Exact Shape Correspondence via 2D Graph Convolution.

Appendices:

A Related work

A.1 point-wise Alignment

Given a set of point-wise descriptors $\mathbf{X}_A \in \mathbb{R}^{n_1 \times d}$ and $\mathbf{X}_B \in \mathbb{R}^{n_2 \times d}$ for two 3D shapes $\mathcal{M}_1$ and $\mathcal{M}_2$, such as the ones based on (1) the Laplacian Beltrami operator (LBO) e.g., Heat Kernel Signatures (HKS) [11], Wave Kernel Signatures (WKS) [6], Global point Signatures (GPS) [53], which all use a truncated basis of the eigen-functions (eigen-vectors) of the LBO rather than using the full basis, or given point-wise descriptors based on (2) the Geodesic distances e.g., Geodesic distance descriptor (GDD) [62], or (3) others based on signatures of histograms such as SHOT [68], the *point-wise alignment* problem can be formulated as:

$$\min_{\mathbf{P}} ||\mathbf{P}\mathbf{X}_A - \mathbf{X}_B||_F^2 = \max_{\mathbf{P}} \mathbf{Tr}(\mathbf{P}^T \mathbf{X}_A \mathbf{X}_B^T), \tag{13}$$

As observed by Lähner et al. [40], there are two main problems with this approach. Which are: (a) despite the variety of point-wise shape descriptors proposed, most assume near-isometry (i.e., assume no noise), and (b) shapes usually have intrinsic symmetries (e.g., humans left side and right side are almost perfect copies of each other) which is not easily captured by point-wise descriptors. The later problem will result in serious discontinuities in the map as some points will be mapped to their desired location, but others to their symmetric counterparts e.g., the left eye and right eye of a human face both being mapped to the right eye of the other face. As such For 2D-GEM we not only use poit-wise descriptors, but combine them with pair-wise descriptors via the 2D-graph convolution.

A.2 pair-wise Alignment.

Given a set of pair-wise descriptors $\mathbf{A} \in \mathbb{R}^{n_1 \times n_1}$ and $\mathbf{B} \in \mathbb{R}^{n_2 \times n_2}$ for two 3D shapes $\mathcal{M}_1$ and $\mathcal{M}_2$ respectively, such as their; Laplacians, adjacency matrices, geodesic distance matrices, mass matrices, or stiffness matrices [77, 40], the alignment problem is given as the quadratic assignment problem (QAP) [21]:

$$\min_{\mathbf{P}} ||\mathbf{P}\mathbf{A}\mathbf{P}^T - \mathbf{B}||_F^2 = \max_{\mathbf{P}} \mathbf{Tr}(\mathbf{P}^T \mathbf{A}\mathbf{P}\mathbf{B}), \tag{14}$$

The QAP problem is computationally intractable. Thus, for 2D-GEM we employ the 2D-graph convolution which helps us reduce the alignment problem to a Linear Assignment Problem (LAP). It is notable to mention however that several relaxations have been proposed for the QAP problem (some of which are grouped as spectral methods and Quadratic Programming methods below).

Spectral Methods. These methods replace the requirement for permutation matrices with a single constraint on the Frobenious norm of the matrices to obtain a maximal eigenvalue problem [63, 41, 48]. They are used both for testing graph isomorphism [35, 7] and for graph matching. They usually reduce (14) into a Linear Assignment Problem (LAP) of the form $\max_{\mathbf{P}} \mathbf{Tr}(\mathbf{P}\mathbf{C})$, where we try to find the $\mathbf{P}$ permutation matrix that achieves the best alignment of the eigenvectors, where $\mathbf{C}$ is the similarity (cost) matrix of the eigen-vectors. Given two graphs $\mathcal{A}$ and $\mathcal{B}$ with adjacencies $\mathbf{A} = \sum_i \lambda_i \mathbf{u}_i \mathbf{u}_i^T$ and $\mathbf{B} = \sum_j \mu_j \mathbf{v}_j \mathbf{v}_j^T$, where $\lambda_i, \mathbf{u}_i$ are the eigenvalues and eigenvectors of $\mathbf{A}$ and $\mu_j, \mathbf{v}_j$ those of $\mathbf{B}$, Umeyama [70] defined his cost function as $\mathbf{C} = \sum_{i,j} s_i \mathbf{u}_i \mathbf{v}_i^T$, for some appropriately chosen signs $s_i \in \{\pm 1\}$ (similar to Xu and King [80]). The motivation being that, this is the solution to the orthogonal relaxation of (14), where the feasible set is relaxed to the set the orthogonal matrices [21]). Furthermore Umeyama [70] suggested the construction $\mathbf{C} = \sum_{i,j} |\mathbf{u}_i||\mathbf{v}_i|^T$, where $|\mathbf{u}_i|$ is the element wise absolute value of $\mathbf{u}_i$. Kazemi et al. [30] and Feizi et al. [19] proposed other low rank cost functions, the simplest of which is $\mathbf{C} = \mathbf{u}_1 \mathbf{v}_1^T$. Fan et al. [18] proposed GRAMPA, suggesting that using a combination of all eigenvectors rather than only those where $i = j$ was significantly more

resilient to noise and proposed $\mathbf{C} = \sum_{i,j} w(\lambda_i, \mu_j)\mathbf{u}_i\mathbf{u}_i^T\mathbf{J}\mathbf{v}_j\mathbf{v}_j^T$, where $w(\lambda, u)$ is a monotonically decreasing function such as the Cauchy spectral weight $w(\lambda, \mu) = \frac{1}{(\lambda-\mu)^2+\gamma^2}$, γ being a bandwidth parameter.

