Differentiable Spectral Normalization for Large-Scale Ising Optimization

Thinh Nguyen-Cong[1] , **Thang Ngoc Dinh**[1,*]
[1]Virginia Commonwealth University, Virginia, United States
{nguyencongt, tndinh}@vcu.edu

Abstract

Spectral relaxation is widely used for large-scale combinatorial optimization due to its computational efficiency. Yet its effectiveness depends critically on the choice of graph normalization, a design decision typically made heuristically. Here, we show that normalization can be treated as a continuous optimization variable rather than a fixed preprocessing choice. Our method, Differentiable Spectral Normalization (DSN), parameterizes the spectral relaxation through a diagonal metric and maximizes the resulting lower bound via projected gradient ascent. Exact gradients are obtained through the Hellmann-Feynman theorem using only the principal eigenpair, maintaining linear complexity per iteration. On benchmark instances ranging from 10^3 to 8.4×10^6 nodes, DSN improves solution quality by 3–15% over static spectral methods. Its performance comes within 1–3% of state-of-the-art metaheuristics, such as simulated annealing, at up to $190\times$ lower computational cost on large-scale instances. These results suggest that learning problem-specific relaxation geometry can substantially close the gap between spectral scalability and metaheuristic solution quality.

1 Introduction

Ising optimization arises in diverse applications including correlation clustering [Bansal *et al.*, 2004], MAP inference in graphical models [Wainwright and Jordan, 2008], and energy-based learning [Hopfield, 1982; Ackley *et al.*, 1985]. Modern instances involve sparse, signed graphs with 10^5–10^7 variables [Lucas, 2014; Kunegis *et al.*, 2010], where computational efficiency and solution quality are essential.

Existing approaches exhibit a tradeoff between these objectives. Spectral relaxation reduces the discrete problem to a sparse eigenvalue computation, achieving near-linear time complexity [Delorme and Poljak, 1993; Trevisan, 2012]. However, the quality of spectral solutions depends on the choice of graph normalization, typically the adjacency matrix or degree-normalized Laplacian, which is selected a priori without reference to the problem instance. Empirical studies indicate that no normalization performs well across diverse graph families [Cucuringu *et al.*, 2019; Cucuringu *et al.*, 2021]. Combinatorial metaheuristics such as simulated annealing [Kirkpatrick *et al.*, 1983] and parallel tempering [Swendsen and Wang, 1986] achieve better solution quality by directly exploring the discrete energy landscape, but their reliance on sequential sampling limits scalability: runtimes grow prohibitively for graphs beyond 10^5 nodes.

This paper investigates whether the normalization that defines a spectral relaxation can be optimized. We introduce Differentiable Spectral Normalization (DSN), which treats normalization as a continuous optimization variable. The method parameterizes the relaxation through a positive diagonal matrix W that defines a weighted inner product, and maximizes the spectral lower bound via projected gradient ascent. The key computational insight is that gradients of eigenvalues with respect to W admit a closed-form expression via the Hellmann-Feynman theorem [Zhang and George, 2002; Milgrom and Segal, 2002], requiring only the principal eigenpair and preserving $O(|E|)$ complexity per iteration.

The main contributions are as follows:

- We formulate normalization selection as bound maximization over a family of diagonal metrics, providing a principled alternative to heuristic operator design.
- We derive an efficient gradient expression that avoids backpropagation through the eigensolver, enabling optimization on graphs with millions of nodes.
- We demonstrate empirically that the learned normalization yields tighter bounds that correlate strongly with improved discrete solutions (Figure 1).

On standard benchmarks (Gset, signed social networks) and synthetic graphs up to 8.4×10^6 nodes, DSN improves solution quality by 3–15% over static spectral baselines and achieves energies within 1–3% of simulated annealing at up to $190\times$ lower runtime. These results indicate that optimizing relaxation geometry can narrow the gap between spectral efficiency and metaheuristic quality, without sacrificing the scalability advantages of eigenvalue-based methods.

The remainder of this paper is organized as follows. Section 2 defines the Ising problem and spectral relaxation framework. Section 3 derives the DSN update rule. Section 4

*Corresponding author

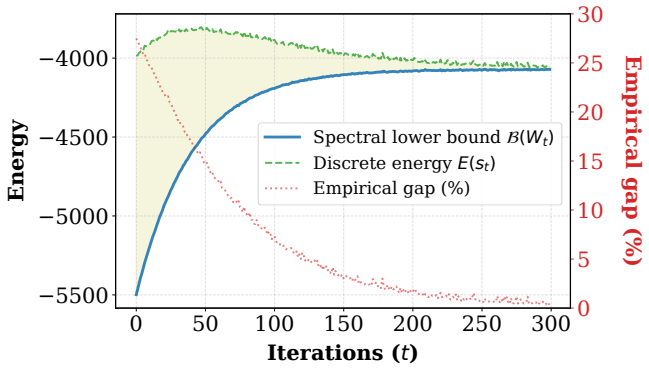


Figure 1: Joint convergence of the spectral bound and discrete energy. DSN trajectory over $T = 300$ iterations: the spectral lower bound $\mathcal{B}(W^{(t)})$ (solid, left axis) and the discrete energy $\mathcal{E}(s^{(t)})$ (dashed, left axis) tighten in tandem, closing the empirical gap (dotted, right axis) from $\sim$30% to near zero.

analyzes smoothness and convergence properties. Section 5 presents experimental results.

Related works. Approaches for large-scale Ising optimization can be broadly categorized into three paradigms: spectral relaxations, semidefinite programming, and combinatorial metaheuristics.

Spectral relaxation serves as a cornerstone for scalable graph clustering and Max-Cut due to its efficiency in leveraging sparse linear algebra [Delorme and Poljak, 1993; Trevisan, 2012]. These methods relax discrete constraints to a spherical manifold and solve for the principal eigenvectors of a graph operator. To handle signed interactions, foundational works have proposed specific operator constructions, such as the signed Laplacian and generalized eigenproblems [Kunegis *et al.*, 2010; Cucuringu *et al.*, 2019]. More recently, regularization techniques have been introduced to stabilize embeddings for sparse, heterogeneous graphs [Cucuringu *et al.*, 2021]. However, a critical limitation remains: these solvers rely on *fixed, axiomatic normalizations* (e.g., degree-based scaling) that are selected heuristically rather than optimized. Such static choices often fail to capture the complex geometry of ground-truth solutions in massive networks. DSN generalizes this paradigm by treating the normalization matrix not as a hyperparameter, but as a continuous variable learned directly from the Ising objective.

