Journal of Machine Learning Research 23 (2022) 1-57 Submitted 7/19; Revised 4/22; Published 5/22

# Manifold Coordinates with Physical Meaning

**Samson J. Koelle**[1] SJKOELLE@UW.EDU
**Hanyu Zhang**[1] HANYUZ6@UW.EDU
**Marina Meilă**[1,2] MMP@STAT.WASHINGTON.EDU
**Yu-Chia Chen**[2] YUCHAZ@UW.EDU
[1] *Department of Statistics*
*University of Washington*
*Seattle, WA 98195-4322, USA*
[2] *Department of Electrical and Computer Engineering*
*University of Washington*
*Seattle, WA 98195, USA*

**Editor:** Aapo Hyvärinen

## Abstract

Manifold embedding algorithms map high-dimensional data down to coordinates in a much lower-dimensional space. One of the aims of dimension reduction is to find *intrinsic coordinates* that describe the data manifold. The coordinates returned by the embedding algorithm are abstract, and finding their physical or domain-related meaning is not formalized and often left to domain experts. This paper studies the problem of recovering the meaning of the new low-dimensional representation in an automatic, principled fashion. We propose a method to explain embedding coordinates of a manifold as *non-linear* compositions of functions from a user-defined dictionary. We show that this problem can be set up as a sparse *linear Group Lasso* recovery problem, find sufficient recovery conditions, and demonstrate its effectiveness on data.

**Keywords:** dimension reduction, manifold learning, functional regression, gradient, group lasso

## 1. Introduction

Manifold learning (ML) algorithms, also known as embedding or unsupervised learning algorithms, map data from high or infinite-dimensional spaces to coordinates of a much lower-dimensional space. In the sciences, one of the motivating goals of dimension reduction is the discovery of descriptors of the data generating process. Linear dimension reduction algorithms like principal component analysis (PCA) and non-linear algorithms such as diffusion maps (Coifman and Lafon, 2006) are used in applications from genomics to astronomy to uncover the variables describing large-scale properties of the interrogated system.

For example, in chemistry, a common problem is to discover so-called *collective coordinates* describing the evolution of molecular configurations at long time scales, which correspond to macroscopically interesting transformations of the molecule, and can explain some of its properties (Clementi et al., 2000; Noé and Clementi, 2017). The molecular configuration is represented by the $3N_a$ vector of spatial locations of the $N_a$ atoms comprising the molecule. A *molecular dynamics (MD) simulation* produces a sample of molecular configurations;

the distribution of this sample describes the molecule's behavior in the given experimental conditions. It has been shown empirically that manifolds approximate these high-dimensional distributions (Dsilva et al., 2013). Figure 1a shows the toluene molecule, consisting of $N_a = 15$ atoms, and 1d shows the mapping of an MD simulated trajectory into $m = 2$ dimensions (the embedding coordinates) by a manifold learning algorithm. Visual inspection shows that this configuration space is well-approximated by a one-dimensional manifold parametrized by a geometric quantity, the *torsion* $g_1$ of the methyl bond, which is the angle formed by the planes inscribing the first three and last three atoms of the orange lines joining four atoms in Figure 1d. Thus, $g_1$ is a collective coordinate which explains the large scale data manifold by the rotation of the $CH_3$ methyl group relative to the plane of the other carbon atoms, filtered out from the faster modes of vibration by the manifold learning algorithm. Similarly, as shown in Figures 1e, 1h 1f, and 1i, the large scale geometry of the ethanol and malonaldehyde MD data is explained by two torsion angles each.

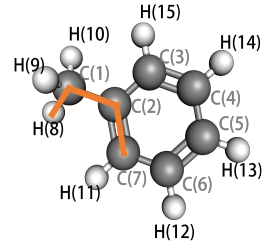
In this example, while the embedding algorithm was able to uncover the manifold structure of the data, finding the physical meaning of the manifold coordinates was done by visual inspection. In general, a scientist scans through many torsions and other functions of the configuration, in order to find ones that can be identified with the abstract coordinates output by a PCA or ML algorithm. Manual inspection of such denoised coordinates for correspondences with features of interest is pervasive in a variety of scientific fields (Chen et al., 2016; Herring et al., 2018; Banville et al., 2019). The goal of this paper is to put this process on a formal basis and to devise a method for automating this identification, thus removing the time consuming visual inspections from the shoulders of the scientist. We introduce a method to automate association of the meaningless abstract coordinates output by an embedding algorithm with functions of the data that are meaningful or interesting in the domain of the problem.

In our paradigm, the scientist has a *dictionary* $\mathcal{G}$ of functions to be considered as possible manifold coordinates. For the examples in Figure 1, $\mathcal{G}$ could be a set of candidate torsions. In other applications like single-cell genomics or astronomy, data are measurements in high-dimensional feature spaces such as gene counts, or spectra of stars and galaxies. The dictionary $\mathcal{G}$ then could consist of functions like cell-type specific signatures for the former, or element-specific spectral signatures for the latter (Blanton and Bershady, 2017; McQueen et al., 2016; Zhang et al., 2020).

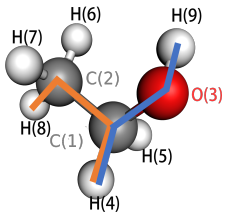
We assume that the data lie on a low-dimensional, smooth manifold $\mathcal{M}$ and that some embedding algorithm maps the data to coordinates denoted by $\phi_1, \dots \phi_m$. We propose an algorithm, ManifoldLasso, that replaces the abstract data-driven coordinates $\phi$ with an "equivalent" set of coordinates consisting of functions $g_1, \dots g_s$ from $\mathcal{G}$. Since our dictionary $\mathcal{G}$ is constructed using functions with physical meaning, this new set of coordinates may be considered to explain the manifold structure $\mathcal{M}$ of the data.

To keep the approach as general as possible, we do not rely on a particular embedding algorithm, making only the minimal assumption that it produces a smooth embedding. We also do not assume a parametric relationship between the embedding and the functions in the dictionary $\mathcal{G}$. We only assume that the mapping between the data manifold and the functions is sufficiently smooth.

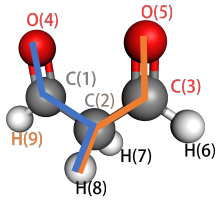
The next section defines the problem formally, and Section 3 presents the necessary background in manifold estimation. Section 4 develops our ManifoldLasso method.



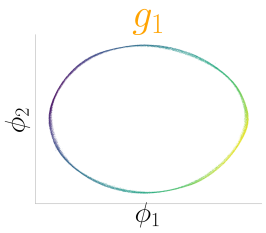

(a) Toluene



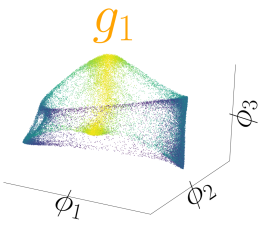

(b) Ethanol



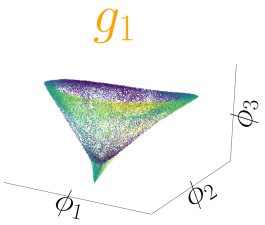

(c) Malonaldehyde



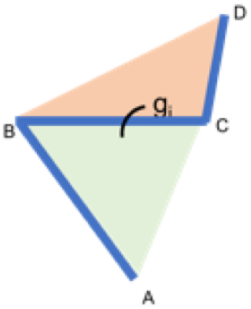

(d)



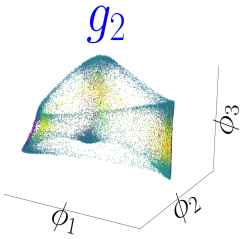

(e)



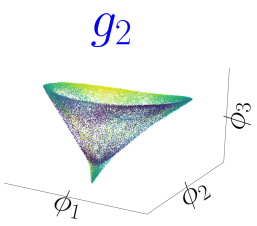

(f)




(g) Torsion example




(h)




(i)

Figure 1: Manifold coordinates with physical meaning in molecular dynamics (MD) simulations. 1a-1c Diagrams of the toluene ($C_7H_8$), ethanol ($C_2H_5OH$), and malonaldehyde ($C_3H_4O_2$) molecules, with the carbon (C) atoms in grey, the oxygen (O) atoms in red, and the hydrogen (H) atoms in white. Bonds defining important torsions $g_j$ are marked in orange and blue (see Section 7 for more details). The bond torsion is the angle of the planes inscribing the first three and last three atoms on the line (1g). 1d Embedding of the configurations of toluene into $m = 2$ dimensions, showing a manifold of $d = 1$. The color corresponds to the values of the orange torsion $g_1$. 1e, 1h Embedding of the configurations of the ethanol in $m = 3$ dimensions, showing a manifold of dimension $d = 2$, respectively colored by the blue and orange torsions in Figure 1b. 1f, 1i. Embedding of the configurations of malonaldehyde in $m = 3$ dimensions, showing a manifold of dimension $d = 2$, respectively colored by the blue and orange torsions in Figure 1c.

The relationship to previous work is discussed in Section 5. Section 6 presents theoretical recovery results, Section 7 presents experiments, and Section 8 concludes the paper. The Appendices present additional information including details about the functional dictionaries

used, adaptions necessary to make our method work in the rotation and translation invariant molecular configuration space, and a useful adaptation to our main algorithm for when support recovery conditions are violated.

## 2. Problem Formulation, Assumptions and Challenges

We make a number of standard manifold learning assumptions. Observed data $\mathcal{D} = \{\xi_i \in \mathbb{R}^D : i \in 1 \dots n\}$ are sampled i.i.d. from a *smooth manifold* [1] $\mathcal{M}$ of intrinsic dimension $d$ embedded in a feature space $\mathbb{R}^D$ by the inclusion map. In this paper, we will call *smooth* any function or manifold of class at least $\mathcal{C}^4$. We assume that the intrinsic dimension $d$ of $\mathcal{M}$ is known; for example, by having been estimated previously by the method in Kleindessner and von Luxburg (2015). The manifold $\mathcal{M}$ is a *Riemannian manifold* with *Riemannian metric* inherited from the ambient space $\mathbb{R}^D$. Furthermore, we assume the existence of a smooth *embedding map* $\phi : \mathcal{M} \to \phi(\mathcal{M}) \subset \mathbb{R}^m$, where typically $m << D$. That is, $\phi$ restricted to $\mathcal{M}$ is a diffeomorphism onto its image, and $\phi(\mathcal{M})$ is a submanifold of $\mathbb{R}^m$. We call the coordinates $\phi(\xi_i)$ in this $m$ dimensional ambient space the *embedding coordinates* $\phi_{1:m}$. In practice, the mapping of the data $\mathcal{D}$ onto $\phi(\mathcal{D})$ represents the output of an embedding algorithm, and we only have access to $\mathcal{M}$ and $\phi$ via $\mathcal{D}$ and its image $\phi(\mathcal{D})$.

As mentioned in the previous section, we are given a dictionary of user-defined and domain-related smooth functions $\mathcal{G} = \{g_1, \dots g_p$, with $g_j : U \subseteq \mathbb{R}^D \to \mathbb{R}\}$, where $U$ is an open set containing $\mathcal{M}$. We assume that $\phi(x) = h(g_{j_1}(x), \dots g_{j_s}(x))$, where $h : O \subseteq \mathbb{R}^s \to \mathbb{R}^m$ is a smooth function of $s$ variables, defined on a open subset of $\mathbb{R}^s$ containing the ranges of $g_{j_1}, \dots g_{j_s}$. Let $S = \{j_1, \dots j_s\}$, and $g_S = [g_{j_1}(x), \dots g_{j_s}(x)]^T$. We call this set the *functional support* or *explanation*. In differential geometric terms, $g_S$ is strongly related to finding *coordinate systems*, *charts* and *parameterizations* of $\mathcal{M}$. For example, in the toluene example, the functions in $\mathcal{G}$ are all the torsions in the molecule, $s = 1$, and $g_S = g_1$ is a chart for the 1-dimensional manifold traced by the configurations. Hence, it is natural to associate $s = d$.

Since the map $\phi$ given by the embedding algorithm is determined only up to diffeomorphism, the map $h$ cannot be uniquely determined, and it can therefore be overly restrictive to assume a parametric form for $h$. Hence, this paper aims to find the support set $S$ while circumventing the estimation of $h$. Indeterminacies w.r.t. the support $S$ itself are also possible. For instance, the support $S$ may not be unique whenever the relationship $g_1 = t(g_2)$ holds for two functions in $\mathcal{G}$ and for $t$ a smooth monotonic function. In Section 6 we give conditions under which $S$ can be recovered uniquely; intuitively, they consist of functional independencies between the functions in $\mathcal{G}$. For instance, it is sufficient to assume that that the dictionary $\mathcal{G}$ is a *functionally independent* set, i.e., there is no $g \in \mathcal{G}$ that can be obtained as a smooth function of other functions in $\mathcal{G}$.

## 3. Manifold Learning and Intrinsic Geometry

Our method relies on statistical estimators of several geometric quantities. One of the most important is the embedding map $\phi$. In addition to the embedding map itself, we also

1. The reader is referred to Lee (2003) for the definitions of the differential geometric terms used in this paper.

estimate the tangent spaces of $\mathcal{M}$ and $\phi(\mathcal{M})$. This will allow us to perform support recovery on the differential level. These estimation tasks are accomplished as follows.

### 3.1 The Neighborhood Graph and Kernel Matrix

The *neighborhood graph* is a data structure that encodes topological information about the dataset. It associates to each data point $\xi_i \in \mathcal{D}$ its set of *neighbors* $\mathcal{N}_i = \{i' \in [n], \text{with } ||\xi_{i'} - \xi_i|| \leq r_N\}$, where $r_N$ is a *neighborhood radius* parameter. The neighborhood relation is symmetric, and determines an undirected graph with nodes represented by the data points $\xi_{1:n}$. We denote $|\mathcal{N}_i|$ by $k_i$.

This graph is used in construction of the local position matrices $\Xi_i = [\xi_{i'} : i' \in \mathcal{N}_i] \in \mathbb{R}^{k_i \times D}$, local embedding coordinate matrices $\Phi_i = [\phi(\xi_{i'}) : i' \in \mathcal{N}_i] \in \mathbb{R}^{k_i \times m}$, and the *kernel matrix* $K \in \mathbb{R}^{n \times n}$ whose elements are

$$K_{i,i'} = \begin{cases} \exp\left(-\frac{||\xi_i - \xi_{i'}||^2}{\epsilon_N^2}\right) & \text{if } i' \in \mathcal{N}_i \text{ ;} \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

This matrix encodes geometric information about the dataset, and so is of crucial importance; $K$ is sparse, with sparsity structure induced by the neighborhood graph. Typically, the radius $r_N$ and the *bandwidth* parameter $\epsilon_N$ are related by $r_N = c\epsilon_N$ with $c$ a small constant greater than 1. This value ensures that the entries in $K$ that are zeroed out are small. Rows of the kernel matrix $K$ will be denoted $K_{i,\mathcal{N}_i}$ to emphasize that when a particular row is passed to an algorithm, only $k_i$ values need to be passed. Next, we show how the neighborhood graph, local position matrices, and kernel matrix are used in manifold estimation algorithms.

### 3.2 The Renormalized Graph Laplacian

The neighborhood graph and kernel matrix play essential roles in estimation of the *renormalized graph Laplacian*, also known as the *sample Laplacian*, or *diffusion maps Laplacian* $L$. This estimator, constructed by the Laplacian algorithm, converges to the manifold Laplace operator $\Delta_{\mathcal{M}}$; This estimator is unbiased w.r.t. the sampling density on $\mathcal{M}$ (Hein et al., 2005; Coifman and Lafon, 2006; Hein et al., 2007; Ting et al., 2010). $L$ is a sparse matrix; its $i$-th row contains non-zero values only for $i' \in \mathcal{N}_i$. Thus, as for $K$, elements and rows of this matrix will be denoted by $L_{i,i'}$ and $L_{i,\mathcal{N}_i}$, respectively, and the sparsity pattern of $L$ is given by the neighborhood graph. Construction of this neighborhood graph is the computational bottleneck of this algorithm; performed naively, constructing the neighborhood graph can have $\mathcal{O}(n^2)$ run time, but algorithms that pre-process the data can reduce the computational cost (Bernhardsson, 2015).

We use the $m$ principal eigenvectors of $L$ (or alternatively, of the matrix $\tilde{L}$ of Algorithm Laplacian) corresponding to its smallest non-zero eigenvalues as embedding coordinates. This embedding is known as the *diffusion map* (Coifman and Lafon, 2006) or the *Laplacian eigenmap* (Belkin and Niyogi, 2002) of $\mathcal{D}$. Although any algorithm which asymptotically generates a smooth embedding is acceptable for our general support recovery method, use of the eigenfunctions of the Laplacian as manifold embedding coordinates has special relevance to quantum systems (Heller, 1983; Zelditch, 2006; Landsman, 2007; Sogge, 2014).

LAPLACIAN (neighborhoods $\mathcal{N}_{i:n}$, local data $\Xi_{1:n}$, bandwidth $\epsilon_N$)

1: Compute kernel matrix $K$ using (1)
2: Compute normalization weights $w_i \leftarrow \sum_{i' \in \mathcal{N}_i} K_{i,i'}$, $i = 1, \dots n$, $W \leftarrow \mathrm{diag}(w_i \; i = 1:n)$
3: Normalize $\tilde{L} \leftarrow W^{-1} K W^{-1}$
4: Compute renormalization weights $\tilde{w}_i \leftarrow \sum_{i' \in \mathcal{N}_i} \tilde{L}_{i,i'}$, $i = 1, \dots n$, $\tilde{W} = \mathrm{diag}(\tilde{w}_i \; i = 1:n)$
5: Renormalize $L \leftarrow \frac{4}{\epsilon_N^2}(\tilde{W}^{-1}\tilde{L} - I_n)$
6: **Output** Kernel matrix $K$, Laplacian $L$, [optionally $\tilde{w}_{1:n}$]

### 3.3 Estimating Tangent Spaces in the Ambient Space $\mathbb{R}^D$

The differential quantities associated with $\phi$ and $\mathcal{G}$ that are used in our support estimation approach are computed w.r.t. the tangent bundle of $\mathcal{M}$. Since $\mathcal{M}$ is a submanifold of $\mathbb{R}^D$, the tangent space at data point $\xi_i$, denoted $\mathcal{T}_{\xi_i}\mathcal{M}$ is representable by an orthogonal basis matrix $T_i \in \mathbb{R}^{D \times d}$. The estimation of this matrix by *weighted local principal component analysis* (Chen et al., 2013) is described in the LOCALPCA algorithm. For this algorithm and others we denote by $\mathrm{SVD}(X, d)$ an algorithm as outputting $V, \Lambda$, where $\Lambda$ and $V$ are the largest $d$ eigenvalues and corresponding $d$ orthonormal eigenvectors, of symmetric matrix $X$, respectively. Denote a column vector of ones of length $k$ by $\mathbf{1}_k$.

LOCALPCA (local data $\Xi_i$, kernel row $K_{i,\mathcal{N}_i}$, intrinsic dimension $d$)

1: Compute normalization weights $w_i \leftarrow \sum_{i' \in \mathcal{N}_i} K_{i,i'}$
2: Compute weighted mean $\bar{\xi}_i \leftarrow \frac{1}{w_i} K_{i,\mathcal{N}_i} \Xi_i$
3: Compute weighted local differences
$Z_i \leftarrow \mathrm{diag}(K_{i,\mathcal{N}_i}^{1/2})(\Xi_i - \mathbf{1}_{k_i} \bar{\xi}_i)$
4: Compute $T_i, \Lambda \leftarrow \mathrm{SVD}(Z_i^T Z_i, d)$
5: **Output** $T_i$

### 3.4 The Pushforward Riemannian Metric

Geometric quantities such as angles and lengths of vectors in the tangent bundle $\mathcal{T}\mathcal{M}$ and distances along curves in $\mathcal{M}$ are captured by Riemannian geometry. Recall our assumption that $(\mathcal{M}, \mathbf{id})$ is a Riemannian manifold, with the metric $\mathbf{id}$ induced from $\mathbb{R}^D$. With this we associate to $\phi(\mathcal{M})$ a Riemannian metric $\mathbf{g}$ which preserves the geometry of $(\mathcal{M}, \mathbf{id})$. This metric, called *the pushforward metric*, is defined by

$$\langle u, v \rangle_{\mathbf{g}} \;=\; \langle D\phi^{-1}(\xi)u, D\phi^{-1}(\xi)v \rangle \quad \text{for all } u, v \in \mathcal{T}_{\phi(\xi)}\phi(\mathcal{M}). \tag{2}$$

In the above, $D$ denotes the differential operator, $D\phi^{-1}(\xi)$ the *pull-back* operator that maps vectors from $\mathcal{T}_{\phi(\xi)}\phi(\mathcal{M})$ to $\mathcal{T}_{\xi}\mathcal{M}$, and $\langle , \rangle$ the Euclidean scalar product.

For each $\phi(\xi_i)$, the pushforward Riemannian metric expressed in the coordinates of $\mathbb{R}^m$ is a symmetric, semi-positive definite $m \times m$ matrix $G_i$ of rank $d$. The scalar product $\langle u, v \rangle_{\mathbf{g}}$ takes the form $u^T G_i v$. The matrices $G_i$ can be estimated by the algorithm RMETRIC of Perraul-Joncas and Meila (2013). The algorithm uses only local information, and thus

can be run efficiently using the Laplacian, the neighborhood graph, and local embedding coordinate matrices. In the next section, we will use the output of this algorithm to estimate the differential $D\phi$.

RMetric (Laplacian row $L_{i,\mathcal{N}_i}$, local embedding coordinates $\Phi_i$, intrinsic dimension $d$)

1: Compute centered local embedding coordinates $\tilde{\Phi}_i = \Phi_i - \mathbf{1}_{k_i}\phi(\xi_i)^T$
2: Form matrix $H_i$ by $H_i \leftarrow [H_{i,k,k'}]_{k,k'\in 1:m}$ with $H_{i,k,k'} = \sum_{i'\in\mathcal{N}_i} L_{i,i'}\tilde{\Phi}_{i,i',k}\tilde{\Phi}_{i,i',k'}$ for $k, k' = 1:m$.
3: Compute $V_i, \Lambda_i \leftarrow \mathrm{SVD}(H_i, d)$
4: $G_i \leftarrow V_i\Lambda_i^{-1}V_i^T$.
5: **Output** $G_i$, optionally $V_i, \Lambda_i$

## 4. The ManifoldLasso Algorithm

The main idea of our approach is to exploit the well-known mathematical fact that, for any differentiable functions $f, g, h$, when $f = h \circ g$, the differentials $Df, Dh, Dg$ at any point are in the *linear* relationship $Df = DhDg$. Since, given coordinate functions $\phi_{1:m}$ and dictionary functions $g_{1:p}$ on a smooth manifold $\mathcal{M}$, our goal is to recover a subset $g_S$ of $g_{1:p}$ such that $\phi_{1:m} = h \circ g_S$ without knowing $h$, we propose to recover the subset $g_S$ by solving a set of *dependent linear sparse recovery problems*, one for each data point. This linear relationship $D\phi = DhDg_S$ will be written in terms of gradients $\mathrm{grad}_{\mathcal{M}}\,\phi_{1:m}$ and $\mathrm{grad}_{\mathcal{M}}\,g_{1:p}$. This section describes how to obtain the relevant gradients and solve the resulting optimization problem corresponding to sparse recovery.

### 4.1 Algorithm Overview

The ManifoldLasso algorithm, the main algorithm of this paper, implements this idea. It takes as input data $\mathcal{D}$ sampled from an unknown manifold $\mathcal{M}$, a dictionary $\mathcal{G}$ of functions defined on $\mathcal{M}$ (or alternatively on an open subset of the ambient space $\mathbb{R}^D$ that contains $\mathcal{M}$), and an embedding $\phi(\mathcal{D})$ in $\mathbb{R}^m$. The output of ManifoldLasso is a set $S$ of indices in $\mathcal{G}$, representing the functions in $\mathcal{G}$ that explain $\mathcal{M}$.

The first part of the algorithm contains preparatory steps for geometric analysis covered in Section 3. Steps 1 and 2 construct the neighborhood graph and the Laplacian matrix used for manifold learning and tangent space estimation.

The second part of ManifoldLasso calculates the necessary gradients; this comprises Steps 9–11. In Step 9, we estimate orthogonal bases of tangent subspaces by the LocalPCA algorithm described in Section 3. The gradients of the dictionary w.r.t. the manifold are then obtained as columns of the $d \times p$ matrix $X_i$ in Steps 5, 6, and 10. These operations are described in detail in Section 4.3. In Step 11, the gradients at $\xi_i$ of the coordinates $\phi_{1:m}$, also w.r.t. $\mathcal{M}$, are calculated as columns of the $d \times m$ matrix $Y_i$ by the PullBackDPhi algorithm described in Section 4.4.

In the last part of ManifoldLasso, Step 14 finds the support $S$ by solving the sparse regression. A GroupLasso algorithm is called to perform the sparse regression of the manifold coordinates' gradients $Y_{1:n}$ on the gradients of the dictionary functions, represented

by $X_{1:n}$. The indices of those dictionary functions whose $\beta$ coefficients are not identically null represent the support set $\operatorname{supp}\beta$. This is described in Section 4.5. Scaling of functions is addressed through normalization in Steps 6 and 13; this procedure is described in more detail in Section 4.7.

There are several optional steps and substitutions in our algorithm. An embedding can be computed in Step 3, or input separately by the user - we denote this step generically as EmbeddingAlg. Finally, although we explicitly describe tangent space estimation methods of both $\mathcal{T}_{\xi_i}\mathcal{M}$ and $\mathcal{T}_{\phi(\xi_i)}\phi(\mathcal{M})$ in our algorithms, other approaches to estimate them may be used.

