



**HAL**  
open science

# Graph Matching via convex relaxation to the simplex

Ernesto Araya Valdivia, Hemant Tyagi

► **To cite this version:**

Ernesto Araya Valdivia, Hemant Tyagi. Graph Matching via convex relaxation to the simplex. 2024.  
hal-04394319

**HAL Id: hal-04394319**

**<https://hal.science/hal-04394319>**

Preprint submitted on 15 Jan 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Graph Matching via convex relaxation to the simplex

Ernesto Araya\*  
 ernesto-javier.araya-valdivia@inria.fr

Hemant Tyagi\*  
 hemant.tyagi@inria.fr

November 1, 2023

## Abstract

This paper addresses the Graph Matching problem, which consists of finding the best possible alignment between two input graphs, and has many applications in computer vision, network deanonymization and protein alignment. A common approach to tackle this problem is through convex relaxations of the NP-hard *Quadratic Assignment Problem* (QAP).

Here, we introduce a new convex relaxation onto the unit simplex and develop an efficient mirror descent scheme with closed-form iterations for solving this problem. Under the correlated Gaussian Wigner model, we show that the simplex relaxation admits a unique solution with high probability. In the noiseless case, this is shown to imply exact recovery of the ground truth permutation. Additionally, we establish a novel sufficiency condition for the input matrix in standard greedy rounding methods, which is less restrictive than the commonly used ‘diagonal dominance’ condition. We use this condition to show exact one-step recovery of the ground truth (holding almost surely) via the mirror descent scheme, in the noiseless setting. We also use this condition to obtain significantly improved conditions for the GRAMPA algorithm [26] in the noiseless setting. Our method is evaluated on both synthetic and real data, demonstrating superior statistical performance compared to existing convex relaxation methods with similar computational costs.

## 1 Introduction

In the context of the Graph Matching (GM) problem, also referred to as *network alignment*, we consider two input undirected graphs,  $G = ([n], \mathcal{E})$  and  $G' = ([n], \mathcal{E}')$ , which are assumed to share the same set of vertices, denoted as  $[n] = \{1, \dots, n\}$ . The objective is to find a bijective map, represented by a permutation  $x : [n] \rightarrow [n]$ , between the vertices of  $G$  and  $G'$ , which is intended to align their edges as effectively as possible. The alignment is considered optimal when the largest possible number of edges in  $G$  and  $G'$  satisfy the relation  $\{i, j\} \in \mathcal{E} \iff \{x(i), x(j)\} \in \mathcal{E}'$ . This problem has numerous applications, such as computer vision [33], network de-anonymization [51], pattern recognition [18, 24], protein-protein interactions and computational biology [67, 60], to name a few. For instance, it has been used in computer vision to track a moving object by identifying the correspondence between the different portions in two different frames.

Mathematically, the Graph Matching problem can be formulated as an instance of the NP-hard *quadratic assignment problem*, as follows

$$\max_{x \in \mathcal{S}_n} \sum_{i,j=1}^n A_{ij} B_{x(i)x(j)} \equiv \max_{X \in \mathcal{P}_n} \langle AX, XB \rangle_F, \quad (1.1)$$

---

\*Inria, Univ. Lille, CNRS, UMR 8524 - Laboratoire Paul Painlevé, F-59000

where  $A, B \in \mathbb{R}^{n \times n}$  are the (possibly weighted) adjacency matrices of the pairs of input graphs to be matched and  $\mathcal{P}_n$  (resp.  $\mathcal{S}_n$ ) is the set of  $n \times n$  permutation matrices (resp. the set of permutation maps). In words, a solution of (1.1) will maximize the total edge alignment between  $A$  and  $B$ . For our purposes, it will be more useful to consider the following equivalent formulation,

$$\min_{X \in \mathcal{P}_n} \|AX - XB\|_F^2, \quad (1.2)$$

which has the advantage of possessing a convex objective function that we define by

$$E(X) := \|AX - XB\|_F^2, \quad (1.3)$$

where the input graphs  $A$  and  $B$  will be clear from the context. It will be useful to consider the vector form of (1.2), and for that we write with a slight abuse of notation<sup>1</sup>

$$E(\text{vec}(X)) = \text{vec}(X)^T H \text{vec}(X), \quad (1.4)$$

where  $H := (\text{Id} \otimes A - B \otimes \text{Id})^2$  and  $\text{vec}(X)$  represents the vector form of  $X$  obtained by stacking its columns (notations are described in Section 1.2). Despite the convexity of  $E(\cdot)$  the problem (1.2) is NP-Hard in the worst case [17], which can be mainly attributed to its combinatorial constraints, and even finding a constant factor approximation remains a hard problem [59]. On the other hand, notice that when  $B = X^{*T}AX^*$  for some  $X^* \in \mathcal{P}_n$ , (1.2) is equivalent to the well-known Graph Isomorphism Problem [9].

**A general algorithmic strategy** In spite of its worst-case complexity, several efficient algorithms have been proven to solve the problem (1.5) within a specific class of instances. Most of the time, these instances are sampled by a statistical model, which generates matrices  $A, B$  according to a ground truth permutation  $X^* \in \mathcal{P}_n$  (we will formally introduce statistical models in Section 2.3). The meta-strategy used by most graph matching algorithms consists of two steps that can be summarized as follows.

1. **Similarity finding stage.** We construct a data-dependent similarity matrix  $\hat{X}(A, B) \in \mathbb{R}^{n \times n}$ , where we expect that  $\hat{X}_{ij}(A, B)$  captures the likelihood of  $X_{ij}^* = 1$ , where  $X^*$  is the ground truth permutation.
2. **Rounding stage.** The similarity matrix  $\hat{X}$  is then projected onto the set of permutations, which, when minimizing the Frobenius norm, is equivalent to solving a *maximum linear assignment problem*. Since analyzing linear assignment can be quite challenging, many algorithms opt for using a greedy rounding strategy. This approach is simpler to analyze, computationally more efficient, and often exhibits comparable error rates to linear assignment in experiments.

**Convex relaxations of graph matching.** One common approach for the similarity finding stage consists of defining  $\hat{X}(A, B)$  as the solution of a relaxation of (1.2), where the set of permutation is relaxed to a continuous constraint set. Of special importance is the family of convex relaxations which can be expressed in general as

$$\min_{X \in \mathcal{K}} \|AX - XB\|_F^2. \quad (1.5)$$

---

<sup>1</sup>Allowing  $E$  to take vectors in  $\mathbb{R}^{n^2}$  as arguments, using the canonical matrix-vector identification.

where  $\mathcal{K} \subseteq \mathbb{R}^{n \times n}$  is a convex set such that  $\mathcal{P}_n \subseteq \mathcal{K}$ . Some of the choices for the set  $\mathcal{K}$  that have been investigated in the literature include the hyperplane defined by  $\mathbf{1}^T X \mathbf{1} = n$  [26] and the Birkhoff polytope of doubly stochastic matrices [1]. The latter can be considered the gold standard, in terms of tightness of convex relaxations, but its analysis has proven to be challenging, and obtaining theoretical guarantees remains largely open.

**Graph matching on the simplex and mirror descent.** We consider the set of matrices

$$\Delta_{n^2} := \{X \in \mathbb{R}^{n \times n} : \mathbf{1}^T X \mathbf{1} = 1, X_{ij} \geq 0, \forall i, j \in [n]\},$$

which can be identified with the usual unit simplex in  $\mathbb{R}^{n^2}$ , in the sense that  $X \in \Delta_{n^2}$  if and only if  $\text{vec}(X)$  belongs to the unit simplex (we will refer to  $\Delta_{n^2}$  as the unit simplex in the sequel). By choosing  $\mathcal{K} = \Delta_{n^2}$  in (1.5), we arrive to the following convex relaxation

$$\min_{X \in \Delta_{n^2}} \|AX - XB\|_F^2. \tag{1.6}$$

This formulation is referred to as graph matching on the simplex. Despite (1.6) representing a natural problem and being an advancement over other convex relaxations (e.g., it is tighter than the  $\mathbf{1}^T X \mathbf{1} = n$  constraint in [26]), it has been largely overlooked in the literature. We argue that (1.6) presents at least two advantages over existing work.

- **Regularization through positivity.** In the works [26, 25], the hyperplane constraint  $\mathcal{K} = \{X \in \mathbb{R}^{n \times n} : \mathbf{1}^T X \mathbf{1} = n\}$  was considered and an additional regularizing term of the form  $\eta \|X\|_F^2$  was added to the objective  $E(\cdot)$ . By directly incorporating the positivity constraints, we achieve a similar regularization effect without the need for a cumbersome explicit regularizer. This approach helps us avoid the task of parameter selection for such a parameter in practice. We show in experiments that it outperforms [26, 25].
- **Efficient algorithms.** we can solve (1.6) using iterative first-order methods such as *projected gradient descent* (PGD) and *entropic mirror descent* (EMD). Although the existence of efficient projection algorithms onto  $\Delta_{n^2}$  can make PGD within the simplex more expedited compared to PGD within the Birkhoff polytope, employing the EMD algorithm allows us to completely bypass this projection step, as it possesses an explicit update formula in the form of *multiplicative weights updates* [16, 12].

**Contributions.** The primary contributions of this paper are outlined below. To our knowledge, these findings mark the first results for both convex graph matching on the simplex and the mirror descent (MD) algorithm for graph matching.

- We propose an efficient iterative algorithm based on the mirror descent to solve the simplex relaxation of graph matching, which improves in  $\mathcal{O}(\log n)$  in running time with respect to the more common projected gradient descent method.
- We introduce a novel sufficient condition for achieving exact recovery, which is notably less stringent than the commonly used diagonal dominance property. We employ this condition to establish exact one-step recovery (i.e., in one iteration) of the ground truth permutation via EMD, in the noiseless case of the Correlated Gaussian Wigner model. This result holds almost surely, and for any positive step size in EMD. Furthermore, we use this condition to improve a pivotal lemma from [26], yielding stronger results with a simpler proof.

- In the noiseless case of the Correlated Gaussian Wigner model, we prove that (1.6) has a unique solution with high probability, with the solution corresponding to a scaled version of the ground truth permutation. This, along with [11, Theorems 9.16 and 9.18], ensures that EMD and PGD converge to a unique point (under certain conditions on step size), which in the noiseless setting leads to exact recovery of the ground truth permutation.
- Through extensive experiments, we demonstrate that our proposed algorithm performs well on synthetic data, for several correlated random graph models (described in Section 2.3). In addition, we show the applicability of our algorithm to real-world datasets where it is seen to outperform other methods either in speed or accuracy.

## 1.1 Related work

**Convex relaxations.** Although many algorithms based on solving a problem of the form (1.5) have been proposed [29, 27, 46, 68], some with very good experimental performance, there is still a lack of theoretical understanding regarding their statistical guarantees. In the case where  $\mathcal{K}$  is the set of doubly stochastic matrices, some early theoretical work, which applies under restrictive conditions on the inputs graphs and mostly in the deterministic setting, can be found in [1]. A spectral estimator dubbed GRAMPA, which is the solution of a regularized version of (1.5), taking  $\mathcal{K}$  to be the hyperplane  $\mathbf{1}^T X \mathbf{1} = n$ , was studied in [26]. They prove that, after greedy rounding, their estimator perfectly recovers the ground truth permutation when the graphs are generated with a correlated Gaussian Wigner model, in the regime where the correlation is  $\sqrt{1 - O\left(\frac{1}{\log^2 n}\right)}$ . There is still a big gap between these guarantees and the sharp information theoretic threshold for recovery, as discusses below.

**Information-theoretic limits of graph matching.** The necessary and sufficient conditions for correctly estimating the matching between two graphs when they are generated at random, from either the correlated Gaussian Wigner or the correlated Erdős-Rényi models (see Section 2.3), have been investigated by several authors in [19, 34, 65, 31, 28, 49, 32]. For the correlated Gaussian Wigner model, it has been shown in [65, Thm.1] that the ground truth permutation  $x^*$  can be exactly recovered w.h.p. only when  $\sigma^2 \leq 1 - \frac{(4+\epsilon)\log n}{n}$ . When  $\sigma^2 \geq 1 - \frac{(4-\epsilon)\log n}{n}$  no algorithm can even partially recover  $x^*$ . One natural question pertains to the possibility of closing the gap between the performance bound for GRAMPA as outlined in [26] and the information-theoretic threshold by utilizing convex relaxations.

**MD algorithm and Multiplicative Weights Updates.** The MD algorithm, introduced in [52], has been extensively studied in convex optimization [12, 16], learning theory, and online learning [35]. On the other hand, the EMD algorithm, also known as Exponentiated Gradient Descent (EGD) [40] or exponential Multiplicative Weights Updates (MWU) [7], has found diverse applications spanning machine learning [13], optimization [8, 6, 3], computer science [7, 2], and game theory [21], among others. Although prior work, such as [22] and [38], has explored MWU strategies for graph matching, there are notable distinctions from our research. Firstly, they consider different objective functions and constraint sets, yielding fundamentally different update strategies. Secondly, their approach lacks a principled assessment of robustness against noise due to the absence of statistical generative models for inputs  $A$  and  $B$ . It is worth noting that the same exponential MWU technique we explore here was previously considered in the context of the nonconvex formulation of Graph Isomorphism in [56]. However, it was primarily proposed as a heuristic inspired by

the replicator equations from theoretical biology, and no connection with the MD algorithm was established in that work.

**Other algorithms for seedless graph matching.** Although less related to our work, it is worth mentioning that there exist several approaches for seedless graph matching not directly based on convex relaxations. When  $\mathcal{K}$  is the set of orthogonal matrices in (1.5), its solution can be expressed in closed form in terms of the spectrum of the graphs, as was proven in the celebrated work [63]. In [48], the authors prove that a two-step algorithm can attain exact recovery in the context of the sparse Erdős-Rényi graphs, even in the constant correlation regime. For sparse Erdős-Rényi graphs, other approaches based on counting combinatorial structures include [50, 30]. Exploring the possibility of extending those approaches to the dense case remains an open question.

## 1.2 Notation.

We denote  $\mathcal{P}_n$  (resp.  $\mathcal{S}_n$ ) the set of permutation matrices (resp. permutation maps) of size  $n$ . We define the vectorization operator  $\text{vec} : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n^2 \times 1}$ , which, for a matrix  $M$ ,  $\text{vec}(M)$  gives the corresponding vector formed by stacking the columns of  $M$ . For any given matrix  $M$ , we define the norm  $\|M\|_{1,1} := \sum_{i,j} |M_{ij}|$  and the usual Frobenius norm  $\|M\|_F := (\sum_{i,j} |M_{ij}|^2)^{\frac{1}{2}} = \sqrt{\langle M, M \rangle_F}$ , where for matrices  $M, M'$ ,  $\langle M, M' \rangle_F := \langle \text{vec}(M), \text{vec}(M') \rangle$  (here  $\langle \cdot, \cdot \rangle$  is the usual inner product in  $\mathbb{R}^{n^2 \times 1}$ ). Similarly, for  $M, M' \in \mathbb{R}^{n \times n}$ , we define  $M \odot M'$  to be their Hadamard (entrywise) product and  $M \otimes M'$  their usual Kronecker product. For a matrix  $M$  we define its Hadamard (entrywise) exponential  $\exp_{\odot}(M)$  to be matrix with  $(i, j)$ -th entry  $\exp(M_{ij})$ . We use  $\mathbf{1}$  (resp.  $J$ ) to denote the vector (resp. matrix) of all ones in  $\mathbb{R}^n$  (resp.  $\mathbb{R}^{n \times n}$ ), where the size  $n$  will be clear from context. We define  $\Delta_n := \{x \in \mathbb{R}^n : \mathbf{1}^T x = 1\}$  and we identify  $\Delta_{n^2}$  with the set of matrices  $M \in \mathbb{R}^{n \times n}$  such that  $\text{vec}(M) \in \Delta_{n^2}$ . We denote  $\text{Id}$  the identity matrix, where the dimension will be clear from context. We will denote  $e_1, \dots, e_n$  the elements of the canonical basis of  $\mathbb{R}^n$ .

## 2 Algorithms

### 2.1 A greedy rounding procedure and a sufficiency lemma

We consider the following greedy rounding algorithm, named Greedy Maximum Weighted Matching (GMWM) algorithm in [45, 66], which will serve us to accomplish Step 2 in the meta-strategy described earlier. The specific version presented below, as Algorithm 1, has been taken from [5]. To simplify our analysis, we will consider the notion of *algorithmic equivariance*, which, in the

---

#### Algorithm 1 GMWM (Greedy maximum weight matching)

---

**Require:** A cost matrix  $C \in \mathbb{R}^{n \times n}$ .

**Ensure:** A permutation matrix  $X \in \mathcal{P}_n$ .

- 1: Select  $(i_1, j_1)$  such that  $C_{i_1, j_1}$  is the largest entry in  $C$  (break ties arbitrarily). Define  $C^{(1)} \in \mathbb{R}^{n \times n}$ :  $C_{ij}^{(1)} = C_{ij} \mathbf{1}_{i \neq i_1, j \neq j_1} - \infty \cdot \mathbf{1}_{i=i_1 \text{ or } j=j_1}$ .
  - 2: **for**  $k = 2$  to  $N$  **do**
  - 3:   Select  $(i_k, j_k)$  such that  $C_{i_k, j_k}^{(k-1)}$  is the largest entry in  $C^{(k-1)}$  (break ties arbitrarily).
  - 4:   Define  $C^{(k)} \in \mathbb{R}^{n \times n}$ :  $C_{ij}^{(k)} = C_{ij}^{(k-1)} \mathbf{1}_{i \neq i_k, j \neq j_k} - \infty \cdot \mathbf{1}_{i=i_k \text{ or } j=j_k}$ .
  - 5: **end for**
  - 6: Define  $X \in \{0, 1\}^{n \times n}$ :  $X_{ij} = \sum_{k=1}^N \mathbf{1}_{i=i_k, j=j_k}$ .
  - 7: **return**  $X$
-

context of graph matching, means that if an algorithm outputs an estimator  $\hat{X}_{\mathcal{P}}$  for inputs  $A$  and  $B$ , then the same algorithm outputs  $\hat{X}_{\mathcal{P}}\Pi^T$  when the inputs are  $A$  and  $\Pi B\Pi^T$  instead, for any  $\Pi \in \mathcal{P}_n$ . The following lemma tells us that any algorithm that, first, obtains  $\hat{X}$  by solving (1.5) for a permutation-invariant set  $\mathcal{K}$  and, second, rounds  $\hat{X}$  using Algorithm 1, is equivariant. We include its proof in Appendix B for completeness.

