

A learning-based harmonic mapping: Framework, assessment, and case study of human-to-robot hand pose mapping

The International Journal of
Robotics Research
2021, Vol. 40(2-3) 534–557
© The Author(s) 2020
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/0278364920962205
journals.sagepub.com/home/ijr


Eunsuk Chong , Lionel Zhang and Veronica J. Santos

Abstract

Harmonic mapping provides a natural way of mapping two manifolds by minimizing distortion induced by the mapping. However, most applications are limited to mapping between 2D and/or 3D spaces owing to the high computational cost. We propose a novel approach, the harmonic autoencoder (HAE), by approximating a harmonic mapping in a data-driven way. The HAE learns a mapping from an input domain to a target domain that minimizes distortion and requires only a small number of input–target reference pairs. The HAE can be applied to high-dimensional applications, such as human-to-robot hand pose mapping. Our method can map from the input to the target domain while minimizing distortion over the input samples, covering the target domain, and satisfying the reference pairs. This is achieved by extending an existing neural network method called the contractive autoencoder. Starting from a contractive autoencoder, the HAE takes into account a distance function between point clouds within the input and target domains, in addition to a penalty for estimation error on reference points. For efficiently selecting a set of input–target reference pairs during the training process, we introduce an adaptive optimization criterion. We demonstrate that pairs selected in this way yield a higher-performance mapping than pairs selected randomly, and the mapping is comparable to that from pairs selected heuristically by the experimenter. Our experimental results with synthetic data and human-to-robot hand pose data demonstrate that our method can learn an effective mapping between the input and target domains.

Keywords

Learning and adaptive system, human-to-robot hand pose mapping, harmonic mapping, contractive autoencoder

1. Introduction

1.1. Human-to-robot hand pose mapping

Finding a natural mapping between two different kinematic structures or domains is of interest in many robotic applications for which an efficient and intuitive operation on the target domain is desired by the human operator. Such applications include programming by demonstration (Chen and Zelinsky, 2003; Dillmann et al., 2000; Kang and Ikeuchi, 1995) and teleoperation (Griffin et al., 2000; Hu et al., 2004; Rohling et al., 1993). In these applications, common approaches are kinesthetic teaching of a robot (Billard et al., 2008), joystick-based control (Nagata et al., 2001), or operation via a miniature model that has a kinematic structure similar to that of the target robot (Portillo-Vélez et al., 2013). The drawbacks of such approaches are that the operator can only control a limited number of degrees of freedom (DOFs) simultaneously and it is often cumbersome to perform (Kormushev et al., 2011). A sophisticated, but intuitive mapping interface would allow the user to

operate complex robotic systems more efficiently (Calinon, 2018).

In particular, the mapping of human hand pose to robotic gripper state, or simply hand pose, is one of the most actively studied subjects in the last three decades (Ciocarlie and Allen, 2009; Cutkosky and Howe, 1990; Ekvall and Kragic, 2004; Gioioso et al., 2013; Iberall, 1997; Kang and Ikeuchi, 1997; Pao and Speeter, 1989; Peer et al., 2008). Robot dexterity is often required to perform difficult tasks including manufacturing (Falco et al., 2013), minimally invasive surgery (Diodato et al., 2018), and space-based operations (Lovchik and Diftler, 1999). Although the

Biomechatronics Laboratory, Department of Mechanical and Aerospace Engineering, University of California–Los Angeles, Los Angeles, CA, USA

Corresponding author:

Eunsuk Chong, Department of Mechanical and Aerospace Engineering, University of California–Los Angeles, Los Angeles, CA 90095, USA.
Email: eschong@ucla.edu

human hand can be easily configured for grasp, tool use, or even as a tool itself (MacKenzie and Iberall, 1994), a natural mapping from a human hand to a robot gripper is not always obvious. An intuitive human-to-robot hand mapping would enable the transformation of skilled actions that are intuitive for humans into functionally equivalent robot hand poses. However, despite aesthetic similarities between the human hand and anthropomorphic robotic grippers, determining a mapping between their poses is not trivial owing to their different kinematic structures.

Most of the existing hand mapping strategies in the public domain can be categorized into three classes: joint-to-joint mapping, Cartesian mapping (usually of fingertip positions), and object-based mapping, as investigated in a recent review by Salvietti (2018). Joint-to-joint and Cartesian mapping impose constraints on the choice of the reference joints or Cartesian points, respectively, on hands/fingers (Gioioso et al., 2013). Such methods are often developed for specific robotic gripper kinematics, thereby limiting the generalizability of the approach. A generic methodology of automatically determining a desirable mapping between two different kinematic structures can save resources for structure-specific developments, and also serve as a universal tool for analyzing and explaining relationships between structures.

An object-based mapping method that can be applied to a wide range of robotic hands is developed and studied in Gioioso et al. (2013) and Salvietti et al. (2014). The main idea is to use virtual ellipsoid objects, which are virtually grasped by a human hand and a robotic gripper, to define the similarity between the two different kinematic structures. Gioioso et al. (2013) presented a mathematically organized framework for encoding human hand movements based on the concept of the *synergy* into robotic hand control while pursuing effective grasping postures. A synergy is a low-dimensional output of a principal component analysis (PCA) that has been applied to hand poses (Santello et al., 1998). However, the method is only shown for synergy-based grasping poses with manually chosen initial hand poses. The mapping between two hand poses is determined based on the initial poses and incremental changes of the virtual ellipsoids. Thus, it is possible that the output of the mapping undesirably reaches joint limits or singular configurations when initial pose pairs are not chosen properly, are transitioning from one pose to another, and/or the hands are performing tasks other than grasping. In addition, as mentioned by the authors, the method is computationally expensive and is not amenable for real-time applications.

In this study, we propose a novel, generalizable methodology for mapping two different domains in a data-driven way. Given a sufficient number of data samples from each of two domains, our approach enables a continuous domain-to-domain mapping based on a few reference input/target pose pairs. Our task can be viewed as a semi-supervised learning problem. Our goal is to define and demonstrate a systematic and general framework for

mapping two different domains—including kinematic structures—in a natural way that can be applied to, but is not limited to, a real-time human-to-robot hand pose mapping.

It is difficult to define a general performance measure or a similarity metric between human and robot hand poses. This is one of the main challenges in determining an effective mapping between human and robot hand poses. In Liarokapis et al. (2013), workspaces are compared for each finger segment of the human and robotic hands to quantifying anthropomorphism of robotic hands. However, the finger segments to be compared need to be chosen heuristically, since the set of fingers and finger segments of robotic hands are not analogous to the human hand in general.

We consider two key requirements for human-to-robot hand pose mapping: First, the image of the map should cover the entire target domain; this allows a human operator to perform as many robot gripper poses as possible. Second, similar human hand poses should map to similar robot gripper poses in a continuous manner; the mapping should be understood intuitively by the operator in order to promote efficient operation.

1.2. Harmonic Mapping

To satisfy the above-mentioned two requirements, we employ a distortion-minimizing mapping between two manifolds, called *harmonic mapping*. We adopt the manifold hypothesis when analyzing our datasets, which assumes that high dimensional real-world data are likely to lie on much lower dimensional manifolds (Cayton, 2005; Narayanan and Mitter, 2010).

In Park and Brockett (1994), the harmonic mapping is suggested as a natural measure of *kinematic dexterity* in the context of robotic mechanisms. This motivates the use of harmonic maps for human-to-robot hand pose mapping; e.g., teleoperation tasks would benefit from a mapping between human and robot hand poses that maximizes kinematic dexterity. Our experimental results show that harmonic mapping can satisfy the two aforementioned key requirements for the mapping.

Most existing harmonic mapping applications are mainly for image processing in between 2D and/or 3D spaces (Choi et al., 2015; Joshi et al., 2007; Shi et al., 2017; Tang et al., 2000; Zhang and Hebert, 1999). In these applications, the harmonic mapping criterion is numerically calculated based on meshes. As size and dimension are crucial factors of computation time, these methods are not suitable for real-time applications with high dimensionality. Lin et al. (2010) linearly approximated the criterion for finding lower-dimensional embeddings of image data, but the computational complexity still grows cubically.

The self-organizing map is a mapping algorithm often used for discretizing high-dimensional input data into a low-dimensional grid (Kohonen, 1990). Similar to our approach, the self-organizing map aims to find an effective

mapping between two domains. However, the algorithm relies on heuristics because there is no known, analytically defined energy function (Erwin et al., 1992; Flexer, 1997; Kohonen, 2013). The self-organizing map is also limited by computational cost when mapping to a grid with greater than three dimensions (Barhak and Fischer, 2001; Bauer and Pawelzik, 1992; Flexer, 1997; Gorricha and Lobo, 2011).

Recently, there have been increasing attempts of applying neural network methods in various disciplines, including robotics (Chong and Park, 2017; Giorelli et al., 2015; He et al., 2016; Levine et al., 2018; Zhu et al., 2017). This advancement is largely a result of the flexibility of the neural network as a global function approximator and its fast estimation and calculation time after training. One such neural network method is the contractive autoencoder (CAE; Rifai et al., 2011), a generative-type neural network. The CAE encodes input data from an observable data domain to output representations in latent space, while contracting the encoded information as a model regularizer. The regularizer term used in CAEs is analogous to the distortion defined in harmonic mapping and, thus, the CAE model can be seen as a special case of harmonic mapping. We incorporate aspects of the CAE to map to the entire, physical target domain, e.g., robot gripper poses, rather than a latent representation.

1.3. The harmonic autoencoder

In this study, we introduce a learning-based harmonic mapping based on the CAE, which we refer to as the *harmonic autoencoder* (HAE). Training the HAE requires a set of sample points from the input and target domains, and a small number of input–target reference pairs. The resultant mapping minimizes distortion over the input samples and covers the target domain, while satisfying the constraints imposed by the reference data pairs. To achieve harmonic mapping, we introduce two terms and add them to the original CAE cost function. With the first term, we penalize estimation errors on reference data pairs so that the resultant mapping is expected to satisfy the data pairs. With the second term, we minimize a distance measure between the mapping output points and the sample points from the mapping’s target domain in order to encourage the mapping to cover the target domain.

Our motivation for using the autoencoder framework is the regularity it provides in an unsupervised way. Applying the autoencoder network structure to physically meaningful variables is one of the notable achievements of our study. Assuming there exists an effective and intuitive mapping between the input and target domains, we posit that the mapped target data should be a regularized representation of the input data, where the autoencoder structure can help with learning the regularity.

We use both synthetic data and human-to-robot hand pose data to test our method. We construct the synthetic data from various geometric plane shapes that include

convex and concave features lying in 2D or 3D space. A number of synthetic data points are sampled from the input and target shapes so that the shapes can be sufficiently covered by the samples. The number of input and target samples are not required to be the same. The number of human-to-robot hand pose data samples are relatively sparse regarding its high-dimensional ambient space (\mathbb{R}^{25} in our case). In both cases, the experimental results show that our proposed method can learn a mapping that successfully covers the target domain while leveraging the beneficial property of minimal distortion from harmonic mapping.

To automate the entire HAE framework, we suggest a simple strategy of selecting reference data pairs adaptively and autonomously during training. Our experimental results show that the adaptive strategy outperforms the case when the data pairs are chosen randomly or uniformly. The result is comparable to the case when the reference data pairs are chosen heuristically by the experimenter. We discuss observations on map folding and twisting, and show that one of our proposed performance metrics for evaluating the learned mapping is sensitive enough to detect and evaluate such extreme cases.

Using our proposed HAE approach, we present a robust mapping that can continuously map between human and robot hand poses in real time. We demonstrate that continuous human hand poses can be mapped to robot hand poses using a small number of reference pose pairs. This is especially useful for scenarios in which obtaining a large number of input–target data pairs, e.g., matching human and robot hand poses, is expensive with regards to time and effort.

The main contribution of this article is the presentation of a novel and efficient method for mapping between two different physically meaningful spaces (e.g., kinematic structures) using only a small number of reference pose pairs. The proposed HAE method overcomes limitations of existing mapping methods such as latent-only target spaces (Bishop et al., 1998; Rifai et al., 2011) and/or high computational cost (Kohonen, 2013; Shi et al., 2017), thereby making possible more real-time high-dimensional applications, e.g., human-to-robot hand pose mapping. To the best of the authors’ knowledge, our study is the first implementation of harmonic mapping in a data-driven way. We offer an extensive and mathematically exact foundation of the proposed approach, followed by a comprehensive evaluation with experiments using synthesized data and human-to-robot hand pose data.

This article is organized as follows. Section 2 reviews harmonic mapping and CAE concepts, and then describes the overall framework and details of our HAE method. In Section 3, we introduce four metrics to assess the mapping performance, and discuss experimental results with synthetic data and human-to-robot hand pose data. Concluding remarks with future research directions are discussed in Section 4. Notation frequently used in this article are summarized in Appendix B.

2. Learning methods

2.1. Preliminaries

2.1.1. Harmonic mapping. In this section, we briefly describe harmonic mapping, the CAE, and their relationship with one another.

An intuitive physical interpretation is offered in Eells and Sampson (1964) and Park and Brockett (1994): suppose it was desired to cover a solid surface X by an elastic material Q ; the distortion $d(f)$ can be considered as the elastic tension of each point in Q covering X . Harmonic maps result in an equilibrium of minimum elastic energy. We provide in the following a brief overview of harmonic maps. For a complete definition and explanation of harmonic mapping associated with Riemannian geometry, we refer the reader to Eells and Sampson (1964).

A harmonic mapping is defined as a mapping which minimizes the *distortion* between two manifolds. Let $f: Q \rightarrow X$ denote a smooth mapping between two manifolds, $Q \subset \mathbb{R}^{D_Q}$ and $X \subset \mathbb{R}^{D_X}$, and let J be its Jacobian, i.e., $J = \frac{\partial f}{\partial q} \in \mathbb{R}^{D_X \times D_Q}$ where $q \in Q$. A distortion density of f is defined as

$$d(f) = \frac{1}{2} \text{Tr}(J^T G J H^{-1}) \quad (1)$$

where $G \in \mathbb{R}^{D_X \times D_X}$ and $H \in \mathbb{R}^{D_Q \times D_Q}$ are metric tensors on X and Q , respectively, and Tr denotes the trace of a matrix. The mapping f is harmonic if the derivative of $\int_Q d(f) dV$ with respect to f is zero, as a calculus of variations problem, where V is the volume measure on Q .

In most applications, an appropriate set of mapping constraints are required to determine a non-trivial harmonic mapping f between two manifolds. Boundary conditions are needed, such as input and target reference pairs, to prevent the image of the mapping from shrinking to a single point.

2.1.2. Contractive autoencoder (CAE). A closely related neural network model called the CAE was proposed by Rifai et al. (2011). It is defined as

$$\text{CAE}(f, g; \tilde{Q}) = \frac{1}{N_{\tilde{Q}}} \sum_{q \in \tilde{Q}} \left\{ \frac{1}{2} \|q - g \circ f(q)\|^2 + \lambda_1 \|J\|_{\mathcal{F}}^2 \right\} \quad (2)$$

where $\tilde{Q} = \{q_1, \dots, q_{N_{\tilde{Q}}}\}$ is $N_{\tilde{Q}}$ data samples from the input domain Q , g is a reconstruction function from X to Q , $\|\cdot\|_{\mathcal{F}}^2$ denotes the squared Frobenius norm (sum of squared elements within a matrix), and λ_1 is a user-designated scalar coefficient.

