-10665.737
Grids_30	400/760	-2563.274	-2631.862	-2640.044	-2691.091	-2649.462
Segmentation_11	228/624	330.541	349.906	334.882	356.895	337.312
Segmentation_12	231/625	74.705	74.029	155.062	79.151	105.801
Segmentation_13	225/607	67.371	86.064	69.430	72.394	112.516
Segmentation_14	231/632	94.192	96.501	100.582	104.091	96.572
Segmentation_15	229/622	388.223	386.701	397.246	407.731	390.641
Segmentation_16	228/610	99.086	99.690	111.121	98.209	108.360
Segmentation_17	225/612	424.686	426.130	425.192	425.240	427.810
Segmentation_18	235/647	89.905	101.307	94.224	88.854	88.809
Segmentation_19	228/624	76.244	78.337	74.284	69.116	70.770
Segmentation_20	232/635	298.802	301.802	302.673	304.457	312.970

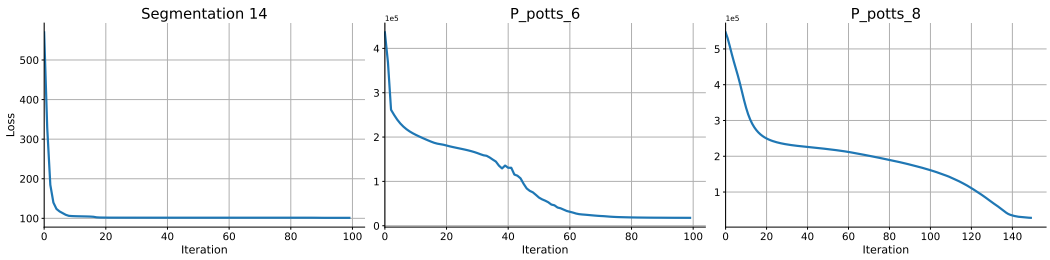


Figure 3: The loss curves of the Segmentation_14, P_potts_6 and P_potts_8 from pairwise potts synthetic problems.

though both graphs are of the same size, the denser graph raises significantly more challenges during optimization. This indicates that increased size and density can complicate the optimization process, and REMAP would need more time to navigate a high quality solution under such cases.

I GNN FORMULATIONS

We summarized the popular GNN message passing formats in Table 11 to show the logic behind the GNN backbone selection of our work.

Table 11: Graph convolutions in typical GNNs

	Graph Convolutions	Neighbor Influence
GCN	$h_i^{(k)} = \sigma\left(W_k \cdot \sum_{j\in\mathcal{N}(i)\cup\{i\}} (\deg(i)\deg(j))^{-1/2} h_j^{(k-1)}\right)$	**Unequal**
GAT	$h_i^{(k)} = \sigma\left(\sum_{j\in\mathcal{N}(i)\cup\{i\}} \alpha_{i,j} W_k h_j^{(k-1)}\right)$	**Unequal**
GraphSAGE	$h_i^{(k)} = \sigma\left(A_k h_i^{(k-1)} + B_k \lvert\mathcal{N}(i)\rvert^{-1} \sum_{j\in\mathcal{N}(i)} h_j^{(k-1)} + b_k\right)$	**Equal**

Loss Landscape Visualization. We visualize loss landscapes using the tool from (Li et al., 2018), with detailed settings in Appendix K. Fig. 4 in Appendix J shows slices for networks of depths $K \in \{1, 2, 5, 8\}$, while Fig. 7 displays converged loss trends. These plots are in the GNN-parameter coordinates, not in logit space. Theorem 4.5 already implies that residual lifting does not enlarge the feasible logit set or create new discrete solutions; depth only changes how the graph branch parameterizes $\rho_v(G)$ and therefore the induced metric $I + JJ^\top$. The two-layer four-hop locality result does not apply to the deeper experimental stacks, which are retained as the original training architecture.

J Results about the analysis experiments

This appendix collects the controls pointed to from Section 5.4: GNN-parameter landscapes (Fig. 4 and Fig. 7), backbone curves (Fig. 5), optimizer curves (Fig. 6), anytime PCI traces (Table 12), and grid-graph wall-clock scaling (Fig. 8).

