

# Instant recovery of shape from spectrum via latent space connections

Riccardo Marin<sup>1</sup>, Arianna Rampini<sup>2</sup>, Umberto Castellani<sup>1</sup>,  
 Emanuele Rodolà<sup>2</sup>, Maks Ovsjanikov<sup>3</sup>, Simone Melzi<sup>3</sup>

<sup>1</sup> University of Verona

<sup>2</sup> Sapienza University of Rome

<sup>3</sup> LIX, Ecole Polytechnique

**Abstract.** We introduce the first learning-based method for recovering shapes from Laplacian spectra. Given an auto-encoder, our model takes the form of a cycle-consistent module to map latent vectors to sequences of eigenvalues. This module provides an efficient and effective linkage between spectrum and geometry of a given shape. Our data-driven approach replaces the need for ad-hoc regularizers required by prior methods, while providing more accurate results at a fraction of the computational cost. Our learning model applies without modifications across different dimensions (2D and 3D shapes alike), representations (meshes, contours and point clouds), as well as across different shape classes, and admits arbitrary resolution of the input spectrum without affecting complexity. The increased flexibility allows us to provide a proxy to differentiable eigendecomposition and to address notoriously difficult tasks in 3D vision and geometry processing within a unified framework, including shape generation from spectrum, mesh super-resolution, shape exploration, style transfer, spectrum estimation from point clouds, segmentation transfer and point-to-point matching.

**Keywords:** Shape from spectrum, Laplacian eigenvalues, latent space, spectral geometry, auto-encoder.

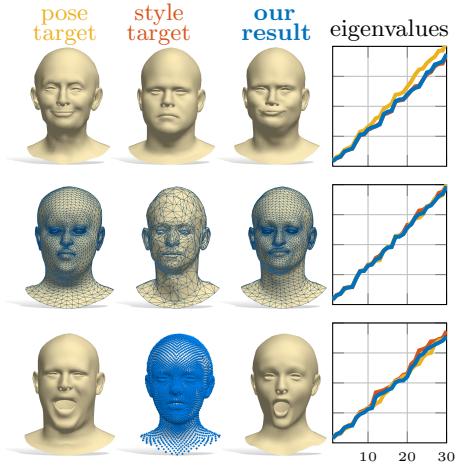
## 1 Introduction

Constructing compact encodings of geometric shapes lies at the heart of 2D and 3D Computer Vision. While earlier approaches have concentrated on handcrafted representations, with the advent of geometric deep learning [12,34], data-driven learned feature encodings have gained prominence. A desirable property in many applications, such as shape exploration and synthesis, is to be able to recover the shape from its (latent) encoding, and various auto-encoder architectures have been designed to solve this problem [2,33,35,21]. Despite significant progress in this area, the structure of the latent vectors is not easy to control or analyze. For example, the dimensions of the latent vectors typically lack a canonical ordering, while invariance to various geometric deformations is often only learned by data augmentations or complex constraints on the intermediate features.

At the same time, a classical approach in the domain of spectral geometry is to encode a shape using the sequence of eigenvalues (spectrum) of its Laplacian operator. This representation is useful since: (1) it does not require any training, (2) it can be computed on various data representations, such as point clouds or meshes, regardless of sampling density, (3) it enjoys many well-known theoretical properties such as a natural ordering of its elements and invariance to isometries, and (4) as shown recently [20,41], alignment of eigenvalues often promotes near-isometries, which is useful in multiple practical tasks such as solving non-rigid shape detection and matching problems.

Unfortunately, although encoding shapes via their Laplacian spectra can be straightforward (at least for meshes), the inverse problem of recovering the shape is very difficult. Indeed, it is well-known that certain pairs of non-isometric shapes can have the same spectrum, or in other words “one cannot hear the shape of a drum” [22]. At the same time, recent evidence suggests that such cases are pathological and that *in practice* it might be possible to recover a shape from its spectrum [20]. Nevertheless, existing approaches [20], while able to deform a shape into another with a given spectrum, can produce highly unrealistic shapes with strong artifacts failing in a large number of cases.

In this paper, we combine the strengths of data-driven auto-encoders with those of spectral methods. Our key idea is to construct a single architecture capable of synthesizing a shape from a learned latent code and from its Laplacian eigenvalues. We show that by explicitly training networks that aim to translate between the learned latent codes and the spectral encoding, we can both recover a shape from its eigenvalues and moreover endow the latent space with certain desirable properties. Remarkably, our shape-from-spectrum solution is extremely efficient since it requires a single pass through a trained network, unlike expensive iterative optimization methods with ad-hoc regularizers [20]. Furthermore, our trainable module acts as a proxy to differentiable eigendecomposition, while encouraging



**Fig. 1.** Our spectral reconstruction approach enables correspondence-free style transfer applications. Given pose and style “donors” (left and middle columns respectively), we synthesize a new shape having the pose of the former and the style of the latter. The generation is driven by a learning-based eigenvalue alignment procedure (rightmost plots, colors correspond to shape names). Our approach does *not* require any input correspondence between pose and style target, it is robust to changes in resolution (middle row) and works across different representations (bottom row; the surface underlying the input point cloud is for visualization purposes only).

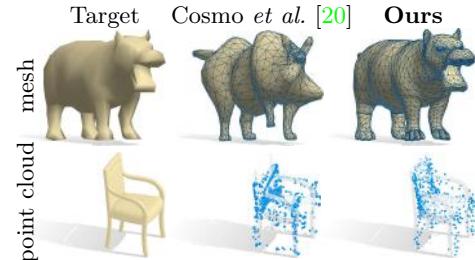
geometric consistency within the network. Overall, our key **contributions** can be summarized as follows:

- We propose the first learning-based model to robustly recover shape from Laplacian spectra *in a single pass*; our model is *end-to-end*, avoiding any iterative optimization or post-processing and so it enables interactive applications;
- For the first time, we provide a bidirectional linkage between learned 3D latent space and spectral geometric properties of 3D shapes;
- Our model is *general*, in that it applies with no variation to different shape classes even across different geometric representations and dimensions and to data that does not belong to the datasets used at training time;
- We showcase our approach in multiple applications, where we compare favorably to the state of the art; see Fig. 1 for an example.

## 2 Related work

Spectral quantities and in particular the eigenvalues of the Laplace-Beltrami operator provide a very informative summary of the intrinsic geometry that is well-established in geometry processing [30]. For example, closed-form estimates and analytical bounds for surface area, genus and curvature in terms of the Laplacian eigenvalues have been obtained [14]. By virtue of these properties, spectral shape analysis has been successfully exploited in many computer vision and computer graphics tasks such as shape retrieval and comparison [45,44], shape description and matching [50,3,11], mesh segmentation [43], sampling [37] and compression [27] among many others. Typically, the intrinsic properties of the shape are computed from its explicit representation and are used to encode compact geometric features invariant to isometric deformations, thus adopting a *data-reduction* principle.

