HOPE: Shape Matching Via Aligning Different K-hop Neighbourhoods

Barakeel Fanseu Kamhoua[1], **Huamin Qu**[1]*,
[1]The Hong Kong University of Science and Technology

Abstract

Accurate and smooth shape matching is very hard to achieve. This is because for accuracy, one needs unique descriptors (signatures) on shapes that distinguish different vertices on a mesh accurately while at the same time being invariant to deformations. However, most existing unique shape descriptors are generally not smooth on the shape and are not noise-robust thus leading to non-smooth matches. On the other hand, for smoothness, one needs descriptors that are smooth and continuous on the shape. However, existing smooth descriptors are generally not unique and as such lose accuracy as they match neighborhoods (for smoothness) rather than exact vertices (for accuracy). In this work, we propose to use different k-hop neighborhoods of vertices as pairwise descriptors for shape matching. We use these descriptors in conjunction with local map distortion (LMD) to refine an initialized map for shape matching. We validate the effectiveness of our pipeline on benchmark datasets such as SCAPE, TOSCA, TOPKIDS, and others.

1 Introduction

Shape matching is a very important task and has been increasingly so with the increase in the availability and affordability of 3D scans [31]. It has important applications including but not limited to shape registration [2, 37], comparison [1, 13], recognition [6], and retrieval [50].

Shapes can undergo different types of transformations which we will group as isometric and non-isometric transformations in this work. Nonrigid isometric transformations preserve most of the geometric properties of the shape i.e., after the transformation, its angles, geodesic distances, scale, connectivity, and other geometric properties are mostly preserved. On the other hand, most geometric properties are not preserved under nonrigid non-isometric transformations. Examples of isometric transformations are rotations and translations (though minute scaling may also be included in this category for nonrigid cases). While examples of non-isometric transformations are splitting, scaling, dilating, and others which change the geometric properties of the shapes.

Several works have been proposed to address shape matching. However, most of these methods will either aim for smoothness [12, 17, 19, 13, 1, 34, 39, 37, 44, 31, 20] or for accuracy [45], but not both. Moreover, most of these methods will focus on specific types of deformations undergone by the shape either isometric [12, 19, 13, 1, 34, 39, 37, 44, 31, 20, 49] or non-isometric [14, 22, 52]. Some methods such as 2D-GEM [22] have also tried to address both accuracy and smoothness in shapes undergoing both isometric and non-isometric deformations. However, 2D-GEM is very parameter-dependent, using different user-defined parameters for isometric shapes than for non-isometric shapes which in real life is not practical as one usually does not know whether one is dealing with isometric or non-isometric deformations.

Given these obsevations, there is a need for a pipeline that aims for accuracy and smoothness across different settings without significantly changing the pipeline. In this light, we propose HOPE, a k-hop

[*] Corresponding author: huamin@cse.ust.hk

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

neighborhood-based refinement technique. The rest of the paper is organized into; (a) preliminaries, (b) related work, (c) HOPE, and (d) experiments, (e) limitations and remarks, and (f) conclusion.

2 Preliminaries

3D shapes are usually represented by their coordinates $\mathcal{X} \in \mathbb{R}^{n\times 3}$ and their triangulations (meshes) which can be used to build an adjacency $\mathcal{A} \in \mathbb{R}^{n\times n}$, with $\mathcal{A}(i,j) = 1$ if vertices i and j are connected in the shape and 0 otherwise.

Given two three-dimensional shapes $\mathcal{M}$ and $\mathcal{N}$ with $n_\mathcal{M}$ and $n_\mathcal{N}$ vertices respectively. Though our proposed pipeline (HOPE) and some other baselines work with $n_\mathcal{M} \neq n_\mathcal{N}$, we will assume $n_\mathcal{M} = n_\mathcal{N} = n$, and use n for simplicity except when specified. The goal of nonrigid shape matching is to find a meaningful correspondence $\mathcal{T} : \mathcal{M} \to \mathcal{N}$, where $\mathcal{T}$ is (a) bijective, (b) continuous (smooth) , and (c) similar vertices should be matched to each other [26, 35, 22]. A measure that captures both the smoothness and the accuracy of the map is the geodesic error. The geodesic error measures how far a map $\mathcal{T}$, maps a vertex from its matching position given by the ground truth map $\hat{\mathcal{T}}$, where a higher map accuracy will correspond to a higher proportion of vertices having geodesic error 0, and a smoother map will correspond to a huge increase in the geodesic error curve as we move slightly away from error 0 (implying that several vertices though not accurate are mapped to their intended neighborhoods).

The two main components of shape matching involve: (1) initializing a map, and (2) refining the initialized map.

First, the map is initialized either by using landmarks vertices (i.e., vertices with known correspondences) or by using robust descriptors. These descriptors can be based on: (a) the spectrum of the shape laplacian [8, 4, 33], (b) The face normals, vertex location, and triangulation [45, 41], geodesic distances [44, 1] or others [14, 46, 15, 16, 11, 43, 23]. In this work, we will assume that one such initialization approach has been utilized and we have an initial map at hand.

Second, the initialized map $\mathcal{T}^0$ can then be refined by either using pair-wise descriptors [36, 26, 52, 48, 14, 38, 22] and solving:

$$\mathcal{T}^t = \arg\min_{\mathcal{T}^t} ||\mathcal{W}_\mathcal{M}(\mathcal{T}^t, \mathcal{T}^{t-1}) - \mathcal{W}_\mathcal{N}||, \tag{1}$$

or by using vertex-wise descriptors [2, 34, 39, 37, 31, 26, 35, 19, 20, 44, 1, 28] and solving:

$$\mathcal{T}^t = \arg\min_{\mathcal{T}^t} ||\mathcal{Q}_\mathcal{M}(\mathcal{T}^t, :) - \mathcal{Q}_\mathcal{N} f(\mathcal{T}^{t-1})||, \tag{2}$$

Where in equation 1, $\mathcal{W}_\mathcal{M}(\mathcal{T}^t, \mathcal{T}^{t-1}) \in (R)^{n\times n}$ is the pair-wise descriptors of vertices for shape $\mathcal{M}$ with its rows and columns aligned using the map $\mathcal{T}^t$ and the previous iterations map $\mathcal{T}^{t-1}$ respectively. $\mathcal{W}_\mathcal{N}$ is the pair-wise descriptor for shape $\mathcal{N}$. Here for conciseness for aligning rows or columns we will use $\mathcal{T}$ (while in reality if the row map is $\mathcal{T} = argmax(\mathcal{P}$, dim=-1) the column should be $\mathcal{T}^{'} = argmax(\mathcal{P}$, dim=-2) where P is the permutation matrix such that $\mathcal{P}\mathcal{Q}_\mathcal{M} = \mathcal{Q}_\mathcal{N}$ and $\mathcal{P}\mathcal{W}_\mathcal{M}\mathcal{P}^T = \mathcal{W}_\mathcal{N}$

While in equation 2, $\mathcal{Q}_\mathcal{M}(\mathcal{T}^t, :) \in \mathbb{R}^{n\times d}$ is the vertex-wise descriptors for shape $\mathcal{M}$ with feature dimension d for each vertex and with its rows aligned according to the map $\mathcal{T}^t$, $f(\mathcal{T}^{t-1})$ is a function (such as functional map[34]) to transfer the previous iteration's map to the descriptor space, and $\mathcal{Q}_\mathcal{N} \in \mathbb{R}^{n\times d}$ is the vertex-wise descriptors for shape $\mathcal{N}$.

For the map $\mathcal{T}$ refined by equations 1 or 2 to have a high accuracy (in the ideal case), the pair-wise descriptors $\mathcal{W}_\mathcal{N}(i, :) \in \mathbb{R}^{1\times n}$ or vertex-wise descriptors $\mathcal{Q}_\mathcal{N}(i, :) \in \mathbb{R}^{1\times d}$ will need to be: (a) unique for each vertex i on shape $\mathcal{N}$, and (b) identical to its ground truth corresponding pair-wise feature $\mathcal{W}_\mathcal{M}(\hat{\mathcal{T}}(i), :) \in \mathbb{R}^{1\times n}$ or vertex-wise feature $\mathcal{Q}_\mathcal{M}(\hat{\mathcal{T}}(i), :)$ on shape $\mathcal{M}$. Where $\hat{\mathcal{T}}$ is the ground truth map. Moreover, for smoothness, vertices should be mapped such that they remain in their relative neighborhoods before and after the mapping.