**Lemma 1.** *Let  $\mathcal{K}$  be a convex subset of  $\mathbb{R}^{n \times n}$  such that  $\mathcal{K}\Pi^T = \mathcal{K}$ , for any  $\Pi \in \mathcal{P}_n$ , and for  $A, B \in \mathbb{R}^{n \times n}$  define  $\mathcal{S}(A, B)$  to be the set of solutions of (1.5). Assume that the set defined by*

$$\{Y \in \mathbb{R}^{n \times n} : Y = \text{GMWM}(\hat{X}) \text{ for } \hat{X} \in \mathcal{S}(A, B)\},$$

*is a singleton. Then the algorithm that outputs  $\hat{X}_{\mathcal{P}} = \text{GMWM}(\hat{X})$  for any  $\hat{X} \in \mathcal{S}(A, B)$  is equivariant.*

The previous lemma implies that we can consider, without loss of generality, the ground truth to be  $X^* = \text{Id}$  (see [26, Section 1.2] for additional discussion). This motivates the following deterministic lemma.

**Lemma 2.** *Let  $C \in \mathbb{R}^{n \times n}$  be any matrix. Then the following statements hold.*

(i) *If  $C$  satisfies*

$$C_{ii} \vee C_{jj} > C_{ij} \vee C_{ji}, \quad \forall i \neq j \in [n], \quad (2.1)$$

*then  $\text{GMWM}(C) = \text{Id}$ .*

(ii) *If  $C$  is symmetric and*

$$C_{ii} + C_{jj} > 2C_{ij}, \quad \forall i \neq j \in [n], \quad (2.2)$$

*then  $\text{GMWM}(C) = \text{Id}$ .*

(iii) *If  $C$  is symmetric and positive definite, we have  $\text{GMWM}(C) = \text{Id}$ .*

*In (i) and (ii), if for each  $i \in [n]$  the set  $\arg \max_j C_{ij}$  is a singleton, then the sufficient conditions (2.1) and (2.2) are also necessary.*

*Proof.* Notice that in the first step of GMWM we select the indices  $(i_1, j_1)$  corresponding to the largest entry of  $C$ . Those indices satisfy  $i_1 = j_1$ , otherwise this would contradict (2.1). Then it suffices to notice that in the subsequent steps the matrix  $C^{(k)}$  still satisfy (2.1) which implies that  $i_k = j_k$  for all  $k \in [n]$ . This proves (i). When  $C$  is symmetric, the property (2.2) implies (2.1). This proves (ii). On the other hand, if  $C$  is positive definite, then for all  $i \neq j \in [n]$  we have  $(e_i - e_j)^T C (e_i - e_j) > 0$ , where  $e_k$  is the  $k$ -th element of the canonical basis of  $\mathbb{R}^n$ . From this, we deduce that  $C_{ii} + C_{jj} > 2C_{ij}$ , which directly implies (2.1), proving (iii). To prove the converse, in the case (i), notice that if  $\text{GMWM}(C) = \text{Id}$  and for each  $i \in [n]$  the set  $\arg \max_j C_{ij}$  is a singleton, then there are no ties in lines 1 and 3 of Algorithm 1, and it is clear that the maximum for each matrix  $C^{(k)}$  is attained by some element in its diagonal. This implies (2.1). The proof for the converse of (ii) is analogous.  $\square$

In terms of the general algorithmic strategy sketched in Section 1, the main use of Lemma 2 will be the following. If we can prove that  $\hat{X}$ , a solution of the convex relaxation (1.5), satisfies (2.1) with certain probability, then Lemma 2 implies that  $\hat{X}_{\mathcal{P}} := \text{GMWM}(\hat{X})$  is equal to  $X^* = \text{Id}$  (that is, we obtain exact recovery), with the same probability. An interesting point is to understand if the properties (2.1) and (2.2) are preserved under typical matrix operations such as additions, scaling, etc. In this direction, we have the following result, whose proof is direct.

**Lemma 3.** Let  $C \in \mathbb{R}^{n \times n}$  be a matrix satisfying (2.1), then for any  $t > 0$ ,  $\alpha \in \mathbb{R}$ , the matrices  $tC$  and  $C + \alpha J$  also satisfy (2.1). Let  $C', C'' \in \mathbb{R}^{n \times n}$  be any two symmetric matrices satisfying (2.2), then  $tC', C + \alpha J$  and  $C' + C''$  satisfy (2.2).

In words, the previous lemma states that property (2.1) is closed under scaling by a positive factor and a global additive translation, while (2.2) is also closed under addition.

## 2.2 Mirror descent algorithm for graph matching

Our proposed method consists of two main steps: first, solving (1.6), which addresses the graph matching problem on the unit simplex and yields  $\widehat{X}$ ; second, rounding this solution using Algorithm 1 to generate  $\widehat{X}_{\mathcal{P}}$  an estimate of the ground truth  $X^*$ . To efficiently solve (1.6), we propose the utilization of the mirror descent (MD) method. This first-order iterative optimization algorithm generalizes projected gradient descent by allowing for non-Euclidean<sup>2</sup> distances [52]. To avoid unnecessary generality, we will focus on two primary examples within the MD family on the unit simplex: Projected Gradient descent (PGD) and Entropic Mirror Descent (EMD). As mentioned earlier, EMD has also received the name of Exponentiated Gradient Descent (EGD) [40] and can be seen as an example of the Multiplicative Weights Update (MWU) [7] strategy.

**EMD and PGD updates.** Both methods receive an initial point  $X^{(0)}$  as input but differ in their approach to performing updates. More specifically, the update process for these methods, in the case of problem (1.6), is as follows,

$$X^{(k+1)} = \arg \min_{X \in \Delta_{n^2}} \sum_{i,j=1}^n \left[ (\gamma_k \nabla E(X^{(k)})_{ij} - X_{ij}^{(k)}) X_{ij} + X_{ij}^2 \right], \quad (\text{PGD update})$$

$$X^{(k+1)} = \arg \min_{X \in \Delta_{n^2}} \sum_{i,j=1}^n \left[ (\gamma_k \nabla E(X^{(k)})_{ij} - 1 - \log X_{ij}^{(k)}) X_{ij} + X_{ij} \log X_{ij} \right], \quad (\text{EMD update})$$

where  $(\gamma_k)_{k \geq 0}$  is a sequence of positive learning rates (also called the step-sizes). The derivation of these updates is well-known (refer to [11, Section 9]) and, without entering into details, stems from a broader form of MD, which permits the selection of different (Bregman) distances to project onto the set of constraints. In (PGD update), the Euclidean distance is used as the Bregman distance, and it is easy to see that (PGD update) is equivalent to

$$X^{(k+1)} = \text{proj}_{\Delta_{n^2}} \left( X^{(k)} - \gamma_k \nabla E(X^{(k)}) \right), \quad (2.3)$$

where  $\text{proj}_{\Delta_{n^2}}$  corresponds to the Euclidean (Frobenius norm) projection onto  $\Delta_{n^2}$  and  $\nabla E(X) = A^2 X + X B^2 - 2 A X B$ . In the case of (EMD update), the Bregman distance employed is the Kullback-Liebler divergence, which gives rise to the term ‘entropic mirror descent’. Perhaps more importantly, from (EMD update) one can easily derive (as shown in [11, Example 3.17]) the following simplified form (which corresponds to a MWU rule)

$$X_{ij}^{(k+1)} = X_{ij}^{(k)} \exp \left( -\gamma_k \nabla E(X^{(k)})_{ij} + c \right), \quad (2.4)$$

where  $c$  is a constant which ensures that  $X^{(k+1)} \in \Delta_{n^2}$ . Our proposed algorithm, written in matrix language, is summarized in Algorithm 2.

<sup>2</sup>Here a non-Euclidean metric means simply a metric different from  $\|\cdot\|_2 = \sqrt{\langle \cdot, \cdot \rangle}$ .



**Initialization.** We chose to always start with  $X^{(0)} = J/n^2$ , the barycenter of the unit simplex, which can be seen as an agnostic initialization, given that the Frobenius distance from  $X^{(0)}$  to any permutation is the same. Additionally, this choice satisfies the known boundedness hypothesis for Kullback-Leibler divergence, with respect to the initial point, that guarantees  $\mathcal{O}\left(\frac{1}{\sqrt{N}}\right)$  convergence to the optimal solution in (1.6) (see [11, Theorem 9.16] and [11, Example 9.17]).

---

**Algorithm 2** Entropic mirror descent for Graph Matching (EMDGM)

---

**Require:**  $A, B \in \mathbb{R}^{n \times n}$ ,  $N \in \mathbb{N}$  and sequence  $\{\gamma_k\}_{k=0}^N \in \mathbb{R}_+$ .

**Ensure:** A permutation  $\widehat{X}_{\mathcal{P}} \in \mathcal{P}_n$ .

- 1: Define the initial point  $X^{(0)} = J/n^2$ .
- 2: Initialize  $X_{\text{best}} \leftarrow X^{(0)}$
- 3: **for**  $k = 0$  to  $N$  **do**
- 4:     Compute  $N_k = \|X^{(k)} \odot \exp_{\odot}(-\gamma_k(A^2 X^{(k)} + X^{(k)} B^2 - 2AX^{(k)}B))\|_{1,1}$
- 5:     Update  $X^{(k+1)} = X^{(k)} \odot \exp_{\odot}(-\gamma_k(A^2 X^{(k)} + X^{(k)} B^2 - 2AX^{(k)}B))/N_k$
- 6:     **if**  $E(X^{(k+1)}) < E(X_{\text{best}})$  **then**
- 7:         Reassign  $X_{\text{best}} \leftarrow X^{(k+1)}$
- 8:     **end if**
- 9: **end for**
- 10: **return**  $\widehat{X}_{\mathcal{P}} := \text{GMWM}(X_{\text{best}})$

---

**Remark 1** (Complexity of Algorithm 2). *Each iteration within the for loop of Algorithm 2 has two main steps. First, the gradient computation, which has the same time complexity as matrix multiplication,  $\mathcal{O}(n^\omega)$  where  $\omega \leq 2.373$  (as established in [42]). Second, the Hadamard product of  $X^{(k)}$  and the entrywise exponentiated gradient, which has complexity  $n^2$ . Similarly, it is clear (see discussion in [66]) that Algorithm 1 has complexity  $\mathcal{O}(n^2)$ . Hence, overall, Algorithm 2 has a time complexity of  $\mathcal{O}(Nn^\omega)$ .*

### 2.3 Generative models for correlated random graphs

We now recall the two most popular models for the statistical study of the graph matching problem. These models generate a pair of correlated adjacency matrices  $A, B$ , based on a ground truth permutation  $X^* \in \mathcal{P}_n$ . Noteworthy fact for these models is that the solution of the QAP formulation (1.1) coincides with the maximum likelihood estimator of  $X^*$ . This is not true for other models, as shown in [64], in the case of random graph models with an underlying geometric structure.

**Correlated Gaussian Wigner (CGW) model  $\mathcal{W}(n, \sigma, X^*)$ .** The Correlated Gaussian Wigner (CGW) model, introduced in [23], defines a distribution on pairs of complete weighted graphs, such that each of them is (marginally) distributed as a random symmetric Gaussian matrix and their correlation is described by a single parameter  $\sigma$ . More specifically, we will say that  $A, B \in \mathbb{R}^{n \times n}$  follow the CGW model, and write  $A, B \sim \mathcal{W}(n, \sigma, X^*)$  if  $B = X^{*T} A X^* + \sigma Z$  where  $\sigma > 0$  and  $A, Z$  are i.i.d GOE( $n$ ) and  $X^* \in \mathcal{P}_n$ . We recall that a  $n \times n$  matrix  $A$  follows the GOE( $n$ ) law if its entries are distributed as

$$A_{ij} \sim \begin{cases} \mathcal{N}(0, \frac{1}{n}) & \text{if } i < j, \\ \mathcal{N}(0, \frac{2}{n}) & \text{if } i = j, \end{cases}$$

and  $A_{ij} = A_{ji}$  for all  $i, j \in [n]$ .

**Correlated Erdős-Rényi (CER) model**  $G(n, \sigma, p, X^*)$ . For  $\sigma, p \in [0, 1]$ , we describe the correlated Erdős-Rényi model with latent permutation  $X^* \in \mathcal{P}_n$  by the following two-step sampling process.

1.  $A$  is sampled according to the classic single graph Erdős-Rényi model  $G(n, p)$ , i.e. for all  $i < j$ ,  $A_{ij}$  are independent Bernoulli's random variables with parameter  $p$ ,  $A_{ji} = A_{ij}$  and  $A_{ii} = 0$ .
2. Conditionally on  $A$ , the entries of  $B$  are i.i.d according to the following law

$$(X^{*T} B X^*)_{ij} \sim \begin{cases} \text{Bern}(1 - \sigma^2(1 - p)) & \text{if } A_{ij} = 1, \\ \text{Bern}(\sigma^2 p) & \text{if } A_{ij} = 0. \end{cases} \quad (2.5)$$

Another equivalent description of this model in the literature involve first sampling an Erdős-Rényi “mother” graph, and then obtaining  $A, B$  as independent subsamples with certain density parameter (related to  $\sigma$ ). We refer to [55] for details on this alternative formulation.

### 3 Theoretical results

This section is dedicated to presenting our theoretical results. In Section 3.1, we prove that (1.6) has a unique solution with high probability, which, in the noiseless setting, corresponds to the scaled ground truth permutation. This, along with [11, Theorems 9.16 and 9.18], ensures that EMD and PGD converge to a unique point (under certain conditions on  $\{\gamma_k\}_{k=0}^{N-1}$  and  $N$ ), which in the noiseless setting leads to exact recovery of the ground truth permutation. In Section 3.2, we examine the EMD dynamic (2.4) in the noiseless setting ( $\sigma = 0$ ). We establish, as demonstrated in Theorem 2 below, that in this scenario, a single iteration of Algorithm 2 is sufficient to almost surely recover the ground truth permutation for any positive step-size  $\gamma_0$ . In Section 3.3, we prove a stronger version, as shown in Theorem 3, of a critical lemma originally found in [26] (specifically, [26, Lemma 2.3]) for the GRAMPA algorithm. This lemma forms the cornerstone of GRAMPA's theoretical guarantees for the general noisy setting.

#### 3.1 Uniqueness of the solution of simplex Graph Matching

In this section, we will prove that under the CGW model, the solution of (1.6) is unique with high probability. As usual, we assume without loss of generality that  $X^* = \text{Id}$ . Note that if  $\sigma > 0$ , then the objective  $\|AX - XB\|_F^2$  is strongly convex a.s, so the solution of (1.6) is unique a.s. Hence, we focus on the noiseless case where (1.6) reduces to

$$\min_{X \in \Delta_{n^2}} \|AX - XA\|_F^2. \quad (3.1)$$

Notice that under the CGW model,  $A \sim \text{GOE}(n)$ , and it is well-known that  $A$  has distinct eigenvalues almost surely [4, Theorem 2.5.2]. In this case, we know that  $X = \frac{1}{n} \text{Id} \in \Delta_{n^2}$  is a solution of (3.1), which implies that any solution has to satisfy  $\|AX - XA\|_F = 0$  or, equivalently,  $AX - XA = 0$ . In vector notation, this can be written as

$$\text{vec}(AX - XA) = H \text{vec}(X) = 0,$$

where we recall that  $H = (\text{Id} \otimes A - A \otimes \text{Id})$ . It is easy to see that the null space of  $H$  is given by  $\text{span}\{v_i \otimes v_i\}_{i=1}^n$ , where  $v_1, \dots, v_n$  are the eigenvectors of  $A$ . Therefore, all the solutions of (3.1) satisfy two conditions: first, they are of the form  $X = \sum_{i=1}^n \mu_i v_i v_i^T$  for some reals  $\mu_1, \dots, \mu_n$  (i.e.  $X \in \text{span}\{v_1 v_1^T, \dots, v_n v_n^T\}$ ), and second, they belong to  $\Delta_{n^2}$ .