Pioneered by Goemans and Williamson [Goemans and Williamson, 1995], Semidefinite programming (SDP) relaxations provide rigorous approximation guarantees by optimizing the embedding geometry over the cone of positive semidefinite matrices. Unlike spectral methods that project data onto a low-dimensional subspace, SDP allows for a high-dimensional representation that tightens the relaxation gap. However, standard interior-point methods for SDP scale poorly (typically $O(N^{3.5})$ or worse), rendering them intractable for graphs with $N > 10^4$. Our proposed DSN strikes a balance: it can be viewed as a scalable "diagonal SDP" that restricts the metric structure to a diagonal form. This restriction allows DSN to retain the near-linear $O(|E|)$ complexity of spectral methods while capturing the essential geometric flexibility required for high-quality solutions.

Metaheuristics such as Simulated Annealing (SA) [Kirkpatrick *et al.*, 1983], Parallel Tempering [Swendsen and Wang, 1986], and Tabu Search directly explore the discrete energy landscape. These methods are capable of escaping local minima and finding near-optimal states on moderate-sized instances. However, their performance relies heavily on extensive sequential sampling, long burn-in phases, and delicate cooling schedules [Lucas, 2014]. For massive graphs ($N > 10^6$), the convergence time becomes prohibitive. DSN complements these approaches by providing a fast, deterministic method to generate high-quality embeddings, which can serve as superior warm-start configurations for fine-tuning heuristics.

Optimizing objectives that depend on spectral quantities requires computing gradients of eigenvalues and eigenvectors. While sensitivity analysis for matrix decompositions is well-established [Magnus, 1985; Kato, 1995], applying it to large-scale optimization is challenging. Recent machine-learning approaches, spectral normalization [Miyato *et al.*, 2018] and differentiable eigendecomposition [Ionescu *et al.*, 2015; Wang *et al.*, 2019], instead rely on automatic differentiation through iterative eigensolvers, which is memory-intensive and prone to numerical instability on large graphs. Our work circumvents these issues by leveraging the Hellmann-Feynman theorem [Zhang and George, 2002; Milgrom and Segal, 2002]. This allows us to derive exact analytical gradients using only the principal eigenpair, enabling efficient geometry updates without the need for expensive backpropagation. Complementary work applies learning to combinatorial optimization, GNN solvers [Schuetz *et al.*, 2022], RL [Dai *et al.*, 2017], unsupervised objectives [Karalias and Loukas, 2020], but typically requires training data; DSN is orthogonal, optimising relaxation geometry per-instance without any.

2 Preliminaries

2.1 Ising Optimization Problem

We consider the problem of finding a binary spin configuration $s \in \{-1, +1\}^n$ that minimizes the Ising energy (Hamiltonian):

$$\mathcal{E}^\star \triangleq \min_{s\in\{-1,+1\}^n} \left(\mathcal{E}(s) \triangleq -s^\top J s = -\sum_{i,j} J_{ij} s_i s_j \right). \tag{1}$$

In this convention, a positive interaction $J_{ij} > 0$ (ferromagnetic) encourages alignment ($s_i = s_j$) to lower the energy contribution to $-J_{ij}$, while $J_{ij} < 0$ (antiferromagnetic) encourages disagreement. Finding the exact global minimizer $\mathcal{E}^\star$ is generally NP-hard.

2.2 Signed Laplacian and Spectral Duality

To extend spectral graph theory to signed interactions, we utilize the *Signed Laplacian* $\bar{L}$. Let D be the diagonal signed degree matrix with entries $D_{ii} \triangleq \sum_j |J_{ij}|$. The Signed Laplacian is defined as $\bar{L} \triangleq D - J$. Leveraging the boolean constraint $s_i^2 = 1$, we establish the duality between the Laplacian

quadratic form and the Ising energy:

$$s^\top \bar{L} s = \sum_i D_{ii} s_i^2 - s^\top J s = C + \mathcal{E}(s), \tag{2}$$

where $C = \sum_{i,j} |J_{ij}|$ is a constant total weight. Consequently, minimizing the Ising energy $\mathcal{E}(s)$ is mathematically equivalent to minimizing the quadratic form of the Signed Laplacian.

2.3 Spectral Relaxations and Lower Bounds

The combinatorial hardness of Eq. (1) stems from the discrete hypercube constraint $s \in \{-1, +1\}^n$. Spectral methods address this by relaxing the domain to a continuous manifold.

A standard approach relaxes the discrete condition to a spherical constraint $\mathcal{S}^{n-1} = \{u \in \mathbb{R}^n : u^\top u = n\}$. Substituting the continuous variable u into the objective yields the eigenvalue problem: $\min_{u \in \mathbb{R}^n} -u^\top J u \quad \text{s.t.} \quad u^\top u = n$.

Using the algebraic identity $\min(-x) = -\max(x)$, the optimal solution is given by the eigenvector corresponding to the *largest* eigenvalue $\lambda_{\max}(J)$. This provides a loose spectral lower bound: $\mathcal{E}^\star \geq -n \cdot \lambda_{\max}(J)$. More generally, the relaxation geometry can be parameterized by a positive diagonal matrix $W = \mathrm{diag}(w) \succ 0$. Since any discrete spin s satisfies $s^\top W s = \sum w_i s_i^2 = \mathrm{Tr}(W)$, we can relax the domain to a weighted hypersphere:

$$\mathcal{M}(W) = \{u \in \mathbb{R}^n \mid u^\top W u = \mathrm{Tr}(W)\}. \tag{3}$$

This generalization leads to a fundamental lower bound that depends on the choice of the metric W.