3 Related Work

ZoomOut [31] and other related works[19, 13, 1, 34, 39, 37, 26, 35] all aim for smoothness of the map and focus on the settings in which the shapes have undergone an isometric deformation, such

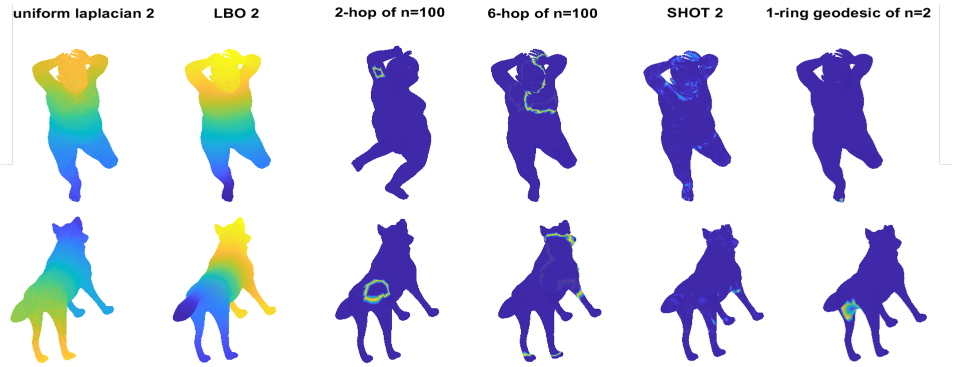


Figure 1: For sample shapes from the TOPKIDS (first row) and TOSCA (second row), this figure shows the second eigenvector of the mesh laplacian (LBO 2), the second eigenvector of the uniform shape laplacian from the triangulation adjacency (uniform shape laplacian 2), the 2-hop and 6-hop neighborhoods of vertex 100, the second SHOT descriptor and the LMD of vertex 2.

that geometric shape properties are preserved. These works use vertex-wise descriptors usually based on some truncated (reduced) basis of specific shape properties, generally leading to some loss of information such as uniqueness. For example, consider the d dimensional vector $\mathcal{U}(i,:) \in \mathbb{R}^d$ as the vertex-wise descriptors of vertex i, supposing that the matrix $\mathcal{U} \in \mathbb{R}^{n\times d}$ contains the first d eigenvectors of the uniform shape laplacian built from $\mathcal{A}$, it can be seen that $\mathcal{U}(i,:)$ is the soft cluster assignment of vertex i, and as such most vertices in the same cluster as i will have similar vertex-wise descriptors. This can be seen in figure 1 where the second eigenvector $\mathcal{U}(:,2) \in \mathbb{R}^n$ is shown for example shapes from TOSCA and TOPKIDS. One can see that $\mathcal{U}(:,2) \in \mathbb{R}^n$ is indeed a soft cluster assignment grouping some vertices in the same cluster together. This explains the success in achieving smoothness by some methods that use the d dimensional spectrum of the shape laplacian.

Theorem 3.1 *Given the shape descriptor $\mathcal{U}(i,:) \in \mathbb{R}^d$ as the vertex-wise descriptors of vertex i, supposing that the matrix $\mathcal{U} \in \mathbb{R}^{n\times d}$ contains the first d eigenvectors or left singular vectors of some unique shape pairwise descriptor $\mathcal{W}$. Using $\mathcal{W}$ for the map refinement via Functional maps helps group nearby clusters together assuming the functional map is perfectly accurate.*

Proof 3.1 *Recall that given a map $\mathcal{T}^{t-1}$, and two vertex-wise descriptors $\mathcal{U}_{\mathcal{M}}$ and $\mathcal{U}_{\mathcal{N}}$ both $\in R^{n\times d}$ the functional map $\mathcal{C} \in R^{d\times d}$ is obtained as:*

$$\mathcal{C}^t = \min_{\mathcal{C}^t} ||\mathcal{U}_{\mathcal{M}}(\mathcal{T}^{t-1},:) - \mathcal{U}_{\mathcal{N}}\mathcal{C}^t||$$

This functional map can then be used to convert functions in the basis of shape $\mathcal{M}$ into those of shape $\mathcal{N}$ and vice versa. This is then used to refine the map via solving equation 2 where $f(\mathcal{T}^t) = \mathcal{C}^t$. One can see that $\mathcal{U}_{\mathcal{N}}\mathcal{C}^t$ converts the soft clusters (left singular vectors or first eigenvectors) of shape $\mathcal{U}_{\mathcal{N}}$ into the basis space of the descriptors (soft clusters) of shape $\mathcal{U}_{\mathcal{M}}$, and so solving 2 is basically aligning the soft clusters in the same basis space. Thus solving equation 2 is matching aligned soft clusters.

In fact, methods such as ZoomOut [31, 20, 34] which iteratively refine the map $\mathcal{T}$ by using an increasing number of eigenvectors of the shape laplacian can be said to be refining the map by increasingly aligning more fine-grained clusters of the shapes. Figure 2 shows that though ZoomOut's recovered maps $\mathcal{T}$ on the isometric SCAPE[26] dataset are generally smooth (seen by the rapid increase of the geodesic curve), they are generally not accurate due to the non-uniqueness of the descriptors it uses to refine the initialized map (seen by the proportion of vertices at geodesic error 0).

Moreover, though geometric properties may be preserved even for some mild non-isometric deformations, they usually are not preserved when the transformation deviates significantly from isometry. For example, consider the shape Laplace Beltrami Operator (LBO) on which many vertex-wise and pair-wise descriptors are built:

$$\mathcal{L} = \mathcal{V}^{-1}\mathcal{S} \tag{3}$$

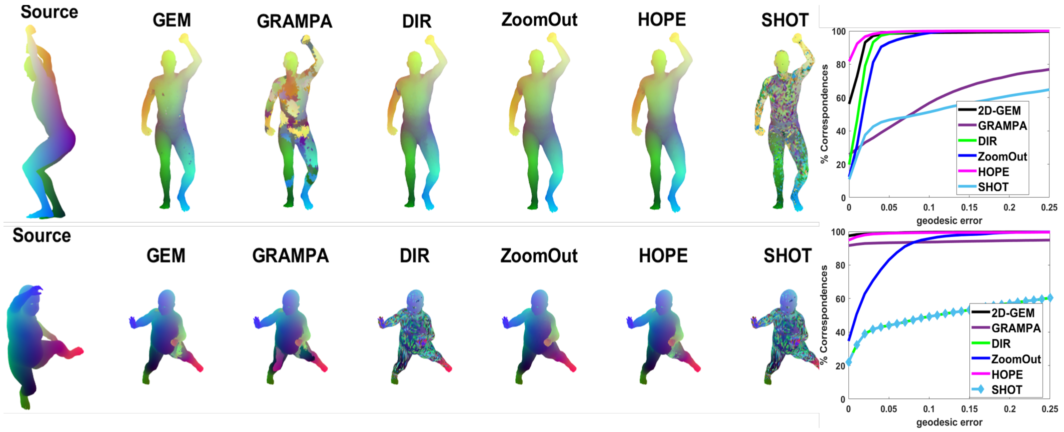


Figure 2: Showing mating of sample shapes from the isometric SCAPE dataset (first row) and the non-isometric TOPKIDS (second row), this figure shows the performance of different baselines.

where $\mathcal{V}$ is the diagonal mass matrix with $\mathcal{V}(i,i)$ entry along the diagonal being the vertex area of vertex i, and the matrix $\mathcal{S}$ being defined as:

$$\mathcal{S} = \begin{cases} \frac{1}{2}(cot\alpha_{i,j} + cot\beta_{ij}), & j \in neigh(i) \\ -\sum_{k \in neigh(i)} \mathcal{S}_{i,k} & i = k \\ 0, & otherwise \end{cases} \tag{4}$$

where $neigh(i)$ denotes the neighborhood of i. It can be observed from the way the LBO is defined that a non-isometric deformation which changes angles, areas, or even merges part of the shape (such as gluing hands to face for example) will change the LBO and by implication its spectrum, hence changing the vertex-wise and pair-wise descriptors based on the LBO. Figure 2 shows the poor performance of DIR [49] and ZoomOut [31] (which use the spectrum of the LBO as vertex-wise descriptors) on a pair of sample shapes from the non-isometric dataset TOPKIDS [26] (which contains shapes with topological noise).

Other methods aim for accuracy use more unique descriptors. However, these descriptors are generally not smooth since they generally do not capture neighborhoods and as such do not match neighborhoods. Figure 1 shows the first SHOT [45] descriptor and figure 2 show that while it generally provides an accurate map [45] (seen by the proportion of vertices at geodesic error 0) the map is not smooth (as seen by the lack of rapid increase of the geodesic curve). That is why methods aiming for smoothness [31, 37, 12, 17] only use these unique descriptors for initialization and then use the LBO or other smooth basis for refinement.