Recently, several works have started to address the inverse problem: namely, recovering an extrinsic embedding from the intrinsic encoding [9,20]. This is closely related to the fundamental theoretical question of “hearing the shape of the drum” [26,22]. Although counterexamples have been proposed to show that isospectral shapes can be non-isometric, thus highlighting that in certain scenarios multiple shapes might have the same spectrum, there is recent work that proposes effective practical solutions to this problem. In [9] the shape-from-operator method was proposed aiming at obtaining the extrinsic shape from a Laplacian matrix where the 3D reconstruction was recovered after the estimation of Riemannian metric in terms of edge lengths. In [19] the intrinsic



**Fig. 2.** Comparison in shape from spectrum estimation between [20] (center) and our method (right) for a mesh and a point cloud. The target shapes are shown on the left. We refer to the supplementary materials for further comparisons and details.

and extrinsic relations of geometric objects have been extensively defined and evaluated from both theoretical and practical aspects. The authors revised the framework of functional shape differences [47] to account of extrinsic structure extending the reconstruction task to non-isometric shapes and models obtained from physical simulation and animation. Several works have also been proposed to recover shapes purely from Laplacian *eigenvalues* [16,1,38] or with mild additional information such as excitation amplitude in the case of musical key design [8]. Most closely related to ours in this area is the recent *isospectralization* approach introduced in [20], that aims directly to estimate the 3D shape from the spectrum by using a continuous optimization strategy combined with strong geometric regularizers. This approach works well in the vicinity of a good solution but is both computationally expensive and, as we show below, can quickly produce very unrealistic instances, failing in a large number of cases in 3D, as shown in Fig.2 for two examples (an animal and a chair).

In this paper we contribute to this line of work, and propose to replace the heuristics used in previous methods such as [20] with a purely data-driven approach for the first time. Our key idea is to design a deep neural network, that both constraints the space of solutions based on the set of shapes given at training, and at the same time, allows us to solve the isospectralization problem with a *single forward pass*, thus avoiding expensive and error-prone optimization.

We note that a related idea has been recently proposed in [25] via the so-called OperatorNet architecture. However, that work is based on shape difference operators [47] and as such requires a fixed source shape and functional maps to each shape in the dataset to properly synthesize a shape. Our approach is based on Laplacian eigenvalues alone and thus is completely correspondence-free.

Our approach also builds upon the recent work on learning generative shape models. A range of techniques have been proposed using the volumetric representations [51], point cloud auto-encoders [4,2], generative models based on meshes and implicit functions [49,23,33,29,15], and part structures [31,35,21,52], among many others.

Although generative models, and in particular auto-encoders, have shown impressive performance, the structure of the latent space is typically difficult to control or analyze directly. To address this problem, some methods proposed a disentanglement of the latent space [52,4] in order to split it in more semantic regions. Perhaps most closely related to ours in this domain, is the work in [4], where the shape spectrum is used to promote disentanglement of the latent space into intrinsic and extrinsic components, that can be controlled separately. Nevertheless, the resulting network does not allow to synthesize shapes from their spectra.

Extending the observations of these previous approaches, our work provides the first way to connect the learned latent space to the spectral space, thus inheriting the benefits and providing the versatility of moving across the two representations. Moreover, by combining the Laplacian spectra with a learned generative latent model, our network allows us not only to synthesize shapes from their spectra, but also to relate shapes with very different input structure. (*e.g.*,

meshes and point clouds) across a wide range of sampling densities, enabling several novel applications.

### 3 Background

In this work we model shapes as connected 2-dimensional Riemannian manifolds  $\mathcal{X}$  embedded in  $\mathbb{R}^3$ , possibly with boundary  $\partial\mathcal{X}$ , equipped with the standard metric. On each shape  $\mathcal{X}$  we consider its positive semi-definite Laplace-Beltrami operator  $\Delta_{\mathcal{X}}$ , generalizing the classical notion of Laplacian from the Euclidean setting to curved surfaces.

**Laplacian spectrum.**  $\Delta_{\mathcal{X}}$  admits an eigendecomposition

$$\Delta_{\mathcal{X}}\phi_i(x) = \lambda_i\phi_i(x) \quad x \in \text{int}(\mathcal{X}) \quad (1)$$

$$\langle \nabla\phi_i(x), \hat{n}(x) \rangle = 0 \quad x \in \partial\mathcal{X} \quad (2)$$

into eigenvalues  $\{\lambda_i\}$  and associated eigenfunctions  $\{\phi_i\}$ <sup>4</sup>.

The Laplacian eigenvalues of  $\mathcal{X}$  (its *spectrum*) form a discrete set, which is canonically ordered into a non-decreasing sequence

$$\text{Spec}(\mathcal{X}) := \{0 = \lambda_0 < \lambda_1 \leq \lambda_2 \leq \dots\}. \quad (3)$$

In the special case where  $\mathcal{X}$  is an interval in  $\mathbb{R}$ , the eigenvalues  $\lambda_i$  correspond to the (squares of) oscillation frequencies of Fourier basis functions  $\phi_i$ . This provides us with a connection to classical Fourier analysis, and with a natural notion of hierarchy induced by the ordering of the eigenvalues. In the light of this analogy, in practice, one is usually interested in a limited bandwidth consisting of the first  $k > 1$  eigenvalues; typical values in geometry processing applications range from  $k = 30$  to 100.

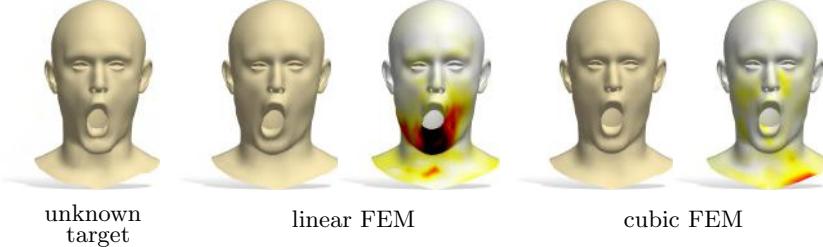
Furthermore, the spectrum is *isometry-invariant*, *i.e.*, it does not change with deformations of the shape that preserve geodesic distances (*e.g.*, changes in pose). This property is directly inherited from the isometry invariance of  $\Delta_{\mathcal{X}}$ , and it will play a key role in our learning model.

**Discretization.** In the discrete setting, we represent shapes as triangle meshes  $X = (V, T)$  with  $n$  vertices  $V$  and  $m$  triangular faces  $T$ ; depending on the application, we will also consider unorganized point clouds. Vertex coordinates in both cases are represented by a matrix  $\mathbf{X} \in \mathbb{R}^{n \times 3}$ .