**Theorem 1.** *Let  $A \sim \text{GOE}(n)$ , then with probability larger than  $1 - \frac{n}{2^{n-1}}$ , the problem (3.1) has a unique solution.*

Before proving Theorem 1, we first recall some concepts that will be useful in the proof. The following definitions are taken from [37, Section 6.2]. A matrix  $M \in \mathbb{R}^{n \times n}$  is reducible if there exist  $P \in \mathcal{P}_n$ ,  $r \in [n - 1]$  and matrices  $M_{11} \in \mathbb{R}^{r \times r}$ ,  $M_{12} \in \mathbb{R}^{(n-r) \times r}$ ,  $M_{22} \in \mathbb{R}^{(n-r) \times (n-r)}$  such that

$$PMP^T = \begin{bmatrix} M_{11} & M_{12} \\ 0_{(n-r),r} & M_{22} \end{bmatrix},$$

where  $0_{(n-r),r}$  is the matrix with all zero entries in  $\mathbb{R}^{(n-r) \times r}$ . On the other hand, a square matrix is irreducible if it is not reducible. We will denote  $\mathcal{I}$  the set of irreducible matrices and  $\mathcal{R}$  the set of reducible matrices (their dimension will be clear from context). It is easy to see that if  $M$  is symmetric, then

$$\begin{aligned} M \in \mathcal{R} &\iff \exists m < n, (B_{kk})_{k=1}^m \text{ with } B_{kk} \in \mathbb{R}^{n_k \times n_k} \text{ and } B_{kk} \in \mathcal{I} \\ &\exists P \in \mathcal{P}_n \text{ s.t. } PMP^T = \text{blockdiag}(B_{11}, \dots, B_{mm}). \end{aligned} \quad (3.2)$$

This block diagonal representation (3.2) is referred to as the Frobenius (or irreducible) normal form of  $M$ . We will use repeatedly the fact that if  $M \in \mathcal{I}$  and has nonnegative entries, then the ‘‘strong’’ form of the Perron-Frobenius theorem (see [37, Theorem 8.4.4]) applies. As a consequence, the largest modulus eigenvalue of  $M \in \mathcal{I}$  is (algebraically) simple and its associated eigenvector can be chosen with all its entries strictly larger than zero.

*Proof of Theorem 1.* Let  $\mathcal{S}$  be the (random) set of solutions<sup>3</sup> of (3.1). By the remarks at the beginning of this section, we have

$$\mathcal{S} = \text{span} \{v_i v_i^T\}_{i=1}^n \cap \mathbf{\Delta}_{n^2},$$

where  $v_1, \dots, v_n$  are the eigenvectors of  $A$ . Note that  $\mathcal{S}$  is nonempty, because clearly  $\frac{1}{n} \text{Id} \in \mathcal{S}$ . Our objective is to control the probability of the event

$$\mathcal{E} := \left\{ \mathcal{S} \neq \left\{ \frac{1}{n} \text{Id} \right\} \right\}.$$

Consider  $\mathcal{R}_1$  to be the class of reducible matrices with a Frobenius normal form having at least one block of size one, and similarly,  $\mathcal{R}_2$  will denote the class of reducible matrices with a Frobenius normal form with all of its blocks of size two or more (so clearly  $\{\mathcal{R}_1, \mathcal{R}_2\}$  form a partition of  $\mathcal{R}$ ). With this definition, it is easy to see that

$$\mathcal{E} = \underbrace{\{\mathcal{S} \cap \mathcal{I} \neq \emptyset\}}_{=: \mathcal{E}_1} \cup \underbrace{\left\{ \mathcal{S} \cap \mathcal{R}_1 \neq \left\{ \frac{1}{n} \text{Id} \right\} \right\}}_{=: \mathcal{E}_2} \cup \underbrace{\{\mathcal{S} \cap \mathcal{R}_2 \neq \emptyset\}}_{=: \mathcal{E}_3}.$$

Here, the events  $\mathcal{E}_1$ ,  $\mathcal{E}_2$ , and  $\mathcal{E}_3$  are clearly disjoint. We will bound their probabilities separately.

---

<sup>3</sup>This set corresponds to  $\mathcal{S}(A)$  introduced in Section 2. Here we drop the dependence on  $A$  to ease the notation.

**Bounding  $\mathbb{P}(\mathcal{E}_1)$ .** Notice that if there exists a matrix  $X$  in  $\mathcal{S} \cap \mathcal{I}$ , then for that  $X$ , the Perron-Frobenius theorem applies, which justifies the implication

$$X \in \mathcal{S} \cap \mathcal{I} \implies X \in \text{span} \{v_i v_i^T\}_{i=1}^n, \text{ and} \\ \exists i' \in [n] \text{ s.t. } v_{i'}(k) > 0 \ \forall k \in [n], \text{ or } v_{i'}(k) < 0 \ \forall k \in [n],$$

where  $v_{i'}(k)$  denotes the  $k$ -th coordinate of  $v_{i'}$ . From this we deduce that

$$\mathcal{E}_1 \subseteq \{\exists i' \text{ s.t. } v_{i'}(k) > 0, \forall k \in [n]\} \cup \{\exists i' \text{ s.t. } v_{i'}(k) < 0, \forall k \in [n]\}.$$

We will need the following lemma, whose proof can be found in Appendix C.

**Lemma 4.** *Let  $v_1, v_2, \dots, v_n$  be a set of (not necessarily independent) vectors uniformly distributed on the unit sphere  $\mathbb{S}^{n-1}$ . Then it holds*

$$\mathbb{P}(\exists i' \in [n] \text{ s.t. } v_{i'}(k) > 0, \forall k \in [n]) \leq \frac{n}{2^n}.$$

Given that  $v_1, \dots, v_n$  are uniformly distributed in the unit sphere  $\mathbb{S}^{n-1}$ , we use Lemma 4 and the union bound, to conclude that

$$\mathbb{P}(\mathcal{E}_1) \leq \frac{n}{2^{n-1}}. \quad (3.3)$$

**Bounding  $\mathbb{P}(\mathcal{E}_2)$ .** Fix any  $X \in \mathcal{S} \cap \mathcal{R}_1$ . We have that there exists at least one block in the Frobenius normal form of  $X$  (c.f. (3.2)) with size one. For simplicity, we can assume w.l.o.g. that its first block is of size one, i.e.,  $B_{11} \in \mathbb{R}$ . Given that entries of  $X$  are nonnegative, it is clear that  $B_{11} \geq 0$ . Then, the following chain of implications holds (denoting  $e_1$  the first element of the canonical basis of  $\mathbb{R}^n$ )

$$\begin{aligned} PXP^T e_1 &= B_{11} e_1 \\ \implies XP^T e_1 &= B_{11} P^T e_1 \\ \implies \sum_{i=1}^n \mu_i v_i \langle v_i, P^T e_1 \rangle &= B_{11} P^T e_1 \\ \implies VD\boldsymbol{\mu} &= B_{11} P^T e_1, \end{aligned} \quad (3.4)$$

where  $V$  has as columns  $v_1, \dots, v_n$ ,  $D$  is a diagonal matrix with entries  $D_{ii} = \langle v_i, P^T e_1 \rangle$  and  $\boldsymbol{\mu} = (\mu_1, \dots, \mu_n)^T$ . We have the following simple result.

**Claim 1.**  *$D$  is almost surely invertible.*

*Proof.* Clearly  $D$  is invertible iff  $D_{ii} \neq 0$  for all  $i \in [n]$ . Note that, there exists  $k' \in [n]$  such that  $P^T e_1 = e_{k'}$ . Hence,  $D_{ii} = \langle v_i, e_{k'} \rangle$  for some  $k' \in [n]$ . We claim that

$$\mathbb{P}(\forall k, i \in [n], \langle v_i, e_k \rangle \neq 0) = 1. \quad (3.5)$$

Indeed, for any given pair  $i, k \in [n]$ , we have that  $\mathbb{P}(\langle v_i, e_k \rangle = 0) = 0$  (because  $v_i$  is uniformly distributed on  $\mathbb{S}^{n-1}$ ). Then (3.5) follows from the union bound. From (3.5), we deduce that  $D_{ii} \neq 0$  a.s.  $\square$

Claim 1 implies that  $VD$  is almost surely invertible, which in turn implies that there exists a unique  $\boldsymbol{\mu}$  that satisfies  $VD\boldsymbol{\mu} = B_{11} P^T e_1$ . On the other hand, it is clear that  $\boldsymbol{\mu} = B_{11} \mathbf{1}$  is a solution of (3.4), so it has to be the unique solution. But, if  $\boldsymbol{\mu} = B_{11} \mathbf{1}$ , then  $X = B_{11} \text{Id}$ . However, the fact that  $X \in \mathcal{S}$  implies that  $B_{11} = \frac{1}{n}$ . In summary, we have demonstrated that for any  $X \in \mathcal{S} \cap \mathcal{R}_1$ , it holds that  $X = \frac{1}{n} \text{Id}$  almost surely. In other words, with probability one, there are no elements in  $\mathcal{S} \cap \mathcal{R}_1$  other than  $\frac{1}{n} \text{Id}$ . Equivalently,  $\mathbb{P}(\mathcal{E}_2) = 0$ .

**Bounding**  $\mathbb{P}(\mathcal{E}_3)$ . Fix any  $X \in \mathcal{S} \cap \mathcal{R}_2$  (assuming it exists). Then for some  $P \in \mathcal{P}_n$ , we have

$$PXP^T = \underbrace{\text{blockdiag}(B_{11}, \dots, B_{mm})}_{=:B}, \quad (3.6)$$

with all the blocks  $B_{kk} \in \mathbb{R}^{n_k \times n_k}$  having size  $n_k \geq 2$ , and  $B_{kk} \in \mathcal{I}$ . We note that

$$AX = XA \iff PAP^T B = BPAP^T.$$

We will use the following lemma.

**Lemma 5.** For  $A \sim \text{GOE}(n)$  and  $B$  as defined in (3.6), it holds

$$\mathbb{P}(\forall Q \in \mathcal{P}_n, QAQ^T B \neq BQ AQ^T) = 1.$$

*Proof.* Fix  $Q \in \mathcal{P}_n$ . Consider the following block decomposition of  $\tilde{A} := QAQ^T$ .

$$\tilde{A} = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} & \cdots & \tilde{A}_{1m} \\ \tilde{A}_{21} & \tilde{A}_{22} & \ddots & \vdots \\ \vdots & & \ddots & \\ \tilde{A}_{m1} & \cdots & & \tilde{A}_{mm} \end{bmatrix}$$

where  $\tilde{A}_{kl}$  belongs to  $\mathbb{R}^{n_k \times n_l}$  (we use bold notation here to distinguish  $\tilde{A}_{kl}$  from the  $(k, l)$  entry of  $\tilde{A}$ ). Observe that  $\tilde{A}_{kl} = \tilde{A}_{lk}^T$ . With this, it is easy to verify that

$$\tilde{A}B = B\tilde{A} \iff \tilde{A}_{kl}B_{ll} = B_{kk}\tilde{A}_{kl}, \quad \forall k, l \in [m],$$

which implies that

$$\tilde{A}_{lk}\tilde{A}_{kl}B_{ll} = B_{ll}\tilde{A}_{lk}\tilde{A}_{kl}, \quad \forall k, l \in [m] \text{ with } k \neq l,$$

and

$$\tilde{A}_{ll}B_{ll} = B_{ll}\tilde{A}_{ll}, \quad \forall l \in [m].$$

In the sequel, we will assume w.l.o.g. that  $n_1 \leq n_2$  and only use the following implication

$$\tilde{A}B = B\tilde{A} \implies \tilde{A}_{12}\tilde{A}_{21}B_{11} = B_{11}\tilde{A}_{12}\tilde{A}_{21}, \quad \text{and} \quad \tilde{A}_{11}B_{11} = B_{11}\tilde{A}_{11}. \quad (3.7)$$

It can be seen that the following facts hold.

- (i)  $\tilde{A} \stackrel{\text{dist.}}{=} A \sim \text{GOE}(n)$ . In particular,  $\tilde{A}_{11} \sim \text{GOE}(n_1)$ , which implies that  $\tilde{A}_{11}$  has  $n_1$  different eigenvalues almost surely (c.f. [4, Thm. 2.5.2]). Hence, its eigenvectors, denoted  $\tilde{v}_1^{(1)}, \dots, \tilde{v}_{n_1}^{(1)}$  are uniquely defined up to sign change.
- (ii)  $\tilde{A}_{12}\tilde{A}_{21} = \tilde{A}_{12}\tilde{A}_{12}^T \sim \text{Wish}(n_1, n_2)$ . Here  $M \sim \text{Wish}(n_1, n_2)$  means that  $M \in \mathbb{R}^{n_1 \times n_1}$  can be written  $M = \tilde{M}\tilde{M}^T$ , where  $\tilde{M} \in \mathbb{R}^{n_1 \times n_2}$  is a matrix with independent Gaussian entries with variance  $\frac{1}{n}$ . From [4, Prop. 4.1.3], we see that  $\tilde{A}_{12}\tilde{A}_{21}$  has  $n_1$  different eigenvalues almost surely. Hence, its eigenvectors, denoted  $\tilde{v}_1^{(1,2)}, \dots, \tilde{v}_{n_1}^{(1,2)}$  are uniquely determined up to sign change.
- (iii)  $\tilde{A}_{12}\tilde{A}_{21}$  is independent from  $\tilde{A}_{11}$ .

Given that  $B_{11} \in \mathcal{I}$ , we deduce, by Perron-Frobenius theorem, that its largest eigenvalue has multiplicity one. On the other hand, by (3.7) we have that

$$B_{11} \in \text{span} \left( \{ \tilde{v}_i^{(1)} \tilde{v}_i^{(1)T} \}_{i=1}^{n_1} \right) \cap \text{span} \left( \{ \tilde{v}_i^{(1,2)} \tilde{v}_i^{(1,2)T} \}_{i=1}^{n_1} \right).$$

In other words,  $\{ \tilde{v}_1^{(1)}, \dots, \tilde{v}_{n_1}^{(1)} \}$  and  $\{ \tilde{v}_1^{(1,2)}, \dots, \tilde{v}_{n_1}^{(1,2)} \}$  are both basis of eigenvectors for  $B_{11}$ . Assume w.l.o.g that the largest eigenvalue of  $B_{11}$  (of multiplicity one) is associated with  $\tilde{v}_1^{(1)}$ , in the first basis, and  $\tilde{v}_1^{(1,2)}$ , in the second. This implies that either  $\tilde{v}_1^{(1)} = \tilde{v}_1^{(1,2)}$  or  $\tilde{v}_1^{(1)} = -\tilde{v}_1^{(1,2)}$ . From facts (i) and (ii),  $\tilde{v}_1^{(1)}$  and  $\tilde{v}_1^{(1,2)}$  are uniformly distributed in  $\mathbb{S}^{n_1-1}$ . From (iii),  $\tilde{v}_1^{(1)}$  and  $\tilde{v}_1^{(1,2)}$  are independent. We deduce that  $\mathbb{P}(\tilde{v}_1^{(1)} = \tilde{v}_1^{(1,2)}) = \mathbb{P}(\tilde{v}_1^{(1)} = -\tilde{v}_1^{(1,2)}) = 0$ . Summarizing, for our fixed permutation  $Q$  we have

$$\mathbb{P}(Q A Q^T B = B Q A Q^T) \leq \mathbb{P}(\tilde{v}_1^{(1)} = \tilde{v}_1^{(1,2)}) + \mathbb{P}(\tilde{v}_1^{(1)} = -\tilde{v}_1^{(1,2)}) = 0.$$

Then, the stated result follows from a union bound over the (finite) set  $\mathcal{P}_n$ .  $\square$

Using Lemma 5, we conclude that  $\mathbb{P}(\mathcal{E}_3) = 0$ .

**Concluding the proof.** We have obtained

$$\begin{aligned} \mathbb{P}(\mathcal{E}) &= \mathbb{P}(\mathcal{E}_1) + \underbrace{\mathbb{P}(\mathcal{E}_2)}_{=0} + \underbrace{\mathbb{P}(\mathcal{E}_3)}_{=0} \\ &\leq \frac{n}{2^{n-1}}, \end{aligned}$$

which concludes the argument.  $\square$

### 3.2 Noiseless EMD dynamics

We now consider the “noiseless” version of the GM problem, where  $\sigma = 0$  or equivalently  $A = B$  (given that we assumed  $X^* = \text{Id}$ ), which makes the gradient  $\nabla E(X)$  that appears in (2.4) equal to  $A^2 X + X A^2 - 2 A X A$ . We consider the CGW model, or equivalently, when  $A \sim \text{GOE}(n)$ . Interestingly, in this case one iteration of Algorithm 2 suffices to recover the ground truth (i.e., we can fix  $N = 1$ ).

**Theorem 2.** *Let  $A \sim \text{GOE}(n)$  and let  $X^{(1)}$  be the first iterate of the dynamic defined by (2.4) with initialization  $X^{(0)} = J/n^2$ . Then  $\text{GMWM}(X^{(1)}) = \text{Id}$  a.s., for any  $\gamma_0 > 0$ .*

To prove the previous theorem, we will need the following auxiliary result.