In most autoencoder applications, including Rifai et al. (2011), Q and X are of a visible and latent variable space, respectively. Note that if G and H are the identity matrices (i.e., Euclidean distances are assumed in Q and X such

that all dimensions in Q and X are equally weighted), then Equation (1) is reduced to $\|J\|_{\mathcal{F}}^2/2$ (see Appendix C). This suggests that the second term of Equation (2), the regularizer of CAE, is analogous to the distortion in the context of harmonic maps. More specifically, the term can be interpreted as the expectation of the distortion density (when G and H are identity matrices) over the data sample distribution, $p(q)$:

$$\frac{1}{N_{\tilde{Q}}} \sum_{q \in \tilde{Q}} \|J\|_{\mathcal{F}}^2 \approx \int_Q \|J\|_{\mathcal{F}}^2 p(q) dq \quad (3)$$

Note that if $p(q)$ is a uniform distribution, the volume measure on Q can be defined as $dv = p(q) dq$, where all the sample points will occupy the same volume. Thus, the regularizer term of CAE can be seen as a statistical approximation of harmonic mapping. Although the CAE does not set explicit boundary constraints on the targeted latent space, the image of f in latent space typically does not shrink to a single point. This is because the image needs to contain sufficient information for reconstructing the visible variable, enforced by the first term in Equation (2), which itself is the basic autoencoder. The CAE aims to find a good representation of input features in a high-dimensional target latent space without labeled data. This latent representation is fed to the next layer of a machine learning model, which improves the overall performance of a target task, such as regression or classification.

In this study, we propose to use CAE to implement harmonic mapping given two domains. Given samples of input states, $q \in \mathbb{R}^{D_Q}$, and samples of output states, $x \in \mathbb{R}^{D_X}$, we seek relationships between the two states, f and g . A small number of reference input–target sample pairs are assumed to be available. The reference pairs provide the only explicit information on the input–target matching. The distortion of f over the input domain is minimized while the mapping covers the target domain sufficiently.

Note that the baseline CAE method does not promote coverage of the target domain. We aim to find a mapping to a physically meaningful space while covering the target domain with minimal distortion given only a few reference data pairs. In the next part, we introduce two additional terms to Equation (2), the original CAE, in order to implement harmonic mapping in a data-driven way. With the addition of these two terms, we define our proposed method, the HAE.

2.2. Pinning and boundary attraction

We constrain the model to satisfy a set of L reference data pairs (or “pinned points”), $\mathcal{S} = \{(q_i, x_i) | i = 1, \dots, L\}$. This is achieved by assigning a “pinning” cost term to the model optimization as follows:

$$\text{Pin}(f, g; \mathcal{S}) = \frac{1}{L} \sum_{(q, x) \in \mathcal{S}} (\|x - f(q)\|^2 + \alpha \|q - g(x)\|^2) \quad (4)$$

where α is a weighting coefficient for balancing the two Euclidean distances. The two terms in Equation (4) are ordinary error functions for feed-forward neural networks in each mapping direction. By minimizing the error terms in Equation (4), both f and g are driven to match the reference data set \mathcal{S} . This constraint resists the mapping's tendency to shrink to a single point by pinning the given input points to corresponding target points. Still, unless a sufficient number of reference pairs are provided, the mapping will shrink as much as possible to form concave shapes (and, in some cases, wrinkles) while being stretched to the reference pairs. Imagine stretching an elastic sheet using only pushpins to tack the sheet down at certain locations. Note that this approach differs from the traditional training of neural networks with an error-minimizing cost function, which requires a sufficiently large number of training samples. We assume that only a small number of reference data pairs are available for training and treat them as constraints.

To cover the target domain more fully, we introduce a "boundary attraction" cost term based on a distance measure between two point clouds, known as the *undirected Chamfer distance* (UCD) (Athitsos and Sclaroff, 2003):

$$\begin{aligned} \text{UCD}(A, B) = & \frac{1}{N_A} \sum_{a \in A} \min_{b \in B} \|a - b\|^2 \\ & + \frac{1}{N_B} \sum_{b \in B} \min_{a \in A} \|a - b\|^2 \end{aligned} \quad (5)$$

where N_A and N_B denote the number of elements in sets A and B , respectively. For computational efficiency, both terms on the right-hand side of Equation (5) can be obtained by forming an N_B -by- N_A distance matrix in which element (i, j) is $D_{ij} = \|a_i - b_j\|^2$; the first term is the average of the minimum values of each row, and the second term is the average of the minimum values of each column. Hence, the entire UCD term can be efficiently calculated matrix-wise. Then, the boundary attraction in the target domain is defined as $\text{UCD}(\tilde{X}, f(\tilde{Q}))$, where \tilde{X} is the set of target domain samples and $f(\tilde{Q})$ is the mapping outputs from the input samples \tilde{Q} . The boundary attraction in the input domain can also be defined in the same manner, $\text{UCD}(\tilde{Q}, g(\tilde{X}))$. Combining the two, we define a bidirectional boundary attraction as follows:

$$\text{BA}(f, g; \tilde{Q}, \tilde{X}) = \text{UCD}(\tilde{X}, f(\tilde{Q})) + \alpha \text{UCD}(\tilde{Q}, g(\tilde{X})) \quad (6)$$

where we use the same weighting coefficient α as in Equation (4) in order to equally control for the effects of the Pin and BA terms in the input domain. In our case, we set the value of α to 0 or 1. Note that \tilde{Q} and \tilde{X} are independently sampled (not necessarily the same number of samples) from the input and target domains, respectively. The BA term can be evaluated without input-target pairing information.

By minimizing Equation (6), boundaries of the two point clouds in each domain are attracted to each other. That is, any point in one point cloud that is farther from another point cloud is penalized more. The pinning and

boundary attraction constraints encourage the mapping to match reference data points and cover both sample domains, respectively.

2.3. Activation functions, Jacobian, and distortion

In this study, we employ two-layer network models for f and g using the rectified linear unit ReLU (Nair and Hinton, 2010) and linear transformation as activation functions:

$$\hat{x} = f(q) = W_2 h_1 + b_2, \quad h_1 = \text{ReLU}(W_1 q + b_1) \quad (7)$$

$$\hat{q} = g(\hat{x}) = W_4 h_2 + b_4, \quad h_2 = \text{ReLU}(W_3 \hat{x} + b_3) \quad (8)$$

where the matrix W_i and vector b_i for $i = 1, 2, 3, 4$ are model parameters.

Applying the chain rule to Equations (7) and (8), the Jacobian matrix J , and, accordingly, the distortion $\|J\|_F^2$, can be calculated as follows, per Appendices D and E:

$$J = W_2 \cdot \text{Diag}(z) \cdot W_1 \quad (9)$$

$$\|J\|_F^2 = \sum_{ij} (zz^T \odot W_2^T W_2 \odot W_1 W_1^T)_{ij} \quad (10)$$

where $\text{Diag}(z)$ is a diagonal matrix of which the i th diagonal element $z_i = 1$ if $(W_1 q + b_1)_i > 0$ and 0 otherwise. The \odot operator denotes element-wise multiplication.

2.4. Adaptive reference point selection

As we assume that obtaining an input-target data pair is expensive, we propose an iterative method for efficiently selecting these reference pairs based on intermediate training performance. Given an existing reference data pair set \mathcal{S}_ℓ , and trained model function f_ℓ , at learning session step l , we first determine the farthest point from the area covered by the mapping f_ℓ in the target domain x^* :

$$a(x) = \min_{q \in \tilde{Q}} \|x - f_\ell(q)\|^2 \quad (11)$$

$$x^* = \underset{x \in X}{\text{argmax}} \ a(x) \quad (12)$$

In practice, we use \tilde{X} instead of X in Equation (12), assuming that the target samples sufficiently cover the target domain. After identifying q^* that maps to x^* , the next set of reference pairs is updated to $\mathcal{S}_{\ell+1} = \mathcal{S}_\ell \cup \{(q^*, x^*)\}$. The mapping is then stretched to the new reference pair (pinned points) using Equation (4).

In this study, we constrain the order to first choosing x^* , then q^* . We choose this order by considering a human-to-robot hand pose mapping scenario in practice, where the reference pairs are determined on demand during the mapping-learning phase. This can be viewed as an iterative human-in-the-loop training process. Once x^* is determined, the corresponding robot hand pose is presented to the human operator (e.g., via the actual robot or a computer

simulation). The human operator then uses intuition to mimic the robot hand pose to set q^* . Our underlying assumption is that human intuition is advantageous in achieving a natural mapping. However, this order can be modified according to the application. From the process, we obtain the reference pairs (q^*, x^*) that are used to define the reference data set \mathcal{S} in Equation (4). A demonstration of this process is shown in Extension 1.

2.5. The HAE framework

Our overall HAE cost function is formulated as

$$\text{HAE}(\tilde{Q}, \tilde{X}, \mathcal{S}; \Theta) = \text{CAE}(\tilde{Q}) + \lambda_2 \text{Pin}(\mathcal{S}) + \lambda_3 \text{BA}(\tilde{X}, \tilde{Q}) \quad (13)$$

where CAE refers to the baseline CAE in Equation (2), Pin is the “pinning” cost term defined in Equation (4), BA represents the boundary attraction defined in Equation (6), λ_2 and λ_3 are weighting coefficients, and $\Theta = \{W_1, W_2, W_3, W_4, b_1, b_2, b_3, b_4\}$ is a set of model parameters. We refer to the model in Equation (13) as the HAE.

Our learning framework is stated in Algorithm 1. Note that L is the total number of reference pairs used for training the model. As a result, learning occurs over a total of $L + 1$ sessions.

In our experiments, we also compare the HAE model performance with and without applying the adaptive reference point selection. When the adaptive selection is not applied, a fixed set of reference pairs $\mathcal{S}_{\text{fixed}}$ are provided. In those cases, we set $\mathcal{S}_0 = \mathcal{S}_{\text{fixed}}$ and skip lines 3–5 in Algorithm 1.

3. Experimental results and discussion

3.1. Performance measures

To assess and compare the resultant mappings in our experiments, we define four performance measures.

Algorithm 1 HAE framework

Given:

- \tilde{Q} : data samples from the input domain
- \tilde{X} : data samples from the target domain
- \mathcal{S}_0 : an initial set of reference data pairs ($= \emptyset$)

Parameters:

- Θ : model parameters of HAE in Equation (13)
 - 1: **for** $\ell \leftarrow 0, \dots, L-1$ **do**
 - 2: $\Theta \leftarrow \underset{\Theta}{\text{argmin}} \text{HAE}(\tilde{Q}, \tilde{X}, \mathcal{S}_\ell; \Theta) \triangleright$ with M_1 epochs
 - 3: Get next target reference point x^* from Equations (11) and (12)
 - 4: Identify q^* that maps to x^* \triangleright e.g., using human intuition
 - 5: $\mathcal{S}_{\ell+1} \leftarrow \mathcal{S}_\ell \cup (q^*, x^*)$
 - 6: **end for**
 - 7: $\Theta \leftarrow \underset{\Theta}{\text{argmin}} \text{HAE}(\tilde{Q}, \tilde{X}, \mathcal{S}_L; \Theta) \triangleright$ with M_2 epochs
-

3.1.1. Sample mean distortion. As is commonly done, we calculate the sample mean distortion \bar{d} of Equation (3) over the test set:

$$\bar{d}(f; \tilde{Q}) = \frac{1}{N_{\tilde{Q}}} \sum_{q \in \tilde{Q}} \|J\|_{\mathcal{F}}^2 \quad (14)$$

A small value of the distortion measure indicates small average elastic energy stored in the mapping. In Park and Brockett (1994), $\int_Q d(f)dV$ is suggested to evaluate the global kinematic dexterity of robotic mechanisms when mapping from joint space to end-effector workspace. In Rifai et al. (2011), the measure of Equation (14) is treated as a local contraction measure. The measure was shown to be strongly correlated to the model performance, which was validated by taking the latent output representation as input to various classification tasks; a good latent representation of raw data results in high classification accuracy.

Note that the distortion measure cannot be the sole indicator of performance since a mapping to a single point would output the minimum value. Thus, we also need a performance measure to evaluate coverage of the mapping over the target domain, which we describe next.

3.1.2. Target domain volume occupancy. To evaluate the mapping coverage of the target domain, we define a volume occupancy measure VO based on sample data points as follows:

$$c(x; \tilde{Q}) \triangleq \begin{cases} 1, & \text{if } \min_{q \in \tilde{Q}} \|x - f(q)\|^2 < \epsilon \\ 0, & \text{otherwise} \end{cases} \quad (15)$$

$$VO(f; \tilde{Q}, \tilde{X}) = \frac{1}{N_{\tilde{X}}} \sum_{x \in \tilde{X}} c(x; \tilde{Q}) \quad (16)$$

where ϵ is a user-defined distance threshold in the target domain. We choose ϵ such that a set of $N_{\tilde{Q}}$ small hyperspheres with radius $\sqrt{\epsilon}/2$, uniformly distributed in the target domain, reflects the entire volume of the target domain. The choice of ϵ is based on the assumption that \tilde{Q} and \tilde{X} are uniformly sampled from their respective domains.

This VO performance measure evaluates the percentage of the target domain volume that is occupied by the mapping's output point cloud. As available, the target domain samples \tilde{X} can be replaced by dense grid points covering the target domain, which will allow a deterministic evaluation of the coverage. However, as dimensionality increases, such a dense grid becomes computationally expensive or impossible to obtain experimentally. For a consistent use of the target domain volume occupancy measure, we use random samples \tilde{X} for each of the synthetic and human-to-robot hand pose data experiments, so that the volume occupancy is evaluated in a statistical way.

3.1.3. K -nearest neighbor inverse image dissimilarity. We assume that similar input points should map to similar target points. We evaluate whether neighboring points in the target domain are indeed mapped from neighboring points in the input domain.

We introduce a novel measure, the K -nearest neighbor inverse image dissimilarity metric $d-Knn$, defined as follows:

$$d-Knn(f; \tilde{Q}) = \frac{1}{N_{\tilde{Q}}} \sum_{\tilde{q} \in \tilde{Q}} \frac{\sum_{\hat{x}_{Knn}} \|f^{-1}(\hat{x}) - f^{-1}(\hat{x}_{Knn})\|^2}{\sum_{\hat{x}_{Knn}} \|\hat{x} - \hat{x}_{Knn}\|^2} \quad (17)$$

where $f(\tilde{Q}) \triangleq \{f(q) | q \in \tilde{Q}\}$, and \hat{x}_{Knn} denotes a point in K -nearest neighbors of \hat{x} in the target space. That is, $\hat{x}_{Knn} \in \{x | x \text{ is the } k\text{th nearest point of } \hat{x}, k = 1, \dots, K\}$. We set the total number of neighboring points $K = 10$ in our experiments. Equation (17) measures the average distance between the inverse image of a target point $f^{-1}(\hat{x})$ and the inverse image of the target point's neighboring point $f^{-1}(\hat{x}_{Knn})$. Note that $f^{-1}(\hat{x})$ denotes $q \in \tilde{Q}$ such that $f(q) = \hat{x}$ from Equation (7). Thus, $f^{-1}(\hat{x})$ and $f^{-1}(\hat{x}_{Knn})$ are directly available by identifying corresponding pairs from \tilde{Q} . The average distance between the inverse images of the target point and its neighboring point is normalized by the average distance between the target point \hat{x} and its neighboring point \hat{x}_{Knn} .

This dissimilarity measure will have large values when very dissimilar inputs are mapped to similar, neighboring points. For example, two distant points in the input domain Q can be mapped to a narrow region in the target domain X . The output mapping will then be undesirably “twisted” or “folded” and the dissimilarity measure will be large.

3.1.4. Normalized mean square error. Since ground-truth data for harmonic mapping are available when using synthetic data, we measure the difference between the ground-truth target and the output from the trained model in our synthetic data experiment (Section 3.2). The normalized mean square error is defined as follows:

$$NMSE(f; \mathcal{D}_{Q,X}) = \frac{1}{N_{\mathcal{D}_{Q,X}} \sigma^2(x)} \sum_{(q,x) \in \mathcal{D}_{Q,X}} \|x - f(q)\|^2 \quad (18)$$

where $\mathcal{D}_{Q,X}$ is a set of $N_{\mathcal{D}_{Q,X}}$ ground-truth input–target pairs (q, x) , and $\sigma^2(x)$ denotes the variance of x over $\mathcal{D}_{Q,X}$. This measure evaluates how close the resultant mapping is to the numerically generated harmonic mapping in terms of squared Euclidean distance sum, normalized with respect to the sample variance. Note that this measure is only applied to the synthetic data experiment to assess the proposed mapping method. It is not applicable in the human-to-robot hand pose data experiment since there are no available ground-truth data pairs available for human-to-robot hand pose mapping.