Fig. 4 slices the GNN-parameter landscape of Segmentation_19 at depths $K \in \{1, 2, 5, 8\}$, using the perturbation setup of Appendix K. These plots live in GNN-parameter coordinates, not logit space. Theorem 4.5 already implies that residual lifting does not enlarge the feasible logit set; depth only changes how $\rho_v(G)$ is parameterized and therefore the induced metric $I + JJ^\top$. Fig. 7 shows the corresponding converged training loss. The two-layer four-hop locality result does not apply to the deeper experimental stacks.

Fig. 5 compares GraphSAGE, GCN, and GAT on UAI 2022 pairwise cases, PCI instances, and synthetic graphs (1000 nodes, average degree 4 or 8, random and Potts energies). GraphSAGE is the default backbone because neighbor aggregation is unweighted, matching MRF neighborhood structure (Table 11 in Appendix I); the curves are consistent with that choice. Fig. 6 compares SGD, RMSprop, and Adam on UAI pairwise instances with learning rate 10^{-4}, width 1024, and eight layers. Adam is the practical optimizer used in the main tables.

Table 12 follows the UAI 1200s protocol on the five real PCI instances, logging every 200s. Toulbar2 solves PCI_1–PCI_3 within 200s. On PCI_4 and PCI_5 it does not terminate within 1200s and improves little after the first interval, while REMAP continues to lower the incumbent. Timing for Toulbar2 on the hardest instances is coarse because the solver's log does not provide a dense trace.

Fig. 8 measures wall-clock time to a relative loss change below 10^{-4} on grid graphs from 5^2 to 50^2 nodes, five repeats per size. The trend is approximately linear in the number of nodes and is consistent with the contraction-cost discussion in Appendix E.

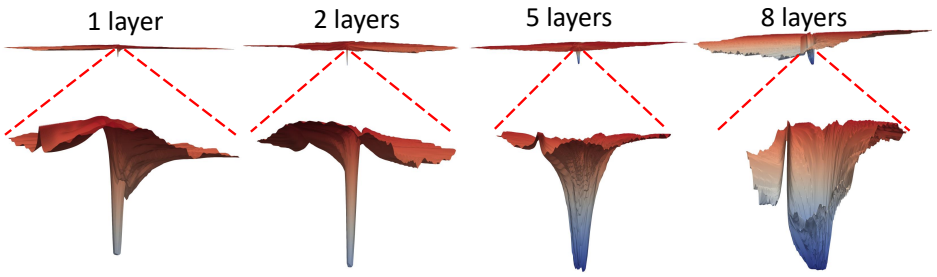


Figure 4: The landscape of instance Segmentation_19. From left to right, the columns correspond to networks with $K \in \{1, 2, 5, 8\}$ layers. The first row is the landscape range from $[-10, +10]$ for both δ and η direction. The second row is the landscape range from $[-1, +1]$ for both δ and η direction.

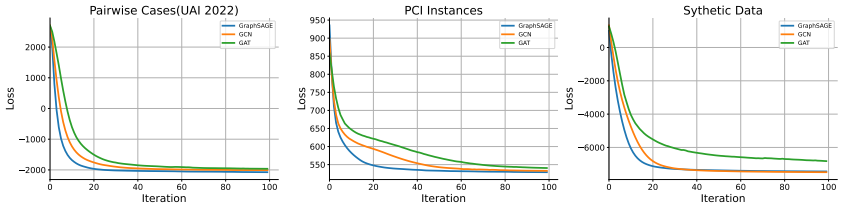


Figure 5: The average loss curves over UAI inference competition 2022 pairwise cases, PCI instances and synthetic instances using GraphSAGE, GCN and GAT as the GNN backbones.

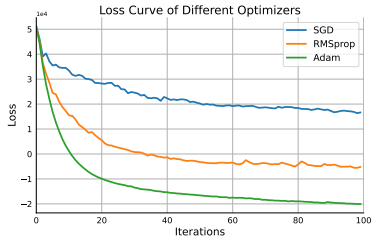


Figure 6: The average loss curves over UAI inference competition 2022 pairwise cases using different optimizers.

K VISUALIZATION SETUP

The core idea of the visualization technique proposed by (Li et al., 2018) involves applying perturbations to the trained network parameters θ^* along two directional vectors, δ and η: $f(\alpha, \beta) = L(\theta^* + \alpha\delta + \beta\eta)$. By doing so, we can generate a 3-D representation of the landscape corresponding to the perturbed parameter space.