**Lemma 6.** *Let  $A \sim \text{GOE}(n)$ . Then for any  $i, j \in [n]$  with  $i \neq j$ ,  $-\nabla E(X^{(0)})$  satisfies a.s. the property (2.2), i.e. the following holds almost surely*

$$\nabla E(J)_{ii} + \nabla E(J)_{jj} - 2\nabla E(J)_{ij} < 0.$$

*Proof.* We use the decomposition

$$\nabla E(J)_{ii} + \nabla E(J)_{jj} - 2\nabla E(J)_{ij} = M_{ii} + M_{jj} - 2M_{ij} + M'_{ii} + M'_{jj} - 2M'_{ij}$$

where  $M := A^2J + JA^2$  and  $M' := -2AXA$ . Define  $a_k$  as the  $k$ -th column of  $A$ , with this we have for  $i, j \in [n]$

$$\begin{aligned} (A^2J + JA^2)_{ij} &= \langle a_i, \sum_{k=1}^n a_k J_{kj} \rangle + \langle \sum_{k=1}^n a_k J_{ik}, a_j \rangle \\ &= \langle \sum_{k=1}^n a_k, a_i + a_j \rangle, \end{aligned}$$

which clearly implies  $M_{ii} + M_{jj} = 2M_{ij}$ . For the term  $M'$  we have

$$M'_{ii} + M'_{jj} - 2M'_{ij} = - \underbrace{\langle (a_i - a_j), J(a_i - a_j) \rangle}_{= \langle (a_i - a_j), \mathbf{1} \rangle^2}.$$

Since  $\langle a_i - a_j, \mathbf{1} \rangle = \sum_{k \notin \{j, i\}} [A_{ik} - A_{jk}]$ , it is clear that  $\langle a_i - a_j, \mathbf{1} \rangle \neq 0$  almost surely.  $\square$

Observe that in this case, it follows from its definition that  $X^{(1)}$  is symmetric. Indeed, it is the Hadamard product of  $X^{(0)}$  and  $\exp_{\odot}(-\gamma_0 \nabla E(X^{(0)}))$ , and both are symmetric matrices.

*Proof of Theorem 2.* Take  $i, j \in [n]$ . From (2.4) and  $X^{(0)} = J/n^2$  we deduce

$$\begin{aligned} X_{ij}^{(1)} &= N_0^{-1} e^{-\gamma_0 \nabla E\left(\frac{J}{n^2}\right)_{ij}} \\ &= N_0^{-1} e^{-\frac{\gamma_0}{n^2} (A^2J + JA^2 - 2AJA)_{ij}} \end{aligned}$$

where the scalar random variable  $N_0 = \|e^{-\frac{\gamma_0}{n^2} (A^2J + JA^2 - 2AJA)_{ij}}\|_{1,1}$  is clearly almost surely strictly positive (we will only need this fact in the sequel). Define the following quantity  $q_{ij} = \frac{\sqrt{X_{ii}^{(1)} X_{jj}^{(1)}}}{X_{ij}^{(1)}}$ , which is well-defined given that  $X_{ij}^{(1)} > 0$ , almost surely. On the other hand,

$$q_{ij} = \exp(-\gamma_0(\nabla E(J)_{ii} + \nabla E(J)_{jj} - 2\nabla E(J)_{ij})).$$

By Lemma 6 we deduce that for  $i \neq j$ , we have almost surely that  $q_{ij} > 1$ . Which means that  $X_{ij}^{(1)} < \sqrt{X_{ii}^{(1)} X_{jj}^{(1)}}$ , so we deduce that  $X_{ij}^{(1)} < \frac{1}{2}(X_{ii}^{(1)} + X_{jj}^{(1)})$ . Consequently,  $X^{(1)}$  satisfies (2.2), and given the symmetry of  $X^{(1)}$ , we deduce that  $\text{GMWM}(X^{(1)}) = \text{Id}$  almost surely.  $\square$

**Remark 2** (The Erdős-Rényi case). *Clearly, if we take an Erdős-Rényi graph  $A \sim G(n, p)$ , the argument in Lemma 6 will not work, since there are, with probability one, at least two vertices (say  $i, j \in [n]$ ) with the same degree, hence  $\langle a_i - a_j, \mathbf{1} \rangle = 0$ . Using the degree profile distribution of Erdős-Rényi graphs, which is a well-studied problem [14, 15], one can give a bound on the number of vertices that will be correctly assigned by Algorithm 2 with  $N = 1$ . We remark that for the graph isomorphism problem, more elaborated “signatures” using the degrees have been used with success under the Erdős-Rényi model [10, 20, 54, 23].*

Given a similarity matrix  $\widehat{X}$ , several existing theoretical results for exact recovery (see [26, 66, 48, 5]) use the following sufficient condition for ensuring  $\text{GMWM}(\widehat{X}) = \text{Id}$ ;

$$\widehat{X}_{ii} > \max_{j \neq i} \widehat{X}_{ij}, \quad \forall i \in [n], \quad (3.8)$$

this is known as diagonal dominance. It is clear that (3.8) implies (2.1) (which is equivalent to (2.2) in the symmetric case). The approach of using (3.8) instead of (2.2) to prove Theorem 2 fails. Indeed, when  $\hat{X} = X^{(1)}$  as in Theorem 2, showing (3.8) is equivalent to showing  $\nabla E(J)_{ii} < \nabla E(J)_{ij}$ , for all  $j \neq i$ . A simple calculation shows that

$$\begin{aligned} \mathbb{E}[\nabla E(J)_{ii}] &= -1 + o(1), \quad \mathbb{E}[\nabla E(J)_{ij}] = o(1) \quad \forall i \neq j, \\ \text{and } \text{Var}[\nabla E(J)_{ij}] &= 3 - o(1) \quad \forall i, j. \end{aligned}$$

Then using Chebyshev's inequality, we deduce that the event  $\nabla E(J)_{ii} \geq \nabla E(J)_{ij}$  occurs with at least a constant probability, for any given  $i \neq j$ . Hence it follows that (3.8) will fail to hold with at least a constant probability.

### 3.3 Alternative analysis for GRAMPA in noiseless setting

To further illustrate the advantage of using (2.2) over (3.8), we will prove a stronger version of [26, Lemma 2.3] which provides guarantees for exact recovery (in the noiseless case) for the GRAMPA algorithm that operates as follows. First, for a regularization parameter  $\eta$ , it computes the similarity matrix

$$\hat{X}_{\text{grampa}} := \sum_{i,j=1}^n \frac{1}{\eta^2 + (\lambda_i - \lambda_j)^2} v_i v_i^T J v_j v_j^T. \quad (3.9)$$

Here,  $A = \sum_{i=1}^n \lambda_i v_i v_i^T$ . Next, the algorithm rounds  $\hat{X}_{\text{grampa}}$  using any rounding procedure (e.g., linear assignment, greedy rounding etc.). For the result in [26, Lemma 2.3], however, conditions are derived on  $\eta$  which ensure that  $\hat{X}_{\text{grampa}}$  is diagonally-dominant<sup>4</sup>, with high probability.

In the following theorem, we show for the noiseless setting that exact recovery is in fact possible for any choice of  $\eta$ .

**Theorem 3.** *Let  $A$  be a  $GOE(n)$  distributed matrix with spectral expansion  $A = \sum_{i=1}^n \lambda_i v_i v_i^T$ . Then for any  $\eta$ , the matrix  $\hat{X}_{\text{grampa}}$ , defined in (3.9), satisfies*

$$GMWM(\hat{X}_{\text{grampa}}) = \text{Id},$$

almost surely.

*Proof.* Define the matrix  $\Lambda_\eta$ , with entries

$$(\Lambda_\eta)_{ij} = (\lambda_i - \lambda_j)^2 + \eta^2.$$

Observe that  $\hat{X}_{\text{grampa}}$  can be rewritten as

$$\hat{X}_{\text{grampa}} = V \left( \Lambda_\eta^{(-1)} \odot (V^T J V) \right) V^T, \quad (3.10)$$

where  $\Lambda_\eta^{(-1)}$  denotes the entrywise inverse of  $\Lambda_\eta$  and  $V$  is the matrix with columns  $v_1, \dots, v_n$ . Recall that a matrix  $M \in \mathbb{R}^{n \times n}$  is conditionally p.s.d (resp. conditionally p.d) if, for all  $v \neq 0$  such that  $v^T \mathbf{1} = 0$ , we have  $v^T M v \geq 0$  (resp.  $v^T M v > 0$ ); see e.g., [36, Section 6.3]. We now have the following claim.

**Claim 2.**  *$-\Lambda_\eta$  is conditionally p.s.d for all  $\eta \in \mathbb{R}$ .*

<sup>4</sup>In fact, they consider an even stronger version of diagonal-dominance than (3.8).



*Proof.* We will first prove that  $-\Lambda_0$  is conditionally p.s.d. For that, notice that

$$\Lambda_0 = (\boldsymbol{\lambda} \odot \boldsymbol{\lambda})\mathbf{1}^T + \mathbf{1}(\boldsymbol{\lambda} \odot \boldsymbol{\lambda})^T - 2\boldsymbol{\lambda}\boldsymbol{\lambda}^T,$$

where  $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_n)^T$ . Take  $v \in \mathbb{R}^n$  such that  $v \neq 0$  and  $v^T \mathbf{1} = 0$ . Then

$$\begin{aligned} v^T(-\Lambda_0)v &= 2(v^T \boldsymbol{\lambda})^2 - \underbrace{v^T(\boldsymbol{\lambda} \odot \boldsymbol{\lambda})\mathbf{1}^T v + v^T \mathbf{1}(\boldsymbol{\lambda} \odot \boldsymbol{\lambda})v}_{=0} \\ &\geq 0. \end{aligned}$$

On the other hand, it is evident that  $\Lambda_\eta = \Lambda_0 + \eta^2 J$ . Since adding a scalar times the all ones matrix preserves the conditional p.s.d property, we deduce that  $-\Lambda_\eta$  is conditionally p.s.d for any  $\eta$ .  $\square$

Claim 2 is useful in light of Theorem 5 (see Appendix E), which says that the entrywise exponential of a conditionally p.s.d matrix is p.s.d. We will use this to prove the following claim.

**Claim 3.**  $\Lambda_\eta^{(-1)}$  is almost surely p.d for all  $\eta \in \mathbb{R}$ .

*Proof.* We will first prove that  $\exp_\odot(-t\Lambda_\eta)$  is p.d, for all  $t > 0$ . For that, using Claim 2 we obtain that  $-t\Lambda_\eta$  is conditionally p.s.d. for all  $t \geq 0$ . By Theorem 5 part (iii),  $\exp_\odot(-t\Lambda_\eta)$  is p.d. if and only if  $-t\Lambda_\eta$  satisfies (2.2). It is clear that for all  $i \neq j$  and  $t > 0$

$$\underbrace{-t(\Lambda_\eta)_{ii} - t(\Lambda_\eta)_{jj}}_{=-2t\eta^2} > \underbrace{-2t(\Lambda_\eta)_{ij}}_{=-2t(\lambda_i - \lambda_j)^2 - 2t\eta^2} \quad \text{almost surely,}$$

since  $\lambda_i \neq \lambda_j$  almost surely [4, Theorem 2.5.2]. Hence,  $\exp_\odot(-t\Lambda_\eta)$  is almost surely p.d. for all  $t > 0$ . Given that  $\frac{1}{s} = \int_0^\infty e^{-st} dt$ , we have the representation

$$\Lambda_\eta^{(-1)} = \int_0^\infty \exp_\odot(-t\Lambda_\eta) dt,$$

where the integral of a matrix is understood entrywise. From Lemma 12, we have the integral of positive definite matrices is positive definite. Given that  $\exp_\odot(-t\Lambda_\eta)$  is almost surely p.d. for all  $t > 0$ , we conclude that  $\Lambda_\eta^{(-1)}$  is almost surely p.d.  $\square$

From Claim 3,  $\Lambda_\eta^{(-1)}$  is almost surely p.d and, on the other hand, it is evident that  $V^T J V$  is p.s.d. In addition, the diagonal entries of  $V^T J V$  (observe that  $(V^T J V)_{ii} = (v_i^T \mathbf{1})^2$ ) are almost surely different from zero, since  $v_i$ 's are uniformly distributed on the sphere (see e.g., [4]). Then by Lemma 11 we obtain that  $\Lambda_\eta^{(-1)} \odot (V^T J V)$  is almost surely p.d. From (3.10) it is easy to see that  $\hat{X}_{\text{grampa}}$  is almost surely p.d. Combining this with Lemma 2 part (iii), the conclusion follows.  $\square$

**Remark 3** (Comparison with Lemma 2.3 in [26]). *Theorem 3 is stronger than [26, Lemma 2.3] in two aspects. First, it provides an almost sure guarantee instead of the high probability bound in [26, Lemma 2.3]. Second, it holds for any  $\eta \in \mathbb{R}$ , while in [26, Lemma 2.3] the result requires that  $\frac{1}{n^{0.1}} < |\eta| < \frac{c}{\log n}$  for some constant  $c$ . In addition, our approach also leads to a shorter and arguably simpler proof.*

**Remark 4** (Extension to the Erdős-Rényi case). *Notice that in the proof of Theorem 3, the assumption that  $A \sim \text{GOE}(n)$  is crucial at two key points. First, to establish Claim 3, it is necessary that  $A$  almost surely possesses a simple spectrum. Second, we rely on this assumption to demonstrate that the quantities  $v_i^T \mathbf{1}$  are almost surely non-zero for  $i \in [n]$ . In light of [61, Corollary 1.4], we believe that it should be relatively straightforward to extend Theorem 3 to the case where  $A \sim G(n, p)$ , at least in a high probability sense.*

## 4 Numerical experiments

We now provide numerical experiments using both synthetic and real data to assess the performance of Algorithm 2. We evaluate its performance along with other competing graph matching algorithms, considering both accuracy and running time as key metrics. In graph matching, the customary measure of accuracy is the overlap with the ground truth  $X^* \in \mathcal{P}_n$ , also known as the *recovery fraction*. This measure, denoted as  $\text{overlap}(X^*, \widehat{X}_{\mathcal{P}})$ , is defined as:

$$\text{overlap}(X^*, \widehat{X}_{\mathcal{P}}) := \frac{1}{n} \langle X^*, \widehat{X}_{\mathcal{P}} \rangle_F,$$

where  $\widehat{X}_{\mathcal{P}}$  represents the output of any graph matching algorithm, and we recall that  $\langle X^*, \widehat{X}_{\mathcal{P}} \rangle$  denotes the Frobenius inner product between  $X^*$  and  $\widehat{X}_{\mathcal{P}}$ .

All the experiments were conducted using MATLAB R2021a (MathWorks Inc., Natick, MA) on an Apple M1 machine with 8 cores, 3.2 GHz clock speed, and 16GB of RAM. The reported running times are based on this setup.

### 4.1 Synthetic data setup

For the synthetic data experiments, we follow this general setup: we generate matrices  $A$  and  $B$  using the CER and CGW models described in Section 2.3, with  $X^*$  chosen uniformly at random from  $\mathcal{P}_n$ . We then apply various graph matching methods based on convex relaxations, using  $A$  and  $B$  as inputs.

**Standardization of the inputs.** To ease the comparison between different models, we will consider their standardized version. Following [26], we consider the following standardized version of the CER model described in Sec. 2.3

$$\tilde{A} = \frac{(A - \mathbb{E}(A))}{p(1-p)n}, \quad \tilde{B} = \frac{(B - \mathbb{E}(B))}{p(1-p)n},$$

where  $A, B \sim G(n, \sigma, p, X^*)$ . This can be regarded as a simple preprocessing step in the pipeline of the three graph matching algorithms that will be experimentally tested here.

**Step-size selection.** An obvious question for any gradient descent-like method is the selection of a step-size updating rule. We investigate the following popular strategies for EMD and PGD.

- **Fixed step-sizes.** In [11, Theorem 9.16 and Example 9.17] it is proven that MD converges in  $\mathcal{O}\left(\frac{1}{\sqrt{N}}\right)$  iterations, under the following constant step-size rule

$$\gamma_k^{\text{md}} = \frac{\sqrt{2 \log n} L_{E,\infty}}{\sqrt{N+1}}, \quad \forall k \in [N], \tag{4.1}$$

where  $L_{E,\infty} := \max_{X \in \Delta_{n,2}} \|\nabla E(X)\|_{\infty}$  and for  $v \in \mathbb{R}^d$ , with  $d \in \mathbb{N}$ ,  $\|v\|_{\infty} := \max_{j \in \mathbb{N}} |v_j|$ . Given that  $L_{E,\infty}$  is, in general, not observed, for each  $k$  we replace it (as a heuristic) with the following observed quantity

$$\hat{L}_{E,\infty}^{(k)} = \max_{X \in \{X^{(0)}, \dots, X^{(k)}\}} \|\nabla E(X)\|_{\infty}. \tag{4.2}$$

Notice that, contrary to (4.1), the rule obtained by replacing  $L_{E,\infty}$  by the empirical quantity  $\hat{L}_{E,\infty}^{(k)}$  is not a fixed step strategy. Analogously, for the PGD algorithm, based on the same theoretical result we define the following update rule

$$\gamma_k^{\text{pgd}} = \frac{\sqrt{2}\hat{L}_{E,2}^{(k)}}{\sqrt{N+1}}, \quad \forall k \in [N], \quad (4.3)$$

where

$$\hat{L}_{E,2}^{(k)} = \max_{X \in \{X^{(0)}, \dots, X^{(k)}\}} \|\nabla E(X)\|_F. \quad (4.4)$$

- **Dynamic step-sizes.** The  $\mathcal{O}\left(\frac{1}{\sqrt{N}}\right)$  convergence of EMD and PGD has also been established, see [11, Theorem 9.18], under the following alternative strategy

$$\gamma_k^{\text{md}} = \begin{cases} \frac{\sqrt{2}}{\|\nabla E(X^{(k)})\|_\infty \sqrt{k+1}} & \text{if } \nabla E(X^{(k)}) \neq 0 \\ 0 & \text{if } \nabla E(X^{(k)}) = 0, \end{cases} \quad (4.5)$$

Notice that here  $\gamma_k^{\text{md}}$  can change from iteration to iteration, as it is adapting to the current gradient information. In the case of PGD the analogous rule is given by

$$\gamma_k^{\text{pgd}} = \begin{cases} \frac{\sqrt{2}}{\|\nabla E(X^{(k)})\|_F \sqrt{k+1}} & \text{if } \nabla E(X^{(k)}) \neq 0 \\ 0 & \text{if } \nabla E(X^{(k)}) = 0. \end{cases} \quad (4.6)$$

In the case that  $\nabla E(X^{(k)}) = 0$ , we have chosen to impute  $\gamma_k^{\text{md}}$  and  $\gamma_k^{\text{pgd}}$  with a zero value, but this choice is arbitrary and has no effect in the value of the iterates.