The Laplace-Beltrami operator  $\Delta_{\mathcal{X}}$  is discretized as a  $n \times n$  matrix via the standard finite element method (FEM) [17]. In the simplest setting (*i.e.*, linear finite elements), this discretization corresponds to the cotangent Laplacian [39]; however, in this paper we use *cubic* FEM (see *e.g.*[43, Sec. 4.1] for a clear algorithmic treatment), since it yields a more accurate discretization as shown in Fig. 3. We remark that, differently from [20,41], this comes at virtually no additional cost for our pipeline, as we show in the sequel. On point clouds,  $\Delta_{\mathcal{X}}$  can be discretized using the approach described in [18,10].

---

<sup>4</sup> Similarly to [20] we use homogeneous Neumann boundary conditions; see Eq. (2), where  $\hat{n}(x)$  denotes the outward normal to the boundary.



**Fig. 3.** Reconstruction examples of our shape-from-spectrum pipeline. We show the results obtained with two different inputs: the eigenvalues of the Laplacian discretized with linear FEM, and those of the cubic FEM discretization. The heatmap encodes point-wise reconstruction error, growing from white to dark red.

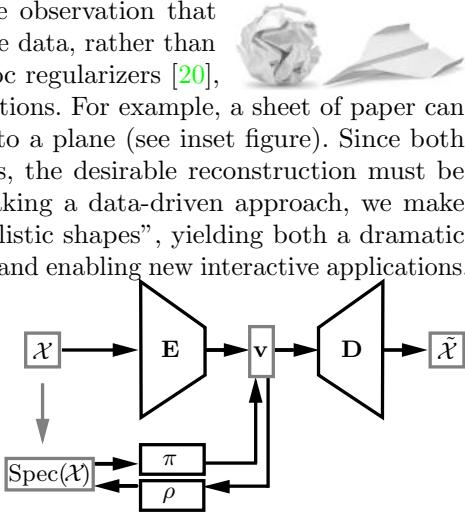
## 4 Method

We propose a deep learning model for recovering shapes from Laplacian eigenvalues. Our model operates in an end-to-end fashion: given a spectrum as input, it directly yields an embedded shape at the output with a single forward pass of the network, thus avoiding possibly expensive iterative optimization.

**Motivation.** Our rationale lies in the observation that shape semantics can be learned from the data, rather than by relying upon the definition of ad-hoc regularizers [20], often resulting in unrealistic reconstructions. For example, a sheet of paper can be *isometrically* crumpled or folded into a plane (see inset figure). Since both embeddings have the same eigenvalues, the desirable reconstruction must be imposed in the form of a prior. By taking a data-driven approach, we make our method aware of the “space of realistic shapes”, yielding both a dramatic improvement in accuracy and efficiency, and enabling new interactive applications.

**Latent space connections.** Our key idea is to construct an auto-encoder (AE) neural network architecture, augmented by explicitly modeling the connections between the latent space of the AE and the Laplacian spectrum of the input shape; see Fig. 4 for an illustration of our learning model.

Loosely speaking, our approach can be seen as implementing a coupling between two latent spaces – a learned one that operates directly on the shape embedding  $\mathcal{X}$ , and the one provided by the eigenvalues  $\text{Spec}(\mathcal{X})$ . Note that in the former case, the *encoder*  $E$  is trainable, whereas the mapping  $\mathcal{X} \rightarrow \text{Spec}(\mathcal{X})$  is



**Fig. 4.** Our network model. The input shape  $\mathcal{X}$  and its Laplacian spectrum  $\text{Spec}(\mathcal{X})$  are passed, respectively, through an AE enforcing  $\mathcal{X} \approx \tilde{\mathcal{X}}$ , and an invertible module  $(\pi, \rho)$  mapping the eigenvalue sequence to a latent vector  $\mathbf{v}$ . The two branches are trained simultaneously, forcing  $\mathbf{v}$  to be updated accordingly. The trained model allows to recover the shape purely from its eigenvalues via the composition  $D(\pi(\text{Spec}(\mathcal{X}))) \approx \mathcal{X}$ .

provided via the eigen-decomposition and thus fixed a priori. Finally, we introduce the two coupling mappings  $\pi, \rho$ , trained with a bidirectional loss, to both enable communication across the two latent spaces and to tune the learned latent space by endowing it with structure contained in  $\text{Spec}(\mathcal{X})$ .

We phrase our overall training loss as follows:

$$\ell = \ell_{\mathcal{X}} + \alpha \ell_{\lambda}, \quad \text{with} \quad (4)$$

$$\ell_{\mathcal{X}} = \frac{1}{n} \|D(E(\mathbf{X})) - \mathbf{X}\|_F^2 \quad (5)$$

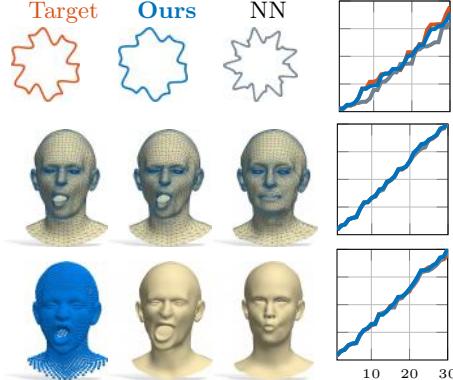
$$\ell_{\lambda} = \frac{1}{k} (\|\pi(\boldsymbol{\lambda}) - E(\mathbf{X})\|_2^2 + \|\rho(E(\mathbf{X})) - \boldsymbol{\lambda}\|_2^2) \quad (6)$$

where  $\boldsymbol{\lambda}$  is a vector containing the first  $k$  eigenvalues in  $\text{Spec}(\mathcal{X})$ ,  $\|\cdot\|_F$  denotes the Frobenius norm, and  $\alpha = 10^{-4}$  controls the relative strengths of the reconstruction loss  $\ell_{\mathcal{X}}$  and the spectral term  $\ell_{\lambda}$ . The blocks  $D$ ,  $E$ ,  $\pi$ , and  $\rho$  are learnable and parametrized by a neural network (see below for details). Also observe that Eq. (6) enforces  $\rho \approx \pi^{-1}$ ; in other words,  $\pi$  and  $\rho$  form a translation block between the latent vector and the spectral encoding of the shape.

**Remark.** Our architecture takes  $\text{Spec}(\mathcal{X})$  as an input, *i.e.*, the eigenvalues are *not* computed at training time. By learning an *invertible* mapping to the latent space, we avoid expensive backpropagation steps through the spectral decomposition of the Laplacian  $\Delta_{\mathcal{X}}$ . In this sense, the mapping  $\rho$  acts as an efficient proxy to differentiable eigendecomposition, which we exploit in several applications below.

The latter remark also highlights that since eigenvalue computation is only incurred as an offline cost, it can be performed with arbitrary accuracy (we use cubic FEM, see Fig. 3) without noticeable efficiency cost.