Lemma 1. *For any positive diagonal matrix W, the optimal Ising energy is lower-bounded by the principal eigenvalue of the normalized coupling matrix:*

$$\mathcal{E}^\star \geq \mathcal{B}(W) \triangleq -\mathrm{Tr}(W) \cdot \lambda_{\max}\left(W^{-1/2} J W^{-1/2}\right). \tag{4}$$

Proof. Let $N(W) = W^{-1/2} J W^{-1/2}$. By setting $v = W^{1/2} u$, the problem becomes $\min_{v:\|v\|^2 = \mathrm{Tr}(W)} -v^\top N(W) v$, which simplifies to $-\mathrm{Tr}(W)\lambda_{\max}(N(W))$ via the Rayleigh-Ritz theorem. □

Existing spectral methods typically commit to a fixed, heuristic geometry ($W = I$ or $W = D$), often resulting in loose bounds. In contrast, our methodology (Section 3) treats $\mathcal{B}(W)$ as an objective function to be maximized.

3 Methodology

3.1 Learning The Relaxation Geometry via A Diagonal Metric

Classical spectral solvers fix the relaxation geometry by committing to a static operator (the unweighted matrix J or the normalized Laplacian). We instead relax the discrete hypercube onto a learnable weighted hypersphere parameterized by a positive diagonal matrix $W = \mathrm{diag}(w_1, \ldots, w_n) \succ 0$. Consider the continuous relaxation of the Ising energy minimization problem defined in Eq. (1):

$$\min_{v \in \mathbb{R}^n} -v^\top J v \quad \text{subject to} \quad v^\top W v = \mathrm{Tr}(W). \tag{5}$$

Geometrically, W defines a diagonal metric that deforms the constraint surface, effectively stretching or compressing the embedding space to align the spectral solution with the discrete problem structure. Solving this constrained optimization via Lagrange multipliers involves finding the stationary points of the Rayleigh quotient. Using the identity $\min(-x) = -\max(x)$, the minimum relaxed energy is determined by the principal eigenpair $(\lambda_{\max}, u)$ of the normalized operator: $N(W) \triangleq W^{-1/2} J W^{-1/2}, \quad \text{with } \|u\|_2 = 1.$

This relaxation induces a rigorous spectral lower bound on the optimal ground-state energy: $\mathcal{E}^\star \geq \mathcal{B}(W) \triangleq -\lambda_{\max}(N(W)) \cdot \mathrm{Tr}(W)$.

Our goal is to learn the optimal normalization by maximizing this lower bound (tightening the relaxation) over a compact domain:

$$\max_{W \in \mathcal{D}_\varepsilon} \mathcal{B}(W), \quad \text{where } \mathcal{D}_\varepsilon = \left\{ W \in \mathbb{R}^{n \times n}_{\mathrm{diag}} : \varepsilon \leq w_i \leq \varepsilon^{-1} \right\}. \tag{6}$$

The domain $\mathcal{D}_\varepsilon$ prevents degenerate scalings and ensures the numerical stability of the eigensolver.

3.2 Gradient Derivation via Eigenvalue Perturbation

A central technical challenge is that the objective $\mathcal{B}(W)$ depends on $\lambda_{\max}(N(W))$. Differentiating through iterative eigensolvers (Lanczos) via unrolling is memory-intensive and often numerically unstable. Instead, we leverage first-order eigenvalue perturbation theory (Hellmann–Feynman theorem). Let (λ, u) be the dominant eigenpair of $N(W)$ with $\|u\|_2 = 1$. The gradient of the eigenvalue with respect to a diagonal parameter w_i is: $\frac{\partial \lambda}{\partial w_i} = u^\top \frac{\partial N(W)}{\partial w_i} u$. Using the identity $\frac{\partial W^{-1/2}}{\partial w_i} = -\frac{1}{2} w_i^{-3/2} E_{ii}$, where $E_{ii} \in \mathbb{R}^{n \times n}$ is the elementary matrix with a 1 at entry (i, i) and 0 elsewhere, the derivative of the operator is:

$$\frac{\partial N(W)}{\partial w_i} = -\frac{1}{2 w_i} \left(E_{ii} N(W) + N(W) E_{ii} \right). \tag{7}$$

Substituting this into the bilinear form and utilizing $N(W)u = \lambda u$, we obtain:

$$\frac{\partial \lambda}{\partial w_i} = -\frac{\lambda}{w_i} u_i^2. \tag{8}$$

Applying the product rule to the full objective $\mathcal{B}(W) = -\lambda \mathrm{Tr}(W)$ (where $\mathrm{Tr}(W) = \sum w_k$), we derive the closed-form gradient: $\frac{\partial \mathcal{B}(W)}{\partial w_i} = -\left(\frac{\partial \mathrm{Tr}(W)}{\partial w_i} \lambda + \mathrm{Tr}(W) \frac{\partial \lambda}{\partial w_i} \right) = -\left(1 \cdot \lambda + \mathrm{Tr}(W) \left(-\frac{\lambda u_i^2}{w_i} \right) \right) = \lambda \left(\frac{\mathrm{Tr}(W)}{w_i} u_i^2 - 1 \right)$.

This gradient expression relies solely on the leading eigenpair (λ, u), enabling efficient computation without backpropagation. Intuitively, $\nabla_{w_i} \mathcal{B}$ acts as a *localization penalty on node i*: if a node i exhibits localization (large u_i^2), the gradient is positive (assuming $\lambda > 0$), prompting an increase in w_i. This penalizes the node in the weighted norm constraint, effectively "flattening" the eigenvector to enforce a more distributed embedding.

Algorithm 1 Differentiable Spectral Normalization (DSN)

Input: Coupling matrix J; step size η; iterations T.
Output: Discrete spin $s \in \{\pm 1\}^n$.