Two other notable Spectral methods related to 2D-GEM are Fast Sinkhorn Filters [55] and Structured Regularization of Functional Map Computations [57]. [55] provides a unifying framework for spectral methods. Following the Meta Algorithm from [55], 2D-GEM can equally be seen as a spectral alignment regularised algorithm. [55] proposed the Sinkhorn filter instead of the KNN and Auction (and GMWM) matching. It is worth considering to incorporate the sinkhorn filter into 2D-GEM. Concerning the differences, besides the difference in the motivation for Fast Sinhorn Filters (Regularized optimal transport) and 2D-GEM (spectral graph theory), the main differences between them are first the fact that the cost matrix proposed in Fast Sinkhorn Filters is a distance matrix (euclidean) between the eigen-vectors, while that in 2D-GEM is based on the spectrum of the graphs of the corresponding shapes. Second, rather than using single eigen-vectors on each shape, 2D-GEM uses pairs of eigen-vectors for its cost matrix. Finally, the regularizers used for 2D-GEM enforce both neighbourhoods preservation (via the 2-Hop algorithm) and geodesic distortion preservation (via the LMD), while the regularizer in fast Sinkhorn Filters is mainly to make the problem strictly convex which gives the interpretation of a regularized transport plan. [57] pointed out the fact that only penalizing the failure of the unknown map to commute with the Laplace-Beltrami operator operators on the source and target shapes may fail in some cases. To address this problem, [57] further proposed additional regularizers to update the functional map (namely, a descriptor-commutative regularizing term, an orientation-preserving term, and a resolvent term). 2D-GEM may be used to attempt to bridge the gap between the popular Functional Map and the 2D-Graph convolution. In this regard, 2D-GEM can also be interpreted as a regularised spectral alignment method to update the map. Thus, it is related to [57] with the difference that the regularizers in our case are neighborhood preservation (enforced via the 2-Hop) and distortion preservation (enforced via the LMD) in addition to the fact that the map in our case is computed in the spectrum of the graphs associated with the shape's triangulation rather than in the traditional shape LBO's spectrum.

Quadratic Programming (QP), Doubly Stochastic (DS) and Semi Definite Programing (SDP) Methods. Here we have other relaxations of (14) e.g., denoting the set of permutation matrices in $\mathbb{R}^{n\times n}$ by $\mathbf{P}$, notice that (14) can be reformulated as:

$$\min_{\mathbf{P}\in\Pi} ||\mathbf{AP} - \mathbf{PB}||_F^2 \tag{15}$$

Relaxing the set of permutation matrices Π to its convex hull (the Birkoff polytope of doubly stochastic matrices $\mathcal{P}_B$), leads to the doubly stochastic relaxation [67, 1]. Other methods here [16] add different regularization terms to (15) or use different relaxations like the Linear Programming (LP) [3] or the Semi Definite Programming (SDP) [31]. GRAMPA can also be placed in this category as its cost matrix is a solution to the regularized QP.

Neural Network Methods. Various Graph neural networks (GNNs) have been proposed for various graph related tasks [33, 34, 24, 71, 69, 43, 73, 85, 76, 79, 73, 22]. It was recently shown that these GNNs can work as the Weisfeiler-Lehman [79, 15]. It was shown that for the graph isomorphism task amongst these GNNs, those that update a node's representation as a sum of itself and its neighbours features work better than those which use other aggregation techniques. As such, many of these GNNs have been used for the graph matching and graph similarity tasks [83, 25, 86, 74, 75, 8, 9, 45]. Many of these methods leverage the graph convolution for the shape matching task as well. However, they perform the convolution on the features of each individual 3D shape as a way to combine the pair-wise and point-wise descriptors of each of the shapes and then feed these combined features into a learning algorithm for supervised or unsupervised alignment using a suitable loss function [20]. Our method on the other hand is unsupervised and does not use neural nets, and our proposed filter is derived from our new definition of noise in the context graph matching and can be regarded as a 2D graph convolution over the point-wise descriptor similarity space or over the initialized correspondence.

Other Methods. Other related methods for the graph alignment task include; Symmetric-Non-Negative-Matrix-Factorization methods for graph and bipartite graph matching [13, 38]. We equally have the popular Graduated Assignment [23] in this category, as well as the ISORANK algorithm [66] which is based on the adjacency matrix of product graphs. Kollias et al. [37] proposed Network Similarity Decomposition (NSD) to approximate ISORANK by using decomposition techniques,

thus avoiding the quadratic size of the product graph. In "noisy" settings where the graphs are not exactly isomorphic, there have been many proposed algorithms [81, 49, 50, 14] besides GRAMPA.

A.3 Joint Alignment (Both)

Using some of the aforementioned pair-wise descriptors $\mathbf{A} \in \mathbb{R}^{n_1 \times n_1}$ and $\mathbf{B} \in \mathbb{R}^{n_2 \times n_2}$, and point-wise descriptors $\mathbf{X}_A \in \mathbb{R}^{n_1 \times d}$ and $\mathbf{X}_B \in \mathbb{R}^{n_2 \times d}$ for two 3D shapes $\mathcal{M}_1$ and $\mathcal{M}_2$ respectively , these methods are either of the form:

$$\min_{\mathbf{P} \in \Pi} ||\mathbf{P}\mathbf{A}\mathbf{P}^T - \mathbf{B}||_F^2 + \alpha ||\mathbf{P}\mathbf{X}_A - \mathbf{X}_B||_F^2, \tag{16}$$

such as SEQA, where the parameter α acts as a trade of parameter between the *pair-wise alignment* and the *point-wise alignment*. Or of the form:

$$\max_{\mathbf{P} \in \Pi} \mathbf{Tr}(\mathbf{P}^T\mathbf{A}\mathbf{P}\mathbf{B}) + \alpha \mathbf{Tr}(\mathbf{P}^T\mathbf{X}_A\mathbf{X}_B^T), \tag{17}$$

such as Kernel-Match. The first term in both (16) and (17) will normally lead to the QAP which is computationally intractable, but to avoid this, these two methods relax $\mathbf{P}$ to be in the set of doubly stochastic matrices $\mathcal{P}_B$ rather the set of permutation matrices Π. One can observe that methods in this category try to find a trade-off between the contribution of the point-wise descriptors and those of the pair-wise descriptors to the assignment problem. Moreover these methods have traditionally been shown [40, 77] to outperform *pair-wise alignment* and *point-wise alignment* methods. Hence, inspired by the performance of these methods, we proposed to leverage the 2D graph convolution instead for better performance at a relatively good time complexity as it equally leads to an LAP rather than the QAP.

Other methods in this category usually initialize the solution $\mathbf{P}$ from point-wise alignment by solving (13), and then iteratively update the solution using a relaxation of pair-wise alignment e.g., using a truncated basis space via the notion of Functional Maps [54]. These methods include; ICP [4], ZoomOut [52], and others [56, 62].

For nearly-isometric shapes, the approach used by most of these methods (i.e., first finding an initial correspondence, and then iteratively improving this initial correspondence using all points and eigen-vectors corresponding to lower frequencies (eigen-values)), leads to an erroneous final correspondence due to accumulation of errors in the updates given that the initial correspondence is usually not exact [78]. Moreover, these methods are usually not very noise robust as they usually assume near-isometry between the shapes, thus, they equally do not perform well on the task of exact matching on non-isometric shapes. Hence, for 2D-GEM we propose a 2D Graph convolution-based framework that is robust to noise on non-isometric shapes, and can incorporate more tight constraints (such as preservation of geodesic distances) to reduce the error accumulation in the correspondence update on near-isometric shapes.

