A SELECTIVE-AND-ADAPTIVE FRAMEWORK FOR DISTANCE-BASED RANDOM GRAPH MATCHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph matching is an NP-hard problem of recovering vertex correspondences between correlated graphs. Degree-profile methods, which compare vertices based on their local statistics, are attractive due to their exact recovery guarantees and computational efficiency. However, these methods typically require computing distances between all pairs of vertices, and their performance may deteriorate in high-noise regimes. To address these challenges, we propose a two-stage framework for random graph matching. First, selective distance computation preserves the exact recovery guarantee of degree-profile matching while reducing the complexity from $O((n^2 + nd^2)\log n)$ to $O(n^2 + nd^2)$. When recovery fails under high noise, the computed distances warm-start a continuous optimization method with convergence guarantees. Under suitable assumptions, we establish exact recovery for stationary points with sufficiently small objective gaps to the ground truth. To the best of our knowledge, this is the first such guarantee for near-optimal stationary points in nonconvex doubly stochastic graph matching. Experiments on synthetic and real-world networks show $10\times$–$20\times$ speedups in low-noise regimes and substantially improved robustness under high noise.

1 INTRODUCTION

Graph matching, also referred to as network alignment (Feizi et al., 2019), aims to identify correspondences between related graphs. It has found broad applications in computer vision, biological networks, natural language processing, knowledge graphs, and other intelligent information-processing tasks. Graph matching between adjacency matrices A and B can be formulated as the quadratic assignment problem (QAP), defined as

$$\max_{P\in\Pi_{n\times n}} \langle A, PBP^\top\rangle, \tag{1}$$

with $\Pi_{n\times n}$ denoting the permutation matrix set and $\langle\cdot,\cdot\rangle$ the Frobenius inner product. For convience, we mark $\mathcal{Z}(P) = \langle A, PBP^\top\rangle = \mathrm{Tr}(A^\top PBP^\top)$ where $\mathrm{Tr}(\cdot)$ is trace. Although QAP is NP-hard and inapproximable within $2^{\log^{1-\epsilon}(n)}$ for $\epsilon > 0$ in the worst case (Makarychev et al., 2014), these theoretical limits often vary from practical scenarios. Real-world networks typically exhibit latent stochastic structures rather than adversarial worst-case patterns. This distinction has motivated a shift in recent scholarship towards high-probability recovery of the perfect matching (Cullina & Kiyavash, 2017; Lyzinski et al., 2014) rather than worst-case optimization.

A common approach to graph matching replaces the original quadratic assignment problem with more tractable formulations based on pairwise vertex similarities or distances. Representative examples include methods based on learned structural embeddings (Heimann et al., 2018), spectral information (Hermanns et al., 2023; Shen et al., 2025), optimal transport (Xu et al., 2019b;a; Song et al., 2026), and continuous optimization (Gold & Rangarajan, 1996; Lu et al., 2016; Shen et al., 2024). However, many of these methods lack theoretical guarantees for exact recovery. In contrast, Ding et al. (2021) developed an $\mathcal{O}((nd^2 + n^2)\log n)$-time algorithm that, by comparing local vertex statistics, perfectly recovers the true vertex correspondence with high probability under suitable conditions. In higher-noise regimes, they further introduce a continuous linear-assignment-based *clean-up* procedure to refine the matching obtained in the initial stage, yielding improved empirical performance.

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.

However, this framework Ding et al. (2021; 2025) has two key limitations. First, it requires constructing the full n^2-scale pairwise distance matrix, which can be computationally prohibitive for large graphs. Second, the clean-up procedure lacks a convergence guarantee, and its suboptimality relative to the global optimum remains theoretically uncharacterized. Motivated by the scalability and robustness limitations of existing degree profile-based methods, we develop **Li**ght **D**egree **P**rofile (LiDP) as a two-stage framework:

1. **Selective computation in low-noise regimes.** We prioritize distance computations for candidate pairs with higher matching likelihoods and stop once a perfect matching is certified. For profile-distance matching, this avoids exhaustive evaluation and $O(n^2 \log n)$ sorting, reducing the matrix-construction and selection cost to $O(n^2)$ while preserving exact recovery in low-noise regimes. Empirically, evaluating only 3%–5% of profile distances yields a 10–$20\times$ speedup. The same strategy extends to other distance-based methods, achieving over $20\times$ speedup for bin-distance matching (Ding et al., 2025).
2. **Convergence-guaranteed continuous optimization in high-noise regimes.** In the refining optimization stage, we relax the original formulation to leverage information from the previous step and adopt an adaptive step-size scheme to ensure convergence. Beyond convergence, we establish, to the best of our knowledge, the first exact-recovery guarantee for near-optimal stationary points of the nonconvex doubly stochastic QAP relaxation.

Together, these two stages address the computational and theoretical limitations of existing degree profile-based methods, yielding a scalable and theoretically grounded framework across noise regimes.

2 Related Work

Random Graph Matching. Provably efficient algorithms for random graph matching exhibit different statistical and computational trade-offs. Local-statistics methods, such as degree-profile matching (Ding et al., 2021; 2025), are simple and interpretable, but typically require relatively strong correlation or density and involve exhaustive comparisons over candidate vertex pairs. Neighborhood-, tree-, and subgraph-based methods exploit richer local structures and extend recovery guarantees to more challenging regimes, at the cost of more elaborate constructions and higher computational overhead (Ganassali & Massoulié, 2020; Mao et al., 2023; 2026). More recently, iterative amplification methods achieve polynomial-time exact recovery under substantially weaker correlation conditions, but rely on multi-stage procedures that are considerably more intricate and model-specific (Ding & Li, 2026). Our work pursues a complementary direction: improving the scalability and robustness of degree-profile matching by reducing unnecessary pairwise computations in low-noise regimes and providing a convergence-guaranteed refinement for more challenging regimes.

Graph Matching Applications. Graph matching has found applications across a variety of domains. In computer vision, it has been used for activity detection (Chen & Grauman, 2012), shape matching (Michel et al., 2011), and fabricated-image detection (Shen et al., 2020). It has also been applied to medical imaging, where graph matching is used to align structural representations for representation learning (MH Nguyen et al., 2024), and to autonomous driving, where it facilitates multimodal feature alignment for 3D object detection (Song et al., 2023). In network and knowledge applications, graph matching supports network de-anonymization (Narayanan & Shmatikov, 2008) and cross-lingual knowledge graph alignment (Xu et al., 2019c). In computational biology, aligning protein–protein interaction networks across species can reveal conserved structures and assist protein-function prediction (Singh et al., 2008; Ma & Liao, 2020). More recently, multi-graph matching has also been used to improve vision–language alignment in medical multimodal models (Nguyen et al., 2024).

3 Preliminary

3.1 Erdős-Rényi Graph Model

The statistical analysis of graph matching has recently emerged as a significant area of research, driven by applications in social networks and biology (Pedarsani & Grossglauser, 2011; Barak et al.,

2019). Recent work typically assumes that the two graphs, A and B, are generated from a random model. A fundamental and widely studied framework is the correlated Erdős-Rényi graph model:

Definition 1 (Correlated Erdős-Rényi Model $\mathcal{G}(n, q, s)$)**.** *Let n be the number of vertices, and let $s \in [0, 1]$ and $p \in [0, 1)$ be probability parameters. Let A and B denote the adjacency matrices of a pair of correlated random graphs $\mathcal{G}(n, q)$ with the same vertex $[n]$. Denote a permutation $\pi^* : [n] \to [n]$. Conditioned on A, for all $i < j$, $B_{\pi^*(i)\pi^*(j)}$ are independent and have the distribution*

$$B_{\pi^*(i)\pi^*(j)} \sim \begin{cases} \textit{Bern}(s) & \textit{if } A_{ij} = 1, \\ \textit{Bern}\left(\frac{q(1-s)}{1-q}\right) & \textit{if } A_{ij} \neq 1, \end{cases} \tag{2}$$

where Bern(p) denotes a Bernoulli distribution with mean p.