Initialize: $s_{\text{best}} \leftarrow \emptyset$, $E_{\text{best}} \leftarrow \infty$.
Initialize v_{init} randomly, $W^{(0)} \leftarrow D$.
for $t = 0$ to T **do**
  1. Spectral solving: Compute (λ_t, u_t) of:

$$N(W^{(t)}) = (W^{(t)})^{-1/2} J (W^{(t)})^{-1/2}$$

  2. Discrete sampling: $s^{(t)} \leftarrow \text{sign}((W^{(t)})^{-1/2} u_t)$

$$E(s^{(t)}) \leftarrow -(s^{(t)})^\top J s^{(t)}$$

  3. Update best solution:
  if $E(s^{(t)}) < E_{\text{best}}$ **then**
    $s_{\text{best}} \leftarrow s^{(t)}$, $E_{\text{best}} \leftarrow E(s^{(t)})$
  end if
  4. Gradient evaluation: Compute $\nabla \mathcal{B}$:

$$\nabla_{w_i} \mathcal{B} \leftarrow \lambda_t \left(\frac{\text{Tr}(W^{(t)})}{w_i} (u_t)_i^2 - 1 \right) \tag{10}$$

  5. Geometry update:

$$W^{(t+1)} \leftarrow \Pi_{\mathcal{D}_\varepsilon} \left(W^{(t)} + \eta \nabla \mathcal{B}(W^{(t)}) \right)$$

  Project $W^{(t+1)}$ to $\mathcal{D}_\varepsilon$ to ensure positive definiteness.
  6. Warm-start: $v_{\text{init}} \leftarrow u_t$
end for
return s_{best}

3.3 Algorithm

We optimize $\mathcal{B}(W)$ via projected gradient ascent on the manifold $\mathcal{D}_\varepsilon$. The update rule at iteration t is:

$$W^{(t+1)} \leftarrow \Pi_{\mathcal{D}_\varepsilon} \left(W^{(t)} + \eta \nabla \mathcal{B}(W^{(t)}) \right), \tag{9}$$

where $\eta > 0$ is the step size and $\Pi_{\mathcal{D}_\varepsilon}$ denotes element-wise projection onto $[\varepsilon, \varepsilon^{-1}]$. The complete procedure is summarized in Algorithm 1. The computational cost of each iteration is dominated by the sparse matrix-vector products required to compute the leading eigenpair of $N(W^{(t)})$. By using efficient iterative solvers (LOBPCG or Lanczos), the complexity remains near-linear in the number of edges $|E|$, preserving the scalability of classical spectral methods.

Although DSN optimizes a continuous spectral bound, the final objective is a discrete spin configuration. We employ standard sign rounding to discretize the optimized embedding. Regarding convergence, DSN performs projected gradient ascent over a compact domain. Under mild assumptions (Lipschitz continuity of the eigenvalue function, which holds given a non-zero spectral gap), the procedure is guaranteed to converge to a stationary point of the spectral bound $\mathcal{B}(W)$.

4 Theoretical Foundations

Definition 2. *Let $\mathcal{D}_{++}^n = \{W = diag(w) \in \mathbb{R}^{n \times n} \mid w_i > 0, \forall i\}$ be the cone of positive definite diagonal matrices. For any $W \in \mathcal{D}_{++}^n$, the weighted spectral operator $N(W) \in \mathbb{R}^{n \times n}$ is defined as the similarity transformation of the coupling matrix J induced by the metric W:*

$$N(W) \triangleq W^{-1/2} J W^{-1/2}.$$

The eigenvalues of $N(W)$, denoted by $\lambda_k(N(W))$, correspond to the generalized eigenvalues of the pencil (J, W), satisfying the characteristic equation $\det(J - \lambda W) = 0$.

This operator generalizes standard spectral graph matrices. The choice $W = I$ yields the coupling matrix J itself, while $W = D$ (where $D_{ii} = \sum_j |J_{ij}|$) yields the normalized signed operator associated with the random walk Laplacian. The mapping $W \mapsto N(W)$ is smooth on $\mathcal{D}_{++}^n$, ensuring that the spectral properties evolve continuously with respect to the weights w.

Theorem 3. *Let $\mathcal{D} \subseteq \mathcal{D}_{++}^n$ be the domain where the principal eigenvalue $\lambda_{\max}(N(W))$ is simple. For any $W \in \mathcal{D}$, the spectral lower bound function $\mathcal{B}(W)$ is differentiable. Let (λ, u) denote the principal eigenpair of $N(W)$ with $\|u\|_2 = 1$. The partial derivative of $\mathcal{B}(W)$ with respect to the weight w_i is given by:*

$$\frac{\partial \mathcal{B}}{\partial w_i} = \lambda \left(\frac{Tr(W)}{w_i} u_i^2 - 1 \right).$$

Proof. The objective function is defined as $\mathcal{B}(W) = -\lambda(W)\text{Tr}(W)$, where we denote $\lambda(W) = \lambda_{\max}(N(W))$ for brevity. By the product rule of differentiation:

$$\frac{\partial \mathcal{B}}{\partial w_i} = -\left(\frac{\partial \lambda(W)}{\partial w_i} \text{Tr}(W) + \lambda(W) \frac{\partial \text{Tr}(W)}{\partial w_i} \right).$$

The derivative of the trace term is trivial: $\frac{\partial \text{Tr}(W)}{\partial w_i} = \frac{\partial}{\partial w_i}(\sum_j w_j) = 1$. To evaluate the eigenvalue derivative, we invoke the Hellmann-Feynman theorem extended to matrix operators. Since λ is simple, it varies smoothly with W, and its first derivative is given by the expectation of the operator derivative with respect to the eigenvector:

$$\frac{\partial \lambda(W)}{\partial w_i} = u^\top \left(\frac{\partial N(W)}{\partial w_i} \right) u.$$

Recall that $N(W) = W^{-1/2} J W^{-1/2}$. Let E_{ii} denote the elementary matrix with 1 at entry (i, i) and 0 elsewhere. Differentiating the diagonal scaling matrix yields $\frac{\partial}{\partial w_i} W^{-1/2} = -\frac{1}{2} w_i^{-3/2} E_{ii} = -\frac{1}{2w_i} E_{ii} W^{-1/2}$. Applying the product rule to the operator $N(W)$:

$$\frac{\partial N(W)}{\partial w_i} = -\frac{1}{2w_i} \left(E_{ii} N(W) + N(W) E_{ii} \right).$$

Substituting this into the eigenvalue derivative expression:

$$\frac{\partial \lambda(W)}{\partial w_i} = u^\top \left[-\frac{1}{2w_i} (E_{ii} N(W) + N(W) E_{ii}) \right] u.$$