3.2. Experiment with synthetic data

In order to analyze and verify our models with various controlled conditions, we ran experiments using synthetic data on our model. The results were evaluated using the performance measures mentioned previously.

To validate our model, we evaluated seven different cost functions (Table 1): pinning (Pin), CAE with pinning (CAE + Pin), Pin with boundary attraction (Pin + BA), Pin with boundary attraction and distortion minimization ($\|J\|_{\mathcal{F}}^2 + \text{Pin} + \text{BA}$), HAE without adaptive reference point selection (HAE) for the cases $\alpha = 0$ and $\alpha = 1$, and HAE with adaptive reference point selection (HAE (adap)) with $\alpha = 1$. Recall that α is the weighting coefficient for the input domain cost terms used in Equations (4) and (6). The cost term Pin is defined by Equation (4); CAE + Pin is Equation (13) with $\lambda_3 = 0$. Note that CAE without any constraint will result in an arbitrary mapping which is of little interest. We set $\alpha = 0$ for all models unless otherwise noted. The set of reference pairs \mathcal{S} for all the models, except for HAE (adap), were randomly selected from available samples using the same random seed over the compared models.

3.2.1. Synthetic data. We chose various shapes, both 2D and 3D, that include convex and concave features. By

Table 1. Cost function terms and applied constraint features for the models compared in our experiments. In the second column, AE denotes autoencoder. In the last column, equation numbers are referenced.

Cost functions	Constraints				Equations
	AE	$\ J\ _{\mathcal{F}}^2$	Pin	BA	
(Baselines)					
Pin			✓		λ_2 (4)
CAE + Pin	✓	✓	✓		(2) + λ_2 (4)
<hr/>					
Pin + BA			✓	✓	λ_2 (4) + λ_3 (6)
$\ J\ _{\mathcal{F}}^2 + \text{Pin} + \text{BA}$		✓	✓	✓	λ_1 (3) + λ_2 (4) + λ_3 (6)
HAE ($\alpha=0$)	✓	✓	✓	✓	(13)
HAE ($\alpha=1$)	✓	✓	✓	✓	(13)
HAE (adap, $\alpha=1$)	✓	✓	✓	✓	(13)

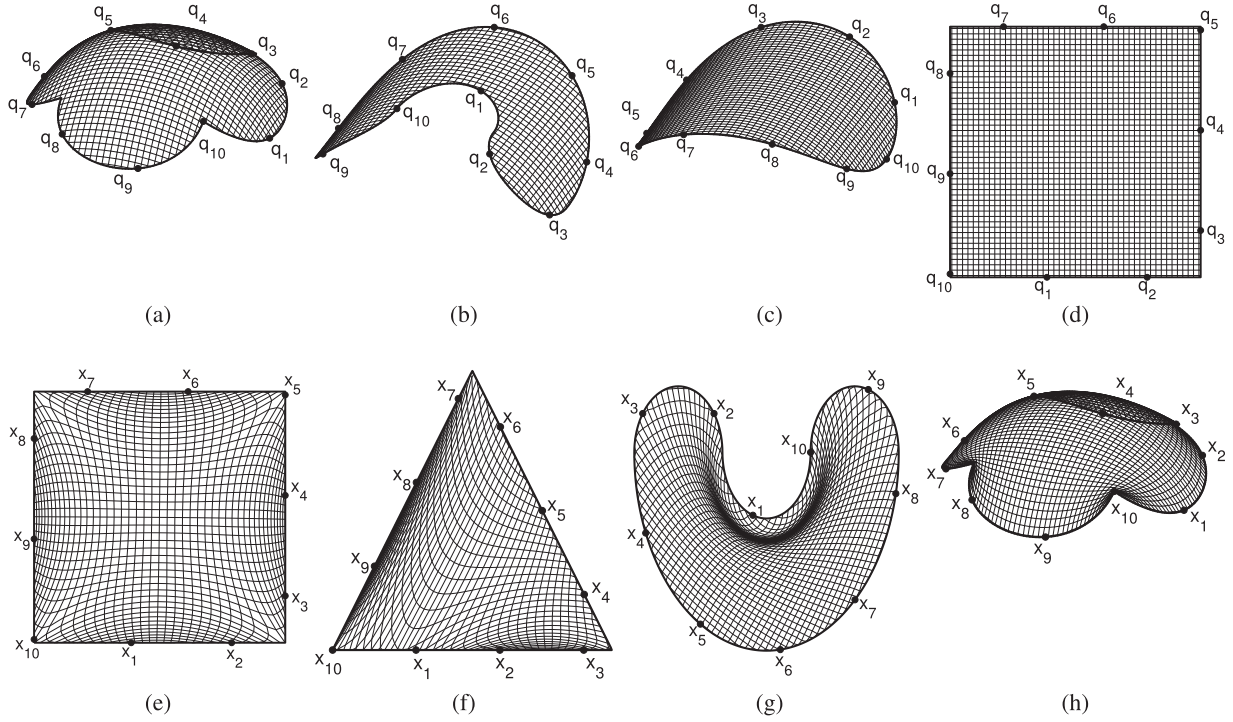


Fig. 1. Synthetic harmonic mapping data of four types of input–target pairs. Top row: input domains Q , Bottom row: target domains X ; (a), (e) SP1 = 3D clover to 2D square; (b), (f) SP2 = 3D U-shape to 2D triangle; (c), (g) SP3 = 3D circle to 2D U-shape; (d), (h) SP4 = 2D square to 3D clover. Boundary segment points with number of 436–456 are matched between input and target domains. For clarity, only 10 representative matching points are shown for each shape: (q_i, x_i) for $i \in \{1, \dots, 10\}$.

choosing from a wide variety of shapes, we show the versatility of our method and its potential as a universal mapping tool. The experiment with synthetic data not only allows us to compare the resultant mappings with ground-truth maps, but also to visually inspect the resultant mappings for topological errors, e.g., twisting and folding.

We generated synthetic data from \mathbb{R}^3 to \mathbb{R}^2 for three cases and from \mathbb{R}^2 to \mathbb{R}^3 for one case. The four input–target domain pairs are shown in Figure 1. The grid covering each input domain was determined by a 48-by-48 2D square grid which compactly covers bounded areas within each shape: clover, U-shape, circle, and square. The clover grid was projected onto a sphere in 3D space (Figure 1(a)), and the grids of the U-shape and circle were projected onto a cylinder in 3D space (Figure 1(b) and (c)). The square grid was retained as 2D input (Figure 1(d)). The number of resultant grid points in the input shapes 3D clover, 3D U-shape, 3D circle, and 2D square were 1,409, 952, 1,709, and 2,209, respectively. We took 50% randomly permuted samples from the grid points of each input shape as a training set, and used the rest as a test set.

The input shapes were paired with 2D square, 2D triangle, 2D U-shape, and 3D clover, respectively (Figure 1(e)–(h)). This resulted in four input Q and target X shape pairs (SP): **SP1** = 3D clover input and 2D square output, **SP2** = 3D U-shape input and 2D triangle output, **SP3** = 3D circle input and 2D U-shape output, and **SP4** = 2D square input and 3D clover output. For each of the shape pairs, the ground-truth set of input–target sample pairs $\mathcal{D}_{Q,X}$ was

synthesized using numerically implemented harmonic maps given the input grid points, as illustrated in Figure 1. Each element of the Jacobian matrix was constructed using numerical differentiation over grid points defined on each of the input shapes. The ground-truth pairs were utilized for evaluating the NMSE defined in Section 3.1.4, and compared with the output of the HAE model. Note that the ground-truth target data were not used for training the HAE model. Instead, 8,000 data points were sampled from each of the target shapes for training in order to ensure that input samples and target samples are sampled independently.

3.2.2. Model learning configuration. Using the four sets of input–target shape pairs, SP1 to SP4, the HAE of Equation (13) was minimized over the training set with $8,000 + 100L$ learning epochs ($M_1 = 100$, $M_2 = 8,000$) for L number of reference pairs. To determine the number of reference pairs for learning an effective mapping, we trained the HAE model with a range of reference pairs (between 1 and 8). In each session, the next reference pair is determined and added to \mathcal{S} for the next session.

The hidden variables h_1 and h_2 were both set to values of 300. We set the coefficients $(\lambda_1, \lambda_2, \lambda_3)$ as $(0.005, 1.0, 5.0)$ for all four shape pairs. The coefficient λ_3 was kept at zero until the 1,000th epoch of the last session, then linearly increased to its full value over 2,000 epochs to reduce sudden impact during optimization. The purpose of this dynamic parameter adaptation is further explained in Section 3.2.3. One can modify these epoch numbers

depending on the learning rate of the HAE model. Having λ_3 start at its full value was found to cause folding and twisting of the mapping.

These hyperparameters were manually selected so that the map learning process would be performed robustly over the four different shape pairs with an identical set of hyperparameter values. Of the hyperparameter set, only λ_2 was changed when switching from the synthetic data to the human and robot hand pose data in Section 3.3.

The epoch number for the final learning session M_2 was set larger than the epoch number for the previous learning sessions M_1 in order to enable model flexibility during the reference pair selections, and then to stabilize the model once all of the reference pairs had been selected. Controlling the number of epochs in this way helps the model to avoid getting stuck in local minima before all of the reference pairs have been selected.

The training was conducted using Tensorflow-gpu 1.3.0 in Python 2.7 running on a PC with an Intel Core i7-6800K CPU at 3.40 GHz, and a GeForce GTX 1080 GPU. We used the Adam optimizer with Nesterov momentum (Nadam) (Dozat, 2016) with an initial learning rate of 5×10^{-4} and exponential weight decay of 0.99 after 300 steps. The training procedure consumed, on average, 76 seconds per model (with $L = 4$) in our experimental environment.

3.2.3. Results and discussion. Each of the seven model types were evaluated using the four measures defined in Section 3.1: sample mean distortion (\bar{d}), target domain volume occupancy (VO), K -nearest neighbor inverse image dissimilarity (d -Knn), and NMSE. For each model type, all the data sampling, training, and testing procedures were repeated 10 times; mean and standard deviation values of the measure evaluations are reported in Table 2 for $L = 4$.¹

The main purpose for evaluating the Pin and CAE + Pin models is to highlight the role of each term in the HAE cost function, and to provide baseline evaluation values. Another purpose is to show that our method achieves desired properties of harmonic mapping compared with those baseline methods. Note that Pin and CAE + Pin do not utilize information from the target domain (data samples \tilde{X}), which was used to evaluate the boundary attraction terms in other models.

Not surprisingly, CAE + Pin achieved the best performance for all the shape pairs with respect to sample mean distortion \bar{d} , because CAE + Pin explicitly minimizes the distortion term with relatively loose constraints. On the other hand, CAE + Pin failed to cover the target domains, resulting in low performances with respect to the volume occupancy measure, VO . However, by adding the boundary attraction constraint, we show that significantly higher performance with respect to VO can be achieved simultaneously.

Among the methods achieving comparable performance for VO , HAE ($\alpha = 0$) performed the best out of all shape pair cases in terms of distortion, \bar{d} . For all shape pairs, the improvement of the \bar{d} values of HAE ($\alpha = 0$) are marginal

Table 2. Model performance comparison for synthetic data (test set) with four reference pairs ($L = 4$). Seven different models are compared with respect to sample mean distortion (\bar{d}), target domain volume occupancy (VO), K -nearest neighbor inverse image dissimilarity (d -Knn), and NMSE over four cases: (a) SP1 = 3D clover to 2D square, (b) SP2 = 3D U-shape to 2D triangle, (c) SP3 = 3D Circle to 2D U-shape, and (d) SP4 = 2D square to 3D clover. Each model is trained and tested for 10 repetitions. Mean and standard deviation (in parentheses) values are shown. Bold text indicates the best performance for the given measure.

(a) 3D clover to 2D square				
Cost functions	Measures			
($L = 4$)	\bar{d}	VO	d -Knn	NMSE
<hr/>				
(Baselines)				
Pin	1.715 (0.359)	0.561 (0.118)	90.413 (72.694)	0.271 (0.168)
CAE + Pin	0.914 (0.286)	0.464 (0.093)	11.646 (0.961)	0.319 (0.099)
<hr/>				
Pin + BA	2.896 (0.212)	0.974 (0.009)	13.595 (4.870)	0.084 (0.029)
$\ J\ _F^2$ + Pin + BA	1.937 (0.098)	0.971 (0.006)	10.506 (4.130)	0.096 (0.043)
HAE ($\alpha=0$)	1.786 (0.140)	0.969 (0.003)	5.177 (1.068)	0.108 (0.052)
HAE ($\alpha=1$)	1.839 (0.081)	0.971 (0.007)	4.144 (0.372)	0.109 (0.040)
HAE (adap, $\alpha=1$)	2.085 (0.034)	0.982 (0.004)	4.503 (0.166)	0.027 (0.002)
<hr/>				
(b) 3D U-shape to 2D triangle				
Cost functions	Measures			
($L = 4$)	\bar{d}	VO	d -Knn	NMSE
<hr/>				
(Baselines)				
Pin	1.154 (0.267)	0.548 (0.134)	168.100 (118.162)	0.572 (0.330)
CAE + Pin	0.565 (0.136)	0.472 (0.090)	13.845 (5.984)	0.594 (0.338)
<hr/>				
Pin + BA	1.860 (0.207)	0.965 (0.021)	71.037 (33.390)	0.387 (0.459)
$\ J\ _F^2$ + Pin + BA	1.186 (0.140)	0.958 (0.019)	46.145 (27.227)	0.445 (0.415)
HAE ($\alpha=0$)	1.132 (0.144)	0.950 (0.030)	13.121 (7.267)	0.516 (0.484)
HAE ($\alpha=1$)	1.394 (0.225)	0.948 (0.039)	9.827 (3.012)	0.298 (0.206)
HAE (adap, $\alpha=1$)	1.533 (0.128)	0.966 (0.016)	8.018 (1.226)	0.082 (0.057)
<hr/>				
(c) 3D circle to 2D U-shape				
Cost functions	Measures			
($L = 4$)	\bar{d}	VO	d -Knn	NMSE
<hr/>				
(Baselines)				
Pin	1.015 (0.208)	0.388 (0.134)	547.117 (286.404)	0.529 (0.235)

(continued)

Table 2. Continued

(c) 3D circle to 2D U-shape

Cost functions	Measures			
($L = 4$)	\bar{d}	VO	d -Knn	NMSE
CAE + Pin	0.583 (0.106)	0.390 (0.059)	14.526 (3.753)	0.525 (0.397)
Pin + BA	2.263 (0.199)	0.954 (0.022)	62.308 (42.230)	0.361 (0.403)
$\ J\ _F^2 + \text{Pin} + \text{BA}$	1.598 (0.116)	0.923 (0.045)	110.493 (107.044)	0.566 (0.727)
HAE ($\alpha=0$)	1.533 (0.141)	0.924 (0.051)	11.260 (6.792)	0.341 (0.449)
HAE ($\alpha=1$)	1.820 (0.339)	0.920 (0.045)	13.567 (12.567)	0.295 (0.182)
HAE (adap, $\alpha=1$)	1.671 (0.032)	0.968 (0.005)	5.584 (0.229)	0.133 (0.010)

(d) 2D square to 3D clover

Cost functions	Measures			
($L = 4$)	\bar{d}	VO	d -Knn	NMSE
(Baselines)				
Pin	2.439 (0.973)	0.108 (0.060)	2.981 (4.154)	0.764 (0.399)
CAE + Pin	1.026 (0.438)	0.052 (0.025)	6.318 (1.764)	0.604 (0.308)
Pin + BA	5.533 (0.486)	0.873 (0.020)	5.113 (4.706)	0.274 (0.158)
$\ J\ _F^2 + \text{Pin} + \text{BA}$	4.789 (0.613)	0.858 (0.021)	10.037 (8.678)	0.325 (0.134)
HAE ($\alpha=0$)	4.565 (0.167)	0.868 (0.015)	1.877 (0.589)	0.275 (0.093)
HAE ($\alpha=1$)	4.813 (0.111)	0.874 (0.012)	1.184 (0.250)	0.218 (0.071)
HAE (adap, $\alpha=1$)	5.340 (0.189)	0.878 (0.007)	1.319 (0.121)	0.146 (0.029)

(except for SP1) compared with the $\|J\|_F^2 + \text{Pin} + \text{BA}$ model, which also explicitly minimizes the distortion (p -values of the standard one-sample t -test for the shape pairs are 8.2×10^{-4} , 0.21, 0.19, 0.24 for SP1, SP2, SP3, SP4, respectively), but are significantly larger compared with Pin + BA without the distortion-reducing term $\|J\|_F^2$ (p -values of the t -test are 4.6×10^{-7} , 3.2×10^{-5} , 1.3×10^{-5} , 1.5×10^{-4} for SP1, SP2, SP3, SP4, respectively).