3.2 Distance matrix based degree profile

Ding et al. (2021) propose a random graph matching method based on *degree profiles* for recovering vertex correspondences. The degree profile of a vertex summarizes the empirical distribution of the degrees of its neighbors and serves as a local structural signature. The underlying intuition is that, for a truly matched pair of vertices, correlation between the two graphs induces many corresponding common neighbors, leading to similar degree profiles. By contrast, for an incorrectly matched pair, the two neighborhoods are largely independent, so their degree profiles typically differ. After discretizing the profiles, vertex pairs can therefore be compared through a suitable profile-based distance, reducing graph matching to a sorting-based procedure.

This framework accommodates different choices of pairwise distance. In this paper, we consider two representative examples. For homogeneous Erdős-Rényi graphs, we use the degree profile distance of Ding et al. (2021). Let $\bar{\mu}_i$ and $\bar{\nu}_j$ denote the degree profiles of vertex i in graph A and vertex j in graph B, respectively. Given a partition $I_1, \ldots, I_L$ of the profile domain (detailed in Appendix B), the corresponding entry of the distance matrix is

$$D_{ij} = \sum_{\ell \in [L]} \left|\bar{\mu}_i(I_\ell) - \bar{\nu}_j(I_\ell)\right| := \left|\bar{\mu}_i - \bar{\nu}_j\right|_{1,L}. \tag{3}$$

For inhomogeneous random graphs, we instead use the bin-distance of Ding et al. (2025), which is based on a balls-into-bins construction (detailed in Appendix B).

Given a vertex-pair distance, the exhaustive baseline provides a generic reference procedure: it computes all n^2 pairwise distances, sorts them, and checks whether the n smallest entries form a valid permutation. Ding et al. (2021) show that whenever such a permutation is obtained, the resulting matching is a perfect matching. Otherwise, the full distance matrix D is retained for subsequent optimization. Thus, the same baseline applies to both distance constructions above, as summarized in Algorithm 1.

Algorithm 1 Graph matching via a pairwise distance matrix

Require: Graphs A and B on n vertices; a pairwise distance $d(\cdot, \cdot)$.
Ensure: A permutation $\widehat{\pi}$ of n elements.

```
Compute D_ik = d(i,k) for all i,k ∈ [n].
Sort {D_ik : i,k ∈ [n]} and let S be the indices of the smallest n entries.
if S = {(i, π̂(i)) : i ∈ [n]} forms a permutation π̂ then
    return π̂.
else
    return D.
end if
```

Algorithm 2 Momentum-Adaptive Continuous Optimization

Require: Adjacency matrices $A, B \in \mathbb{R}^{n \times n}$, initial D, maximum iterations T, tolerance ϵ.
Ensure: Permutation matrix P.

```
P^0 ← P(−D)
for t = 0 to T do
    P̃^t ← P(AP^tB)
    Compute step size α^t via (8)
    P^{t+1} ← P^t + α^t(P̃^t − P^t)
    if ‖P^{t+1} − P^t‖_F / ‖P^t‖_F < ε then break
end for
P ← P(P^{t+1})
return P
```

3.3 CLEAN-UP STRATEGY

Since the theoretical guarantees of Algorithm 1 are restricted to limited noise regimes, Ding et al. (2021) employed an iterative clean-up strategy to enhance robustness in high-noise environments. Initialized with the permutation matrix from Algorithm 1, this procedure refines the matching by iteratively solving a linear assignment problem,

$$P^0 = \arg\max_{P \in \Pi_{n\times n}} \langle P, -D\rangle,\ P^{t+1} = \arg\max_{P \in \Pi_{n\times n}} \langle P,\ AP^tB\rangle, \tag{4}$$

which aims to maximize the number of common neighbors consistent with the current alignment.

4 SELECTIVE-AND-ADAPTIVE MATCHING METHOD

4.1 SELECTIVE COMPUTATION SCHEME

The degree-profile-based algorithms (Ding et al., 2021; 2025) find the optimal matching if and only if the positions of the n smallest elements of the distance matrix form a permutation matrix. Motivated by this property, we find that the full n^2 distance matrix need not be computed to attain the same theoretical recovery guarantees. Our approach is based on two principles: (i) prioritizing the computation of distances that are likely to belong to the optimal matching, and (ii) designing a sound certification procedure that can detect when all distances required by the optimal matching have been revealed.

The key to selective distance computation is to identify distances associated with node pairs that are more likely to be matched. Our prioritization is motivated by two observations. First, matched nodes tend to have similar degrees, making pairs with smaller degree differences more plausible matching candidates. Second, among the n smallest distances computed thus far, any rows or columns that remain uncovered must contain plausible matching candidates; otherwise, these entries cannot collectively support a permutation matrix. This is supported by the following theorem.

Theorem 1. *Let D be an $n \times n$ real matrix, and assume that its n-th smallest entry is strictly smaller than its $(n+1)$-st smallest entry. Then the positions of the n smallest entries of D form a permutation matrix if and only if each of these entries is the unique minimum in its row and column.*

Motivated by these two observations, we compute distances progressively in ascending order of degree difference. At iteration d, we compute only those entries whose degree difference is at most d and whose corresponding rows or columns are not yet covered by the current set $\mathcal{S}_n^{(d)}$ of the n smallest computed distances. This process continues until either $\mathcal{S}_n^{(d)}$ is identified as corresponding to the optimal matching or it is determined that the optimal matching cannot be found. The whole procedure is detailed in Figure 1.

To check whether $\mathcal{S}_n^{(d)}$ corresponds to the optimal matching consistent with DP (Ding et al., 2021), we only need to verify, based on Theorem 1, whether the largest element in $\mathcal{S}_n^{(d)}$ is the smallest distance within its corresponding row and column. Algorithm 3, along with Figure 2, details the subroutine used for this verification. Moreover, this verification process covers some potentially relevant entries that were previously left uncomputed, thereby preventing omissions.

Theorem 2 (Guarantee of Selective Computation Scheme)**.** *Let $D \in \mathbb{R}^{n\times n}$ be the distance matrix whose n smallest entries, indexed by $\mathcal{S}$, induce a permutation matrix. Then the set $\mathcal{S}_n$ returned by the selective computation scheme satisfies*

$$\mathcal{S}_n = \mathcal{S}.$$

Thus, selective computation yields the same matching as full computation.

The assumption ensures that the distance matrix permits exact recovery. Theorem 2 shows that our scheme attains the same statistical guarantees as (Ding et al., 2021; 2025) under the same noise regimes, while substantially reducing the computational cost without compromising the probability of exact recovery.

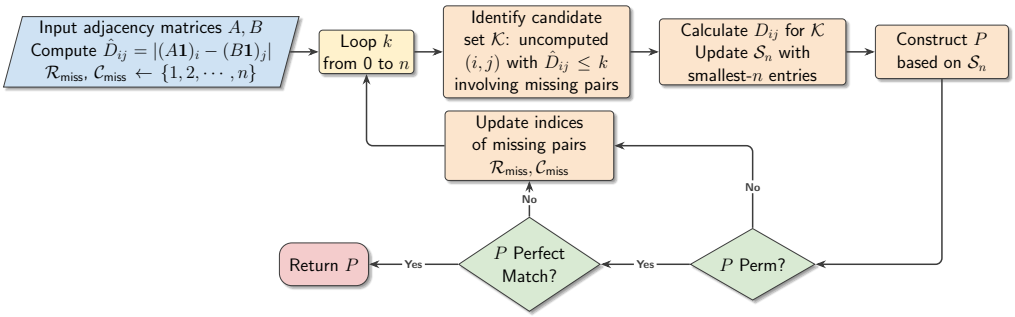


Figure 1: The exact recovery procedure of the LiDP algorithm. $\mathcal{R}_{\text{miss}}$ and $\mathcal{C}_{\text{miss}}$ represent the row and column indices from $\{1, \ldots, n\}$ that are not yet present in the current construction of P.