Others have used the 1-hop [14] or 2-hop [52, 22] neighborhoods of vertices as witnesses to improve an initialized map. Though these methods aim for both accuracy (via uniqueness of vertex neighborhoods), and smoothness via using neighborhoods as witnesses to improve the map (matching neighborhoods that agree best), they are nonetheless affected by the fact that 1-hop or 2-hop neighborhoods may not be unique enough and distant vertices may have similar 1-hop and or 2-hop neighborhoods especially in the presence of symmetries in $\mathcal{A}$. For example, consider GRAMPA [14] which initializes the map by using a graph matching kernel in the full graph spectrum and then uses the 1-hop (or 2-hop as in this figure) neighborhood to refine the map. Figure 2 shows GRAMPA [14] performing well on non-isometric shapes from TOPKIDS where there are few symmetries (and other graph properties) which normally cause the neighborhoods of the vertices to be non-unique. However, figure 2 shows that GRAMPA struggles on a pair of shapes from the nearly isometric dataset SCAPE [26] where shapes have symmetries (and other graph properties) that normally cause the neighborhoods of the vertices to be non-unique. To address this challenge, 2D-GEM[22] proposed to use the concept of local map distortion LMD[49, 48] and the spectrum of the uniform shape laplacian. However, though 2D-GEM seems to perform well on non-isometric and isometric shapes (figure 2), 2D-GEM does not generalize, but rather significantly changed their model (depending on user-defined parameters) in order to handle isometric cases as discussed in Section 4.4.

4 HOPE

To address both accuracy and smoothness for correlated meshes undergoing isometric and non-isometric deformations we propose HOPE (k-HOP niEghborhood matching). HOPE is based on the iterative refinement of an initialized map by: (1) using the local map distortion (LMD)[49, 48] to identify poorly matched vertices, and (2) using noise robust k-hop neighborhood-based descriptors for refinement of the maps of these poorly matched vertices.

This section is organized as: (a) the local map distortion, (b) k-hop pairwise descriptors, and (c) iterative refinement pipeline and algorithm.

4.1 Local Map Distortion (LMD)

Let $\mathcal{T} : \mathcal{M} \to \mathcal{N}$ be a map between two shapes. The LMD [49, 22] of the map $\mathcal{T}$ at the vertex x_i is given as follows:

$$\mathcal{D}_\gamma(\mathcal{T})(x_i) = \frac{\sum_{x_j \in \mathcal{B}_\gamma(x_i)} \mathcal{V}_\mathcal{M}(j) DE_\mathcal{T}(x_i, x_j)}{\sum_{x_j \in \mathcal{B}_\gamma(x_i)} \mathcal{V}_\mathcal{M}(j)}, \tag{5}$$

where $\mathcal{B}_\gamma(x_i) = \{x_j \in \mathcal{M} \mid d_\mathcal{M}(x_i, x_j) \leq \gamma\}$ is the γ-geodesic ball of x_i, $\mathcal{V}_\mathcal{M}$ is the area element of the mesh of shape $\mathcal{M}$, and $DE_\mathcal{T}(x_i, x_j) = |d_\mathcal{M}(x_i, x_j) - d_\mathcal{N}(\mathcal{T}(x_i), \mathcal{T}(x_j))|/\gamma$ represents a pairwise distance distortion of mapping nearby vertices x_i and x_j to $\mathcal{T}(x_i)$ and $\mathcal{T}(x_j)$. A smaller value of $\mathcal{D}_\gamma(\mathcal{T})(x_i)$ means a better map continuity of $\mathcal{T}$ at the vertex x_i, in other words, the local distance at the point x_i is well preserved. Based on the above definition of LMD, one can check that if $\mathcal{T}$ is an isometric map, then $\mathcal{D}_\gamma(\mathcal{T})(x_i) = 0, \forall x_i \in \mathcal{M}, \gamma > 0$. Conversely, if $\mathcal{D}_\gamma(\mathcal{T})(x_i) = 0, \forall x_i \in \mathcal{M}$ for some $\gamma > 0$, then $\mathcal{T}$ is isometric. We use the LMD to find well-matched pairs ($lmks$) i.e., $lmks = \{(x_i, \mathcal{T}(x_i)) | \mathcal{D}_\gamma(\mathcal{T})(x_i) \leq \epsilon\}$, where ϵ is a threshold, We fix ϵ to be the same values for all dataset for all our experiments. We then call the rest of the vertices on shape $\mathcal{M} \notin lmks$ as non-landmarks $Nlmks$ (i.e., poorly mapped vertices).

4.2 K-Hop Pairwise Descriptors

Given an initial map $\mathcal{T}^0$ matching a fraction β of the vertices n correctly and the fraction $1 - \beta$ incorrectly, it has been shown that one can refine the map by (a) using the 1-hop [29] neighborhood contained in the graph adjacencies of the triangulations $\mathcal{A}_\mathcal{M}$ and $\mathcal{A}_\mathcal{N}$, or (b) using k-hop [52] neighborhoods contained in binary matrices $\mathcal{A}_{\mathcal{M},k}$ and $\mathcal{A}_{\mathcal{N},k} \in \mathbb{R}^{n \times n}$ indicating whether vertices are connected at a path of length k on the shape mesh adjacencies (note that $k = 1$ is the adjacency). The refinement is done by solving:

$$\mathcal{T}^t = \arg\max_{\mathcal{T}} \mathbf{Tr}(\mathcal{A}_{\mathcal{M},k}(\mathcal{T}, \mathcal{T}^{t-1})\mathcal{A}_{\mathcal{N},k}), \tag{6}$$

the main assumption being that 2 the meshes $\mathcal{M}$ and $\mathcal{N}$ are correlated i.e., their adjacencies are assumed to come from the same parent graph, with their correlation ratio being s. Specifically, given a $\mathcal{G}(n, p)$ Erdos Regny parent graph and the correlation s, $1 - s$ is the probability of independently randomly deleting edges from this graph to either obtain the adjacency of $\mathcal{M}$ or that of $\mathcal{N}$. For $k = 1$ and $k = 2$, rigorous analysis and tight bounds were given for recovering the ground truth map $\hat{\mathcal{T}}$ (for given values of s, p and β) by solving equation 6 (see [51, 30, 52, 29] for exact bounds).

Theorem 4.1 *Given the k-hop based descriptors $\mathcal{A}_{\mathcal{M},k}$ and $\mathcal{A}_{\mathcal{N},k}$, solving equation 6 is matching vertices whose neighborhoods agree best under $\mathcal{T}^{t-1}$.*

Proof 4.1 *Notice that:*

- *columns of $\mathcal{A}_{\mathcal{M},k}$ and$\mathcal{A}_{\mathcal{N},k}$ are descriptors for the vertices (rows), and these columns are the k-hop neighborhoods of vertices (rows)*
- *$\mathcal{T}^{t-1}$ is used in equation 6 for first aligning these descriptors i.e., $\mathcal{A}_{\mathcal{M},k}(:, \mathcal{T}^{t-1})$ rearranges the columns of $\mathcal{A}_{\mathcal{M},k}$ thus realigning the descriptors of each vertex (the rows),*
- *each entry $\mathcal{K}(i, j)$ in the product $\mathcal{K} = \mathcal{A}_{\mathcal{M},k}(:, \mathcal{T}^{t-1})\mathcal{A}_{\mathcal{N},k}$ will be the number of vertices that are common in the k-hop neighborhoods of vertices i and j, after the alignment of the k-hop neighborhoods of vertices in $\mathcal{M}$ by $\mathcal{A}_{\mathcal{M},k}(:, \mathcal{T}^{t-1})$*

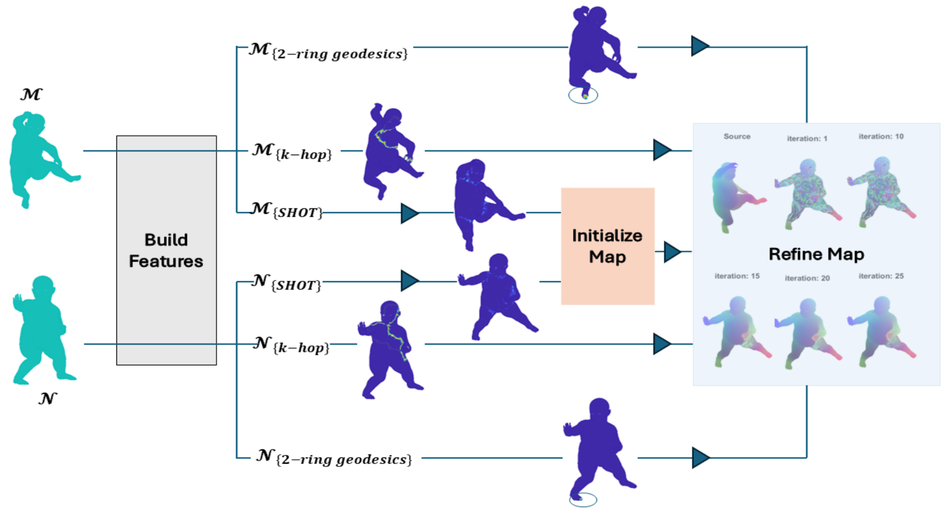


Figure 3: HOPE (a pipeline) for shape matching.