Using the eigenvalue equation $N(W)u = \lambda u$ and the symmetry $u^\top N(W) = \lambda u^\top$:

$$\frac{\partial \lambda(W)}{\partial w_i} = -\frac{\lambda}{2w_i} (2 u^\top E_{ii} u). \tag{11}$$

Since $u^\top E_{ii} u = u_i^2$, we obtain the scalar derivative $\frac{\partial \lambda(W)}{\partial w_i} = -\frac{\lambda u_i^2}{w_i}$. Finally, substituting terms back into the objective gradient equation:

$$\frac{\partial \mathcal{B}}{\partial w_i} = -\left(\left[-\frac{\lambda u_i^2}{w_i}\right] \mathrm{Tr}(W) + \lambda \cdot 1\right) \tag{12}$$

$$= \lambda \left(\frac{\mathrm{Tr}(W)}{w_i} u_i^2 - 1\right). \tag{13}$$

□

Standard matrix perturbation theory establishes that the principal eigenvalue $\lambda_{\max}(N(W))$ is a real-analytic function of the matrix entries provided the eigenvalue is simple [Kato, 1995]. Since our feasible set $\mathcal{D}_\varepsilon$ is compact, the existence of a uniform spectral gap $\gamma > 0$ (discussed below) ensures that both the objective function $\mathcal{B}(W)$ and its gradient are Lipschitz continuous, satisfying the regularity conditions required for optimization.

Lemma 4. *Let $\mathcal{D}_\varepsilon := \{W = \mathrm{diag}(w) \in \mathbb{R}^{n\times n} : \varepsilon \le w_i \le \varepsilon^{-1}\}$ be the feasible set. Assume there exists a uniform lower bound $\gamma > 0$ such that trajectory $\{W^{(t)}\}$ remains within the region of non-degenerate spectral gaps:*

$$\mathcal{D}_{\varepsilon,\gamma} := \{W \in \mathcal{D}_\varepsilon | \lambda_1(N(W)) - \lambda_2(N(W)) \ge \gamma\}.$$

Then $\mathcal{B}(W)$ is L-smooth on $\mathcal{D}_{\varepsilon,\gamma}$. For a step size $0 < \eta \le 1/L$, the projected gradient updates satisfy:

1. *The objective strictly increases unless a stationary point is reached. With the standard mapping is defined as $\mathcal{G}_\eta(W^{(t)}) = \frac{1}{\eta}(W^{(t)} - W^{(t+1)})$, we have:*

$$\mathcal{B}(W^{(t+1)}) - \mathcal{B}(W^{(t)}) \ge \frac{\eta}{2} \|\mathcal{G}_\eta(W^{(t)})\|_F^2$$

2. *$\{W^{(t)}\}$ admits accumulation points, and every W^* is a first-order stationary point satisfying: $\langle \nabla\mathcal{B}(W^*), W - W^* \rangle \le 0, \quad \forall W \in \mathcal{D}_\varepsilon$.*

3. *Let $\mathcal{B}^* := \max_{W\in\mathcal{D}_\varepsilon} \mathcal{B}(W)$. The minimum squared norm of gradient mapping decays at a rate of $O(1/T)$:*

$$\min_{0\le k<T} \|\mathcal{G}_\eta(W^{(k)})\|_F^2 \le \frac{2(\mathcal{B}^* - \mathcal{B}(W^{(0)}))}{\eta T}.$$

Proof. (Sketch) $\mathcal{D}_{\varepsilon,\gamma}$ is a compact subset. Within this region, the principal eigenvalue is simple, implying that $\mathcal{B}(W)$ is C^2 differentiable [Kato, 1995]. Since the Hessian is continuous on a compact set, it is bounded, implying that $\nabla\mathcal{B}$ is L-Lipschitz continuous along the trajectory.

1. By the descent lemma for L-smooth functions and the variational property of the projection $\Pi_{\mathcal{D}_\varepsilon}$, we derive the sufficient ascent inequality: $\mathcal{B}(W^{(t+1)}) - \mathcal{B}(W^{(t)}) \ge \frac{1}{2\eta}\|W^{(t+1)} - W^{(t)}\|_F^2$. Noting that $\|\mathcal{G}_\eta\|_F^2 = \frac{1}{\eta^2}\|W^{(t)} - W^{(t+1)}\|_F^2$, the result follows.
2. Since $\mathcal{B}$ is continuous and bounded above on $\mathcal{D}_\varepsilon$ and the sequence is strictly increasing (non-stationary), the values converge. Since $\{W^{(t)}\} \subset \mathcal{D}_\varepsilon$, accumulation points exist. As $\|\mathcal{G}_\eta(W^{(t)})\|_F \to 0$, the fixed-point condition $W^* = \Pi_{\mathcal{D}_\varepsilon}(W^* + \eta\nabla\mathcal{B}(W^*))$ holds for any limit point W^*, verifying stationarity.
3. Summing the sufficient ascent inequality over T iterations yields $\mathcal{B}^* - \mathcal{B}(W^{(0)}) \ge \frac{\eta T}{2} \min_k \|\mathcal{G}_\eta(W^{(k)})\|_F^2$. Rearranging terms provides the $O(1/T)$ rate for the squared norm.

□

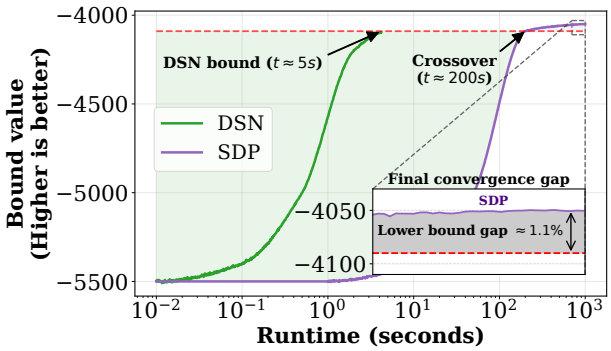


Figure 2: Time-to-quality comparison between DSN and SDP bounds. DSN reaches its plateau within seconds, while the SDP bound improves more slowly and overtakes after substantially longer runtime; the final bound gap remains small.