Algorithm 3 Check Perfect

Require: Adjacency matrices A, B, distance metric $d(\cdot, \cdot)$, Current matching candidates $\mathcal{I}, \mathcal{J}$ and corresponding distance $\mathcal{D}$
Ensure: Updated $\mathcal{I}$, $\mathcal{J}$, $\mathcal{D}$ and Boolean flag $isPerfect$
Initialization $isPerfect \leftarrow$**True**
Let (r^*, c^*) be the last element of $(\mathcal{I}, \mathcal{J})$
for $k = 1$ **to** n **do**
  Compute $d_{row} = D_{r^*,k}$ and $d_{col} = D_{k,c^*}$ by the defined distance metric
  if $d_{row} < \text{last}(\mathcal{D})$ **then**
    Remove the current last element from $\mathcal{J}$ and $\mathcal{D}$
    Insert d_{row} to $\mathcal{D}$ and maintain order, then insert k to $\mathcal{J}$ corresponding to $\mathcal{D}$
    $isPerfect \leftarrow$**False**
  end if
  if $d_{col} < \text{last}(\mathcal{D})$ **then**
    Remove the current last element from $\mathcal{I}$ and $\mathcal{D}$
    insert d_{col} to $\mathcal{D}$ and maintain order, then insert k to $\mathcal{I}$ corresponding to $\mathcal{D}$
    $isPerfect \leftarrow$**False**
  end if
end for
return $\mathcal{I}$, $\mathcal{J}$, $\mathcal{D}$, $isPerfect$

4.2 Momentum-Adaptive Continuous Optimization

While the selective computation scheme ensures exact recovery with high probability under moderate noise, the perturbations exceeding the theoretical threshold may prevent finding the global optimum. Nevertheless, this resulting approximate matching provides a highly informative warm start. To refine this candidate solution and rectify remaining mismatches, we relax the problem domain from $\Pi_{n\times n}$ to doubly stochastic matrix domain

$$\mathcal{B}_n := \left\{X \in \mathbb{R}^{n\times n} : X \geq 0,\ X\mathbf{1} = \mathbf{1},\ X^\top \mathbf{1} = \mathbf{1}\right\} \tag{5}$$

and extend the clean-up strategy (Ding et al., 2021) to a momentum-based continuous optimization procedure that incorporates an adaptive step size, which is a kind of Frank-Wolfe algorithms (Frank et al., 1956). Let P^t denote the doubly stochastic matrix at iteration t. The update rule is defined as:

$$P^{t+1} = P^t + \alpha^t(\tilde{P}^t - P^t), \tag{6}$$

$$\tilde{P}^t = \mathcal{P}(AP^tB) = \arg\max_{P\in\Pi_{n\times n}} \langle P, AP^tB\rangle, \tag{7}$$

$$\alpha^t = \arg\max_{\alpha\in[0,1]} \mathcal{Z}(P^t + \alpha(\tilde{P}^t - P^t)). \tag{8}$$

(7) is solved by the Hungarian algorithm, yielding a permutation matrix $\tilde{P}^t$. According to the convergence analysis in the following theorem, the sequence $\{P^t\}$ generated by the above rules

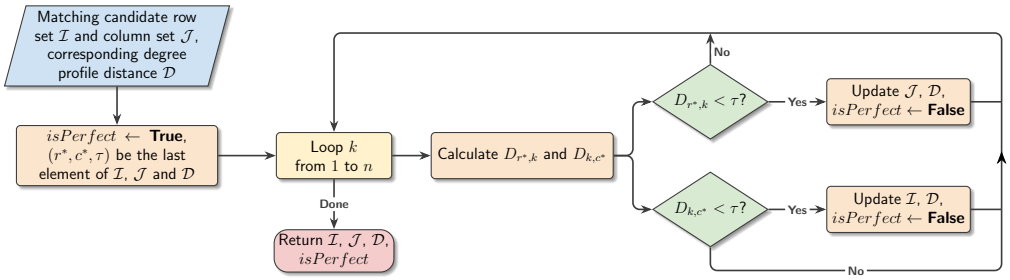


Figure 2: Check perfect algorithm description. The input $\mathcal{I}$, $\mathcal{J}$ are the row and column indices of non-zero element in P, its corresponding distance is $\mathcal{D}$.

is guaranteed to converge to a first-order stationary point. The whole algorithm is summarized in Algorithm 2.

Theorem 3 (Momentum-Adaptive continuous optimization convergence property)**.** *Let $A, B \in \mathbb{R}^{n\times n}$ be two symmetric adjacency matrices. For arbitrary initialization D, consider the sequence $\{P^t\}$ generated by Algorithm 2. Then, every accumulation point of $\{P^t\}$ is a first-order stationary point of $\mathcal{Z}$ over $\mathcal{B}_n$.*

We next investigate conditions under which a first-order stationary point of $\mathcal{Z}$ yields the ground-truth matching. Under the separation condition (9) stated below, a sufficiently small objective gap relative to the ground truth guarantees exact recovery for any first-order stationary point over $\mathcal{B}_n$.

Theorem 4 (Exact Recovery from First-Order Stationarity)**.** *Let $A, B \in [0,1]^{n\times n}$ be symmetric with zero diagonal. For convenience, set the ground-truth correspondence as I. Suppose that, for some $c_n > 0$,*

$$\mathcal{Z}(I) - \mathrm{Tr}(APBQ^\top) \geq c_n\big(2n - \mathrm{Tr}(P) - \mathrm{Tr}(Q)\big), \qquad \forall P, Q \in \Pi_{n\times n}. \tag{9}$$

If a first-order stationary point $\widehat{X}$ of $\mathcal{Z}$ over $\mathcal{B}_n$ satisfies

$$\mathcal{Z}(I) - \mathcal{Z}(\widehat{X}) < \frac{c_n^2}{2}, \tag{10}$$

then $\widehat{X} = I$.

4.3 The Proposed Selective-and-Adaptive Framework

Algorithm 4 integrates these two strategies into a unified framework. It employs a selective computation scheme that permits early termination when a candidate passes. If verification remains unsuccessful after the computed fraction is progressively increased to 20%, further selective computation is empirically unlikely to yield successful certification. LiDP then computes the remaining entries to complete the distance matrix and switches to the momentum-adaptive continuous optimization stage, using the precomputed distance information as a warm start. By avoiding sorting the entire distance matrix, we eliminate the $O(n^2 \log n)$ sorting cost included in DP (Ding et al., 2021). Consequently, the computational complexity of the first stage is only $O(n^2 + nd^2)$, dominated by distance computations. The second stage has a computational complexity of $O(n^3)$, primarily due to matrix multiplications.

5 Experiments

We evaluate the performance of the proposed LiDP algorithm from the following three aspects:

- **Q1.** How does LiDP benchmark against the standard distance-based algorithm in low-noise regimes?

Algorithm 4 Selective-and-Adaptive Matching Algorithm (LiDP Algorithm)

Require: Adjacency matrices $A, B \in \{0,1\}^{n\times n}$, distance metric $d(\cdot,\cdot)$, maximum iterations T, tolerance ϵ
Ensure: Permutation matrix P
Compute $\hat{D}_{ij} = |\,(A\mathbf{1})_i - (B\mathbf{1})_j\,|$.
Initialize $\mathcal{S}_n \leftarrow \varnothing$, $\Omega \leftarrow \varnothing$, $\mathcal{R}_{\text{miss}} \leftarrow [n]$, $\mathcal{C}_{\text{miss}} \leftarrow [n]$
▷ # *Distance matrix constructed by selective computation*
for $k = 0$ **to** $\max(\hat{D})$ **do**
  $\mathcal{K} \leftarrow \{(i,j) \mid (i \in \mathcal{R}_{\text{miss}} \vee j \in \mathcal{C}_{\text{miss}}) \wedge (\hat{D}_{ij} \le k)\} \setminus \Omega$
  for all $(i,j) \in \mathcal{K}$ **do**
    Compute D_{ij} by the defined distance metric and add (i,j) to Ω.
    Update $\mathcal{S}_n$ to keep the n-smallest pairs corresponding to D
  end for
  Extract $\mathcal{I}_n$ and $\mathcal{J}_n$ from $\mathcal{S}_n$, and let $\mathcal{D}_n$ be the n-smallest elements in D
  if $\mathcal{S}_n$ forms a valid bijection **then** $(\mathcal{I}_n, \mathcal{J}_n, \mathcal{D}_n, isPerfect) \leftarrow$ Check Perfect$(\mathcal{I}_n, \mathcal{J}_n, \mathcal{D}_n)$
    if $isPerfect$ **then** Construct P based on $\mathcal{S}_n$; **return** P
  $\mathcal{R}_{\text{miss}} \leftarrow \{1, \cdots, n\} \setminus \mathcal{I}_n$, $\mathcal{C}_{\text{miss}} \leftarrow \{1, \cdots, n\} \setminus \mathcal{J}_n$
end for
▷ # *If needed, refine permutation matrix by continuous optimization*
$P \leftarrow$ Algorithm 2 with input (A, B, D, T, ϵ); **return** P

- **Q2.** How robust is the momentum-adaptive continuous optimization strategy under higher noise variations?