---

ManifoldLasso (Dataset $\mathcal{D}$, dictionary $\mathcal{G}$, embedding coordinates $\phi(\mathcal{D})$, intrinsic dimension $d$, kernel bandwidth $\epsilon_N$, neighborhood cutoff size $r_N$, regularization parameter $\lambda$)

---

1: Construct $\mathcal{N}_i$ for $i = 1 : n$; $i' \in \mathcal{N}_i$ iff $||\xi_{i'} - \xi_i|| \leq r_N$, and local data matrices $\Xi_{1:n}$
2: Construct kernel matrix and Laplacian $K, L \leftarrow$ Laplacian$(\mathcal{N}_{1:n}, \Xi_{1:n}, \epsilon_N)$
3: [Optionally compute embedding: $\phi(\xi_{1:n}) \leftarrow$EmbeddingAlg$(\mathcal{D}, \mathcal{N}_{1:n}, m, \ldots)$]
4: **for** $j = 1, 2, \ldots p$ **do**
5: Compute $\nabla_\xi g_j(\xi_i)$ for $i = 1, \ldots n$
6: Compute $\zeta_j^2$ by (11) and normalize $\nabla_\xi g_j(\xi_i) \leftarrow (1/\zeta_j)\nabla_\xi g_j(\xi_i)$ for $i = 1, \ldots n$
7: **end for**
8: **for** $i = 1, 2, \ldots n$ **do**
9: Compute basis $T_i^{\mathcal{M}} \leftarrow$LocalPCA$(\Xi_i, K_{i,\mathcal{N}_i}, d)$
10: Project $X_i \leftarrow (T_i^{\mathcal{M}})^T \nabla_\xi g_{1:p}$
11: Compute $Y_i \leftarrow$PullBackDPhi$(\Xi_i, \Phi_i, T_i^{\mathcal{M}}, L_{i,\mathcal{N}_i}, d)$
12: **end for**
13: Compute $\zeta_k^2 \leftarrow \frac{1}{n}\sum_{i=1}^n \|y_{ik}\|^2$ (i.e., (10)), for $k = 1, \ldots m$ and normalize $Y_i \leftarrow Y_i \operatorname{diag}\{1/\zeta_{1:m}\}$, for $i = 1, \ldots n$.
14: $\beta \leftarrow$ GroupLasso$(X_{1:n}, Y_{1:n}, \lambda)$
15: **Output** $S = \operatorname{supp}\beta$

---

### 4.2 Gradients and Coordinate Systems

Our algorithm regresses the gradients of the embedding coordinate functions against the gradients of the dictionary functions. Both sets of gradients are with respect to the manifold $\mathcal{M}$, and so this requires calculating or estimating various gradients in the same $d$-dimensional coordinate system. This and the following two sections explain these procedures.

First, note that by assumption we have two Euclidean spaces $\mathbb{R}^D$ and $\mathbb{R}^m$, in which manifolds $\mathcal{M}$ and $\phi(\mathcal{M})$ of dimension $d$ are embedded. Denote gradients w.r.t. the Euclidean coordinate systems in $\mathbb{R}^D$ and $\mathbb{R}^m$ by $\nabla_\xi$ and $\nabla_\phi$, respectively. Since our interest is in functions on manifolds, we also define the gradient of a function on a manifold $\mathcal{M}$. The *gradient* of $f$ at $\xi$, on a Riemannian manifold $(\mathcal{M}, \mathbf{g})$, denoted $\operatorname{grad}_{\mathcal{M}} f(\xi) \in \mathcal{T}_\xi\mathcal{M}$, is defined by the identity

$$\langle \operatorname{grad}_{\mathcal{M}} f(\xi), u\rangle_{\mathbf{g}} = Df(\xi)u \qquad \text{for any } u \in \mathcal{T}_\xi\mathcal{M}. \tag{3}$$

At each data point $\xi_i$, we fix bases $T_i^{\mathcal{M}}$ in $\mathcal{T}_{\xi_i}\mathcal{M}$ and $T_i^{\phi}$ in $\mathcal{T}_{\phi(\xi_i)}\phi(\mathcal{M})$. Gradients expressed in these coordinate systems are denoted by $\operatorname{grad}_{T_i^{\mathcal{M}},\mathbf{g}}$ and $\operatorname{grad}_{T_i^{\phi},\mathbf{g}}$ respectively. For a manifold

$\mathcal{M}$ which is a submanifold of $\mathbb{R}^D$, we denote by $\operatorname{grad}_{T_i^{\mathcal{M}}}(\xi)$ the value of $\operatorname{grad}_{T_i^{\mathcal{M}},\mathbf{id}}(\xi)$ w.r.t. the identity metric **id** inherited from $\mathbb{R}^D$, and by $\langle , \rangle$ the Euclidean scalar product.

Note that in coordinates, $\operatorname{grad}_{\mathcal{M}} f$ depends on the metric $\mathbf{g}$, but at the same time, this definition shows that $\operatorname{grad}_{\mathcal{M}} f$ as a linear operator on $\mathcal{T}_{\xi_i}\mathcal{M}$ is invariant to the metric; it is just the first order derivative reflecting how $f$ changes along the manifold. Hence, the left hand side must also be invariant to the metric. It follows that $Df(\xi)u = u^T \operatorname{grad}_{T_i^{\mathcal{M}}}(\xi_i)$ for any $u \in \mathcal{T}_{\xi}\mathcal{M}$, and, furthermore, that $\operatorname{grad}_{T_i^{\mathcal{M}},\mathbf{g}} = G_i^{-1} \operatorname{grad}_{T_i^{\mathcal{M}}}$ for any other Riemannian metric $\mathbf{g}$.

### 4.3 Calculating the Gradients of the Dictionary Functions

Our goal is to obtain $X_i$, the matrix defined by

$$X_i \;=\; [\operatorname{grad}_{T_i^{\mathcal{M}}} g_j(\xi_i)]_{j=1:p} \in \mathbb{R}^{d\times p}. \tag{4}$$

Let $g_j$ be a function in the dictionary $\mathcal{G}$. By definition, for any basis $T_i^{\mathcal{M}} \in \mathbb{R}^{D\times d}$ of $\mathcal{T}_{\xi_i}\mathcal{M}$,

$$\operatorname{grad}_{T_i^{\mathcal{M}}} g_j(\xi_i) \;=\; (T_i^{\mathcal{M}})^T \nabla_{\xi} g_j(\xi_i).$$

In other words, $\operatorname{grad}_{T_i^{\mathcal{M}}} g_j$ is the projection of $\nabla_{\xi} g_j$ on the basis $T_i^{\mathcal{M}}$. These bases of $\mathcal{T}_{\xi_i}\mathcal{M}$, for every $i$, are estimated by LocalPCA as described in Section 3. The gradients $\nabla_{\xi} g_j(\xi_i)$ are known analytically, by assumption. We thus construct matrices $X_i$, for $i = 1, \dots n$, with $p$ columns representing the gradients of the $p$ dictionary functions as $X_i = (T_i^{\mathcal{M}})^T \nabla_{\xi} g_{1:p}$, as in Step 10 of Algorithm ManifoldLasso. Now we turn to obtaining the manifold gradients of the coordinate functions $\phi_k$ in the same coordinate system.

### 4.4 Estimating the Coordinate Gradients by Pull-back

Since $\phi$ is implicitly determined by a manifold embedding algorithm, the gradients of $\phi_k$ are often not analytically available, and $\phi_k$ is known only through its values at the data points. We therefore introduce an estimator of these gradients based on the notion of vector *pull-back* between tangent spaces. Instead of estimating gradients naively from differences $\phi_k(\xi_i) - \phi_k(\xi_{i'})$ between neighboring points, we first estimate their values in $\mathcal{T}_{\phi(\xi_i)}\phi(\mathcal{M})$, where they have a simple expression, then pull them back in the coordinate system $T_i^{\mathcal{M}}$. This estimation method is novel, and of some independent interest. A schematic of this approach is given in Figure 2.

The PullBackDPhi Algorithm takes as inputs the local neighborhoods $\Xi_i$, $\Phi_i$ of point $\xi_i$ in the original and embedding spaces, respectively, the basis $T_i^{\mathcal{M}}$ of $\mathcal{T}_{\xi_i}\mathcal{M}$, and the row of the Laplacian matrix corresponding to $i$, $L_{i,\mathcal{N}_i}$. From this local information, the algorithm first computes the tangent space $\mathcal{T}_{\phi(\xi_i)}\phi(\mathcal{M})$, then obtains the gradients of the coordinate functions $\phi$ in this space by projection, and finally pulls back these gradients in the coordinate system given by $T_i^{\mathcal{M}}$ by solving a least squares regression.

#### 4.4.1 The Tangent Space $\mathcal{T}_{\phi(\xi_i)}\phi(\mathcal{M})$

When $m = d$, this space is trivially equal to $\mathbb{R}^d$, so the problem is interesting in the case $m > d$. If the embedding induced by $\phi$ were an isometry, the estimation of $\mathcal{T}_{\phi(\xi_i)}\phi(\mathcal{M})$ could

be performed by LocalPCA, and the subsequent pull-back could be done as described in Luo et al. (2009). Here we do *not* assume that $\phi$ is isometric.

The method we introduce uses the push-forward Riemannian metric $\mathbf{g}$, expressed as $G_i$ in the coordinates $\phi$ at $\xi_i$, to estimate the $\mathcal{T}_{\phi(\xi_i)}\phi(\mathcal{M})$. By definition, the theoretical rank of $G_i$ equals $d$ and the $d$ principal eigenvectors of $G_i$ represent an orthonormal basis of $\mathcal{T}_{\phi(\xi_i)}\phi(\mathcal{M})$. $G_i$ and its decomposition are estimated by the RMetric algorithm described in Section 3. We denote this basis $T_i^{\phi} \in \mathbb{R}^{m\times d}$.

#### 4.4.2 The Gradient $\mathrm{grad}_{\phi(\mathcal{M})}\,\phi_k$

Trivially, the gradients of $\phi_{1:m}$ in the embedding space $\mathbb{R}^m$, are equal to the $m$ basis vectors of $\mathbb{R}^m$, i.e., $\nabla_{\phi}\phi_{1:m} = I_m$. Therefore $\mathrm{grad}_{\phi(M)}\,\phi_k$, expressed in the basis $T_i^{\phi}$, given by the top $d$ the eigenvectors of $G_i$, is equal to the projection of the corresponding basis vector onto the tangent subspace $\mathcal{T}_{\phi(\xi_i)}\phi(\mathcal{M})$. In matrix form we have

$$[\mathrm{grad}_{T_i^{\phi}}\,\phi_k(\xi_i)]_{k=1}^{m} \;=\; (T_i^{\phi})^T I_m.$$

#### 4.4.3 Pulling Back $\mathrm{grad}_{\phi(\mathcal{M})}\,\phi$ into $\mathcal{T}_{\xi}\mathcal{M}$

In order to bring these gradients into the same coordinate system as our dictionary functions, we define the following matrices, with $\mathrm{Proj}_T\, v$ denoting the Euclidean projection of vector $v$ onto subspace $T$.

$$Y_i \;=\; [y_{ik}]_{k=1}^{m} \;=\; [\mathrm{grad}_{T_i^{\mathcal{M}}}\,\phi_k(\xi_i)]_{k=1}^{m} \in \mathbb{R}^{d\times m},$$

$$A_i \;=\; \Big[\mathrm{Proj}_{\mathcal{T}_{\xi_i}\mathcal{M}}(\xi_{i'} - \xi_i)\Big]_{i'\in\mathcal{N}_i} \in \mathbb{R}^{d\times k_i}\,,$$

and

$$B_i \;=\; [\phi(\xi_{i'}) - \phi(\xi_i)]_{i'\in\mathcal{N}_i} \;\in \mathbb{R}^{m\times k_i}, \quad \tilde{B}_i \;=\; \Big[\mathrm{Proj}_{\mathcal{T}_{\phi(\xi_i)}\phi(\mathcal{M})}\,[\phi(\xi_{i'}) - \phi(\xi_i)]\Big]_{i'\in\mathcal{N}_i}, \in \mathbb{R}^{d\times k_i}.$$

The columns of $A_i$ and $Y_i$ are vectors in $\mathcal{T}_{\xi_i}\mathcal{M}$, the columns of $B_i$ are in $\mathbb{R}^m$ and the columns of $\tilde{B}_i$ are in $\mathcal{T}_{\phi(\xi_i)}\phi(\mathcal{M})$. These vectors are shown schematically in Figure 2. Note that when $m = d$, $B_i = \tilde{B}_i$.

The key property that enables our estimator of $Y_i$ is that the columns of $A_i$ and $\tilde{B}_i$ are in correspondence, because they represent (approximately) the same vectors in two different coordinate systems, namely the *logarithmic maps* of point $i'$ in $\mathcal{M}$ and $\phi(\mathcal{M})$ with respect to point $i$. The accuracy of this approximation is shown in Appendix A. The idea of the algorithm is then to use this correspondence in order to pull back the gradient of the coordinate function $\phi_k$ into the coordinates $T_i^{\mathcal{M}}$.

Specifically, since $D\phi_k$, the differential of $\phi_k : \mathcal{M} \to \mathbb{R}$, as a linear functional on the tangent bundle $\mathcal{T}\mathcal{M}$ is invariant to coordinate system, we calculate its value on the columns of $\tilde{B}_i$ in the coordinate system given by $\phi$ itself, and equate these values with $\mathrm{grad}_{T_i^{\mathcal{M}}}\,\phi_k$ applied to the columns of $A_i$, an expression in the coordinates $T_i^{\mathcal{M}}$. By (3) and Appendix A, we have that

$$(\mathrm{grad}_{T_i^{\mathcal{M}}}\,\phi_k(\xi_i))^T A_i \;=\; (\mathrm{grad}_{T_i^{\phi}}\,\phi_k(\xi_i))^T \tilde{B}_i + o(r_N).$$

In coordinates, $A_i = (T_i^{\mathcal{M}})^T(\Xi_i^T - \xi_i \mathbf{1}_{k_i}^T)$ and $\tilde{B}_i = (T_i^{\phi})^T(\Phi_i^T - \phi(\xi_i)\mathbf{1}_{k_i}^T)$. These matrices are computed by Steps 2 and 3 of Algorithm PullBackDPhi, while $Y_i$ contains the gradients we want to estimate. The error term comes from approximating the logarithmic map applied to points $\xi_{i'}$ and $\phi(\xi_{i'})$ for $i' \in \mathcal{N}_i$ with the columns of $A_i$ and $\tilde{B}_i$. Recalling that $Y_i = [\text{grad}_{T_i^{\mathcal{M}}} \phi_k(\xi_i)]_{k=1:m}$ we obtain

$$Y_i^T A_i \;=\; [(T_i^{\phi})^T I_m]^T (T_i^{\phi})^T B_i + o(r_N). \tag{5}$$

We solve this linear system in the least squares sense

$$Y_i \;=\; \arg\min_{Y \in \mathbb{R}^{d\times m}} \|A_i^T Y - B_i^T T_i^{\phi}(T_i^{\phi})^T\|^2 \tag{6}$$

to obtain

$$Y_i \;=\; A_i^{\dagger} B_i^T T_i^{\phi}(T_i^{\phi})^T. \tag{7}$$

This solution is effectively the regression of the columns of $B_i T_i^{\phi}(T_i^{\phi})^T$ on the columns of $A_i$ at each data point $\xi_i$. We call estimator (7) the *pullback gradient estimator* because of its implicit invocation of the notion of vector pullback.

To see this a different way, note that by equation (2), for any function $f : \phi(\mathcal{M}) \to \mathbb{R}$,

$$\langle D\phi^{-1}u, D\phi^{-1}\,\text{grad}_{\phi(\mathcal{M})}\, f\rangle \;=\; \langle u, \text{grad}_{\phi(\mathcal{M})}\, f\rangle_{\mathbf{g}}, \quad \text{for all } u \in \mathcal{T}_{\phi(\xi_i)}\phi(\mathcal{M})$$

where $g$ is the push-forward metric associated with $\phi$. Using this fact, and the invariance of gradient to metric, we have that, for any $w \in \mathcal{T}_{\xi_i}\mathcal{M}$, $D\phi^{-1}\,\text{grad}_{\phi(\mathcal{M})}\, f = \text{grad}_{\mathcal{M}}(f \circ \phi)$ for any smooth function $f : \phi(\mathcal{M}) \to \mathbb{R}$. The above claims give us $\langle D\phi^{-1}u, \text{grad}_{\mathcal{M}}(f\circ\phi)\rangle = \langle u, \text{grad}_{\phi(\mathcal{M})}\, f\rangle$ where $u \in \mathcal{T}_{\phi(\xi_i)}\phi(\mathcal{M})$ is an arbitrary tangent vector. In coordinates $T_i^{\phi}$ and $T_i^{\mathcal{M}}$, we can write this equivalence as

$$\langle D\phi^{-1}u, \text{grad}_{T_i^{\mathcal{M}}}(f\circ\phi)\rangle = \langle u, \text{grad}_{T_i^{\phi}}\, f\rangle.$$

If we then replace values of $(T_i^{\phi})^T e^k$, $(T_i^{\mathcal{M}})^T(\xi_{i'} - \xi_i)$ and $(T_i^{\phi})^T(\phi(\xi_{i'}) - \phi(\xi_i))$ for $\text{grad}_{T_i^{\phi}}\phi_k$, $D\phi^{-1}u$ and $Du$, respectively, we obtain (5).

---

PullBackDPhi local data $\Xi_i$, local embedding coordinates $\Phi_i$, basis $T_i^{\mathcal{M}}$ (Optional: $T_i^{\phi}$ or Laplacian row $L_{i,\mathcal{N}_i}$, intrinsic dimension $d$)

---

1: Compute pushforward metric eigendecomposition $T_i^{\phi}, G_i \leftarrow \text{RMetric}(L_{i,\mathcal{N}_i}, \Phi_i, d)$.
2: Compute $B_i \leftarrow (\Phi_i^T - \phi(\xi_i)\mathbf{1}_{k_i}^T)$
3: Compute $A_i \leftarrow (T_i^{\mathcal{M}})^T(\Xi_i^T - \xi_i \mathbf{1}_{k_i}^T)$
4: Calculate $Y_i \leftarrow A_i^{\dagger} B_i^T T_i^{\phi}(T_i^{\phi})^T$ by solving linear system (6)
5: **Output** $Y_i$

---

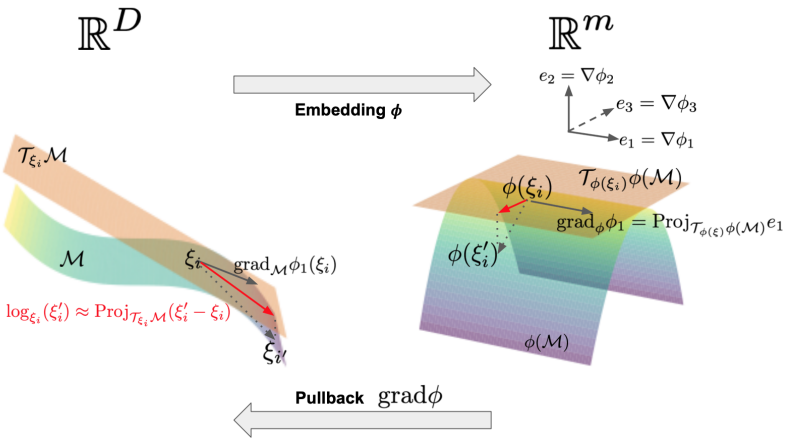



Figure 2: Left: $\mathcal{M}$ with a tangent subspace at $\xi_i$, $\xi_{i'} - \xi_i$ (in black, dotted), the projection $\text{Proj}_{\mathcal{T}_{\xi_i}\mathcal{M}}(\xi_{i'} - \xi_i)$ (in red), and the manifold gradient $\text{grad}_{\mathcal{M}}\phi_1(\xi_i)$ of the first embedding coordinate $\phi_1$ (in black). Right: $\phi(\mathcal{M})$ and tangent subspace at $\phi(\xi_i)$, with $\phi(\xi_{i'}) - \phi(\xi_i)$ (in black, dotted), $\text{Proj}_{\mathcal{T}_{\phi(\xi_i)}\phi(\mathcal{M})}(\phi(\xi_{i'}) - \phi(\xi_i))$ (in red) and the manifold gradient $\text{grad}_{\phi(\mathcal{M})}\phi_1$ (in black). $A_{i,i'}$ is the (approximate) mapping of $B_{i,i'}$ through $D\phi^{-1}(\xi_i)$, as in (2). The gradient $\text{grad}_{\phi(\mathcal{M})}\phi_1$ is the the projection of the first unit vector onto $\mathcal{T}_{\phi(\xi_i)}\phi(\mathcal{M})$.

### 4.5 The GroupLasso Formulation

With the estimated gradients, we are now ready to resolve the functional support problem. Recall that $X_i$ defined in (4) contains the gradients of the dictionary functions $g_j$, and that $y_{ik} \in \mathbb{R}^d$, the $k$-th column of $Y_i$, represents the coordinates of $\text{grad}_{\mathcal{M}}\phi_k(\xi_i)$ in the chosen basis of $\mathcal{T}_{\xi_i}\mathcal{M}$. Further, given our assumption that $\phi = h \circ g_S$, let $h_k$ be the $k$-th component of the vector valued function $h$, and denote

$$\beta_{ijk} = \frac{\partial h_k}{\partial g_j}(g_j(\xi_i)), \quad \beta = [\beta_{ijk}]_{i,k,j=1}^{n,m,p},$$

$$\beta_j = \text{vec}(\beta_{ijk},\ i = 1:n, k = 1:m) \in \mathbb{R}^{mn}, \quad \beta_{ik} = \text{vec}(\beta_{ijk},\ j = 1:p) \in \mathbb{R}^p.$$

Then since $\text{grad}_{\mathcal{M}}\phi_k = \text{grad}_{\mathcal{M}}(h_k \circ g_S)$ , for any point $\xi_i$ and any $v \in \mathcal{T}_{\xi_i}\mathcal{M}$ it holds from chain rule that

$$\begin{aligned}\langle \text{grad}_{\mathcal{M}}(h_k \circ g_S)(\xi_i), v\rangle_{\mathbf{g}} &= D(h_k \circ g_S)(\xi_i)v \\ &= Dh_k(g_S(\xi_i)[Dg_S(\xi_i)(v)] \\ &= Dh_k(g_S(\xi_i))\langle \text{grad}_{\mathcal{M}}\, g_S(\xi_i), v\rangle_{\mathbf{g}} \end{aligned} \tag{8}$$

Note that both inner product and $Dh_k(g_S(\xi_i))$ are linear mappings, we conclude that $\text{grad}_{\mathcal{M}}\phi_k = Dh_k(g_S(\xi_i))\,\text{grad}_{\mathcal{M}}\, g_S(\xi)$. In coordinates, we have

$$y_{ik} \;=\; \sum_{j=1}^{p} \beta_{ijk}x_{ij} \;=\; X_i\beta_{ik} \quad \text{for all } i = 1:n, \text{ and } k = 1:m.$$

In the above regression of $Y_{1:n}$ on $X_{1:n}$, $\beta_{ik}$ is the set of regression coefficients of $y_{ik}$ onto $X_i$. If there is some $h$ such that $\phi = h \circ g_S$, then the non-zero $\beta_{ijk}$ coefficients are estimates of

$\frac{\partial h}{\partial g_j}$ for $j \in S$. Further, $\beta_j$ represents the vector of regression coefficients corresponding to the effect of function $g_j$; therefore, the zero $\beta_j$ vectors indicate that $j \notin S$. Hence, in each $\beta_{ik}$, only $|S|$ elements are non-zero.

The key characteristic of the functional support that we leverage is that the same set $S$ of coefficients will be non-zero for all $i$ and $k$. Since finding this set $S \subset [p]$ is underdetermined, we use a sparsity inducing regularization that simultaneously zeros out entire $\beta_j$ vectors. Thus, our problem can be naturally expressed as a *group lasso* (Yuan and Lin, 2006), with $p$ groups of size $mn$, consisting of the $\beta_{1:p}$ groups of coefficients of $\text{grad}_{\mathcal{M}}\, g_{1:p}$. To solve it we minimize the following objective function w.r.t. $\beta$:

$$J_\lambda(\beta) \;=\; \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{m} ||y_{ik} - X_i\beta_{ik}||^2 + \frac{\lambda}{\sqrt{mn}}\sum_{j=1}^{p} ||\beta_j||. \tag{9}$$

The first term of the objective is the least squares loss of regressing $Y_{1:n}$ onto $X_{1:n}$. The second is a regularization term, which penalizes each group $\beta_j$ by its Euclidean norm. This encourages most $\beta_j$ groups to be identically 0. The normalization of the regularization coefficient $\lambda$ by the group size $mn$ follows Yuan and Lin (2006) takes into account that the least squares loss also grows proportionally to $mn$. The use of group lasso for sparse functional regression was introduced in Meila et al. (2018).

Note that $J_\lambda(\beta)$ is convex in $\beta$ and invariant to the change of basis $T_i$. Let $\tilde{T}_i = T_i\Gamma$ be a different basis, with $\Gamma \in \mathbb{R}^{d\times d}$ a unitary matrix. Then, $\tilde{y}_{ik} = \Gamma^T y_{ik}$, $\tilde{X}_i = \Gamma^T X_i$, and $||\tilde{y}_{ik} - \tilde{X}_i\beta_{ik}||^2 = ||y_{ik} - X_i\beta_{ik}||^2$ for any $\beta_{ik} \in \mathbb{R}^p$.

### 4.6 Computation

The first two steps of ManifoldLasso are construction of the neighborhood graph and estimation of the Laplacian $L$. As shown in Section 3, $L$ is a sparse matrix, hence RMetric can be run efficiently by only passing values corresponding to one neighborhood at a time. Therefore, the computational cost may be significantly reduced by fast nearest neighbor approximation based on random projections and trees (Bernhardsson, 2015). Note that in our examples and experiments, diffusion maps is our chosen embedding algorithm, so the neighborhoods and Laplacian are already available, though in general this is not the case. The second part of the algorithm estimates the gradients and constructs matrices $Y_{1:n}, X_{1:n}$. The gradient estimation runtime, with Cholesky decomposition-based solvers, is $O(qd^2 + nd^3)$ where $q = \sum_{i=1}^{n} k_i$ is the number of edges in the neighborhood graph. The last major step is a call to the GroupLasso solver, which estimates the support $S$ of $\phi$. The computation time of each iteration in GroupLasso is $O(nmpd)$. Note that when using a standard group lasso solver, the computation time is $O(n^2m^2pd)$ due to the block-diagonal structure of the problem implicit in flattening the $n$ by $p$ by $d$ covariate tensor. We therefore use our own implementation of accelerated proximal gradient descent to solve this problem (Boyd and Vandenberghe, 2004; Jas et al., 2020; Beck and Teboulle, 2009). Finally, for large data sets, we perform the 'for' loop over a subset $\mathcal{I} \subset [n]$ of the original data while retaining the geometric information from the full data set. This replaces the $n$ in the computation time with the smaller factor $n' = |\mathcal{I}|$.