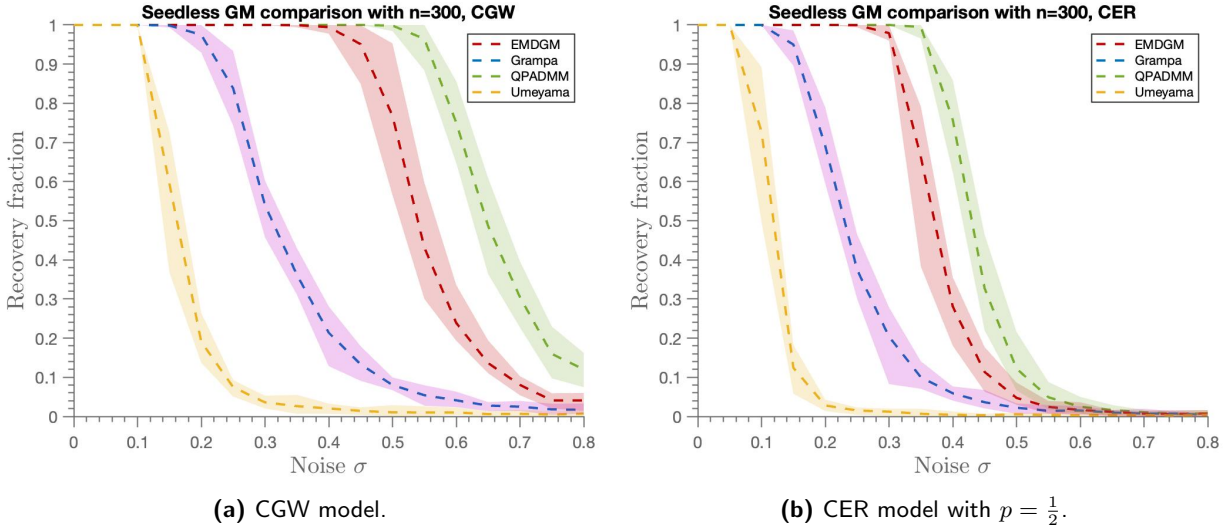
- **Other heuristic rules.** We also consider certain purely heuristic rules, which are not based on theoretical convergence results. In our experiments, these rules sometimes outperform the theoretically established rules described above. For PGD we consider

$$\gamma_k = \theta \frac{\|\nabla E(X^{(k)})\|_F^2}{\|E(X^{(k)})\|_F^2}, \quad (4.7)$$

where  $\theta$  is a positive constant, which is either fixed to 1 (default value) or determined by line search.

#### 4.1.1 Comparison with other convex optimization-based methods

We compare some of the state-of-art seedless graph matching methods based on convex relaxations with our proposed method, Algorithm 2, using a fixed number of iterations  $N = 125$ . In terms of efficient methods for convex graph matching, two of the better performing ones are **Grampa** [26] and **QPADMM**, the latter of which solves (1.5) with  $\mathcal{K}$  the set of doubly stochastic matrices, using the method of Alternating Direction Method of Multipliers (ADMM). The **QPADMM** algorithm has also been considered in the numerical experiments sections in [23, 26], under the name **QS-DS**. We recall that **Grampa** is a regularized spectral method, which adds the term  $\eta\|X\|_F^2$  to the objective function  $E(\cdot)$  in (1.5) with  $\mathcal{K} = \{\mathbf{1}^T X \mathbf{1} = n\}$ . Unless otherwise specified, we adopt the recommended value of  $\eta = 0.2$  (see [26, Section 4]). In our experiments, we observed that variations in this parameter have minimal impact on the overall performance of **Grampa**, which aligns with findings reported in [26]. We also include the classic **Umeyama** algorithm, which was originally introduced in [63], and corresponds to an unregularized spectral method.



**Figure 1:** Comparison of the performance of seedless convex methods for CGW (Fig. 1a) and CER (Fig. 1b) models. We plot the average overlap with the ground truth (recovery fraction) over 15 Monte Carlo runs for graphs of size  $n = 300$ . We used  $N = 125$  iterations in EMDGM and in Grampa we use the regularization parameter  $\eta = 0.2$ . For EMDGM we used the dynamic step-size rule (4.5).

**Accuracy comparison.** Figure 1 illustrates that, in terms of accuracy, the performance of EMDGM sits between Grampa and QPADMM, while Umeyama is outperformed by all three methods. While this result is not unexpected, considering the well-defined hierarchy of relaxations from the hyperplane to the simplex and, finally, to the Birkhoff polytope, these experiments aim to provide a clearer understanding of the gap between them. In this regard, taking the case of the CGW model in Fig.1a, we see that EMDGM achieves exact recovery for noise levels as high as  $\sigma \approx 0.5$ , while Grampa can recover only up to  $\sigma \approx 0.25$ . On the other hand, QPADMM outperforms all the rest, with a recovery of up to  $\sigma \approx 0.6$ . These experiments show that EMDGM (with  $N = 125$ ) is closer, in terms of accuracy, to the tightest relaxation of QPADMM. As we will see in the sequel, the performance of EMDGM is highly correlated with the number of iterations. In the case of the CER model, the gap between methods reduce and even QPADMM can recover only up to  $\sigma \approx 0.4$ . This confirms that matching CER graphs is more challenging for comparable levels of noise. One possible explanation for this fact is that the (continuous) weights in the CGW model help to discriminate the correct from incorrect matches (“breaking the symmetry”) more effectively than in the binary weights case, where one relies solely on the graph structure. From the theoretical point of view, we saw a manifestation of this phenomenon in Section 3, when trying to extend Theorem 1 to the CER case (see Remark 2). Overall, these experiments show that EMDGM is closer, in terms of accuracy, to the tightest relaxation of QPADMM.

**Running time comparison.** In Table 1, we summarize the average running time and standard deviation over the 15 Monte Carlo runs, for each algorithm. In these experiments, we observe that both Grampa and Umeyama are of comparable speed, while EMDGM operates about eight times slower than them. Conversely, QPADMM is the slowest, being about a thousand times slower than Grampa. In the next subsection, we will explore how it is possible to achieve good accuracy with EMDGM using fewer iterations, thus speeding up the overall process.

|     | Umeyama        | EMDGM          | Grampa         | QPADMM          |
|-----|----------------|----------------|----------------|-----------------|
| CGW | 0.0356(0.0211) | 0.8609(0.0282) | 0.0490(0.0048) | 34.2666(1.1065) |
| CER | 0.0382(0.015)  | 0.9226(0.0168) | 0.0432(0.0063) | 29.1306(1.1302) |

**Table 1:** Average (over 15 Monte Carlos) running time in seconds (standard deviation in parentheses).

### 4.1.2 Comparison with PGD

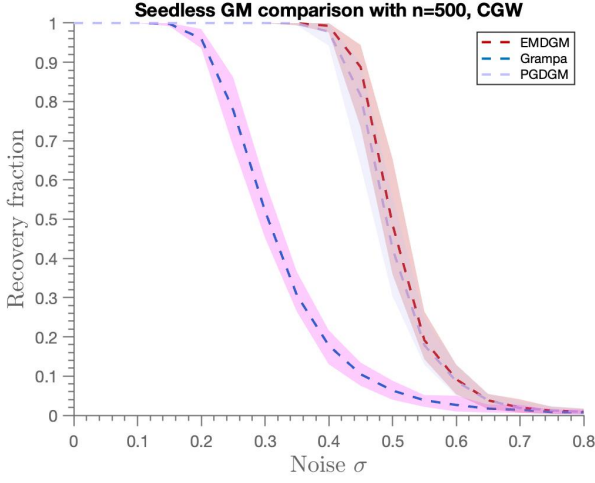
As mentioned earlier, the EMD algorithm is one of several algorithmic alternatives to solve (1.6). In this section, we have two objectives: Firstly, to compare it with other algorithms that achieve the same task, and secondly, to evaluate its performance with different values of  $N$ . For runtime efficiency, we confine our comparison to the category of first-order methods, among which PGD is arguably the most popular one. We will call PGDGM the algorithm that solves (1.6) using a gradient descent approach, receiving the same inputs and following the same pipeline as Algorithm 2, but changing the update rule from (2.4) to (2.3). Given the existing theoretical convergence guarantees for MD and PGD, see [11, Sections 8 and 9], we know that there exists rate sequences  $(\gamma_k)_{k \geq 0}$  such that both methods will converge to a minimizer of (1.6). Our objective here is to provide a practical comparison of both algorithms within a unified setup, focusing on the number of iterations and the selection of step-size rules. In this section, we will demonstrate that the overall results of both approaches often diverge on synthetic data (a pattern consistent with real data, as we demonstrate later in Section 4.2 below).

**Accuracy comparison.** In Fig. 2 we plot the average overlap between  $X^*$  and the output of EMDGM, PGDGM and Grampa. Among the step-size rules described in Section 4.1 above, the better performing ones in our experiments were the dynamic adaptive rule (4.5) for EMDGM and the heuristic rule for PGDGM with  $\theta$  close to 1 (we chose to include the default value  $\theta = 1$  in the plot). We see that all three algorithms perform better under the CGW model than under the CER model. Indeed, one can see in Fig. 2a that EMDGM and PGDGM can achieve perfect recovery for  $\sigma \approx 0.45$ , and Grampa for  $\sigma$  around 0.25. Interestingly, in the case of the CER model, we see that EMDGM performs better in terms of perfect recovery with a gap of at least 0.1 in the noise level it tolerates. When we restrict the number of the iterations to  $N = 25$ , we see that all models have an accuracy of 90% or more under a noise level as high as  $\sigma = 0.25$ . On the other hand, as seen in Fig. 2b, for the same noise level, only the EMDGM algorithm achieves 90%, under the setting of this experiment. Overall, EMDGM outperforms both Grampa and PGDGM in this experiments.

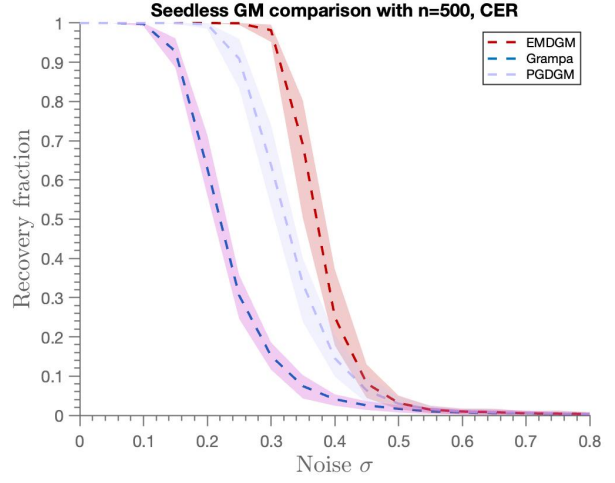
**Running time comparison.** Table 2 contains the running time information for the experiments shown in Figure 2. We see that Grampa is the fastest algorithm, running approximately five times as fast as EMDGM (with  $N = 25$ ), which in turn is approximately two times faster than PGDGM. Despite EMDGM being slower than Grampa, we demonstrate here that when terminated at  $N = 25$ , EMDGM still outperforms Grampa, achieving an average speed five times faster than the case with  $N = 125$ .

### 4.1.3 Experimental comparison between (2.1) and diagonal dominance

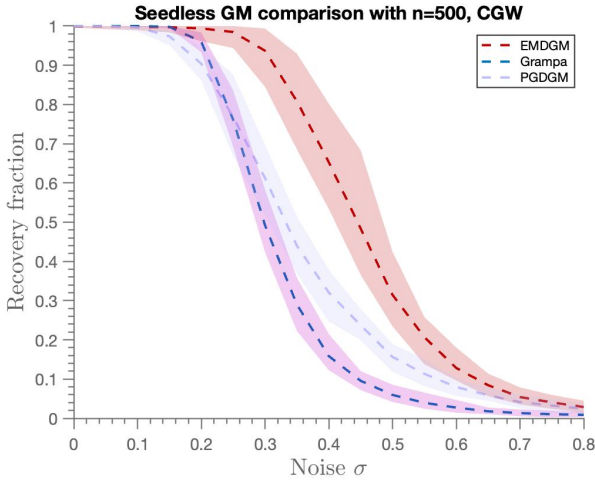
As previously noted, both EMDGM and Grampa adhere to the meta-strategy outlined in the Introduction: their output is the rounded version of a similarity matrix (specific for each algorithm). To prove exact recovery, a large part of the literature [26, 66, 48, 5] has focused on proving that



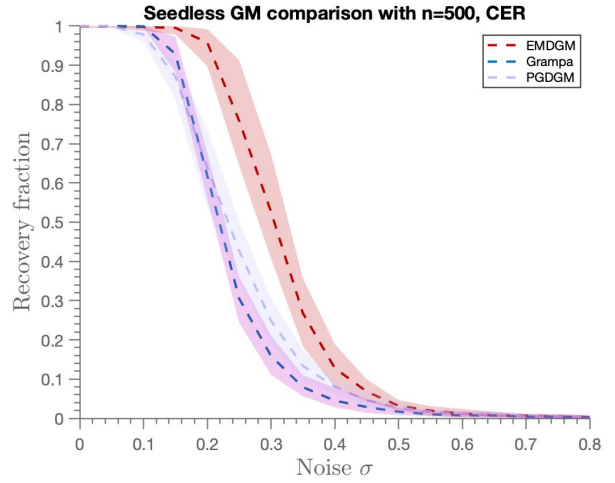
(a)  $N = 125$  in the CGW model.



(b)  $N = 125$  in the CER model with  $p = \frac{1}{2}$ .



(c)  $N = 25$  in the CGW model.



(d)  $N = 25$  in the CER model with  $p = \frac{1}{2}$ .

**Figure 2:** We compare EMDGM, Grampa and PGD under the CGW and CER models for  $n = 500$ . For EMDGM we used the dynamic step-size rule (4.5) and PGD we used the heuristic rule (4.7) with  $\theta = 1$ . In both cases, we fix the number of iterations to  $N = 25$  (Figs.2c and 2d) and  $N = 125$  (Figs. 2a and 2b). The Grampa regularization parameter  $\gamma$  is set to 0.2 as recommended in [26]. We present the average overlap (recovery fraction) for 25 Monte Carlo runs. The shaded area indicates where 90% of the data falls (excluding the top and bottom 5%).

|     | EMDGM( $N = 25$ ) | EMDGM( $N = 125$ ) | PGDGM( $N = 25$ ) | PGDGM( $N = 125$ ) | Grampa         |
|-----|-------------------|--------------------|-------------------|--------------------|----------------|
| CGW | 0.5822(0.0150)    | 2.5410(0.0202)     | 0.9587(0.0695)    | 4.2685(0.02800)    | 0.1210(0.0072) |
| CER | 0.5959(0.01621)   | 2.4803(0.0157)     | 0.9914(0.0723)    | 4.2382(0.0215)     | 0.1494(0.0082) |

**Table 2:** Average (over 25 Monte Carlos) running time in seconds (standard deviation in parentheses).

the similarity matrix  $\hat{X}$  satisfies the following diagonal dominance property with high probability (under different random graph model assumptions and assuming that  $X^* = \text{Id}$ ), which we recall is

$$\hat{X}_{ii} > \max_{j \neq i} \hat{X}_{ij}, \forall i \in [n].$$

Clearly, (3.8) implies that  $\text{GMWM}(\hat{X}) = \text{Id}$ . We compare (3.8) with the property (2.1), which is weaker than (3.8) and also sufficient to ensure  $\text{GMWM}(\hat{X}) = \text{Id}$  (as proven in Lemma 2 part (ii)). The objective is to see if for moderately small values of  $n$ , we see the diagonal dominance of the similarity matrix. The setup is as follows: we run EMDGM ( $N = 25$ ) and Grampa with  $n = 500$  using the CGW model and compute the following two metrics for the similarity matrix of each method.