**Implementation details.** Our proposed model can be easily adapted to different shape representations simply by changing the AE architecture; in this paper we investigate 3D meshes and unorganized point clouds. In our experiments, we only adopt a single simple AE focusing on the power of the proposed bidirectional mapping between the latent space and the spectrum. Future work could exploit our method by adopting more complex generative methods. The latent space dimension is fixed to 30 (the same as  $k$ ). We refer to



**Fig. 5.** Examples of shape reconstruction from eigenvalues using our data-driven approach on three different representations (top to bottom: 2D contours, 3D meshes and point clouds). The eigenvalues of the shapes on the left are given as input to our network, which generates the shapes in the middle column. For each representation, the eigenvalues are computed on the appropriate discretization of the Laplacian as per Sec. 3. The NN column shows the nearest-neighbor solution sought in the training set.

the supplementary material for details about the architecture, both in the case of meshes and point clouds.

## 5 Analysis

**Shape representation.** To better reflect different practical settings, in this paper we make the following distinction: triangle meshes are assumed to be in point-to-point correspondence *at training time* (typical setting in graphics and geometry processing), while unorganized point clouds are *not* assumed to have a consistent vertex labeling (typical in 3D computer vision).

In Fig. 5 we show an example of shape-from-spectrum reconstruction for each case, demonstrating robustness across different representations. For the sake of illustration, similarly to [20,41] we also include 2D contours among the examples, discretized as regular cycle graphs. We refer to Sec. 6 for details on the datasets, and for more specific applications with meshes and point clouds.

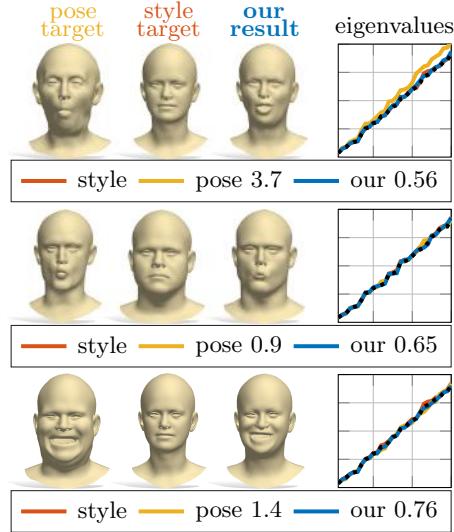
**Comparison.** We trained two separate architectures (with and without the  $\rho$  block) and compared them, in terms of reconstruction accuracy, to the nearest-neighbors baseline as well as with the state-of-the-art isospectralization method of Cosmo *et al.* [20]. Measuring performance without the  $\rho$  block is an ablation study we carry out to validate the importance of having an *invertible* module connecting the spectral encoding to the learned latent codes. As the results suggest, this network component both contributes to reducing the reconstruction error, and it enables novel applications as we will see in Sec. 6.

In Table 1 we report the results as the mean squared error between the reconstructed shape and the ground-truth. For this evaluation, we trained our model on 1,853 3D shapes from the COMA dataset [42] of human faces; 100 shapes of an unseen subject are used for the test set. We repeated this test at four different mesh resolutions:  $\sim 4K$  (full resolution), 1K, 500 and 200 vertices respectively. For each resolution, we independently compute the Laplacian spectrum and use these spectra to recover the shape.

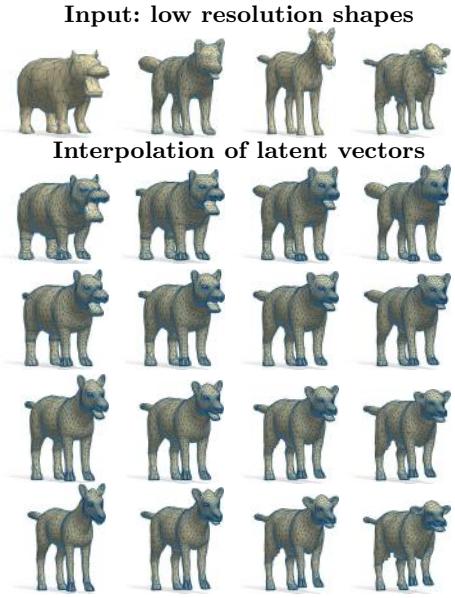
**Spectral bandwidth** (the number  $k$  of input eigenvalues) has a direct effect on reconstruction accuracy, since increasing this number brings in more high-frequency detail into the representation. This arises naturally from the Fourier-like, multi-scale nature of the spectrum. Following prior work [20,41,46], in all our experiments we use  $k = 30$  eigenvalues. In the supplementary material we additionally report results with different choices of  $k$ .

	full res	1000	500	200
<b>Ours</b>	<b>1.61</b>	<b>1.62</b>	<b>1.71</b>	<b>2.13</b>
Ours without $\rho$	1.89	1.82	2.06	2.42
NN	4.45	4.63	4.01	2.65
Cosmo <i>et al.</i> [20]	—	16.4	7.11	4.08

**Table 1.** Shape-from-spectrum reconstruction comparisons with the NN baseline (nearest neighbors between spectra) and a state of the art spectral reconstruction approach; we report average error over 100 shapes of an unseen subject from the COMA dataset [42]. Best results (in bold) are obtained with our full pipeline. ‘—’ denotes out of memory; all errors must be rescaled by  $10^{-5}$ .



**Fig. 6.** Examples of style transfer. The target style (middle) is applied to the target pose (left) by solving problem (7) and then decoding the resulting latent vector, obtaining the result shown on the right. For each example we also report the corresponding eigenvalue alignment (rightmost plots). The black dotted line is the image of  $\rho$ . The numbers in the legend denote the distance from the target “style” spectrum to the source pose and to our generated shape; a small number suggests that the generated shape is a near-isometry of the style target.



**Fig. 7.** Latent space interpolation of four low-resolution shapes with different mesh connectivity (top row, unseen at training time). The spectra of the input shapes are mapped via  $\pi$  to the latent space, where they are bilinearly interpolated and then decoded to  $\mathbb{R}^3$ . The reconstructions of the four input shapes are depicted at the four corners of the  $4 \times 4$  grid.

## 6 Applications

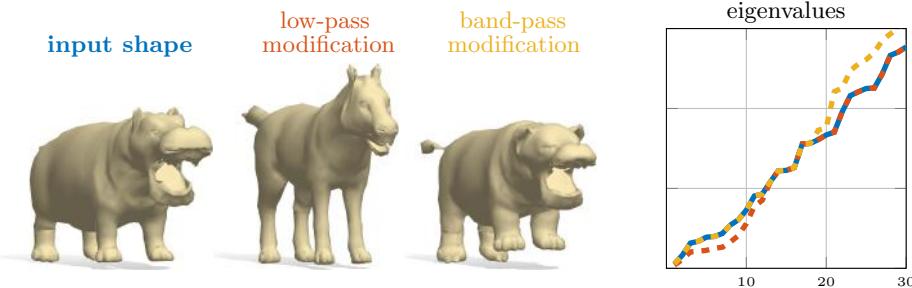
We showcase the application of our generative model on multiple tasks. As we show empirically, our data-driven method bridges the gap between previous spectral generation approaches and downstream applications. Due to the limited space, we collect in the supplementary materials a complete description of the training and test sets and the parameters selected in our experiments.

