Solving Partial Assignment Problems Using Random Clique Complexes
(Supplementary)

1. Proofs

1.1. **Upper Bound to Clique Size in a Random Graph.** Let $G(n,p)$ denote the Erdős-Rényi random graph on n vertices, i.e., $G(n,p) = \{G_{ij} | 1 \leq i < j \leq n\}$, where $G_{ij} \sim Ber(p)$ are i.i.d Bernoulli random variables. We denote the number of k-cliques in the realization of $G(n,p)$ as $X_n(k)$. By definition, a k-clique in a graph G is a subset A of k vertices, which induce a complete subgraph of G. Additionally, no other vertex in G can be joined by edges to all vertices of A. Therefore, we can represent $X_n(k)$ as a sum of indicator random variables $\mathbb{1}_A$, where

$$\mathbb{1}_A = \begin{cases} 1 & \text{if } A \text{ is a } k\text{-clique in } G(n,p) \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

It is clear that $X_n(k) = \sum_{|A|=k} \mathbb{1}_A$. Hence, we get

$$\mathbb{E}(X_n(k)) = \mathbb{E}(\sum_{|A|=k} \mathbb{1}_A) = \sum_{|A|=k} \mathbb{E}(\mathbb{1}_A) = \binom{n}{k} p^{\binom{k}{2}}$$

Using Stirling's formula, we upper bound $X_n(k)$ as $\left(\frac{en}{k}\right)^k$, where e is the *Euler's number*.

1.2. **Quadratic Assignment Problem.** We begin by defining the general quadratic assignment problem (QAP) using the *Koopman-Beckmann* version. Let $A = (a_{vv'})$, $B = (b_{vv'}) \in \mathbb{R}^{n \times n}$. Let Π denote the set of all possible bijections (permutations) $\pi : N \to N$, where $N = \{1, 2, \dots n\}$. We define the QAP as:

$$\begin{aligned} \text{minimize} \quad & \sum_{v,v'} b_{vv'} a_{\pi(v)\pi(v')} \\ \text{subject to} \quad & \pi \in \Pi \end{aligned}$$

For now on, for ease of notation, we denote the cost function $b_{vv'} a_{\pi(v)\pi(v')}$ as $\mathcal{C}_{vv'}$.

1.3. **Asymptotic Analysis of Higher-order Clique Assignment (Proof of Theorem 3).** Given that the QAP is a combinatorial optimization problem, in the case of random symmetric matrices, the subset of *feasible solutions* S_π is of the form:

$$S_\pi = \{(\pi(v), \pi(v') \mid v < v', u, v = 1, \dots, n\}$$

where, $|S_\pi| = \binom{n+1}{2}$ and $|\Pi| = n!$.

Recall that our cost function $\mathcal{C}_{vv'}$ has expectation λ_e and variance λ_v. For notational convenience, we set $\epsilon' = \lambda_v - \epsilon$. Then, there exists a bijection $\pi \in \Pi$, for which the following holds by the

definition of variance

$$\mathbb{P}\left\{\frac{1}{|S_\pi|}\left|\sum_{v,v'\in\pi}(\mathcal{C}_{vv'}-\lambda_e)\right|\geq\epsilon'\right\}$$

To proceed further with our proof, we make use of the following lemma by Renyi et. al. [7].

Lemma 1. *Let $X_1,\ldots,X_n$ be independent random variables with $|X_k-\mathbb{E}(X_k)|\leq 1$, $k=1,\ldots,n$. Denote*

$$D:=\sqrt{\sum_{k=1}^{n}\mathbb{V}(X_k)}$$

and let μ be a positive real number with $\mu\leq D$. Then

$$\mathbb{P}\left\{\left|\sum_{k=1}^{n}(X_k-\mathbb{E}(X_k))\right|\geq\mu D\right\}\leq 2\exp\left(-\frac{\mu^2}{2(1+\mu/2D)^2}\right)$$

□

In order to apply Lemma 1, we change the form of the inequality as follows:

$$\mathbb{P}\left\{\frac{1}{|S_\pi|}\left|\sum_{v,v'\in\pi}(\mathcal{C}_{vv'}-\lambda_e)\right|\geq\epsilon'\right\} \tag{2}$$

$$\leq\sum_{\pi\in\Pi}\mathbb{P}\left\{\frac{1}{|S_\pi|}\left|\sum_{v,v'\in\pi}(\mathcal{C}_{vv'}-\lambda_e)\right|\geq\epsilon'\right\} \tag{3}$$

$$\leq|\Pi|\mathbb{P}\left\{\frac{1}{|S_\pi|}\left|\sum_{v,v'\in\pi}(\mathcal{C}_{vv'}-\lambda_e)\right|\geq\epsilon'\right\} \tag{4}$$

Before applying Lemma 1, we compute D as,

$$D=\sqrt{\sum_{v,v'\in\pi}\lambda_v}=\sqrt{|S_\pi|\lambda_v}$$

We can rewrite (4) as

$$|\Pi|\mathbb{P}\left\{\left(\frac{\sqrt{\lambda_v}}{\sqrt{|S_\pi|}}\right)\left(\frac{1}{\sqrt{|S_\pi|\lambda_v}}\right)\left|\sum_{v,v'\in\pi}(\mathcal{C}_{vv'}-\lambda_e)\right|\geq\epsilon'\right\}$$

$$=|\Pi|\mathbb{P}\left\{\left|\sum_{v,v'\in\pi}(\mathcal{C}_{vv'}-\lambda_e)\right|\geq\underbrace{\left(\frac{\epsilon'\sqrt{|S_\pi|}}{\sqrt{\lambda_v}}\right)}_{\mu}\underbrace{\left(\sqrt{|S_\pi|\lambda_v}\right)}_{D}\right\}$$