- **Number of non-diagonally dominant rows.** We consider the quantity

$$|\{i \in [n] : \hat{X}_{ii} \leq \max_{j \neq i} \hat{X}_{ij}\}| \quad (4.8)$$

- **Number of pairs not satisfying (2.1).** In this case, we consider

$$|\{(i, j) \in [n]^2 : i \neq j \text{ and } \hat{X}_{ii} \vee \hat{X}_{jj} \leq \hat{X}_{ij}\}| \quad (4.9)$$

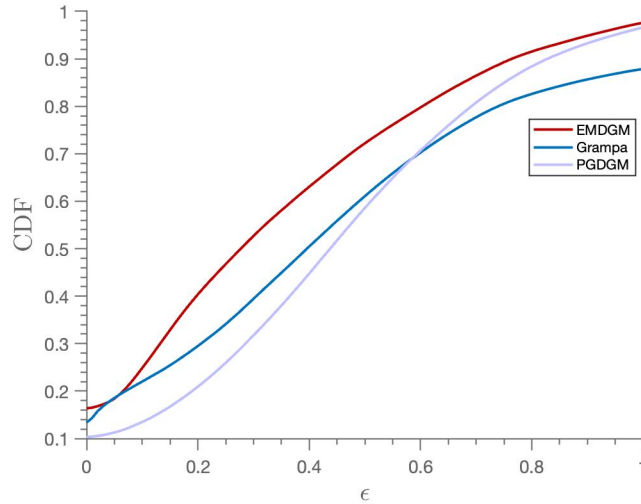
In Fig. 3 we show the heatmaps for the metrics (4.8) and (4.9), for Grampa and EMDGM algorithms. For both algorithms, we do not see diagonal dominance in any of the Monte Carlo samples, however in the low noise regime, both algorithms have perfect recovery (as seen in Fig. 2c). If one compares both algorithms by the metric (4.8), the conclusion would be that Grampa performs slightly better than EMDGM, however by the results in Fig. 2c, we know the opposite is true. On the other hand, we see in Figs. 3c and 3d that the metric (4.9) is 0 for algorithms for values of  $\sigma$  up to around 0.2, which coincides with the results in Fig. 2c. In addition, under this metric, EMDGM outperforms Grampa, which is also the conclusion of Section 4.1.2. Overall, in this experiment, metric (2.1) seems to be much better correlated with the perfect recovery cases in both methods.

## 4.2 Real data

This section focuses on evaluating the performance of the EMDGM algorithm on real data in comparison to PGDGM and Grampa. Before proceeding, it is important to note a few general remarks. Firstly, since some of the datasets are relatively large (approximately ten thousand vertices per graph) and sparse, we have omitted the standardization preprocessing discussed in Section 4.1 to more effectively leverage their sparsity. Secondly, unless specified otherwise, we employ dynamic step sizes derived from (4.5) for EMDGM and (4.7) for PGDGM. This choice is based on their performance with synthetic data. The Grampa regularizer  $\eta$  is fixed to 0.2.







**Figure 4:** Performance comparison of EMDGM, PGDGM, and Grampa on the SHREC'16 dataset, based on the error cumulative distribution function (CDF) defined in (4.10). We conducted 100 iterations each for EMDGM and PGDGM.

SHREC'16 dataset is that all images have varying sizes, resulting in inputs of the form  $A \in \mathbb{R}^{n_A \times n_A}$  and  $B \in \mathbb{R}^{n_B \times n_B}$  with  $n_A \neq n_B$ . It is worth noting that both EMDGM and PGDGM naturally extend to accommodate this variation, and the same can be said for Grampa (see [26, Section 4]). We will apply and compare the EMDGM, PGDGM, and Grampa algorithms using the Princeton benchmark protocol [39]. This benchmark protocol essentially involves a CDF comparison, defined by the following steps.

- **Normalized geodesic errors.** Given inputs  $A$  and  $B$ , and the output  $\hat{\pi}$  of a graph matching algorithm, we calculate the normalized geodesic error as follows: for each node  $i$  in  $A$ , we compute  $d_B(\hat{\pi}(i), \pi^*(i))$ , where  $\pi^*$  represents the ground truth matching, and  $d_B$  denotes the geodesic distance on  $B$  (computed as the weighted shortest path distance using the triangulation representation of the image [57]). The normalized error is defined as

$$\varepsilon(i) := d_B(\hat{\pi}(i), \pi^*(i)) / \sqrt{\text{Area}(B)},$$

where  $\text{Area}(B)$  is the surface area of  $B$  (computed using the triangulation representation).

- **CDF computation.** The error cumulative distribution function (CDF) is defined for  $t \geq 0$  as

$$\text{CDF}(t) = \sum_{i=1}^{n_A} \mathbb{1}_{\varepsilon(i) \leq t}, \quad (4.10)$$

where  $n_A$  is the number of nodes in  $A$ . Notice that  $\text{CDF}(0)$  corresponds to the recovery fraction, or overlap as defined in Section 4.1.

In Fig.4 we show the average performance of the three algorithms on the SHREC'16 dataset. We see that the performance of EMDGM is the best of the three, while for a small value of  $\epsilon$  (around 0.05), Grampa has a similar performance. It is interesting that for the other values of  $\epsilon$  mirror descent has a big gap in performance compared to Grampa, because in the graph matching problem we are not trying to explicitly optimize the errors beyond  $\epsilon = 0$  (which by the definition (4.10) is

related to exact recovery). In this case, **EMDGM** appears to introduce an additional level of regularity beyond what is specified in the optimization problem. We attribute the suboptimal performance of **PGDGM** to our difficulty in properly tuning the step-sizes.

### 4.3 Autonomous systems

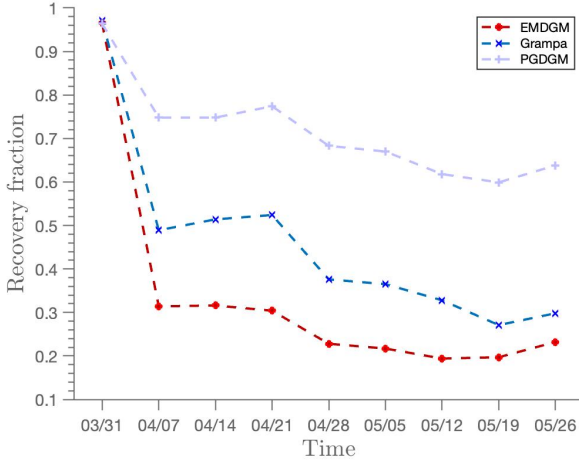
We use data from the Autonomous Systems dataset, which belongs to the University of Oregon Route Views Project [53], and is also available at the Stanford Network Analysis Project repository [44, 43]. This data has also been used to evaluate the performance of **Grampa** and other graph matching algorithms in [26, Section 4.5]. The dataset consists of a dynamically evolving network of autonomous systems (a collection of IP networks under a common routing policy) that exchange traffic through the Internet. There are a total of nine graphs, with each one representing weekly observations of the autonomous system from March 31, 2001, to May 26, 2001. The node count varies between 10,670 and 11,174, while the edge count fluctuates between 22,002 and 23,409 over time. Since the number of nodes and edges exhibits these fluctuations over time (see [44] for more summarizing statistics), we have implemented a subsampling strategy adopted in [26, Section 4.5]. We focus on a subset of 1,000 vertices with the highest degrees, found in all nine networks. The nine resulting graphs can be seen as perturbations of each other. Fixing the first graph (from March 31), as input  $A$ , and comparing it to the subsequent ones, we observe that the correlation approximately decreases over time (although is not fully monotone). This provides a comparison framework similar to the synthetic data experiments, where the correlation was controlled by the parameter  $\sigma$ .

In Figure 5 we compare the output of the three methods **EMDGM**, **Grampa**, **PGDGM** under two different metrics. The first is the recovery fraction, plotted in Figure 5a and the other is the rescaled objective  $\langle A, \widehat{X}_{\mathcal{P}}^T B \widehat{X}_{\mathcal{P}} \rangle_F$  (here  $\widehat{X}_{\mathcal{P}}$  represent the output of a graph matching algorithm), plotted in Figure 5b. As noted in [26], the latter metric is more instructive for this dataset due to the presence of numerous degree-one vertices connected to high-degree vertices, resulting in symmetries that render the ground truth unidentifiable. In this experiment, we apply all three graph matching algorithms to nine instances created by fixing input  $A$  as the first graph. Input  $B$  is obtained by random node relabeling for each of the nine graphs. In Figures 5c and 5d we plot the same metrics, but with standardized inputs. For that, we subtract to each matrix their average value and divide them by their standard deviation. Notably, the performance of **EMDGM** and **PGDGM** improves, while that of **Grampa** declines. This standardization alters the objective function and appears to introduce a ‘regularization’ effect for the methods on the simplex. Further investigation into this phenomenon is warranted, but we leave it for future research.

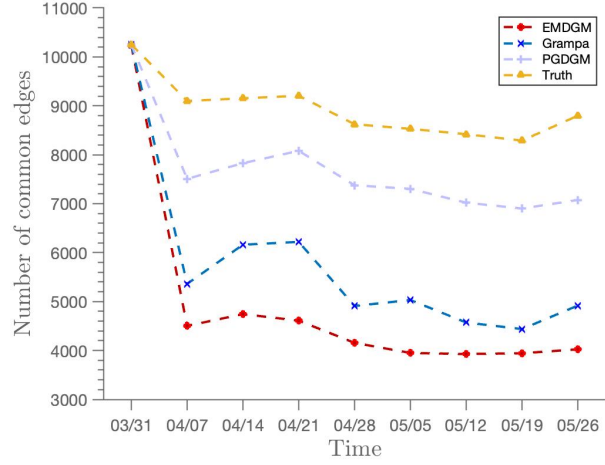
While we do not have a full explanation for why **PGDGM** outperforms **EMDGM** in this case, we can provide some partial insights. In certain scenarios, the “regularity” of the objective function may be more influential for achieving better results, whereas in other cases, the performance disparity may be primarily attributed to the selection of an appropriate step-size strategy. We believe that the latter plays a greater role in this case, based on the following experiment. We fix the step-size rule, using (4.1) for MD and (4.3), and define the efficiency ratio of the function  $E$  (as seen in [11, Example 9.17]) as:

$$\rho_E = \frac{\sqrt{\log n}, L_{E,\infty}}{L_{E,2}},$$

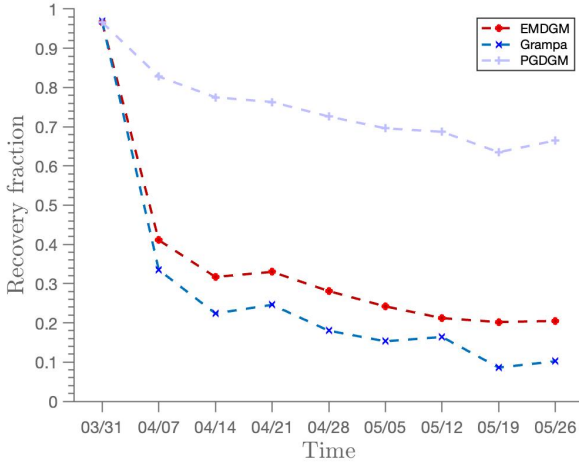
where it is clear that  $\frac{\sqrt{\log n}}{\sqrt{n}} \leq \rho_E \leq \sqrt{\log n}$ . When  $\rho_E$  is closer to the lower bound, MD tends to outperform PGD, while the reverse is true when  $\rho_E$  is closer to the upper bound. We compute



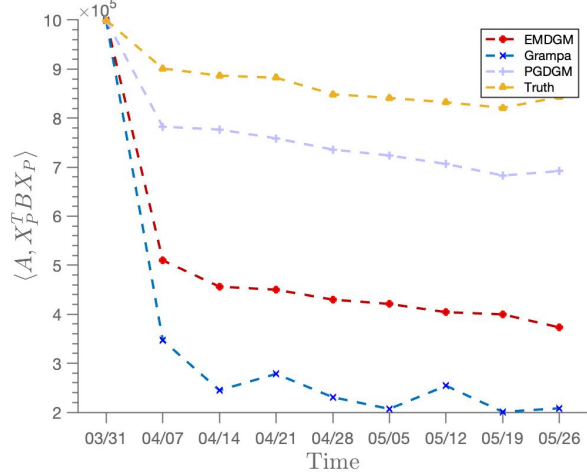
(a) Recovery fraction in a high-degree subsampled graph.



(b) Common edges in a high-degree subsampled graph.



(c) Recovery fraction in a high-degree standardized graph.



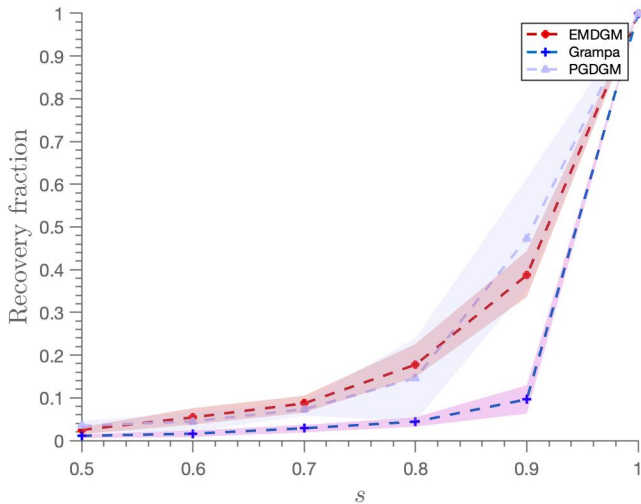
(d) Rescaled objective in a high-degree standardized graph.

**Figure 5:** Comparison of EMDGM, Grampa, and PGDGM using a high-degree subsampled graph with 1,000 vertices for each time instant. The input graph  $A$  remains fixed as the graph from March 31, while the input graph  $B$  varies (with random permutations of its rows and columns), as indicated on the x-axis. We used the dynamic step rule for EMDGM, derived from (4.5) and the heuristic rule (4.7) for PGDGM, with 125 iterations in both cases.

an empirical version of the unobserved quantities  $L_{E,\infty}$  and  $L_{E,2}$  (which can be regarded as regularity measures for  $E$ ), by taking 10,000 uniformly distributed samples on  $\Delta_{n^2}$  and computing the maximum of the  $\ell_\infty$  and  $\ell_2$  norms of the gradient evaluated on those points. The value of  $\rho_E$  obtained varies from 0.46 to 0.53, while  $\frac{\sqrt{\log n}}{\sqrt{n}} = 0.0831$  and  $\sqrt{\log n} = 2.62$ . Using this metric, there is not a clear advantage between the two methods. However, when applying rule (4.3), we observe that PGDGM significantly underperforms EMDGM (approximately ten times worse). It appears that the heuristic rule (4.7) plays a crucial role in accelerating the convergence of PGDGM. Unfortunately, we currently lack a similar rule for EMDGM, but this is an avenue for future exploration.

#### 4.4 Facebook networks

We use Facebook’s friendship network data from [62], consisting of 11,621 individuals affiliated with Stanford University. In contrast to the previous examples, where multiple networks were available, here, we have a single large network. As a result, we need to employ a subsampling strategy to generate correlated graphs for evaluating the graph matching methods. In the context of seeded graph matching, [66] employed a specific sampling strategy to evaluate methods based on seed quality. However, since our objective is different, our sampling strategy also differs from theirs. The complete dataset contains 136,660 (unweighted) edges, with a degree distribution closely resembling a power-law pattern (density proportional to  $d^{-1}$ ), and an average degree of approximately 100. We obtain two correlated graphs  $A, B$  as follows. First, we create graph  $H$  by selecting 1,000 vertices uniformly at random, forming the induced subgraph. From the parent graph  $H$ , we retain each edge with a probability of  $s$  to create graph  $A$ . We independently repeat this process to generate graph  $B$  (as usual, we shuffle the labels of  $B$  uniformly before applying the graph matching algorithms). Intuitively, a higher value of  $s$  corresponds to a stronger correlation between the graphs, with  $s = 1$  implying isomorphism. Notice that our sampling scheme closely resembles the “parent graph” definition in [55] for the CER model.



**Figure 6:** Comparison of graph matching algorithms on subsamples of 1,000 edges of a Facebook friendship network. We plot the average over 15 Monte Carlo (over the uniform subsamples). We used the dynamic step rule for EMDGM, derived from (4.5) and the heuristic rule (4.7) for PGDGM, with 125 iterations in both cases.

In Figure 6 we plot the recovery fraction for EMDGM, PGDGM and Grampa. As expected, all three methods achieve exact recovery in the isomorphic case  $s = 1$ . However, for smaller values of  $s$ , there is a significant gap in performance between EMDGM and PGDGM (which are comparable) and Grampa. The more pronounced drop in performance, compared to the synthetic data experiments presented in Section 4.1, is also expected given the power-law behavior exhibited by this network. An interesting observation can be made from the confidence intervals in Figure 6: EMDGM shows less variance across the Monte Carlo subsamples. To quantify this, we computed the average standard deviation over the 6 instances of  $s$ . For EMDGM, this value is 0.0126, while for PGDGM, it is significantly higher at 0.0604.

## 5 Concluding remarks

In this work, we present a novel graph matching algorithm based on a convex relaxation of the permutation set to the unit simplex. We introduce an iterative mirror descent scheme that offers closed-form expressions for the iterates, resulting in enhanced efficiency compared to competing projecting gradient methods. Theoretical analysis demonstrates that, when coupled with a standard greedy rounding procedure, our method can exactly recover the ground truth matching in the noiseless case of the correlated Gaussian Wigner model. Experimental results support this claim, showing that our proposed algorithm can achieve exact recovery even in the presence of significant noise. In terms of statistical performance, it consistently outperforms other convex relaxation-based methods with similar time complexity.

The following are potential avenues for future research.