- **Q3.** Can the selective computation scheme accelerate other distance-based methods?

The appendix provides visual comparisons of LiDP with existing graph-matching methods, experiments on graph-density sensitivity, and evaluations of our framework under bin-distance (Ding et al., 2025) on both synthetic and real-world networks.

Datasets include synthetic correlated Erdős-Rényi graphs $\mathcal{G}(n, p, 1-\delta)$ and two real-world datasets: the Slashdot social network and the yeast PPI network. For the real-world cases, we utilize the following as parent graphs: a subgraph induced by the first 750 users from Slashdot, and the high-confidence *S. cerevisiae* PPI network (Collins et al., 2007). The relevant setting details are summarized in Table 1. For the Erdős-Rényi experiments, we report the noise level using $\sqrt{\delta}$, while δ itself is used for the other datasets.

Table 1: Summary of the datasets and experimental settings. δ is the deletion probability .

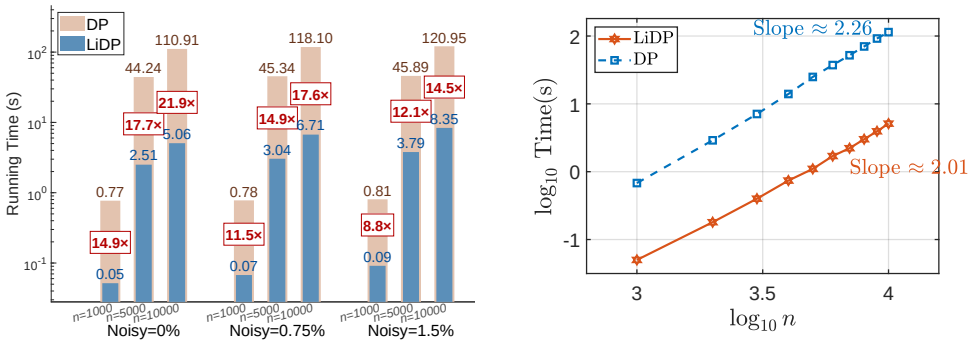
Dataset	Dims (n)	Noise Level
Synthetic	$[1000, 10000]$	$\sqrt{\delta} \in [0, 0.4]$
Slashdot	750	$\delta \in [0, 0.4]$
Yeast PPI	1004	$\delta \in \{0\%, 5\%, 10\%, 15\%\}$

Baselines. For the synthetic and Slashdot datasets, we benchmark against the standard DP and DP+ (Ding et al., 2021) to validate the robustness of our clean-up strategy. For the Yeast PPI network, we evaluate performance against comprehensive state-of-the-art methods, including GA (Gold & Rangarajan, 1996), IsoRank (Singh et al., 2008), AIPFP (Leordeanu et al., 2009; Lu et al., 2016), GHOST (Patro & Kingsford, 2012), FAQ (Vogelstein et al., 2015), MAGNA++ (Vijayan et al., 2015), DSPFP (Lu et al., 2016), S-GWL (Xu et al., 2019a), GWL(Xu et al., 2019b), GRASP (Hermanns et al., 2023), ASM (Shen et al., 2024), and LoBCD-GW (Song et al., 2026). All methods were benchmarked in terms of runtime using MATLAB R2024b on an Intel Core i7-12800HX processor.

5.1 Low Noise Regime Comparison

We compare the proposed LiDP algorithm, equipped with the 'Check Perfect' subroutine, with the DP algorithm of (Ding et al., 2021) on correlated Erdős-Rényi graphs $\mathcal{G}(n, q, 1-\delta)$. The comparison focuses on the perfect-matching regime, where both methods recover the ground-truth matching.

Figure 3(a) demonstrates the substantial computational advantage of LiDP over DP. In scenarios where the standard DP algorithm successfully identifies a perfect matching, our method achieves a remarkable speedup of up to $21\times$. Moreover, Table 2 indicates that the acceleration of LiDP gains from only calculating a small fraction of distances.



(a) Speedup of our LiDP over DP across different n and noise levels on Erdős-Rényi graph, values inside the red boxes represent the speedup ratios.

(b) Empirical scalability comparison between DP and LiDP on Erdős-Rényi graphs with different problem size.

Figure 3: Runtime scalability and speedup comparison between DP and LiDP on correlated Erdős-Rényi graphs. The results evaluate the effect of graph size and noise level on the computational efficiency of both methods.

Table 2: Comparison of computed distances between DP and LiDP on Erdős-Rényi graphs.

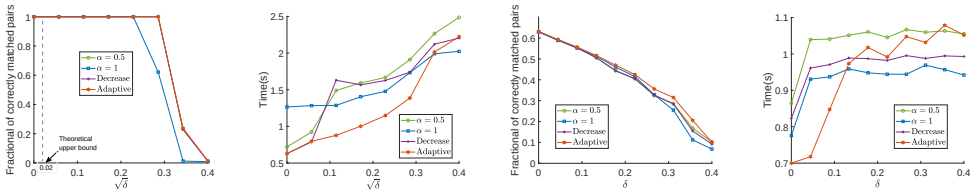
Problem Size	**DP**	**LiDP**		**Cost Reduction**	
		Noisy 0%	**Noisy 2%**	**Noisy 0%**	**Noisy 2%**
$n = 1{,}000$	1.00×10^6	4.31×10^4	5.08×10^4	95.69%	94.92%
$n = 5{,}000$	2.50×10^7	8.34×10^5	1.02×10^6	96.66%	95.92%
$n = 10{,}000$	1.00×10^8	3.10×10^6	3.86×10^6	96.90%	96.14%

We further investigate the empirical scalability of LiDP by measuring its runtime as the graph size n increases from 10^3 to 10^4, as shown in Figure 3(b). These results show that LiDP exhibits an empirical complexity of approximately $O(n^2)$, while DP is approximately $O(n^{2.26})$. This efficiency gain supports that our selective computation strategy is better suited for recovering the ground-truth of large-scale graph matching problems.

5.2 Continuous Optimization Comparison

Figure 4 compares fixed step size with the proposed momentum-adaptive continuous optimization strategy and shows that the adaptive strategy improves robustness under increasing noise. For the fixed-step baseline, we set the step size to $\alpha = 0.5, 1$ throughout. For the decrease step size, we set $\alpha^t = \alpha^{t-1}/10$.

As shown in Figure 4(a) and 4(c), our proposed strategy effectively mitigates the impact of structural noise. Although perfect recovery is no longer guaranteed beyond the theoretical upper bound, LiDP variants consistently outperform the fixed step size across all noise levels, exhibiting a slower decay in accuracy as δ increases. From Figure 4(b) and 4(d), our momentum-adaptive strategy induces a moderate increase in computational time. However, this overhead is justifiable given the substantial gains in accuracy.



(a) ER Accuracy (b) ER time (c) Slashdot Acc. (d) Slashdot time

Figure 4: Accuracy and runtime improvement of momentum-adaptive step size on Erdős-Rényi graph and Slashdot network, the dashed vertical line indicates the theoretical upper bound for noise tolerance.

To assess the performance of the proposed algorithm in practical scenarios, we extend our evaluation to the Yeast PPI matching. Table 3 presents the node accuracy and runtime comparisons between LiDP and several classic baseline methods under varying noise levels, which also reflects the advantages of our algorithm on real-world graphs.