Now, we make use of Lemma 1 and get

$$
\begin{aligned}
&|\Pi|\mathbb{P}\left\{\left|\sum_{v,v'\in\pi}(\mathcal{C}_{vv'}-\lambda_e)\right| \geq \left(\frac{\epsilon'\sqrt{|S_\pi|}}{\sqrt{\lambda_v}}\right)\left(\sqrt{|S_\pi|\lambda_v}\right)\right\} \\
\leq & 2|\Pi|\exp\left(-\frac{\left(\frac{\epsilon'\sqrt{|S_\pi|}}{\sqrt{\lambda_v}}\right)^2}{2\left(1+\frac{\epsilon'\sqrt{|S_\pi|}}{\sqrt{\lambda_v}}\frac{1}{2\sqrt{|S_\pi|\lambda_v}}\right)^2}\right) \\
= & 2|\Pi|\exp\left(-2|S_\pi|\left(\frac{\epsilon'\sqrt{\lambda_v}}{\epsilon'+2\lambda_v}\right)^2\right)
\end{aligned}
$$

Equation 2 can now be written as

$$
\mathbb{P}\left\{\frac{1}{|S_\pi|}\left|\sum_{v,v'\in\pi}(\mathcal{C}_{vv'}-\lambda_e)\right| \leq \epsilon'\right\} \geq 1-2|\Pi|\exp\left(-2|S_\pi|\left(\frac{\epsilon'\sqrt{\lambda_v}}{\epsilon'+2\lambda_v}\right)^2\right)
$$

It can easily be verified that the expression in the R.H.S. of the above inequality tends to 1 as $n\to\infty$.

We know that for the expression $\left|\sum_{v,v'\in\pi}(\mathcal{C}_{vv'}-\lambda_e)\right| \leq \epsilon'|S_\pi|$, the following bounds hold.

$$
|S_\pi|(\lambda_e-\epsilon') \leq \sum_{v,v'\in\pi}\mathcal{C}_{vv'} \leq |S_\pi|(\lambda_e+\epsilon')
$$

It follows that

$$
\frac{\max\limits_{\pi\in\Pi}\sum\limits_{vv'}\mathcal{C}_{vv'}}{\min\limits_{\pi\in\Pi}\sum\limits_{vv'}\mathcal{C}_{vv'}} \leq \frac{|S_\pi|(\lambda_e+\epsilon')}{|S_\pi|(\lambda_e-\epsilon')} \leq 1+\epsilon
$$

This completes the proof. □

1.4. **Eigenvalue Bounds on Lawler's QAP Formulation on Random Matrices (Proof of Theorem 2).** As illustrated in [1], we will make use of Talagrand's concentration inequality. We provide a tighter bound in the case of our affinity matrix using the Rayleigh's quotient.

Theorem 1. *[8] Let $\Omega=\prod_{i=1}^m\Omega_i$ be a product space of probability spaces. Let $\mathcal{A}$ and $\mathcal{A}_t$ be subsets of Ω and if for each $y=(y_1,\ldots,y_m)\in\mathcal{A}_t$, there exists a real vector $\alpha=(\alpha_1,\ldots,\alpha_m)$, such that for every $x=(x_1,\ldots,x_k)\in\mathcal{A}$, the following inequality holds*

$$
\sum_{i:x_i\neq y_i}|\alpha_i| \leq t\left(\sum_{i=1}^m\alpha_i^2\right)^{1/2}
$$

Then,

$$
\mathbb{P}[\mathcal{A}]\mathbb{P}[\bar{\mathcal{A}}_t] \leq e^{-t^2/4}.
$$

Here, $\mathcal{A}_t$ denotes the set with *Talagrand distance* at most t from $\mathcal{A}$ and $\bar{\mathcal{A}}_t$ denotes the *complement* of set $\mathcal{A}_t$. □

Theorem 2. *For a real symmetric matrix $A = (a_{ij}) \in R^{m\times m}$ and for positive constant t,*

$$\mathbb{P}[|\lambda_1(A) - \mathcal{M}| \geq t] \leq 4e^{-t^2/8},$$

where $\mathcal{M}$ is the median of $\lambda_1(A)$.

Proof. [1] Given a real symmetric matrix $A = (a_{ij}) \in R^{m\times m}$ and a non-zero vetor x, the *Rayleigh Quotient* $\mathcal{R}(A, x)$ is defined as

$$\mathcal{R}(A, x) = \frac{x^T Ax}{x^T x}$$

Given the eigenvalues of A in decreasing order as $\lambda_1(A) \geq \cdots \geq \lambda_m(A)$, we know that $\mathcal{R}(A, x) \in [\lambda_m(A), \lambda_1(a)]$. It is well known that $\mathcal{R}(A, x)$ attains its maximum value at $\lambda_1(A)$ when $x = v$, where v is the eigenvector corresponding to $\lambda_1(A)$. Therefore, we have

$$\lambda_1(A) = \mathcal{R}(A, v) = \frac{v^T Av}{v^T v} \tag{5}$$

In our proof, we omit the constant factor $v^T v$ and normalize the eigenvector v, hence $\|v\| = 1$.

Consider the product space Ω of entries a_{ij}, $1 \leq i \leq j \leq m$. Let $t, \mathcal{M}$ be real numbers, where $t > 0$ and $\mathcal{M}$ is the median of $\lambda_1(A)$. Let $\mathcal{A}$ be the set of matrices $A = (a_{ij}) \in \Omega$, for which $\lambda_1(A) \leq \mathcal{M}$. By definition, $\mathbb{P}[\mathcal{A}] \geq 1/2$. Additionally, let $\mathcal{B}$ be the set of matrices $B = (b_{ij}) \in \Omega$, for which $\lambda_1(B) \geq \mathcal{M} + t$. Using Rayleigh's equation (5) for $\lambda_1(A)$, we rewrite it as a summation of diagonal and off-diagonal terms

$$\lambda_1(A) = \mathcal{R}(A, v) = v^T Av = \underbrace{\sum_{1\leq i<j\leq m} (v_i^T v_j + v_j^T v_i)a_{ij}}_{\text{off-diagonal}} + \underbrace{\sum_{i=1}^{m} v_i^T v_i a_{ii}}_{\text{diagonal}} \leq \mathcal{M} \tag{6}$$

and

$$\lambda_1(B) = \mathcal{R}(B, v) = v^T Bv = \underbrace{\sum_{1\leq i<j\leq m} (v_i^T v_j + v_j^T v_i)b_{ij}}_{\text{off-diagonal}} + \underbrace{\sum_{i=1}^{m} v_i^T v_i b_{ii}}_{\text{diagonal}} \geq \mathcal{M} + t \tag{7}$$

In order to apply Talagrand's inequality (Theorem 1), we set a real vector $\alpha = (\alpha_{ij})_{1\leq i\leq j\leq m}$ as follows: For off-diagonal $(1 \leq i < j \leq m)$ terms, we set

$$\alpha_{ij} = (v_i^T v_j + v_j^T v_i)$$

For diagonal $(1 \leq i \leq m)$ terms, we set

$$\alpha_{ii} = v_i^T v_i$$

We proceed by first proving two claims that will be used in this proof.