We sampled 250000 points in the $\alpha - \beta$ plane, where both α and β range from -10 to 10, to obtain an overview of the loss function landscape. Subsequently, we focused on the region around the parameter θ^* by sampling an additional 10,000 points in a narrower range, with α and β both from -1 to 1.

L READ UAI FORMAT FILES

An example data file in UAI format is provided in Box. L. This Markov Random Field consists of 3 variables, each with 2 possible states. Detailed information can be found in the box, where we illustrate the meanings of different sections of the file. Notably, the potential tables are not normalized.

Table 12: Time comparison between Toulbar2 and REMAP on PCI instances. "-" if the solving process is already stopped.

Instances	Algorithm/Solver	200s	400s	600s	800s	1000s	1200s
PCI_1	Toulbar2	18.134	-	-	-	-	-
	REMAP	18.211 (10s in total)	-	-	-	-	-
PCI_2	Toulbar2	98.364	-	-	-	-	-
	REMAP	98.446 (16s in total)	-	-	-	-	-
PCI_3	Toulbar2	1003.640	-	-	-	-	-
	REMAP	1003.640 (71s in total)	-	-	-	-	-
PCI_4	Toulbar2	428.299	426.806	426.806	426.806	426.806	426.806
	REMAP	408.508	407.6304	407.419	-	-	-
PCI_5	Toulbar2	1128.244	1121.325	1121.325	1121.325	1121.325	1121.325
	REMAP	1222.281	1086.899	1077.858	1074.3094	1070.8013	1069.875

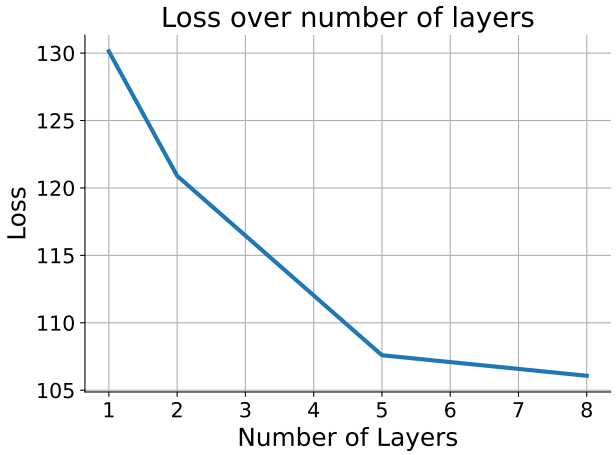


Figure 7: The training loss of instance Segmentation_19 after convergence of using network layer number $\{1, 2, 5, 8\}$.

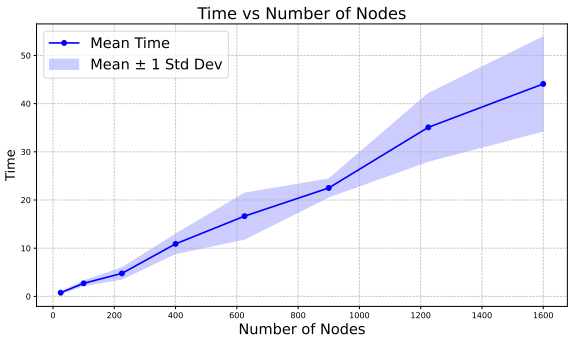


Figure 8: Wall-clock convergence time of REMAP on grid graphs from 5^2 to 50^2 nodes. Each size is repeated five times. Convergence is a relative loss change below 10^{-4}.

During BP, these tables may be normalized to avoid numerical issues. For the energy transformation, however, we do not normalize them; each positive potential value ϕ is converted directly to $-\log \phi$.