HAE (adap, $\alpha = 1$) outperformed all other models with respect to VO and NMSE for SP1 and SP4 (Table 2(a) and (d)), and with respect to VO , d -Knn, and NMSE for the cases of SP2 and SP3 (Table 2(b) and (c)). The four performance measure values for SP1 are plotted as a function of learning epoch in Figure 2 (see Appendix F for SP2, SP3, and SP4 results). The HAE (adap, $\alpha = 1$) converged more slowly than other models, but gained its above-stated competitiveness after 2,000 epochs of learning. Note that it took less than two minutes to complete 8,000 epochs for the number of reference pairs ranging from 1 to 8.

Snapshots of the map learning process of HAE (adap, $\alpha = 1$, $L=4$) for the four input–target shape pairs are illustrated in Figure 3. Each image of the maps started in a contracted state. Then, the images were stretched to the next reference point which is chosen as the most vacant point within the target domain. All of the reference pairs were matched after five sessions ($L + 1$ sessions given $L=4$ reference pairs) of training (including the initial session, $l = 0$), then the boundaries were matched via boundary attraction from Equation (6).

The final learned mappings are shown in the last column of Figure 3 for each of the four shape pairs. In each case, it is shown that the learned mapping for each shape pair successfully covers the target domain area with matched boundaries. The resultant maps showed similar grid structures to the original harmonic maps as shown in Figure 1(e)–(h), however, they also had noticeable wrinkles compared with their ground-truth counterparts. Note that the original harmonic maps were calculated by numerical computation of derivatives with sufficient time and number of boundary matching point information. Our method serves as an efficient tool for approximating harmonic maps given data samples and limited input–target observations, with fast model construction and real-time estimation for new data points.

Volume occupancy and NMSE performances for SP1 with respect to the number of provided reference pairs are shown in Figure 4 and 5 (see Appendix F for SP2, SP3, and SP4 results). In addition to HAE and HAE (adap), HAE with heuristically chosen reference pairs, HAE (fixed), was also compared; uniformly segmented boundary points in each of the input and target domains were selected for HAE (fixed). All of the models consistently reached target volume occupancy larger than 0.84. In particular, HAE (adap) outperformed others with more than three reference pairs ($L > 3$). Although NMSE values (i.e., difference to target values from the ground-truth mapping) were all comparable, HAE (adap) and HAE (fixed) slightly outperformed the basic HAE in most cases. HAE (adap) also tended to converge faster with a smaller number of reference pairs than the other two cases.

In some cases a “folded” or “twisted” mapping was learned, as shown in Figure 6. The phenomenon is not new and has been mentioned in previous literature related to the self-organizing map. Folded and twisted grid structures were found and investigated in Kiviluoto (1996) and Aoki et al. (2007), respectively. It is known that choosing appropriate criteria for determining the neighborhood points is the key to avoiding such defects.

A similar approach for the HAE can be achieved by tuning the hyperparameters. Empirically, we found that dynamically adjusting the hyperparameters was more effective than using constant values for avoiding such extreme defects as folding or twisting. We limited the initial effect of the boundary attraction term during training in order to reduce topological defects in the learned mapping. When the boundary attraction term is active early in the model

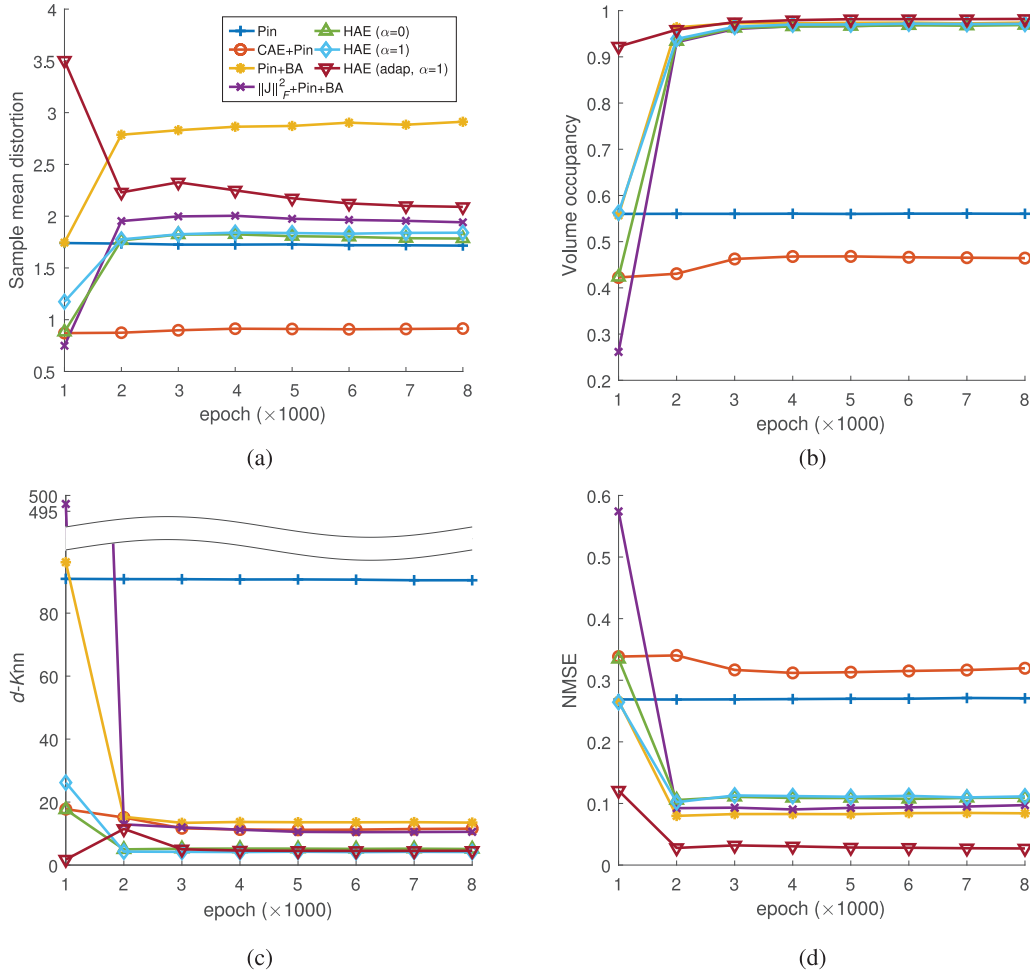


Fig. 2. Mapping evaluation curves ($L=4$) with respect to learning epoch for synthetic data (SP1 = 3D clover to 2D square): (a) sample mean distortion \bar{d} , (b) volume occupancy VO , (c) K -nearest neighbor inverse image dissimilarity d -Knn, and (d) NMSE. Mean values over 10 independent repetitions for each method are shown. For the results of other shape pairs, see Appendix F.

training procedure, the mapping will tend to stretch towards the boundaries quickly, causing subsequent pinnings to distort the mapping and result in twisting. To minimize defects such as twisting, we effectively turned off the boundary attraction constraint (by initially setting λ_3 to zero) until all reference pairs had been matched and stabilized. Then we activated the boundary attraction constraint and linearly increased λ_3 to its full value over a number of epochs (2,000 epochs in our experiments).

In Figure 6, Equation (17) defined for d -Knn is evaluated for each data point (not averaged over $x \in f(Q)$) and shown as a heatmap. The results highlight regions of folding and/or twisting because similar outputs are not necessarily from similar inputs for those cases. One of the important roles of reconstruction (the first term of Equation (2)) in the input domain is to encourage the image of the mapping to contain complete, as possible, information of the original input, so that the mapping is learned close to one-to-one between the two manifolds, which results in lower d -Knn values. This implies that the autoencoder structure of the HAE also reduces the defects in the learned mapping, in addition to enabling dynamic hyperparameter

adjustment. As can be read from Table 2, d -Knn values were reduced significantly from the $\|J\|_F^2 + \text{Pin} + \text{BA}$ model to the HAE ($\alpha=0$) model where the only difference is the autoencoder reconstruction term.

The performances of HAE models with respect to α were mixed, for both HAE and HAE (adap). We interpret that α , a ratio coefficient between Euclidean distances of input and target domains, can be treated as a hyperparameter that can be tuned according to specific domains of the applications. For the purposes of this article, we did not tune this hyperparameter; we compared the models with fixed values of α (either 1 or 0).

3.3. Experiment with human and robot hand pose data

Here, we compare the same seven models (Table 1) evaluated in Section 3.2 using human and robot hand pose data. To make subsequent descriptions clear, we introduce anatomical terms for the human hand, as shown in Figure 7. The bones in each finger starting from the tip are termed distal, middle, and proximal phalanges, respectively. Proximal

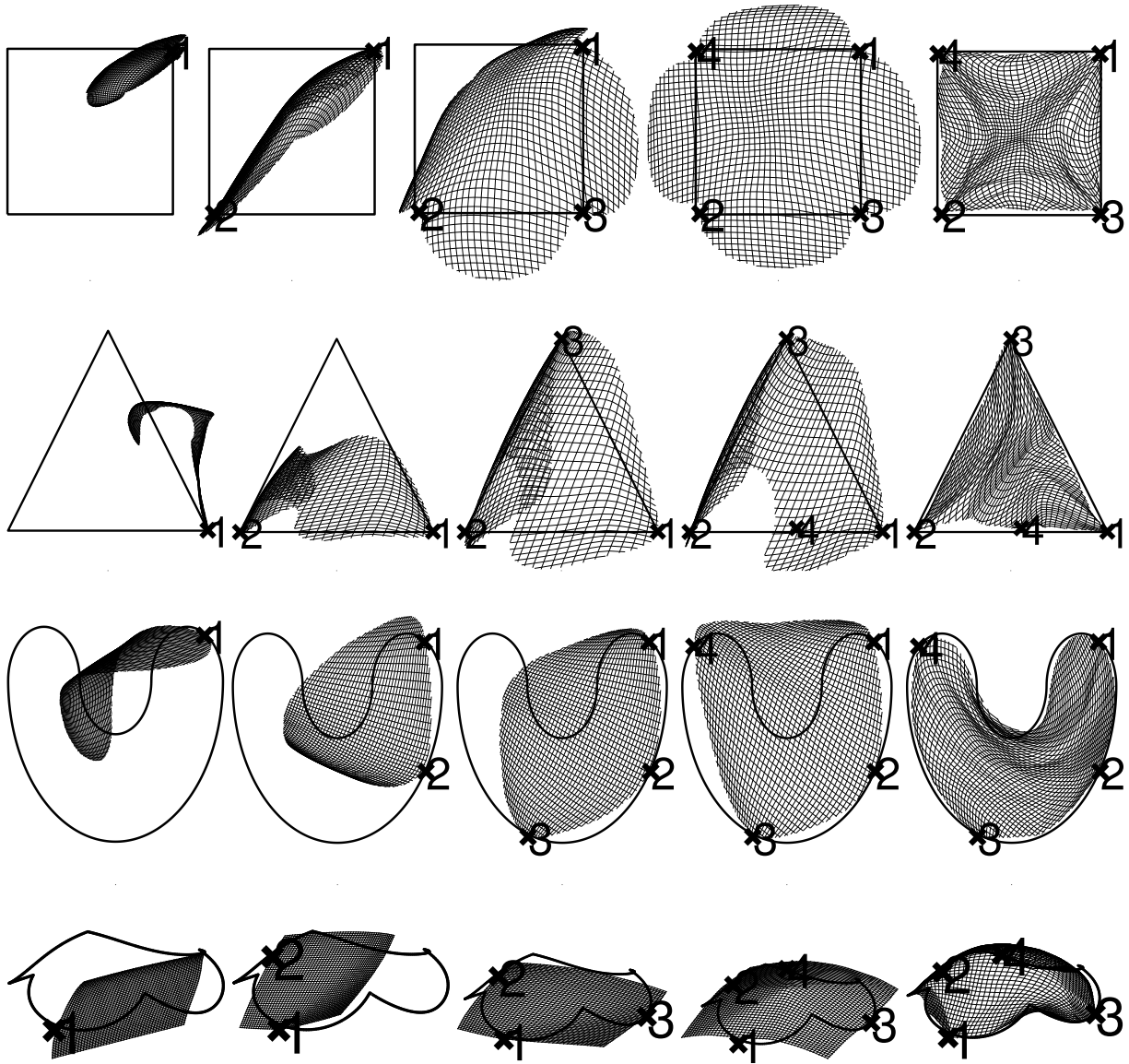


Fig. 3. Mapping results of HAE (adapt, $L=4$) during the learning process with synthetic data (test set). Row 1: SP1 = 3D clover to 2D square, row 2: SP2 = 3D U-shape to 2D triangle, row 3: SP3 = 3D circle to 2D U-shape, row 4: SP4 = 2D square to 3D clover. The five columns, from left to right, correspond to training session 1 to 5, respectively. The numbered points denote the reference points (Pins), which are updated sequentially during the learning process. The coefficient λ_3 in Equation (13) is set to zero until the 1,000th epoch of session 5, after which λ_3 is linearly increased to a value of one over the course of 2,000 epochs.

phalanges are connected to metacarpal bones, which are connected to the wrist bones. It has been reported that a human hand can be modeled with 21 DOFs, excluding wrist motion Lin et al. (2000). The joints of each finger starting from the tip are named distal interphalangeal (DIP), proximal interphalangeal (PIP), and metacarpophalangeal (MCP) joints, respectively. For the DIP and PIP joints, 1-DOF hinges are typically assumed. The MCP is modeled as a 2-DOF universal joint. Exceptionally, the thumb has no middle phalanx and has different joint DOFs; Hollister et al. (1992) suggested five non-intersecting, non-orthogonal DOFs for the thumb.

3.3.1. Hardware setup. Encoding human hand pose. A custom sensor glove was used for collecting human hand pose data, as shown in Figure 8(a). A total of 16 inertial measurement unit (IMU) sensors (BNO055, Bosch Sensortec, Germany) were mounted on a fabric-based glove. One IMU was placed on the dorsal aspect of the hand. For the thumb, one IMU was placed over each of the phalanges and the first metacarpal. For the remaining fingers, one IMU was placed over each of the phalanges. Each 9-DOF IMU is composed of a 3D accelerometer, 3D gyroscope, and 3D magnetometer.

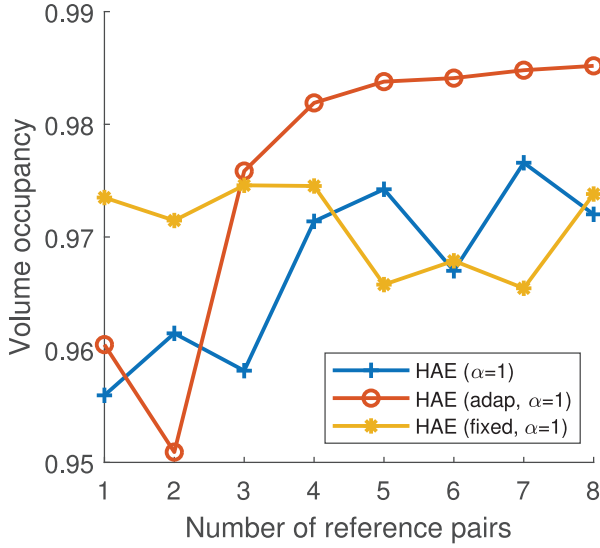


Fig. 4. Volume occupancy VO with respect to number of reference pairs for synthetic data (SP1 = 3D clover to 2D square). Results for the remaining shape pairs are shown in Appendix F.