The assumption of a non-zero spectral gap holds generically. Results in random matrix theory show that for random graphs and generic perturbations, the Laplacian spectrum is simple with high probability [Luh *et al.*, 2025; Kato, 1995]. Eigenvalue degeneracy requires exact structural symmetry, which is rare in real-world instances. Nonetheless, for robustness, our implementation monitors the spectral gap $\gamma_t = \lambda_1(N(W)) - \lambda_2(N(W))$ at each iteration. If γ_t falls below a safeguard threshold $10^{-2}\|J\|_2$, we reduce the step-size η proportionally. Across all reported experiments, this safeguard was never triggered, confirming the validity of the smoothness assumption in practice; quantitative profiling is reported in §5.

SDP solves a convex relaxation and can produce very tight bounds; its computational cost typically scales cubically or worse, which limits scalability on large instances. In contrast, DSN optimizes a non-convex spectral objective with per-iteration cost dominated by sparse matrix–vector products, yielding near-linear scaling in $|E|$. As shown in Figure 2, DSN reached its converged bound within seconds ($t \approx 5$s), whereas the SDP took longer ($t \approx 200$s) to crossover. While SDP may eventually surpass DSN given sufficient runtime, the final bound gap remains small in our experiments, indicating that DSN provides a highly efficient approximation in the large-scale regime.

5 Experiments

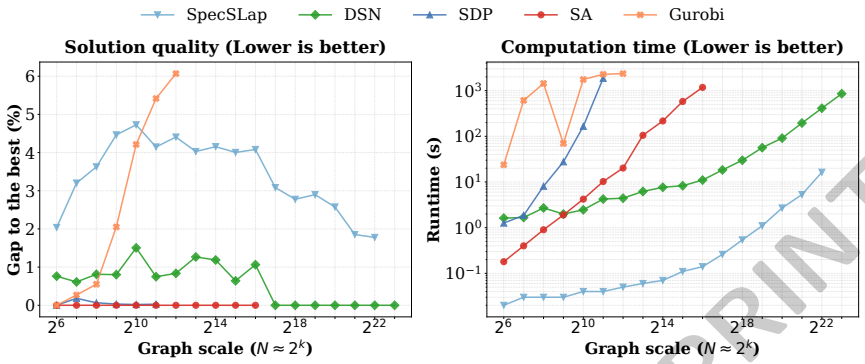
5.1 Experimental Setup

Datasets We evaluate DSN on three instance families:

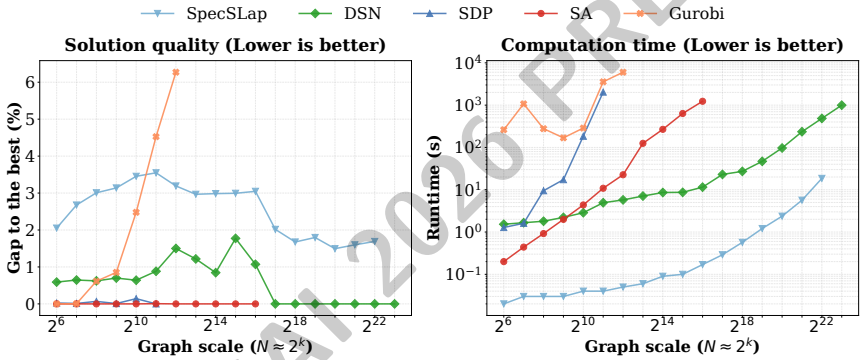
- Gset benchmark: 66 widely-used Max-Cut instances ($N \in [800, 40\,000]$) exhibiting diverse topologies, including random and toroidal graphs [Ye, 2003].

Instance	SpecSLap Energy	SpecSLap Time	DSN Energy	DSN Time	SDP Energy	SDP Time	SA Energy	SA Time	Gurobi Energy	Gurobi Time
Avg. (Gset)	-5090.5 ± 4.2	0.13 ± 0.01	-5515.0 ± 1.8	3.74 ± 0.15	-3038.5 ± 45.2[1]	1980.3 ± 15.4	-5674.7 ± 5.2	7.19 ± 0.24	-4573.8 ± 87.3[1]	2222.8 ± 42.8
Epinions	-589389.6 ± 245.0	0.08 ± 0.01	-606858.0 ± 12.5	6.32 ± 0.21	–	–	-606869.0 ± 28.4	1210.1 ± 14.5	–	–
Slashdot	-338955.4 ± 156.2	0.04 ± 0.01	-350960.5 ± 8.2	5.98 ± 0.18	–	–	-351208.0 ± 18.6	744.1 ± 9.2	–	–
Wikielec	-62068.4 ± 38.5	0.03 ± 0.01	-71524.8 ± 4.5	4.57 ± 0.12	–	–	-71537.0 ± 10.3	25.44 ± 0.65	$-71537.0^{\dagger}$	128.07 ± 2.1

Table 1: Benchmark performance on real and synthetic instances. We report final energy (lower is better) and wall-clock time for spectral baselines, DSN, and reference solvers. The symbol "–" indicates that the solver failed to return a feasible solution within a 2-hour time limit (Time Limit Exceeded) or ran out of memory (OOM). See footnote 1 for converged-subset breakdown.



(a) Barabási-Albert dataset.



(b) Erdős-Rényi dataset.

Figure 3: Scalability and efficiency on large-scale benchmarks (Barabási-Albert and Erdős-Rényi). DSN scales up to 2^{23}, uniquely bridging the gap between the speed of spectral heuristics and the quality of an exact solver. In contrast, combinatorial (Gurobi, SA) and SDP solvers exhibit prohibitive complexity, failing to scale beyond medium-sized instances. Standard spectral clustering (SpecSlap), though fast, yields significantly degraded solutions. "Gap to the best" is calculated relative to the best solution found by any solver in our evaluation.

- Real-world signed networks: Three large social graphs (Epinions, Slashdot, Wikielec) with up to 131K nodes and mixed-sign interactions, representing correlation clustering tasks [Leskovec *et al.*, 2010].
- Synthetic stress tests: Massive scale-free (Barabási-Albert) and random (Erdős-Rényi) graphs with mixed weights (up to $N \approx 10^6$) generated to validate scalability and robustness to degree heterogeneity.