- An intriguing theoretical question is to extend the guarantees to the case where  $\sigma \neq 0$  in the CGW model. This presents two main challenges. First, understanding the eigenvector structure of the iterates in (2.4) is nontrivial, given its Hadamard product definition and the complexity of the gradient expression. Second, using Lemma 2 becomes less straightforward when not all pairs satisfy the condition (2.1). To address this, one possible approach is to demonstrate that the number of pairs not satisfying (2.1), as given in (4.9), converges to zero. However, this seems to present significant technical challenges.
- Extending recovery guarantees to other models is also an important theoretical objective. A common approach in the literature of iterative methods involves using uniform concentration bounds, which is essentially a worst-case scenario approach. To follow this strategy, it will be crucial to gain a deeper understanding of the structure of the iterates in (2.4) beyond  $X^{(1)}$ . In particular, we need to improve our ability to “localize”  $X^{(k)}$ . Otherwise, there is a risk that, in the worst-case scenarios, the gradient may point in the wrong direction. It is not hard to provide examples of such scenarios.
- From an experimental perspective, it is intriguing to explore acceleration in the mirror descent dynamic. Although accelerated MD schemes have been considered in the literature, as discussed in [41], they involve Bregman projections that are fast in theory but lack a closed form, making their implementation more involved. It would be interesting to assess the practical performance of these schemes.
- Another experimental question involves finding a rate selection method for EMD that enhances its statistical performance, similar to the heuristic (4.7) in the case of PGD. In general, any other techniques used to improve the performance of first-order methods can also be tested here, including adaptive restart, early stopping, and more.

## References

- [1] Yonathan Aflalo, Alexander Bronstein, and Ron Kimmel. On convex relaxation of graph isomorphism. *Proceedings of the National Academy of Sciences*, 112(10):2942–2947, 2015.
- [2] Zeyuan Allen-Zhu, Zhenyu Liao, and Lorenzo Orecchia. Spectral sparsification and regret minimization beyond matrix multiplicative updates. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '15, page 237–245, New York, NY, USA, 2015. Association for Computing Machinery.
- [3] David G. Anderson, Ming Gu, and Christopher Melgaard. An efficient algorithm for unweighted spectral graph sparsification, 2014.
- [4] Greg W. Anderson, Alice Guionnet, and Ofer Zeitouni. *An Introduction to Random Matrices*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2009.
- [5] Ernesto Araya, Guillaume Braun, and Hemant. Tyagi. Seeded graph matching for the correlated Wigner model via the projected power method. *arXiv:2204.04099*, 2022.
- [6] S. Arora, E. Hazan, and S. Kale. Fast algorithms for approximate semidefinite programming using the multiplicative weights update method. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 339–348, 2005.
- [7] S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.
- [8] Sanjeev Arora and Satyen Kale. A combinatorial, primal-dual approach to semidefinite programs. *J. ACM*, 63(2), may 2016.
- [9] László Babai. Group, graphs, algorithms: the graph isomorphism problem. In *Proceedings of the International Congress of Mathematicians: Rio de Janeiro 2018*, pages 3319–3336. World Scientific, 2018.
- [10] Laszlo Babai, Paul Erdos, and Stanley Selkow. Random graph isomorphism. *SIAM Journal on Computing*, 9(3):628–635, 1980.
- [11] Amir Beck. *First Order Methods in Optimization*. Society for Industrial and Applied Mathematics, 2017.
- [12] Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.
- [13] Jeremy Bernstein, Jiawei Zhao, Markus Meister, Ming-Yu Liu, Anima Anandkumar, and Yisong Yue. Learning compositional functions via multiplicative weight updates. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc.
- [14] Béla Bollobás. The distribution of the maximum degree of a random graph. *Discrete Mathematics*, 32(2):201–203, 1980.
- [15] Béla Bollobás. Degree sequences of random graphs. *Discrete Mathematics*, 33(1):1–19, 1981.
- [16] Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Found. Trends Mach. Learn.*, 8(3–4):231–357, nov 2015.

- [17] Rainer Ernst Burkard, Eranda Dragoti-Cela, P.M. Pardalos, and L.S. Pitsoulis. *The quadratic assignment problem*, volume 2, pages 241–337. Kluwer Academic Publishers, Netherlands, 1998.
- [18] Dajana Conte, Pasquale Foggia, Carlo Sansone, and Mario Vento. Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(03):265–298, 2004.
- [19] Daniel Cullina and Negar Kiyavash. Exact alignment recovery for correlated Erdős-Rényi graphs. *arXiv:1711.06783*, 2017.
- [20] Tomek Czajka and Gopal Pandurangan. Improved random graph isomorphism. *Journal of Discrete Algorithms*, 6(1):85–92, 2008.
- [21] Constantinos Daskalakis, Alan Deckelbaum, and Anthony Kim. Near-optimal no-regret algorithms for zero-sum games. *Games and Economic Behavior*, 92:327–348, 2015.
- [22] Chris Ding, Tao Li, and Michael I. Jordan. Nonnegative matrix factorization for combinatorial optimization: Spectral clustering, graph matching, and clique finding. In *2008 Eighth IEEE International Conference on Data Mining*, pages 183–192, 2008.
- [23] Jian Ding, Zongming Ma, Yihong Wu, and Jiaming Xu. Efficient random graph matching via degree profiles. *Probability Theory and Related Fields*, 179:29–115, 2021.
- [24] Frank Emmert-Streib, Matthias Dehmer, and Yongtang Shi. Fifty years of graph matching, network alignment and network comparison. *Inf. Sci.*, 346(C):180–197, jun 2016.
- [25] Zhou Fan, Cheng Mao, Yihong Wu, and Jiaming Xu. Spectral graph matching and regularized quadratic relaxations II: Erdős-Rényi graphs and universality. *arXiv:1907.08883*, 2019.
- [26] Zhou Fan, Cheng Mao, Yihong Wu, and Jiaming Xu. Spectral graph matching and the quadratic relaxation I: the Gaussian model. *arXiv: 1907.08880v1*, 2019.
- [27] Soheil Feizi, Gerald T. Quon, Mariana Recamonde-Mendoza, Muriel Médard, Manolis Kellis, and Ali Jadbabaie. Spectral alignment of graphs. *IEEE Transactions on Network Science and Engineering*, 7:1182–1197, 2020.
- [28] Luca Ganassali. Sharp threshold for alignment of graph databases with gaussian weights. In Joan Bruna, Jan Hesthaven, and Lenka Zdeborova, editors, *Proceedings of the 2nd Mathematical and Scientific Machine Learning Conference*, volume 145 of *Proceedings of Machine Learning Research*, pages 314–335. PMLR, 16–19 Aug 2022.
- [29] Luca Ganassali, Marc Lelarge, and Laurent Massoulié. Spectral alignment of correlated gaussian matrices. *Advances in Applied Probability*, 54(1):279–310, 2022.
- [30] Luca Ganassali and Laurent Massoulié. From tree matching to sparse graph alignment. In Jacob D. Abernethy and Shivani Agarwal, editors, *Conference on Learning Theory, COLT 2020, 9-12 July 2020, Virtual Event [Graz, Austria]*, volume 125 of *Proceedings of Machine Learning Research*, pages 1633–1665. PMLR, 2020.
- [31] Luca Ganassali, Laurent Massoulié, and Marc Lelarge. Impossibility of partial recovery in the graph alignment problem. In Mikhail Belkin and Samory Kpotufe, editors, *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pages 2080–2102. PMLR, 15–19 Aug 2021.

- [32] Luca Ganassali, Laurent Massoulié, and Guilhem Semerjian. Statistical limits of correlation detection in trees, 2022.
- [33] W. Zhou H. Sun and M. Fei. A survey on graph matching in computer vision. *2020 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pages 225–230, 2020.
- [34] Georgina Hall and Laurent Massoulié. Partial recovery in the graph alignment problem. *Operations Research*, Aug 2022.
- [35] E. Hazan. Introduction to online convex optimization. *Foundations and Trends R© in Optimization*, 2(3-4):157–325, 2015.
- [36] Roger Horn and Charles Johnson. *Topics in matrix analysis*. Cambridge University Press, 1991.
- [37] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 2 edition, 2012.
- [38] Bo Jiang, Jin Tang, Chris Ding, Yihong Gong, and Bin Luo. Graph matching via multiplicative update algorithm. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17*, page 3190–3198, Red Hook, NY, USA, 2017. Curran Associates Inc.
- [39] Vladimir G. Kim, Yaron Lipman, and Thomas Funkhouser. Blended intrinsic maps. *ACM Trans. Graph.*, 30(4), jul 2011.
- [40] J. Kivinen and M.K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, pages 1–63, 1997.
- [41] Walid Krichene, Alexandre Bayen, and Peter L Bartlett. Accelerated mirror descent in continuous and discrete time. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.
- [42] François Le Gall. Algebraic complexity theory and matrix multiplication. In *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, page 23, 2014.
- [43] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: Densification laws, shrinking diameters and possible explanations. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, KDD ’05*, page 177–187, New York, NY, USA, 2005. Association for Computing Machinery.
- [44] Jure Leskovec and Andrej Krevl. Snap datasets: Stanford large network dataset collection.
- [45] Joseph Lubars and R. Srikant. Correcting the output of approximate graph matching algorithms. *IEEE conference on Computer Communications*, pages 1745–1753, 2018.
- [46] Vince Lyzinski, Donniell E. Fishkind, Marcelo Fiori, Joshua T. Vogelstein, Carey E. Priebe, and Guillermo Sapiro. Graph matching: relax at your own risk. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):60–73, 2016.



- [47] Zorah Löhner, Emanuele Rodolà, Michael M. Bronstein, Daniel Cremers, Oliver Burghard, Luca Cosmo, Andreas Dieckmann, Reinhard Klein, and Yusuf Sahillioglu. Matching of Deformable Shapes with Topological Noise. In *Eurographics Workshop on 3D Object Retrieval*, 2016.
- [48] Cheng Mao, Mark Rudelson, and Konstantin Tikhomirov. Exact matching of random graphs with constant correlation. *arXiv:2110.05000*, 2021.
- [49] Cheng Mao, Yihong Wu, Jiaming Xu, and Sophie H. Yu. Testing network correlation efficiently via counting trees, 2022.
- [50] Cheng Mao, Yihong Wu, Jiaming Xu, and Sophie H. Yu. Random graph matching at otter’s threshold via counting chandeliers. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, page 1345–1356, New York, NY, USA, 2023. Association for Computing Machinery.
- [51] A. Narayanan and V. Shmatikov. De-anonymizing social networks. *2009 30th IEEE Symposium on Security and Privacy*, pages 173–187, 2009.
- [52] A. S. Nemirovsky and D. B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley-Interscience, New York, 1983.
- [53] University of Oregon Route Views Project. Autonomous systems peering networks.
- [54] Negar Kiyavash Osman Emre Dai, Daniel Cullina and Matthias Grossglauser. Analysis of a canonical labeling algorithm for the alignment of correlated erdős-rényi graphs. *arXiv:1804.09758v2*, 2019.
- [55] 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*, pages 1235–1243, 2011.
- [56] Marcello Pelillo. Replicator equations, maximal cliques, and graph isomorphism. *Neural Computation*, 11(8):1933–1955, 1999.
- [57] Gabriel Peyré. Numerical mesh processing, 2008.
- [58] Robert Reams. Hadamard inverses, square roots and products of almost semidefinite matrices. *Linear Algebra and its Applications*, 288:35–43, 1999.
- [59] Sartaj Sahni and Teofilo Gonzalez. P-complete approximation problems. *J. ACM*, 23(3):555–565, jul 1976.
- [60] 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, 2009.
- [61] Terence Tao and Van Vu. Random matrices have simple spectrum. *Combinatorica*, 37(3):539–553, Jun 2017.
- [62] Amanda L. Traud, Peter J. Mucha, and Mason A. Porter. Social structure of facebook networks. *Physica A: Statistical Mechanics and its Applications*, 391(16):4165–4180, 2012.

- [63] Shinji Umeyama. An eigendecomposition approach to weighted graph matching problems. *IEEE transactions on pattern analysis and machine intelligence*, 10(5):695–703, 1988.
- [64] Haoyu Wang, Yihong Wu, Jiaming Xu, and Israel Yolou. Random graph matching in geometric models: the case of complete graphs. *arXiv:2202.10662*, 2022.
- [65] Yihong Wu, Jiaming Xu, and H Yu Sophie. Settling the sharp reconstruction thresholds of random graph matching. *2021 IEEE International Symposium on Information Theory*, pages 2714–2719, 2021.
- [66] Liren Yu, Jiaming Xu, and Xiaojun Lin. Graph matching with partially-correct seeds. *Journal of Machine Learning Research*, 22(2021):1–54, 2021.
- [67] Mikhail Zaslavskiy, Francis Bach, and Jean-Philippe Vert. Global alignment of protein–protein interaction networks by graph matching methods. *Bioinformatics*, 25(12):1259–1267, 2009.
- [68] Mikhail Zaslavskiy, Francis Bach, and Jean-Philippe Vert. A path following algorithm for the graph matching problem. *IEEE transactions on pattern analysis and machine intelligence*, 31(12):60–73, 2009.

## A The population dynamics

We introduce and analyze the population (or average) version of the mirror descent dynamics. For that, notice that all randomness in the dynamics defined by (2.4) is given by the input matrices  $A, B$ . In addition, the dependence on these matrices in the iterative process is captured by the term  $\nabla E(X) = A^2X + XB^2 - 2AXB$  (as it is clear from line 3 of Algorithm 2). We define the *population dynamics*, by replacing the (random) loss function  $E(X)$  by the (deterministic) *population loss*, defined as  $\mathbb{E}_{A,B}E(X)$ . This is equivalent to replace the term  $\nabla E(X)$  in (2.4) by its expectation  $\overline{\nabla E}(X) := \mathbb{E}_{A,B}\nabla E(X)$ , which is proven as part of Lemma 10. More formally, this dynamic, which is deterministic, is defined given a sequence  $(\gamma_k)_{k=1}^\infty$  of positive rates, by the recursion

$$X^{(0)} = J/n^2, \tag{A.1}$$

$$X^{(k+1)} = n \cdot \frac{X^{(k)} \odot \exp_\odot(-\gamma_k \overline{\nabla E}(X^{(k)}))}{\|X^{(k)} \odot \exp_\odot(-\gamma_k \overline{\nabla E}(X^{(k)}))\|_{1,1}}. \tag{A.2}$$

For each  $k$ , we define the normalization factor by

$$\overline{N}_k := \frac{1}{n} \|X^{(k)} \odot \exp_\odot(-\gamma_k \overline{\nabla E}(X^{(k)}))\|_{1,1},$$

which is a positive real number.

### A.1 One iteration ( $N = 1$ )

The following result shows that Algorithm 2 succeeds in finding the ground truth  $X^*$  in one step (i.e., when  $N = 1$ ). We will assume without loss of generality that  $X^* = \text{Id}$ , and consequently in the definition of the population gradient, we have  $\overline{\nabla E}(X) = \mathbb{E}_{A,B}\nabla E(X)$ , where the joint law of  $A, B$  is  $\mathcal{W}(n, \sigma, X^* = \text{Id})$ . The following result shows that in the case of the population dynamic, one step is sufficient for recovering the ground truth if we applied Algorithm 1 directly after the first iteration.

**Proposition 1.** *Let  $X^{(1)} \in \mathbb{R}^{n \times n}$  be the first iterate in the dynamic defined by (A.1) and (A.2), defined by  $\gamma_0 \in \mathbb{R}_+$ . Then  $\text{GMWM}(X^{(1)}) = \text{Id}$ , for any  $\gamma_0 > 0$ .*

*Proof.* From (A.1) and (A.2), we have

$$X^{(1)} = \overline{N}_0^{-1} X^{(0)} \odot \exp_\odot(-\gamma_0 \overline{\nabla E}(X^{(0)})),$$

where  $\overline{N}_0 > 0$  is the normalization term. Using the definition for the population gradient, Lemma 10 and by a simple calculation, we get

$$\begin{aligned} X^{(1)} &= \overline{N}_0^{-1} J \odot \exp_\odot\left(\frac{-\gamma_0}{n^2} \left(2 + \frac{n+1}{n} \sigma^2\right) J - 2 \text{Id}\right) \\ &= \overline{N}_0^{-1} J \odot \exp_\odot\left(\frac{-\gamma_0}{n^2} \left(2 + \frac{n+1}{n} \sigma^2\right) J\right) \odot \exp_\odot\left(\frac{2\gamma_0}{n^2} \text{Id}\right) \\ &= \overline{N}'_0 \exp_\odot\left(\frac{2\gamma_0}{n^2} \text{Id}\right), \end{aligned}$$

where we define  $\overline{N}'_0 := \frac{\overline{N}_0^{-1}}{n} e^{-\frac{\gamma_0}{n} (2 + \frac{n+1}{n} \sigma^2)}$ . Given that  $e^{\frac{2\gamma_0}{n^2}} > 1$  for  $\gamma_0 > 0$ , we deduce that  $X^{(1)}$  satisfies (3.8), which implies that  $\text{GMWM}(X^{(1)}) = \text{Id}$ .  $\square$

## A.2 Multiple iterations ( $N > 1$ )

In practice, we do not have a priori information on how many iterations we need to perform before the rounding step, but we expect Algorithm 2 to be relatively robust with respect to the choice of the number of iterations  $N$  (at least in some range), for some choice of the step size sequence  $(\gamma_k)_{k=1}^N$ . The next result proves that this is the case for the population dynamic, in the sense that all its iterates can be rounded to the ground truth, provided that a certain condition on the rate sequence  $(\gamma_k)_{k \geq 0}$  holds.