B GRAMPA for Pair-wise Alignment

In this section we review GRAMPA [18] for 3D shape matching via *pair-wise alignment*. This algorithm consists of two parts; (1) Formulation/Initialization of the permutation matrix, and (2) iterative improvement of the initialized correspondence.

B.1 Formulation

In graph theory, an isomorphism of graphs $\mathcal{A}$ and $\mathcal{B}$ is a bijection between the vertex sets of $\mathcal{A}$ and $\mathcal{B}$ i.e., $f\colon V(\mathcal{A}) \to V(\mathcal{B})$, such that any two vertices u and v of $\mathcal{A}$ are adjacent in $\mathcal{A}$ if and only if $f(u)$ and $f(v)$ are adjacent in $\mathcal{B}$. If an isomorphism exists between two graphs, then the graphs are called isomorphic and denoted as $\mathcal{A} \simeq \mathcal{B}$. From linear algebra, two graphs $\mathcal{A}$ and $\mathcal{B}$ are isomorphic iff the adjacency matrix of one can be expressed as the permutation of the rows and columns of the other i.e.,

$$\mathbf{A} = \mathbf{P}^T\mathbf{B}\mathbf{P}, \tag{18}$$

where , and $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$ are the adjacency matrices of $\mathcal{A}$ and $\mathcal{B}$ respectively. As such, the graph matching problem [42, 20, 74, 18, 12, 17] is the attempt at finding the permutation matrix $\mathbf{P}$ such that (18) holds.

This problem has several applications which include but are not limited to action identification in computer vision [72], protein network matching in bioinformatics[64, 66], comparing molecules in cheminformatics [36], and linking user accounts (de-anonymization) in social network analysis [84]. GRAMPA was proposed to address this problem in a noisy setting, where both graphs may not be exactly isomorphic.

For 3D shape matching, graph matching can be seen as a *pair-wise alignment* problem in which the set of pair-wise descriptors are the adjacency matrices. Using the adjacency matrices $\mathbf{A} \in \mathbb{R}^{n_1 \times n_1}$ and $\mathbf{B} \in \mathbb{R}^{n_2 \times n_2}$ for two 3D shapes $\mathcal{M}_1$ and $\mathcal{M}_2$ respectively, the alignment problem is given as the Quadratic Assignment Problem (QAP) [21] as in equation (14).

Several relaxations of this problem have been proposed [63, 41, 48, 70, 30, 19, 67, 1, 16, 3, 31]. Amongst the Spectral Relaxations, GRAMPA has been proven to achieve the SOTA results, being tobust to high noise levels[3]. GRAMPA's relaxed alignment is $\max_{\mathbf{P}} \mathbf{Tr}(\mathbf{P}^T\mathbf{C})$, where $\mathbf{C}$ is their cost matrix given as:

$$\mathbf{C} = \sum_{i,j} w(\lambda_i, \mu_j)\mathbf{u}_i\mathbf{u}_i^T\mathbf{J}_{A,B}\mathbf{v}_j\mathbf{v}_j^T, \tag{19}$$

$\mathbf{Tr}(\cdot)$ is the matrix Trace operator, $w(\lambda, \mu) = \frac{1}{(\lambda-\mu)^2+\gamma^2}$, and γ the noise control parameter. This cost matrix $\mathbf{C}$ in (19) was shown by Fan et al. [18] to be the solution $\mathbf{X}$ to the regularized relaxed QAP:

$$\min_{X \in \mathbb{R}^{n \times n}} \frac{1}{2}||\mathbf{AX} - \mathbf{XB}||_F^2 + \frac{\gamma^2}{2} + ||\mathbf{X}||_F^2 - \mathbf{1X1}, \tag{20}$$

first relaxing the set of permutations Π to its convex hull (the Birkhoff polytope of doubly stochastic matrices), and further relaxing to only be required to satisfy the constraint $\mathbf{1X1} = n$. The proof being tedious, we refer the reader to their paper [18].

In the case where $n_1 \neq n_2$, the permutation matrix $\mathbf{P} \in \mathbb{R}^{n_2 \times n_1}$ is adjusted to be a rectangular assignment matrix. This problem has been widely addressed not only on 3D shapes, but also generally on graphs.

B.2 2-Hop Iterative improvement

Once the permutation matrix $\mathbf{P}$ is obtained, GRAMPA proposed to use the iterative cleanup procedure: $\mathbf{P}_{t+1} = \arg\max_{\mathbf{P} \in \Pi} \mathbf{Tr}(\mathbf{P}^T\mathbf{A}\mathbf{P}_t\mathbf{B})$. More recently it has been suggested to use j-hop neighbourhood adjacency matrices $\mathbf{A}_j$ and $\mathbf{B}_j$ for the iterative improvement instead [82]. Specifically, YU et al. [82] suggested to use the 2-hop neighborhood and gave performance guarantees. The 2-hop adjacency matrices $\mathbf{A}_2$ and $\mathbf{B}_2$ (using the Delauney triangulations), are such that $[\mathbf{A}_2]_{i,j} = 1$ and $[\mathbf{B}_2]_{i,j} = 1$ if i and j are 2-hop neighbors, and 0 otherwise. Using these, the iterative improvement becomes $\mathbf{P}_{t+1} = \arg\max_{\mathbf{P} \in \Pi} \mathbf{Tr}(\mathbf{P}^T\mathbf{A}_2\mathbf{P}_t\mathbf{B}_2)$.

B.3 Relationship Between 2D-GEM and GRAMPA

Looking at GRAMPA's cost function in equation (19), one can see that using the adjacency instead of the Laplacian, as well as using no features, 2D-GEM as in equation (7) becomes exactly GRAMPA with a similar but different filter $w(\lambda_i, \mu_j)$. As such, GRAMPA can also be seen as filtering [61] the initialized indistinguishable 2D signals of graphs using the adjacency basis and a graph-matching-suitable-filter (i.e., $w(\lambda, \mu)$), and then iteratively updating the correspondence i.e., GRAMPA is also a filter (de-noise), then initialize, then update algorithm. However, 2D-GEM as a 2D graph convolution allows us to utilise features (such as the *point-wise* descriptors in the context of shape matching), which causes 2D-GEM to achieve better performance than GRAMPA when these features are informative. Besides comparison on the 3D shape matching task, we equally do more empirical comparisons between 2D-GEM and GRAMPA on random graphs following the Erdos Regny model (see Appendix D). These experiments show that indeed when we have features that are informative 2D-GEM may not only outperform GRAMPA, but also the QP (i.e., the full Quadratic Assignment Problem with full doubly stochastic constraints [18]) depending on how informative the features are.