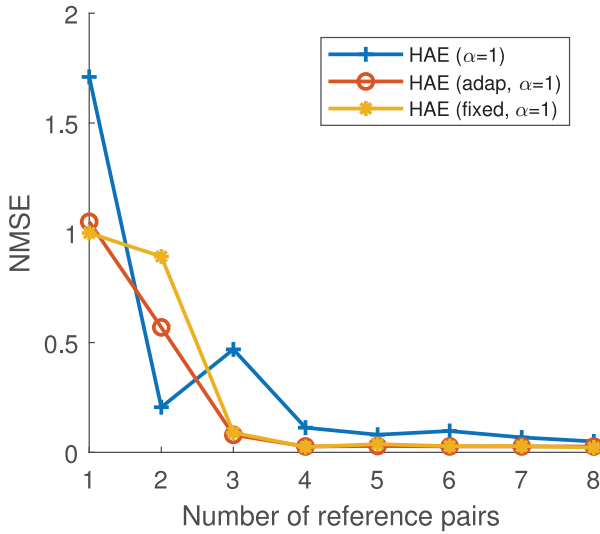


Fig. 5. NMSE with respect to number of reference pairs for synthetic data (SP1 = 3D clover to 2D square). Results for the remaining shape pairs are shown in Appendix F.

Robotic gripper. We used the Robotiq 3-Finger Adaptive Gripper (Robotiq Inc., QC, Canada), which is a robot gripper designed for industrial applications as shown in Figure 8(b). Each of the three articulated fingers has three joints and is underactuated, which allows the gripper to mechanically conform to the object being grasped. As a result, the configuration of the hand cannot be directly computed because there is only one position reading at the base of each finger. There are a total of four DOFs: open-close for each of the three fingers, and a lateral, scissor-like motion between two fingers.

We surmised that mapping poses from a human hand to a 3-finger robot hand (4 DOFs) would be the most

interesting among popular gripper configurations. First, mapping to a simple 1-DOF gripper is relatively trivial. Second, mapping to a 5-digit robot hand can be made intuitive by using a finger-to-finger mapping. For our demonstration, we purposely selected a non-anthropomorphic 3-finger robot hand, which is standard in industry and common in research labs. The additional lateral DOF of the Robotiq 3-finger gripper introduces additional complexity to our task by allowing the mapping of human hand abduction/adduction capabilities.

3.3.2. Data collection protocol and specification. The placement of the IMU sensors and the collected joint DOFs are shown in Figure 8(a). During data collection via the sensor glove, there were additional DOFs owing to sliding between the glove fabric and the human hand, especially at the MCP joint of each finger. Thus, we assumed one extra DOF each for the CMC joint of the thumb and the MCP joint of each remaining finger, resulting in a total of 25 DOFs for the entire sensor glove system.

Human hand pose data were collected via the sensor glove by a human operator performing continuous movements: making a fist, flattening the palm, flexing and extending each finger individually as well as multiple fingers simultaneously, and gathering the fingertips together. Data were collected at a sampling rate of 9.2 Hz, resulting in 381 samples for the training set \tilde{Q}_{train} , and 254 samples for the test set \tilde{Q}_{test} .

Random uniform samples from the robot gripper \tilde{X} were collected according to the configuration space of the specific robot gripper structure shown in Figure 8(b). We sampled 8,000 data points \tilde{X} for training the HAE. The maximum gripper opening is 155 mm; the maximum lateral separation for the scissor-like motion is 32° (Robotiq Inc., 2016).

The robot gripper pose can be represented with four state values: $x = [x_A, x_B, x_C, x_S]$. The values x_A , x_B , and x_C correspond to the open-close state of fingers A, B, and C, respectively, and the value x_S denotes the state for lateral motion between fingers B and C. The configuration space CS can be described as $CS = CS_0 \cap CS_1$, where $CS_0 = \{x | x_k \in [0, 255] \ \forall \ k \in \{A, B, C, S\}\}$. That is, all four states range from 0 to 255 real values. In addition, $CS_1 = \{x | x_S < \beta_{\text{pinch}} \text{ OR } \max(x_A + x_B, x_A + x_C) < \gamma_{\text{pinch}}\}$, such that the open-close motions of the three fingers are constrained when their configuration is in a “pinch grasp” mode (i.e., $x_S \geq \beta_{\text{pinch}}$). The following thresholds were set: $\beta_{\text{pinch}} = 180$ and $\gamma_{\text{pinch}} = 290$.

Here, we aim to cover the entire configuration space of the robot gripper so that CS is the target domain of the mapping. However, the user can elect to define the target domain in order to take into account additional constraints, e.g., obstacles or self-collision.

3.3.3. Model learning configuration. We set the coefficients $(\lambda_1, \lambda_2, \lambda_3)$ as $(0.005, 2.0, 5.0)$ in Equations (2) and

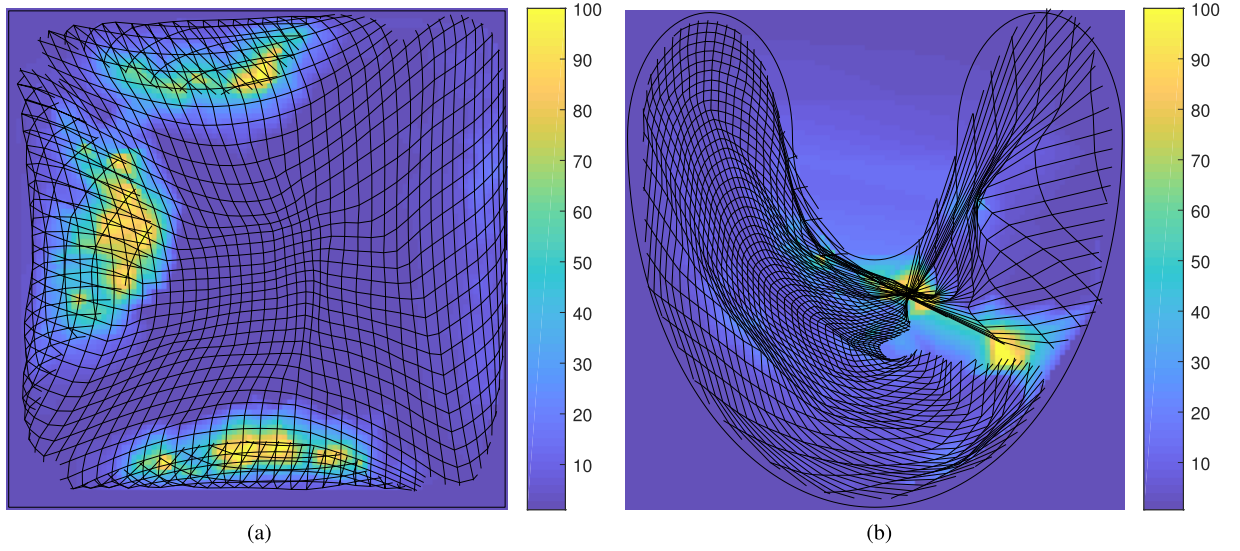


Fig. 6. The K -nearest neighbor inverse image dissimilarity measure d -Knn of Equation (17) is evaluated in an element-wise way (i.e., not averaged over $x \in f(Q)$) over target domain and shown as a heatmap; values are capped at 100 to show clearer heatmap distributions. (a) A folding example for the SP1 3D clover to 2D square case using $\|J\|_F^2 + \text{Pin} + \text{BA}$ ($L=4$) and (b) a twisting example for the SP3 3D circle to 2D U-shape case using HAE ($\alpha=0$, $L=4$). The measure successfully captures regions of folding or twisting where similar outputs are not necessarily from similar inputs.

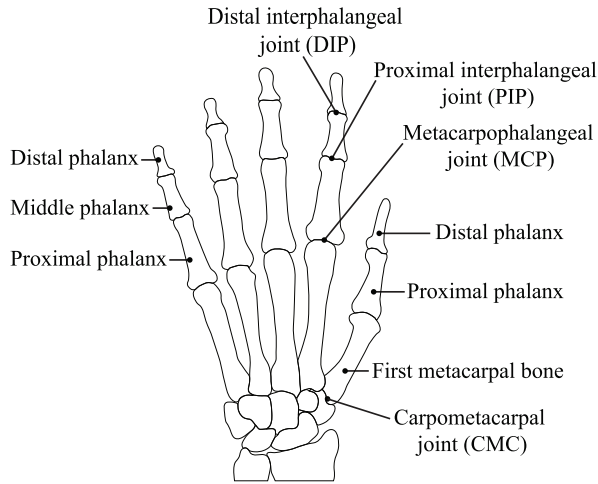


Fig. 7. A schematic of bones and joints of the human (left) hand.

(13), and set 300 dimensions for both of the hidden variables h_1 and h_2 in Equations (7) and (8). For the rest of the experimental settings, we used the same values as the synthetic data experiment including the number of hidden variables and the computational environment. The training procedure took 188 seconds for the HAE (adap) model (with $L=8$), including time for the operator to provide human hand pose demonstrations during the training by using intuition to mimic the robot gripper poses, which are automatically selected by the adaptive model.

3.3.4. Results and discussion. As with the synthetic data experiment in Section 3.2, we compare and discuss the seven neural network structures shown in Table 1.

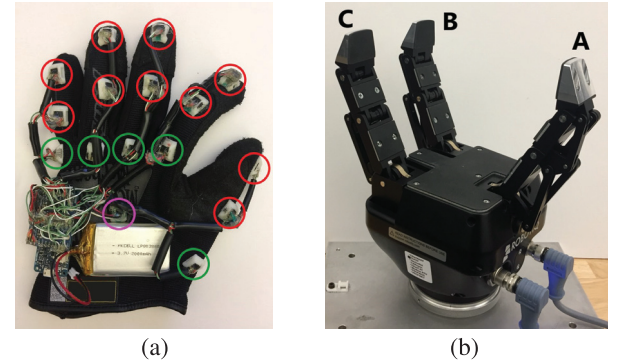


Fig. 8. (a) Custom sensor glove equipped with 16 IMU sensors. Red and green circles mark the sensors used to estimate joint angles for presumed 1-DOF and 3-DOF joints, respectively. The sensor on the dorsum of the hand, circled in magenta, was used as the base frame for all digits. (b) 4-DOF underactuated robot gripper (Robotiq 3-Finger Adaptive Gripper). Each finger (A, B, C) has one open-close DOF. An additional lateral, scissor-like DOF exists between fingers B and C.

Reference point pairs S for all of the models except for the HAE (adap) model were selected heuristically from 10 poses as listed in Table 3.

The first L poses among the 10 poses listed were chosen as L reference pairs for training each model. The three performance measures \bar{d} , VO , and d -Knn were evaluated for each model. Mapping results are reported in Table 4 for $L=8$. Each model was trained and evaluated for 10 independent repetitions.

As with the synthetic data experiment, the CAE + Pin achieved the best performance with respect to \bar{d} , but failed

Table 3. Robot gripper poses that were selected heuristically and used to provide target points x of reference pairs to all models except for the HAE (adap).

Index	Description	$x = [x_A, x_B, x_C, x_S]$
i	pinch close	125, 125, 125, 255
ii	pinch open	0, 0, 0, 255
iii	finger A open	0, 255, 255, 130
iv	finger B open	255, 0, 255, 130
v	finger C open	255, 255, 0, 130
vi	finger A, B open	0, 0, 255, 130
vii	finger B, C open	255, 0, 0, 130
viii	finger C, A open	0, 255, 0, 130
ix	all fingers close	255, 255, 255, 130
x	all fingers open	0, 0, 0, 0

Table 4. Model performance comparison for human-to-robot hand pose data (test set) with eight reference pairs ($L = 8$). Seven different models are compared with respect to sample mean distortion, target domain volume occupancy, and K -nearest neighbor inverse image dissimilarity as defined in Section 3.1. Bold text indicates the best performance for the given measure.

Cost functions		Measures	
($L = 8$)	\bar{d}	VO	$d-Knn$
(Baselines)			
Pin	3.606 (0.174)	0.468 (0.024)	11.994 (1.323)
CAE + Pin	0.971 (0.030)	0.417 (0.008)	18.395 (0.437)

Pin + BA	13.412 (0.744)	0.936 (0.011)	19.517 (3.382)
$\ J\ _F^2 + \text{Pin} + \text{BA}$	6.116 (0.340)	0.936 (0.008)	15.638 (1.667)
HAE ($\alpha=0$)	5.672 (0.187)	0.934 (0.005)	10.814 (1.012)
HAE ($\alpha=1$)	5.980 (0.139)	0.931 (0.010)	9.948 (0.817)
HAE (adap, $\alpha=1$)	5.950 (0.402)	0.936 (0.006)	11.031 (0.936)

to cover the target domain resulting in low values for VO . Among the methods with comparable volume occupancy performance ($VO > 0.93$), the HAE models outperformed others in terms of \bar{d} and $d-Knn$, but the performance within HAE models were mixed. HAE ($\alpha = 0$) yielded the lowest distortion, HAE (adap, $\alpha = 1$) yielded the largest volume occupancy, and HAE ($\alpha = 1$) yielded the smallest $d-Knn$ value.

In fact, we found that the robot poses automatically selected by HAE (adap) closely resembled the poses selected heuristically, as shown in Figure 9. This likely explains the comparable performances of HAE ($\alpha = 1$) and HAE (adap, $\alpha = 1$).

In addition, similar to the synthetic data experimental results, the effect of α on model performance was found to be marginal in our human-to-robot hand pose mapping

experiment. The two non-adaptive HAE models ($\alpha = 0$ and $\alpha = 1$) performed similarly when varying the number of reference pairs between 1 and 10 for the three performance measures (Figure 10).

An error measure such as NMSE used in Section 3.2 may be useful to evaluate how the model output is different from the ground-truth target data. In this experiment, such an evaluation of error is not conducted because a ground-truth dataset was not available. Instead, we visually show that our model is capable of mapping a continuously performed human hand motion to corresponding robot poses in a natural way and in real time (Figure 12). Next, we describe the trained model and a visualization of the mapping in more detail.

The reference poses selected by HAE (adap, $\alpha = 1$, $L = 8$) during the on-line training phase are shown in Figure 9. The robot gripper poses were selected by the model autonomously and sequentially, according to Equations (11) and (12). Corresponding human hand poses were determined and performed by the operator in real time, then both the robot and human hand poses were fed into the model to update the set of reference pairs and continue the training. This training phase is demonstrated in Extension 1.

The selected robot and human hand poses are shown within input and target domains, respectively, represented in PCA-transformed 3D spaces as shown in Figure 13. Three principal axes from 25-dimensional human hand pose data and 4 dimensional robot gripper pose data are each represented. We observe that the adaptively selected reference points are primarily in the bounding areas of the sample space. Based on the PCA representation shown in Figure 13(b), the image of test samples $f(\tilde{Q}_{\text{test}})$, is well distributed within \tilde{X} samples. This implies that most configurations of the robot gripper could be reached by the learned mapping.

We demonstrate a continuous representative hand motion to visually assess mapping performance of our model from human to robot hand poses in Figure 12. The human motion input was collected at 20 Hz from the human hand via the sensor glove. The mapped robot hand pose output was estimated at 50 kHz. The fast estimation time of the model is promising for real-time applications. Snapshots of human and robot hand poses are shown for every 500 ms, denoted as q_0-q_{24} for human hand poses and x_0-x_{24} for corresponding robot poses, within a 6-second time period. The motion is also plotted in the PCA-represented 3D space in Figure 13. A real-time human to robot hand pose mapping is demonstrated in Extension 1. For clarity, the demonstration is performed at a slower speed (within a 14-second time period) than would otherwise be necessary in practice.