Example.uai

```
MARKOV      //Instance type
3     //Number of variables
2 2 2     //State number of each variable
5     //Number of cliques that has potentials
1 0     //1 means this clique is a variable, and the variable is 0.
1 1
1 2
2 0 1     //2 means this clique is an edge, the edge is (0, 1).
3 0 1 2     //3 means this clique includes 3 variables, and the clique is (0, 1, 2).

2     //The number 2 indicates that the potential in the next line has two values.
0.1 0.9     //The potential of variable 0 is 0.1 for state 0 and 0.9 for state 1.

2
0.1 10

2
0.5 0.5

4
0.1 1.0 1.0 0.1//The potential of the state combinations for variables 0 and 1 is given in the order of
(0,0), (0,1), (1,0) and (1,1).

8
0.1 2.0 0.1 0.1 0.1 0.1 0.1 2.0     //The potential of the state combinations for variables 0, 1, and 2 is
given in the order of (0,0,0), (0,0,1), (0,1,0), (0,1,1), (1,0,0), and so on.
```

Since the transformation of variable energies and clique energies follows the same procedure, we will use the edge $(0, 1)$ to illustrate the transformation. Table 13 shows the unnormalized potentials for the edge $(0, 1)$; Table 14 is obtained by applying $-\log(\cdot)$ entrywise to Table 13.

Table 13: $P(x_0, x_1)$

x_0 \ x_1	0	1
0	0.1	1.0
1	1.0	0.1

Table 14: $\theta_C(x_0, x_1)$

x_0 \ x_1	0	1
0	2.303	0
1	0	2.303

M PCI PROBLEM FORMULATION

This appendix records the MIP-to-MRF conversion used for the PCI experiments in Section 5. Table 3 in the main text reports five real instances from a Chinese city. Table 17 below reports five larger synthetic PCI-style graphs generated from the same pairwise interference model, using the protocol of Section 5.1 with 100 iterations and a 3600s Toulbar2 limit. On the synthetic family, LBP and TRBP remain far above the other two methods; Toulbar2 improves over message passing but is consistently above residual REMAP as the graph grows.

The Mixed Integer Programming format of PCI problems is as follows:

where n is the index for devices, and N is the set of these indices. P stands for the possible states of each device. M_{ih} stands for the possible states set for node n_i. L_{ij} is the cost when given a certain choices of the states of device i and device j, a_{ij} is the coefficient of the cost in the objective function. There is an $(i, j) \in \mathcal{E}$ means there exists interference between these two devices.

When using MRF to model PCI problems, each random variable represents the identity state of the given node and the interference between devices would be captured by the pairwise energy functions. Next we will introduce how to transform the PCI problem from MIP form to MRF form.

$$\min_{z,L} \quad \sum_{(i,j)\in\mathcal{E}} a_{ij} L_{ij} \tag{19}$$

$$\text{s.t.} \quad z_{np} \in \{0,1\}, \quad \forall n \in N, p \in P \tag{19a}$$

$$\sum_{p\in P} z_{np} = 1, \quad \forall n \in N. \tag{19b}$$

$$\sum_{p\in M_{ih}} z_{n_i p} + \sum_{p\in M_{jh}} z_{n_j p} - 1 \leq L_{ij}, \forall (i,j) \in \mathcal{E}, \forall h \in \{0,1,2\}. \tag{19c}$$

In the original MIP formulation of the PCI problems, three types of constraints are defined. By combining Eq. 19a and Eq. 19b, we establish that each device must select exactly one state at any given time. Furthermore, the constraint in Eq. 19c indicates that interference occurs between two devices only when they select specific states. The overall impact on the system is governed by the value of L_{ij} and its corresponding coefficient. Given that interference is always present, the objective is to minimize its extent.

To transform these problems into an MRF framework, we utilize Eq. 19b to represent the nodes, where each instance of Eq. 19a corresponds to the discrete states of a specific node. The constraints set forth in Eq. 19a and Eq. 19b ensure that only one state can be selected at any given time, thus satisfying those conditions automatically. By processing Eq. 19c, we can identify the edges and their associated energies. If $z_{n_i p}$ and $z_{n_j p}$ appear in the same constraint from Eq. 19c, we can formulate an edge (i, j). By selecting different values for $z_{n_i p}$ and $z_{n_j p}$, we can determine the minimum value of L_{ij} that maintains the validity of the constraint.

The product of L_{ij} and a_{ij} represents the energy associated with the edge (i, j) under the combination of the respective states. Once the states of all nodes are fixed, the values of the edge costs also become fixed. This leads to the conclusion that the objective function is the summation of the energies across all edges. Since the PCI problems do not include unary terms, we can omit them during the transformation process. This establishes a clear pathway for converting the MIP formulation into an MRF representation, allowing us to leverage MRF methods for solving the PCI problems effectively.