[3] we refer the reader to their paper [18].

C On the In-variance of the Graph Convolution to Permutation

In this section we want to review graph isomorphism from linear algebra and revisit the in-variance of graph convolution to permutation.

C.1 Review of Isomorphism from Linear algebra

From linear algebra, to graphs $\mathcal{A}$ and $\mathcal{B}$ are isomorphic iff the adjacency matrix of one can be expressed as the permutation of the rows and columns of the other i.e.,

$$\mathbf{A} = \mathbf{P}^T\mathbf{B}\mathbf{P}, \tag{21}$$

where $\mathbf{P} \in \mathbb{R}^{n\times n}$ is a permutation matrix, and $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times n}$ are the adjacency matrices of $\mathcal{A}$ and $\mathcal{B}$. This implies that the diagonal degree matrix $\mathbf{D}_A$ of $\mathbf{A}$ is also a permutation of the diagonal degree matrix $\mathbf{D}_B$ of matrix $\mathbf{B}$:

$$\mathbf{d}_A = \mathbf{A}\mathbf{1} = \mathbf{P}^T\mathbf{B}\mathbf{P}\mathbf{1} = \mathbf{P}^T\mathbf{d}_B, \tag{22}$$

where $\mathbf{d}_A, \mathbf{d}_B \in \mathbb{R}^n$ are the vectors of degree, $\mathbf{1} \in \mathbb{R}^n$ is the all ones vector, $\mathbf{P}\mathbf{1} = \mathbf{1}$. This can be written in matrix form as:

$$\mathbf{D}_A = \mathbf{P}^T\mathbf{D}_B\mathbf{P}, \tag{23}$$

where the extra $\mathbf{P}$ on the left of $\mathbf{D}_B$ ensures that the permutation keeps the node degree with the corresponding node index of the rows and columns of $\mathbf{D}_B$. One can observe that a linear combination of (21) and (23) will imply that isomorphism also implies that the Laplacian matrix of $\mathcal{A}$ is also a permutation of that of $\mathcal{B}$, i.e., taking (23) - (21), we have:

$$\begin{aligned}\mathbf{L}_A = \mathbf{D}_A - \mathbf{A} &= \mathbf{P}^T\mathbf{D}_B\mathbf{P} - \mathbf{P}^T\mathbf{B}\mathbf{P}\\ &= \mathbf{P}^T(\mathbf{D}_B - \mathbf{B})\mathbf{P} = \mathbf{P}^T\mathbf{L}_B\mathbf{P}\\ &\rightarrow \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T = \mathbf{P}^T\mathbf{V}\mathbf{M}\mathbf{V}^T\mathbf{P}\\ &\rightarrow \mathbf{U}g(\mathbf{\Lambda})\mathbf{U}^T = \mathbf{P}^T\mathbf{V}h(\mathbf{M})\mathbf{V}^T\mathbf{P},\end{aligned} \tag{24}$$

because $\mathbf{\Lambda} = \mathbf{M}$ for isomorphic graphs, we also have $g(\mathbf{\Lambda}) = h(\mathbf{M})$ where $\mathbf{L}_A \in \mathbb{R}^{n\times n}$ and $\mathbf{L}_B \in \mathbb{R}^{n\times n}$ are the Laplacians of $\mathcal{A}$ and $\mathcal{B}$ respectively, $\mathbf{\Lambda}, \mathbf{M} \in \mathbb{R}^{n\times n}$ the eigen-value diagonal matrices, $g(\mathbf{\Lambda})$, $h(\mathbf{M})$ functions of the eigen-values (where g and h are the same function e.g., $g(\lambda) = 1 - \alpha\lambda$, and $h(\mu) = 1 - \alpha\mu$), and $\mathbf{U}, \mathbf{V}$ and the eigen-vectors of the Laplacians $\mathbf{L}_A$ and $\mathbf{L}_B$ respectively.

Moreover, if $\mathcal{A}$ and $\mathcal{B}$ are associated with node attributes $\mathbf{X}_A \in \mathbb{R}^{n\times d}$ and $\mathbf{X}_B \in \mathbb{R}^{n\times d}$, one can also observe that:

$$\mathbf{X}_A = \mathbf{P}^T\mathbf{X}_B, \tag{25}$$

i.e., there exists a permutation of the node attribute matrix of $\mathcal{B}$ that reorders it into the node attribute matrix of $\mathcal{A}$. However, one should note that here isomorphism implies equation (25) holds, but equation (25) holding does not necessarily imply isomorphism. So, here we can be seen as using some form of relaxation. Notice also that this relaxation is reasonable (and made by other baselines [40]) as in real life meaningful features often lead to some level of information on the permutation matrix.

Putting (24) and (25) together, we have the in-variance of the graph convolution to permutation [20], i.e., for any convolution function h that allows the diagonalization $h(\mathbf{L}_B) = \mathbf{V}h(\mathbf{M})\mathbf{V}^T$ we have that:

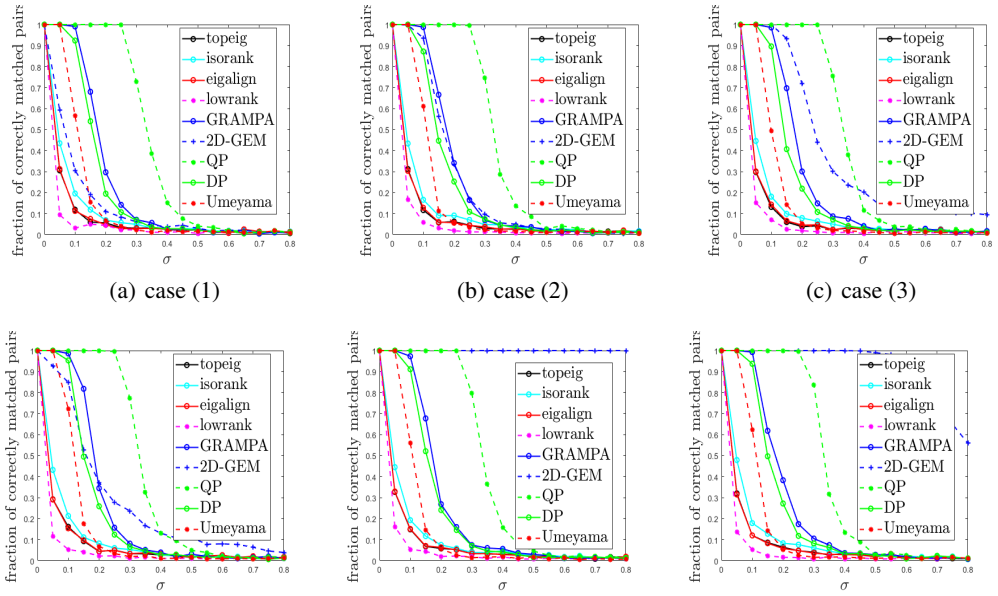
$$\mathbf{P}^T\mathbf{V}h(\mathbf{M})\mathbf{V}^T\mathbf{P}\mathbf{P}^T\mathbf{X}_B = \mathbf{P}^T\mathbf{V}h(\mathbf{M})\mathbf{V}^T\mathbf{X}_B \tag{26}$$