### 4.7 Normalization

As with many sparse regression methods, normalization is necessary to balance the relative influence of dictionary elements and embeddings coordinates. Multiplying $g_j$ by a non-zero constant and dividing its corresponding $\beta_j$ by the same constant leaves the reconstruction error of all $y$'s invariant, but affects the norm $||\beta_j||$. Therefore, the relative scaling of the dictionary functions $g_j$ can influence the recovered support $S$, by favoring the dictionary functions whose columns have larger norm. A similar effect is present if a particular embedding coordinate $\phi_k$ is rescaled by a constant. For example, multiplying a certain $\phi_k$ by a number close to zero will cause the penalty accrued by learned coefficients for that coordinate to be smaller than for the other coefficients, and for that $\phi_k$ to dominate support recovery.

We therefore normalize all $\operatorname{grad}_{T_i^{\mathcal{M}}} \phi_{1:m}$ and $\operatorname{grad}_{T_i^{\mathcal{M}}} g_{1:p}$ as follows. Denote $f$ a function on $\mathcal{M}$, which can be either a coordinate function or a dictionary function. When $f$ is defined on $\mathcal{M}$, but not outside $\mathcal{M}$, we calculate the *normalizing constant*

$$\zeta^2 \;=\; \frac{1}{n}\sum_{i=1}^{n} \|\operatorname{grad}_{T_i^{\mathcal{M}}} f(\xi_i)||^2 \,, \tag{10}$$

then we set $f \leftarrow f/\zeta$. The above $\zeta$ is the finite sample version of $\|\operatorname{grad}_{\mathcal{T}} f\|_{L_2(\mathcal{M})}$, integrated w.r.t. the data density on $\mathcal{M}$. We apply this normalization to coordinate functions $\phi_k$, but it could also be applied to functions $g_j$ when they are defined only on $\mathcal{M}$. A similar approach was used in Haufe et al. (2009).

When function $f$ is defined on a neighborhood around $\mathcal{M}$ in $\mathbb{R}^D$, we compute the normalizing constant with respect to $\nabla_\xi f$. That is,

$$\zeta^2 \;=\; \frac{1}{n}\sum_{i=1}^{n} \|\nabla_\xi f(\xi_i)\|^2. \tag{11}$$

Then, once again, we set $f \leftarrow f/\zeta$. We apply this normalization to our dictionary functions $g_j$. This favors dictionary functions whose gradients are nearly tangent to the manifold $\mathcal{M}$, and penalizes the $g_j$'s which have large gradient components perpendicular to $\mathcal{M}$.

### 4.8 Tuning

Tuning parameters are often selected by cross-validation in Lasso-type problems. However, in our setting, the recovered support generally span the tangent space, and as discussed in Section 6, we are theoretically motivated to identify a size $d$ support. Since the cardinality of the support decreases as the tuning parameter $\lambda$ is increased, we base our choice of $\lambda$ on matching the cardinality of the support to $d$. Sufficient conditions for this estimation strategy are given in Section 6. To identify this $\lambda$, which we call $\lambda_0$, we perform a simple binary search over $\lambda$ in the range $[0, \lambda_{\max}]$ where $\lambda_{\max}$, the theoretical maximum $\lambda$ value, is $\lambda_{\max} = \max_j(\sum_{i=1}^n \sum_{k=1}^m (\operatorname{grad}_{T_i^{\mathcal{M}}} g_j(\xi_i))^T (\operatorname{grad}_{T_i^{\mathcal{M}}} \phi_m(\xi_i)))^{1/2}$.

### 4.9 Variants and Extensions

The ManifoldLasso algorithm presented here can be extended in several interesting ways. First, our current approach explains the embedding coordinates $\phi$ produced by a

particular embedding algorithm. However, the same approach can be used to directly explain the tangent subspace of $\mathcal{M}$, independently of any embedding. Second, one could set up GroupLasso problems that explain a single coordinate function. In general, manifold coordinates may not have individual meaning, so it will not always be possible to find a good explanation for a single $\phi_k$. However, Figure 1 shows that for the ethanol molecule, whose manifold is a torus, there exists a canonical association of certain coordinates to particular torsions. Third, one could apply the same group lasso machinery to gradients in the coordinates of the ambient space. Finally, metric properties may be used in order to distinguish between various valid explanations.

It is well-established in the support recovery and sparse coding literature (Chen et al., 1998; Hesterberg et al., 2008; Breheny and Huang, 2011; Lederer and Müller, 2015; Hastie and Tibshirani, 2015) that at large $\lambda$, shrinkage can cause problems including variable selection inconsistency; and furthermore that intermediate $\lambda$ values can have desirable properties as a variable pruning rather than selection step. Therefore, we also exhibit a combination of the group lasso formulation (9) with group sparse basis pursuit (Qu et al., 2018). The so-called basis pursuit problems (Chen et al., 1998), intimately related to regularized regression, are discussed in more detail in Appendix E. In the case of ManifoldLasso, the corresponding basis pursuit problem is

$$\arg\min_{\beta:s=d} \sum_{j=1}^{p} \|\beta_j\| s.t. \operatorname{grad}_{T_i^{\mathcal{M}}} \phi_k(\xi_i) = \sum_{j=1}^{p} \beta_{ijk} \operatorname{grad}_{T_i^{\mathcal{M}}} g_j(\xi_i) \quad \text{for all } i = 1:n, \text{ and } k = 1:m. \tag{12}$$

This problem is evidently not tractable, as it involves searching over all $d$-sets of dictionary functions. We suggest, following Hesterberg et al. (2008), to initially use an intermediate $\lambda$ values in ManifoldLasso in order to prune the dictionary to a smaller size. Subsequently, we solve problem (12) with the pruned dictionary.

## 5. Related Work

We draw a firm distinction between our approach and purely non-parametric methods that attempt to learn a parameterization of $\mathcal{M}$. For example, the early works of Saul and Roweis (2003) and Teh and Roweis (2002) (and references therein) propose parametrizing the manifold by finite mixtures of local linear models, aligned so as to provides global coordinates, in a way reminiscent of Local Tangent Space Alignment (Zhang and Zha, 2004). Another idea is to use $d$ eigenfunctions of the Laplace-Beltrami operator $\Delta_{\mathcal{M}}$ as a parametrization of $\mathcal{M}$. Hence, the Diffusion Maps coordinates could be considered such a parametrization (Coifman and Lafon, 2006; Coifman et al., 2005; Gear, 2012). However, these are not in and of themselves interpretable, and it is not clear how many such coordinates are needed (Chen and Meilă, 2019). In Mohammed and Narayanan (2017), it was shown that principal curves and surfaces can provide an approximate manifold parametrization. These methods can often be used as embedding algorithms in our approach, but make no attempts at synergizing with an interpretable dictionary. Dsilva et al. (2018) tackle the related problem of choosing among the infinitely many Laplacian eigenfunctions $d$ which provide a $d$-dimensional parametrization of the manifold. Their approach is to solve a sequence of Local Linear Embedding (Roweis and Saul, 2000) problems, each aiming to

represent an eigenfunction as a combination of the preceding ones. Similarly, Chen and Meilă (2019) is another method for reducing the number of "covarying" eigenfunctions. However, these methods fail to provide physical meaning for the selected functions.

Our work differs from the above entirely non-parametric methods in two key ways: (1) the explanations we obtain are endowed with the meaning of the domain specific dictionaries, (2) less obviously, descriptors like principal curves or Laplacian eigenfunctions are generally still non-parametric (i.e exist in infinite dimensional function spaces), while the parameterizations by dictionaries we obtain (e.g., the torsions) are in finite dimensional spaces. This distinction is mirrored in comparison with the many so-called *dictionary learning* methods in which a low-dimensional transformation is learned simultaneously with its inverse. We note that our method is not dictionary learning per se, but rather sparse coding, in which the dictionary is given (Szabó et al., 2011).

The *symbolic regression* methods of Brunton et al. (2016), Rudy et al. (2019), and Champion et al. (2019) for estimating governing laws of dynamical systems are perhaps most similar to this work. These methods use sparse regression with respect to a dictionary and the idea of differential composition. Their goal is to identify the functional equations of non-linear dynamical systems by regressing the time derivatives of the state variables on a subset of functions in the dictionary selected using a sparsity inducing penalty. This provides a natural interpretability. However, although these methods can loosely be considered univariate analogs, they do not consider the multidimensional data-manifold, and their synergies with dimension-reduction algorithms are developed in separate directions.

With respect to sparse regression, the seminal Group Lasso paper of Yuan and Lin (2006) and support recovery analyses of Elyaderani et al. (2017); Wainwright (2009) are central to our approach. However, our use of replicates in experiments is reminiscent of the Stability Selection method of Meinshausen and Bühlmann (2010). Such methods address instabilities of the variable selection, in particular, when restrictive theoretical conditions are violated (Zhao and Yu, 2006; Huan Xu et al., 2012). The empirically-based two-stage OLS-hybrid approach we elucidate in Appendix E for resolving this issue is based on ideas in Efron et al. (2004); Meinshausen (2007); Hesterberg et al. (2008). Some attractive alternate approaches to this problem that we do not pursue are the use of non-convex penalties such as SCAD (Fan and Li, 2001; Breheny and Huang, 2011) and weighted data points in the Adaptive Lasso (Zou, 2006). We note the method of Haufe et al. (2009), which applies group lasso to analyze sparse decomposition of vectors fields, albeit in a different setting.

As for our method, gradient estimation on manifolds is typically derived from the perspective of local linear regression and tangent space estimation (Mukherjee and Zhou, 2006; Aswani et al., 2011). However, as in Luo et al. (2009), we make explicit the logarithmic map by estimating and projecting upon the tangent space of $\phi(\mathcal{M})$, and our estimates of this tangent space are made using the pushforward metric of Perraul-Joncas and Meila (2013).

The role of our work with in the molecular dynamics literature is particularly relevant to enhanced sampling methods (Rohrdanz et al., 2011, 2013; Fiorin et al., 2013; Fleming et al., 2016). In these methods, exploration of the molecular state space is accelerated through biasing of simulation towards directions of large scale variation, which are typically identified through visual inspection. Note that this method is practically useful despite the need to perform initial simulations in order to identify collective coordinates. More recently,

reinforcement-learning type syntheses of these ideas have been applied (Wang et al., 2019; Pant et al., 2020; Sidky et al., 2020; Buenfil et al., 2021).

Although in this paper the dictionary consists of functions with physical meaning, our general principle of finding parametric geometrically-motivated approximations of learned representations is relevant to a range of machine learning contexts. Examining functions in embedding coordinates is quite typical in genomics (Amir et al., 2013), and much deep learning work also makes use of explicit traversal of a latent space (Lin et al., 2020; Shukla et al., 2018). It is also known in a range of settings that learned gradients provide interpretable (Adebayo et al., 2018) or otherwise statistically-useful information (Wu et al., 2010; Constantine et al., 2014; Yang, 2020). Our approach relies on the classical weighted local PCA method for tangent space estimation (Joncas et al., 2017; Aamari and Levrard, 2019). Improvement of this estimator in the presence of noise is an active area of research (Puchkin and Spokoiny, 2022).

## 6. Theoretical Results

We investigate the conditions under which $f = h \circ g_S$ can be represented over a dictionary $\mathcal{G}$ that contains $g_S$. Not surprisingly, we will show that these are *functional independency* conditions on the dictionary. Subsequently, we prove recovery conditions in the finite sample case.

### 6.1 Functional Dependency

We first study when a set of functions on an open subset $U \subset \mathbb{R}^d$ can be *almost smoothly* represented with a subset of functionally independent functions. The following lemma implies that if a set of non-full-rank smooth functions has a constant rank in a neighborhood, then locally we can choose a subset of these functions such that the other functions can be smoothly represented by them. This is a direct result from the constant rank theorem.

**Lemma 1 (Remark 2 after Zorich (2004) Theorem 2 in Section 8.6.2)** *Let $f : U \to \mathbb{R}^m$ be a mapping defined in an open neighborhood $U \subset \mathbb{R}^d$ of a point $x^\star \in \mathbb{R}^d$. Suppose $f \in C^\ell$, the rank of the mapping $f$ is $k$ at every point in $U$, and $k < m$. Moreover, assume that the principal minor of order $k$ of the matrix $Df$ is not zero at $x^\star$. Then in some neighborhood $U_{x^\star} \subset U$ there exist $m - k$ $C^\ell$ functions $g_i, i = k+1, \cdots, m$ such that for any $x = (x_1, \cdots, x_d) \in U(x^\star)$,*

$$f_i(x_1, x_2, \cdots, x_d) = g_i(f_1(x_1, x_2, \cdots, x_d), f_2(x_1, x_2, \cdots, x_d), \cdots, f_k(x_1, x_2, \cdots, x_d)). \quad (13)$$

Applying this lemma we can construct a local representation of a subset in $g_S$. Partitions of unity enable us to expand the above lemma from local to global. Mathematically, a *smooth partition of unity subordinate to* $\{U_\alpha\}$ is an indexed family $(\psi_\alpha)_{\alpha \in A}$ of smooth functions $\psi_\alpha : \mathcal{M} \to \mathbb{R}$ with the following properties:

(i) $0 \leq \psi_\alpha(\xi)$ for all $\alpha \in A$ and all $\xi \in \mathcal{M}$;

(ii) supp $\psi_\alpha \subset U_\alpha$ for each $\alpha \in A$;

(iii) Every $\xi \in \mathcal{M}$ has a neighborhood that intersects supp $\psi_\alpha$ for only finitely many values of $\alpha$;

(iv) $\sum_{\alpha \in A} \psi_\alpha(\xi) = 1$ for all $\xi \in \mathcal{M}$.

**Lemma 2 (Lee (2003) Theorem 2.23)** *Suppose that $\mathcal{M}$ is a smooth manifold, and $\{U_\alpha\}_{\alpha \in A}$ is any indexed open cover of $\mathcal{M}$. Then there exists a smooth partition of unity subordinate to $\{U_\alpha\}$.*

Now we state our main results.

**Theorem 3** *Assume $\mathcal{G} = \{g_i\}_{i=1}^p$ is the dictionary where $g_{1:p}$ are $C^\ell$ functions in an open set $U \subset \mathbb{R}^d$. For a subset $S \subset [p]$, let $g_S$ is defined as $\{g_i : i \in S\}$. Consider $S' \subset [p], S' \neq S, |S'| < d$ such that $\operatorname{rank} Dg_{S'} = |S'|$ at a point. Suppose that $\ell \geq d+1$. Then there exists a function $\tau : \mathbb{R}^{|S'|} \to \mathbb{R}^{|S|}$ that is almost everywhere $C^\ell$ on the range of $g_{S'}$, w.r.t. Lebesgue measure on $\mathbb{R}^{|S'|}$, such that $g_S = \tau \circ g_{S'}$ if*

$$\operatorname{rank} \begin{pmatrix} Dg_S \\ Dg_{S'} \end{pmatrix} = \operatorname{rank} Dg_{S'} \text{ on } ;, \tag{14}$$

*holds globally. If $\tau$ is smooth everywhere on the range of $g_{S'}$, then (14) holds globally.*

**Proof** First, we show the existence of $\tau$. We claim that it suffices to prove the existence of function composition on the set where $\operatorname{rank} Dg_{S'} = |S'|$. Consider $U = U_1 \cup U_2$, where $U_1 := \{x : \operatorname{rank} Dg_{S'} = |S'|\}$, and $U_2 = U - U_1$. $U_1$ is not empty by the assumption. Note that we can select an $|S'| \times |S'|$ submatrix $A_{S',\xi}$ in $Dg_{S'}$ and $\det A_{S',\xi}$ is a continuous function (and thus nonzero) in a neighborhood. This shows that $U_1$ is a nonempty open set. Locally, $g_{S'}$ is a diffeomorphism to its image; therefore $g_{S'}(U_1)$ contains an interior point, and thus has positive measure in $\mathbb{R}^{|S'|}$. From Sard's theorem (Lee, 2003), we know that the range of $g_{S'}(U_2)$ is of Lebesgue measure zero in $\mathbb{R}^{|S'|}$. Therefore it suffices to show that there exists a $\tau \in C^\ell$ on $g_{S'}(U_1)$. To simplify the notation we use $U$ to denote $U_1$ in the following proof. By definition of $U$, we know that $g_{S'}$ is a diffeomorphism between $U$ and $g_{S'}(U)$. So the inverse $g_{S'}^{-1}$ is well defined and $C^\ell$. Also denote $s = |S|$ and $s' = |S'|$. Let

$$g_{S' \sqcup S}(\xi) = \begin{pmatrix} g_{S'}(\xi) \\ g_S(\xi) \end{pmatrix},$$

and use $Dg_{S' \sqcup S}$ to denote the l.h.s. matrix in (14). Here $\sqcup$ means disjoint union. To be specific, we use $g_{j_i}$ to denote the $i-$th function in the collection $[g_{S'}; g_S]$. When the rank of $Dg_{S \sqcup S'}$ equals the rank of $D_{g_{S'}}$, Lemma 1 implies that there exists some neighborhood $U_x \in \mathbb{R}^d$ of $x$ and $C^\ell$ functions $\tau_x^i : \mathbb{R}^{|S'|} \to \mathbb{R}, i = s'+1, s'+2, \cdots, s'+s$ such that

$$g_{j_i}(\xi) = \tau_x^i(g_{j_1}(\xi), \cdots, g_{j_{s'}}(\xi)) = \tau_x^i(g_{S'}(\xi)), \text{ for } i = s'+1, s'+2, \cdots, s'+s, \ \xi \in U_x.$$

Here we should notice that $\tau_x^i$ is defined only on $g_{S'}(U_x)$. Since this holds for every $x \in U$, we can find an open cover $\{U_x\}$ of the original open set $U$. Since each open set in $\mathbb{R}^d$ is a manifold, the result of partition of unity in Lemma 2 holds, namely that $U$ admits a smooth partition of unity subordinate to the cover $\{U_x\}$. We denote this partition of unity by $\psi_x(\cdot)$.

Hence we can define

$$\tau^i(y) = \sum_{x \in U} \psi_x(g_{S'}^{-1}(y))\tau_x^i(y), \quad y \in g_{S'}(U).$$

where $\tau^i$ is a function mapping from $g_{S'}(U) \to \mathbb{R}$. For each fixed $x \in U$, the function $y \to \psi_x(g_{S'}^{-1}(y))\tau_x^i(y)$ for $y \in g_{S'}(U)$ is $C^\ell$. According to the properties of partition of unity, in a local neighborhood of each point, this is a summation of finitely many smooth functions. Then this $\tau^i$ will be a $C^\ell$ function on $g_{S'}(U)$. Also, by $1 = \sum_x \psi_x(\xi)$, it holds that $\tau^i(g_{S'}(\xi)) = g_{j_i}(\xi)$ for any $i = s'+1, \cdots, s'+s$.

Therefore, globally in $U$ we have

$$g_{S \sqcup S'}^i(\xi) = \tau^i(g_1(\xi), \cdots, g_{s'}(\xi)), \text{ for } i = s'+1, s'+2, \cdots, s'+s, \ \xi \in U.$$

Now we prove the reverse implication. If $\operatorname{rank} Dg_{S \sqcup S'} > \operatorname{rank} Dg_{S'}$, then there is $j \in S$, so that $Dg_j \notin \operatorname{rowspan} Dg_{S'}$. Pick $\xi^0 \in U$ such that $Dg_j(\xi^0) \neq 0$; such an $\xi^0$ must exist because otherwise it will be in $\operatorname{rowspan} Dg_{S'}$. By the theorem's assumption, $Dg_S = D\tau Dg_{S'}$. Therefore, $(Dg_S)^T$ is in $\operatorname{rowspan}(Dg_{S'})^T$ for any $\xi$. However, this is impossible at $\xi^0$. ■

This theorem essentially gives a condition for the existence of the explanation. Further, if $S$ is the set found by ManifoldLasso, then checking that there is no subset satisfying the rank condition implies that the explanation is unique in the dictionary. We say that a set of functions $g_S$ on a metric space $X$ is $C^\ell$ *(smooth) functionally dependent* at $\xi$ if there is a subset $S' \subset S, S' \neq S$, a function $\tau : \mathbb{R}^{|S'|} \to \mathbb{R}^{|S|}$ and a neighborhood $U$ around $\xi$ such that

(i) $g_S = \tau \circ g_{S'}$ on $U$;

(ii) $\tau$ is $C^\ell$ (smooth) globally on $g_{S'}(U) \subset \mathbb{R}^{|S'|}$;

(iii) $y - \tau(y_{S'}) \not\equiv 0$ on any neighborhood $O(g_S(\xi)) \subset \mathbb{R}^{|S|}$. Here $y = (y_1, \cdots, y_{|S|}) \in \mathbb{R}^{|S|}, y_{S'} = (y_i)_{i \in S'} \in \mathbb{R}^{|S'|}$.

The condition (iii) here eliminates the possibility of a trivial $\tau$. $S$ is *functionally independent* if it is nowhere functionally dependent. Based on Theorem 3, we formulate the rank condition below as a necessary and sufficient condition of functional independence.

**Corollary 4 (Functional Independence)** *Suppose $\mathcal{M}$ is a $d-$dimensional smooth manifold and $g_S : \mathcal{M} \to \mathbb{R}^d$ are $d$ $C^\ell$ functions. Suppose $g_S(\mathcal{M})$ has a positive measure in $\mathbb{R}^d$. Then they are functionally independent on $\mathcal{M}$ iff* $\operatorname{rank} Dg_S(\xi)$ *is $d$ everywhere on $\mathcal{M}$ except for a closed subset $W \subset \mathcal{M}$ with no interior point.*

**Proof** First we show that the rank condition implies functional independence. Suppose $g_S$ is functionally dependent. Then by definition we have that $g_S = \tau \circ g_{S'}$ on a neighborhood for some $S'$ with $|S'| < |S| = d$. Then on this neighborhood $\operatorname{rank} Dg_S \leq \operatorname{rank} Dg_{S'} \leq d-1$. This contradicts the assumption. On the other hand, suppose $Dg_S$ is functionally independent. We claim that for any $\xi' \in V_0$, $\operatorname{rank} Dg_S(\xi') \geq \operatorname{rank} Dg_S(\xi)$: For any $\xi$ there is a $\operatorname{rank} Dg_S \times \operatorname{rank} Dg_S$ non-degenerate submatrix of $Dg_S(\xi)$ whose determinant is non-zero. Therefore, by the smoothness of $g_S$, there exists a neighborhood $V_0$ such that this submatrix of the Jacobian is invertible in this neighborhood and the claim holds.

We therefore start from a point $\xi$ where $\operatorname{rank} Dg_S = d-1$. Select functions that are full rank at this point, and denote them by $g_{S'}$. There is a neighborhood $V_1$ of $\xi$ with $\operatorname{rank} g_{S'}(\xi) = d-1$. If $\operatorname{rank} Dg_S = d-1$ holds on some neighborhood $V_2$ of $\xi$, then after selecting a chart $(U, \varphi)$ containing $V_1 \cap V_0$, we have that

$$\operatorname{rank} \begin{pmatrix} Dg_S \circ \varphi^{-1} \\ Dg_{S'} \circ \varphi^{-1} \end{pmatrix} = \operatorname{rank} Dg_{S'} \circ \varphi^{-1}$$

holds on $V_2 \cap V_1 \cap V_0$. Thus, Theorem 3 implies that $g_S$ cannot be functionally independent (consider a composition with $\varphi$). Therefore, the set $\{x : \operatorname{rank} Dg_S = d-1\}$ has empty interior in $\mathcal{M}$. Similarly, in every neighborhood $V_k$ of any point where $0 \leq \operatorname{rank} Dg_S = k < d-1$, there must be a point such that $\operatorname{rank} Dg_S = k+1$. Then in every neighborhood $V_{k+1} \cap V_k$ of this new point there must be a point such that $\operatorname{rank} Dg_S = k+2$. By induction, there must be a point in $V_k$ such that $\operatorname{rank} Dg_S = d$. Therefore we conclude that the set $W = \{\xi : \operatorname{rank} Dg_S \leq d-1\}$ contains no interior point. Also, it is closed because $\{\xi : \operatorname{rank} Dg_S = d\}$ is open. ■

**Theorem 5** *Let $\mathcal{G}$ and $g_S$ be defined as before. $\mathcal{M}$ is a smooth manifold with dimension $d$ embedded in $\mathbb{R}^D$. Suppose that $\psi : \mathcal{M} \subset \mathbb{R}^D \to \mathbb{R}^m$ is also an embedding of $\mathcal{M}$ and has a decomposition $\psi(\xi) = h \circ g_S(\xi)$ for every $\xi \in \mathcal{M}$ where $h$ is smooth. If the dictionary $g_S$ contains $d$ functions denoted by $g_{S'}$, that are smooth functionally independent on $\mathcal{M}$, then there exists a $\widetilde{h}$ such that $\psi = \widetilde{h} \circ g_{S'}$ on every $\xi \in \mathcal{M}$. Here, the function $\widetilde{h}$ is smooth almost everywhere in the range of $g_{S'}$.*

**Proof** Consider the set $U = \{x : \operatorname{rank} Dg_{S'} = d\}$. It is an open subset of the manifold $\mathcal{M}$ and therefore a smooth manifold. For each point $x \in U$, select a local chart $(V, \varphi)$ such that $V \subset U$. With the same argument in the proof of Corollary 4, we know that there exists a smooth functions $\tau_x$ on $V$ such that $g_S = \tau \circ g_{S'}$ holds on V. Also, since $V$ is an open neighborhood, we conclude that the measure of $g_{S'}(U) \geq g_{S'}(V)$ should be strictly positive. Therefore the partition of unity technique used in the proof of Theorem 3 can show that there exists a function $\tau$ on $U$ that is smooth over $g_{S'}(U)$ such that $g_S = \tau \circ g_{S'}$ holds globally on $U$. We can define $\tau$ on $\mathcal{M} \setminus U$ to be anything, and Sard's theorem implies that $g_{S'}(U)$ would be a measure zero set in $\mathbb{R}^d$. Finally, we just write $\widetilde{h} = h \circ \tau$ ■

The assumptions of these theorems are reasonable. Even though any smooth map $f : \mathcal{M} \to \mathbb{R}^d$ on a compact manifold $\mathcal{M}$ must have at least one singular point, Theorem 3 will still hold almost everywhere as long as $g_S$ is smooth almost everywhere. The existence of such functions $g_S$ with rank $d$ almost everywhere is guaranteed by the fact that a single coordinate chart can cover any compact manifold except for a set of measure zero, known as the cut-locus of the chart (Sheng, 2009; Bishop, 2013). One can, for example, find one function explaining the whole circle $S^1$ embedded in $\mathbb{R}^2$ except one point. Thus, these theoretical results should be considered conditions on the dictionary, rather than the manifold itself.