Table 3: Comparisons on yeast PPI with node accuracy (%) and runtime (s).

Yeast network	0% noise		5% noise		10% noise		15% noise	
Methods	Node Acc	time	Node Acc	time	Node Acc	time	Node Acc	time
GA (Gold & Rangarajan, 1996)	24.0%	16s	14.0%	12.2s	11.8%	12.3s	9.6%	12.3s
IsoRank (Singh et al., 2008)	7.8%	0.3s	2.3%	0.3s	0.1%	0.3s	0.1%	0.4s
AIPFP (Leordeanu et al., 2009; Lu et al., 2016)	56.2%	63.8s	43.1%	52.7s	35.1%	45.2s	27.1%	37.6s
GHOST (Patro & Kingsford, 2012)	86.1%	10.1s	74.8%	12.9s	56.1%	14.0s	39.9%	15.1s
FAQ (Vogelstein et al., 2015)	53.4%	11.3s	43.4%	11.8s	34.6%	12.1s	33.7%	12.7s
MAGNA++ (Vijayan et al., 2015)	88.3%	290.3s	77.6%	301.7s	61.9%	308.5s	45.8%	315.3s
DSPFP (Lu et al., 2016)	85.9%	4.9s	78.1%	5.1s	69.5%	5.1s	60.8%	5.1s
S-GWL (Xu et al., 2019a)	83.6%	63.1s	81.3%	41.2s	71.9%	41.2s	62.4%	41.1s
GWL(Xu et al., 2019b)	83.7%	140.8s	83.7%	113.2s	75.0%	120.3s	66.3%	127.4s
DP+ (Ding et al., 2021)	100%	0.63s	92.8%	1.75s	87.6%	1.91s	86.2%	2.43s
GRASP (Hermanns et al., 2023)	98.3%	0.4s	38.6%	**0.6s**	23.5%	**0.7s**	8.3%	**0.7s**
ASM (Shen et al., 2024)	94.0%	10s	89.8%	17.8s	82.2%	14.6s	80.9%	11.4s
LoBCD-GW (Song et al., 2026)	93.8%	0.8s	45.6%	0.8s	24.8%	0.9s	16.1%	0.9s
LiDP	**100%**	**0.13s**	**92.8%**	1.2s	**87.8%**	1.7s	**86.4%**	2.2s

6 Conclusion

In this paper, we proposed LiDP, a two-stage framework for distance-based random graph matching that improves computational efficiency and noise robustness. LiDP selectively computes pairwise matching distances, reducing evaluated entries by about 97% and achieving up to a $21\times$ speedup on Erdős-Rényi graphs while preserving exact recovery in low-noise regimes. When certification fails, it warm-starts a momentum-adaptive continuous optimization procedure with convergence guarantees, improving robustness under high noise. Experiments on bin-distance matching further demonstrate the generality of selective computation. Finally, we establish an exact-recovery guarantee for near-optimal stationary points of the nonconvex doubly stochastic graph-matching relaxation, providing a theoretical link between continuous optimization and exact matching recovery.

Future work will focus on extending the theoretical analysis of the selective-computation principle beyond degree-profile distances to broader distance-based matching methods and random graph models.

AI Use Statement. This work uses generative AI solely to improve the language and readability of the manuscript. All AI-assisted edits have been reviewed by the authors, and no AI tools were used for other research or writing tasks. The authors take full responsibility for the final content of the paper.

Ethics Statement. This work focuses on the development and analysis of algorithms for random graph matching. It does not involve human subjects, sensitive data, or applications with foreseeable ethical risks. To the best of our knowledge, this research raises no ethical concerns.

Reproducibility Statement. To ensure reproducibility, we provide the source code with our submission. All algorithmic steps, parameter settings, and experimental configurations are explicitly described in the paper. With the released code and detailed specifications, independent researchers should be able to fully reproduce our results.

REFERENCES

Boaz Barak, Chi-Ning Chou, Zhixian Lei, Tselil Schramm, and Yueqi Sheng. (Nearly) Efficient algorithms for the graph matching problem on correlated random graphs. *Advances in Neural Information Processing Systems*, 32, 2019.

Chao-Yeh Chen and Kristen Grauman. Efficient activity detection with max-subgraph search. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1274–1281. IEEE, 2012.

Sean R Collins, Patrick Kemmeren, Xue-Chu Zhao, Jack F Greenblatt, Forrest Spencer, Frank CP Holstege, Jonathan S Weissman, and Nevan J Krogan. Toward a comprehensive atlas of the physical interactome of saccharomyces cerevisiae. *Molecular & Cellular Proteomics*, 6(3):439–450, 2007.

Daniel Cullina and Negar Kiyavash. Exact alignment recovery for correlated erdös-rényi graphs. *arXiv preprint arXiv:1711.06783*, 2017.

Jian Ding and Zhangsong Li. A polynomial-time iterative algorithm for random graph matching with nonvanishing correlation. *Mathematics of Operations Research*, 2026. doi: 10.1287/moor.2024.0487. Articles in Advance.

Jian Ding, Zongming Ma, Yihong Wu, and Jiaming Xu. Efficient random graph matching via degree profiles. *Probability Theory and Related Fields*, 179(1):29–115, 2021.

Jian Ding, Yumou Fei, and Yuanzheng Wang. Efficiently matching random inhomogeneous graphs via degree profiles. *The Annals of Statistics*, 53(4):1808–1832, 2025.

Soheil Feizi, Gerald Quon, Mariana Recamonde-Mendoza, Muriel Medard, Manolis Kellis, and Ali Jadbabaie. Spectral alignment of graphs. *IEEE Transactions on Network Science and Engineering*, 7(3):1182–1197, 2019.

Marguerite Frank, Philip Wolfe, et al. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

Luca Ganassali and Laurent Massoulié. From tree matching to sparse graph alignment. In *Proceedings of the Thirty-Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pp. 1633–1665. PMLR, 2020.

Steven Gold and Anand Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(4):377–388, 1996.

Mark Heimann, Haoming Shen, Tara Safavi, and Danai Koutra. Regal: Representation learning-based graph alignment. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pp. 117–126. ACM, 2018. doi: 10.1145/3269206.3271788.

Judith Hermanns, Konstantinos Skitsas, Anton Tsitsulin, Marina Munkhoeva, Alexander Kyster, Simon Nielsen, Alexander M Bronstein, Davide Mottin, and Panagiotis Karras. GRASP: Scalable Graph Alignment by Spectral Corresponding Functions. *ACM Transactions on Knowledge Discovery from Data*, 17(4):1–26, 2023.

Marius Leordeanu, Martial Hebert, and Rahul Sukthankar. An integer projected fixed point method for graph matching and map inference. In *Advances in Neural Information Processing Systems*, pp. 1114–1122, 2009.

Yao Lu, Kaizhu Huang, and Cheng-Lin Liu. A fast projected fixed-point algorithm for large graph matching. *Pattern Recognition*, 60:971–982, 2016.

Vince Lyzinski, Donniell E Fishkind, and Carey E Priebe. Seeded graph matching for correlated erdös-rényi graphs. *J. Mach. Learn. Res.*, 15(1):3513–3540, 2014.

Cheng-Yu Ma and Chung-Shou Liao. A review of protein–protein interaction network alignment: From pathway comparison to global alignment. *Computational and Structural Biotechnology Journal*, 18:2647–2656, 2020.

Konstantin Makarychev, Rajsekar Manokaran, and Maxim Sviridenko. Maximum quadratic assignment problem: Reduction from maximum label cover and lp-based approximation algorithm. *ACM Transactions on Algorithms (TALG)*, 10(4):1–18, 2014.

Cheng Mao, Mark Rudelson, and Konstantin Tikhomirov. Exact matching of random graphs with constant correlation. *Probability Theory and Related Fields*, 186(1–2):327–389, 2023. doi: 10.1007/s00440-022-01184-3.

Cheng Mao, Yihong Wu, Jiaming Xu, and Sophie H. Yu. Random graph matching at otter's threshold via counting chandeliers. *Operations Research*, 74(1):430–445, 2026. doi: 10.1287/opre.2023.0574.