**Theorem 4.** *Consider  $X^{(0)}, X^{(1)}, \dots, X^{(N)} \in \mathbb{R}^{n \times n}$  the iterates of the dynamic defined by (A.1) and (A.2), defined by the rate sequence  $(\gamma_k)_{k=0}^{N-1} \in \mathbb{R}_+$ . If  $\sigma \leq 1$  and it holds*

$$\sum_{k=0}^{N-1} \gamma_k < \frac{n-1}{4} \log 2, \quad (\text{A.3})$$

then  $\text{GMWM}(X^{(k)}) = \text{Id}$ , for all  $1 \leq k \leq N$ .

Before proving this theorem, we will prove two lemmas that characterize the structure of iterates in the population dynamic.

**Lemma 7.** *Let  $(X^{(k)})_{k=0}^\infty$  be a trajectory of the population dynamic given by some step size sequence  $(\gamma_k)_{k=0}^\infty \in \mathbb{R}_+$ . We have that  $X^{(k)}$  is symmetric for all  $k$  and that*

$$X_{ij}^{(k)} = \begin{cases} X_{11}^{(k)} & \text{if } i = j \\ X_{12}^{(k)} & \text{if } i \neq j \end{cases} \quad (\text{A.4})$$

*Proof.* The proof is by induction. Clearly  $X^{(0)}$  is symmetric. Assume that  $X^{(k)}$  is symmetric. Notice that

$$X^{(k+1)} = \overline{N}_k^{-1} X^{(k)} \odot \exp_{\odot} \left( -\gamma_k \overline{\nabla E}(X^{(k)}) \right), \quad (\text{A.5})$$

for a constant  $\overline{N}_k^{-1} > 0$ . The proof follows by the inductive hypothesis, by the fact that  $\overline{\nabla E}(X)$  is symmetric if  $X$  is symmetric (see (D.5)) and the fact that the Hadamard product of two symmetric matrices is symmetric. We also prove (A.4) by induction. The base case  $k = 0$  is evident. We assume that  $X^{(k)}$  satisfies the property (A.4). Using (A.5), the fact that  $\overline{\nabla E}(X)$  satisfies (A.4) if  $X$  does, and the fact that (A.4) is closed under Hadamard products, the conclusion follows.  $\square$

The previous lemma asserts that each entry of  $X^{(k)}$  can take only two values, depending on its location inside the matrix (diagonal or off-diagonal). In this case, it is easy to see that the greedy rounding of  $X^{(k)}$ , for any  $k$ , will be equal to the identity (the ground truth) if and only if  $X_{11}^{(k)} > X_{12}^{(k)}$ , which is in fact equivalent to say that  $X^{(k)}$  is diagonally dominant, i.e. for all  $i \in [n]$  we have  $X_{ii}^{(k)} > X_{ij}^{(k)}$ , for all  $j \neq i$ . Since the iterates of the population dynamics  $X^{(k)}$ , up to the step  $K$  (for a given  $K \in \mathbb{N}$ ), are completely determined by the sequence of learning rates  $(\gamma_k)_{k=0}^K$ , we can characterize them as a function of them. To prove the diagonal dominance, it will be convenient to define the ratio between the off-diagonal and diagonal entries, for  $k \geq 1$ , as follows,

$$r_k(\gamma_0, \dots, \gamma_{k-1}) := \frac{X_{12}^{(k)}}{X_{11}^{(k)}}.$$

Note that, in the previous definition,  $r_k$  is a function from  $\mathbb{R}^k$  to  $\mathbb{R}$ . The following lemma provides a recursive characterization for these ratios.

**Lemma 8.** For any given sequence  $(\gamma_k)_{k=0}^\infty$ , the ratio between off-diagonal and diagonal entries  $r_k$  defined above, satisfy the following

$$\begin{aligned} r_k(\gamma_0, \dots, \gamma_{k-1}) &= r_{k-1}(\gamma_0, \dots, \gamma_{k-2}) e^{-\gamma_{k-1} \frac{a_\sigma r_{k-1}(\gamma_0, \dots, \gamma_{k-2}) - (a_\sigma - 2)}{n(n-1)r_{k-1}(\gamma_0, \dots, \gamma_{k-2}) + n}}, \text{ for } k \geq 2 \\ r_1(\gamma_0) &= e^{-2\gamma_0/n^2}, \end{aligned}$$

where  $a_\sigma := 2 + \frac{n+1}{n}\sigma^2$ .

*Proof.* To save notation, we will write  $r_k$  instead of  $r_k(\gamma_0, \dots, \gamma_{k-1})$ . Using Lemma 7 and (D.5) we get that

$$\overline{\nabla E}(X^{(k)})_{12} - \overline{\nabla E}(X^{(k)})_{11} = a_\sigma X_{12}^{(k)} - (a_\sigma - 2)X_{11}^{(k)}.$$

From this and the definition of the population dynamics, we see that

$$\begin{aligned} r_k &= r_{k-1} e^{-\gamma_{k-1}(\overline{\nabla E}(X^{(k)})_{12} - \overline{\nabla E}(X^{(k)})_{11})} \\ &= r_{k-1} e^{-\gamma_{k-1}(a_\sigma X_{12}^{(k-1)} - (a_\sigma - 2)X_{11}^{(k-1)})}, \end{aligned} \tag{A.6}$$

for  $k \geq 2$ . Similarly,

$$\begin{aligned} r_1 &= \frac{X_{12}^{(0)}}{X_{11}^{(0)}} e^{-\gamma_0(a_\sigma X_{12}^{(0)} - (a_\sigma - 2)X_{11}^{(0)})} \\ &= e^{-\frac{\gamma_0}{n^2}(a_\sigma - (a_\sigma - 2))} \\ &= e^{-2\frac{\gamma_0}{n^2}}. \end{aligned}$$

On the other hand, given the fact that  $X^{(l)} \in \Delta_{n^2}$  we have, for any  $l \in \mathbb{N}$ ,

$$nX_{11}^{(l)} + n(n-1)X_{12}^{(l)} = 1,$$

which together with the fact  $X_{12}^{(k-1)} = r_{k-1}X_{11}^{(k-1)}$ , imply the following,

$$\begin{aligned} X_{12}^{(k-1)} &= \frac{r_{k-1}}{n(n-1)r_{k-1} + n} \\ X_{11}^{(k-1)} &= \frac{1}{n(n-1)r_{k-1} + n}. \end{aligned}$$

Plugging this on (A.6) finishes the proof.  $\square$

*Proof of Thm.4.* Given Lemma 7, it suffices to compare  $X_{11}^{(k)}$  and  $X_{12}^{(k)}$  for each  $k > 1$ . Indeed, if we prove for a given  $k$  that  $X_{12}^{(k)} < X_{11}^{(k)}$  or, equivalently, that  $r_k < 1$ , then  $X^{(k)}$  will satisfy the diagonal dominance property on all of its rows, thus we will get  $\text{GMWM}(X^{(k)}) = \text{Id}$ . We now fix a  $k \in [N-1]$ . From Lemma 8, it is easy to see that

$$r_k = r_1 \prod_{j=1}^{k-1} e^{-\gamma_j \frac{a_\sigma r_j - a_\sigma + 2}{n(n-1)r_j + n}} \tag{A.7}$$

On the other hand, notice that  $a_\sigma < 4$  (because  $\sigma \leq 1$ ), from which we deduce that

$$\frac{a_\sigma r_j - a_\sigma + 2}{n(n-1)r_j + n} \leq \frac{a_\sigma r_j}{n(n-1)r_j + n} < \frac{4}{n(n-1)},$$

for all  $j \leq k - 1$ . This in turn implies that

$$r_k \geq e^{-2\gamma_0/n^2} \prod_{j=1}^{k-1} e^{-4\gamma_j/n(n-1)} \geq e^{-\frac{4}{n(n-1)} \sum_{j=0}^{k-1} \gamma_j}.$$

Given that the sequence of rates satisfies

$$\frac{n(n-1)}{4} \log 2 > \sum_{j=0}^{N-1} \gamma_j \geq \sum_{j=0}^{k-1} \gamma_j,$$

we have that

$$r_k > \frac{1}{2} \geq \frac{a_\sigma - 2}{a_\sigma},$$

where the last inequality follows from the fact that  $\sigma \leq 1$ . Noticing that  $k$  was arbitrary in  $[N - 1]$ , we conclude that  $r_j > \frac{a_\sigma - 2}{a_\sigma}$ , for all  $j \in [N - 1]$ . From this, and from the fact that  $\gamma_j > 0$ , we deduce that for all  $2 \leq j \leq N - 1$  we have that

$$e^{-\gamma_j \frac{a_\sigma r_j - a_\sigma + 2}{n(n-1)r_j + n}} < 1,$$

which combined with (A.7) gives that  $r_k < 1$ , for  $2 \leq k \leq N$  (the fact that  $r_1 < 1$  is evident).  $\square$

## B Proof of Lemma 1

Notice that for any  $\Pi \in \mathcal{P}_n$  and  $X \in \mathbb{R}^{n \times n}$  we have

$$\|AX - X\Pi B\Pi^T\|_F^2 = \|AX\Pi - X\Pi B\|_F^2,$$

by the unitary invariance of the Frobenius norm. Then, given the permutation invariance of the set  $\mathcal{K}$ , if  $\hat{X} \in \mathcal{S}(A, B)$ , then  $\hat{X}\Pi^T \in \mathcal{S}(A, \Pi B\Pi^T)$ . It remains to show that if  $\hat{X}_{\mathcal{P}}$  is the output of the algorithm described in the lemma with input  $A, B$ , then the  $\hat{X}_{\mathcal{P}}\Pi^T$  will its output for the inputs  $A, \Pi B\Pi^T$ . For this is sufficient to notice that (given the mechanism of Algorithm 1) that if the largest element of  $\hat{X}$  is the entry  $(i, j)$  then the largest of  $\hat{X}\Pi^T$  is the entry  $(i, \pi(j))$ , where  $\pi$  is the permutation map associated to  $\Pi$ .

## C Proof of Lemma 4

Note that for all  $i \in [n]$ , we have  $v_i$  is marginally distributed uniformly on  $\mathbb{S}^{n-1}$ . This is equivalent to  $v_i \stackrel{dist}{=} \frac{g}{\|g\|_2}$ , where  $g$  is a standard normal vector in  $\mathbb{R}^n$ . It is easy to see that  $\text{sign}(v_i(k)) \stackrel{dist}{=} \text{sign}(g(k))$ , and the variables  $\text{sign}(g(1)), \dots, \text{sign}(g(n))$  are i.i.d Rademacher distributed. Then, for any given  $i \in [n]$ , we have

$$\mathbb{P}(\forall k, v_i(k) > 0) = \frac{1}{2^n}.$$

The result follows by applying the union bound.

## D Useful calculations

We gather here some useful calculations, used at different parts of the analysis.

**Lemma 9.** *Let  $A$  be a  $GOE(n)$  distributed random matrix, then*

$$\mathbb{E}(A^2) = \frac{n+1}{n} \text{Id} \quad (\text{D.1})$$

$$\mathbb{E}(A \otimes A) = \frac{1}{n} \text{vec}(\text{Id}) \text{vec}(\text{Id})^T + \frac{1}{n} T, \quad (\text{D.2})$$

where  $T$  is a  $n^2 \times n^2$  matrix that represents the transposition operator defined on  $\mathbb{R}^{n \times n}$ .

*Proof.* Take  $i, j \in [n]$ . We have

$$\begin{aligned} \mathbb{E}(A^2)_{ij} &= \sum_{k=1}^n \mathbb{E}(A_{ik} A_{kj}) \\ &= \sum_{k=1}^n \frac{1}{n} \delta_{ij} (1 + \delta_{ki}), \end{aligned}$$

from which we get (D.1). To prove (D.2), we take any matrix  $X \in \mathbb{R}^{n \times n}$  and compute

$$\begin{aligned} \mathbb{E}(AXA)_{ij} &= \sum_{k,k'=1}^n \mathbb{E}(A_{ik} X_{kk'} A_{k'j}) \\ &= \sum_{k=1}^n \frac{1}{n} X_{kk} \delta_{ij} (1 + \delta_{ki}) + \frac{1}{n} X_{ji} \\ &= \frac{1}{n} \delta_{ij} (\text{Tr}(X) + X_{ii}) + \frac{1}{n} (1 - \delta_{ij}) X_{ji}, \end{aligned}$$

and from this we deduce that

$$\begin{aligned} \mathbb{E}(AXA) &= \frac{1}{n} (\text{Tr}(X) \text{Id} + X^T), \\ &= \frac{1}{n} (\langle X, \text{Id} \rangle_F \text{Id} + X^T). \end{aligned}$$

Given that  $(A \otimes A) \text{vec}(X) = AXA$ , it is clear that the previous implies the expression (D.2).  $\square$

**Lemma 10.** *Let  $A, B \sim \mathcal{W}(n, \sigma, X^* = \text{Id})$ . Recall the definition  $E(X) = \|AX - XB\|_F^2$  or, in vector notation,  $E(\text{vec}(X)) = \text{vec}(X)^T H \text{vec}(X)$ , where  $H = (\text{Id} \otimes A - B \otimes \text{Id})^2$ . We have*

$$\nabla_{\mathbb{E}_{A,B}} E(\text{vec}(X)) = \mathbb{E}_{A,B} \nabla E(\text{vec}(X)) = \mathbb{E}(H) \text{vec}(X) \quad (\text{D.3})$$

and

$$\mathbb{E}(H) = (2 + \sigma^2) \left( \frac{n+1}{n} \right) (\text{Id} \otimes \text{Id}) - \frac{2}{n} (\text{vec}(\text{Id}) \text{vec}(\text{Id})^T + T), \quad (\text{D.4})$$

where  $T \in \mathbb{R}^{n^2 \times n^2}$  is the matrix representing the transpose transformation in  $\mathbb{R}^{n \times n}$ . In particular, we have (in matrix language)

$$\mathbb{E}_{A,B} \nabla E(X) = (2 + \sigma^2) \left( \frac{n+1}{n} \right) X - \frac{2}{n} (\text{Tr}(X) \text{Id} + X^T) \quad (\text{D.5})$$

*Proof.* Both equalities in (D.3) follow from  $\mathbb{E}_{A,B}(E(\text{vec}(X))) = \text{vec}(X)^T \mathbb{E}_{A,B}(H) \text{vec}(X)$  and linearity of expectation. On the other hand, a simple calculation shows that

$$\begin{aligned} \mathbb{E}(H) &= \mathbb{E}(\text{Id} \otimes A^2) + \mathbb{E}(B^2 \otimes \text{Id}) - 2\mathbb{E}(B \otimes A) \\ &= \mathbb{E}(\text{Id} \otimes A^2) + \mathbb{E}(A^2 \otimes \text{Id}) + \sigma^2 \mathbb{E}(Z^2 \otimes \text{Id}) - 2\mathbb{E}(A \otimes A). \end{aligned}$$

From this and Lemma 9, the expression (D.4) follows. Equality (D.5) follows by translation of (D.3) and (D.4) to the matrix language.  $\square$

## E Useful facts about positive semidefinite matrices

The following result proves that the entrywise exponential of a conditionally positive definite matrix is positive definite. Recall that a  $n \times n$  symmetric matrix  $M$  is said to be conditionally p.s.d (resp. conditionally p.d) if for all  $v \in \mathbb{R}^{n \times n}$  such that  $v^T \mathbf{1} = 0$  and  $v \neq 0$  we have  $v^T M v \geq 0$  (resp.  $v^T M v > 0$ ).

**Theorem 5.** *Let  $M \in \mathbb{R}^{n \times n}$  be conditionally p.s.d. Then the following statements hold.*

- (i)  $\exp_{\odot}(tM)$  is p.s.d. for all  $t \geq 0$ .
- (ii) If, in addition,  $M$  is conditionally p.d, then  $\exp_{\odot}(tM)$  is p.d. for all  $t > 0$ .
- (iii)  $\exp_{\odot}(M)$  is p.d. if and only if  $M_{ii} + M_{jj} > 2M_{ij}$  for all  $i \neq j$ .

Theorem 5 parts (i) and (iii) are part of [58, Lemma 2.5]. Notice that part (ii) follows from part (iii).

**Lemma 11.** *Let  $A, B \in \mathbb{R}^{n \times n}$  be symmetric matrices. If  $A$  is p.s.d with nonzero diagonal entries, and  $B$  is p.d., then  $A \odot B$  is p.d.*

*Proof.* Denote  $\lambda_{\min}(B)$  to be the smallest eigenvalue of  $B$  (which is strictly positive by assumption). Then by the Schur product theorem we have that

$$A \odot (B - \lambda_{\min}(B) \text{Id}) \succcurlyeq 0 \iff A \odot B \succcurlyeq \lambda_{\min}(B) \text{diag}(A_{11}, A_{22}, \dots, A_{nn}) \text{Id}.$$

Since  $\lambda_{\min}(B) > 0$  and the entries  $A_{11}, \dots, A_{nn}$  are nonzero, we conclude that  $A \odot B \succ 0$ .  $\square$

**Lemma 12.** *Let  $M(t) \in \mathbb{R}^{n \times n}$  be a symmetric matrix which is also integrable w.r.t the Lebesgue measure over the positive reals. If  $M(t)$  is p.s.d (resp. p.d) for  $t > 0$ , then the matrix  $\int_0^{\infty} M(t) dt$  is p.s.d (resp. p.d).*

*Proof.* For any  $v \in \mathbb{R}^n$  we have from the integrability of  $M(t)$  that

$$v^T \left( \int_0^{\infty} M(t) dt \right) v = \int_0^{\infty} v^T M(t) v dt$$

from which the statement follows directly.  $\square$