[1] Our proof technique follows the technique outlined in [1]

Claim 1.

$$\sum_{1\le i\le j\le m} \alpha_{ij}^2 \le 2$$

Proof. By definition,

$$\begin{aligned}\sum_{1\le i\le j\le m} \alpha_{ij}^2 &= \sum_{i=1}^{m}(v_i^T v_i)^2 + \sum_{1\le i<j\le m}(v_i^T v_j + v_j^T v_i)^2 \\ &< 2\left(\sum_{i=1}^{m} {v_i^T}^2\right)\left(\sum_{i=1}^{m} {v_i}^2\right) \quad \text{(by AM-GM inequality)} \\ &= 2 \quad (\text{since } \|v\| = 1)\end{aligned}$$

This completes the proof. □

Claim 2. For every $A \in \mathcal{A}$,

$$\sum_{1\le i\le j\le m; a_{ij}\neq b_{ij}} |\alpha_{ij}| \ge t$$

Proof. Recall that for matrix $A \in \mathcal{A}$, v is the eigenvector with unit-norm corresponding to $\lambda_1(A)$. We know that,

$$v^T A v \le \lambda_1(A) \le \mathcal{M} \quad (\text{from set } \mathcal{A})$$

while,

$$v^T B v \ge \lambda_1(B) \ge \mathcal{M} + t \quad (\text{from set } \bar{\mathcal{A}}_t)$$

We observe that the entries in affinity matrices A and B, are *affinity scores* in interval $[0, 1]$. Therefore, we have $|b_{ij} - a_{ij}| \le 1$, for all $1 \le i, j \le m$. For ease of notation, let us denote by P, the set of ordered pairs ij with $1_1, j \le m$ where $a_{ij} \neq b_{ij}$. Then,

$$\begin{aligned} t \le v^T(B - A)v &= \sum_{i,j\in P}(b_{ij} - a_{ij})v_i^T v_j \\ &\le \sum_{i,j\in P} |v_i^T||v_j| \le \sum_{i,j\in P} |\alpha_{ij}|\end{aligned}$$

This completes the proof. □

By the above two claims, we get the following form:

$$\sum_{x_i\neq y_i} |\alpha_i| \ge t > \left(\frac{t}{\sqrt{2}}\right)\left(\sum_{1\le i\le j\le m} \alpha_{ij}^2\right)^{1/2}$$

Applying Talagrand's inequality, we get

$$\mathbb{P}[\lambda_1(A) \le \mathcal{M}]\mathbb{P}[\lambda_1(B) \ge \mathcal{M} + t] \le e^{\frac{-1}{4}\left(\frac{t}{\sqrt{2}}\right)^2} \le e^{-t^2/8}$$

Since $\mathcal{M}$ is the median of $\lambda_1(A)$, by definition $\mathbb{P}[\lambda_1(A) \leq \mathcal{M}] \geq 1/2$, then

$$\mathbb{P}[\lambda_1(A) \geq \mathcal{M} + t] \leq 2e^{-t^2/8} \tag{8}$$

Accordingly, we also have that,

$$\mathbb{P}[\lambda_1(A) \leq \mathcal{M} - t] \leq 2e^{-t^2/8} \tag{9}$$

Combining results (8) and (9), we have

$$\mathbb{P}[|\lambda_1(A) - \mathcal{M}| \geq t] \leq 4e^{-t^2/8} \tag{10}$$

This completes the proof. □

1.5. **Proof of Lemma 1.** For ease of understanding, we drop the (A) as it is obvious from context. Let λ_i be the i-th eigenvalue of A, and let $x_i \neq 0$ be its corresponding eigenvector. From $Ax_i = \lambda_i x_i$, we have

$$AX_i = \lambda_i X_i, \qquad \text{where } X_i := \begin{bmatrix} x_i \mid \ldots \mid x_i \end{bmatrix} \in \mathcal{M}_n \setminus \{0\}$$

It follows,

$$|\lambda_i| |||X_i||| |||X_i||| = |||\lambda_i X_i||| = |||AX_i||| \leq |||A||| |||X_i|||.$$

As $|||X|||$ is non-negative, we get $|\lambda_i| \leq |||A|||$. Thus, every eigenvalue of A is upper bounded by the matrix norm $|||A|||$. Applying the triangle inequality, we get that $\delta_i(A) = |\lambda_{i+1}(A) - \lambda_i(A)| \leq 2|||A|||$, which completes the proof. □