While the trajectory of human-demonstrated motion did not necessarily pass through training samples or reference points, our learned model was capable of naturally mapping human hand poses to robot poses (Figure 12), which implies that the mapping was successfully defined over

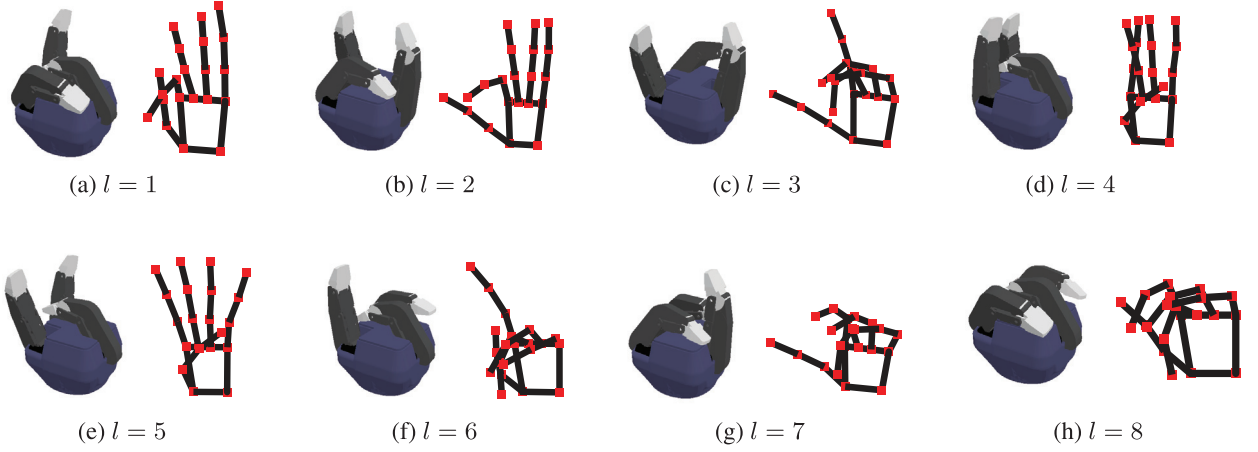


Fig. 9. Illustration of eight reference pairs automatically selected from HAE (adap, $\alpha = 1$) model ($L = 8$). The robot gripper poses appear similar to poses selected heuristically by the human operator: (a) finger B open, (b) finger A, B open, (c) finger A, C open, (d) finger B, C narrow open, (e) finger B, C wide open, (f) finger C open, (g) finger A open, and (h) all fingers closed. Corresponding human hand poses were selected for a left hand as shown alongside each robot gripper pose and were fed into the HAE model during the adaptive learning process.

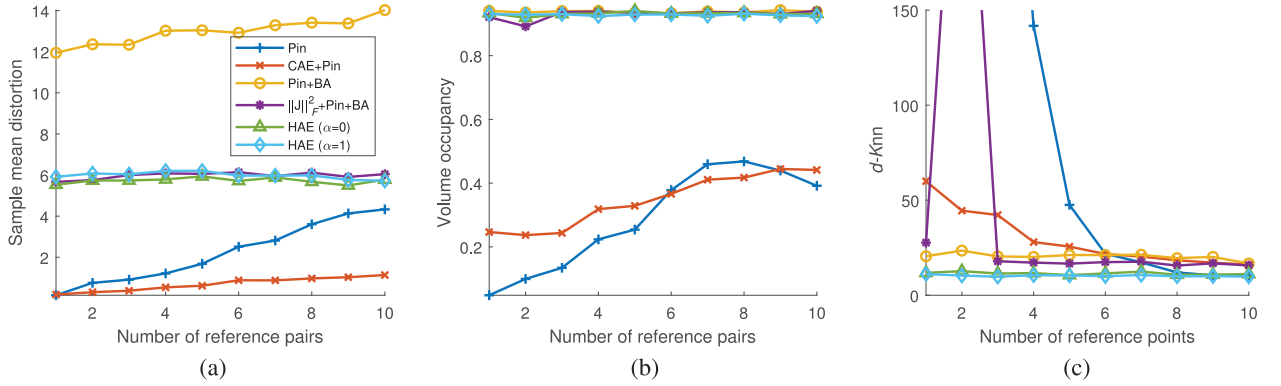


Fig. 10. Model performance curves with respect to number of reference pairs for human-to-robot hand pose data: (a) sample mean distortion \bar{d} , (b) volume occupancy VO , and (c) K -nearest neighbor inverse image dissimilarity $d\text{-Knn}$. In (c), $d\text{-Knn}$ values over 150 are omitted from the plot for clarity; the omitted datapoints are reported here as (label, x -value, y -value)-tuples: (Pin, 2, 1.20×10^3), (Pin, 3, 371.40), and ($\|J\|_F^2 + \text{Pin} + \text{BA}$, 2, 314.69).

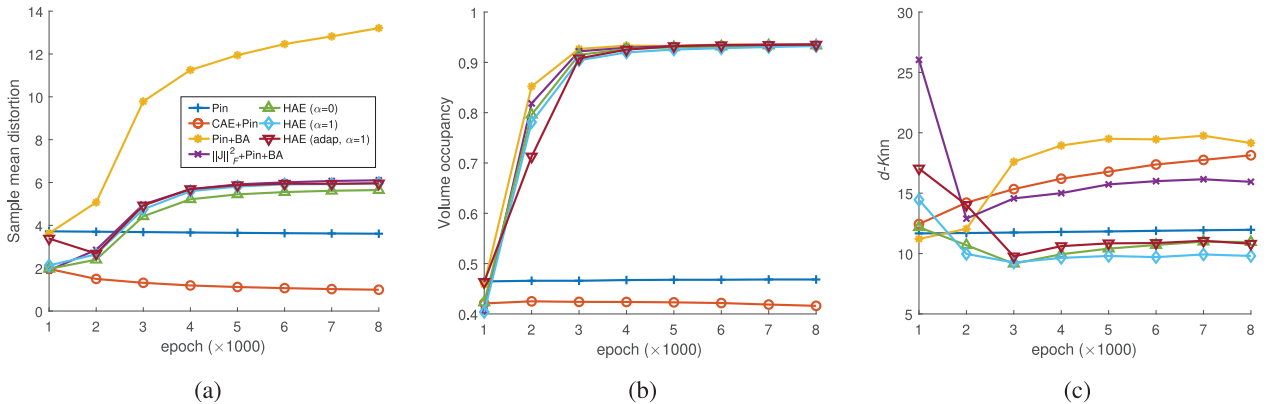


Fig. 11. Model performance curves with respect to learning epoch for human-to-robot hand pose data: (a) sample mean distortion \bar{d} , (b) volume occupancy VO , and (c) K -nearest neighbor inverse image dissimilarity $d\text{-Knn}$.

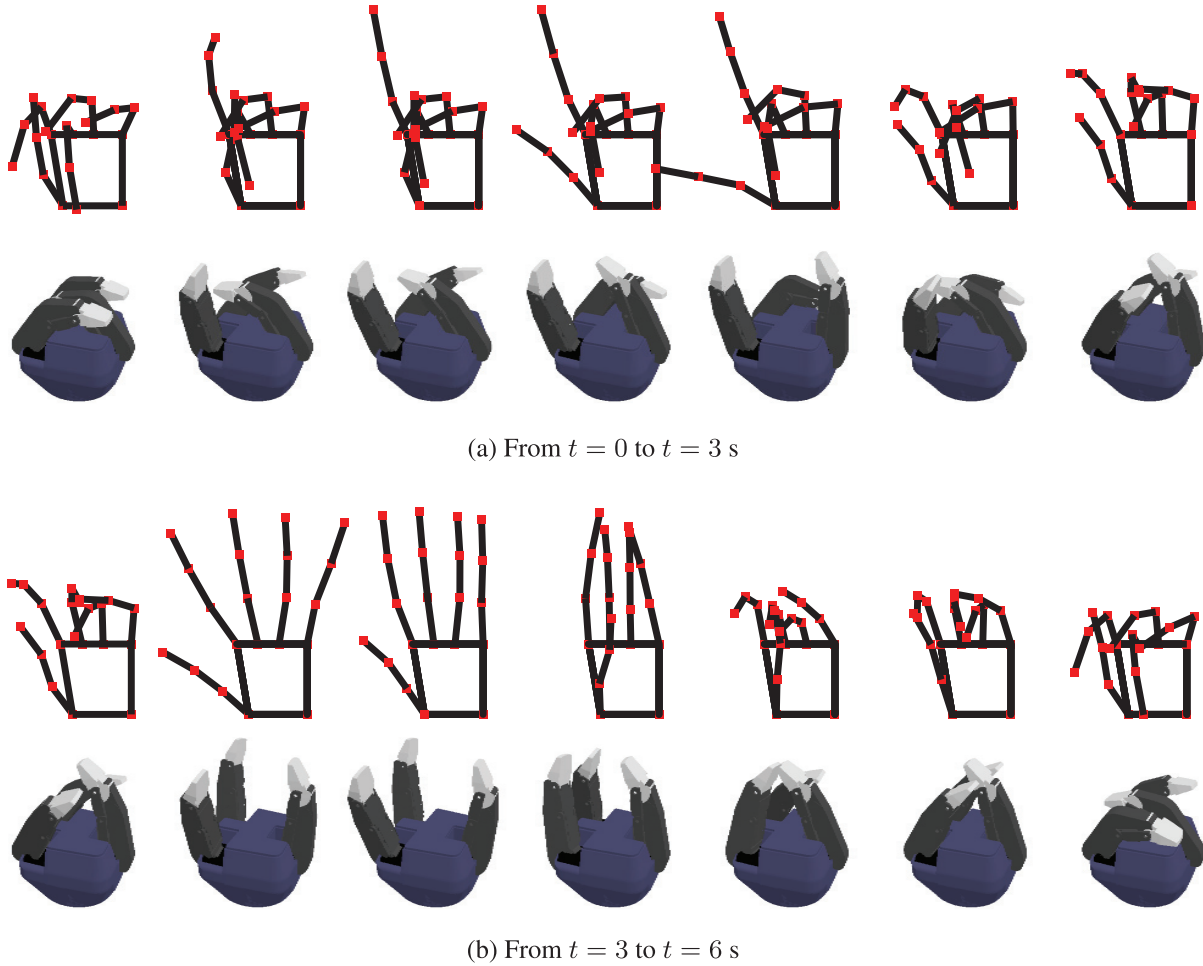


Fig. 12. A continuous 6-second human hand motion trajectory and its corresponding robot gripper motion trajectory mapped from learned HAE (adap, $\alpha = 1$) model ($L = 8$) are illustrated. Snapshots of the human and robot hand poses are shown for every 500 ms. Each subfigure shows human hand poses in the upper row and robot gripper poses in the lower row. (a) Upper row: $[q_0, q_2, \dots, q_{12}]$; lower row: $[x_0, x_2, \dots, x_{12}]$. (b) Upper row: $[q_{12}, q_{14}, \dots, q_{24}]$; lower row: $[x_{12}, x_{14}, \dots, x_{24}]$.

input human hand pose and target robot gripper pose domains.

3.4. Assessment of HAE and comparison with related works

Our HAE approach is based on the well-established method of harmonic mapping distortion from Park and Brockett (1994) and Rifai et al. (2011). Aside from harmonic mapping, traditional mapping approaches between two different domains include the self-organizing map (Kohonen, 1990), the generative topographic mapping (Bishop et al., 1998), and the autoencoder. However, in most cases, the aim is to obtain a latent variable representation for visualization, dimensionality reduction, or feature extraction. We could not identify in the literature an established benchmark method for mapping between two physical spaces for a direct quantitative comparison with HAE.

In order to extend the mapping problem to two physical spaces, we modeled functions f and g using neural

networks and defined two constraint terms, Pin and BA. We chose the neural networks model for its flexibility as a global approximator and to take advantage of existing optimization techniques and resources developed for neural networks. We easily modified the cost function with two novel constraints that enabled our model to learn a distortion-minimizing mapping that satisfies given reference pairs and covers the target domain.

Instead of neural networks, one might consider using a different model structure such as the Gaussian process model, which is a powerful approximation tool when only a small number of data points are available. One can also derive the distortion, Pin, and BA terms using the mean estimation function. A further study on the relationship between different model structures and the characteristics of a learned mapping would be an interesting topic for future work.

Experimental results in Sections 3.2 and 3.3 clearly show how each term of the HAE cost function contributes to the metrics of performance. For all methods, a Pin term

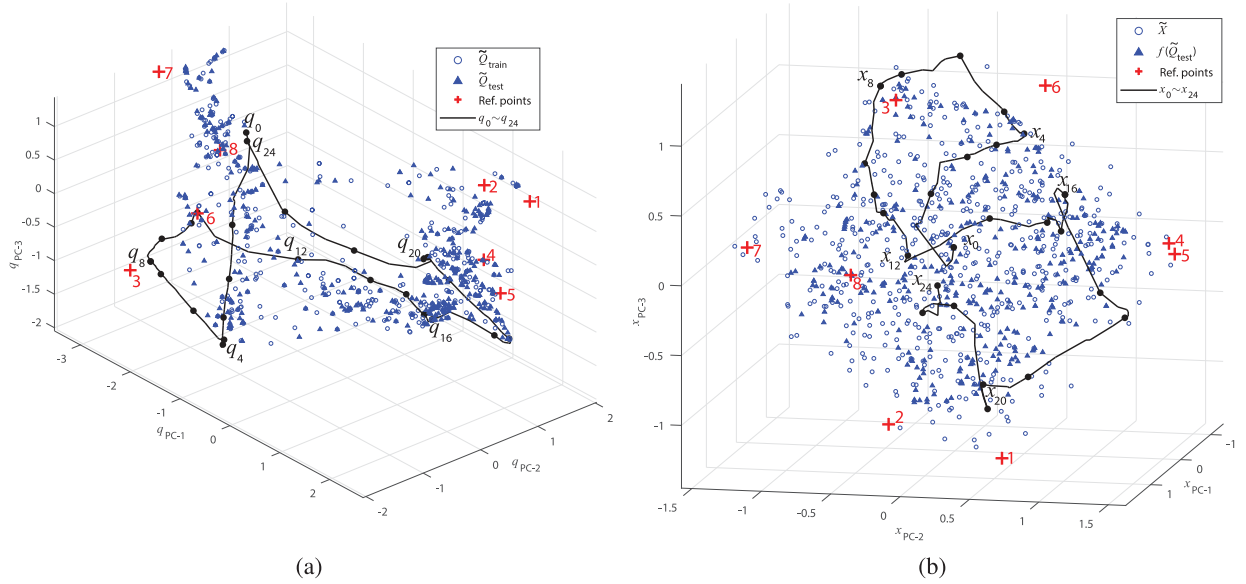


Fig. 13. Visualization of input and target domains for human and robot hand pose data with learned results from HAE (adap, $\alpha = 1$) model ($L = 8$). Three principal axes from the 25-dimensional human hand pose data and 4-dimensional robot gripper pose data were extracted using PCA for representation in 3D space. (a) Human hand pose in the 3D space: samples of the training set \tilde{Q}_{train} and test set are shown as blue marks. (b) Robot gripper pose in the 3D space: robot hand pose samples \tilde{X} and mapping outputs from the human hand pose test set $f(\tilde{Q}_{\text{test}})$ are shown. In both (a) and (b), red crosses denote reference points autonomously selected by the adaptive model. A continuous sample trajectory passing through human hand poses from q_0 to q_{24} and its image trajectory passing through robot gripper poses from x_0 to x_{24} are shown as black solid lines in (a) and (b), respectively.

was necessary in order to constrain input–target reference pairs. The BA term was necessary for covering the target domain. The baseline methods (Pin, CAE + Pin) that did not use the BA term had smaller target domain volume occupancy values VO (Tables 2 and 4).