### 6.2 Discussion of Practical Recovery from Samples

In a finite sample setting, Theorem 3 states that $S$ and $S'$ are equivalent explanations for $f$ whenever (14) holds on open sets around the sample points. In this situation, it is very

likely to find many subsets $S' \subset [p]$ of cardinality $d$ that are full rank in neighborhoods of all data points. Still assuming that all gradients are exact, for all such $S'$ the first term of $J_\lambda(\beta)$ in (9) will be zero; in other words there will be many equivalent explanations of $\phi$ in $\mathcal{G}$. However, one can define a subset $S$ as the expected minimizer in (12), or in a purely oracle sense. For our theoretical analyses, we thus fix a subset $S$ and regard it as the true support throughout this section.

The tendency of ManifoldLasso to select a support with a low value of $\sum_{j=1}^{p} \|\beta_j\|$ is reasonable, and even desirable because, according to Obozinski et al. (2011), a particular group $S$ will be recovered by Group Lasso methods, if (i) it is close to perpendicular to the linear subspace generated by all other groups, and (ii) group features in $S$ are close to orthogonal matrix. The first condition will be discussed later in this section. As for condition (ii), we note that if a set $S'$ is not full rank on $\mathcal{M}$, the Jacobian $Dg_{S'}$ will be ill-conditioned at the data near the critical points, which will result in very large $\beta_{ji}$ values. Hence, such a subset will be heavily penalized. Moreover, features $g_j$ which vary much in a direction normal to $\mathcal{M}$ will have, due to the gradient normalization, smaller values for $\mathrm{grad}_{T_i^{\mathcal{M}}} g_j$; therefore their $\beta_j$ coefficients will be large relatively to the coefficients of functions that vary within $\mathcal{M}$.

We now analyze the situation when $\mathrm{grad}_{T_i^{\mathcal{M}}} g_j$ and $\mathrm{grad}_{T_i^{\mathcal{M}}} \phi_k$ are estimated with noise, showing that it is qualitatively similar to noiseless case. Specifically, we provide recovery guarantees for the (GroupLasso) problem that highlight the influence of the aforementioned factors, as well as of condition (i). The guarantees are deteriministic, but they depend on the noise sample variance, hence they can lead to statistical guarantees holding w.h.p. in the usual way. For simplicity, we analyze support recovery for $m = 1$, hence for a single dependent variable $y$. Namely we assume that that the data $y_{1:n} \in \mathbb{R}^d$ satisfy

$$y_i \;=\; \sum_{j=1}^{p} \beta_{ij}^* x_{ij} + \epsilon_i \quad \text{for } i = 1:n\,, \tag{15}$$

and we rewrite the GroupLasso problem as

$$\min_{\beta} \tfrac{1}{2} \sum_{i=1}^{n} ||y_i - X_i\beta_i||_2^2 + \lambda\sqrt{n} \sum_{j=1}^{p} ||\beta_j||, \tag{16}$$

according to the notations of equation (9) with the index $k$ dropped. A first theorem deals with support recovery, proving that all coefficients outside the support are zeroed out, under conditions that depend only on the dictionary, the fixed support $S$, and the noise. The second result completes the previous with error bounds on the estimates $\hat{\beta}_j$, assuming an additional condition on the magnitude of the true $\beta_j$ coefficients. Since these coefficients are partial derivatives w.r.t. the dictionary functions, the condition implies that the dependence on each function must be strong enough to allow the accurate estimation of the partial derivatives.

We introduce the following quantities. The *incoherence* of $\mathcal{G}$ is defined as

$$\mu \;=\; \max_{i=1:n, j\in[p], j'\in S, j\neq j'} \frac{|x_{ij}^T x_{ij'}|}{\|x_{ij}\|\|x_{ij'}\|}.$$

This definition differs in two ways from the standard definition of incoherence as $\max_i \max_{j,j' \in [p]} |x_{ij}^T x_{ij'}|$. First, here we do not make the common assumption that the columns of the GroupLasso design matrix are norm 1 (details to be found in the proof of Theorem 7). Rather, the recovery result we pursue assumes the normalizations in Section 4.7; hence to preserve a measure of incoherence independent of the column norms, we must rescale by $\|x_{ij}\|\|x_{ij'}\|$. Second, because we condition on the set $S$, it is not necessary to require that the gradients outside the support $S$ be incoherent.

We further consider the internal collinearity of the support $S$ as follows. Let

$$\Sigma_i \;=\; \left[x_{ij}^T x_{ij'}\right]_{j,j' \in S} \quad \text{and} \quad \Sigma \;=\; \operatorname{diag}\{\Sigma_{1:n}\}.$$

**Lemma 6**

$$\|\Sigma_i^{-1}\| \;\leq\; \frac{1}{(\min_{j \in S} \|x_{ij}\|^2)[1-(s-1)\mu]} \quad \textit{for all } i = 1:n.$$

**Proof** It is easy to see that

$$\Sigma_i \;=\; \operatorname{diag}\{\|x_{ij}\|_{j \in S}\} \tilde{\Sigma}_i \operatorname{diag}\{\|x_{ij}\|_{j \in S}\} \quad \text{with} \quad \tilde{\Sigma}_i \;=\; \left[\frac{x_{ij}^T x_{ij'}}{\|x_{ij}\|\|x_{ij'}\|}\right]_{i,j \in S}.$$

By the Gershgorin Theorem, since all the off-diagonal elements of $\tilde{\Sigma}_i$ are bounded in absolute value by $\mu$, the minimum eigenvalue of $\tilde{\Sigma}_i$ is bounded below by $1-(s-1)\mu$. When this quantity is positive, then the maximum eigenvalue of $\tilde{\Sigma}_i^{-1}$ is

$$\|\tilde{\Sigma}_i^{-1}\| \;\leq\; \frac{1}{1-(s-1)\mu} \;=\; \nu.$$

A smaller $\nu$ means that the $x_{ij}$ gradients are closer to being orthogonal at each datapoint $i$. Furthermore, $\|\Sigma_i^{-1}\| \leq \|\tilde{\Sigma}_i^{-1}\| \|\operatorname{diag}\{\|x_{ij}\|_{j \in S}^{-1}\}\|^2 \leq \frac{\nu}{\min_{j \in S} \|x_{ij}\|^2}$. ■

Finally, the noise level $\sigma$ is defined by

$$\max_{i=1:n} ||\epsilon_i||^2 = d\sigma^2.$$

**Theorem 7 (Support recovery)** *Assume that equation (15) holds, and that* $\sum_{i=1}^n ||x_{ij}||^2 = \gamma_j^2$ *for all* $j = 1:p$. *Let* $\gamma_{\max} = \max_{j \notin S} \gamma_j$, $\kappa_S = \max_{i=1:n} \frac{\max_{j \in S} \|x_{ij}\|}{\min_{j \in S} \|x_{ij}\|}$. *Denote by* $\bar{\beta}$ *the solution of (16) for some* $\lambda > 0$. *If* $1-(s-1)\mu > 0$ *and*

$$\gamma_{\max}\left(\frac{\mu}{1-(s-1)\mu} \frac{\kappa_S}{\min_{i=1}^n \min_{j' \in S} \|x_{ij'}\|} + \frac{\sigma\sqrt{d}}{\lambda\sqrt{n}}\right) \;\leq ;, \tag{17}$$

*then* $\bar{\beta}_{ij} = 0$ *for* $j \notin S$ *and all* $i = 1, \ldots n$.

**Proof** We structure equation (15) in the form

$$y \;=\; \bar{\bar{X}}\bar{\beta}^* + \bar{\epsilon} \quad \text{with } y = [y_i]_{i=1:n} \in \mathbb{R}^{nd},\; \bar{\beta} = [\beta_i]_{i=1:n} \in \mathbb{R}^{np},$$

$\tilde{X}_{ij} \in \mathbb{R}^{nd}$ is obtained from $x_{ij}$ by padding with zeros for the entries not in the $i$-th segment, $\bar{\bar{X}} = [[\tilde{X}_{ij}]_{j=1:p}]_{i=1:n} \in \mathbb{R}^{nd\times np}$, and $\bar{\bar{X}}_j = [\tilde{X}_{ij}]_{i=1:n} \in \mathbb{R}^{nd\times n}$ collects the columns of $\bar{\bar{X}}$ that correspond to the $j$-th dictionary entry. Note that

$$\tilde{X}_{ij}^T\tilde{X}_{ij'} \;=\; x_{ij}^T x_{ij'} \quad \text{and} \quad \tilde{X}_{ij}^T\tilde{X}_{i'j'} \;=\; 0 \text{ whenever } i \neq i'.$$

The proof is by the *primal dual witness* method, following Elyaderani et al. (2017); Obozinski et al. (2011). It can be shown Elyaderani et al. (2017); Wainwright (2009) that $\bar{\beta}$ is a solution to (GroupLasso) iff, for all $j = 1:p$,

$$\bar{\bar{X}}_j^T\bar{\bar{X}}(\bar{\beta} - \bar{\beta}^*) - \bar{\bar{X}}_j^T\bar{\epsilon} + \lambda z_j \;=\; 0 \;\in \mathbb{R}^n \text{ with } z_j = \frac{\beta_j}{||\beta_j||} \text{ if } \beta_j \neq 0 \text{ and } ||z_j|| < 1 \text{ otherwise.} \tag{18}$$

The matrix $\bar{\bar{X}}_j^T\bar{\bar{X}}$ is a diagonal matrix with $n$ blocks of size $1 \times p$, hence the first term in (18) becomes

$$[x_{ij}^T X_i(\bar{\beta}_{i:} - \bar{\beta}^*_{i:})]_{i=1:n} \in \mathbb{R}^n.$$

Similarly $\bar{\bar{X}}_j^T\bar{\epsilon} = [x_{ij}^T\epsilon_i]_{i=1:n} \in \mathbb{R}^n$.

We now consider the solution $\hat{\beta}$ to problem (16) under the additional constraint that $\beta_{ij'} = 0$ for $j' \notin S$. In other words, $\hat{\beta}$ is the solution we would obtain if $S$ was known. Let $\hat{z}$ be the optimal dual variable for this problem, and let $\hat{z}_S = [\hat{z}_j]_{j\in S}$.

We will now complete $\hat{z}_S$ to a $z \in \mathbb{R}^{np}$ so that the pair $(\hat{\beta}, z)$ satisfies (18). If we succeed, then we will have proved that $\hat{\beta}$ is the solution to the original GroupLasso problem, and in particular that the support of $\hat{\beta}$ is included in $S$. For simplicity we denote $\lambda' = \lambda\sqrt{n}$.

From (18) we obtain values for $z_j$ when $j \notin S$.

$$z_j \;=\; \frac{-1}{\lambda'}\bar{\bar{X}}_j^T\left[\bar{\bar{X}}^T(\hat{\beta} - \bar{\beta}^*) - \bar{\epsilon}\right]. \tag{19}$$

At the same time, if we consider all $j \in S$, we obtain from (18) that $\bar{\bar{X}}_S = [\bar{\bar{X}}_j]_{j\in S}$ (here the vectors $\beta_S, \beta_S*$ and all other vectors are size $ns$, with entries sorted by $j$, then by $i$).

$$\bar{\bar{X}}_S^T\bar{\bar{X}}_S(\hat{\beta}_S - \beta_S^*) - \bar{\bar{X}}_S^T\bar{\epsilon} + \lambda'\hat{z}_S \;=\; 0. \tag{20}$$

Solving for $\hat{\beta}_S - \beta_S^*$ in (20), we obtain

$$\hat{\beta}_S - \beta_S^* \;=\; (\bar{\bar{X}}_S^T\bar{\bar{X}}_S)^{-1}\left(\bar{\bar{X}}_S^T\bar{\epsilon} - \lambda'\hat{z}_S\right) \;=\; \Sigma^{-1}\left(\bar{\bar{X}}_S^T\bar{\epsilon} - \lambda'\hat{z}_S\right).$$

After replacing the above in (19) we have

$$z_j \;=\; \frac{-1}{\lambda'}\bar{\bar{X}}_j^T\left[\bar{\bar{X}}_S\Sigma^{-1}\bar{\bar{X}}_S^T w - \bar{\bar{X}}_S\Sigma^{-1}\lambda'\hat{z}_S - \bar{\epsilon}\right] \;=\; \bar{\bar{X}}_j^T\bar{\bar{X}}_S\Sigma^{-1}\hat{z}_S + \frac{1}{\lambda'}\bar{\bar{X}}_j^T(I - \bar{\bar{X}}_S\Sigma^{-1}\bar{\bar{X}}_S^T)\bar{\epsilon}.$$

Finally, by noting that $\Pi = I - \bar{\bar{X}}_S\Sigma^{-1}\bar{\bar{X}}_S^T$ is the projection operator on the subspace $\text{span}(\bar{\bar{X}}_S)^\perp$, we obtain that

$$z_j \;=\; (\bar{\bar{X}}_j^T\bar{\bar{X}}_S)\Sigma^{-1}\hat{z}_S + \frac{1}{\lambda'}\bar{\bar{X}}_j^T\Pi\bar{\epsilon}, \quad \text{for } j \notin S. \tag{21}$$

We must show that $||z_j|| < 1$ for $j \notin S$. To bound the first term, we note that $\bar{\bar{X}}_j^T \bar{\bar{X}}_S$ is $n \times ns$, block diagonal, with blocks of size $1 \times s$, and with all non-zero entries bounded in absolute value by $\mu$. Hence, for any vector $v = [v_i]_{i=1:n} \in \mathbb{R}^{ns}$,

$$||\bar{\bar{X}}_j^T \bar{\bar{X}}_S v||^2 = ||[(x_{ij}^T x_{iS}) v_i]_{i=1:n}||^2 \leq \sum_{i=1}^n ||(x_{ij}^T x_{iS}) v_i||^2 \leq \sum_{i=1}^n \|(x_{ij}^T x_{iS})\|^2 \|v_i\|^2. \tag{22}$$

In our case $v_i = \Sigma_i^{-1} \hat{z}_{iS}$, hence by by Lemma 6

$$\|v_i\| \leq ||\Sigma_i^{-1}|| ||\hat{z}_{iS}|| \leq \nu \frac{1}{\min_{j' \in S} \|x_{ij'}\|^2}. \tag{23}$$

On the other hand,

$$\|x_{ij}^T x_{iS}\|^2 = \sum_{j' \in S} (x_{ij}^T x_{ij'})^2 \leq \sum_{j' \in S} \mu \|x_{ij}\|^2 \|x_{ij'}\|^2 \leq \mu \max_{j' \in S} (\|x_{ij'}\|^2) \|x_{ij}\|^2. \tag{24}$$

Bounds (23) and (24) together with equation (22) yield

$$\begin{aligned}
(\bar{\bar{X}}_j^T \bar{\bar{X}}_S) \Sigma^{-1} \hat{z}_S &\leq n u^2 \mu^2 \sum_{i=1}^n \|x_{ij}\|^2 \max_{j' \in S} (\|x_{ij'}\|^2) \frac{1}{(\min_{j' \in S} \|x_{ij'}\|^2)^2} \\
&\leq \nu^2 \mu^2 \kappa_S^2 \sum_{i=1}^n \|x_{ij}\|^2 \frac{1}{\min_{j' \in S} \|x_{ij'}\|^2} \\
&\leq \nu^2 \mu^2 \kappa_S^2 \frac{1}{\min_{i=1}^n \min_{j' \in S} \|x_{ij'}\|^2} \sum_{i=1}^n \|x_{ij}\|^2 \\
&= \nu^2 \mu^2 \kappa_S^2 \frac{1}{\min_{i=1}^n \min_{j' \in S} \|x_{ij'}\|^2} \gamma_j^2.
\end{aligned} \tag{25}$$

To bound the second term, we note that $\Pi$ is a block diagonal matrix, $\Pi = \text{diag}\{\Pi_{1:n}\}$, with $\Pi_i = I_d - x_{iS}^T \Sigma_i^{-1} x_{iS}$. Hence, the norm squared of this term is bounded above by $\sum_{i=1}^n ||\Pi_i \epsilon_i||^2 ||x_{ij}||^2 / (\lambda')^2 \leq \sum_{i=1}^n ||\epsilon_i||^2 ||x_{ij}||^2 / (\lambda')^2 \leq d\sigma^2 / (\lambda')^2 \sum_{i=1}^n ||x_{ij}||^2 = (d\sigma^2 / (\lambda')^2) \gamma_j^2$.

Replacing these bounds in (21) we obtain that

$$||z_j|| \leq ||\bar{\bar{X}}_j^T \bar{\bar{X}}_S \Sigma^{-1} \hat{z}_S|| + ||\frac{1}{\lambda'} \bar{\bar{X}}_j^T \Pi \bar{\epsilon}|| \leq \left( \frac{\mu \nu \kappa_S}{\min_{i=1}^n \min_{j' \in S} \|x_{ij'}\|} + \frac{\sigma \sqrt{d}}{\lambda'} \right) \gamma_j \text{ for any } j \notin S. \tag{26}$$

■

The first term inside the parenthesis relates to the properties of the support $S$. The factor $\frac{\mu}{1-(s-1)\mu}$ measures the near-orthogonality of the gradients in $S$, while the factors $(\min_{i=1}^n \min_{j' \in S} \|x_{ij'}\|)^{-1}$ and $\kappa_S$ measure the conditioning of $S$ with respect to the gradient norms. They are optimal when all gradients in $S$ are bounded away from 0, and when their sizes are relatively equal. The second term depends on the noise amplitude, and can be made arbitrarily small by increasing the regularization coefficient $\lambda$.

We now consider the recovery of all the non-zero coefficients, which will complete the exact support recovery proof. From the result below, we shall see that having non-zero $\hat{\beta}_j$

for a $j \in S$ requires that the original $\beta_j$ is large enough w.r.t. noise level and condition numbers of the problem. This conflicts with the requirement that $\beta_j$ is small, suggesting one possible way that the GroupLasso recovery may fail, namely that the smallest of the non-zero $\beta_j$'s may be "regularized out" before all the nuisance $\hat{\beta}_j$ are.

**Corollary 8** *Assume that equation* (16) *and condition* (17) *hold. Let* $\kappa = \frac{\mu}{1-(s-1)\mu} \frac{\kappa_S}{\min_{i=1}^n \min_{j' \in S} \|x_{ij'}\|}$ *and* $\gamma_S = \|\bar{\bar{X}}_S\|$. *Denote by* $\hat{\beta}$ *the solution to problem* (16) *for some* $\lambda > 0$. *If (1)* $\lambda = c\frac{\gamma_{\max}\sigma\sqrt{d}}{1-\kappa\gamma_{\max}}$, $c > 1$, *and (2)* $||\beta_j^*|| > \sigma\sqrt{d}(\gamma_{\max} + \gamma_S) + \lambda(1+\sqrt{s})$ *for all* $j \in S$, *then the support* $S$ *is recovered exactly and*

$$||\hat{\beta}_j - \beta_j^*|| < \sigma\sqrt{d}(\gamma_{\max}+\gamma_S)+\lambda(1+\sqrt{s}) = \sigma\sqrt{d}\gamma_{\max}\left[1 + \gamma_S/\gamma_{\max} + c\frac{1+\sqrt{s}}{1-\kappa\gamma_{\max}}\right] \quad \text{for all } j \in S. \tag{27}$$

**Proof of Corollary 8** According to Theorem 7, $\hat{\beta}_j = 0$ for $j \notin S$. It remains to prove the error bound for $j \in S$. According to Lemma V.2 of Elyaderani et al. (2017), for any $j \in S$,

$$\begin{aligned} ||\hat{\beta}_j - \beta_j^*|| &\leq ||\bar{\bar{X}}_j^T\bar{\epsilon}|| + ||\bar{\bar{X}}_S^T\bar{\epsilon}|| + \lambda(1+\sqrt{s}) \\ &\leq (||\bar{\bar{X}}_j|| + ||\bar{\bar{X}}_S||)||\bar{\epsilon}|| + \lambda(1+\sqrt{s}) \\ &\leq \sigma\sqrt{d}(\gamma_{\max}+\gamma_S) + \lambda(1+\sqrt{s}) \\ &= \sigma\sqrt{d}\gamma_{\max}\left[1 + \gamma_S/\gamma_{\max} + c\frac{1+\sqrt{s}}{1-\kappa\gamma_{\max}}\right]. \end{aligned}$$

Hence, if $||\beta_j^*||$ is greater than the r.h.s. of the above, $\hat{\beta}_j \neq 0$ and the support is recovered exactly. ■

In equation (27), the factor $\sigma\sqrt{d}$ represents the noise amplitude, while $\gamma_{\max}$ bounds the amplitude of the nuisance covariates $\bar{\bar{X}}_j$ outside of $S$. A smaller $\gamma_{\max}$ means that the contribution of these nuisance covariates will be smaller. The term $\gamma_S$ bounds the collinearity of the noise with the true support covariates. The last term measures the bias introduced in $\hat{\beta}_j$ by the regularization; note that $\lambda$ itself depends on the noise amplitude $\sigma\sqrt{d}$.

Recall from Sections 4.5 and 4.7 that $\gamma_j$ represents the finite sample estimate of the $L_2$ norm of $\text{grad}_{T_i^{\mathcal{M}}} g_j$. When the dictionary functions $g_j$ are defined on $\mathcal{M}$, but not outside $\mathcal{M}$, then $\text{grad}_{T_i^{\mathcal{M}}} g_j$ is normalized by equation (10) and consequently $\gamma_j = \sqrt{n}$ for all $j$. If first the gradients $\nabla_\xi g_j$ are computed and then normalized in ambient space $\mathbb{R}^D$ by (11) after projection on the tangent bundle $\mathcal{TM}$, then $\gamma_j \leq \sqrt{n}$. Thus, by explicitly considering the variability in the norms of $\|x_{ij}\|$ for $j \notin S$, we see that features $g_j$ whose gradient $\nabla_\xi g_j$ is not tangent to the manifold are easier to rule out. Regarding scaling of the l.h.s. of equation (17), it is easy to see that the term $\frac{\sigma\sqrt{d}\gamma_{\max}}{\lambda\sqrt{n}}$ is $\mathcal{O}(1)$ w.r.t. $n$; the first term $\kappa\gamma_{\max}$ is invariant to any rescaling of the $\bar{\bar{X}}$ by a scalar.

## 7. Experiments

We demonstrate the ability of ManifoldLasso to identify explanations of manifolds and their embedding coordinates in both toy and scientific manifold learning problems. Section

7.1 describes the general experimental procedure, while Section 7.2 describes some specific adjustments to this protocol necessary for analyzing molecular dynamics (MD) data. Sections 7.3.1–7.4 describe our experimental results. [2]

| Dataset | $n$ | $N_a$ | $D$ | $d$ | $\epsilon_N$ | $m$ | $n'$ | $p$ | $\omega$ |
|---|---|---|---|---|---|---|---|---|---|
| **SwissRoll** | 10000 | NA | 49 | 2 | .18 | 2 | 100 | 51 | 1 |
| **RigidEthanol** | 10000 | 9 | 50 | 2 | 3.5 | 3 | 100 | 12 | 25 |
| **Ethanol** | 50000 | 9 | 50 | 2 | 3.5 | 3 | 100 | 12 | 25 |
| **Malonaldehyde** | 50000 | 9 | 50 | 2 | 3.5 | 3 | 100 | 12 | 25 |
| **Toluene** | 50000 | 16 | 50 | 1 | 1.9 | 2 | 100 | 30 | 25 |
| **Ethanol** | 50000 | 9 | 50 | 2 | 3.5 | 3 | 100 | 756 | 25 |
| **Malonaldehyde** | 50000 | 9 | 50 | 2 | 3.5 | 3 | 100 | 756 | 25 |

Table 1: Summary of experiments. **SwissRoll** and **RigidEthanol** are toy data, while **Toluene**, **Ethanol**, and **Malonaldehyde** are from quantum molecular dynamics simulations by Chmiela et al. (2017). The columns list the following experimental parameters: $n$ is the sample size for manifold embedding, $N_a$ is the number of atoms in the molecule, $D$ is the dimension of $\xi$, $d$ is the intrinsic dimension, $\epsilon_N$ is the kernel bandwidth, $m$ is the embedding dimension, $n'$ is the size of the subsample used for ManifoldLasso, $p$ is the dictionary size, and $\omega$ is the number of independent repetitions of ManifoldLasso. More details are in Section 7.1

### 7.1 Experimental Setup

For all of the following experiments, the data consist of $n$ data points in $D$ dimensions, as well an embedding $\phi_{1:m}(\mathcal{D})$. We assume access to the manifold dimension $d$, a kernel bandwidth $\epsilon_N$ used in the estimation of the tangent spaces, and $p$ dictionary functions. Except where otherwise specified, $m$ and $\epsilon_M$ are used in the preliminary step of generating embeddings $\phi_{1:m}$ using the diffusion maps algorithm as EmbeddingAlg. ManifoldLasso is applied to a uniformly random subset of size $n' = |\mathcal{I}|$ and this process is repeated $\omega$ number of times. These parameters are passed to the Laplacian, LocalPCA, RMetric, and PullBackDPhi algorithms, and are summarized in Table 1. The regularization parameter $\lambda$ ranges over $[0, \lambda_{\max}]$ as described in Section 4.8.