Duy MH Nguyen, Hoang Nguyen, Nghiem Diep, Tan Ngoc Pham, Tri Cao, Binh Nguyen, Paul Swoboda, Nhat Ho, Shadi Albarqouni, Pengtao Xie, et al. Lvm-med: Learning large-scale self-supervised vision models for medical imaging via second-order graph matching. *Advances in Neural Information Processing Systems*, 36, 2024.

D. Michel, I. Oikonomidis, and A. Argyros. Scale invariant and deformation tolerant partial shape matching. *Image and Vision Computing*, 29(7):459–469, 2011.

Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pp. 111–125. IEEE, 2008.

Duy M. H. Nguyen, Nghiem T. Diep, Trung Q. Nguyen, Hoang-Bao Le, Tai Nguyen, Tien Nguyen, TrungTin Nguyen, Nhat Ho, Pengtao Xie, Roger Wattenhofer, James Zhou, Daniel Sonntag, and Mathias Niepert. LoGra-Med: Long Context Multi-Graph Alignment for Medical Vision-Language Model, 2024. URL https://arxiv.org/abs/2410.02615.

Rob Patro and Carl Kingsford. Global network alignment using multiscale spectral signatures. *Bioinformatics*, 28(23):3105–3114, 2012.

Pedram Pedarsani and Matthias Grossglauser. On the privacy of anonymized networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1235–1243, 2011.

Binrui Shen, Qiang Niu, and Shengxin Zhu. Fabricated Pictures Detection with Graph Matching. In *Proceedings of the 2020 2nd Asia Pacific Information Technology Conference*, pp. 46–51, 2020.

Binrui Shen, Qiang Niu, and Shengxin Zhu. Adaptive Softassign via Hadamard-Equipped Sinkhorn. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17638–17647, 2024.

Binrui Shen, Qiang Niu, and Shengxin Zhu. Lightning graph matching. *Journal of Computational and Applied Mathematics*, 454:116189, 2025.

Binrui Shen, Qiang Niu, and Shengxin Zhu. Csgo: Constrained-softassign gradient optimization for large graph matching. *Pattern Recognition*, 177:113329, 2026. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2026.113329. URL https://www.sciencedirect.com/science/article/pii/S0031320326002943.

Rohit Singh, Jinbo Xu, and Bonnie Berger. Global alignment of multiple protein interaction networks with application to functional orthology detection. *Proceedings of the National Academy of Sciences*, 105(35):12763–12768, 2008.

Jingni Song, Jiawei Huang, Kangke Cheng, Bangxian Han, and Hu Ding. Lobcd-gw: a fast and data-dependent algorithm for computing gromov-wasserstein distance via localized block coordinate descent. In *Forty-third International Conference on Machine Learning*, 2026.

Ziying Song, Haiyue Wei, Lin Bai, Lei Yang, and Caiyan Jia. Graphalign: Enhancing accurate feature alignment by graph matching for multi-modal 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3358–3369, 2023.

Vipin Vijayan, Vikram Saraph, and Tijana Milenković. Magna++: maximizing accuracy in global network alignment via both node and edge conservation. *Bioinformatics*, 31(14):2409–2411, 2015.

Joshua T Vogelstein, John M Conroy, Vince Lyzinski, Louis J Podrazik, Steven G Kratzer, Eric T Harley, Donniell E Fishkind, R Jacob Vogelstein, and Carey E Priebe. Fast approximate quadratic programming for graph matching. *PLOS one*, 10(4):e0121002, 2015.

Hongteng Xu, Dixin Luo, and Lawrence Carin. Scalable gromov-wasserstein learning for graph partitioning and matching. In *Advances in Neural Information Processing Systems*, volume 32, 2019a.

Hongteng Xu, Dixin Luo, Hongyuan Zha, and Lawrence Carin Duke. Gromov-wasserstein learning for graph matching and node embedding. In *International conference on machine learning*, pp. 6932–6941. PMLR, 2019b.

Kun Xu, Liwei Wang, Mo Yu, Yansong Feng, Yan Song, Zhiguo Wang, and Dong Yu. Cross-lingual Knowledge Graph Alignment via Graph Matching Neural Network. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3156–3161, 2019c.

A Comparison of LiDP with Existing Methods

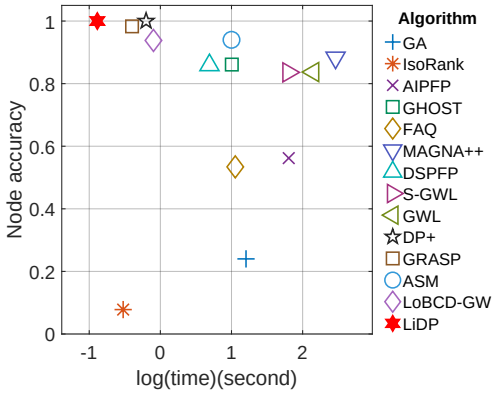


Figure 5: Matching accuracy and running time of different algorithms on protein network matching with 0% noise level.

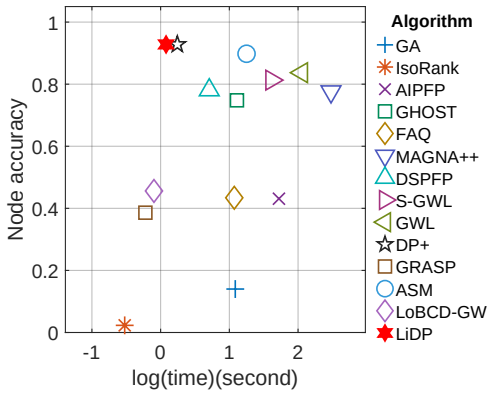


Figure 6: Mean matching accuracy and running time of different algorithms on protein network matching with 5%, 10% and 15% noise level.

B Definitions of Pairwise Distances

B.1 Degree Profile Distance

We recall the degree profile distance used for homogeneous Erdős-Rényi graph pairs. For graph A, let $N_A(i)$ and $N_A[i]$ denote the open and closed neighborhoods of vertex i, respectively, and let $a_i = |N_A(i)|$. For each neighbor $j \in N_A(i)$, define the standardized out-degree of j relative to i by

$$a_j^{(i)} = \frac{\sum_{\ell \notin N_A[i]} (A_{\ell j} - q)}{\sqrt{(n - a_i - 1)q(1-q)}}. \tag{11}$$

The empirical degree profile of vertex i in A is

$$\mu_i = \frac{1}{a_i} \sum_{j \in N_A(i)} \delta_{a_j^{(i)}}. \tag{12}$$

The corresponding profile ν_k for vertex k in graph B is defined analogously, with $b_k = |N_B(k)|$ and B in place of A. Let $\overline{\mathrm{Binom}}(m, q)$ denote the law of $(X - mq)/\sqrt{mq(1-q)}$, where $X \sim \mathrm{Binom}(m, q)$. The centered profiles are

$$\bar{\mu}_i = \mu_i - \overline{\mathrm{Binom}}(n - a_i - 1, q), \qquad \bar{\nu}_k = \nu_k - \overline{\mathrm{Binom}}(n - b_k - 1, q). \tag{13}$$

Given a partition $I_1, \ldots, I_L$ of the profile domain, the degree profile distance between vertex i in A and vertex k in B is

$$D_{ik}^{\mathrm{dp}} = \sum_{\ell=1}^{L} |\bar{\mu}_i(I_\ell) - \bar{\nu}_k(I_\ell)| := \|\bar{\mu}_i - \bar{\nu}_k\|_{1,L}. \tag{14}$$

B.2 Bin-Distance

For inhomogeneous random graphs, the bin-distance compares local neighborhood information through a balls-into-bins construction. Let $L \geq 5$ be an integer. For $t \in [0, 1)$ and $m \in \mathbb{Z}$, define

$$I_{t,m} = \left[t + m - \frac{1}{L}, t + m + \frac{1}{L} \right], \qquad I_t = \bigcup_{m \in \mathbb{Z}} I_{t,m}. \tag{15}$$