- *finding $\mathcal{T}$ via equation 6 matches the vertices i, and j whose k-hop neighborhood have most vertices in common based on the alignment $\mathcal{T}^{t-1}$.*

However, in the presence of symmetries the next 1-hop and 2-hop neighborhoods may not be unique enough as even distant nodes may have the same 1 and or 2-hop connectivity. To address this non-uniqueness, we follow 2D-GEM, and use the LMD to first detect $Nlmks$ vertices. Unlike 2D-GEM which used the laplacian spectrum to refine the map $\mathcal{T}(Nlmks)$ for $Nlmks$ vertices, we instead propose to iteratively use different k-hop neighborhoods for refining the map $\mathcal{T}(Nlmks)$ for $Nlmks$ vertices, i.e., iteratively using nodes at different lengths k to refine $\mathcal{T}(Nmlks)$. We propose this strategy because: (b) first, 2D-GEM's strategy of using the spectrum will suffer from non-uniqueness according to theorm 3.1, and will also not adapt to non-isometric shapes as discussed in Sections 3 and 4.4. (b) Second, according to theorem 4.1 our strategy will ensure that $\mathcal{T}$ is consistent across different neighborhoods. We do so for $1 \leq k \leq k_{max}$ where k_{max} is large. This strategy is inspired by the fact that it was shown that large neighborhood statistics are essential in graph matching [51, 30, 52, 29, 10, 32], and other graph tasks [18, 5, 47, 21, 22, 53].

4.3 Pipeline for HOPE

Here we show the general pipeline for HOPE which consists of the two aforementioned steps, namely:

- (a) Map initialization: we use any robust map initializations e.g., the map initialized from SHOT[45] in our experiments
- (b) map refinement for t iteration steps consisting of:
  - Using the LMD [22, 49, 48] to detect the poorly matched $Nlmks$ vertices
  - refine the map for the $Nlmks$ vertices via enforcing different noise robust k-hop neighborhoods consistency (for $1 \leq k \leq k_{max}$) by solving:

$$\mathcal{T}^t(Nlmks) = \arg\max_{\mathcal{T}} \mathbf{Tr}(\mathcal{A}_{\mathcal{M},k}(\mathcal{T}(Nlmks), \mathcal{T}^{t-1})\mathcal{A}_{\mathcal{N},k}), \tag{7}$$

- (c) returning the final map $\mathcal{T}^t$

4.4 HOPE vs 2D-GEM

In this section, we briefly introduce 2D-GEM [22] and highlight some key differences between 2D-GEM and HOPE. Like us, 2D-GEM initializes its map via SHOT[45] and then at each iteration t 2D-GEM refined the map by:

- (1) updating the map $\mathcal{T}$ by solving one iteration of equation 6 using the 2-hop neighborhoods, meaning $k = 2$ in equation 6,
- (2) finding the $lmks$ well-matched pairs, and poorly matched pairs $Nlmks$ pairs using the LMD.
- (3) using the $lmks$ vertices to update the map of the $Nlmks$ vertices in the spectrum by using the GMWM [52] on a cost matrix built from the spectrum of the laplacian of the two shapes (see their paper for more details). They showed similarity between this approach and the functional map [34] approach.

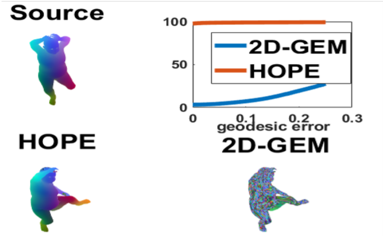


Figure 4: Comparison on TOPKIDS showing HOPE using ϵ as in Section 5.5 for its LMD threshold (same as in the isometric cases), against 2D-GEM with ϵ set as in the isometric case (i.e., using the isometric parameters in [22] rather than $\epsilon = 100$ as in the non-isometric case).

The main two differences between 2D-GEM and HOPE are: (a) HOPE completely removes their step (1), and (b) they used the laplacian (LBO) spectrum via their proposed 2D-graph convolution to refine the map for the $Nlmks$ in step (3). However, as mentioned in section 3 and shown in figure 1, using LBO descriptors loses uniqueness since vertices in similar clusters will be grouped together, and in addition, these descriptors are not robust to non-isometric deformations as discussed in Section 3.

A consequence of this is that, 2D-GEM[22] needed to deactivate step (2) and (3) of their algorithms (the $Nlmks$ refinement step) by setting $\epsilon = 100$ as their LMD threshold when dealing with non-isometric shapes. Hence though effective, their algorithm can be seen as two disjoint algorithms that are used for isometric or non-isometric shapes based on the setting of ϵ. But in real life, in some cases, it is not obvious whether the deformation used is isometric or not. On the other hand, we propose refining the $Nlmks$ map based on the preservation of different k-hop connectivity which is a constraint that holds for isometric and non-isometric shapes with correlated triangulations [51, 30, 52].

Figure 4 shows a comparison between HOPE and 2D-GEM on a non-isometric pair of shapes from TOPKIDS using the 2D-GEM ϵ parameters proposed for isometric shapes. It can be seen that when we use the isometric parameters of 2D-GEM on non-isometric shapes, 2D-GEM fails to perform well (the same holds when using their non-isometric parameter $\epsilon = 100$ on isometric shapes.

4.5 Time complexity analysis

See the algorithm for HOPE in Appendix A. Given that for step (1) HOPE follows DIR[49] and [22] in using LMD, checking the LMD takes $O(n)$. The GMWM used to solve step (2) of HOPE takes $O(|Nlmks|^2 logn)$. Hence the total time complexity of HOPE for all t iterations is $O(t(|Nlmks|^2 logn + n))$.

5 Experiments

We report experimental results that validate the effectiveness, efficiency, and generalization ability of HOPE in the matching of nearly-isometric and non-isometric 3D shapes.

5.1 Experimental Set-up

All experiments are conducted in Matlab 2023 on a Windows 11 system with 32GB RAM and Intel(R) i5 13500 CPU @ 2.50-4.8GHz.

5.2 Datasets

We evaluate the performance of HOPE on two nearly isometric benchmark datasets TOSCA [7], and SCAPE [3], as well as on the non-isometric dataset SHREC'16 (TOPKIDS) [26], TOPKIDS contains 25 shapes of the same class with up to 12K vertices, undergoing near-isometric deformations in addition to large topological noise (such as merging hands to thighs) which results in $n_{\mathcal{M}} \neq n_{\mathcal{N}}$. TOSCA consists of 80 shapes in 8 different categories (human and animal shapes) with vertex numbers ranging from 4k to 50k. SCAPE has 71 shapes (12,500 vertices for each) of the same person with different poses.

Furthermore, to see the generalization abilities of HOPE, we equally used datasets SCAPE_r[9], FAUST_r[9], and TOSCA_r[9], which are remeshed shape datasets. The SCAPE_r consists of the same 71 shapes from the SCAPE dataset, but remeshed, while the FAUST_r (TOSCA_r) likewise contains the same FAUST (TOSCA) datasets but remeshed. We used the same 71 test pairs SCAPE_r as for SCAPE, and for FAUST_r and TOSCA_r we followed [9]

5.3 Evaluation Metrics

We use the geodesic error as our error metric [14, 22]. Given that the map of an algorithm maps $x_i \in \mathcal{M}$ to $x_j \in \mathcal{N}$, and the true map maps x_i to x_j^*, the geodesic error is defined as $e(x_i) = \frac{d_{\mathcal{N}}(x_j, x_j^*)}{diam(\mathcal{N})}$, where $d_{\mathcal{N}}$ denotes the geodesic distance on $\mathcal{N}$, and $diam(\mathcal{N})$ is the geodesic diameter of $\mathcal{N}$.

5.4 Baselines

Following 2D-GEM [22], we compare HOPE with the following methods: EM [42], GE [25], RF [40], PFM [39], FSPM [28], Kernel-Matching [26], and GRAMPA [14], [22], SGMDS[1], FM[34], BIM[24], Mobius[27], Best-Conformal [24], Kernel-Matching [26], DIR-500 [49] which uses 500 eigenvectors, DIR-1000 [49] which uses 1000 eigenvectors.