where $\mathbf{P}\mathbf{P}^T = \mathbf{P}^T\mathbf{P} = \mathbf{I}$ for permutation matrices $\mathbf{P}$. As such we have that for isomorphism,

$$\begin{aligned}\mathbf{U}g(\mathbf{\Lambda})\mathbf{U}^T\mathbf{X}_A &= \mathbf{P}^T\mathbf{V}h(\mathbf{M})\mathbf{V}^T\mathbf{X}_B\\ &\rightarrow \mathbf{P}\mathbb{M}_A\mathbf{X}_A = \mathbb{M}_B\mathbf{X}_B,\end{aligned} \tag{27}$$

where $\mathbb{M}_A = \mathbf{U}g(\mathbf{\Lambda})\mathbf{U}^T = g(\mathbf{L}_A)$ is the convolution function for graph $\mathcal{A}$, and likewise $\mathbb{M}_B = \mathbf{V}h(\mathbf{M})\mathbf{V}^T = h(\mathbf{L}_B)$ is the convolution function for graph $\mathcal{B}$. As such the graph matching problem can be reformulated using the convolution as:

$$\min_{\mathbf{P}} ||\mathbf{P}\mathbb{M}_A\mathbf{X}_A - \mathbb{M}_B\mathbf{X}_B||_F^2, \tag{28}$$



(a) case (1) (b) case (2) (c) case (3)

(d) case (4) (e) case (5) (f) case (6)

Figure 6: Plots on Erdos Regny random graphs, using $n = 100$ and $p = 0.5$. All plots follow the same legend as case (1), Cases (1)-(2) use undistinguishable features i.e., the all ones as in equation (5)), cases (3)-(4) use features such that 10% of the nodes are distinguishable, and cases (5)-(6) such that 100% of the nodes are distinguishable. Cases (1)-(3), and (5) use all eigen-values, while cases (4) and (6) use 20 eigen-values as in equation (8). All cases use $e = 10$ except case (1) where $e = 1$.Nodes with distinguishable features are chosen at random.

D Validation of 2D Graph Convolution for Graph Matching on Erdos Regny Random Graphs

In this section we conduct experiments to validate the effectiveness of utilizing features available via 2D graph convolution for graph matching. We compare the 2D graph convolution (referred to as 2D-GEM also in this Section, where we do not use any LMD nor 2-hop refinement, but just the cost in equation (7) or equation (8) and then the normal rounding procedure "matchpairs" provided by Matlab) with GRAMPA and few other methods (which all cannot utilize features). We make use of the $\mathcal{G}(n, q)$ Erdos Regny random graphs model, where n determines the number of nodes and q the number of edges in the graph.

D.1 Datasets

Given a parent Erdos Regny random graph $\mathcal{G}(n, q)$ with adjacency $\mathbf{G}$, we first sample a graph from it which has the same number of nodes n but deleting edges (each independently) from it with probability $1 - s$, where $s \in [0, 1]$ to obtain a source graph with adjacency $\mathbf{A} \sim \mathcal{G}(n, p)$, where $q \triangleq ps$. We then define another graph (target graph) with adjacency $\mathbf{B} \sim \mathcal{G}(n, p)$ as:

$$\mathbf{B}_{i,j} \sim \begin{cases} \text{Bern}(s) & \text{if } \mathbf{A}_{ij} = 1 \\ \text{Bern}\left(\frac{p(1-s)}{1-p}\right) & \text{if } \mathbf{A}_{ij} = 0 \end{cases}, \tag{29}$$

where $\text{Bern}(s)$ denotes a Bernoulli distribution with mean s. Note that $\sigma^2 = 1 - s$ can be seen as the square of the noise level determining the number of edges that differ between $\mathbf{A}$ and $\mathbf{B}$. Suppose we observe a pair of graphs with adjacency matrices $\mathbf{A}$ and $\mathbf{B}' = \mathbf{P}^T\mathbf{B}\mathbf{P}$, where $\mathbf{P}$ is the permutation matrix. We then wish to recover $\mathbf{P}$.

D.2 Evaluation Metrics

For these experiments on random Erdos Regny graphs, we visualise the $\%$ of correct correspondences against the noise level σ.

D.3 Baselines

We compare 2D-GEM to GRAMPA [18], and others such as: Umeyama [70], Degree Profile (DP) [14], ISORANK [29], QP (where we follow [18] and use the relaxation of the permutation matrix in the Quadratic Assignment Problem to a doubly stochastic matrix), EigenAlign [19], LowrankAlign [19] and TopEigenVec [18].

D.4 Parameter Settings

We use graphs with $n = 100$ and set $p = 0.5$ and σ starting from 0, and using a step size of 0.05 up to 0.8. We repeat the experiments 10 times each and each time we generate features for the graphs we generate features $\mathbf{X}_A$ for the graph with adjacency $\mathbf{A}$ and use $\mathbf{X}_B = \mathbf{P}^T\mathbf{X}_A$ as features for $\mathbf{B}$.