Comparing $\|J\|_{\mathcal{F}}^2 + \text{Pin} + \text{BA}$ and HAE ($\alpha = 0$), the inclusion of the HAE autoencoding constraint (AE in Table 1) offers a distinct advantage in terms of \bar{d} and $d\text{-Knn}$. This suggests that AE provides additional useful regularization while reconstructing the input data. Furthermore, a comparison between HAE ($\alpha = 0$) and HAE ($\alpha = 1$) shows that adding Pin and BA terms to the reconstruction function g enhances overall mapping quality; mapping performance improved in terms of $d\text{-Knn}$ with comparable VO and NMSE. Although AE is commonly used for obtaining latent representations for dimensionality reduction or feature extraction, our results suggest that the AE structure is also useful for learning a physically meaningful representation.

Recall that we purposely consider applications in which it is costly to obtain a large number of reference data pairs. As a result, the Pin term acts only on a small number of input–target pairs. With the addition of a distortion term $\|J\|_{\mathcal{F}}^2$ in the cost function, a learned mapping will contract as much as possible while being stretched to the pinned reference pairs. With the addition of BA and AE terms, the mapping can leverage additional information using independent samples from input and target domains. As more reference pairs become available, we expect that the

benefits of incorporating the BA and AE terms will diminish until the benefits are outweighed by the computational expense for training the two terms. If a sufficient number of reference pairs are available, standard regression approaches can be used; this scenario is not of interest in this study. The addition of the BA and AE terms to Equation (13) also introduces additional hyperparameters: λ_2 , λ_3 , and α (whereas λ_1 is from the original CAE model). Note that we use the same set of hyperparameters for all experiments while only changing λ_2 for the Pin term (1.0 or 2.0), thereby avoiding excessive offline tuning effort. We are able to minimize hyperparameter tuning efforts while still achieving good mapping performance.

There are a number of benefits of the HAE method. First, the reference pairs required for the training phase of the mapping can be performed by a non-expert operator based on the operator’s intuition. Knowledge of how to achieve a joint-to-joint mapping from the input to the target domain for a specific task or robot platform is not necessary. Furthermore, the training inputs can be customized to the capabilities of the operator, which would be especially useful for assistive robotics applications. Second, the HAE architecture is not restricted to a specific kinematic structure such as a robotic gripper, but can be generally applied to any of a number of high-DOF systems, e.g., a swarm of robots. Lastly, the HAE uses physically meaningful cost terms, resulting in an interpretable, straightforward implementation.

4. Concluding remarks

In this article, we have proposed a novel method, the HAE, for learning a harmonic mapping by using a set of sample points from an input domain, target domain, and a small number of reference input–target data pairs. Our model extends an existing generative neural network, the CAE, as a building block to minimize distortion over the input samples, while covering the target domain and satisfying the reference data pairs. We have provided detailed and reproducible mathematical explanations on how the CAE can be utilized to implement a learning-based, data-driven harmonic mapping. To achieve a desired harmonic mapping, a distance measure between point clouds is taken into account within both the input and target domains (boundary attraction), and enacts a penalizing estimation error on the input–target reference pairs (pinning).

Although identifying reference pairs (including boundary examples) can be expensive, it is a strength that our approach works with a small number of reference pairs by incorporating the BA term. With a small number of reference pairs, a map trained using a traditional autoencoder framework (e.g., a baseline CAE) will not be able to efficiently cover the target domain. In contrast, the BA term can be calculated from independent samples from each input/output domain with unmatched sample sizes and without pairing information.

A set of input–target reference data pairs could be selected automatically in an effective way by using an adaptive optimization criterion while training the HAE (adap). We have demonstrated that pairs selected adaptively yield a higher-performance mapping than pairs selected randomly, and the mapping result is comparable to that from pairs selected heuristically by the experimenter using intuition.

Our experimental results with synthetic data and human-to-robot hand pose data have shown that, with several input–target reference pairs, our method can learn a mapping that successfully covers the target domain and inherits the beneficial property of minimal distortion from harmonic mapping. By applying our model to human-to-robot hand pose data with eight pose demonstrations, a natural and continuous robotic hand motion trajectory was mapped from human hand motion in real time.

Our method can be applied directly to applications of gesturing and pre-shaping of grippers for grasp. For gripper contact with an object or the environment, however, additional modifications would be necessary. For instance, one could apply constraints to the gripper configuration space or impose costs based on the degree of intrusion of the gripper through the object. Another approach, as implemented in our hardware demonstration with the Robotiq 3-Finger Adaptive Gripper, is to leverage an internal controller that automatically limits grip forces. Regardless, our method could be modularized to output mappings that are refined prior to control of the robot hardware. At this time, accounting for contact dynamics is beyond the scope of this work.

Our analysis on variations of the HAE has highlighted the purpose and importance of each term added to the CAE.

We have discussed our observations on map folding and twisting, which can be detected by one of our proposed performance metrics, d -Knn. The HAE models have produced low d -Knn values, which implies that the reconstruction and boundary attraction cost terms calculated over the input domain could help avoid these extreme cases. We expect that exhaustive tuning of the hyperparameters (e.g., the number of hidden variables, learning rate, optimization coefficients) may lead to better performance than that illustrated in this work. The tuning task requires additional computer resources, and is beyond the scope of our current research.

To the best of the authors' knowledge, this is the first approach to implement a learning-based harmonic mapping in a data-driven way. Our proposed method enables more versatile and higher-dimensional applications by breaking through limitations of existing harmonic mapping and self-organizing map approaches. Whereas most existing harmonic mapping and self-organizing map applications are limited by computational cost to numerically mapping images between 2D and/or 3D spaces, the HAE approximates harmonic mapping in a data-driven way so that the mapping can be applied to high-dimensional applications, such as human-to-robot hand pose mapping. The data-driven approach allows our method to take advantage of a neural network's low computational complexity for estimation, as well as its ability to learn useful representations from high-dimensional raw data.

As our method does not rely on a specific kinematic structure, we expect that the HAE can be generalized to various robotic structures or domains. We have shown that the proposed method enables a continuous mapping of human-to-robot hand poses with only a few reference pairs and in real time. We believe that our study opens new pathways for future research directions beyond the human-to-robot hand pose mapping application.


Acknowledgements

The authors thank Ja Yeun Heo for assistance with figure illustrations.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported in part by the National Science Foundation (award number 1463960), Office of Naval Research (award numbers N00014-16-1-2468 and N00014-18-1-2814), and Advanced Robotics for Manufacturing Institute (award number ARM-TEC-18-01-F-14).

ORCID iD

Eunsuk Chong  <https://orcid.org/0000-0002-4629-0918>

Note

1. For the synthetic data, we could obtain effective mapping results with $L \geq 4$ in terms of volume occupancy and NMSE (see Figures 4 and 5).

References

- Aoki T, Ota K, Kurata K and Aoyagi T (2007) Ordering process of self-organizing maps improved by asymmetric neighborhood function. In: *International Conference on Neural Information Processing*. New York: Springer, pp. 426–435.
- Athitsos V and Sclaroff S (2003) Estimating 3D hand pose from a cluttered image. In: *Proceedings 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2. IEEE, pp. II–432.
- Barhak J and Fischer A (2001) Adaptive reconstruction of free-form objects with 3D SOM neural network grids. In: *Proceedings Ninth Pacific Conference on Computer Graphics and Applications*, 2001. IEEE, pp. 97–105.
- Bauer HU and Pawelzik KR (1992) Quantifying the neighborhood preservation of self-organizing feature maps. *IEEE Transactions on Neural Networks* 3(4): 570–579.
- Billard A, Calinon S, Dillmann R and Schaal S (2008) *Robot Programming by Demonstration (Handbook of Robotics*, Vol. 59). Berlin: Springer, pp. 1371–1394.
- Bishop CM, Svensén M and Williams CK (1998) GTM: The generative topographic mapping. *Neural Computation* 10(1): 215–234.
- Calinon S (2018) Learning from demonstration (programming by demonstration). In: Ang M, Khatib O and Siciliano B (eds) *Encyclopedia of Robotics*. Berlin: Springer. DOI: 10.1007/978-3-642-41610-1_27-1
- Cayton L (2005) Algorithms for manifold learning. *Technical Report*, University of California at San Diego. Available at: <https://www.lcayton.com/resexam.pdf> (accessed 19 September 2020).
- Chen J and Zelinsky A (2003) Programing by demonstration: Coping with suboptimal teaching actions. *The International Journal of Robotics Research* 22(5): 299–319.
- Choi PT, Lam KC and Lui LM (2015) FLASH: Fast landmark aligned spherical harmonic parameterization for genus-0 closed brain surfaces. *SIAM Journal on Imaging Sciences* 8(1): 67–94.
- Chong E and Park FC (2017) Movement prediction for a lower limb exoskeleton using a conditional restricted Boltzmann machine. *Robotica* 35(11): 2177–2200.
- Ciocarlie MT and Allen PK (2009) Hand posture subspaces for dexterous robotic grasping. *The International Journal of Robotics Research* 28(7): 851–867.
- Cutkosky MR and Howe RD (1990) Human grasp choice and robotic grasp analysis. In: *Dextrous Robot Hands*. Berlin: Springer, pp. 5–31.
- Dillmann R, Rogalla O, Ehrenmann M, Zöliner R and Bordegoni M (2000) Learning robot behaviour and skills based on human demonstration and advice: The machine learning paradigm. In: *Robotics Research*. Berlin: Springer, pp. 229–238.
- Diodato A, Brancadoro M, De Rossi G, et al. (2018) Soft robotic manipulator for improving dexterity in minimally invasive surgery. *Surgical Innovation* 25(1): 69–76.
- Dozat T (2016) Incorporating Nesterov momentum into Adam. In: *Proceedings of 4th International Conference on Learning Representations, Workshop*, pp. 1–4.
- Eells J and Sampson JH (1964) Harmonic mappings of Riemannian manifolds. *American Journal of Mathematics* 86(1): 109–160.
- Ekval S and Kragic D (2004) Interactive grasp learning based on human demonstration. In: *Proceedings 2004 IEEE International Conference on Robotics and Automation (ICRA'04)*, volume 4. IEEE, pp. 3519–3524.
- Erwin E, Obermayer K and Schulten K (1992) Self-organizing maps: Ordering, convergence properties and energy functions. *Biological Cybernetics* 67(1): 47–55.
- Falco J, Marvel J and Messina E (2013) Dexterous manipulation for manufacturing applications workshop. In: *NISTIR 7940*.
- Flexer A (1997) Limitations of self-organizing maps for vector quantization and multidimensional scaling. In: *Advances in Neural Information Processing Systems*, pp. 445–451.
- Gioioso G, Salvietti G, Malvezzi M and Prattichizzo D (2013) Mapping synergies from human to robotic hands with dissimilar kinematics: An approach in the object domain. *IEEE Transactions on Robotics* 29(4): 825–837.
- Giorelli M, Renda F, Calisti M, Arienti A, Ferri G and Laschi C (2015) Neural network and Jacobian method for solving the inverse statics of a cable-driven soft arm with nonconstant curvature. *IEEE Transactions on Robotics* 31(4): 823–834.
- Gorricha JM and Lobo VJ (2011) On the use of three-dimensional self-organizing maps for visualizing clusters in georeferenced data. In: *Information Fusion and Geographic Information Systems*. Berlin: Springer, pp. 61–75.
- Griffin WB, Findley RP, Turner ML and Cutkosky MR (2000) Calibration and mapping of a human hand for dexterous telemanipulation. In: *ASME IMECE 2000 Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems*, pp. 1–8.
- He W, Chen Y and Yin Z (2016) Adaptive neural network control of an uncertain robot with full-state constraints. *IEEE Transactions on Cybernetics* 46(3): 620–629.
- Hollister A, Buford WL, Myers LM, Giurintano DJ and Novick A (1992) The axes of rotation of the thumb carpometacarpal joint. *Journal of Orthopaedic Research* 10(3): 454–460.
- Hu H, Gao X, Li J, Wang J and Liu H (2004) Calibrating human hand for teleoperating the hit/dlr hand. In: *Proceedings 2004 IEEE International Conference on Robotics and Automation, 2004 (ICRA'04)*, Vol. 5. IEEE, pp. 4571–4576.
- Iberall T (1997) Human prehension and dexterous robot hands. *The International Journal of Robotics Research* 16(3): 285–299.
- Joshi AA, Shattuck DW, Thompson PM and Leahy RM (2007) Surface-constrained volumetric brain registration using harmonic mappings. *IEEE Transactions on Medical Imaging* 26(12): 1657–1669.
- Kang SB and Ikeuchi K (1995) Robot task programming by human demonstration: Mapping human grasps to manipulator grasps. In: *Intelligent Robots and Systems*. Amsterdam: Elsevier, pp. 119–136.
- Kang SB and Ikeuchi K (1997) Toward automatic robot instruction from perception-mapping human grasps to manipulator grasps. *IEEE Transactions on Robotics and Automation* 13(1): 81–95.
- Kiviluoto K (1996) Topology preservation in self-organizing maps. In: *IEEE International Conference on Neural Networks, 1996*, Vol. 1. IEEE, pp. 294–299.
- Kohonen T (1990) The self-organizing map. *Proceedings of the IEEE* 78(9): 1464–1480.
- Kohonen T (2013) Essentials of the self-organizing map. *Neural Networks* 37: 52–65.
- Kormushev P, Nenchev DN, Calinon S and Caldwell DG (2011) Upper-body kinesthetic teaching of a free-standing humanoid robot. In: *2011 IEEE International Conference on Robotics and Automation*. IEEE, pp. 3970–3975.

- Levine S, Pastor P, Krizhevsky A, Ibarz J and Quillen D (2018) Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research* 37(4–5): 421–436.
- Liarokapis MV, Artemiadis PK and Kyriakopoulos KJ (2013) Quantifying anthropomorphism of robot hands. In: *2013 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 2041–2046.
- Lin B, He X, Zhou Y, Liu L and Lu K (2010) Approximately harmonic projection: Theoretical analysis and an algorithm. *Pattern Recognition* 43(10): 3307–3313.
- Lin J, Wu Y and Huang TS (2000) Modeling the constraints of human hand motion. In: *Proceedings of the Workshop on Human Motion, 2000*. IEEE, pp. 121–126.
- Lovchik C and Diftler MA (1999) The robonaut hand: A dexterous robot hand for space. In: *Proceedings 1999 IEEE International Conference on Robotics and Automation*, Vol. 2. IEEE, pp. 907–912.
- MacKenzie CL and Iberall T (1994) *The Grasping Hand*. Amsterdam: Elsevier.
- Nagata F, Watanabe K, Kiguchi K, et al. (2001) Joystick teaching system for polishing robots using fuzzy compliance control. In: *Proceedings 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation*. IEEE, pp. 362–367.
- Nair V and Hinton GE (2010) Rectified linear units improve restricted Boltzmann machines. In: *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 807–814.
- Narayanan H and Mitter S (2010) Sample complexity of testing the manifold hypothesis. In: *Advances in Neural Information Processing Systems*, pp. 1786–1794.
- Pao L and Speeter TH (1989) Transformation of human hand positions for robotic hand control. In: *Proceedings 1989 IEEE International Conference on Robotics and Automation*. IEEE, pp. 1758–1763.
- Park FC and Brockett RW (1994) Kinematic dexterity of robotic mechanisms. *The International Journal of Robotics Research* 13(1): 1–15.
- Peer A, Eidenkel S and Buss M (2008) Multi-fingered telemanipulation-mapping of a human hand to a three finger gripper. In: *The 17th IEEE International Symposium on Robot and Human Interactive Communication, 2008 (RO-MAN 2008)*. IEEE, pp. 465–470.
- Portillo-Vélez RdJ, Cruz-Villar CA, Rodríguez-Ángeles A and Arteaga-Pérez MA (2013) Master/slave robotic system for teaching motion-force manufacturing tasks. *Applied Mechanics and Materials* 307: 84–88.
- Rifai S, Vincent P, Muller X, Glorot X and Bengio Y (2011) Contractive auto-encoders: Explicit invariance during feature extraction. In: *Proceedings of the 28th International Conference on International Conference on Machine Learning*. Omnipress, pp. 833–840.
- Robotiq Inc. (2016) *Robotiq 3-Finger Adaptive Robot Gripper Instruction Manual*. Robotiq Inc.
- Rohling RN, Hollerbach JM and Jacobsen SC (1993) Optimized fingertip mapping: A general algorithm for robotic hand teleoperation. *Presence: Teleoperators and Virtual Environments* 2(3): 203–220.
- Salviati G (2018) Replicating human hand synergies onto robotic hands: A review on software and hardware strategies. *Frontiers in Neurobotics* 12: 27.
- Salviati G, Malvezzi M, Gioioso G and Prattichizzo D (2014) On the use of homogeneous transformations to map human hand movements onto robotic hands. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 5352–5357.
- Santello M, Flanders M and Soechting JF (1998) Postural hand synergies for tool use. *Journal of Neuroscience* 18(23): 10105–10115.
- Shi R, Zeng W, Su Z, et al. (2017) Hyperbolic harmonic mapping for surface registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39(5): 965–980.
- Tang B, Sapiro G and Caselles V (2000) Diffusion of general data on non-flat manifolds via harmonic maps theory: The direction diffusion case. *International Journal of Computer Vision* 36(2): 149–161.
- Zhang D and Hebert M (1999) Harmonic maps and their applications in surface matching. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999*, Vol. 2. IEEE, pp. 524–530.
- Zhu Y, Mottaghi R, Kolve E, et al. (2017) Target-driven visual navigation in indoor scenes using deep reinforcement learning. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 3357–3364.