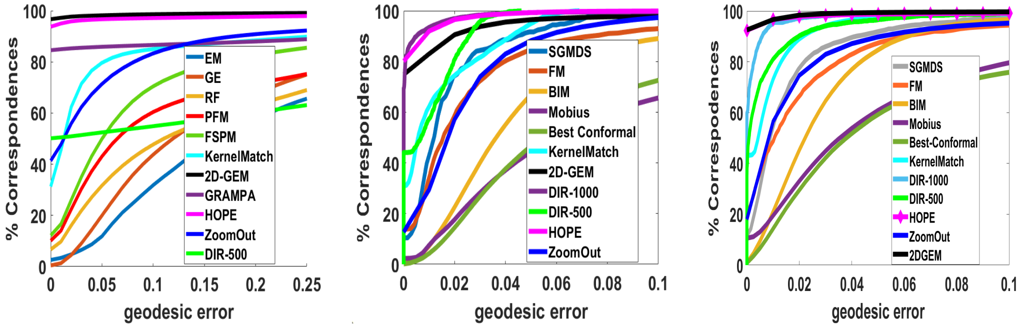
5.5 Parameter Settings

On ZoomOut, we start with a functional map using the first 20 eigenvectors of the LBO, then we iteratively add an eigenvector until we reach the 120th eigenvector after which we stop.

On all other Baselines, we follow the settings from [22].

On HOPE, on all datasets we set the LMD threshold ϵ, staring from $\epsilon = 100$ and 10 equally spaced values to $\epsilon = 0.2$ i.e., we use $\epsilon = linespace(100, 0.2, 10)$ and we set $t = 60$. When the last value of e is reached, it is maintained for the rest of the iterations. We equaly set $k_{max} = 8$ for all datasets. For the LMD, we used the second ring neighborhood following [49, 22].

5.6 Performance Analysis

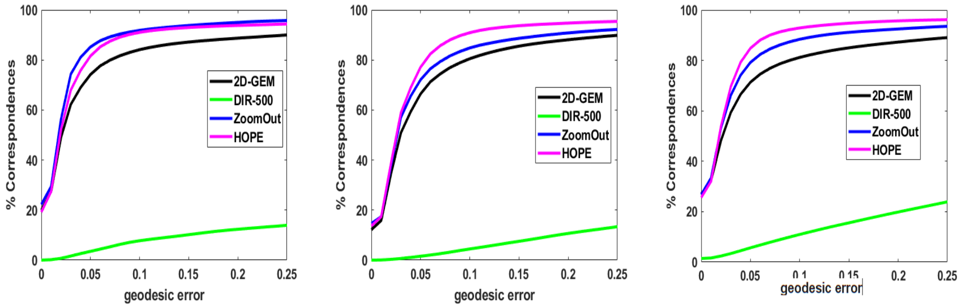


(a) TOPKIDS (b) SCAPE (c) TOSCA

Figure 5: Performance comparison on TOPKIDS 5(a), SCAPE 5(b), and TOSCA 5(c).

Comparison on non-isometric shapes. On the dataset with topological noise TOPKIDS (Figure 5(a)), the top 3 methods are the noise-robust methods 2D-GEM with 96.7% vertices correctly matched, followed by HOPE with 94.9%, and GRAMPA with accuracy of 84.5%. This shows that 2D-GEM, HOPE, and GRAMPA are indeed adapted for non-isometric shapes. Moreover, as discussed in

[22], GRAMPA is very time-consuming as it needs the full basis of the graph adjacencies of the shapes, while HOPE and 2D-GEM do not. For example on a shape with 12500 vertices from SCAPE, GRAMPA takes roughly 118 seconds on our set up while 2D-GEM and HOPE take 56 and 52 seconds respectively. Better still HOPE is a general model as it can be seen that with the same parameter of ϵ (see Section 5.5) for its LMD, it performs well on all datasets.



(a) FAUST_r (b) SCAPE_r (c) TOSCA_r

Figure 6: Performance comparison on FAUST_r 6(a), SCAPE_r 6(b), and TOSCA_r 6(c).

Comparison on isometric shapes. On isometric shapes, methods that enforce stronger geometric constraints on their refinement pipelines such as 2D-GEM (with appropriate parameters), DIR, and HOPE. HOPE outperform all other baselines. 2D-GEM achieves 92.54% accuracy at geodesic error 0 on the TOSCA dataset, as well as 74.8% accuracy on the SCAPE dataset, while HOPE achieves 80.11% on SCAPE and 92.54% on TOSCA using no matrix decomposition. Third is DIR-1000[49] with 1000 eigenvectors which achieves around 69.5% accuracy at geodesic error 0 on SCAPE, and 59.8% accuracy at geodesic error 0 on TOSCA. Moreover, unlike DIR (and 2D-GEM,), HOPE generalizes to this isometric setting using the same parameters (see Section 5.5).

Comparison on remeshed shapes. On the remeshed datasets FAUST_r, SCAPE_r and TOSCA_r in figure 6, methods that enforce stronger geometric constraints such as DIR performed poorly, while those that are more robust to noise such as 2D-GEM (with appropriate parameters), ZoomOut and HOPE perform relatively well. Though 2D-GEM does not outperform ZoomOut in this setting (probably due to differences in the mesh connectivity), HOPE still outperforms ZoomOut though by a smaller margin than on datasets with similar mesh connectivity amongst pairs.

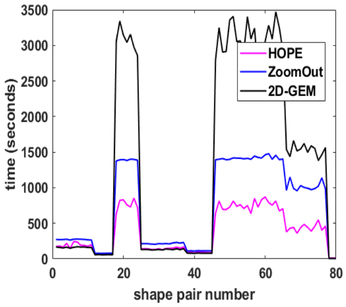


Figure 7: Time Comparison on TOSCA showing HOPE, 2D-GEM, and ZoomOut.

Time Comparison. Here we compare the time usage per shape for both 2D-GEM, HOPE, and ZoomOut on the TOSCA dataset. It can be seen from figure 7 that HOPE is relatively faster than 2D-GEM, and even ZoomoOut (for the given parameters of ZoomOut that we used for our experiments).

6 Limitations

The main limitation of this work is the reliance on correlated triangulations between the pair of shapes to be matched as outlined in Section 4.2. One can see this drawback on figures 6 which show that though HOPE still performs well on remeshed shapes (shapes where the triangulations are not very correlated), and even outperforms other baselines, it nonetheless does not perform as well as on figures 5(b), 5(a) and 5(c) where the triangulations of the pairs matched are strongly correlated. This is a major drawback because triangulations are often hard and expensive to get, especially consistent triangulations between shapes.

7 Conclusion

We introduced an effective and easy-to-implement map refinement strategy consisting of; (a) detecting poorly matched vertices (nodes) using the concept of local map distortion (LMD), and (b) improving the map of these poorly matched vertices via noise robust k-hop pairwise descriptors. We then conducted a series of experiments to show that our framework is effective and generalizable to different shape datasets. We also discussed the main limitation of our work (the fact that it is reliant on the triangulation consistency between the shapes matched).

8 Broader Impact

This work proposes a map refinement strategy for shape matching that is based on matching different k-hop neighborhoods of vertices. It then validates the effectiveness of this strategy on several shape matching datasets. This can spark new research on the importance of large neighborhood statistics for shape matching and other related tasks. As we focus solely proposing a framework for map refinement for shape matching, we do not see clear negative impact of this work.

9 Acknowledgements

This work is partially supported by Hong Kong Research Grants Council under the Areas of Excellence Scheme grant AoE/P-601/23-N.

References

[1] Yonathan Aflalo, Anastasia Dubrovina, and Ron Kimmel. Spectral generalized multi-dimensional scaling. *International Journal of Computer Vision*, 118:380–392, 2016.

[2] B. Amberg, S. Romdhani, and T. Vetter. Optimal step nonrigid icp algorithms for surface registration. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007.

[3] Dragomir Anguelov, Praveen Srinivasan, Hoi-Cheung Pang, Daphne Koller, Sebastian Thrun, and James Davis. The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In *Advances in neural information processing systems*, pages 33–40, 2005.

[4] Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 1626–1633, 2011. doi: 10.1109/ICCVW.2011.6130444.

[5] Federico Barbero, Ameya Velingker, Amin Saberi, Michael M. Bronstein, and Francesco Di Giovanni. Locality-aware graph rewiring in GNNs. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=4Ua4hKiAJX`.

[6] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape context: A new descriptor for shape matching and object recognition. In T. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13. MIT Press, 2000. URL `https://proceedings.neurips.cc/paper_files/paper/2000/file/c44799b04a1c72e3c8593a53e8000c78-Paper.pdf`.

[7] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. *Numerical geometry of non-rigid shapes*. Springer Science & Business Media, 2008.

[8] M. M. Bronstein and I. Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1704–1711, 2010.

[9] Dongliang Cao, Paul Roetzer, and Florian Bernard. Unsupervised learning of robust spectral shape matching. *ACM Transactions on Graphics (TOG)*, 2023. doi: 10.1145/3592107. URL `https://doi.org/10.1145/3592107`.