Baselines We compare DSN against methods spanning three categories: (1) Static spectral (normalized signed Laplacian); (2) Combinatorial metaheuristics: Simulated Annealing (SA), providing high-quality references for solution optimality; (3) Exact solvers (Gurobi [Gurobi Optimization, LLC, 2024], time-limited) for small instances. To ensure strict fairness, all spectral variants utilize identical post-rounding refinement, and all heuristic baselines operate under strictly matched wall-clock time budgets.

Metrics The primary metric is the *Discrete energy* $E(s) = -s^\top Js$ (or Cut value), allowing direct quality comparison across all solver types. To validate the proposed theoretical mechanism, we report the *Spectral bound value* $\mathcal{B}(W)$, quantifying the tightness of the learned relaxation relative to static baselines. We also measure the *Optimality gap* $[E(s) - \mathcal{B}(W)]/|E(s)|$ and *Runtime* to assess scalability. The bound $\mathcal{B}(W) \leq \mathcal{E}^\star \leq E(s)$ further certifies that this gap upper-bounds the per-instance rounding error.

[1]SDP and Gurobi averages include zeros from OOM/timeout failures. On the subset where SDP successfully converges, the en-

Implementation details. We implement DSN in PyTorch with sparse matrix support. We set $\varepsilon = 10^{-5}$ for numerical stability and run $T = 300$ iterations of PGD with initial $\eta = 0.1$ and exponential decay (factor 0.99 every 10 steps). Results are averaged over 10 seeds. Profiling on `Slashdot` gives $\gamma_t \geq 2.61\times$ safeguard, $K_t \in [38, 56]$ matvecs/step (near-linear in $|E|$), $w_i \in [0.81, 81.5]$ (bulk near $\bar{w}$=1, std 1.18; hubs receive large weight per the localisation penalty, Eq. (10)), and $\|\nabla\mathcal{B}\|_F \in [3.61, 4.68]$, confirming Lemma 4.

Hardware Experiments were conducted on a high-performance computing node with dual Intel Xeon Gold 6246 CPUs and 1.5 TB of RAM. To accelerate spectral computations, we utilized NVIDIA A100 GPUs (40GB VRAM).

5.2 Scalability and Performance Trade-offs

We evaluate DSN against a spectrum of solvers ranging from exact combinatorial methods to fast spectral heuristics (as summarized in Table 1 and visualized in Figure 3).

Performance on standard benchmarks. On the Gset benchmark (66 instances), DSN achieves a mean energy of -5514.96, substantially outperforming both the static spectral baseline SpecSLap (-5090.54) and the SDP relaxation (-3038.54). Notably, DSN attains solution quality within **3%** of Simulated Annealing (SA) (-5674.66) while requiring only 3.74 seconds compared to SA's 7.19 seconds. On the converged subset above, DSN comes within 0.55% of Gurobi's certified optimum, but Gurobi hits the 2-hour cutoff on most other Gset instances; its time-limited incumbent (-4573.78 average) consequently lags DSN by over 17%, reflecting the practical scalability ceiling of branch-and-bound at this problem size.

Efficiency on real-world signed networks. DSN demonstrates consistent advantages on large-scale real datasets. For the `Epinions` graph (131K nodes), DSN obtains an energy of -606858.0, matching SA's solution (-606869.0) to within 0.002% while reducing runtime by a factor of **190**$\times$ (6.32s vs. 1210s). Similar patterns emerge on `Slashdot` and `Wikielec`, where DSN achieves energies of -350960.5 and -71524.8 respectively, both within 0.1% of SA, at a fraction of the computational cost. In contrast, SDP and Gurobi fail entirely on these large networks due to memory exhaustion and complexity limits.

Scalability to massive synthetic graphs. Figure 3 presents the central empirical finding of this work: DSN scales to graphs with $N = 2^{23}$ ($\approx 8.4 \times 10^6$ nodes) while maintaining competitive solution quality.

- On `Barabási-Albert` graphs at scale $N = 2^{23}$, DSN achieves an energy of -2134.8×10^6 in 856s. SpecSLap fails to produce results beyond 2^{22}. At $N = 2^{16}$ (the limit of SA's scalability), DSN obtains an energy of -17076.4K compared to SA's -17260.0K, a gap of 1.1%, while running **107**$\times$ faster (11.0s vs. 1177.7s).
- On `Erdȯs-Rényi (ER)` graphs, DSN shows similar robustness. At $N = 2^{23}$, DSN obtains an energy of -2276.1×10^6 in 998 seconds. Across all comparable scales, DSN consistently captures over **99%** of SA's solution quality while requiring only $3-5\%$ of the runtime.

High-precision solvers exhibit limitations in this regime: SDP fails beyond 2^{11} due to memory constraints, while Gurobi cannot scale past 2^{12} within practical time budgets.

These results delineate a fundamental trade-off in large-scale Ising optimization. SpecSLap occupies one extreme: sub-second runtimes, but solutions that lag $3 - 15\%$ behind the best achievable quality. SA and Gurobi occupy the opposite extreme: near-optimal solutions but with runtimes that grow prohibitively, rendering them infeasible beyond $N \approx 2^{16}$. DSN uniquely bridges this gap, scaling to 2^{23} while maintaining solution quality within $1-3\%$ of metaheuristic solvers at a fraction of their computational cost.

5.3 Ablation Study

Component	Variant	Energy (E)	Time (s)
Full	**DSN (Proposed)**	**-71524.8**	**4.57**
Initialization	DSN (Random init)	-70094.3	6.86
	DSN (Identity init ($W = I$))	-70952.6	5.48
Gradient	DSN (AutoDiff)	-71520.0	28.33
Schedule	DSN (Fixed LR ($\eta = 0.1$))	-70451.9	5.03
	DSN (Linear decay)	-71167.2	4.80

Table 2: Ablation study on `Wikielec`. The proposed DSN (degree init, analytic gradient, and exponential decay) achieves the best balance of solution quality and efficiency.