2. Example

Cliques	**Graph 1** (G_1)		**Graph 2** (G_2)	
	Clique	Neighbours	Clique	Neighbours
3-Cliques	{A,B,D}	{A}, {B}, {D}, {A,B}, {A,D}, {B,D}, {B,D,E}	{A,C,D}	{A}, {C}, {D}, {A,C}, {A,D}, {C,D}, {B,D,E}
	{B,D,E}	{B}, {D}, {E}, {B,D}, {B,E}, {D,E}, {A,B,D}	{B,D,E}	{B}, {D}, {E}, {B,D}, {B,E}, {D,E}, {A,C,D}
2-Cliques	{A,B}	{A}, {B}, {A,B,D}	{A,C}	{A}, {C}, {A,C,D}
	{A,C}	{A}, {C}	{A,D}	{A}, {D}, {A,C,D}
	{A,D}	{A}, {D}, {A,B,D}	{B,D}	{B}, {D}, {B,D,E}
	{B,D}	{B}, {D}, {A,B,D}, {B,D,E}	{B,E}	{B}, {E}, {B,D,E}
	{B,E}	{B}, {E}, {B,D,E}	{C,D}	{C}, {D}, {A,C,D}
	{D,E}	{D}, {E}, {B,D,E}	{D,E}	{D}, {E}, {B,D,E}
	{D,F}	{D}, {F}	{D,F}	{D}, {F}
1-Cliques	{A}	{A,B}, {A,C}, {A,D}, {A,B,D}	{A}	{A,C}, {A,D}, {A,C,D}
	{B}	{A,B}, {B,D}, {B,E}, {A,B,D}, {B,D,E}	{B}	{B,D}, {B,E}, {B,D,E}
	{C}	{A,C}	{C}	{A,C}, {C,D}, {A,C,D}
	{D}	{A,D}, {B,D}, {D,E}, {D,F}, {A,B,D}, {B,D,E}	{D}	{A,D}, {B,D}, {C,D}, {D,E}, {D,F}, {A,C,D}, {B,D,E}
	{E}	{B,E}, {D,E}, {B,D,E}	{E}	{B,E}, {D,E}, {B,D,E}
	{F}	{D,F}	{F}	{D,F}

Table 1. Neighbourhood of $3, 2, 1$-cliques of graphs G_1 and G_2 shown in Figure 1.

We explained our method with the help of an example shown in Figure 1, Table 1 and Table 2 for a better understanding. We consider two random graphs G_1 and G_2 with 6 vertices each in Figure 1(a) for which we perform higher-order matching from 3-cliques 1(b) to 1-cliques 1(d). For a higher-order matching, we take the neighbourhood of a barycenter of a clique as the barycenters

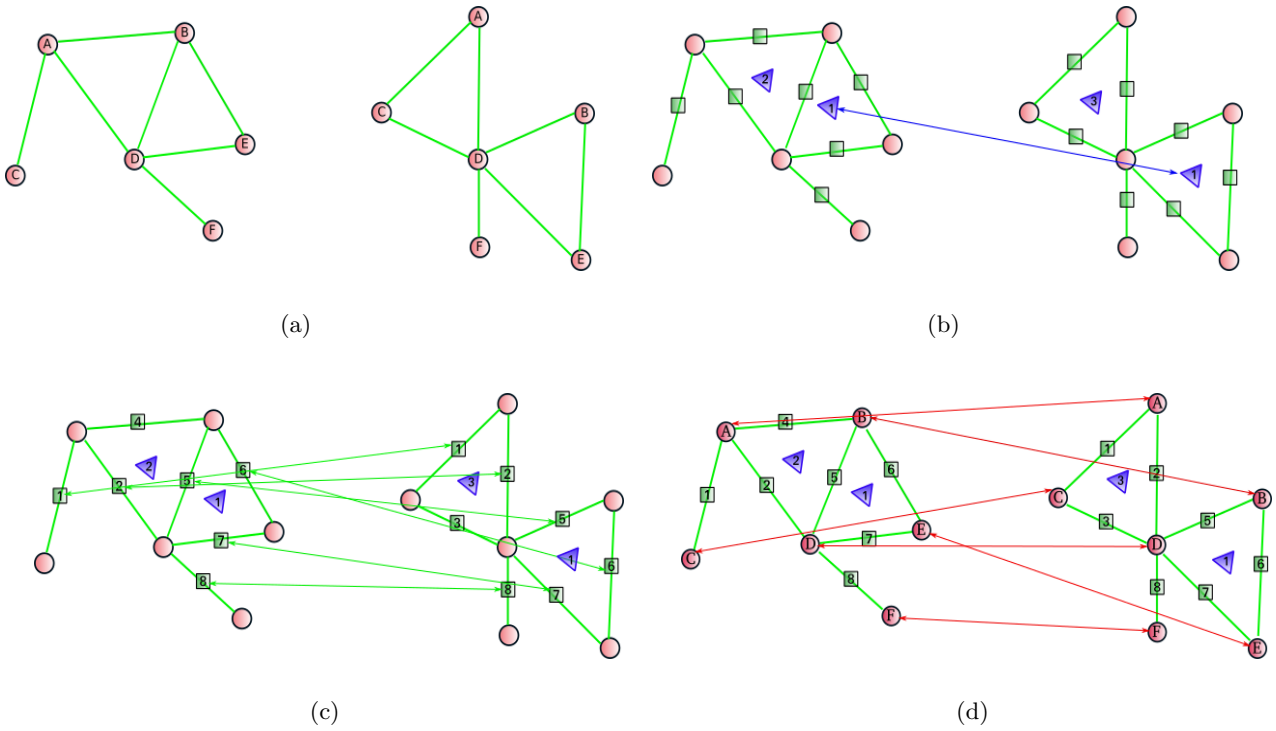


FIGURE 1. Matching corresponding 3-clique, 2-cliques and 1-cliques (b)-(d) respectively in a pair of (a) Erdős-Rényi graph.

Cliques	**Graph 1** (G_1)	**Graph 2** (G_2)
3-Cliques	{B,D,E} → 1	{B,D,E} → 1
2-Cliques	{A,C} → 1 {A,D} → 2 {B,D} → 5 {B,E} → 6 {D,E} → 7 {D,F} → 8	{A,C} → 1 {A,D} → 2 {B,D} → 5 {B,E} → 6 {D,E} → 7 {D,F} → 8
1-Cliques	{A}, {B}, {C} {D}, {E}, {F}	{A}, {B}, {C} {D}, {E}, {F}

TABLE 2. Matchings of $3, 2, 1$-cliques of graphs G_1 and G_2 shown in Figure 1.

of the other cliques it is connected to. Thus, we place additional nodes of different order in the neighbourhood of each clique in addition to the same order cliques. This information would help the cliques to have more accurate matches. The neighbours and matchings of 3-cliques, 2-cliques and 1-cliques are mentioned in the Table 1 and Table 2 for both the graphs G_1 and G_2 respectively.