[10] Tomek Czajka and Gopal Pandurangan. Improved random graph isomorphism. *J. of Discrete Algorithms*, 6(1):85–92, March 2008. ISSN 1570-8667. doi: 10.1016/j.jda.2007.01.002.

[11] Nadav Dym, Haggai Maron, and Yaron Lipman. Ds++: A flexible, scalable and provably tight relaxation for matching problems. *ACM Trans. Graph.*, 36(6), November 2017. ISSN 0730-0301. doi: 10.1145/3130800.3130826.

[12] M. Eisenberger, Z. Lahner, and D. Cremers. Smooth shells: Multi-scale shape registration with functional maps. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12262–12271, Los Alamitos, CA, USA, jun 2020. IEEE Computer Society. doi: 10.1109/CVPR42600.2020.01228. URL `https://doi.ieeecomputersociety.org/10.1109/CVPR42600.2020.01228`.

[13] Marvin Eisenberger, Aysim Toker, Laura Leal-Taixe, and Daniel Cremers. G-msm: Unsupervised multi-shape matching with graph-based affinity priors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, March 2023.

[14] Zhou Fan, Cheng Mao, Yihong Wu, and Jiaming Xu. Spectral graph matching and regularized quadratic relaxations: Algorithm and theory. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2985–2995. PMLR, 13–18 Jul 2020.

[15] Soheil Feizi, Gerald Quon, Mariana Recamonde-Mendoza, Muriel Medard, Manolis Kellis, and Ali Jadbabaie. Spectral alignment of graphs. *IEEE Transactions on Network Science and Engineering*, 7(3):1182–1197, jul 2020. ISSN 2327-4697. doi: 10.1109/TNSE.2019.2913233.

[16] Gerd Finke, Rainer E. Burkard, and Franz Rendl. Quadratic assignment problems. In Silvano Martello, Gilbert Laporte, Michel Minoux, and Celso Ribeiro, editors, *Surveys in Combinatorial Optimization*, volume 132 of *North-Holland Mathematics Studies*, pages 61–82. North-Holland, 1987. doi: https://doi.org/10.1016/S0304-0208(08)73232-8.

[17] Florine Hartwig, Josua Sassen, Omri Azencot, Martin Rumpf, and Mirela Ben-Chen. An elastic basis for spectral shape correspondence. In *ACM SIGGRAPH 2023 Conference Proceedings*, SIGGRAPH '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701597. doi: 10.1145/3588432.3591518. URL `https://doi.org/10.1145/3588432.3591518`.

[18] Max Horn, Edward De Brouwer, Michael Moor, Yves Moreau, Bastian Rieck, and Karsten Borgwardt. Topological graph neural networks. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=oxxUMeFwEHd`.

[19] Ruqi Huang, Jing Ren, Peter Wonka, and Maks Ovsjanikov. Consistent zoomout: Efficient spectral map synchronization. In *Proc. SGP*, volume 39, 2020.

[20] Ruqi Huang, Jing Ren, Peter Wonka, and Maks Ovsjanikov. Consistent zoomout: Efficient spectral map synchronization. *Computer Graphics Forum*, 39(5):265–278, 2020. doi: https://doi.org/10.1111/cgf.14084. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14084`.

[21] Barakeel Fanseu Kamhoua, Lin Zhang, Kaili Ma, James Cheng, Bo Li, and Bo Han. Hypergraph convolution based attributed hypergraph clustering. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021.

[22] Barakeel Fanseu Kamhoua, Lin Zhang, Yongqiang Chen, Han Yang, MA KAILI, Bo Han, Bo Li, and James Cheng. Exact shape correspondence via 2d graph convolution. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=f39vsgpEaY5`.

[23] Ehsan Kazemi, Seyed Hamed Hassani, M. Grossglauser, and H. Modarres. Proper: global protein interaction network alignment through percolation matching. *BMC Bioinformatics*, 17, 2016.

[24] Vladimir G. Kim, Yaron Lipman, and Thomas Funkhouser. Blended intrinsic maps. *ACM Trans. Graph.*, 30(4), July 2011. ISSN 0730-0301. doi: 10.1145/2010324.1964974.

[25] Z. Lähner, E. Rodolà, M. M. Bronstein, D. Cremers, O. Burghard, L. Cosmo, A. Dieckmann, R. Klein, and Y. Sahillioğlu. Matching of deformable shapes with topological noise. In *Proceedings of the Eurographics 2016 Workshop on 3D Object Retrieval*, 3DOR '16, page 55–60, Goslar, DEU, 2016. Eurographics Association. ISBN 9783038680048.

[26] Zorah Lähner, Matthias Vestner, Amit Boyarski, Or Litany, Ron Slossberg, Tal Remez, Emanuele Rodolà, Alexander M. Bronstein, Michael M. Bronstein, Ron Kimmel, and Daniel Cremers. Efficient deformable shape correspondence via kernel matching. *2017 International Conference on 3D Vision (3DV)*, pages 517–526, 2017.

[27] Yaron Lipman and Thomas Funkhouser. Möbius voting for surface correspondence. *ACM Trans. Graph.*, 28(3), July 2009. ISSN 0730-0301. doi: 10.1145/1531326.1531378.

[28] O. Litany, E. Rodolà, A. M. Bronstein, and M. M. Bronstein. Fully spectral partial shape matching. *Computer Graphics Forum*, 36(2):247–258, 2017. doi: https://doi.org/10.1111/cgf.13123.

[29] Joseph Lubars and R. Srikant. Correcting the output of approximate graph matching algorithms. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, pages 1745–1753, 2018. doi: 10.1109/INFOCOM.2018.8486238.

[30] Cheng Mao, Mark Rudelson, and Konstantin Tikhomirov. Exact matching of random graphs with constant correlation. *Probability Theory and Related Fields*, 186(1-2):327–389, 2023.

[31] Simone Melzi, Jing Ren, Emanuele Rodolà, Abhishek Sharma, Peter Wonka, and Maks Ovsjanikov. Zoomout: Spectral upsampling for efficient shape correspondence. *ACM Transactions on Graphics (TOG)*, 38(6):155, 2019.

[32] Elchanan Mossel and Nathan Ross. Shotgun assembly of labeled graphs. *IEEE Transactions on Network Science and Engineering*, 6(2):145–157, 2019. doi: 10.1109/TNSE.2017.2776913.

[33] Maks Ovsjanikov, Jian Sun, and Leo Guibas. Global intrinsic symmetries of shapes. *Comp. Graph. Forum*, 27(5):1341–1348, 2008.

[34] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: A flexible representation of maps between shapes. *ACM Trans. Graph.*, 31(4), July 2012. ISSN 0730-0301. doi: 10.1145/2185520.2185526.

[35] Gautam Pai, Jing Ren, Simone Melzi, Peter Wonka, and Maks Ovsjanikov. Fast sinkhorn filters: Using matrix scaling for non-rigid shape correspondence with functional maps. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 384–393, 2021. doi: 10.1109/CVPR46437.2021.00045.

[36] Gautam Pai, Jing Ren, Simone Melzi, Peter Wonka, and Maks Ovsjanikov. Fast sinkhorn filters: Using matrix scaling for non-rigid shape correspondence with functional maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 384–393, June 2021.

[37] Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. Continuous and orientation-preserving correspondences via functional maps. *ACM Trans. Graph.*, 37(6), dec 2018. ISSN 0730-0301. doi: 10.1145/3272127.3275040.

[38] Jing Ren, Mikhail Panine, Peter Wonka, and Maks Ovsjanikov. Structured regularization of functional map computations. *Comput. Graph. Forum*, 38(5):39–53, 2019. doi: 10.1111/cgf.13788. URL `https://doi.org/10.1111/cgf.13788`.

[39] E. Rodolà, L. Cosmo, M. M. Bronstein, A. Torsello, and D. Cremers. Partial functional correspondence. *Comput. Graph. Forum*, 36(1):222–236, January 2017. ISSN 0167-7055. doi: 10.1111/cgf.12797.

[40] Emanuele Rodolà, Samuel Bulò, Thomas Windheuser, Matthias Vestner, and Daniel Cremers. Dense non-rigid shape correspondence using random forests. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 4177–4184, 2014. doi: 10.1109/CVPR.2014.532.

[41] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE International Conference on Robotics and Automation*, pages 3212–3217, 2009. doi: 10.1109/ROBOT.2009.5152473.

[42] Yusuf Sahillioğlu and Yücel Yemez. Minimum-distortion isometric shape correspondence using em algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11): 2203–2215, 2012. doi: 10.1109/TPAMI.2012.26.

[43] Aliaksei Sandryhaila and José M. F. Moura. Discrete signal processing on graphs. *IEEE Transactions on Signal Processing*, 61(7):1644–1656, 2013. doi: 10.1109/TSP.2013.2238935.