### 7.2 Molecular Dynamics Data

The method of MD simulations is one of the principal tools in the study of molecular systems. Such simulations provide detailed information on the fluctuations and conformational changes of the system, and are now routinely used to investigate the structure, dynamics and thermodynamics of biological macromolecules and their complexes. In such simulations, the positions of atoms within a molecule are sampled as they proceed through time from some initial conditions according to interatomic effects. The distribution of this sample describes the molecule's behavior in the given experimental conditions. It has been shown empirically that manifolds approximate these high-dimensional distributions (Dsilva et al.,

2. Code to run experiments is available at `https://github.com/sjkoelle/montlake`.

2013). Accordingly, application of manifold learning to find the collective coordinates has achieved great success (Das et al., 2006; Tribello et al., 2012; Noé and Clementi, 2017; Sidky et al., 2020). Even though the vector of atomic coordinates can take any value, due to interatomic interactions, the relative positions of atoms within the molecule lie near a low-dimensional *slow manifold*. Performing manifold learning on these data separates the conformational changes, modeled by the manifold, from the fluctuations represented by the "noise" around the manifold.

### 7.2.1 Representing molecular configurations

Our MD data are quantum-simulations from Chmiela et al. (2017). The raw data consists of $X, Y, Z$ coordinates for each of the $N_a$ atoms of the chosen molecule. For a single observation, we denote these by $r_i \in \mathbb{R}^{3N_a}$. The first step in our data analysis pipeline is to featurize the configuration in a way that is invariant to rotation and translation. In the present experiments, we follow Chen et al. (2019) and represent a molecular configuration as a vector $a_i \in \mathbb{R}^{3\binom{N_a}{3}}$ of the *planar angles* formed by triplets of atoms. We then perform an SVD on this featurization, and project the data onto the top $D = 50$ singular vectors to remove linear redundancies; we denote the new data points by $\xi_{1:n}$. The EmbeddingAlg and LocalPCA algorithms work directly with $\xi$ in dimension $D$. Other possible representations such as applying a Procrustes transform to each configuration to align it with the first one give similar results, and no matter which low level representation we choose, large-scale conformational changes are described by the relative rotations of groups of atoms - the bond torsions illustrated in Figure 1 (Chen et al., 2019).

### 7.2.2 Dictionaries for MD Data

Therefore, in the **RigidEthanol**, **Ethanol**, **Malonaldehyde**, and **Toluene** MD datasets, we construct dictionaries consisting of bond *torsions*. We then apply ManifoldLasso to select combinations of these higher-level torsion features that explain the manifold in the lower-level planar angle feature space. Given an ordered 4-tuple of atoms $ABCD$, the torsion $g_{ABCD}$ is the angle of the planes defined by the locations of $ABC$ and $BCD$. Note that $g_{ABCD} \equiv g_{DBCA} \equiv g_{DCBA} \equiv g_{ACBD}$. Any torsion $g$ is expressible in closed form as functions of the planar angles feature vector $a$. In particular, a torsion $g_{ABCD}$ is a function of the angles of the triangles inscribing atoms $ABC$, $ABD$, $ACD$, and $BCD$. We compute the gradients of the torsions by automatic differentiation (Paszke et al., 2019).

One cannot use the obtained gradients directly in ManifoldLasso, since the angular features overparameterize the molecular *shape space* $\Sigma_3^{N_a}$ (Addicoat and Collins, 2010; Kendall, 1989) of dimension $D' = 3N_a - 7$, and off-manifold gradients are therefore not well-defined. For example, whether one chooses to use triangles $ABC$, $ABD$, and $ACD$, or $ABC$, $ABD$, and $BCD$ to compute $g_{ABCD}$ has no effect on the value of $g_{ABCD}$, but changes the value of its partial derivatives in $\mathbb{R}^D$. We therefore project the gradients on the tangent bundle of the shape space as it is embedded in $\mathbb{R}^D$. Details are given in Appendix B. It is on these gradients that we perform the normalization as described in Section 4.6. Remaining specifics of our MD data analytics pipeline are in Appendix C.

### 7.3 Synthetic Data Results

As a prelude to real MD data, we demonstrate the workings of ManifoldLasso in controlled settings by applying it to the well known **SwissRoll** dataset, as well as a simple non-dynamical simulation of a rigidly-rotating ethanol molecule.

#### 7.3.1 ManifoldLasso on **SwissRoll**

We use the **SwissRoll** dataset to demonstrate that ManifoldLasso is invariant to the choice of embedding algorithm. This consists of points sampled from a two dimensional rectangle and rolled up along one of the two axes, then randomly rotated in $D = 49$ dimensions. The dictionary $\mathcal{G}$ consists of $\{g_1, g_2\}$, the two rectilinear coordinates, as well as $g_{j+2} = \xi_j$, for $j = 1, \dots 49$, the coordinates of the feature space. We learn the manifold using three techniques: local tangent space alignment, diffusion maps, and isomap, shown in Figures 3c, 3e and 3g. For comparison, we also analyze the "trivial embedding" $\phi_1^{Internal} = g_1$, $\phi_2^{Internal} = g_2$ (Figure 3a). These rectilinear coordinates are colored in red and blue, and show clear associations with the other embedding coordinates.

Applying ManifoldLasso to the embeddings identifies the set $S = \{g_1, g_2\}$ as the manifold explanation, and identifies the association of the recovered support with individual embedding coordinates $\phi_{1:2}$. By visual inspection of Figures 3a, 3c, 3e, and 3g, we see that all embedding algorithms recover the original manifold, although the embeddings $\phi^{\text{Iso}}, \phi^{DM}, \dots$ are not isometric (this is particularly noticeable with diffusion maps), and sign changes are possible. Figures 3b, 3d, 3f and 3h demonstrate that ManifoldLasso recovers the two manifold-specific coordinate functions in each case, while the coefficients $\beta_{3:51}$ decay rapidly to 0 with $\lambda$. Furthermore, each of $g_1$ and $g_2$ is always mapped to the correct embedding coordinate. The regularization paths are virtually identical for all embeddings, even though the embeddings are not isometric.

#### 7.3.2 ManifoldLasso on a Rigid Ethanol Skeleton

We validate our analytics pipeline for MD data by analyzing a rigid simulation of the ethanol molecule. We construct an ethanol skeleton composed of the atoms shown in Figure 4a. We then sample as we rotate the atoms around the C-C and C-O bonds. In contrast with the MD trajectories, which are simulated according to quantum dynamics, these two angles are distributed uniformly over a grid. We call the resultant dataset **RigidEthanol**. As expected given our two a priori known degrees of freedom, Figures 4b and 4c show that the estimated manifold is a two-dimensional surface with a torus topology parameterized by bond torsions $g_1$ and $g_2$ similar to that observed for the MD **Ethanol** in Figure 1.

The dictionary consists of the twelve torsions implicitly defined by the bond diagram[3] in Figure 4a. All of these torsions circumscribe one of the central C-C and C-O bonds. Counting permutations of peripheral hydrogens, we can see that there are 9 of the former, and 3 of the latter, which we denote $g_{1,1:9}$ and $g_{2,1:3}$ in Figure 4d. Hence, any pair $\{g_{1,l}, g_{2,l'}\}$ with $l \in \{1:9\}, l' \in \{1:3\}$ is a correct coordinate system for this manifold. This is shown in Figure 4d by the incoherences $\mu_{jj'}$, i.e., mean pairwise cosines of the dictionary

3. These are all 4-tuples of atoms connected by a path in the figure, modulo the natural equivalence relation on torsions previously described.

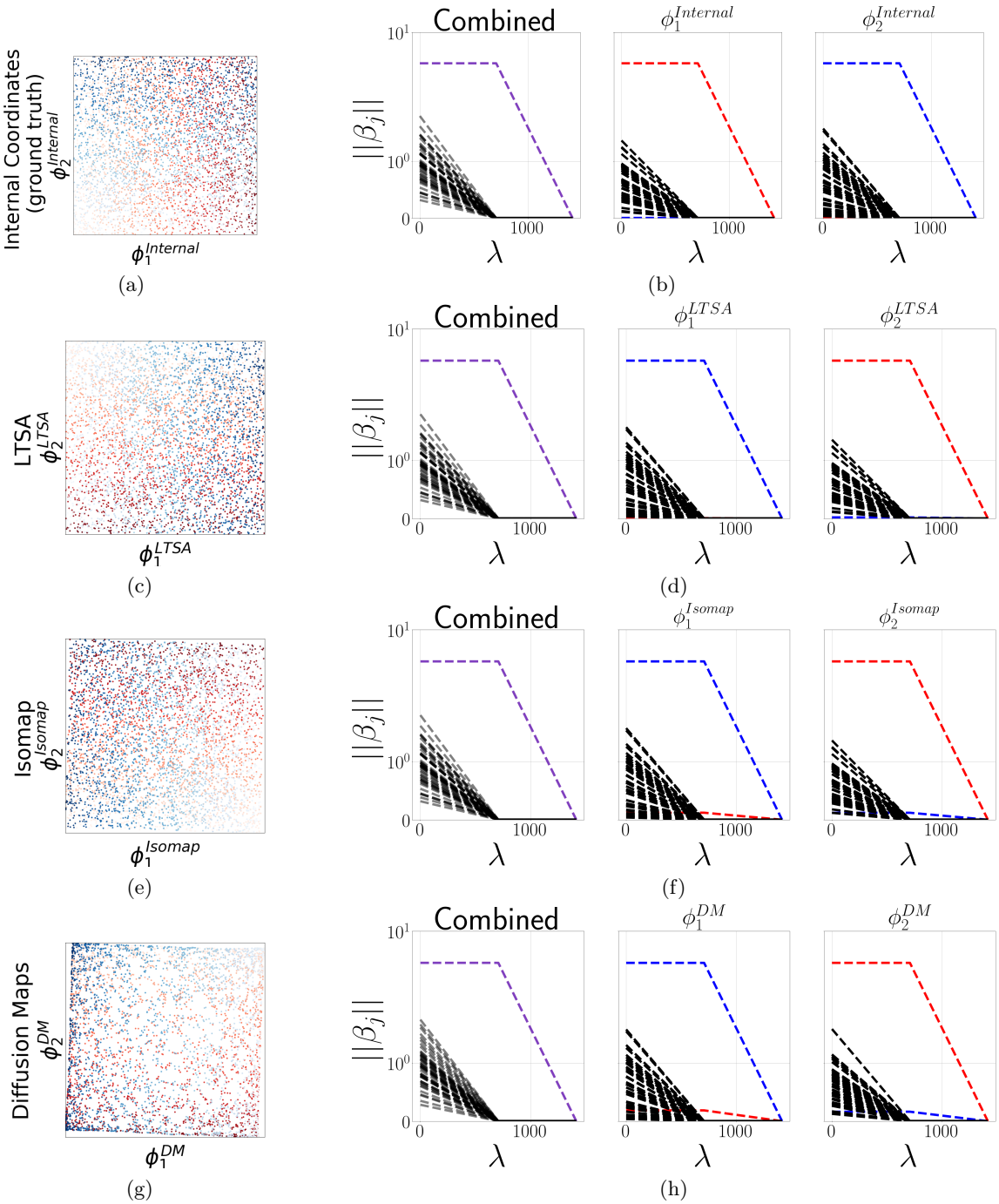
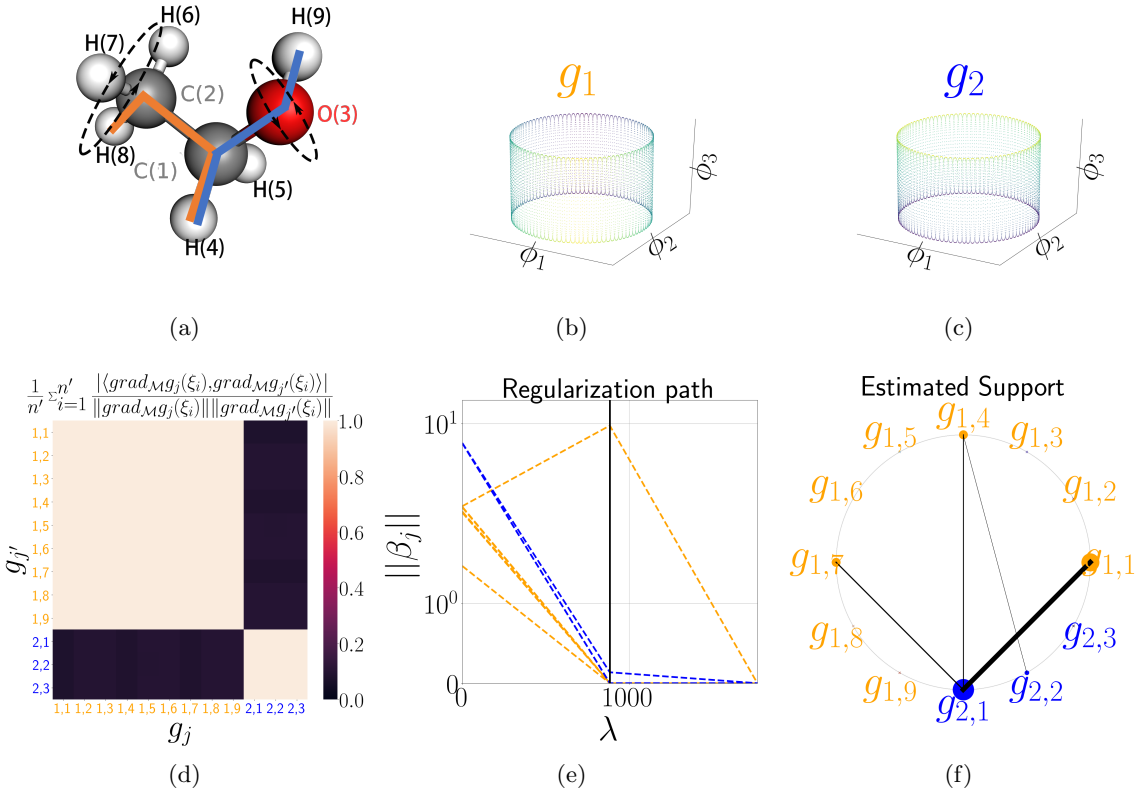



Figure 3: Results for **SwissRoll** embedded using a variety of manifold learning algorithms. Figure 3a shows the data mapped w.r.t. the edges of the rectangle colored by $g_1$ in red and $g_2$ in blue. Figures 3c, 3e, and 3g display embeddings of **SwissRoll** generated by several different manifold learning methods, colored by the rectilinear coordinates in red and blue. Figures 3b, 3d, 3f, and 3h display the regularization paths of ManifoldLasso for these embeddings. The combined norms $\|\beta_j\|$ used in ManifoldLasso are given on the left, and the norms for the individual embedding coordinates $\|\beta_{jk}\|$ on the right.




(a) (b) (c)

(d) (e) (f)

Figure 4: Results of ManifoldLasso for **RigidEthanol**. Figure 4a shows the simplified dynamics of our rigid molecular simulation. Atoms in the rigid ethanol skeleton are articulated around the C-O and C-C bonds by a torus of rotations. Figure 4b shows the learned torus, colored by C-C torsion $g_1$ from Figure 1. Figure 4c shows the same torus, colored by the C-O torsion $g_2$ from Figure 1. Figure 4d displays the incoherences, i.e., pairwise collinearities of dictionary gradients; C-C torsions functionally dependent on $g_1$ are in orange, C-O torsions functionally dependent on $g_2$ are in blue. Figure 4e shows combined regularization paths $\|\beta_j\|$ vs. $\lambda$ for a single replicate. The tuning parameter at which $|S| = d$ is indicated by the vertical black line. The chord diagram in Figure 4f represents the frequency of selecting each pair of torsions in replicate experiments. The frequencies with which individual torsions are selected are given by the sizes of the perimeter dots corresponding to each dictionary element, while the frequencies with which pairs of torsions are selected are given by the line widths connecting the dots.

functions, where the index $j$ ranges over the 12 dictionary functions. Comparing the row and column labels of Figure 4d with Figure 4a shows that the collinearities of these gradients clearly cluster by central bond. Thus, we expect ManifoldLasso to recover one torsion from each group. Indeed, in the regularization path of an individual replicate of ManifoldLasso shown in Figure 4e, collinear torsions are killed off, and a representative torsion is selected from each group. Finally, Figure 4f shows that ManifoldLasso selects such orthogonal pairs in 25 out of 25 random replicates of the $n'$ points.

### 7.4 Molecular Dynamics Results

In the same manner, we use ManifoldLasso to identify torsions that govern the dynamics of the molecules in Figure 1. From the machine learning point of view, MD data from well-studied molecules are an excellent testbed: the manifold hypothesis is believed to hold approximately, there is sufficient data to learn a manifold, and the ground truth is available and can validate our algorithms. Moreover, MD data are challenging problems for manifold learning. Appendix D displays **Toluene**, **Malonaldehyde**, and **Ethanol** in the $\xi$ representation, showing high amplitude noise outside the manifold; indeed, MD data has multiscale structure and the "noise" is non-uniformly distributed and highly correlated in the $\mathbb{R}^D$ space. Since the gradients of the dictionary functions are calculated analytically from the $\xi$ coordinates, and the data does not lie exactly on $\mathcal{M}$, the values of $\text{grad}_{T_i^{\mathcal{M}}} g_j$ will necessarily be noisy as well. From a scientific point of view, high quality MD data are expensive to generate, taking weeks or months of supercomputer time (Bowers et al., 2006; Fiorin et al., 2013). Fast automated analysis of these data by identification of collective coordinates serves both in the scientific understanding of the data and in acceleration sampling methods (Rohrdanz et al., 2013). Moreover, every new simulation represents a new manifold, and a new manifold explanation problem.

We first show that ManifoldLasso can distinguish groups that correspond to the chemical bonds in Figure 1, as would typically be done by a scientist using prior domain knowledge. Next, we repeat the analysis with no prior knowledge. That is, we include all distinct 4-tuples of atoms in the dictionary, even those which are not implicitly defined by the bond diagrams.

#### 7.4.1 Dictionaries Based on Bond Diagrams

Bond diagrams such as the ones in Figure 1 are based on a priori information about molecular structure garnered from historical work. Building a dictionary based on this structure is akin to many other methods in the field (Krenn et al., 2020; Xie et al., 2019). As in the case of **RigidEthanol**, our dictionaries consist of all equivalence classes of 4-tuples of atoms implicitly defined by bond diagrams, and the incoherence plots for **Ethanol** and **Malonaldehyde** in Figures 5a and 5d show two groups of highly dependent torsions, corresponding to the two bonds between heavy atoms in the molecules. We have labelled these by their central bond. For example, $g_1$ of ethanol is described by 9 functionally dependent torsions, since each central carbon has three peripheral atoms, while $g_2$ of ethanol is described by only 3 functionally dependent torsions, since, by the diagram, the oxygen atom only has one peripheral atom. Therefore, success means recovering a pair of incoherent torsions out of these dictionaries. For **Toluene**, the manifold dimension is $d = 1$ and success means recovering one of the 6 torsions associated with the peripheral methyl group bond. For this molecule, there are also $p - 6 = 24$ torsions that do not explain the data manifold. These correspond to bonds within the main benzene ring. We apply ManifoldLasso with these dictionaries to the embeddings shown in Figure 1.

As Figure 5 shows, ManifoldLasso is always able to identify torsions corresponding to the expected labelled bonds. Figures 5b, 5e, and 5g show combined regularization paths for single replicates of ManifoldLasso, and Figures 5c, 5f and 5h show frequencies of support recovery of sets of size $d$ over $w = 25$ replicates. ManifoldLasso finds that the

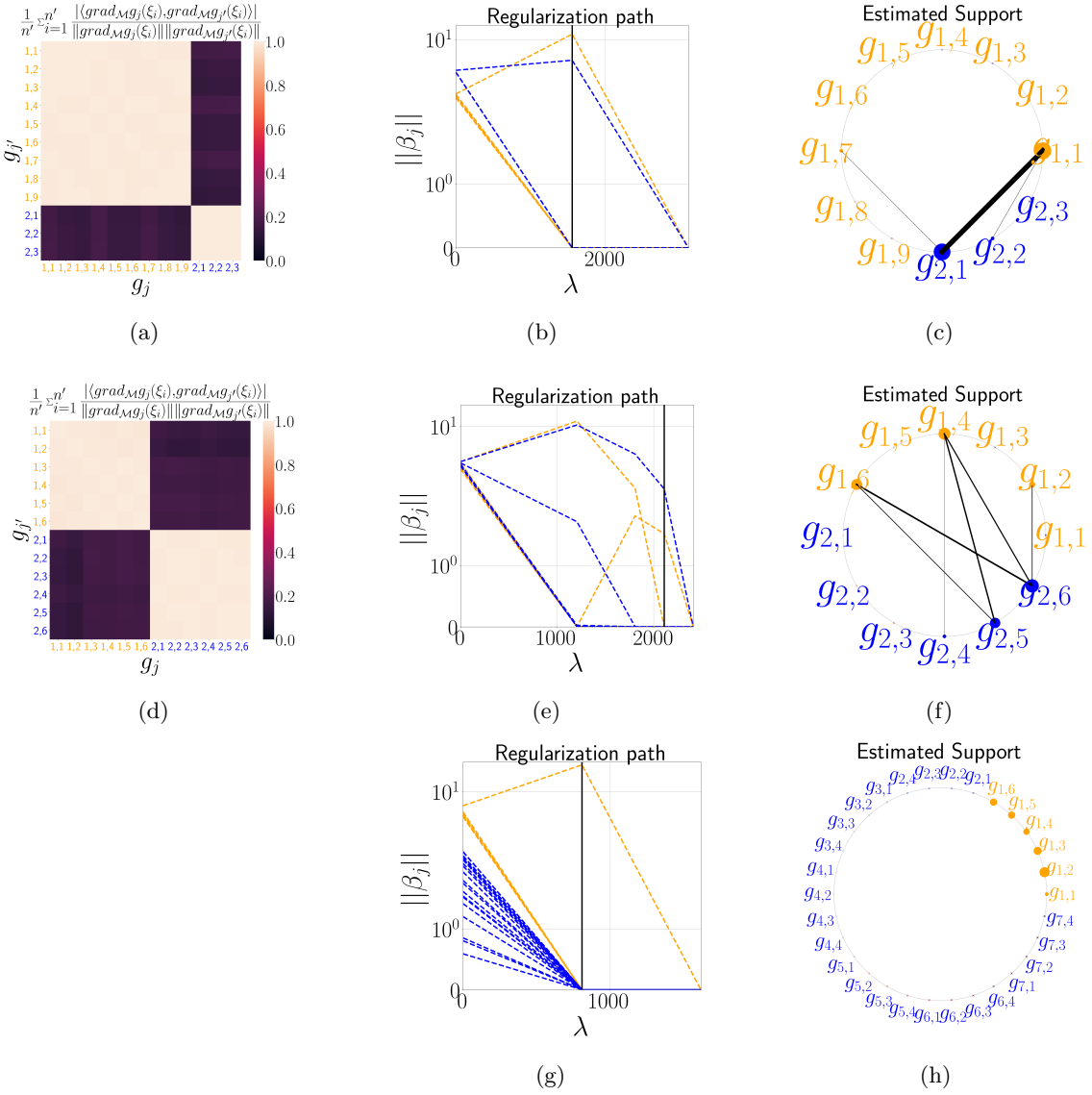



Figure 5: Results for MD data with a priori dictionaries given by the bond diagrams in Figure 1. The three rows correspond to **Ethanol**, **Malonaldehyde**, and **Toluene**, respectively. Figures 5a and 5d display pairwise collinearities of dictionary gradients, colored by bond as in Figure 1. Toluene, a $1-d$ manifold, has trivial cosines, and so these are not shown. Figures 5b, 5e, and 5g show combined regularization paths of $\|\beta_j\|$ for single replicates. Vertical black lines indicate the tuning parameter at which $|S| = d$. Figures 5c, 5f, and 5h show chord diagrams displaying frequency of support recovery of sets of size $d$ for 25 replicates. As for **RigidEthanol**, two-dimensional support recovery frequency is denoted by chord width, and one-dimensional support recovery frequency is denoted by size of perimeter dot. Note that blue in toluene corresponds to torsions in the benzene ring.

toroidal **Ethanol** manifold is explained by pairs of torsions from the C-O and C-C bonds, while **Malonaldehyde** is explained by one of each of the two central bonds. **Toluene** is

explained by the torsion of the peripheral methyl group. These agree with our domain-expert validated parameterizations from Figure 1. Torsion association with individual embedding coordinates is examined in Appendix F.

For all quantum MD experiments, we examine the support recovery condition Theorem 7. We first note that Figures 5a and 5d show that even without foreknowledge of a unique true support, the incoherence parameter $\mu$ must be quite close to 1, since it is a maximum over set of cosines whose mean is plotted. The empirical distributions of the parameters of this Theorem across replicates are listed in Appendix F. The high values of the incoherence parameter $\mu$ and otherwise unfavorable empirical support recovery parameters listed in the table indicate that we cannot expect a unique recovery. However, ManifoldLasso is still successful in obtaining representative torsions from the desired bonds in Figure 1. The similarity between these results on real data with challenging noise and variable sampling density to the result on the synthetic **RigidEthanol** show the robustness of the ManifoldLasso method.

### 7.4.2 Results from Full Dictionary

We test ManifoldLasso in the extreme case when the dictionary consists off all possible torsions, i.e., all $\binom{N_a}{4}$ 4-tuples modulo equivalence. For **Ethanol** and **Malonaldehyde** we obtain $p = 756$ torsions[4]. Such a large $p$ is challenging for $l_1$ regularized estimation, due to the bias mentioned Section 4.9 for large $\lambda$. Moreover, Figures 6a and 6e show that, besides the two main groups of collinear torsions, roughly a quarter of the 756 are coherent with both groups. While we do not necessarily expect ManifoldLasso to succeed, or to be used in such a way in practice, this experiment tests the robustness of ManifoldLasso to a situation that is challenging for sparsity inducing regularization.