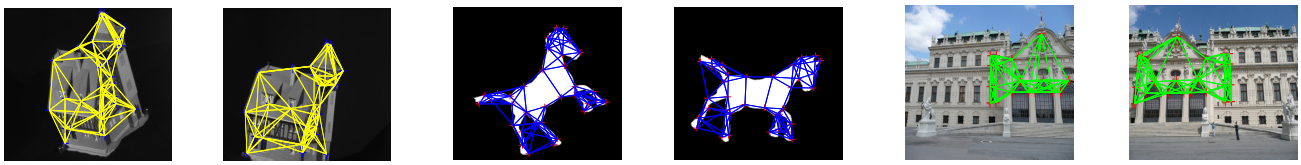
And, the matchings shown in Figure 1(b), 1(c) and 1(d) are based on having the same labels for each barycenter in graph G_1 and G_2.

3. Experiments

3.1. **Setup.** We compare the performance of our proposed method with various other matching algorithms on synthetic and real world datasets. The real world datasets are categorized in Table 3. Here, N is the total number of samples with n landmark points in each image to be matched. We represent random graphs on images in Figure 2 for better understanding and visualization of random graphs for our experiments. Matchings of two images for real world datasets (Table 3) are shown in Figure 9.

Groups	**Dataset**	$N \times n$
Video Frames	CMU House	111×30
	CMU Hotel	101×30
Affine	Horse-Rot [2]	200×35
	Horse-Shear [2]	200×35
Occluded	Books [6]	20×34
	Building [6]	16×28
Non-Affine	Magazine [5]	30×30
	Butterfly [5]	30×19
Object Matching	Car [3]	40×10
	Bike [3]	40×10

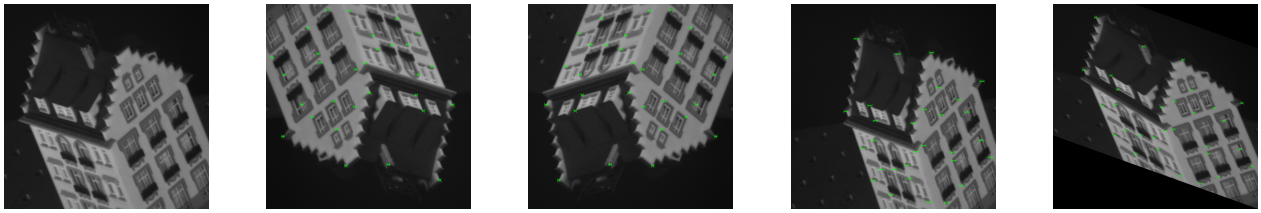
Table 3. Datasets used, where N is the number of samples and n is the dimensionality of each sample.



(a) (b) (c)

Figure 2. Pair of Erdős-Rényi graphs on *CMU House*, *Horse Rotate*, and *Building* datasets.

3.2. **Effect of Affine Transformation.** We created a synthetic dataset from CMU House and Hotel dataset by uniformly sampling 20% and 40% frames from a video sequence and performing affine transformations like rotation, reflection, scaling, and shearing. We have explained the transformations we considered for this experiment which is similar to Figure (2) and Table (1) in main paper. Table(1) in main paper shows the results on the CMU House dataset. Affine transformations



(a) (b) (c) (d) (e)

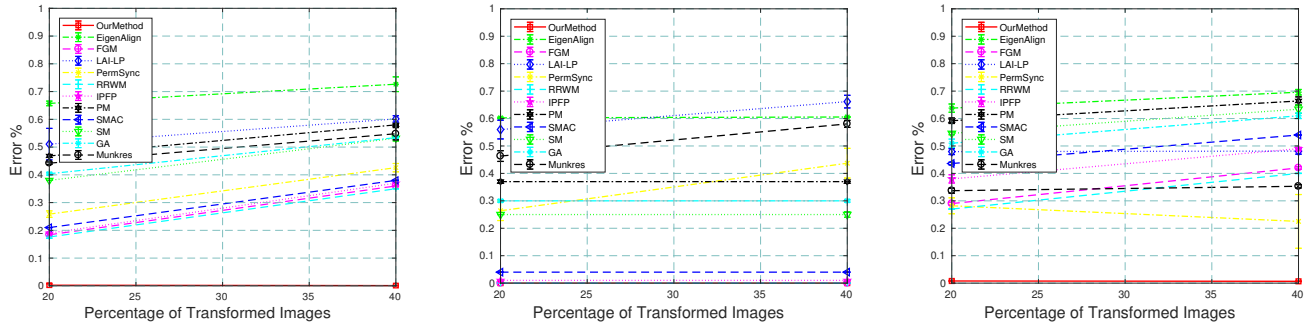
FIGURE 3. (a) Original *Hotel* frame, (b)–(e) four transformations on hotel frame: rotation, reflection, scaling, and shear (green markers show true matching case with the original frame (a)).

on Hotel frame are shown in Figure 3. Figure 4 shows the results of matching for the remaining House (fig. 4(a) and 4(b)) and Hotel synthetic dataset for all the algorithms. We observe that our method produces best results in all the cases, whereas the error for other algorithms either remains stable or increases steeply with the increase in the percentage of transformed frames in the sequence.