### 6.1 Style transfer

As shown in Fig. 1, we can use our trained network to transfer the style of a shape  $\mathcal{X}_{\text{style}}$  to another shape  $\mathcal{X}_{\text{pose}}$  having both a different style and pose. This is done by a search in the latent space, phrased as the optimization problem:

$$\min_{\mathbf{v}} \|\text{Spec}(\mathcal{X}_{\text{style}}) - \rho(\mathbf{v})\|_2^2 + w \|\mathbf{v} - E(\mathcal{X}_{\text{pose}})\|_2^2 \quad (7)$$

Here, the first term seeks a latent vector whose associated spectrum aligns with the eigenvalues of  $\mathcal{X}_{\text{style}}$ ; in other words, we regard style as an intrinsic



**Fig. 8.** Exploring the space of shapes in real time via direct manipulation of the spectrum. The low-pass modification (middle) corresponds to a decrease of the first 12 eigenvalues of the input shape; the band-pass modification (right) is obtained by amplifying the last 12 eigenvalues (within the window plotted on the far right). The observed effect reflects intuition: the damping of low eigenvalues leads to the emergence of more pronounced geometric features (such as longer legs and snout), while amplification of mid-range eigenvalues affect the high-frequency details (note the change in the ears and fingers); see the supplementary video for a wall-clock demo of this procedure.

property of the shape, considering the spectral representation invariant to pose deformations. The second term keeps the latent vector close to that of the input pose (we initialize with  $\mathbf{v}_{\text{init}} = E(\mathcal{X}_{\text{pose}})$ ). We carry out the optimization problem by back-propagating the gradient of the cost function of Eq. (7) with respect to  $\mathbf{v}$  through  $\rho$ . The sought shape is then given by a forward pass on the resulting minimizer. In Fig. 6 we show several examples (others can be found in the supplementary material).

We emphasize here that the style is purely encoded in the input eigenvalues, therefore it does not rely on the test shapes being in point-to-point correspondence with the training set. This leads to the following:

*Property 1.* Our method can be used in a **correspondence-free** scenario. By taking eigenvalues as input, it enables applications that traditionally require a correspondence, but side-steps this requirement.

This observation was also mentioned in other spectrum-based approaches [20, 41]. However, the data-driven nature of our method makes it more robust, efficient and accurate, therefore greatly improving its practical utility.

## 6.2 Shape exploration

The results of Sec. 6.1 suggest that eigenvalues can be used to drive the exploration of the AE’s latent space toward a desired direction. Another possibility is to regard *the eigenvalues themselves* as a parametric model for isometry classes, and explore the “space of spectra” in a similar manner as is typically done with latent spaces. Our bi-directional coupling between spectra and latent codes makes this exploration feasible, as remarked by the following property:

*Property 2.* Latent space connections provide both a means for **controlling** the latent space, and vice-versa, enable **exploration** of the space of Laplacian spectra.

Since eigenvalues change continuously with the manifold metric [5], a small variation in the spectrum will give rise to a small change in the geometry. We can visualize such variations in shape directly, by first deforming a given spectrum (*e.g.*, by a simple linear interpolation between two spectra) to obtain the new eigenvalue sequence  $\mu$ , and then directly computing  $D(\pi(\mu))$ ; due to lack of space, we report these results in the supplementary material.

In Fig. 7 we show a related experiment. Here we train the network on 4,430 animal meshes generated with the SMAL parametric model following the official protocol [54]. Given four *low-resolution* shapes  $\mathcal{X}_i$  as input, we first compute their spectra  $\text{Spec}(\mathcal{X}_i)$ , map these to the latent space via  $\pi(\text{Spec}(\mathcal{X}_i))$ , perform a bilinear interpolation of the resulting latent vectors, and finally reconstruct the corresponding shapes.

Finally, in Fig. 8 we show an example of interactive spectrum-driven shape exploration. Given a shape and its Laplacian eigenvalues as input, we navigate the space of shapes by directly modifying different frequency bands with the aid of a simple user interface. The modified spectra are then decoded by our network in *real time*. We refer to the video in the supplementary material for a further demonstration of this interactive process.

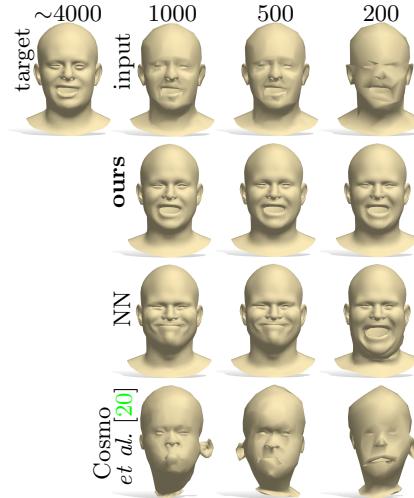
### 6.3 Super-resolution

A key feature that emerges from the experiment in Fig. 7 is the perfect reconstruction of the low-resolution shapes once their eigenvalues are mapped to the latent space via  $\pi$ . This brings us to a fundamental property of our approach:

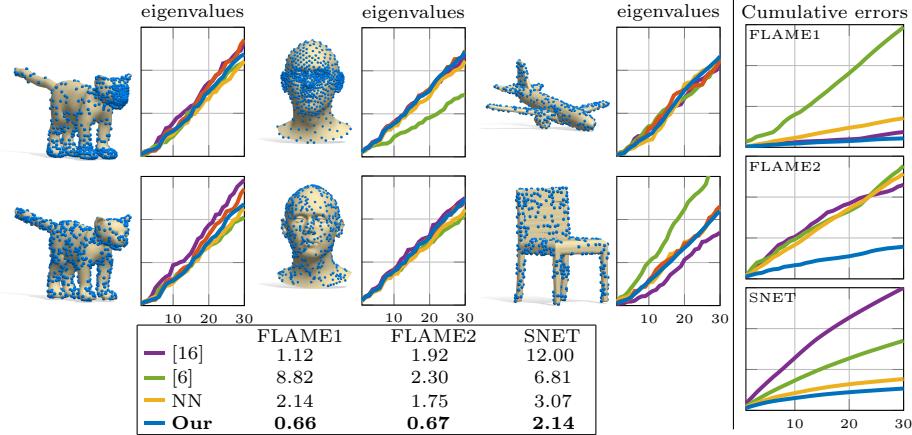
*Property 3.* Since eigenvalues are largely **insensitive to mesh resolution and sampling**, so is our trained network model.