We isolate the impact of DSN's design choices via an ablation study on the *Wikielec* dataset (Table 2). Initializing with node degrees ($W^{(0)} = D$) proves essential, improving solution quality by 2.0% over random initialization and 0.8% over identity ($W = I$), confirming that degree-based weighting offers a geometrically informative warm-start. Our gradient (via Hellmann-Feynman) matches Automatic Differentiation (Autodiff) in solution quality while delivering a **6.2**$\times$ speedup (4.57s vs. 28.33s), by avoiding memory-intensive backpropagation through the iterative eigensolver. The exponential decay schedule yields the lowest energy (-71524.8), outperforming linear decay and fixed rates by 0.5% and 1.5%, respectively, balancing early-stage exploration with late-stage exploitation for fine-grained spectral refinement.

6 Conclusion

We introduced DSN, which treats graph normalization as a learnable diagonal metric to bridge the gap between spectral efficiency and combinatorial precision. Using the Hellmann-Feynman theorem, DSN achieves exact analytical gradients with near-linear complexity, enabling effective optimization on massive graphs with up to 8.4×10^6 nodes. We established that under a generic spectral-gap condition, the DSN objective is smooth and guaranteed to converge to stationary points. Empirically, DSN consistently outperforms static spectral baselines and matches state-of-the-art metaheuristics within 1–3% while reducing computational cost by up to two orders of magnitude.

ergies match theoretical expectations: **Gurobi** (-4372.15) < **SDP** (-4359.64) < **DSN** (-4348.20) < **SpecSLap** (-4125.70).

Acknowledgments

This work was partially supported by NSF AMPS-2229075, Commonwealth Cyber Initiative (CCI) Awards, and the VCU Quest Award.

References

[Ackley *et al.*, 1985] David H. Ackley, Geoffrey Everest Hinton, and Terrence Joseph Sejnowski. A learning algorithm for boltzmann machines. *Cognitive Science*, 9(1):147–169, 1985.

[Bansal *et al.*, 2004] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Mach. Learn.*, 56(1–3):89–113, June 2004.

[Cucuringu *et al.*, 2019] Mihai Cucuringu, Peter Davies, Aldo Glielmo, and Hemant Tyagi. Sponge: A generalized eigenproblem for clustering signed networks. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 1088–1098. PMLR, 16–18 Apr 2019.

[Cucuringu *et al.*, 2021] Mihai Cucuringu, Apoorv Vikram Singh, Déborah Sulem, and Hemant Tyagi. Regularized spectral methods for clustering signed networks. *Journal of Machine Learning Research*, 22(1):1–79, January 2021.

[Dai *et al.*, 2017] Hanjun Dai, Elias Boutros Khalil, Yuyu Zhang, Bistra Dilkina, and Le Song. Learning combinatorial optimization algorithms over graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6351–6361, Red Hook, NY, USA, 2017. Curran Associates Inc.

[Delorme and Poljak, 1993] Charles Delorme and Svatopluk Poljak. Laplacian eigenvalues and the maximum cut problem. *Mathematical Programming*, 62(1–3):557–574, February 1993.

[Goemans and Williamson, 1995] Michel Xavier Goemans and David Paul Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, November 1995.

[Gurobi Optimization, LLC, 2024] Gurobi Optimization, LLC. Gurobi optimizer reference manual. https://www.gurobi.com, 2024.

[Hopfield, 1982] John J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982.

[Ionescu *et al.*, 2015] Catalin Ionescu, Orestis Vantzos, and Cristian Sminchisescu. Matrix backpropagation for deep networks with structured layers. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2965–2973, 2015.

[Karalias and Loukas, 2020] Nikolaos Karalias and Andreas Loukas. Erdos goes neural: an unsupervised learning framework for combinatorial optimization on graphs. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, pages 6659–6672, Red Hook, NY, USA, 2020. Curran Associates Inc.

[Kato, 1995] Tosio Kato. *Perturbation Theory for Linear Operators*, volume 132. Springer, 1995.

[Kirkpatrick *et al.*, 1983] Scott Kirkpatrick, Charles Daniel Gelatt, and Mario Pietro Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[Kunegis *et al.*, 2010] Jérôme Kunegis, Stephan Schmidt, Andreas Lommatzsch, Jürgen Lerner, Ernesto W. De Luca, and Sahin Albayrak. *Spectral Analysis of Signed Graphs for Clustering, Prediction and Visualization*, pages 559–570. Society for Industrial and Applied Mathematics, 2010.

[Leskovec *et al.*, 2010] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Signed networks in social media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, page 1361–1370, New York, NY, USA, 2010. Association for Computing Machinery.

[Lucas, 2014] Andrew Lucas. Ising formulations of many NP problems. *Frontiers in Physics*, 2:5, 2014.

[Luh *et al.*, 2025] Kyle Luh, Ryan Vogel, and Alan Yu. Eigenvalue gaps of random perturbations of large matrices. *Random Matrices: Theory and Applications*, 14(04):2550017, 2025.

[Magnus, 1985] Jan R. Magnus. On differentiating eigenvalues and eigenvectors. *Econometric Theory*, 1(2):179–191, 1985.

[Milgrom and Segal, 2002] Paul Milgrom and Ilya Segal. Envelope theorems for arbitrary choice sets. *Econometrica*, 70(2):583–601, 2002.

[Miyato *et al.*, 2018] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

[Schuetz *et al.*, 2022] 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, 2022.

[Swendsen and Wang, 1986] Robert Haakon Swendsen and Jian-Sheng Wang. Replica monte carlo simulation of spin-glasses. *Phys. Rev. Lett.*, 57:2607–2609, Nov 1986.

[Trevisan, 2012] Luca Trevisan. Max cut and the smallest eigenvalue. *SIAM Journal on Computing*, 41(6):1769–1786, 2012.

[Wainwright and Jordan, 2008] Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 12 2008.

[Wang *et al.*, 2019] Wei Wang, Zheng Dang, Yinlin Hu, Pascal Fua, and Mathieu Salzmann. *Backpropagation-friendly eigendecomposition*, pages 3162–3170. Curran Associates Inc., Red Hook, NY, USA, 2019.

[Ye, 2003] Yinyu Ye. The gset dataset. https://web.stanford.edu/~yyye/yyye/Gset/, 2003.

[Zhang and George, 2002] Guoping Zhang and Thomas F. George. Breakdown of the hellmann-feynman theorem: Degeneracy is the key. *Physical Review B*, 66:033110, Jul 2002.