Let

$$x_j^A = \sum_{\ell=1}^{n} A_{j\ell}, \qquad x_j^B = \sum_{\ell=1}^{n} B_{j\ell} \tag{16}$$

be the degrees of vertices in A and B. For a candidate pair (i, k), the balls are the degree statistics $\sqrt{x_j^A}$ and $\sqrt{x_j^B}$, and a ball is counted in the bin family I_t only when the corresponding vertex is adjacent to i or k. Specifically, define

$$\begin{aligned} A_{ij}^{\text{bin}}(t) &= \mathbf{1}\{\sqrt{x_j^A} \in I_t\}\mathbf{1}\{A_{ij} = 1\}, \\ B_{kj}^{\text{bin}}(t) &= \mathbf{1}\{\sqrt{x_j^B} \in I_t\}\mathbf{1}\{B_{kj} = 1\}. \end{aligned} \tag{17}$$

The bin-distance between vertex i in A and vertex k in B is the L_1 distance between the resulting bin-count signatures:

$$D_{ik}^{\text{bin}} = \int_0^1 \left| \sum_{j=1}^{n} A_{ij}^{\text{bin}}(t) - \sum_{j=1}^{n} B_{kj}^{\text{bin}}(t) \right| \mathrm{d}t. \tag{18}$$

C Proof of Theorems about Selective Computation

Theorem 1. *Let D be an $n \times n$ real matrix, and assume that its n-th smallest entry is strictly smaller than its $(n+1)$-th smallest entry. Then the positions of the n smallest entries of D form a permutation matrix if and only if each of these entries is the unique minimum in its row and column.*

Proof. Let $\mathcal{S}$ be the set of positions of the n smallest entries of D. Fix any $(i, j) \in \mathcal{S}$.

Step 1: D_{ij} is the minimum in its i-th row. Suppose, for the sake of contradiction, that there exists some $j' \neq j$ such that $D_{ij'} < D_{ij}$. Since $D_{ij'}$ is strictly smaller than D_{ij} and $\mathcal{S}$ consists of the n smallest entries of D, the position (i, j') must also be included in $\mathcal{S}$. This contradicts the assumption that S contains exactly one position in each row. Hence D_{ij} must be the minimum entry in its i-th row.

Step 2: D_{ij} is the minimum in its j-th column. A symmetric argument applies to columns. If there exists $i' \neq i$ such that $D_{i'j} < D_{ij}$, then the position (i', j) would also belong to $\mathcal{S}$, contradicting the assumption that each column contains exactly one position from $\mathcal{S}$. Therefore, D_{ij} must also be the minimum entry in its j-th column.

Combining Steps 1 and 2, we conclude that for every $(i, j) \in \mathcal{S}$, the entry D_{ij} is the minimum in both its row and its column. □

Theorem 2 (Guarantee of Selective Computation Scheme)**.** *Let $D \in \mathbb{R}^{n\times n}$ be the distance matrix whose n smallest entries, indexed by $\mathcal{S}$, induce a permutation matrix. Then the set $\mathcal{S}_n$ returned by the selective computation scheme satisfies*

$$\mathcal{S}_n = \mathcal{S}.$$

Thus, selective computation yields the same matching as full computation.

Proof. The proof is divided into two parts: the first case is when our scheme finds a permutation matrix and terminates early by passing the perfect test; the second case is when the algorithm runs to completion.

Case 1: Early termination via the perfect test.

Passing the perfect test in Figure 1 implies that for

$$(i^*, j^*) = \arg \max_{(i,j)\in\mathcal{S}_n} D_{ij}, \tag{19}$$

we have

$$D_{i^*j^*} < \min\Big\{ \min_{j'\neq j^*} D_{i^*j'},\ \min_{i'\neq i^*} D_{i'j^*} \Big\}. \tag{20}$$

Since $\mathcal{S}$ in Algorithm 1 corresponds to the perfect matching, by Theorem 1 we have

$$D_{ij} < \min\Big\{\min_{j'\neq j} D_{ij'},\ \min_{i'\neq i} D_{i'j}\Big\} \ \Leftrightarrow\ (i,j)\in\mathcal{S}. \tag{21}$$

Hence, $(i^*,\ j^*)\in\mathcal{S}$. Moreover, for any other element in $\mathcal{S}_n$,

$$D_{ij} < D_{i^*j^*}, \quad \forall (i,j)\in\mathcal{S}_n. \tag{22}$$

Since

$$|\mathcal{S}| = n, \qquad \forall\,(i,j)\in\mathcal{S},\ \forall\,(a,b)\notin\mathcal{S},\ D_{ij} < D_{ab}, \tag{23}$$

we have

$$\mathcal{S}_n \setminus \{(i^*,j^*)\}\subset\mathcal{S}\ \Rightarrow\ \mathcal{S}_n = \mathcal{S}. \tag{24}$$

Case 2: Runs to completion.

Suppose that the solution P found by our scheme differs from the optimal solution P^* found by Algorithm 1 with profile distance. This implies that $\mathcal{S}_n \neq \mathcal{S}$.

Since the 'Check Perfect' step represented in Figure 2 updates any non-perfect permutation matrix into a 0-1 matrix that is not a permutation matrix, once $\mathcal{S}_n \neq \mathcal{S}$, the matrix P derived from $\mathcal{S}_n$ cannot be a permutation matrix. Hence,

$$\mathcal{R}_{\text{miss}} \cup \mathcal{C}_{\text{miss}} \neq \varnothing, \tag{25}$$

therefore,

$$D_{ij} > D_{ab},\ \forall\, i\in\mathcal{R}_{\text{miss}} \text{ or } j\in\mathcal{C}_{\text{miss}},\ (a,b)\in\mathcal{S}_n. \tag{26}$$

From the definitions of $\mathcal{S}$ and $\mathcal{S}_n$, we have

$$\max_{(i,j)\in\mathcal{S}_n} D_{ij} \geq \max_{(k,l)\in\mathcal{S}} D_{kl}. \tag{27}$$

Therefore,

$$(i,j)\notin\mathcal{S}, \quad \forall\, i\in\mathcal{R}_{\text{miss}} \text{ or } j\in\mathcal{C}_{\text{miss}}. \tag{28}$$

This implies that $\mathcal{S}$ cannot form a permutation matrix, which contradicts the assumption. Hence,

$$\mathcal{S}_n = \mathcal{S}. \tag{29}$$

□

D Proof of Theorems about Continuous Optimization

Theorem 3 (Momentum-Adaptive continuous optimization convergence property)**.** *Let $A, B\in\mathbb{R}^{n\times n}$ be two symmetric adjacency matrices. For arbitrary initialization D, the sequence $\{P^t\}$ generated by Algorithm 2 converges to a first-order stationary point of $\mathcal{Z}$ over $\mathcal{B}_n$.*

Proof. Let $\Delta P^t = \tilde{P}^t - P^t$. Substituting the update rule into the objective function $\mathcal{Z}(P)$, and we expand the term with respect to α

$$\begin{aligned}\mathcal{Z}(P^t+\alpha\Delta P^t) =& \text{tr}\left[(P^t+\alpha\Delta P^t)^\top A(P^t+\alpha\Delta P^t)B\right]\\ =& \text{tr}(P^{t\top}AP^tB) + 2\alpha\text{tr}(\Delta P^{t\top}AP^tB) +\\ & \alpha^2\text{tr}(\Delta P^{t\top}A\Delta P^tB)\\ =& c + b\alpha + a\alpha^2,\end{aligned} \tag{30}$$

where the coefficients are identified as

$$a = \text{tr}(\Delta P^{t\top}A\Delta P^tB), \quad b = 2\text{tr}(\Delta P^{t\top}AP^tB). \tag{31}$$

Since $\tilde{P}^t = \arg\max_P\langle P, \nabla\mathcal{Z}(P^t)\rangle$, we have

$$\langle\tilde{P}^t, \nabla\mathcal{Z}(P^t)\rangle \geq \langle P^t, \nabla\mathcal{Z}(P^t)\rangle, \tag{32}$$

which explicitly implies $b\geq 0$. Therefore, the optimal α^t is determined by

$$\alpha^t = \begin{cases}1, & \text{if } a\geq 0,\\ \min\left(1, -\frac{b}{2a}\right), & \text{if } a<0.\end{cases} \tag{33}$$