This fact is especially evident when using cubic FEM discretization, as we do in all our tests, since it more closely approximates the continuous setting and is thus much less affected by the surface discretization.

**Remark.** It is worth mentioning here that existing methods can in principle employ cubic FEM as well; however, this soon becomes prohibitively expensive due to the *explicit* differentiation of spectral decomposition required by their optimization procedures [20,41].



**Fig. 9.** Mesh super-resolution for input shapes at decreasing resolution (top row, left to right). Our solutions match closely with the original high-resolution version of the input shapes (top left), while other approaches either predict the wrong pose (NN baseline) or generate an unrealistic shape (Cosmo *et al.*).



**Fig. 10.** Qualitative and quantitative evaluation of point cloud spectra estimation. On the left we show the qualitative comparison for different samplings distributions on three classes (animals, human faces and objects). We show the eigenvalues estimated by each method alongside the input point cloud, plotted as surface samplings for better visualization, and the ground truth spectrum (in red). On the last column, we report the average cumulative error curves evaluated on the FLAME dataset for the two different sample distributions (FLAME1 and FLAME2) and on ShapeNet dataset (SNET).

These properties allow us to use our network for the task of mesh super-resolution. Given a low-resolution mesh as input, our aim is to recover a higher resolution counterpart of the mesh. Furthermore, while the input mesh has *arbitrary* resolution and is unknown to the network (and in particular, a correspondence with the training models is *not* given), an additional desideratum is for the new shape to be in dense point-to-point correspondence with models from the training set. We do so in a single shot, by directly predicting the decoded shape as:

$$\mathcal{X}_{\text{hires}} = D(\pi(\text{Spec}(\mathcal{X}_{\text{lowres}}))). \quad (8)$$

As is clear from the equation, this generative model fully exploits the geometric information encoded in the spectrum.

In Fig. 9 we show a comparison with nearest-neighbors between eigenvalues (with shapes within the training set), and the isospectralization method of Cosmo *et al.* [20]. We observe that our solution closely reproduces the high-resolution target. Isospectralization does correctly align the eigenvalues – yet, it recovers unrealistic shapes in the process, due to ineffective regularization. This phenomenon highlights another key property of our method:

*Property 4.* Our data-driven approach replaces ad-hoc regularizers, that are difficult to model axiomatically, with **realistic priors** learned from examples.

This is especially important when dealing with deformable objects, where shapes falling into the same isometry class are often hard to disambiguate without using sophisticated geometric priors.

#### 6.4 Estimating point cloud spectra

As an additional experiment, we show how our network can be used to directly predict Laplacian eigenvalues for unorganized point clouds. This task is particularly challenging due to the lack of a structure in the point set, and existing approaches such as [18,6] often fail at approximating the eigenvalues of the underlying (unknown) surface accurately. The difficulty is even more pronounced when the point sets are irregularly sampled, as we empirically show here.

In our case, estimation of the spectrum for a given input point cloud  $\mathcal{X}$ , simply boils down to computing the single forward pass:

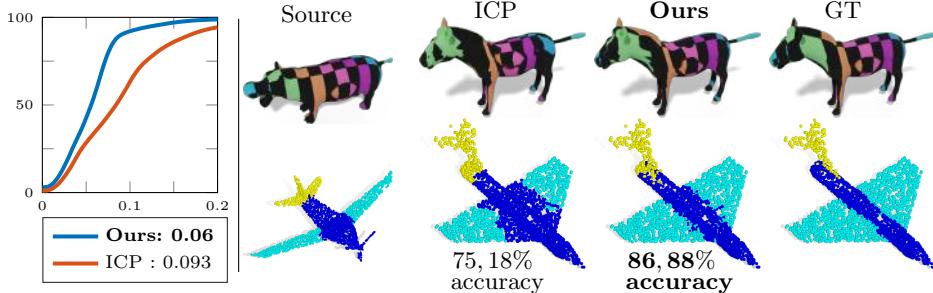
$$\widetilde{\text{Spec}}(\mathcal{X}) = \rho(E(\mathcal{X})). \quad (9)$$

To address this task we train our network by feeding unorganized point clouds as input, together with the spectra computed from the corresponding meshes (which are available at training time). As described in the supplementary materials, for this setting we use a PointNet [40] encoder and a fully connected decoder, and we replace the reconstruction loss of Eq. (5) with the Chamfer distance (which accounts for the lack of a consistent vertex ordering). This application evidences the **generality** of our model, which can accommodate different representations of geometric data, and does not strictly depend on a specific auto-encoder.

We consider two types of point clouds: (1) with similar point density and regularity as in the training set, and (2) with randomized non-uniform sampling. All point clouds have a number of samples equal to around 20% of the mesh vertices at full resolution for heads and animals, while objects have different resolutions. On the left of Fig. 10, we depict a qualitative comparison in point cloud spectrum estimation. We compare the spectrum that we estimate via  $\rho(E(\mathcal{X}))$  with axiomatic methods [18,6], and with the nearest-neighbor baseline (applied in the latent space). These qualitative results are obtained by training on the SMAL [54] (left), COMA [42] (middle) and ShapeNet watertight version [24] (right). We remark that to underline its generalization capability, the network trained on COMA is tested on point clouds from the FLAME dataset, while on Shapenet we consider 4 different classes (airplanes, boats, screens and chairs). We compute the cumulative error curves of the distance between the eigenvalues from the meshes corresponding to the test point clouds and the estimated spectrum in three cases: 393 generated point clouds from the FLAME dataset at two different sample distributions, and 400 generated point clouds from the ShapeNet dataset among the 4 classes. The average curves are shown on the right of Fig. 10; the mean error is reported in the legend for a direct comparison.

#### 6.5 Matching from spectrum

Finally, we used the same models presented in the previous application to obtain a matching between shape pairs using only their spectrum and regardless of their representation. Given a pair of models, we feed their eigenvalues into our network. The 500 output points are naturally ordered by the decoder, and we exploit this to establish a sparse correspondence. In the case of meshes, we extend it to a



**Fig. 11.** On the left, quantitative evaluation of matching [28] between 100 pairs of animals. On the right, the qualitative comparison on texture and segmentation transfer.

dense one by using functional maps framework [36]. In the case of point clouds, we can propagate a semantic segmentation using nearest-neighbor. We perform a quantitative evaluation on the SMAL [54] dataset, testing on 100 non-isometric meshes pairs of animals from different classes. Two applications that benefit from our approach are texture and segmentation transfer; we tested them respectively on animals and segmented ShapeNet [53]. The comparison baseline consists of 100 iterations of Iterative Closest Point [7] (ICP) to rigidly align the two shapes followed by nearest-neighbour assignment as correspondence. The results are reported in Fig. 11. For further details, we refer to the supplementary materials.