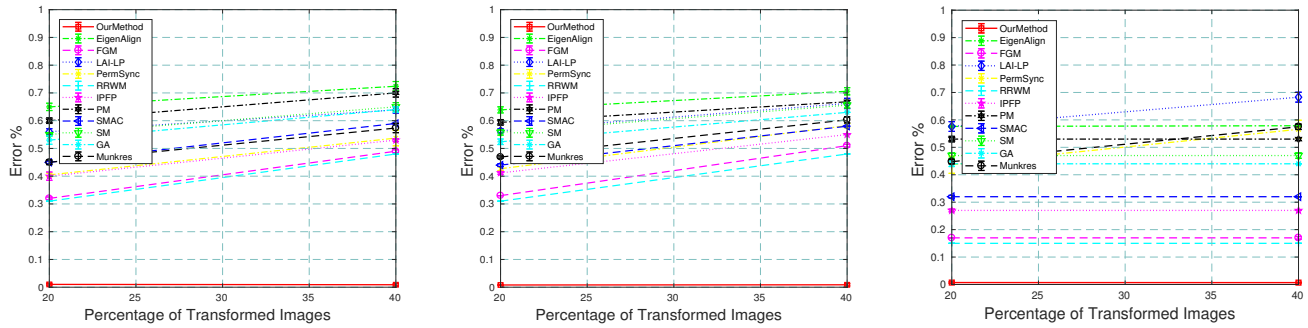
3.3. **Effect of Occlusion.** We considered two datasets with grave occlusions, mentioned in Table 3. Figures 9(i) and 9(j) show the matching of two images for both the datasets, although the matching results are shown in Table (2) in the main paper. We also created a synthetic dataset by removing $2, 4, 6, 8$, and 10 (6.66%, 13.33%, 20%, 26.66%, and 33.33%) points out of total house landmark points (i.e., 30 points) from 20% and 60% of frame sequences randomly. Figures 5(a) and 5(b) show the increase in error as we remove more points from images. We also note the difference in both the results. Since we remove points from more percentage of frames in 5(b), there is more gradual increase in the error. This experimental setup is similar to Figure (4) in our main paper. It shows that affinity based methods like FGM and RRWM perform well but our method still consistently outperforms all the algorithms.

3.4. **Effect of Frame Separation.** Figures 6(a) and 6(b) show the frame separation level result of CMU House and Horse Rotate frame sequences. We select a pair of frames at a time with increase in their frame separation (x-axis). Here, the House dataset consists of 3D rotations of *House* whereas *Horse Rotate* dataset applies rotation with more degree of rotation as the frame separation level increases. We see that most of the algorithms performs well for both the datasets even with 0% error.

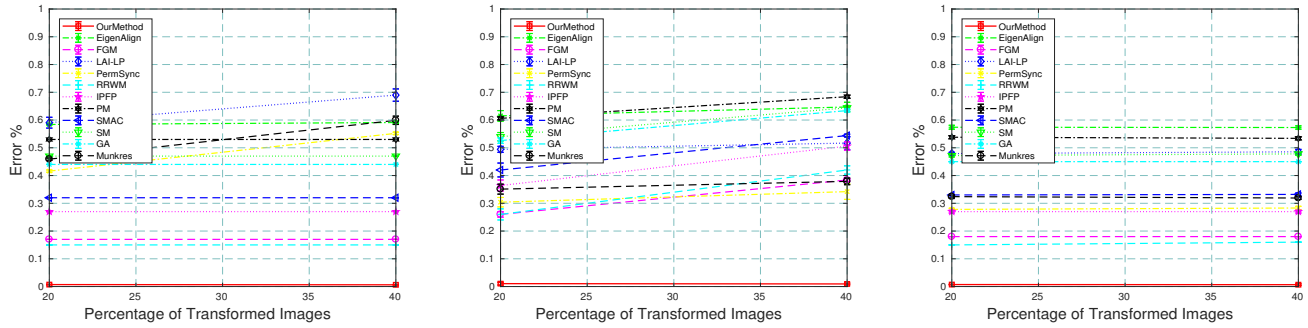
3.5. **Effect of k-Nearest Neighbour.** In Figures 7(a) and 7(b), error and computation time of matching two frames of house are shown with different probability p and nearest neighbor k values. We observe that as the value of p and k increases, the possibility of mismatching decreases which leads to correct matching. On the other hand, the computation time increases since it increases the number of edges in the underlying graph, which in turn leads to a larger number of d-cliques. This also causes a marked increase in the matching algorithm's runtime. The computation time



(a) (b) (c)

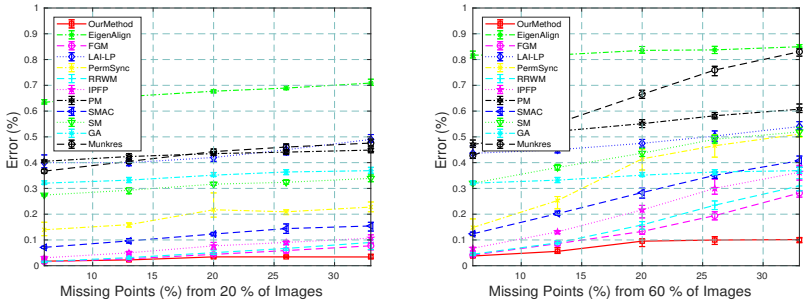


(d) (e) (f)



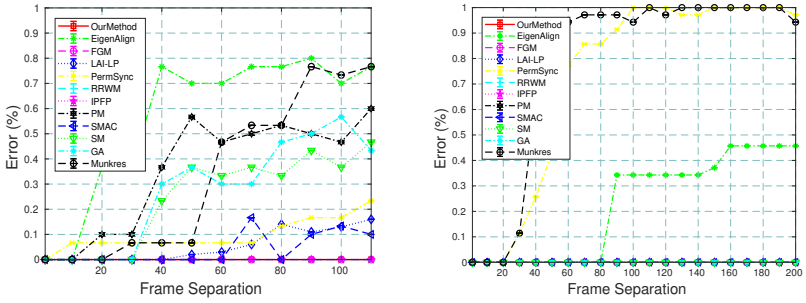
(g) (h) (i)

FIGURE 4. Error(%) in matching when varying the percentage (20% and 40%) of transformed images in the frame sequence of *CMU House* (a)-(b) and *CMU Hotel* (c)-(i) . (a) 40° rotation, (b) 90° rotation, (c) 20° rotation, (d) 40° rotation, (e) 60° Degree rotation, (f) 90° rotation, (g) reflection, (h) scaling, and (i) shearing.



(a) (b)

FIGURE 5. Error (%) in matching when varying the number of missing landmarks in (a) 20% and (b) 60% of the images in *CMU House* frame sequence.