The results of ManifoldLasso with the full dictionary for **Ethanol** and **Malonaldehyde** are displayed in Figure 6. For consistency between replications, we choose *a priori* the ground truth to be represented by torsions $g_{21,35}$ and $g_{75,351}$, which are representative torsions of $g_1$ and $g_2$ for **Ethanol**, respectively for **Malonaldehyde**, depicted in Figure 1. We can evaluate the selected $d = 2$ functions for coherence with this ground truth. ManifoldLasso identifies supports with mean incoherences with the true support of $0.66 \pm .54$ and $.85 \pm .4$ for **Ethanol** and for **Malonaldehyde**, respectively. Note that when both selected functions are more coherent with a single element of the true support, we use the pairwise coherences with higher mean. This is visually apparent from comparing selected torsions in Figures 6c and 6g with their collinearities in Figures 6d and 6h since the latter figures also show collinearities of the selected supports with the example functions from the true support. We can see that the selected support functions are often strongly coherent with the ground truth functions, regardless of the orthogonality of the selected support, although for **Ethanol** in particular, both selected support functions are often strongly coherent with the same the ground truth function. Thus, we can see that ManifoldLasso performs preferably on **Malonaldehyde**.

The results are visualized in Appendix F, which shows the embeddings colored by the selected torsions. There is a visual correspondence between coherences between torsions

4. We do not analyze **Toluene**, because for $d = 1$ the solution is available analytically, making this example somewhat trivial.

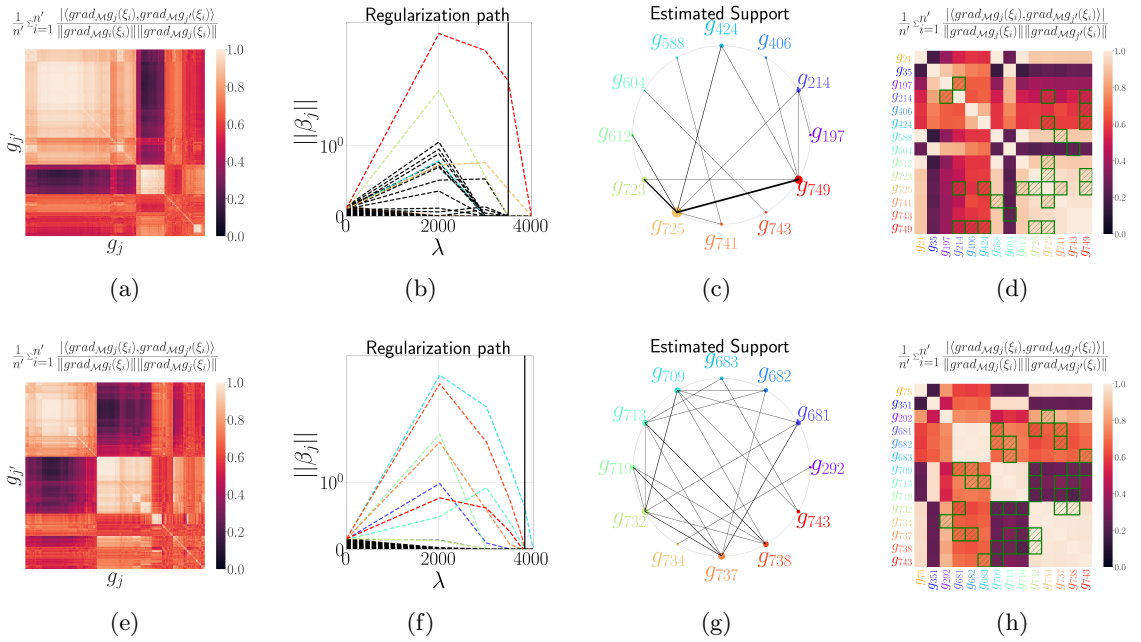



Figure 6: Results for MD data with full dictionaries consisting of all possible torsions. The top and bottom rows show results for **Ethanol** and **Malonaldehyde**, respectively. Figures 6a and 6e show mean cosine collinearity of dictionary gradients ordered by heirarchical clustering. Figures 6b and 6f show examples of combined regularization paths for single replicates that select relatively orthogonal functions. The tuning parameter at which $|S| = d$ is indicated by the vertical black lines. Functions are colored if they are selected in any replicate. Figure 6c and 6g shows support recoveries given by ManifoldLasso over different replicates. Figure 6d and 6h and shows mean cosine collinearity of selected supports. $g_{21,35}$ and $g_{75,351}$ are representative torsions from the true support, while the others are selected in any replicate. Pairs that are selected in any replicate are marked with a green box.

and their colorings of the manifolds learned from **Ethanol** and **Malonaldehyde**. For example, torsions $g_{588}$ and $g_{604}$ of ethanol are orthogonal, while $g_{588}$ and $g_{612}$ are collinear. When orthogonal pairs are selected, we capture information that would otherwise need to be discovered by visual inspection of the embeddings. Moreover, the **Malonaldehyde** plots also demonstrate that even for this simple manifold, associating manifold coordinates to dictionary functions by visual inspection is delicate work. From a chemistry perspective, orthogonal recovered torsions generally flank pairs of hydrogens of which each is attached to one of the central atoms in the putatively true bonds. Thus, it makes sense that these peripheral torsions could geometrically describe the same motion as the putative true support. We also note that certain selected torsions do not appear to parameterize the manifold. In these cases, the functional maximizer used to compute $\lambda_{\max}$ in Section 4.8 is predominantly driven by single outlying data points. Thus, such functions are infrequently selected in replicates.

Examination of the selected regularization paths in Figures 6b and 6f shows that a small number of unselected functions persist quite far into the regularization path. Thus, when

ManifoldLasso fails to select orthogonal functions, for example due to the documented support recovery instability and bias at high values of $\lambda$ for Lasso methods in general (Meinshausen, 2007; Hesterberg et al., 2008; Huan Xu et al., 2012), we propose a two-stage variable selection procedure in which a secondary variable selection step is applied after initial pruning, as in Hesterberg et al. (2008). As described in Appendix E, in the first stage we heuristically choose $\lambda = \lambda_{max}/2$, which eliminates most dictionary functions. For example, in the plotted replicate, the number $p'$ of dictionary elements selected at $\lambda_{max}/2$ is about 15 for **Ethanol** and 8 for **Malonaldehyde**. In the second stage we perform an exhaustive search over the remaining dictionary to optimize (12). This simple variation recovers an explanation with functions that are highly coherent with the ground truth in all cases - $.97 \pm .2$ for **Ethanol** and $.96 \pm .08$ for **Malonaldehyde**, and we avoid selecting pairs of functions that are collinear with the same element of the true support. Inspection of the colored embeddings for **Ethanol** in Appendix F also confirms that these conform to our visual intuition of orthogonally varying torsions. These experiments show that on noisy large $p$ problems with massive violations of the incoherence conditions, ManifoldLasso, while sometimes not successful on its own, can robustly prune the dictionary.

## 8. Conclusion

The approach of ManifoldLasso is to reconstruct the differentials of the manifold coordinates from differentials of functional covariates. It is robust to non-linearity in both the algorithm and the covariates. It requires functions that are smooth, as well as the assumption that the data lie near a smooth manifold. We estimate the differentials of the manifold embedding algorithm, but use differentials of functional covariates that are available analytically. We demonstrate this approach on molecular dynamics simulations that generate high-dimensional point clouds sampled from the configuration space of a given molecule, and for which our functional covariates are bond torsions and the embedding coordinates display a denoised version of the data. Together, these examples demonstrate the efficacy of ManifoldLasso for automated explanation of data manifolds. Its limitations are consistent with the general behavior of l1-regularized methods, and circumventing them with existing tools appears promising.

Both linear and non-linear dimension reduction methods map data to abstract coordinates, derived from agnostic, intrinsic data properties, such as the covariance matrix, in the case of PCA, or the Laplacian, in the case of the diffusion maps algorithm. By regressing the abstract coordinate functions on a dictionary $\mathcal{G}$ of functions of the data that have meaning in the domain of the problem, we automatically establish relationships between the learned manifold and domain knowledge. The expert is freed from the tedious work of visually inspecting each possible function $g_j$ with the manifold coordinates; her expertise is used by specifying covariate functions of the data. The recovered results come with guarantees which can be partially checked in practice. With the obvious simplifications, ManifoldLasso could also be used to assign explanations to coordinates obtained by PCA.

Variations of the methods and results presented here could solve a variety of related problems. For example, suppose that two different experiments produce data sets in the same ambient space $\mathbb{R}^D$ and that, from these, we learn manifolds $\mathcal{M}_1$ and $\mathcal{M}_2$ which are both 2-tori. Explaining $\mathcal{M}_{1,2}$ with a dictionary $\mathcal{G}$ can tell us if the manifolds are "the same"

from the physics point of view. Moreover, one can seek common or overlapping explanations from a single dictionary for different data sources. In other words, by explaining manifolds estimated purely from data with domain-dependent dictionaries, we produce transferable knowledge, that does not depend on the particularities of the sample, or embedding algorithm, and that can be communicated between experts in the language of their domain.

## Acknowledgments

This work was partially supported by NSF DMS 2015272, NSF DMS 1810975, NSF DMS PD 08-1269, U.S. Department of Energy, Solar Energy Technology Office award DE-EE0008563, NSF IGERT 1258485, the Moore-Sloan Foundation, the UW eScience Institute, and a Simons Fellowship to MM from the Institute for Pure and Applied Mathematics (IPAM). Part of this work was conducted while the authors Y-CC, SK and MM were participants in the IPAM long program on Learning for Physics and the Physics of Learning. The authors thank Jim Pfaendtner, Stefan Chmiela, Alexandre Tkatchenko, and the Tkatchenko group for providing both data and expertise.

## Appendix A. Approximating the Logarithmic Map by Orthogonal Projection

In this appendix, we illustrate the details of the approximation to logarithmic map by orthogonal projection in Section 4.4. We assume that $\mathcal{M}$ is a submanifold isometrically embedded in $\mathbb{R}^D$. Also $\mathcal{M}$ is assumed to be at least $C^4$ and compact and the Riemannian metric $\mathbf{g}$ of $\mathcal{M}$ is also $C^4$. The function $\phi$ is also assumed to be at least $C^3$.

Let $\gamma(s)$ be the geodesic pass through a point $\xi$ at $s = 0$ and a different point $\xi'$ in $\mathcal{M}$ for some $s > 0$, where $s$ is the arc length parameter of the geodesic. Then, the *logarithmic map* (do Carmo, 1992) of $\xi'$ w.r.t. to $\xi$ is defined as the vector $\log_\xi \xi' := s\gamma'(0) \in \mathcal{T}_\xi\mathcal{M}$.

**Proposition 9** *For all $\xi$ not on the boundary of $\mathcal{M}$ and all $\xi'$ such that $\|\xi' - \xi\| \leq r$ for some $r > 0$, it holds that*

$$\|\mathrm{Proj}_{\mathcal{T}_\xi\mathcal{M}}(\xi' - \xi) - \log_\xi \xi'\| = o(r), \quad \|\mathrm{Proj}_{\mathcal{T}_{\phi(\xi)}\phi(\mathcal{M})}(\phi(\xi') - \phi(\xi)) - \log_{\phi(\xi)} \phi(\xi')\| = o(r) \ .$$

**Proof** This proposition follows the results in Appendix B in Coifman and Lafon (2006). First, from the assumption it follows that the geodesic $\gamma \in C^3$. This is because when the manifold is at least $C^4$ and the Riemannian metric is also $C^4$, the Christoffel symbol as smooth functions of the Riemannian metric will be at least $C^3$, Hence the solution of geodesic equation will be $C^3$ according to standard ODE theory.

Therefore by Taylor expansion, $\gamma(s) = \gamma(0) + s\gamma'(0) + \frac{s^2}{2}\gamma^{(2)}(0) + \frac{\gamma^{(3)}(\tilde{s})}{6}s^3$, where $\tilde{s} \in (0, s)$. Recall that $\gamma^{(2)}$ is a vector orthogonal to $\mathcal{T}_\xi\mathcal{M}$ for a geodesic; moreover, when the manifold $\mathcal{M}$ is $C^3$ and compact, the magnitude of $\gamma^{(3)}$ is uniformly bounded on $\mathcal{M}$. Denote $l = \log_\xi \xi'$; also, denote by $u$ the orthogonal projection $\mathrm{Proj}_{\mathcal{T}_\xi\mathcal{M}}(\xi' - \xi)$. Note that $\gamma(0) = \xi, \gamma(s) = \xi'$, so $\xi' - \xi = l + O(s^3)$, i.e., $l = \xi' - \xi + O(s^3)$.

Lemma 7 in Coifman and Lafon (2006) implies $\|u\|^2 = s^2 + O(r^4)$. Therefore, $s^2 = \|u\|^2 + O(r^4) \leq \|\xi - \xi'\|^2 + O(r^4) \leq r^2 + O(r^4) = O(r^2)$. Hence, $s^3 = \mathcal{O}(r^3)$. Now consider

$l, u$ and $\xi' - \xi$ as points in $\mathbb{R}^D$. We have that $\|\xi' - \xi - u\| \leq \|\xi' - \xi - l\|$, and by triangle inequality, $\|l - u\| \leq \|\xi' - \xi - u\| + \|\xi' - \xi - l\| \leq 2\|\xi' - \xi - l\| = o(r)$. Hence, we have shown the first part of the desired result.

Now we turn to $\text{Proj}_{\mathcal{T}_{\phi(\xi)}}(\phi(\xi') - \phi(\xi))$. In the pushforward Riemannian metric $G$, $\phi(\gamma)$ is the geodesic between $\phi(\xi)$ and $\phi(\xi')$ in $\phi(\mathcal{M})$. When $\mathcal{M}$ is compact, and $\phi \in C^3(\mathcal{M})$, then $\phi$ and the derivatives $\phi'_{1:m}$ are uniformly continuous, hence the derivatives of $\phi(\gamma)$ remain bounded by the derivatives of $\gamma$, and $\|\phi(\xi') - \phi(\xi)\| = \mathcal{O}(r)$. Therefore, we can apply the previous argument to complete the proof. ■

## Appendix B. The Shape Space

Here we define the shape space, and show how to obtain the gradient of a function $g_j$ of a molecular configuration, at a non-singular point, in the tangent bundle of this space.

We define the *shape space*

$$\Sigma_3^{N_a} = \mathbb{R}^{3N_a}/(E(3) \times \mathbb{R}^+),$$

where $E(3)$ is the three dimensional Euclidean group composed of rigid rotations and translations in $\mathbb{R}^3$, and $\mathbb{R}^+$ is a dilation factor relative to the mean position of the $N_a$ atoms. That is, $\Sigma_3^{N_a}$ is the space of positions of $N_a$ atoms in $\mathbb{R}^3$ with equivalences given by translation, rotation, and dilation. Away from singularities of measure zero, $\Sigma_3^{N_a}$ is a Riemannian manifold (Le and Kendall, 1993; Addicoat and Collins, 2010).

Denote the Euclidean coordinates of $\mathbb{R}^{3N_a}$ by $r$, and the Euclidean position of each data point by $r_i$. Recall that we compute $a_i = a(r_i)$ for $i \in 1, \ldots n$, where $a$ is the vector-valued function

$$a : \mathbb{R}^{3N_a} \to \mathbb{R}^{3\binom{N_a}{3}}$$

that computes the angles formed by all triples of atoms in the molecule. This angular featurization of the data respects the symmetries of the shape space, and embeds the shape space.

We compute bases for the tangent spaces of $\Sigma_3^{N_a}$ as follows. For every analyzed point $i$, we compute the matrix of partial derivatives, also known as the Wilson B-matrix,

$$W_i = \frac{\partial a}{\partial r}(r_i) \in \mathbb{R}^{3N_a \times \mathbb{R}^{3\binom{N_a}{3}}}.$$

Note that $W_i$ is the transpose Jacobian of $a$. This computation is done using automatic differentiation. We then calculate the reduced singular value decomposition

$$W_i = U_i \Lambda_i V_i^T$$

where $\Lambda_i$ is a diagonal matrix of dimension $3N_a - 7$, containing the non-zero singular values of $W_i$. A deductive explanation for the rank of $W_i$ is that translation, rotation, and dilation correspond to a total of 7 degrees of freedom. The $3N_a - 7$ corresponding singular vectors

in $V_i$ are a basis for the tangent space $\mathcal{T}_i\Sigma_3^{N_a}$ in $\mathbb{R}^{3\binom{N_a}{3}}$ (Addicoat and Collins, 2010). Let $a_i = a(r_i)$ for $i \in 1, \dots n$. We can then project

$$\operatorname{grad}_{\Sigma_3^{N_a}} g_j(a_i) = V_i V_i^T \nabla_a g_j(a_i),$$

where $\nabla_a g_j(a_i)$ is obtained with automatic differentiation using a close-form expression for the dictionary function in the angular coordinates $a$ of $\mathbb{R}^{3\binom{N_a}{3}}$, and $\operatorname{grad}_{\Sigma_3^{N_a}}$ is the gradient on the shape manifold in the angular coordinates.

Recall that we apply principal component analysis to the angular features matrix $a_{1:n} \in \mathbb{R}^{n \times D}$. To perform PCA, we use singular value decomposition:

$$a_{1:n} = M \Pi N^T.$$

Denote by $P$ the matrix formed with the first $D$ columns of $N$; $P$ projects the angular features into a lower dimension space that reduces redundancy while capturing the vast majority of the variability. That is,

$$\xi_i = a_i P, \text{ for } i = 1, \dots n.$$

The gradient of $g_j$ with respect to coordinates $\xi$ are given by

$$\operatorname{grad}_\xi g_j(\xi_i) = P^T \operatorname{grad}_{\Sigma_3^{N_a}} g_j(a_i).$$

We use $\operatorname{grad}_\xi g_j(\xi_i)$ as $\nabla_\xi g_j(\xi_i)$ in ManifoldLasso.

## Appendix C. Torsion Computation

For molecular dynamics data, the dictionary $\mathcal{G}$ consists of bond torsions $g$ (see Figure 1), which are computed from planar angles of the faces of the circumscribing triangles. For example, in Figure 1b, the ordered atom 4-tuple $[9, 3, 1, 5]$ describes a torsion corresponding to the hydroxyl rotor containing the red oxygen. Their gradients are obtained using automatic differentiation in PyTorch.

As described in Section 7.2, the association of an ordered atom 4-tuple $(A, B, C, D)$ to a torsion $g(A, B, C, D)$ (where $B$ and $C$ are central, and $A$ and $D$ distal) is not unique. This is a separate issue from that of merely collinear torsions, and reflects the basic geometric properties of the analysis. There is an equivalence

$$g(A, B, C, D) = g(A, C, B, D) = g(D, C, B, A) = g(D, B, C, A).$$

For example, if $[9, 3, 1, 5]$ is explicitly included in our dictionary, then $[5, 1, 3, 9]$ is not, since these are in fact the same function. Thus, each set of 4 atoms defines 6 torsions upon ordering, since we have $\binom{4}{4}$ ordered 4-tuples, and equivalences of groups of 4. This is understandable geometrically by the fact that a tetrahedron (the shape defined by 4 points) has 6 edges, and therefore 6 torsions.

In the first set of experiments of Section 7.4, only torsions involving 3 ordered line segments in the bond diagrams are included in the dictionaries. In the second set of experiments, the dictionaries contain all possible torsions (for example, $g(4, 5, 6, 7)$ from Figure 1b is included in the latter dictionary but not in the former).

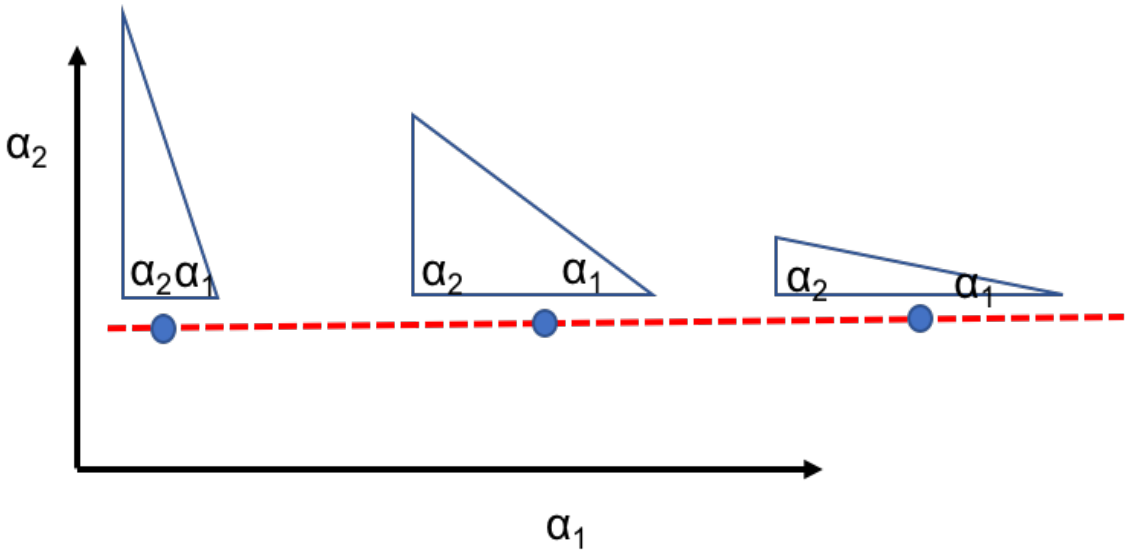



Figure 7: This diagram shows a simplified representation of the neighborhood of a point in the shape space $\Sigma_2^3$. Up to rotation, dilation, and translation, the shape of a triangle is determined by two angles, so we can see that this is a two-dimensional space. The diagram represents the logarithmic map of a region of $\Sigma_2^3$, with the red line indicating the logarithmic map of the subspace of right triangles, in a coordinate system given by $\alpha_1$ and $\alpha_2$, two angles in the triangle.

## Appendix D. Feature Space

In order to demonstrate the multiscale non-i.i.d. noise and non-trivial topology and geometry of our data $\xi$ in the PCA feature space $\mathbb{R}^D$, we display scatterplots of pairs of the top 6 coordinate, i.e., principal components, in this feature space. Note that the manifolds are relatively thin in comparison to some noise dimensions; in other words the manifold *reach* is of the same scale as the noise.

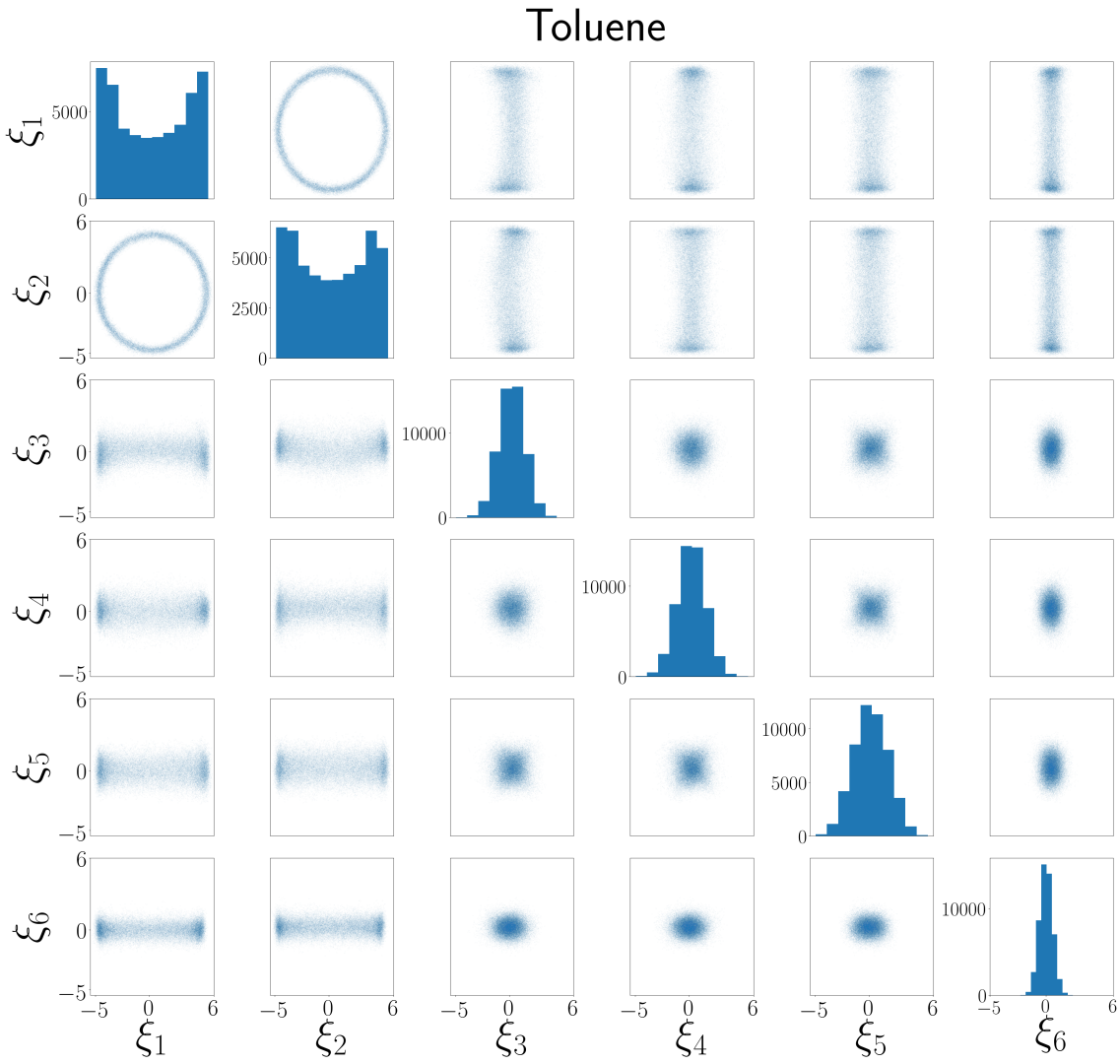



Figure 8: First 6 coordinates in $\mathbb{R}^D$ output by PCA for Toluene.