## 7 Conclusions

We introduced the first data-driven method for shape generation from Laplacian spectra. Our approach consists in enriching a standard AE with a pair of cycle-consistent maps, associating ordered sequences of eigenvalues to latent codes and vice-versa. This explicit coupling brings forth key advantages of spectral methods to generative models, enabling novel applications and leading to a gap in accuracy when compared to competing approaches.

**Limitations.** The main limitation of our method, which we share with spectral methods, is the limited robustness of current Laplacian discretization algorithms for poor-quality meshes, which makes it difficult to compute eigenvalues reliably in all situations. Adopting the recent approach [48] for such borderline cases is a promising possibility.

**Future work.** Among the open directions that we plan to explore is the application of similar ideas to other functional operators. While the Laplacian remains a classical choice due to its Fourier-like properties, spectra of other operators exhibiting different desirable properties may lead to applications in other practical settings. It would also be interesting to estimate and exploit spectral quantities from other representations such as images or noisy volumetric data.

## Acknowledgments

We gratefully acknowledge Luca Moschella and Silvia Casola for the technical support, Nicholas Sharp for the valuable suggestions about pointcloud spectra

estimation. Parts of this work were supported by the KAUST OSR Award No. CRG-2017-3426, the ERC Starting Grant No. 758800 (EXPROTEA), the ERC Starting Grant No. 802554 (SPECGEO), and the MIUR under grant "Dipartimenti di eccellenza 2018-2022" of the Department of Computer Science of Sapienza University and University of Verona.

## References

1. Aasen, D., Bhamre, T., Kempf, A.: Shape from sound: toward new tools for quantum gravity. *Physical review letters* **110**(12), 121301 (2013)
2. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.: Learning representations and generative models for 3d point clouds. In: International Conference on Machine Learning. pp. 40–49 (2018)
3. Aubry, M., Schlickewei, U., Cremers, D.: The wave kernel signature: A quantum mechanical approach to shape analysis. In: Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on. pp. 1626–1633. IEEE (2011)
4. Aumentado-Armstrong, T., Tsogkas, S., Jepson, A., Dickinson, S.: Geometric disentanglement for generative latent shape models. In: International Conference on Computer Vision (ICCV) (2019)
5. Bando, S., Urakawa, H.: Generic properties of the eigenvalue of the laplacian for compact riemannian manifolds. *Tohoku Mathematical Journal, Second Series* **35**(2), 155–172 (1983)
6. Belkin, M., Sun, J., Wang, Y.: Constructing laplace operator from point clouds in rd. In: Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms. pp. 1031–1040. Society for Industrial and Applied Mathematics (2009)
7. Besl, P.J., McKay, N.D.: A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**(2), 239–256 (Feb 1992)
8. Bharaj, G., Levin, D.I., Tompkin, J., Fei, Y., Pfister, H., Matusik, W., Zheng, C.: Computational design of metallophone contact sounds. *ACM Transactions on Graphics (TOG)* **34**(6), 223 (2015)
9. Boscaini, D., Eynard, D., Kourounis, D., Bronstein, M.M.: Shape-from-operator: recovering shapes from intrinsic operators. *Computer Graphics Forum* **34**(2), 265–274 (2015)
10. Boscaini, D., Masci, J., Rodolà, E., Bronstein, M.M., Cremers, D.: Anisotropic diffusion descriptors. *Computer Graphics Forum* **35**(2), 431–441 (2016)
11. Bronstein, A.M., Bronstein, M.M., Guibas, L.J., Ovsjanikov, M.: Shape google: Geometric words and expressions for invariant shape retrieval. *ACM Transactions on Graphics (TOG)* **30**(1), 1 (2011)
12. Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P.: Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine* **34**(4), 18–42 (2017)
13. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: ShapeNet: An Information-Rich 3D Model Repository. Tech. Rep. arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago (2015)
14. Chavel, I.: Eigenvalues in Riemannian Geometry. Academic Press (1984)
15. Chen, Z., Zhang, H.: Learning implicit fields for generative shape modeling. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5939–5948 (2019)

16. Chu, M., Golub, G., Golub, G.H.: Inverse eigenvalue problems: theory, algorithms, and applications, vol. 13. Oxford University Press (2005)
17. Ciarlet, P.G.: The finite element method for elliptic problems, vol. 40. Siam (2002)
18. Clarenz, U., Rumpf, M., Telea, A.: Finite elements on point based surfaces. In: Proceedings of the First Eurographics conference on Point-Based Graphics. pp. 201–211. Eurographics Association (2004)
19. Corman, E., Solomon, J., Ben-Chen, M., Guibas, L., Ovsjanikov, M.: Functional Characterization of Intrinsic and Extrinsic Geometry. ACM Transactions on Graphics **17** (2017)
20. Cosmo, L., Panine, M., Rampini, A., Ovsjanikov, M., Bronstein, M.M., Rodolà, E.: Isospectralization, or how to hear shape, style, and correspondence. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7529–7538 (2019)
21. Gao, L., Yang, J., Wu, T., Yuan, Y.J., Fu, H., Lai, Y.K., Zhang, H.: Sdm-net: Deep generative network for structured deformable mesh. arXiv preprint arXiv:1908.04520 (2019)
22. Gordon, C., Webb, D.L., Wolpert, S.: One cannot hear the shape of a drum. Bulletin of the American Mathematical Society **27**(1), 134–138 (1992)
23. Groueix, T., Fisher, M., Kim, V.G., Russell, B., Aubry, M.: AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In: Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2018)
24. Huang, J., Su, H., Guibas, L.: Robust watertight manifold surface generation method for shapenet models. arXiv preprint arXiv:1802.01698 (2018)
25. Huang, R., Rakotosaona, M.J., Achlioptas, P., Guibas, L., Ovsjanikov, M.: Operatornet: Recovering 3d shapes from difference operators. In: ICCV (2019)
26. Kac, M.: Can one hear the shape of a drum? The american mathematical monthly **73**(4P2), 1–23 (1966)
27. Karni, Z., Gotsman, C.: Spectral compression of mesh geometry. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques. pp. 279–286. SIGGRAPH '00, ACM Press/Addison-Wesley Publishing Co. (2000)
28. Kim, V.G., Lipman, Y., Funkhouser, T.: Blended intrinsic maps. In: ACM Transactions on Graphics (TOG). vol. 30, p. 79. ACM (2011)
29. Kostrikov, I., Jiang, Z., Panozzo, D., Zorin, D., Bruna, J.: Surface networks. In: Proc. CVPR (2018)
30. Levy, B., Zhang, R.H.: Spectral geometry processing. In: ACM SIGGRAPH Course Notes (2010)
31. Li, J., Xu, K., Chaudhuri, S., Yumer, E., Zhang, H., Guibas, L.: Grass: Generative recursive autoencoders for shape structures. ACM Transactions on Graphics (Proc. of SIGGRAPH 2017) **36**(4), 52–56 (2017)
32. Li, T., Bolkart, T., Black, M.J., Li, H., Romero, J.: Learning a model of facial shape and expression from 4D scans. ACM Transactions on Graphics, (Proc. SIGGRAPH Asia) **36**(6) (2017)
33. Litany, O., Bronstein, A., Bronstein, M., Makadia, A.: Deformable shape completion with graph convolutional autoencoders. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1886–1895 (2018)
34. Masci, J., Rodolà, E., Boscaini, D., Bronstein, M.M., Li, H.: Geometric deep learning. In: SIGGRAPH ASIA 2016 Courses. p. 1. ACM (2016)
35. Mo, K., Guerrero, P., Yi, L., Su, H., Wonka, P., Mitra, N., Guibas, L.J.: Structurenet: Hierarchical graph networks for 3d shape generation. arXiv preprint arXiv:1908.00575 (2019)