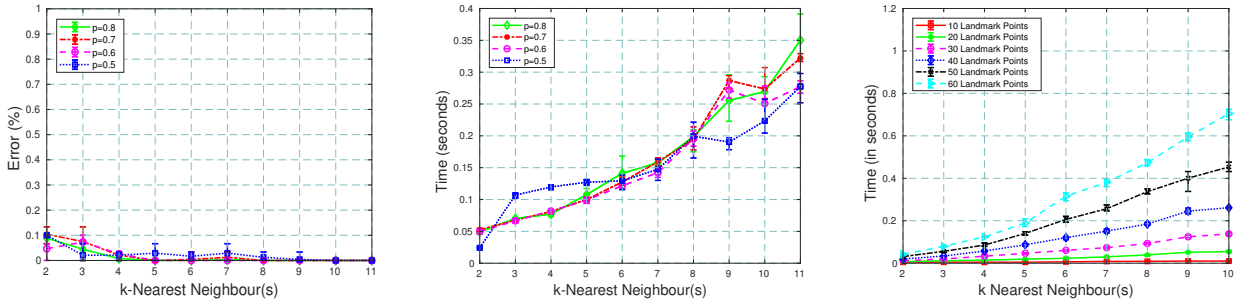
(a) (b)

FIGURE 6. Error (%) in matching by various methods with different frame separation level for (a) *CMU House* and (b) *Horse Rotate*.

of our algorithm considers the time of the *Kuhn-Munkres* algorithm, which is used as a matching algorithm to match two random clique complexes, which takes $O(n^3)$ running time.

The overall time increases as we increase the value of p and k, since it increases the probability of an edge occurrence between two landmark points. As the number of edges increase in a random graph, the number of d-cliques also increase. Due to this phenomenon, the runtime of the *Kuhn-Munkres* algorithm also increases.

Figure 7(c) shows the computation time of matching two images with varying k-NN for different n landmark points in the image. We can clearly see that the time increases with increasing k and a larger number of landmark points. Here, 60 landmark points take maximum time for the highest value of k. On the other hand, if we consider lower values of k, even 60 landmark points take a reasonable amount of time to match, which is comparable to lower values of n. Thus, we set k value as low as possible for matching, depending on the complexity of the dataset.



(a) (b) (c)

FIGURE 7. (a) and (b) Error(%) and runtime of matching two frames of *House* with varying p and k. Error decreases with p and k, whereas computation time increases. (c) computation time of matching two images with varying k-NN for $n = 10 - 60$ landmark points. p is fixed as 0.6 here for all the cases.

3.6. **Noise Model.** We analyze the performance of our method over other pairwise algorithms for two different noise models. We follow the noise model setup mentioned in [4]. We introduce noise in one random graph G_1 and generate a noisy version $\widetilde{G}$ to be matched with G_2. G_1 is a random graph here which is created as $G_1(n, p)$ with n nodes and p probability. We describe two noise models as follows:

Noise Model I:

$$\widetilde{G} = G_1 \odot (1 - A) + (1 - G_1) \odot A \tag{11}$$

$\widetilde{G}$ is generated using the aforementioned equation where A is a binary random symmetric matrix, whose entries are drawn from a *Bernoulli distribution* as $A(n, q)$ with n nodes and q probability and $\odot$ represents the element-wise multiplication of matrices. This model flips the node-node adjacency of G_1 with probability q.

Noise Model II:

$$\widetilde{G} = G_1 \odot (1 - A) + (1 - G_1) \odot B \tag{12}$$

Again, A and B are binary random symmetric matrices, whose entries are drawn from the Bernoulli distribution as $A(n, q)$ and $B(n, r)$ with n nodes and q and r probabilities, respectively. This model flips node-node adjacency of G_1 with probability q, and in addition it also creates edges between non-connected nodes with probability r.

Results of noise model I and II on *CMU Hotel* and *Horse Shear* for frame separation level is shown in Figures 8(a), 8(c) and 8(b), 8(d) respectively. We observe that our method is robust to noise for both the models as compared to other algorithms since there is a very small increase or no increase in error (%) for all the cases.