We conduct four different experiments and run each 10 times and plot the averages. The details for each setting for 2D-GEM are as follows (the other baselines are set following Fan et al. [18] who kindly provided their source code):

- case (1): for figure (6(a)) we use indistinguishable features $\mathbf{J}_{A,B}$, and use all the eigen-values for 2D-GEM as in equation (5) with $e = 1$,
- case (2): for figure (6(b)) we use indistinguishable features $\mathbf{J}_{A,B}$, all eigen-values and use $e = 10$,
- case (3): for figure figure (6(c)) we use features such that 10% of randomly selected nodes are distinguishable (i.e., 10% of nodes in both graphs can be matched using these features) and use all the eigen-values for 2D-GEM as in equation (7), and $e = 10$
- case (4): for figure figure (6(d)) we use features such that 10% of randomly selected nodes are distinguishable and use 20 eigen-values (instead of $n = 100$ eigen-values) for 2D-GEM as in equation (8), and $e = 10$
- case (5): for figure figure (6(e)) we use features such that all nodes are distinguishable and use $e = 10$ and all eigen-values for 2D-GEM as in equation (7),
- case (6): for figure figure (6(f)) we use features such that all nodes are distinguishable and use $e = 10$ and 20 eigen-values (instead of $n = 100$ eigen-values) for 2D-GEM as in equation (8),

D.5 Performance Analysis

Cases (1) and (2) show that without features, 2D-GEM with $e = 1$ performs worse than GRAMPA and even other methods such as DP and Umeyama, while with $e = 10$, 2D-GEM performs on par with GRAMPA. This trend is in fact general and also occurs in other cases i.e., the performance of 2D-GEM increases as e increases. Cases (3)-(6) illustrate the fact that if we have features which are a little informative even as little as 10% (case (3)-(4)), 2D-GEM can outperform even GRAMPA, and if the features are fully informative (case (5)-(6)), 2D-GEM using all eigen-values (case (5)) can recover the exact permutation matrix no mater the noise level in the adjacency, while 2D-GEM using as little as 20 eigen-values also performs significantly well (case (6)).

These experiments demonstrate that using features in addition to the adjacency/Laplacian (i.e., *point-wise* descriptors together with *pair-wise* descriptors) will be more beneficial than using only either. Moreover, these experiments also show that it is non-detrimental to assume that the permutation of the adjacency and those of the features correspond, as even when the features a non-informative such as in cases (1)-(2) where we used the all ones, the performance of 2D-GEM was on par with GRAMPA, and the performance of 2D-GEM only got better as the features became more informative (cases (3)-(6)). Moreover, in real life, most graphs are associated with features and for 3D shapes, one could extract *point-wise* descriptors as well as *pair-wise* descriptors. A caution here is the case where the information contained in both do not agree with each other, though we did not test this case, it is obvious that in such a case the permutation matrix will be different.

Algorithm 3 : case (3)

1. **input** An initial correspondence $\mathbf{P}^0$, $\mathbf{U}_k$, $\mathbf{V}_k, \mathbf{\Lambda}_k, \mathbf{M}_k, e$, maximum iteration t for 2-hop neighborhood algorithm.

while $0 < i \leq iter$ **do**

2. Update $\mathbf{P}^i$ using 2-Hop with algorithm 2, where $t = 1$,
3. De-noise and update correspondence:
   $\mathbf{P}^i = \arg\max_{\mathbf{P}} \mathbf{Tr}(\mathbf{PC})$,
   where $\mathbf{C} = \sum_{i,j}^{k} w(\lambda_i, \mu_j)\mathbf{u}_i\mathbf{u}_i^T\mathbf{P}^i\mathbf{v}_j\mathbf{v}_j$,

end while

4. **return** $\mathbf{P}^t$.

E Ablation Studies on 2D-GEM

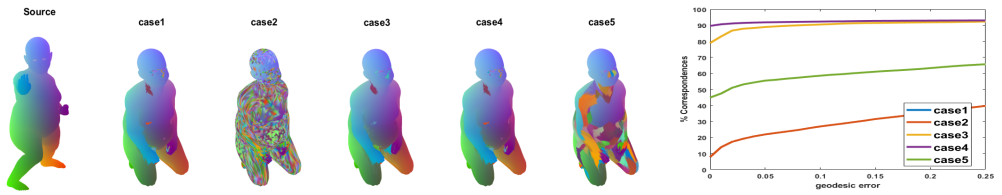
Here we conduct ablation studies on 2D-GEM to show the effectiveness of the overall framework over its individual parts.

E.1 Set-up

We compare 5 different cases (scenarios). For all cases, we use the same parameters as in Section 5.5, except that for all datasets we use 500 LBO eigen-vectors.

- case (1): we use 2D-GEM as it is,
- case (2): 2D-GEM without the 2-Hop improvement i.e., without step (2) in algorithm 1,
- case (3): 2D-GEM without the LMD. As in algorithm 3, with $iter = 60$,
- case (4): 2D-GEM with only the 2-Hop algorithm i.e., without the LMD nor the 2D graph convolution. As in algorithm 3 without step (3),
- case (5): 2D-GEM with only the 2D graph convolution i.e., without the LMD nor the 2-Hop algorithm. As in algorithm 3 without step (2).

E.2 Comparison

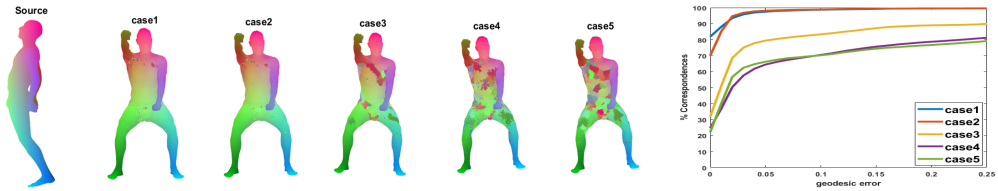


(a) Sample on TOPKIDS (b) Plot of sample on TOPKIDS

Figure 7: Ablation studies on non-isometric dataset (TOPKIDS). In Figure 7(b) above the plots of case 1 and 4 overlap)

Ablation on non-isometric shapes. One can observe from Figure 7(b) that cases (1) and (4) have exactly the same performance, since the plot of case (1) is shadowed by that of case (4). This suggests that for non-isometric shape matching, given a good initialization, the 2-Hop algorithm is as strong as 2D-GEM' framework (However, this is not the case on nearly-isometric shapes as seen in Figure 8(b) where it performs much worse than 2D-GEM). This equally shows that the 2D graph convolution provides such a good initialization. Moreover, it can be seen on this plot that the 2D graph convolution by itself (i.e., case (5)) performs moderately at geodesic 0 though it produces non-smooth maps. On the other hand, it can be observed that 2D-GEM with only the 2D convolution and the 2-Hop algorithm (i.e., case (3)) performs a little worse than 2D-GEM as a whole and than the 2-Hop algorithm by itself. It can also be seen on this plot that the LMD by itself (case (2) is really not suited for non-isometric shapes as for these shapes geodesic distances are not preserved.

Ablation on isometric shapes. Figure 8(b) shows that 2D-GEM as a whole (i.e., case (1)) performs best once again, and this time much better than the 2-Hop algorithm by itself (i.e., case (4)) since it



(a) Sample on SCAPE (b) Plot of sample on SCAPE

Figure 8: Ablation studies on isometric dataset (SCAPE).