[44] Gil Shamai and Ron Kimmel. Geodesic distance descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[45] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *European conference on computer vision*, pages 356–369. Springer, 2010.

[46] S. Umeyama. An eigendecomposition approach to weighted graph matching problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(5):695–703, 1988. doi: 10.1109/34.6778.

[47] Yuhao Wu, Jiangchao Yao, Bo Han, Lina Yao, and Tongliang Liu. Unraveling the impact of heterophilic structures on graph positive-unlabeled learning. In *Forty-first International Conference on Machine Learning*, 2024. URL `https://openreview.net/forum?id=NCT3w7VKjo`.

[48] Rui Xiang, Rongjie Lai, and Hongkai Zhao. Efficient and robust shape correspondence via sparsity-enforced quadratic assignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9513–9522, 2020.

[49] Rui Xiang, Rongjie Lai, and Hongkai Zhao. A dual iterative refinement method for non-rigid shape matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15930–15939, June 2021.

[50] Lei Xu and Irwin King. A pca approach for fast retrieval of structural patterns in attributed graphs. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, 31:812–7, 02 2001. doi: 10.1109/3477.956043.

[51] Joonhyuk Yang, Dongpil Shin, and Hye Won Chung. Efficient algorithms for exact graph matching on correlated stochastic block models with constant correlation. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[52] LIREN YU, Jiaming Xu, and Xiaojun Lin. Graph matching with partially-correct seeds. *J. Mach. Learn. Res.*, 22:280:1–280:54, 2021.

[53] Xiaotong Zhang, Han Liu, Qimai Li, and Xiao-Ming Wu. Attributed graph clustering via adaptive graph convolution. In *IJCAI*, 2019.

EHOPE: Shape Matching Via Aligning Different K-hop Neighbourhoods
Appendices:

A HOPE Algorithm

In this section, we present the algorithm for HOPE. This is given in algorthm 1

Algorithm 1 : HOPE

```
0. input An initial map $\mathcal{T}^0$, distance matrices $d_{\mathcal{M}}$ and $d_{\mathcal{N}}$, LMD threshold $\epsilon$, maximum iteration $t$, and the maximum hop $k_{max}$ and $k = 1$
while $0 \leq i \leq t$ do
    1. build $\mathcal{A}_{\mathcal{M},k}$ and $\mathcal{A}_{\mathcal{M},k}$
    2. set $k = k + 1$
    if $k > k_{max}$ then
        3. $k = 1$,
    end if
    4. Given $\mathcal{T}^i$, use the LMD to locate well-matched points $lmks$ and poorly-matched points $Nlmks$,
    5. Use these $Nlmks$ pairs to update the $\mathcal{T}^i(Nlmks)$ by using the GMWM to solve equation (7),
end while
return $\mathcal{T}^t$.
```

B Ablation and Parameter Sensitivity Studies

In this section, we conduct parameter and ablation studies. We use the following settings:

- HOPE: the hope algortithm as in algorithm 1, with $t = 60$ and $\epsilon = linespace(100, 0.2, 10)$ as in the experiments in Section 5 in the main paper,
- HOPE-M: where we reduce the number of iterations to $t = 20$ in algorithm 1,
- HOPE-th: where we set $\epsilon = linespace(1, 0.2, 10)$ in algorithm 1,
- HOPE-fixhop: where we simply solve equation 6 with $k = 1$ and $k = 2$ alternatively per iteration as in algorithm 1,
- HOPE-LMDvaryhop: where use $k_{max} = 2$ in algorithm 1,
- HOPE-varyhop: where we simply solve equation 6 with $k = [1, 2, \cdots, 8]$ alternatively per iteration, as in algorithm 1.

It can be observed from figure 8 that the best model overall is HOPE-th which is best on isometric (figure 8(b)) shapes and non-isometric shapes (figure 8(a)) and performaing comparatively to other variants on remeshed shapes (figure 8(c)). This is followed by HOPE. This indicates that the ϵ we used for our main experiments in Section 5 is sub-optimal since we randomly selected the range to be $linspace(100, 0.2, 10)$ without any parameter tuning to demonstrate the effectiveness and generalizability of HOPE. We notice that the variants without the LMD (HOPE-varyhop and HOPE-fixhop) all struggled on the isometric shape (figure 8(b)) validating our observation that the k-hop neighborhood of nodes may not be very unique especially when there are symmetries, and as such using a stronger constraint like the LMD is beneficial in such settings (Section 3 and 4)

C Different Initializations

In this section, we conduct studies on the effects of different initializations on HOPE. We use the following settings:

- HOPE-SHOT: where we use SHOT[45] descriptors to initialize t^0 in algorithm 1,
- HOPE-HKS: where we use Heat Kernel Signatures (HKS)[8] descriptors to initialize t^0 in algorithm 1,
- HOPE-WKS: where we use Wave Kernel Signatures (WKS)[4] descriptors to initialize t^0 in algorithm 1.

Figure 9 shows that indeed SHOT [45] which is commonly used in practice is a robust and good descriptor to use as initialization. On the other hand while both the HKS [8] and WKS [4] provided an initialization that could be enhanced by HOPE relatively well on the isometric shape (figure 9(b)), only WKS and SHOT were suitable for the remeshed shape (figure 9(c)), while only SHOT initializations where relatively okay in terms of accuracy on the non-sometric shape (figure 9(a)).

D Partial Shape Matching

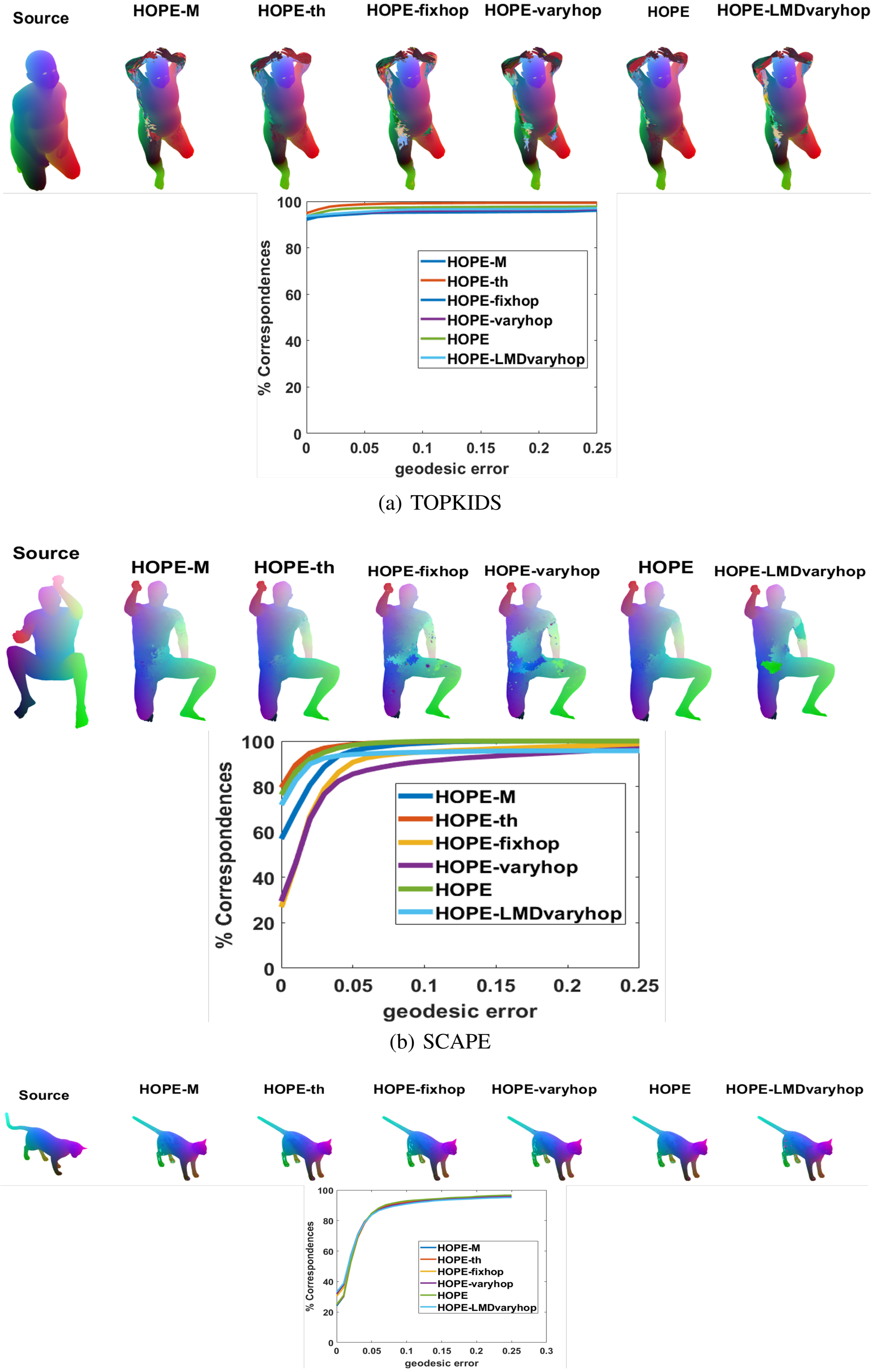
In this section, we show the performance of HOPE on partial shape matching where we match the full shapes to the partial shapes. Here we used the SHREC16 HOLES and SHREC16 CUTS following Cao et al. [9]. Figure 10 shows that although HOPE was not designed specifically for the partial shape setting, it nonetheless performs relatively well on average as seen by the average geodesic curve.