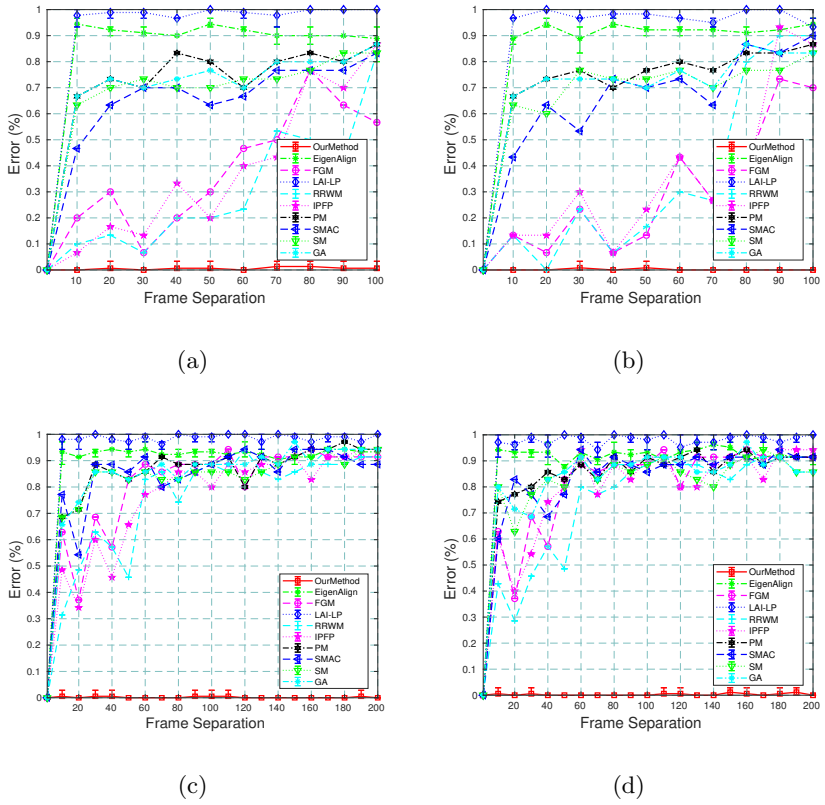
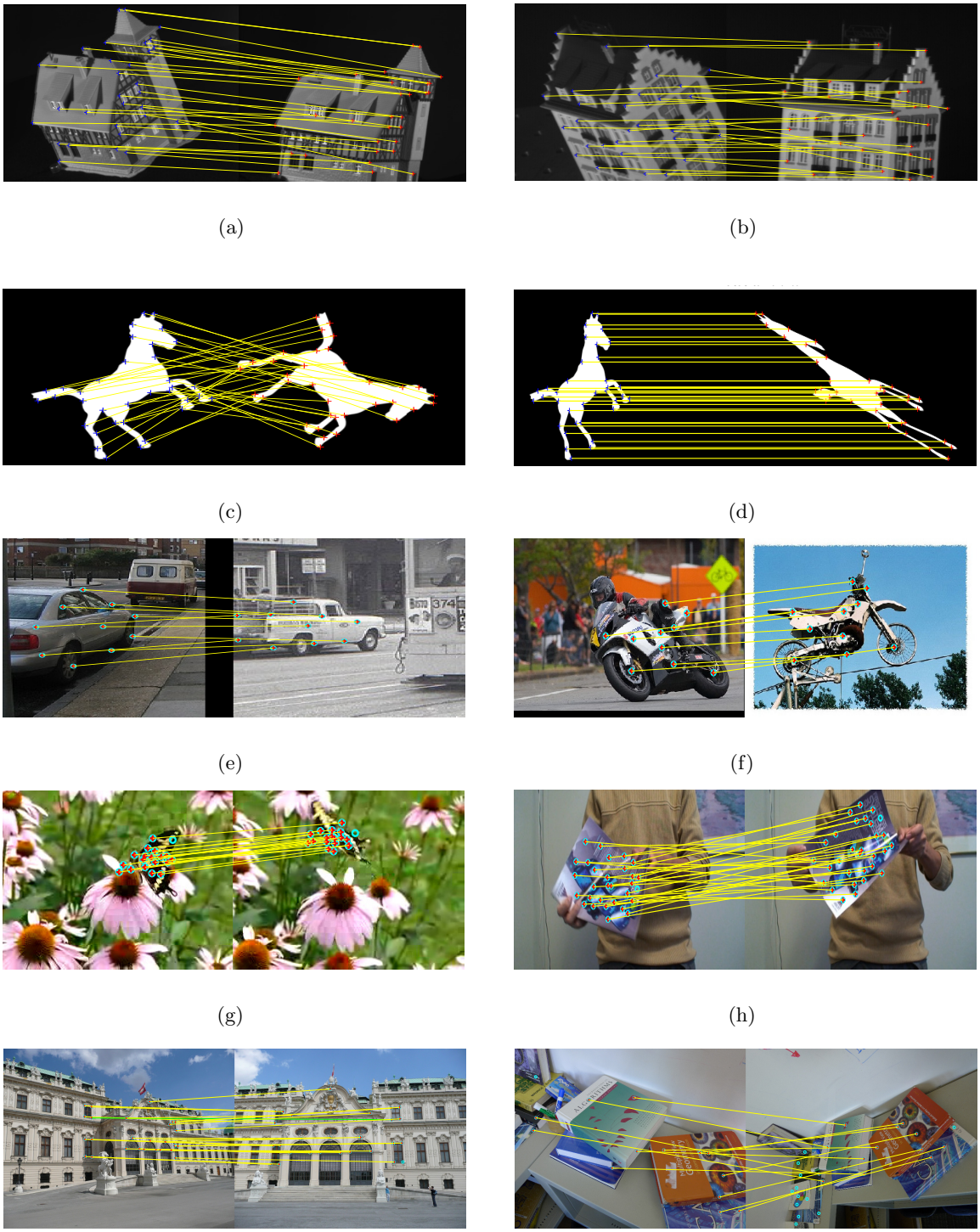


FIGURE 8. Error (%) in matching by various methods with different frame separation level for Noise Model (a) I, (b) II for *CMU Hotel* and (c) I, (d) II for *Horse Shear*.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

(i) (j)

FIGURE 9. Instances of matchings in (a) House, (b) Hotel, (c) Horse Rotate, (d) Horse Shear frame sequences, (e) Car, (f) Bike, (g) Butterfly, (h) Magazine, (i) Building and (j) Books dataset. Yellow/green lines show correct/incorrect matches and isolated points show no matches.

References

[1] N. Alon, M. Krivelevich, and V. H. Vu. On the concentration of eigenvalues of random symmetric matrices. *Israel Journal of Mathematics*, 131(1):259–267, Dec 2002.

[2] T. S. Caetano, J. J. McAuley, L. Cheng, Q. V. Le, and A. J. Smola. Learning graph matching. *IEEE transactions on pattern analysis and machine intelligence*, 31(6):1048–1058, 2009.

[3] M. Cho, K. Alahari, and J. Ponce. Learning graphs to match. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 25–32, 2013.

[4] S. Feizi, G. Quon, M. Recamonde-Mendoza, M. Médard, M. Kellis, and A. Jadbabaie. Spectral alignment of networks. *arXiv preprint arXiv:1602.04181*, 2016.

[5] H. Jiang, X. Y. Stella, and D. R. Martin. Linear scale and rotation invariant matching. *IEEE transactions on pattern analysis and machine intelligence*, 33(7):1339–1355, 2011.

[6] D. Pachauri, R. Kondor, and V. Singh. Solving the multi-way matching problem by permutation synchronization. In *Advances in neural information processing systems*, pages 1860–1868, 2013.

[7] A. Rényi. *Probability Theory*. North-Holland, Amsterdam, 1970.

[8] M. Talagrand. Concentration of measure and isoperimetric inequalities in product spaces. *Publications Mathématiques de l'Institut des Hautes Études Scientifiques*, 81(1):73–205, 1995.