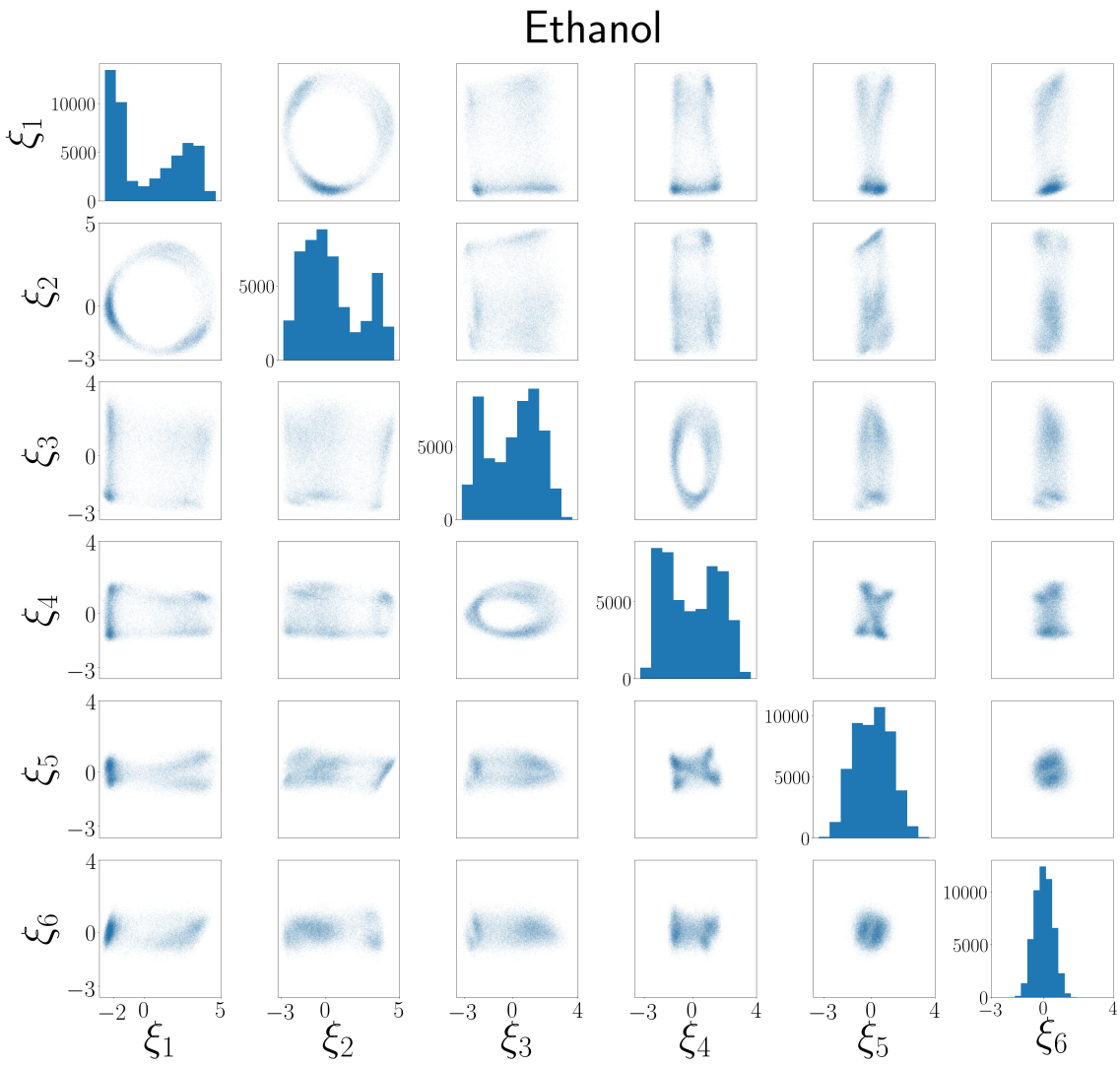



Figure 9: First 6 coordinates in $\mathbb{R}^D$output by PCA for Ethanol.

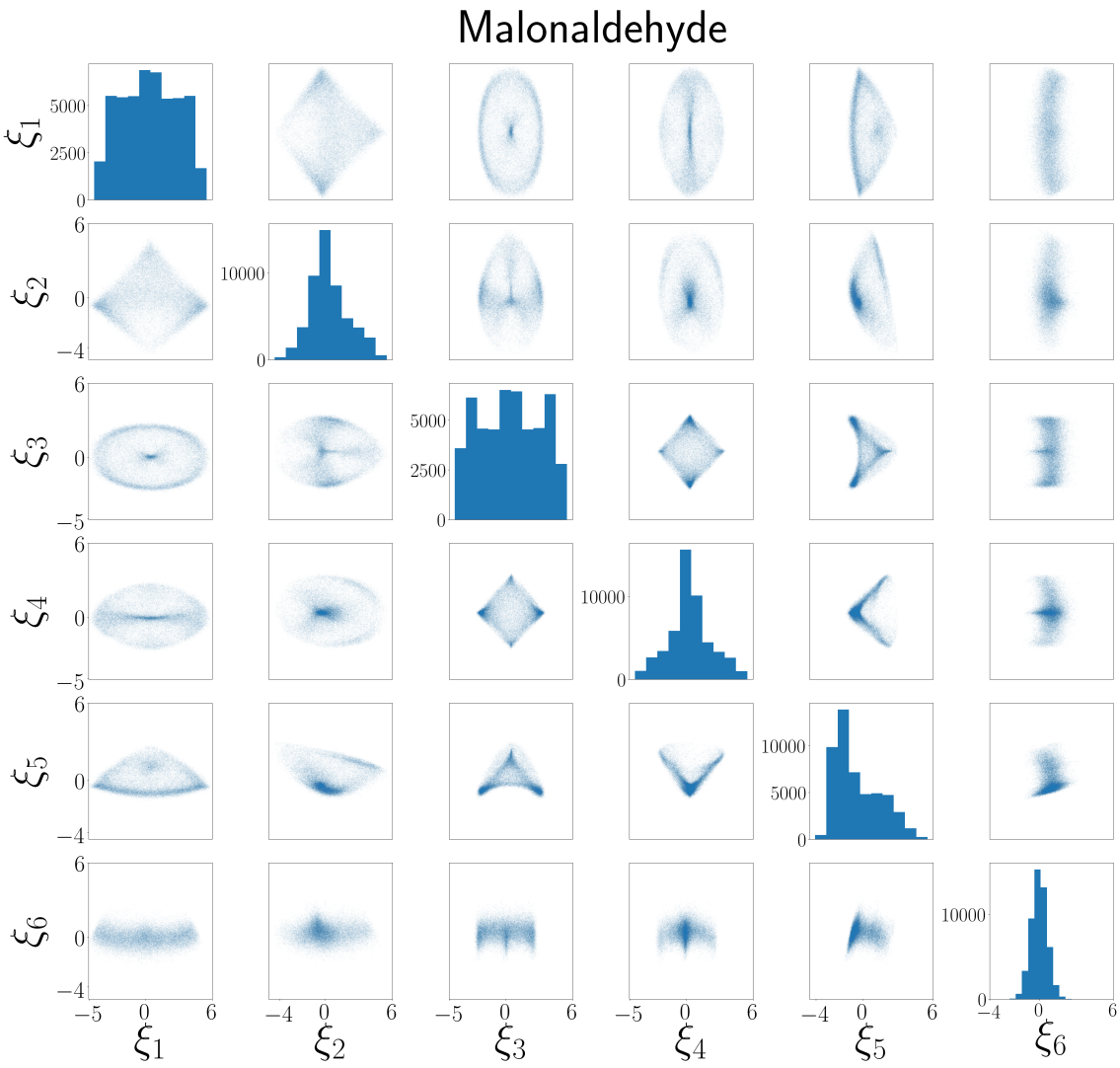



Figure 10: First 6 coordinates in $\mathbb{R}^D$ output by PCA for Malonaldehyde.

## Appendix E. Group Sparse Basis Pursuit

As mentioned in Section 4.9, the combinatorial *group sparse basis pursuit problem*

$$\arg\min_{\beta:s=d} \sum_{j=1}^{p} \|\beta_j\| s.t. \operatorname{grad} \phi_k(\xi_i) = \sum_{j=1}^{p} \beta_{ijk} \operatorname{grad}_{T_i^{\mathcal{M}}} g_j(\xi_i) \quad \text{for all } i = 1:n, \text{ and } k = 1:m \tag{28}$$

has a natural relationship to our approach. That is, for each value of $\lambda$, there is a corresponding constraint ball of radius $\epsilon$ such the solution of the lasso problem is also the solution of the

$$\arg\min_{\beta} \sum_{j=1}^{p} \|\beta_j\| s.t. \sum_{i=1}^{n} \sum_{k=1}^{m} \| \operatorname{grad} \phi_k(\xi_i) - \sum_{j=1}^{p} \beta_{ijk} \operatorname{grad}_{T_i^{\mathcal{M}}} g_j(\xi_i)\|_2^2 < \epsilon.$$

This is a combinatorial problem without restriction on the cardinality of the selected support. It can be exactly solved using the convex regularized approach.

This cardinality-unrestricted version of program (28) has several interesting properties (Candes and Tao, 2007). First, it favors gradients that are orthogonal and evenly varying. This matches our intuitively notion of what is a "good" explanation and mathematically corresponds to the notion of isometry. Second, it has a clear but not entirely obvious relation to the $l_2$ error of our method applied to gradients in $\mathbb{R}^D$ rather than $\mathbb{R}^d$, in the sense that dictionary functions which are non-tangent to $\mathcal{M}$ will accrue a higher penalty. Third, expected minimizers of such dual problems are used to define optima for sparse estimation Meinshausen and Yu (2009). We can thus use the empirical estimate of this minimizer in the cardinality-restricted setting to provide a useful notion of what is a "good" support beyond simply having low pairwise collinearity.

Compared with the standard duality, the major distinguishing feature of ManifoldLasso is the a priori knowledge of $|S|$, the cardinality of the desired support. Unfortunately, we sometimes observe that the shrinkage caused by using a high $\lambda$ to restrict support size causes the recovered support to not be close to the sample optimum of (28). Dictionary functions with large projection $(\sum_{i=1}^{n}\sum_{k=1}^{m}(\operatorname{grad}_{T_i^{\mathcal{M}}} g_j(\xi_i))^T(\operatorname{grad}_{T_i^{\mathcal{M}}} \phi_m(\xi_i)))^{1/2}$ tend to appear early the regularization path, regardless of their orthogonality or consistency of variation. Problems with shrinkage including variable selection inconsistency at large $\lambda$, as well as the desirable properties of an intermediate value, are well-established in the support recovery and sparse coding literature (Chen et al., 1998; Hesterberg et al., 2008; Breheny and Huang, 2011; Lederer and Müller, 2015; Hastie and Tibshirani, 2015).

We can empirically adapt our method to respond to this problem while still leveraging the advantages of the convex algorithm. We follow Hesterberg et al. (2008) in using an intermediate $\lambda$ as a variable filtering step prior to variable selection, in our case, using (28). In this adapted approach, we initially prune the dictionary using ManifoldLasso with an intermediate $\lambda$ value. We then run program (28) on the pruned dictionary. Initial selection of $d' << p$ using our approach prior to selection of $d$ variables using program (28) is often more effective in obtaining the empirical result of program (28) than ManifoldLasso on its own, and is much more computationally feasible than running program (28) on the entire dictionary. The $\lambda$ at which these $d'$ functions are obtained is somewhat arbitrary, since a fully data-driven approach would require computation of program (28) on the entire dictionary, but

relatively generic theoretical arguments provide blanket arguments in favor of $\lambda > O(\log p)$ (Chen et al., 1998). We in general find relatively wide regions of relatively low cardinality, and substantial improvements in the combinatorial loss with minimal computational burden at $\lambda = \lambda_{\max}/2$. Results for this two-stage method for **Ethanol** and **Malonaldehyde** are displayed in Figure 13.

## Appendix F. Supplemental Experiments

We include supplemental experiments showing association of individual embedding coordinates to dictionary functions, orthogonal dictionary function selection using our two-stage variable pruning method, embeddings colored by selected functions, and calculated theoretical quantities.

### F.1 Coordinate Association in a Priori Dictionaries

We show the association of individual embedding coordinates to dictionary functions in **Ethanol** and **Malonaldehyde**. In contrast to **Malonaldehyde**, but similar to **SwissRoll**, **Ethanol** has a distinct association of embedding coordinates with dictionary functions. In particular, $\phi_3$ is associated with different torsions from $\phi_1$ and $\phi_2$. This is clearly evident in Figure 1. In **Malonaldehyde**, there is no clear association with embeddings coordinates. Note that this would also be true for **Toluene**, as Figure 1 clearly shows a circular manifold symmetric in $\phi_1$ and $\phi_2$.

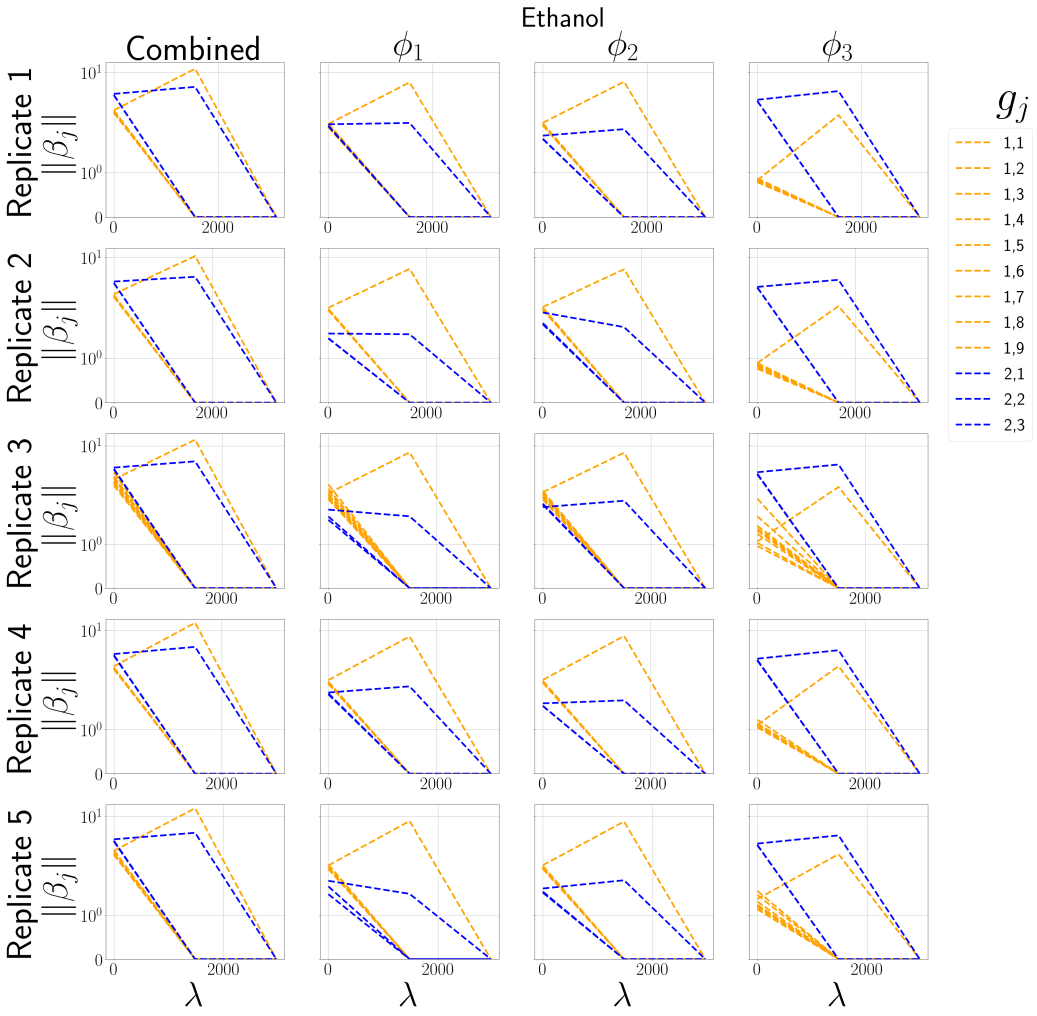



Figure 11: Combined and coordinate-specific regularization paths in five replicates of Manifold-Lasso for **Ethanol** with dictionary given by the bond diagram. There is a clear association of the blue torsion with $\phi_3$, and orange with $\phi_1$ and $\phi_2$.

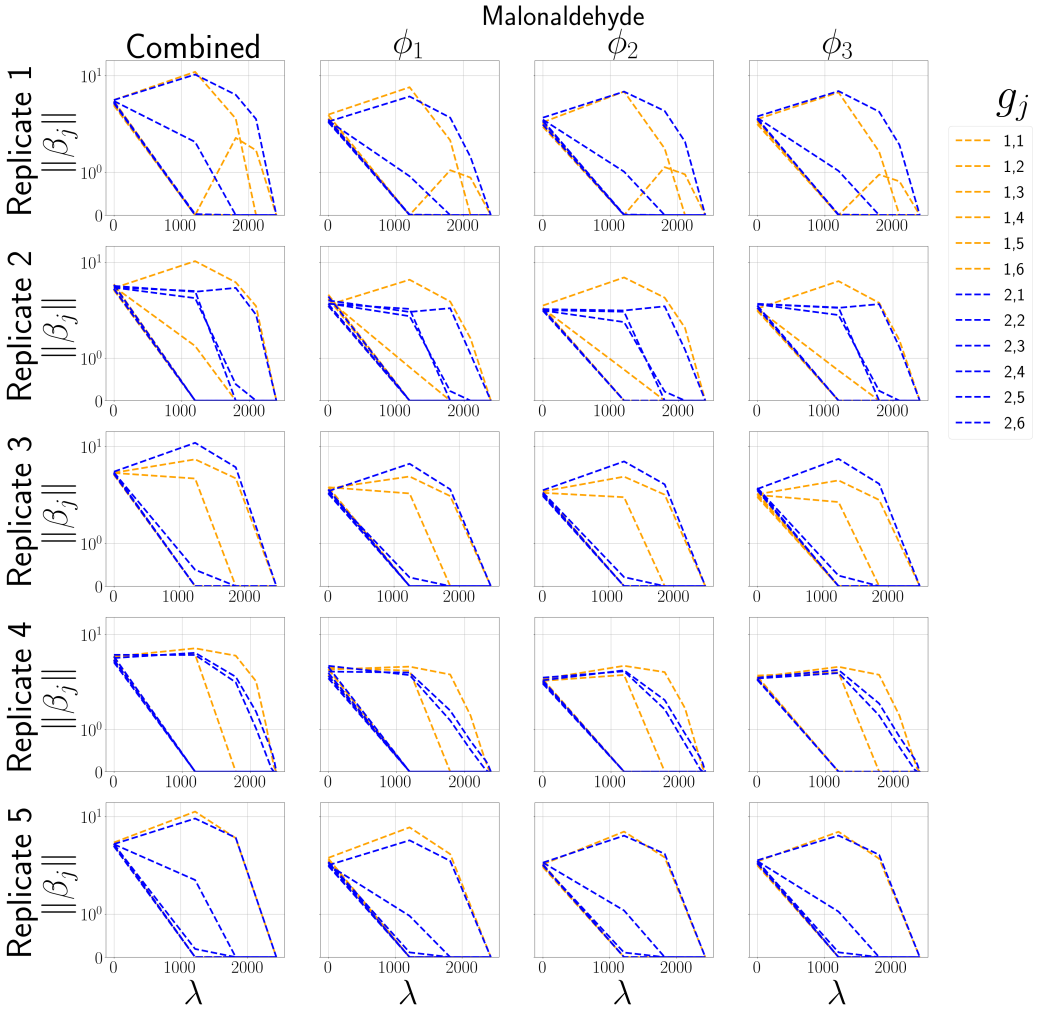



Figure 12: Combined and coordinate-specific regularization paths in five replicates of Manifold-Lasso for **Malonaldehyde** with dictionary given by the bond diagram. There is no clear association of embedding coordinates and covariates.

### F.2 Two-stage Method Results

The two-stage approach obtains highly orthogonal solutions collinear with the true support at minimal computational cost. Comparison of selected functions with the visualizations in Appendix F.4 shows a clear correspondence between variable selection using the two-stage method and visual identification of orthogonal supports based on colored embeddings.

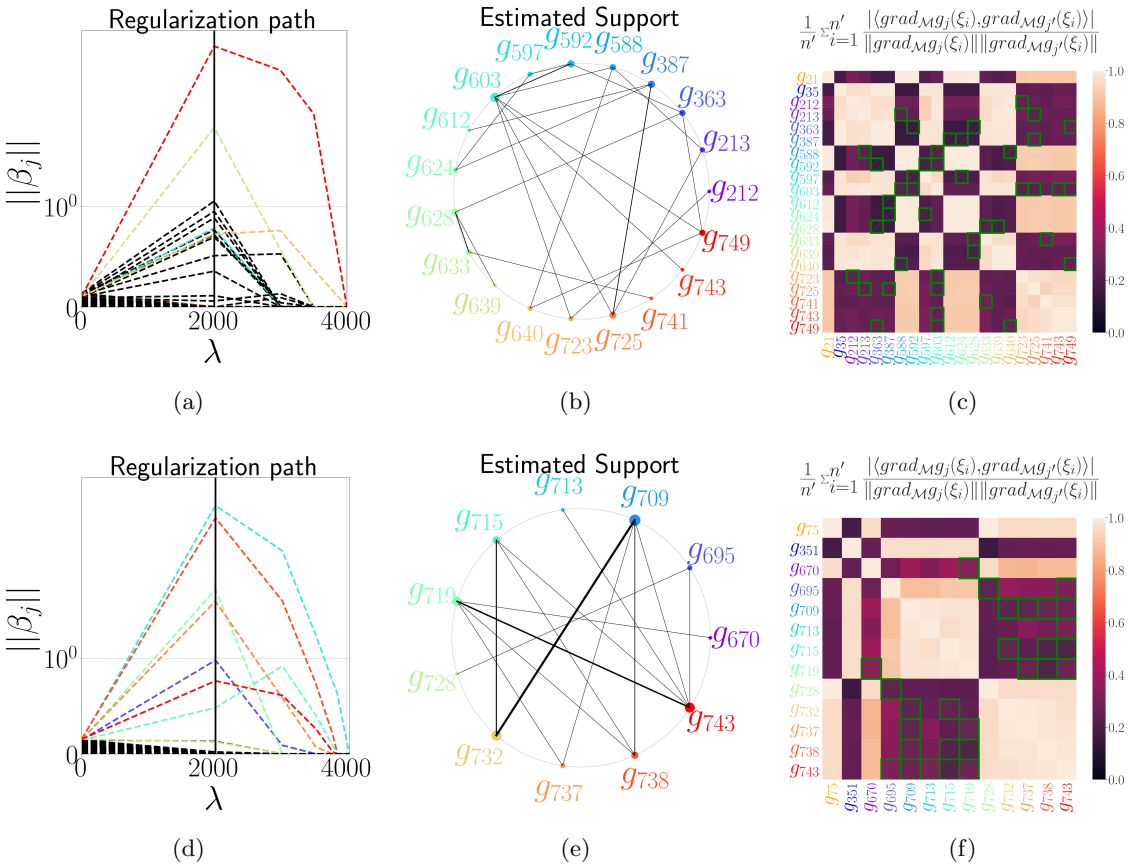



Figure 13: Two-stage results for **Ethanol** and **Malonaldehyde**, respectively, with dictionaries given by all possible torsions. Figures 13a and 13d show individual replicates. The pruning tuning parameter value $\lambda_{\max}/2$ is represented by the vertical black lines. Colors are plotted for functions selected by subsequent combinatorial analysis. Figure 13b and 13e shows support recoveries given by subset selection using group lasso at $\lambda_{\max}/2$ followed by program (28) over $\omega = 25$ different replications. Figure 13c and 13f show mean cosine collinearity of selected supports. $g_{21,35}$ and $g_{35,351}$ are representative torsions from the true support, while the others are included if they are selected in any replicate. Pairs that are selected in any replicate are marked with a green box.

### F.3 Visualizing Functions Selected by ManifoldLasso from Full Dictionaries

We can visualize the selected torsions in the manifold embedding coordinates. The identities of the selected torsions can be compared with the bond diagrams in Appendix C. We first visualize the functions selected using ManifoldLasso from Figure 6. Note that certain functions do not appear to parameterize the manifold.

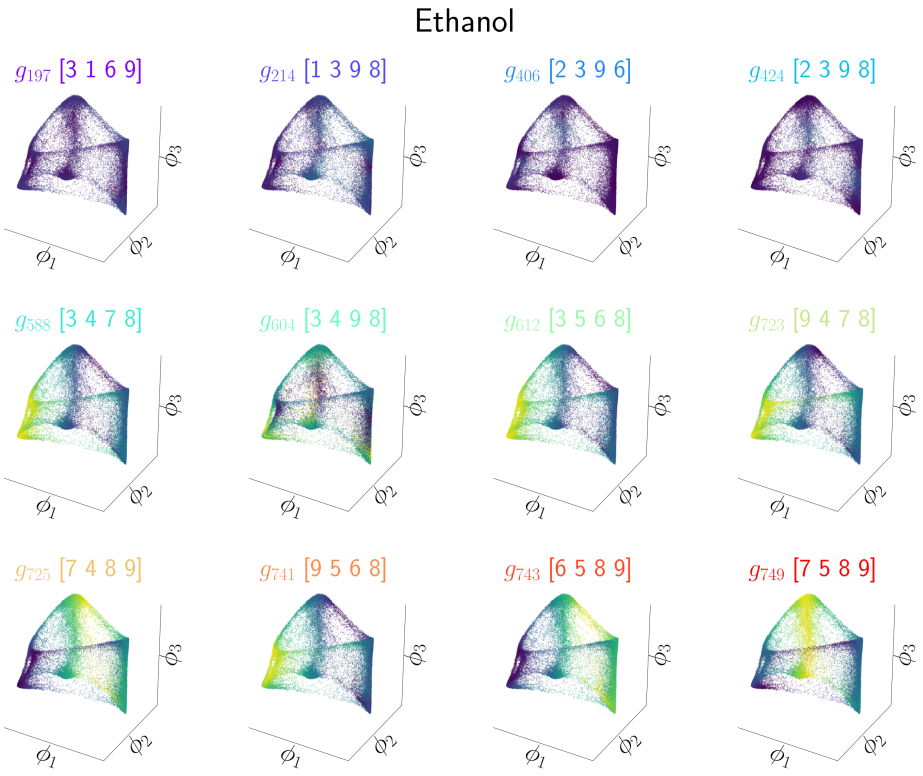



Figure 14: **Ethanol** support estimated using ManifoldLasso with full dictionary. Dictionary function colors should be compared with Figure 6. The four numbers in each subtitle correspond to the atoms in Figure 1 that inscribe the torsion.