By invoking Property 1 and 2 in (Shen et al., 2026) and the compactness of $\mathcal{B}_n$, we establish that any accumulation point of the sequence $\{P^t\}$ converges to a first-order stationary point. □

Theorem 4 (Exact Recovery from First-Order Stationarity). *Let $A, B \in [0,1]^{n\times n}$ be symmetric with zero diagonal. For convenience, set the ground-truth correspondence as I. Suppose that, for some $c_n > 0$,*

$$\mathcal{Z}(I) - \mathrm{Tr}(APBQ^\top) \geq c_n\big(2n - \mathrm{Tr}(P) - \mathrm{Tr}(Q)\big), \qquad \forall P, Q \in \Pi_{n\times n}. \tag{34}$$

If a first-order stationary point $\widehat{X}$ of $\mathcal{Z}$ over $\mathcal{B}_n$ satisfies

$$\mathcal{Z}(I) - \mathcal{Z}(\widehat{X}) < \frac{c_n^2}{2}, \tag{35}$$

then $\widehat{X} = I$.

Proof. For $X \in \mathcal{B}_n$, write $t(X) := n - \mathrm{Tr}(X)$. By the Birkhoff–von Neumann theorem, $X = \sum_P \lambda_P P$, where $\lambda_P \geq 0$ and $\sum_P \lambda_P = 1$. Bilinearity and (34) give

$$\begin{aligned}\mathcal{Z}(I) - \mathcal{Z}(X) &= \sum_{P,Q} \lambda_P\lambda_Q\big[\mathcal{Z}(I) - \mathrm{Tr}(APBQ^\top)\big] \\ &\geq c_n \sum_{P,Q} \lambda_P\lambda_Q\big(2n - \mathrm{Tr}(P) - \mathrm{Tr}(Q)\big) \\ &= 2c_n t(X).\end{aligned} \tag{36}$$

Now let $\widehat{X}$ be a first-order stationary point of $\mathcal{Z}$ over $\mathcal{B}_n$, and set $\Delta := \widehat{X} - I$ and $t := t(\widehat{X})$. Since $I \in \mathcal{B}_n$ and $\nabla\mathcal{Z}(\widehat{X}) = 2A\widehat{X}B$, stationarity gives

$$0 \leq \langle A\widehat{X}B, \Delta\rangle = \langle AB, \Delta\rangle + \mathrm{Tr}(\Delta^\top A\Delta B). \tag{37}$$

Applying (34) with $Q = I$ and averaging over a Birkhoff decomposition of $\widehat{X}$ yields $\langle AB, \Delta\rangle \leq -c_n t$. Moreover, $\sum_{i,j} |\Delta_{ij}| = 2t$, so the entrywise bounds on A and B imply

$$\big|\mathrm{Tr}(\Delta^\top A\Delta B)\big| \leq \left(\sum_{i,j} |\Delta_{ij}|\right)^2 = 4t^2. \tag{38}$$

Consequently, $0 \leq -c_n t + 4t^2$. If $\widehat{X} \neq I$, then $t > 0$ and hence $t \geq c_n/4$. The global growth bound therefore gives

$$\mathcal{Z}(I) - \mathcal{Z}(\widehat{X}) \geq 2c_n t \geq \frac{c_n^2}{2}. \tag{39}$$

Thus $\mathcal{Z}(I) - \mathcal{Z}(\widehat{X}) < c_n^2/2$ forces $\widehat{X} = I$, proving the final claim. □

Next we specialize Theorem 4 to the correlated Erdős-Rényi graphs.

Corollary 1 (Exact Recovery from First-Order Stationarity for Correlated Erdős-Rényi Graph). *Let $A, B \in \{0,1\}^{n\times n}$ be the adjacency matrices generated from the correlated Erdős-Rényi model in Definition 1, with the ground-truth correspondence relabeled as I, and let $\gamma := q(s-q) > 0$. Suppose that*

$$\mathcal{Z}(I) - \mathrm{Tr}(APBQ^\top) \geq \frac{\gamma n}{8}\big(2n - \mathrm{Tr}(P) - \mathrm{Tr}(Q)\big), \qquad \forall P, Q \in \Pi_{n\times n}. \tag{40}$$

If a first-order stationary point $\widehat{X}$ of $\mathcal{Z}$ over $\mathcal{B}_n$ satisfies

$$\mathcal{Z}(I) - \mathcal{Z}(\widehat{X}) < \frac{\gamma^2 n^2}{128} \tag{41}$$

then $\widehat{X} = I$.

Proof. The matrices A and B generated by Defnition 1 are symmetric with zero diagonal and entries in $[0,1]$. Together with (40) verifies the assumptions of Theorem 4 with $c_n = \gamma n/8$. Therefore, the global growth bound and uniqueness follow directly. The stationary-point exactness condition follows from $c_n^2/2 = \gamma^2 n^2/128$. □

E Other Experiments

E.1 Sensitivity to Graph Density

We further evaluate the effect of graph density on the computational efficiency of LiDP in the perfect matching regime. Specifically, we vary the edge existence probability of the parent Erdős-Rényi graph of $n = 1000$ while keeping the noise level fixed at $\delta = 0$.

Table 4 reports the runtime comparison between DP and LiDP under different graph densities. As the edge probability increases from 0.05 to 0.5, the runtime of DP increases substantially due to the higher cost of degree-profile computations. In contrast, LiDP exhibits only marginal runtime growth across different density regimes. Consequently, the speedup ratio improves consistently with graph density, reaching over $35\times$ for dense graphs.

This trend indicates that denser graphs make the full degree-profile computation in DP increasingly expensive, while LiDP benefits more from degree-difference pruning.

Table 4: Runtime comparison between DP and LiDP on correlated Erdős-Rényi graphs with different densities under the noise level $\delta = 0$.

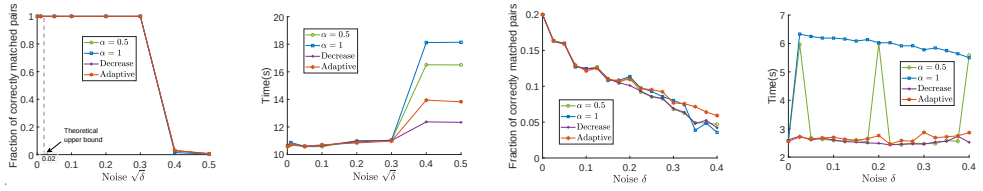
Density	**DP**	**LiDP**	**Speedup Ratio**
0.05	0.53s	0.05s	$10.6\times$
0.1	1.02s	0.05s	$20.4\times$
0.5	2.13s	0.06s	$35.5\times$

E.2 Evaluation on Bin-Distance Matching

To evaluate whether LiDP can serve as a generic selective-computation framework, we apply the same strategy to bin-distance matching on inhomogeneous graphs. On correlated Erdős-Rényi graphs with $n = 1000$, Table 5 compares the original bin(0.5)-distance matching method (Ding et al., 2025) with our selective computation scheme, showing over $20\times$ speedup under both noise levels while finding perfect matching. Figure 7 further shows that, with bin-distance initialization, our momentum-adaptive continuous optimization improves accuracy with comparable runtime on both Erdős-Rényi graphs and the Slashdot networks.

Table 5: Performance comparison under different noise levels on Erdős-Rényi graphs with bin-distance ($n = 1000$), 'lite' represents our selective computation scheme.

Noise ($\sqrt{\delta}$)	**bin(0.5)**	**bin(0.5) lite**	**Speedup Ratio**
0.1%	12.79s	0.52s	$24.6\times$
1.0%	11.94s	0.55s	$21.7\times$



(a) ER Acc. (b) ER time (c) Slashdot Acc. (d) Slashdot time

Figure 7: Accuracy and runtime comparison between fixed and momentum-adaptive step-size strategies with bin-distance initialization on Erdős-Rényi graphs and the Slashdot network.