Example
The original problem is

$$\begin{aligned}
\min_{z,L} \quad & L_{1,2} + 3L_{2,3} \\
\text{s.t.} \quad & z_{np} \in \{0,1\}, && \forall n \in \{1,2,3\}, p \in \{1,2,3\} \\
& \sum_{p\in P} z_{np} = 1, && \forall n \in \{1,2,3\}. \\
& z_{11} + z_{21} - 1 \leq L_{1,2} \\
& z_{13} + z_{22} - 1 \leq L_{1,2} \\
& z_{12} + z_{23} - 1 \leq L_{1,2} \\
& z_{21} + z_{31} - 1 \leq L_{2,3} \\
& z_{22} + z_{32} - 1 \leq L_{2,3} \\
& z_{23} + z_{33} - 1 \leq L_{2,3}
\end{aligned} \tag{20}$$

Then the corresponding MRF problem is

$$\min \theta_{1,2}(x_1, x_2) + \theta_{2,3}(x_2, x_3) \tag{21}$$

the energy on edge (x_1, x_2) and edge (x_2, x_3) are shown in Table 15 and Table 16.

N LIMITATIONS

Preconditioning is not global optimality. The residual bypass makes the lifting lossless and guarantees at least the instantaneous free-logit dissipation. It does not make a nonconvex trajectory reach a global

Table 15: $E(x_1, x_2)$

x_1 \ x_2	z_{21}	z_{22}	z_{23}
z_{11}	1	0	0
z_{12}	0	0	1
z_{13}	0	1	0

Table 16: $E(x_2, x_3)$

x_2 \ x_3	z_{31}	z_{32}	z_{33}
z_{21}	3	0	0
z_{22}	0	3	0
z_{23}	0	0	3

Table 17: Synthetic PCI instances, same pairwise MRF conversion as Table 3. Energy values; lower is better. Best in bold.

Graph	#Nodes/#Edges	LBP	TRBP	Toulbar2	REMAP	
					Energy	Loss
PCI_synthetic_1	280/9678	564198.000	568082.000	522857.923	$\mathbf{496015.5 \pm 6307.363}$	496013.662 ± 6297.169
PCI_synthetic_2	526/34500	2.092e+06	2.084e+06	2.064e+06	$\mathbf{1.923e+06 \pm 9977.015}$	$1.923e+06 \pm 10007.739$
PCI_synthetic_3	1000/49950	2.932e+06	2.908e+06	2.856e+06	$\mathbf{2.665e+06 \pm 4555.868}$	$2.664e+06 \pm 4468.965$
PCI_synthetic_4	1500/78770	4.568e+06	4.532e+06	4.534e+06	$\mathbf{4.215e+06 \pm 13500.602}$	$4.214e+06 \pm 13252.456$
PCI_synthetic_5	2000/120024	6.807e+06	6.904e+06	7.023e+06	$\mathbf{6.542e+06 \pm 19789.758}$	$6.540e+06 \pm 19782.638$

solution from every initialization. The local rate requires entry into a MAP-margin neighborhood, and the factor $(1+\beta)^{-1}$ requires residual alignment $\|J_v^\top q\|_2^2 \geq \beta \|q\|_2^2$; β is trajectory-dependent and is not implied by the architecture.

High order can be exponentially large. For explicit dense tables, time and memory grow with $\prod_{i \in C} s_i$. The method scales to many nodes only when the total factor representation and contraction fit memory. Structured kernels or low-rank factors are necessary for genuinely large order.

The PCI parameterization ablation compares terminal discrete energies of residual, GNN-only, and free logits. It does not estimate $\beta_t = \|J_v^\top q\|_2^2 / \|q\|_2^2$ along the residual trajectory, and it is not a matched-compute anytime test.

Heuristic optimization. Adam, restarts, and optional temperature schedules have no global convergence guarantee here. Conditional rounding guarantees a discrete energy no worse than the current relaxed value, but neither quantity is guaranteed close to $E_\star$ without an external lower bound.

Best use as a portfolio component. The most credible operational role is to generate GPU incumbents for difficult instances and pass them to an exact or dual solver, yielding the certified gap $E(x) - B$.