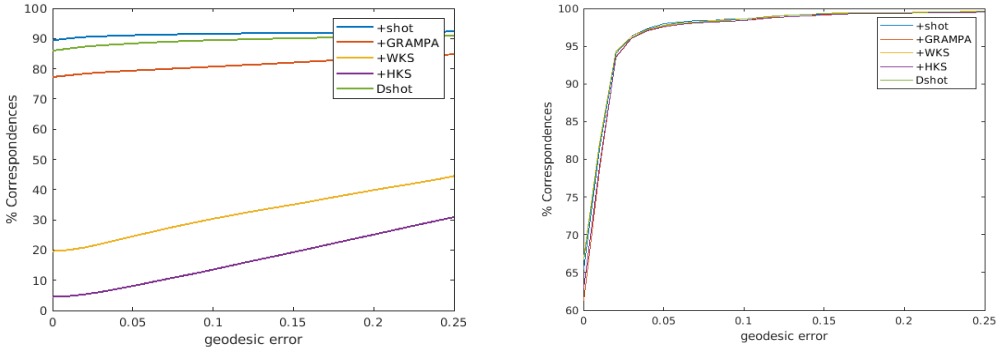
assumes conservation of geodesic distances (a constraint not accounted for by the 2-Hop algorithm). Moreover, the 2D graph convolution by itself (i.e., case (5)) is also seen to be on par and even outmatching the 2-Hop algorithm, while being outmatched by the combination of both (i.e., case (3)). Finally, it can equally be seen that the LMD (i.e., case (2)) is quite good for isometric shapes as it takes into account the preservation of geometric properties such as geodesic distances. However, the LMD (i.e., case (2)) is still outmatched in the exact matching task by 2D-GEM (i.e., case (1)) as can be seen on this figure at geodesic distance 0.

E.3 Ablation Conclusion

Overall from the ablation ablation studies, one can see that 2D-GEM helps one with the flexibility of incorporating stronger constraints such as preservation of geodesic distances or relaxing them, depending on the properties of the shapes at hand. Moreover, one can also see that as on random graphs (see Section D), the 2D graph convolution can indeed serve as a good initialization.

F More Comments and Initialization Studies

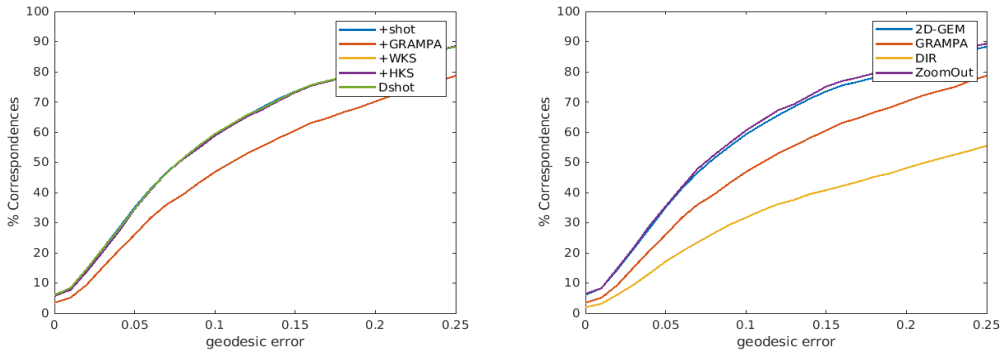
In this section we add some initialization studies for 2D-GEM 1. For these studies we used SCAPE, TOPKIDS and TOSCA-re-meshed [56], where TOSCA-re-meshed is the TOSCA dataset with each shape re-meshed to roughly 5,000 vertices (for TOSCA-re-mershed, we use the TOSCA isometric [56]).



(a) Initializations on TOPKIDS (b) Initializations on SCAPE

Figure 9: 2D-GEM with different initializations on 10 shapes from each dataset, i.e., 10 from TOPKIDS, and 10 from SCAPE.

For figures 9(a), 9(b), 10(b) and 10(a), we used 100 eigen-vectors for the Heat Kernel Signature (+HKS) and Wave Kernel Signatures (+WKS). For 2D-GEM we used 500 eigen-vectors, maxiter 60, LMD 100 for all iterations on non-isometric sample from TOPKIDS (figure 9(a)), maxiter 60 and LMD thresholds [0.5, 0.45, 0.4, 0.35, 0.3] for non-isometric shapes (figures 9(b), where we repeat 0.3 for the rest of the iterations beyond 5), and LMD 100 and maxiter 60 for the remeshed shape (figures 10(b) and 10(a)). DSHoT is directly using the shot feature (i.e., computing $X_A X_B^T$) and



(a) Initializations with LMD 100

(b) Comparison with LMD 100

Figure 10: 2D-GEM with different initializations on a sample shape from TOSCA-re-meshed as well as comparison with other methods on the same sample shape. The correspondence plots of the different initializations are very close to each other in figure 10(a) with the exception of GRAMPA which is a little lower. In figure 10(a) we used +LMD1 in Appendix H

using 2D-GEM as in equation 4 to initialize the map and use this initialized map in the algorithm in Section 4.

As shown in Appendix D, 2D-GEM is robust to initializations on graphs. When no features or initialization are present, 2D-GEM is shown to be comparable with the spectral baseline GRAMPA. However, when an informative initialization or features is present, 2D-GEM performs significantly better. On 3D-shapes, 2D-GEM equally tends to maintain a good performance despite the initializations as show in Figures 9(a), 9(b), and 10(a). This means that 2D-GEM is robust to initializations on 3D-shapes as well.

It can be seen that unlike on isometric shapes, the HKS and WKS descriptors donot perform well with 2D-GEM on non-isometric shapes. We attribute this to the well known fact that these descriptors are not robust to high levels of noise [40]. Though not shown in these plots, for non-isometric shapes when we concatenate these descriptors with the SHOT descriptors before using them in 2D-GEM, their performance is on par with the SHOT descriptors' performance on the plots.

As seen on figures 10(b) and 10(a), 2D-GEM seems not to significantly outperform existing baselines on the exact matching task on remeshed-shapes. This maybe due to the fact that a key component of 2D-GEM is the adjacency associated with the triangulation of the shapes. So, an interesting topic for future work is an in-depth investigation of the re-meshed shapes as well as shapes with very different triangulation such as in SHRECK19 [51].