Appendix A. Index to Multimedia Extensions

Extension	Type	Description
1	Video	Human-to-robot hand pose mapping: Training and real-time performance

Appendix B. Summary of notation

Notation	Description
\mathcal{Q}	Input domain
\mathcal{X}	Target domain
q	Variable in \mathcal{Q}
x	Variable in \mathcal{X}
f	Function from \mathcal{Q} to \mathcal{X}
g	Function from \mathcal{X} to \mathcal{Q}
$\tilde{\mathcal{Q}}$	Set of samples from \mathcal{Q}
$\tilde{\mathcal{X}}$	Set of samples from \mathcal{X}
N_A	Number of elements in a set A
J	Jacobian matrix
$\ J\ _F^2$	Distortion
\mathcal{S}	Set of reference pairs
α	Weighting coefficient in Equation (4) and (6)
\bar{d}	Sample mean distortion

Appendix C. Derivation of $Tr(J^T J) = \|J\|_{\mathcal{F}}^2$

We derive $Tr(J^T J) = \|J\|_{\mathcal{F}}^2$ as follows:

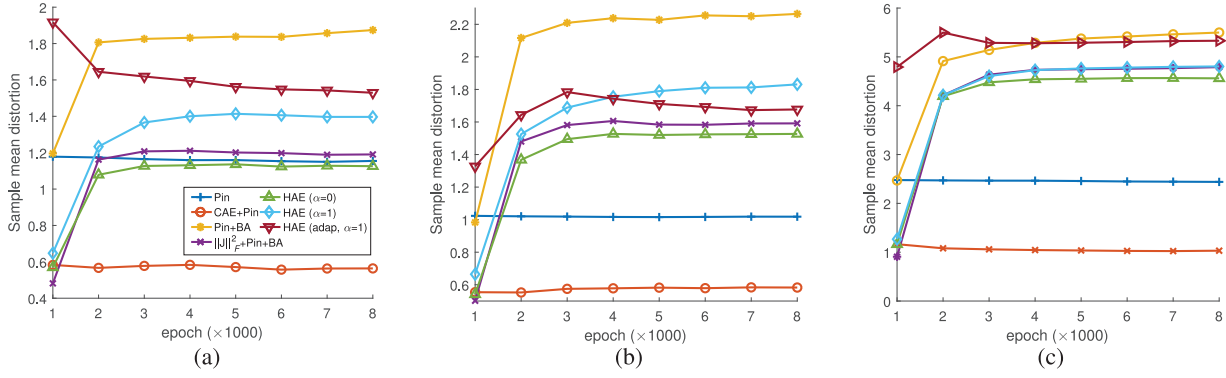


Fig. 14. Sample mean distortion \bar{d} with respect to learning epoch for synthetic data (with the number of reference pairs $L=4$): (a) SP2 = 3D U-shape to 2D triangle, (b) SP3 = 3D circle to 2D U-shape, and (c) SP4 = 2D square to 3D clover. Mean values over 10 repetitions for each method are shown.

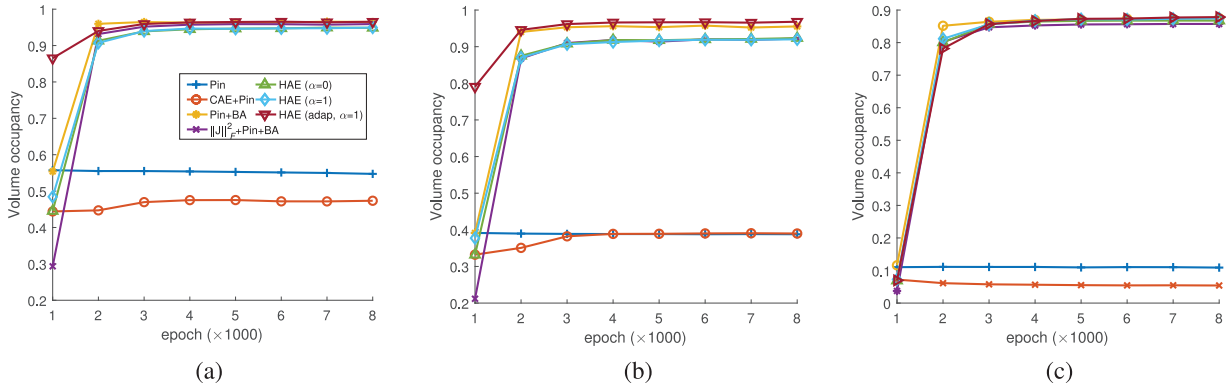


Fig. 15. Volume occupancy VO with respect to learning epoch for synthetic data (with the number of reference pairs $L=4$): (a) SP2 = 3D U-shape to 2D triangle, (b) SP3 = 3D circle to 2D U-shape, and (c) SP4 = 2D square to 3D clover. Mean values over 10 repetitions for each method are shown.

$$\begin{aligned} \text{Tr}(J^T J) &= \sum_{i=j} \sum_k J_{i,k}^T J_{k,j} \\ &= \sum_i \sum_k J_{k,i} J_{k,i} \\ &= \|J\|_{\mathcal{F}}^2 \end{aligned}$$

Appendix D. Derivation of J in Section 2.3

Given Equation (7), let $a = W_1 q + h_1$. Then element (m, n) of J is calculated using the chain rule as follows:

$$\begin{aligned} J_{mn} &= \frac{\partial f_m}{\partial q_n} \\ &= \sum_i \frac{\partial f_m}{\partial h_{1,i}} \frac{\partial h_{1,i}}{\partial a_i} \frac{\partial a_i}{\partial q_n} \\ &= \sum_i W_{2,mi} z_i W_{1,in} \end{aligned}$$

where $z_i = \partial h_{1,i} / \partial a_i = 1$ if $(W_1 q + b_1)_i > 0$; otherwise 0. Thus, $J = W_2 \cdot \text{Diag}(z) \cdot W_1$.

Appendix E. Derivation of $\|J\|_{\mathcal{F}}^2$ in Section 2.3

* From Appendix D,

$$\begin{aligned} \|J\|_{\mathcal{F}}^2 &= \sum_{mn} J_{mn} J_{mn} \\ &= \sum_{mn} \sum_i W_{2,mi} z_i W_{1,in} \sum_j W_{2,mj} z_j W_{1,jn} \\ &= \sum_{ij} z_i z_j \sum_m W_{2,mi} W_{2,mj} \sum_n W_{1,in} W_{1,jn} \\ &= \sum_{ij} z_i z_j (W_2^T W_2)_{ij} (W_1 W_1^T)_{ij} \\ &= \sum_{ij} (zz^T \odot W_2^T W_2 \odot W_1 W_1^T)_{ij} \end{aligned}$$

4.1 Appendix F. Figures for the synthetic data experiment in Section 3.2

Model performance curves for SP2 (3D U-shape to 2D triangle), SP3 (3D circle to 2D U-shape), and SP4 (2D square to 3D clover) are shown in Figures 14–19. Sample mean

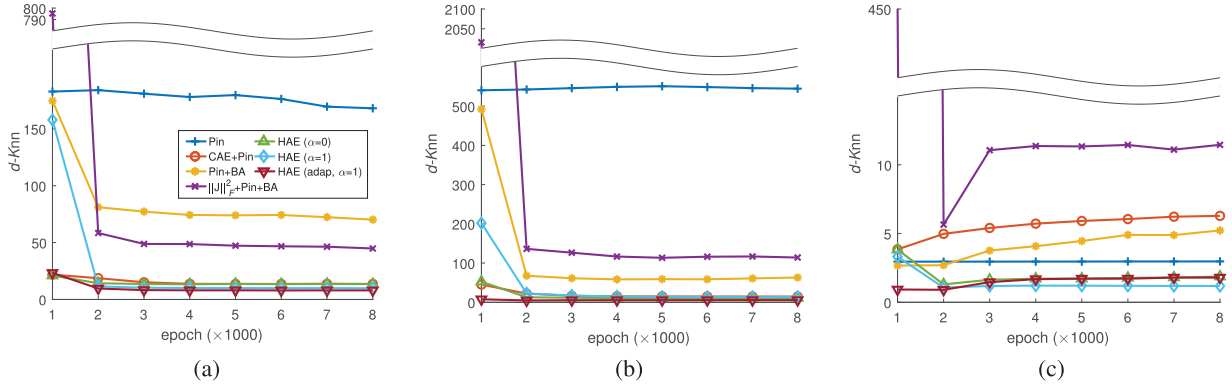


Fig. 16. Knn inverse image dissimilarity $d\text{-Knn}$ with respect to learning epoch for synthetic data (with the number of reference pairs $L = 4$): (a) SP2 = 3D U-shape to 2D triangle, (b) SP3 = 3D circle to 2D U-shape, and (c) SP4 = 2D square to 3D clover. Mean values over 10 repetitions for each method are shown.

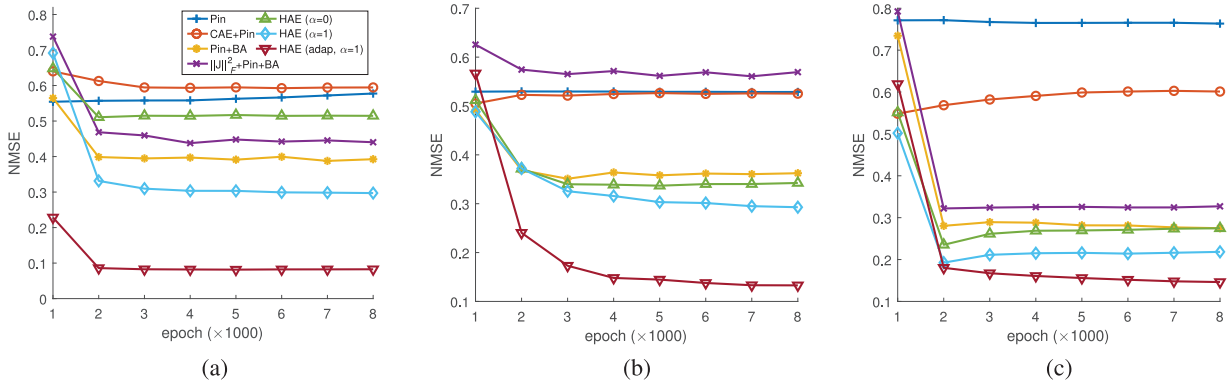


Fig. 17. NMSE with respect to learning epoch for synthetic data (with the number of reference pairs $L = 4$): (a) SP2 = 3D U-shape to 2D triangle, (b) SP3 = 3D circle to 2D U-shape, and (c) SP4 = 2D square to 3D clover. Mean values over 10 repetitions for each method are shown.

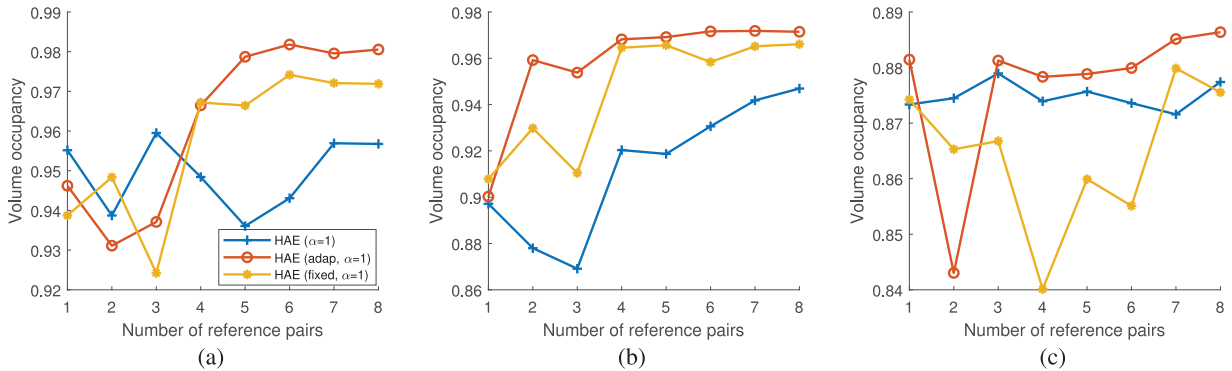


Fig. 18. Volume occupancy VO with respect to the number of reference pairs L for synthetic data: (a) SP2 = 3D U-shape to 2D triangle, (b) SP3 = 3D circle to 2D U-shape, and (c) SP4 = 2D square to 3D clover.

distortion \bar{d} , volume occupancy VO , K -nearest neighbor inverse image dissimilarity $d\text{-Knn}$, and NMSE with respect to learning epoch are shown in Figures 14, 15, 16, and 17,

respectively. The evaluation results for VO and NMSE with respect to the number of reference pairs L are shown in Figures 18 and 19, respectively.

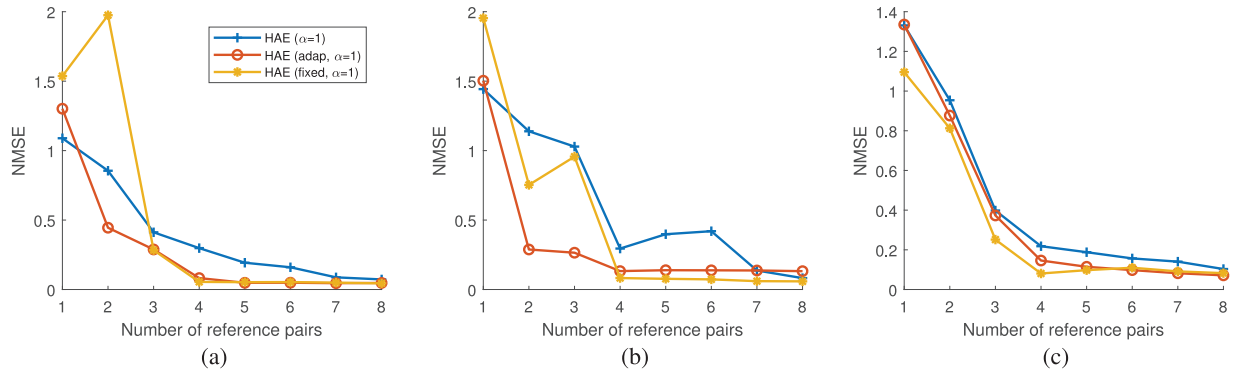


Fig. 19. NMSE with respect to the number of reference pairs L for synthetic data: (a) SP2 = 3D U-shape to 2D triangle, (b) SP3 = 3D circle to 2D U-shape, and (c) SP4 = 2D square to 3D clover.