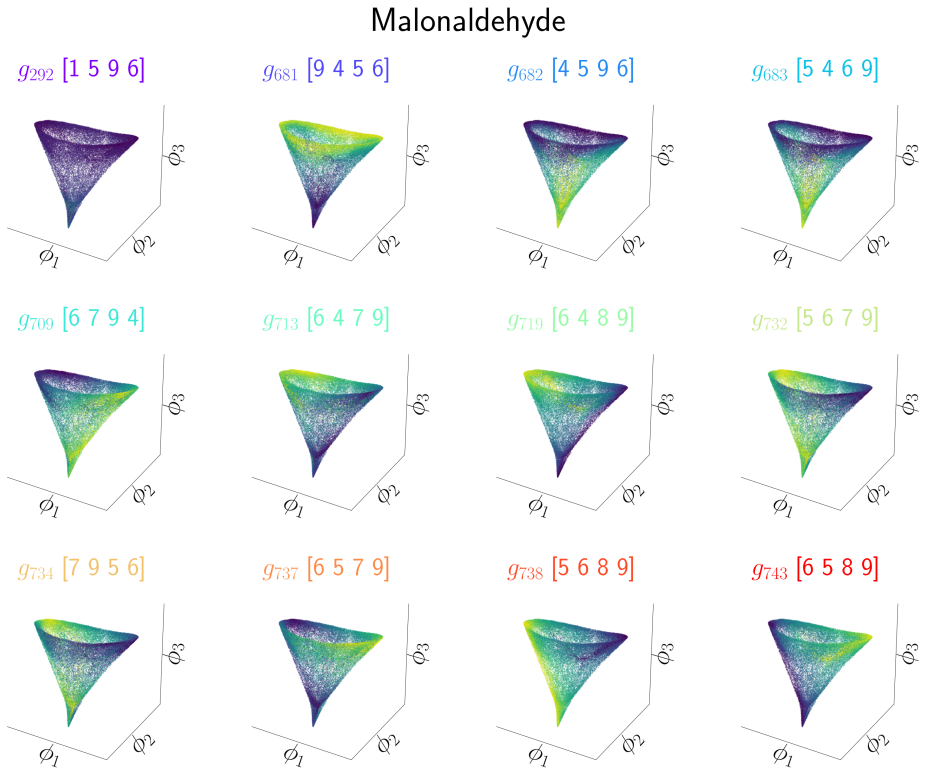



Figure 15: **Malonaldehyde** support using using ManifoldLasso with full dictionary. Dictionary function colors should be compared with Figure 6. The four numbers in each subtitle correspond to the atoms in Figure 1 that inscribe the torsion.

### F.4 Visualizing Functions Selected by the Two-stage Method from Full Dictionaries

We also visualize the selected functions from Figure 13. Selected pairs of functions for **Ethanol** are more orthogonal than found using ManifoldLasso.

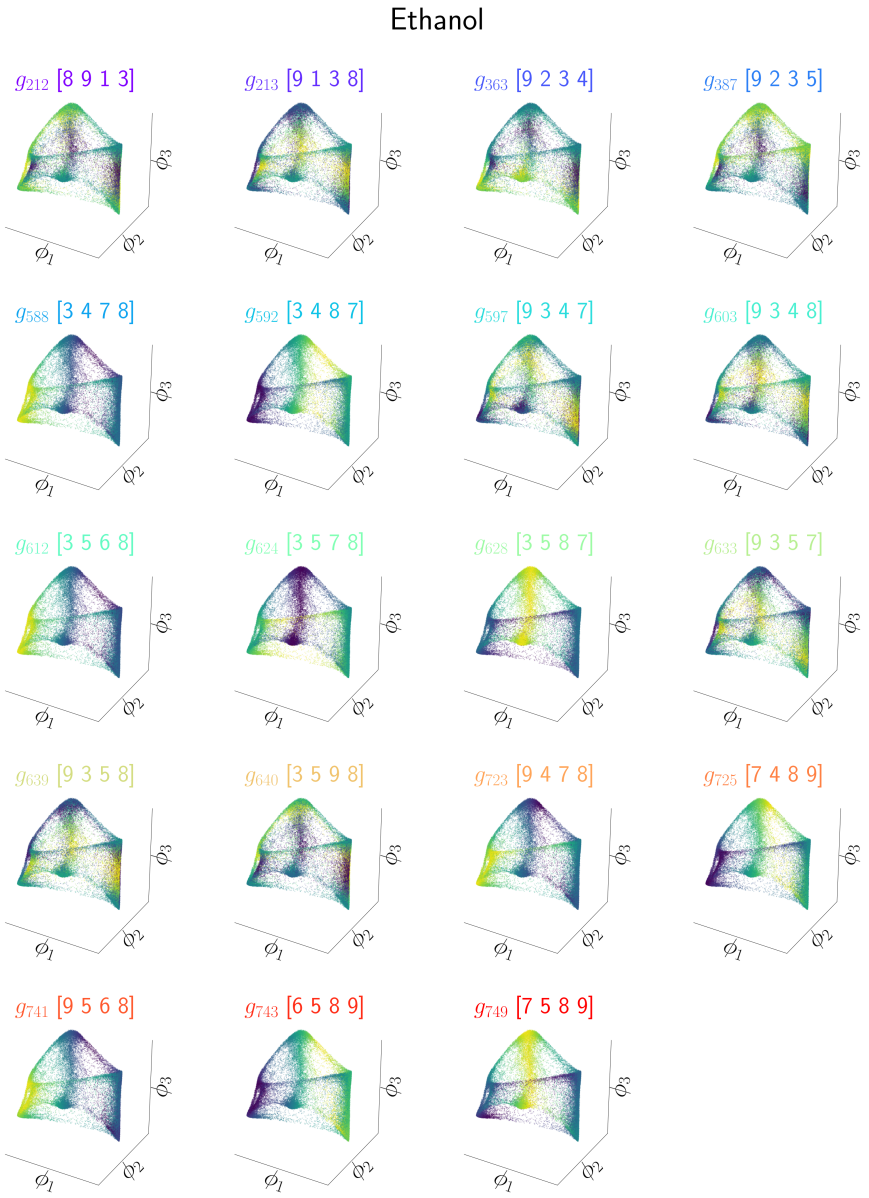



Figure 16: **Ethanol** support using basis pursuit on superset obtained using ManifoldLasso. Colors should be compared with Figure 13. The four numbers in each subtitle correspond to the atoms in Figure 1 that inscribe the torsion.

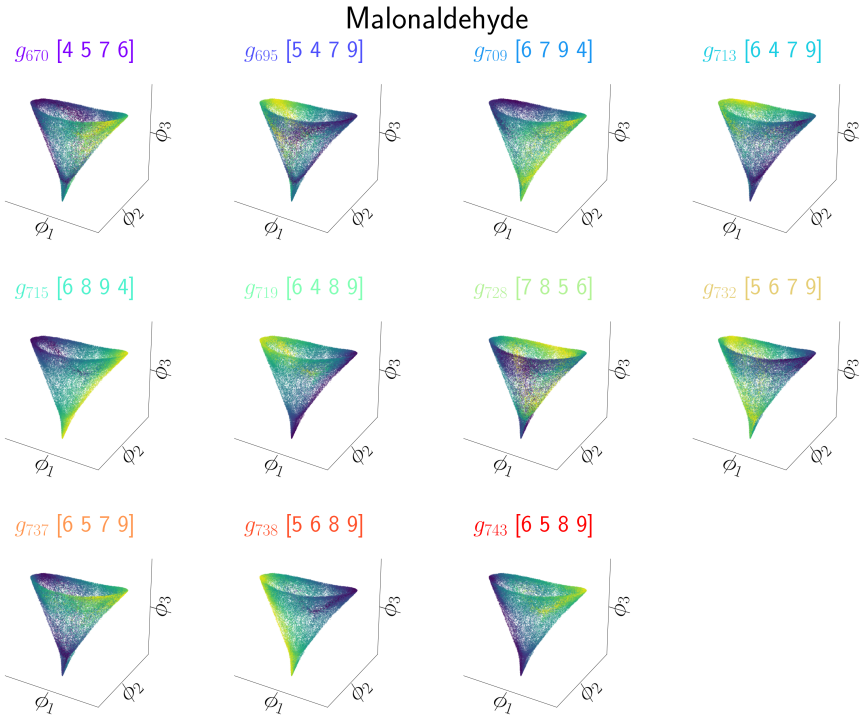



Figure 17: **Malonaldehyde** support using basis pursuit on superset obtained using Manifold-Lasso. Colors should be compared with Figure 13. The four numbers in each subtitle correspond to the atoms in Figure 1 that inscribe the torsion.

### F.5 Calculated Theoretical Quantities

We calculate the theoretical quantities $\mu$, $\gamma_{\max}$, $\kappa_S$, and $\min_{i=1}^{n} \min_{j' \in S} \|x_{ij}\|$ over replicates using the putative true supports from Figure 1.

| | $\bar{\mu}$ | $\sigma_\mu$ | $\bar{\kappa_S}$ | $\sigma_{\kappa_S}$ | $\gamma_{\max}$ | $\sigma_{\gamma_{\max}}$ | $\min_{i=1}^{n} \bar{\min}_{j \in S} \|\|x_{ij}\|\|$ | $\sigma_{\min_{i=1}^{n} \min_{j \in S} \|\|x_{ij}\|\|}$ |
|---|---|---|---|---|---|---|---|---|
| Ethanol (a priori) | 1.0 − 3.83e-8 | 5.57e-8 | 5.83 | 2.83 | 46.3 | 1.6 | .154 | 0.084 |
| Malonaldehyde (a priori) | 1.0 − 7.83e-8 | 9.28e-8 | 1.87 | 0.18 | 21.9 | 0.5 | 0.282 | 0.035 |
| Toluene (a priori) | | | | | 12.4 | 1.3 | .00985 | .00939 |
| Ethanol (agnostic) | 1.0 − 2.30e-11 | 3.55e-11 | 5.83 | 2.83 | 46.3 | 1.6 | 0.154 | 0.084 |
| Malonaldehyde (agnostic) | 1.0 − 2.22e-11 | 4.79e-11 | 1.87 | 0.18 | 21.9 | 0.5 | 0.282 | 0.035 |

Table 2: Mean and standard deviation of theoretical quantities across replicates.

## References

E. Aamari and C. Levrard. Nonasymptotic rates for manifold, tangent space and curvature estimation. *Annals of Statistics*, 47(1):177–204, Feb. 2019.

M. A. Addicoat and M. A. Collins. Potential energy surfaces: the forces of chemistry. In M. Brouard and C. Vallance, editors, *Tutorials in Molecular Reaction Dynamics*, chapter 2, pages 28–49. Royal Society of Chemistry Publishing, London, 2010.

J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, 2018.

E.-A. D. Amir, K. L. Davis, M. D. Tadmor, E. F. Simonds, J. H. Levine, S. C. Bendall, D. K. Shenfeld, S. Krishnaswamy, G. P. Nolan, and D. Pe'er. viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia. *Nat. Biotechnol.*, 31(6):545–552, June 2013.

A. Aswani, P. Bickel, and C. Tomlin. Regression on manifolds: Estimation of the exterior derivative. *Annals of Statistics*, 39(1):48–81, 2011.

H. Banville, I. Albuquerque, A. Hyvärinen, G. Moffat, D.-A. Engemann, and A. Gramfort. Self-supervised representation learning from electroencephalography signals. In *IEEE International Workshop on Machine Learning for Signal Processing*, 2019.

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm with application to wavelet-based image deblurring. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009.

M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems*, 2002.

E. Bernhardsson. Annoy (Approximate Nearest Neighbors Oh Yeah), 2015. URL `https://github.com/spotify/annoy`.

R. L. Bishop. Riemannian geometry. *arXiv e-prints*, Mar. 2013.

M. R. Blanton and M. A. Bershady. Sloan digital sky survey IV: Mapping the milky way, nearby galaxies, and the distant universe. *Astron. J.*, 154:28, July 2017.

K. J. Bowers, D. E. Chow, H. Xu, R. O. Dror, M. P. Eastwood, B. A. Gregersen, J. L. Klepeis, I. Kolossvary, M. A. Moraes, F. D. Sacerdoti, J. K. Salmon, Y. Shan, and D. E. Shaw. Scalable algorithms for molecular dynamics simulations on commodity clusters. In *IEEE Conference on Supercomputing*, 2006.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, USA, 2004.

P. Breheny and J. Huang. Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *Annals of Applied Statistics*, 5(1):232–253, Jan. 2011.

S. L. Brunton, J. L. Proctor, and J. N. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences, USA*, 113(15):3932–3937, 2016.

J. Buenfil, S. Koelle, and M. Meilă. Tangent space least adaptive clustering. In *International Conference on Machine Learning Workshop on Unsupervised Reinforcement Learning*, 2021.

E. Candes and T. Tao. The dantzig selector: Statistical estimation when p is much larger than n. *Annals of Statistics*, 35(6):2313–2351, Dec. 2007.

K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences, USA*, 116(45):22445–22451, Nov. 2019.

G. Chen, A. V. Little, and M. Maggioni. *Multi-Resolution Geometric Analysis for Data in High Dimensions*, pages 259–285. Birkhäuser Boston, Boston, MA, 2013.

S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal of Scientific Computing*, 20(1):33–61, Jan. 1998.

X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2016.

Y.-C. Chen and M. Meilă. Selecting the independent coordinates of manifolds with large aspect ratios. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, 2019.

Y.-C. Chen, J. McQueen, S. J. Koelle, M. Meila, S. Chmiela, and A. Tkatchenko. Modern manifold learning methods for md data – a step by step procedural overview. `www.stat.washington.edu/mmp/Papers/mlcules-arxiv.pdf`, July 2019.

S. Chmiela, A. Tkatchenko, H. Sauceda, I. Poltavsky, K. T. Schütt, and K.-R. Müller. Machine learning of accurate energy-conserving molecular force fields. *Science Advances*, March 2017.

C. Clementi, H. Nymeyer, and J. Onuchic. Topological and energetic factors: what determines the structural details of the transition state ensemble and "en-route" intermediates for protein folding? an investigation for small globular proteins. *Journal of Molecular Biology*, 2000.

R. R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 30(1): 5–30, 2006.

R. R. Coifman, S. Lafon, A. Lee, Maggioni, Warner, and Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the National Academy of Sciences, USA*, pages 7426–7431, 2005.

P. Constantine, E. Dow, and Q. Wang. Active subspace methods in theory and practice: Applications to kriging surfaces. *SIAM Journal on Scientific Computing*, 36(4):A1500–A1524, 2014.

P. Das, M. Moll, H. Stamati, L. Kavraki, and C. Clementi. Low-dimensional, free-energy landscapes of protein-folding reactions by nonlinear dimensionality reduction. *Proceedings of the National Academy of Sciences, USA*, 103(26):9885–9890, 2006.

M. do Carmo. *Riemannian Geometry*. Springer, 1992.

C. J. Dsilva, R. Talmon, N. Rabin, R. R. Coifman, and I. G. Kevrekidis. Nonlinear intrinsic variables and state reconstruction in multiscale simulations. *The Journal of Chemical Physics*, 139(18): 184109, 2013.

C. J. Dsilva, R. Talmon, R. R. Coifman, and I. G. Kevrekidis. Parsimonious representation of nonlinear dynamical systems through manifold learning: A chemotaxis case study. *Applied and Computational Harmonic Analysis*, 44(3):759–773, May 2018.

B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, and Others. Least angle regression. *Annals of Statistics*, 32(2):407–499, 2004.

M. K. Elyaderani, S. Jain, J. Druce, S. Gonella, and J. Haupt. Group-level support recovery guarantees for group lasso estimator. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017.

J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, Dec. 2001.

G. Fiorin, M. L. Klein, and J. Hénin. Using collective variables to drive molecular dynamics simulations. *Molecular Physics*, 111(22-23):3345–3362, Dec. 2013.

K. L. Fleming, P. Tiwary, and J. Pfaendtner. New approach for investigating reaction dynamics and rates with ab initio calculations. *Journal of Physical Chemistry A*, 120(2):299–305, Jan. 2016.

C. W. Gear. Parameterization of non-linear manifolds, Aug. 2012.

T. Hastie and R. Tibshirani. *Statistical Learning With Sparsity : the Lasso and Generalizations*. Monographs on statistics and applied probability, no. 143. CRC Press, special indian ed. edition, 2015.

S. Haufe, V. V. Nikulin, A. Ziehe, K.-R. Müller, and G. Nolte. Estimating vector fields using sparse basis field expansions. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, 2009.

M. Hein, J. Audibert, and U. von Luxburg. From graphs to manifolds - weak and strong pointwise consistency of graph laplacians. In *Learning Theory, 18th Annual Conference on Learning Theory, COLT 2005, Bertinoro, Italy, June 27-30, 2005, Proceedings*, pages 470–485, 2005. doi: 10.1007/11503415_32.

M. Hein, J. Audibert, and U. von Luxburg. Graph laplacians and their convergence on random neighborhood graphs. *Journal of Machine Learning Research*, 8:1325–1368, 2007.

E. J. Heller. The correspondence principle and intramolecular dynamics. *Faraday Discussions of Chemical Society*, 75(0):141–153, Jan. 1983.

C. A. Herring, A. Banerjee, E. T. McKinley, A. J. Simmons, J. Ping, J. T. Roland, J. L. Franklin, Q. Liu, M. J. Gerdes, R. J. Coffey, and K. S. Lau. Unsupervised trajectory analysis of Single-Cell RNA-Seq and imaging data reveals alternative tuft cell origins in the gut. *Cell Systems*, 6(1): 37–51, Jan. 2018.

T. Hesterberg, N. H. Choi, L. Meier, and C. Fraley. Least angle and $\ell_1$ penalized regression: A review. *Statistics Surveys*, 2:61 – 93, 2008.

Huan Xu, C. Caramanis, and S. Mannor. Sparse algorithms are not stable: A No-Free-Lunch theorem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(1), Jan. 2012.

M. Jas, T. Achakulvisut, A. Idrizović, D. Acuna, M. Antalek, V. Marques, T. Odland, R. Garg, M. Agrawal, Y. Umegaki, P. Foley, H. Fernandes, D. Harris, B. Li, O. Pieters, S. Otterson, G. D. Toni, C. Rodgers, E. Dyer, M. Hamalainen, K. Kording, and P. Ramkumar. Pyglmnet: Python implementation of elastic-net regularized generalized linear models, 2020. URL `http://glm-tools.github.io/pyglmnet/`.

D. Joncas, M. Meila, and J. McQueen. Improved graph laplacian via geometric Self-Consistency. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, 2017.

D. G. Kendall. A survey of the statistical theory of shape. *Statistical Science*, 1989.

M. Kleindessner and U. von Luxburg. Dimensionality estimation without distances. In *International Conference on Artificial Intelligence and Statistics*, 2015.

M. Krenn, F. Häse, A. Nigam, P. Friederich, and A. Aspuru-Guzik. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Machine Learning : Science and Technology*, 1(4), Oct. 2020.

N. P. Landsman. Between classical and quantum. *Handbook of the Philosophy of Science*, 2:417–553, 2007.

H. Le and D. G. Kendall. The riemannian structure of euclidean shape spaces: A novel environment for statistics. *Annals of Statistics*, 21(3):1225–1271, 1993.

J. Lederer and C. Müller. Don't fall for tuning parameters: Tuning-free variable selection in high dimensions with the TREX. *AAAI Conference on Artificial Intelligence*, Jan. 2015.

J. M. Lee. *Introduction to Smooth Manifolds*. Springer, 2003.

Z. Lin, K. K. Thekumparampil, G. C. Fanti, and S. Oh. Infogan-cr and modelcentrality: Self-supervised model training and selection for disentangling gans. In *International Conference on Machine Learning*, 2020.

C. Luo, I. Safa, and Y. Wang. Approximating gradients for meshes and point clouds via diffusion metric. *Computer Graphics Forum*, 28(5):1497–1508, July 2009.

J. McQueen, M. Meila, J. VanderPlas, and Z. Zhang. Megaman: Scalable manifold learning in python. *Journal of Machine Learning Research*, 17:148:1–148:5, 2016.

M. Meila, S. Koelle, and H. Zhang. A regression approach for explaining manifold embedding coordinates. *arXiv e-prints*, page 1811.11891, Nov 2018.

N. Meinshausen. Relaxed lasso. *Computational Statistics and Data Analysis*, 52(1):374–393, Sept. 2007.

N. Meinshausen and P. Bühlmann. Stability selection: Stability selection. *Journal of the Royal Statistical Society Series B Statistical Methodology*, 72(4):417–473, July 2010.

N. Meinshausen and B. Yu. Lasso-type recovery of sparse representations for high-dimensional data. *Annals of Statistics*, 37(1):246 – 270, 2009.

K. Mohammed and H. Narayanan. Manifold learning using kernel density estimation and local principal components analysis. *arXiv e-prints*, page 1709.03615, 2017.

S. Mukherjee and D.-X. Zhou. Learning coordinate covariances via gradients. *Journal of Machine Learning Research*, 7:519–549, Mar 2006.

F. Noé and C. Clementi. Collective variables for the study of long-time kinetics from molecular trajectories: theory and methods. *Current Opinions on Structural Biology*, 43:141–147, Apr. 2017.

G. Obozinski, M. J. Wainwright, and M. I. Jordan. Support union recovery in high-dimensional multivariate regression. *Annals of Statistics*, 39(1):1–47, 2011.

S. Pant, Z. Smith, Y. Wang, E. Tajkhorshid, and P. Tiwary. Confronting pitfalls of AI-augmented molecular dynamics using statistical physics. *Journal of Chemical Physics*, 153(23), Dec. 2020.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: an imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8026–8037, Dec. 2019.

D. Perraul-Joncas and M. Meila. Non-linear dimensionality reduction: Riemannian metric estimation and the problem of geometric discovery, May 2013.

N. Puchkin and V. Spokoiny. Structure-adaptive manifold estimation. *Journal of Machine Learning Research*, 23(40):1–62, 2022.

L. Qu, S. An, T. Yang, and Y. Sun. Group sparse basis pursuit denoising reconstruction algorithm for polarimetric Through-the-Wall radar imaging. *International Journal of Antennas and Propagation*, 2018, Aug. 2018.

M. A. Rohrdanz, W. Zheng, M. Maggioni, and C. Clementi. Determination of reaction coordinates via locally scaled diffusion map. *Journal of Chemical Physics*, 134(12), 2011.

M. A. Rohrdanz, W. Zheng, and C. Clementi. Discovering mountain passes via torchlight: Methods for the definition of reaction coordinates and pathways in complex macromolecular reactions. *Annual Review of Physical Chemistry*, 64:295–316, 2013.

S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290 (5500):2323–2326, December 2000.

S. Rudy, A. Alla, S. L. Brunton, and J. N. Kutz. Data-Driven identification of parametric partial differential equations. *SIAM Journal of Applied Dynamical Systems*, 18(2):643–660, Jan. 2019.

L. K. Saul and S. T. Roweis. Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, 4:119–155, Dec. 2003.

W. Sheng. Section 5. geodesics and the exponential map, December 2009. URL `http://www.mathweb.zju.edu.cn:8080/swm/RG_Section_5.pdf`.

A. Shukla, S. Uppal, S. Bhagat, S. Anand, and P. Turaga. Geometry of deep generative models for disentangled representations. In *Indian Conference on Computer Vision, Graphics and Image Processing*, 2018.

H. Sidky, W. Chen, and A. L. Ferguson. Machine learning for collective variable discovery and enhanced sampling in biomolecular simulation. *Molecular Physics*, 118(5), Mar. 2020.

C. D. Sogge. *Hangzhou Lectures on Eigenfunctions of the Laplacian*. Princeton University Press, 2014.

Z. Szabó, B. Póczos, and A. Lőrincz. Online group-structured dictionary learning. In *Conference on Computer Vision and Pattern Recognition*, pages 2865–2872, 2011.

Y. W. Teh and S. T. Roweis. Automatic alignment of local representations. In *Advances in Neural Information Processing Systems*, 2002.

D. Ting, L. Huang, and M. I. Jordan. An analysis of the convergence of graph laplacians. In *International Conference on Machine Learning*, 2010.

G. A. Tribello, M. Ceriotti, and M. Parrinello. Using sketch-map coordinates to analyze and bias molecular dynamics simulations. *Proceedings of the National Academy of Sciences, USA*, 109: 5196–5201, 2012.

M. J. Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using $\ell_1$ -constrained quadratic programming (lasso). *IEEE Transactions on Information Theory*, 55: 2183–2202, 2009.

Y. Wang, J. M. L. Ribeiro, and P. Tiwary. Past-future information bottleneck for sampling molecular reaction coordinate simultaneously with thermodynamics and kinetics. *Nature Communications*, 10(1), Aug. 2019.

Q. Wu, J. Guinney, M. Maggioni, and S. Mukherjee. Learning gradients: predictive models that infer geometry and statistical dependence. *Journal of Machine Learning Research*, 2010.

T. Xie, A. France-Lanord, Y. Wang, Y. Shao-Horn, and J. C. Grossman. Graph dynamical networks for unsupervised learning of atomic scale dynamics in materials. *Nature Communications*, 10(1), June 2019.

G. Yang. Tensor programs II: Neural tangent kernel for any architecture. *arXiv e-prints*, page 2006.14548, June 2020.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B Statistical Methodology*, 2006.

S. Zelditch. Quantum ergodicity and mixing of eigenfunctions. In *Encyclopedia of Mathematical Physics*, pages 183–196. Elsevier, 2006.

S. Zhang, Y. Cui, X. Ma, J. Yong, L. Yan, M. Yang, J. Ren, F. Tang, L. Wen, and J. Qiao. Single-cell transcriptomics identifies divergent developmental lineage trajectories during human pituitary development. *Nature Communications*, 11(1), Oct. 2020.

Z. Zhang and H. Zha. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM Journal of Scientific Computing*, 26(1):313–338, 2004.

P. Zhao and B. Yu. On model selection consistency of lasso. *Journal of Machine Learning Research*, 7:2541–2563, 2006.

V. A. Zorich. *Mathematical Analysis I.* Springer-Verlag Berlin Heidelberg, 2004.

H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, Dec. 2006.