G On Non-Smooth Final Maps Returned By 2D-GEM

Concerning the non-smooth final maps produced by the algorithm, we will indeed try to dive deeper into that in future extensions. However, for now, we note that it may arise due to the initial map being very poor and non smooth as pointed out by Consistent ZoomOut [26]. In most cases, this may be reduced to an extent by (a) either using 2D-GEM in a ZoomOut-like manner, i.e., starting with lower eigen-values and iteratively increasing them (though this strategy works in some cases, it may not always work as pointed out in [26]), or by (b) using more rounds of our pipeline, i.e., the LMD (since the LMD is an effective measure of the continuity and distortion of the map), the map update via convolution (for denoising the map), and the 2-Hop algorithm (since the 2-Hop algorithm also enforces neighbourhood preservation). See Figures 11 and 12 where the initial map given by shot was non-smooth.

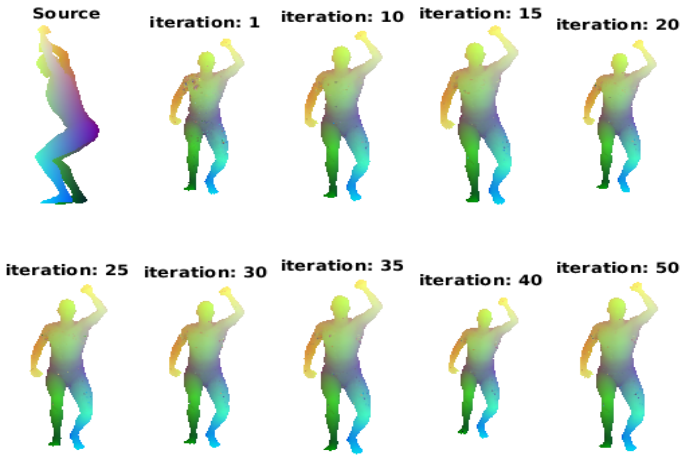


Figure 11: 2D-GEM's action on a non smooth initial map on SCAPE

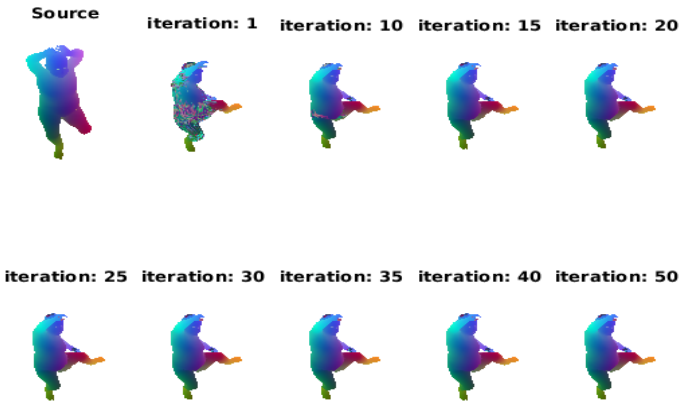


Figure 12: 2D-GEM's action on a non smooth initial map on TOPKIDS

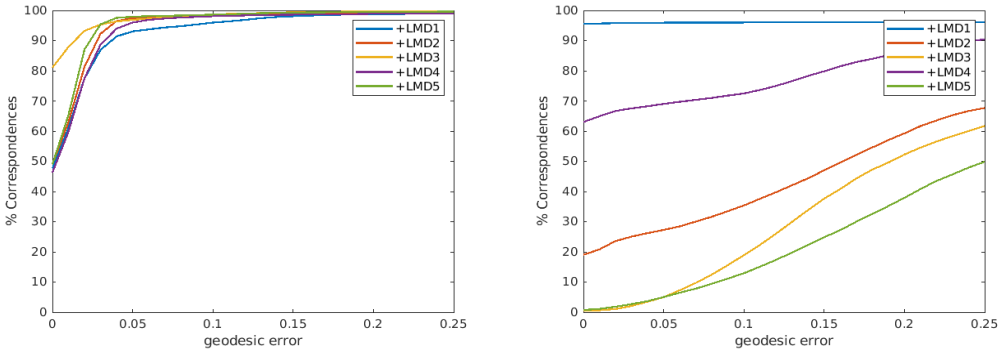
H 2D-GEM's Sensitivity to LMD Thresholds

As noted in Section 5, the LMD thresholds are set differently for 2D-GEM on isometric shapes and non-isometric shapes. Our choices of parameters were motivated by the properties of the shape. Namely, on non-isometric shapes where geometric properties are not preserved, we deactivate the constraint of enforcing preservation of geometric distortions by setting higher values for the LMD (i.e., updating the 2D spectrum with more anchor points) we used LMD 100 in Section 5. On isometric shapes, we set lower values, thereby enforcing this constraint. In this section, we conduct sensitivity studies on the LMD using an isometric pair of shapes, i.e., two random meshes from the SCAPE dataset as well as a non-isometric pair of shapes, i.e., two shapes from the TOPKIDS dataset.

For all figures in this section we used 60 iterations and set 5 different ranges of values for the LMD thresholds as follows:

- +LMD1 corresponds to using thresholds as $\epsilon_i = 100$ for all iterations.
- +LMD2 corresponds to using thresholds as ϵ_i =[100, 89, 78, 67, 56, 45, 34, 23, 12, 1] for the first 10 iterations and 1 for the rest of the iterations.

- +LMD3 corresponds to using thresholds as ϵ_i = [1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1] for the first 10 iterations and 0.1 for the rest of the iterations.
- +LMD4 corresponds to using thresholds as ϵ_i = [5.00, 4.6667, 4.3333, 4.0000, 3.6667, 3.3333, 3.00, 2.6667, 2.3333, 2.00] for the first 10 iterations and 2.00 for the rest of the iterations.
- +LMD5 corresponds to using thresholds as ϵ_i = [3.00, 2.7333, 2.4667, 2.20, 1.9333, 1.6667, 1.40, 1.1333, 0.8667, 0.60] for the first 10 iterations and 0.60 for the rest of the iterations.



(a) LMD thresholds effects on 2D-GEM on SCAPE (b) LMD thresholds effects on 2D-GEM on TOPKIDS

Figure 13: 2D-GEM with different LMD on a sample shape SCAPE and a sample shape on TOPKIDS.

Figures 13(a) and 13(b) show that +LMD3 achieves above 80% accuracy on isometric shapes. This shows that values from 1 and decreasing at a good rate after every iterations are good, which validates the use of the constraint of preserving geometric properties for isometric shape matching (as shown in DIR[78]). On non isometric shapes, +LMD1 achieves the best accuracy (above 95%), showing that geometric distortions are not preserved by non-isometrics, and so the LMD thresholds should be set high for these shapes.