E Non-Isometric Shape Matching

In this section, we show the performance of HOPE on another non-isometric shape dataset SMAL_r Cao et al. [9], where we perform intra-class matching of 298 pairs of shapes. Figure 11 shows that HOPE again outperforms other baselines even when the mesh triangulations are not strongly correlated (due to noise).

F Visual Comparisons

In this section, we show the visual comparisons between HOPE and other baselines, on SCAPE_r (figure 12), SHREC16 (figure 13), and TOPKIDS (figure 14).

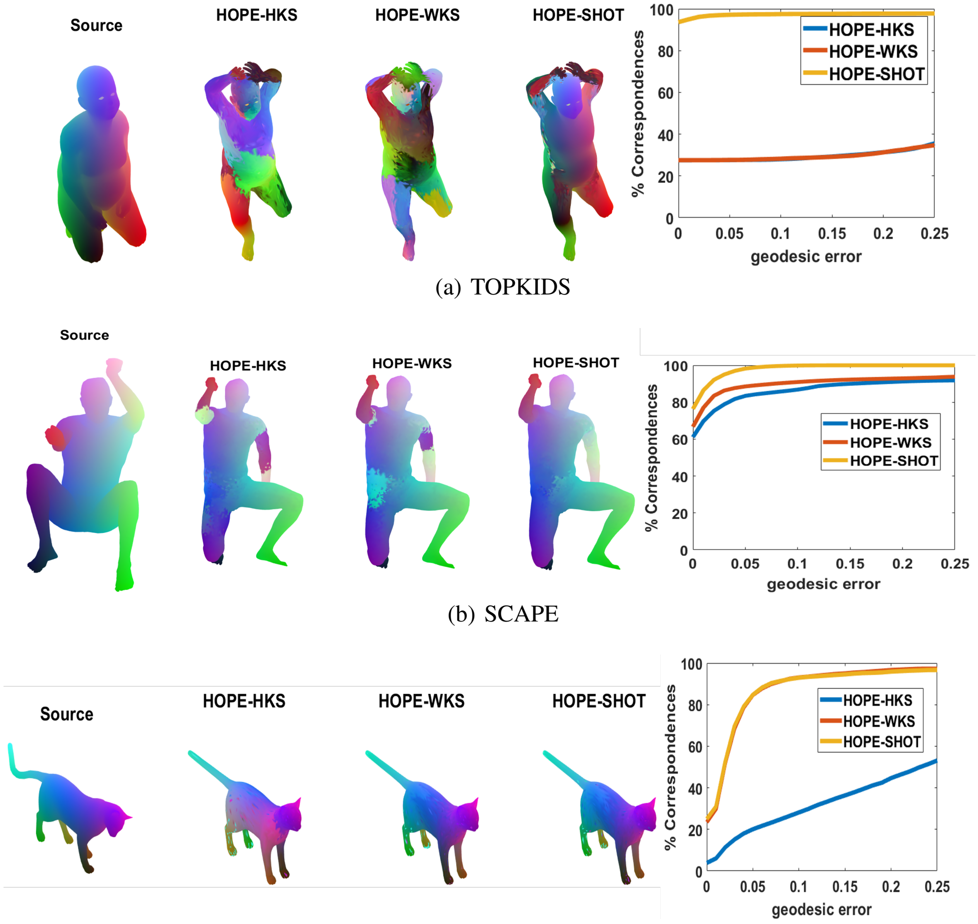


(a) TOPKIDS

(b) SCAPE

(c) TOSCA_r

Figure 8: Ablation and sensitivity studies on TOPKIDS 8(a), SCAPE 8(b), and TOSCA_r 8(c).

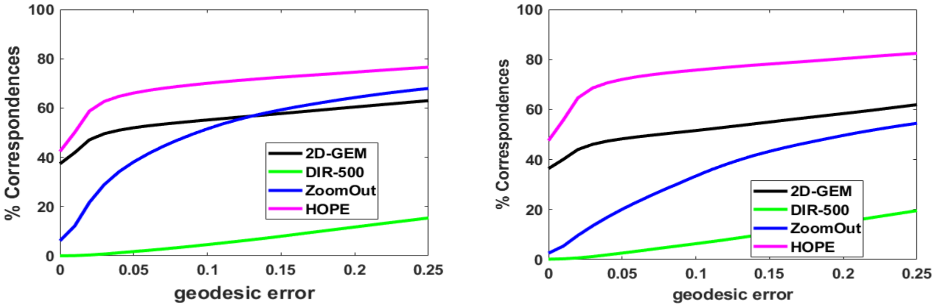


(a) TOPKIDS

(b) SCAPE

(c) TOSCA_r

Figure 9: Different initialzations on TOPKIDS 9(a), SCAPE 9(b), and TOSCA_r 9(c).



(a) SHREC16 HOLES

(b) SHREC16 CUTS

Figure 10: Performance comparisons on SHREC16 HOLES 10(a), SHREC16 CUTS 9(b), and TOSCA_r 10(b).

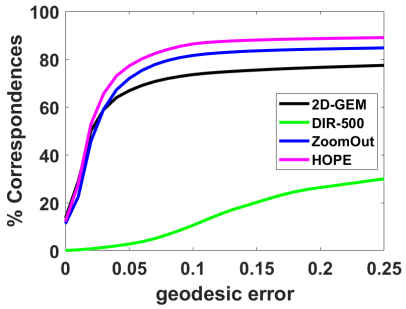


Figure 11: Comparison on intra-class matching on SMAL_r using 298 pairs of shape. Figure showing HOPE, 2D-GEM, ZoomOut, and DIR.

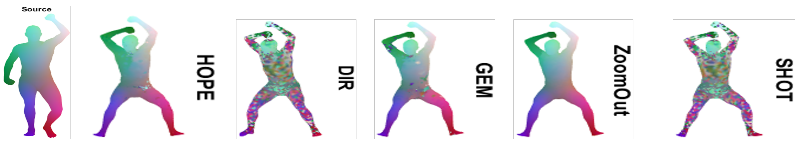


Figure 12: Sample shape from SCAPE_r. Figure showing HOPE, 2D-GEM, ZoomOut, and DIR.

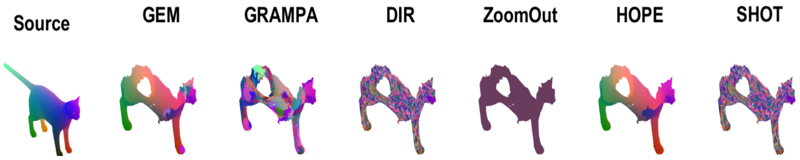


Figure 13: Sample shape from SHREC16-cuts. Figure showing HOPE, 2D-GEM, ZoomOut, and DIR.

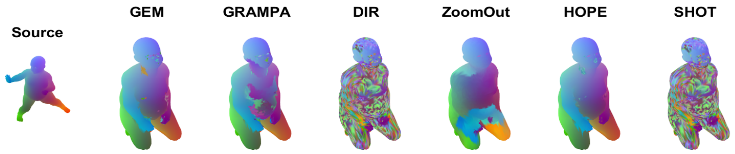


Figure 14: Sample shape from TOPKIDS. Figure showing HOPE, 2D-GEM, ZoomOut, and DIR.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and precede the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper set out to propose the SLNC problem, show that existing baseline GNNs do not perform well in this setting, and propose ELI to improve GNNs performance in SLNC. These were all shown and demonstrated in the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: See Sections 7 and 6

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: See Sections 3 and Appendix 4.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: See Section 5 and attached code.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: See attached code.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: See Section 5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: See Section 5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See Section 5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Anonymous authors, no plagiarism and other ethical concerns from `https://neurips.cc/public/EthicsGuidelines` where followed.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See Section 8

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This does not apply to our task

    Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We implemented the code and cite all packages and dataset used.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide the code and readme file on how to use it.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

Justification: This does not apply to our work.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: This does not apply to our work.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.