36. Ovsjanikov, M., Ben-Chen, M., Solomon, J., Butscher, A., Guibas, L.: Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)* **31**(4), 30:1–30:11 (2012)
37. Öztireli, C., Alexa, M., Gross, M.: Spectral sampling of manifolds. *ACM Transactions on Graphics (TOG)* **29**(6), 168 (2010)
38. Panine, M., Kempf, A.: Towards spectral geometric methods for euclidean quantum gravity. *Physical Review D* **93**(8), 084033 (2016)
39. Pinkall, U., Polthier, K.: Computing discrete minimal surfaces and their conjugates. *Experimental mathematics* **2**(1), 15–36 (1993)
40. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 652–660 (2017)
41. Rampini, A., Tallini, I., Ovsjanikov, M., Bronstein, A.M., Rodolà, E.: Correspondence-free region localization for partial shape similarity via hamiltonian spectrum alignment. In: *International Conference on 3D Vision (3DV)* (2019)
42. Ranjan, A., Bolkart, T., Sanyal, S., Black, M.J.: Generating 3D faces using convolutional mesh autoencoders. In: *European Conference on Computer Vision (ECCV)* (2018)
43. Reuter, M.: Hierarchical shape segmentation and registration via topological features of laplace-beltrami eigenfunctions. *International Journal of Computer Vision* **89**(2-3), 287–308 (2010)
44. Reuter, M., Wolter, F.E., Peinecke, N.: Laplace-spectra as fingerprints for shape matching. In: *Proceedings of the 2005 ACM symposium on Solid and physical modeling*. pp. 101–106. ACM (2005)
45. Reuter, M., Wolter, F.E., Peinecke, N.: Laplace–beltrami spectra as shape-dnaof surfaces and solids. *Computer-Aided Design* **38**(4), 342–366 (2006)
46. Roufosse, J.M., Sharma, A., Ovsjanikov, M.: Unsupervised deep learning for structured shape matching. In: *Proceedings of the IEEE International Conference on Computer Vision*. pp. 1617–1627 (2019)
47. Rustamov, R.M., Ovsjanikov, M., Azencot, O., Ben-Chen, M., Chazal, F., Guibas, L.: Map-based exploration of intrinsic shape differences and variability. *ACM Transactions on Graphics (TOG)* **32**(4) (2013)
48. Sharp, N., Soliman, Y., Crane, K.: Navigating intrinsic triangulations. *ACM Trans. Graph.* **38**(4), 55:1–55:16 (Jul 2019)
49. Sinha, A., Unmesh, A., Huang, Q.X., Ramani, K.: SurfNet: Generating 3d shape surfaces using deep residual networks. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 791–800 (2017)
50. Sun, J., Ovsjanikov, M., Guibas, L.: A concise and provably informative multi-scale signature based on heat diffusion. *Computer Graphics Forum* **28**(5), 1383–1392 (2009)
51. Wu, J., Zhang, C., Xue, T., Freeman, W.T., Tenenbaum, J.B.: Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In: *Advances in Neural Information Processing Systems*. pp. 82–90 (2016)
52. Wu, Z., Wang, X., Lin, D., Lischinski, D., Cohen-Or, D., Huang, H.: Sagnet: Structure-aware generative network for 3d-shape modeling. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2019)* **38**(4), 91:1–91:14 (2019)
53. Yi, L., Shao, L., Savva, M., Huang, H., Zhou, Y., Wang, Q., Graham, B., Engelcke, M., Klokov, R., Lempitsky, V., et al.: Large-scale 3d shape reconstruction and segmentation from shapenet core55. *arXiv preprint arXiv:1710.06104* (2017)

54. Zuffi, S., Kanazawa, A., Jacobs, D., Black, M.J.: 3D menagerie: Modeling the 3D shape and pose of animals. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (Jul 2017)

# SUPPLEMENTARY

## Instant recovery of shape from spectrum via latent space connections

Riccardo Marin<sup>1</sup>, Arianna Rampini<sup>2</sup>, Umberto Castellani<sup>1</sup>,  
Emanuele Rodolà<sup>2</sup>, Maks Ovsjanikov<sup>3</sup>, Simone Melzi<sup>3</sup>

<sup>1</sup> University of Verona

<sup>2</sup> Sapienza University of Rome

<sup>3</sup> LIX, Ecole Polytechnique

**Abstract.** In this document we collect some additional details about the proposed method, architecture and results, that due to lack of space were not included in the main manuscript.

### 1 Architecture

**Meshes.** When  $\mathcal{X}$  is discretized as a mesh (typical in graphics and geometry processing applications), the AE architecture is a multilayer perceptron (MLP) with a bottleneck and dimensions:  $n \times 3 \rightarrow 300 \rightarrow 200 \rightarrow 30 \rightarrow 200 \rightarrow n \times 3$ ; each layer, except for the last one, is followed by tanh activation. The maps  $\pi$  and  $\rho$  are both parametrized by a MLP with dimensions:  $k \rightarrow 80 \rightarrow 160 \rightarrow 320 \rightarrow 640 \rightarrow 320 \rightarrow 160 \rightarrow 80 \rightarrow k$ ; each layer, except for the last one, is followed by batch norm and SeLu activation. We use batch size 16, and the Adam optimizer with learning rate equal to  $10^{-4}$ . Note that we also use this architecture for 2D contours, which are discretized as (2-regular) cycle graphs.

**Unorganized point clouds.** In this case we do *not* assume the training data to have a consistent vertex labeling. To tackle this setting, we use a PointNet [40] encoder and a fully connected decoder<sup>4</sup>; we use the MLP from the mesh case for the decoder. The encoder consists in a shared MLP network on each point with layer output sizes 64 and 128; each layer is followed by batch norm. A maxpool layer is then used to output a 128-dimensional vector, which is reduced with a tanh-activated MLP to dimensions:  $128 \rightarrow 64 \rightarrow 30$ .

After acceptance, we will release a full implementation of the proposed architecture and the link to the data that we used in our experiments.

---

<sup>4</sup> As suggested by the authors of [40] in <https://github.com/charlesq34/pointnet-autoencoder>

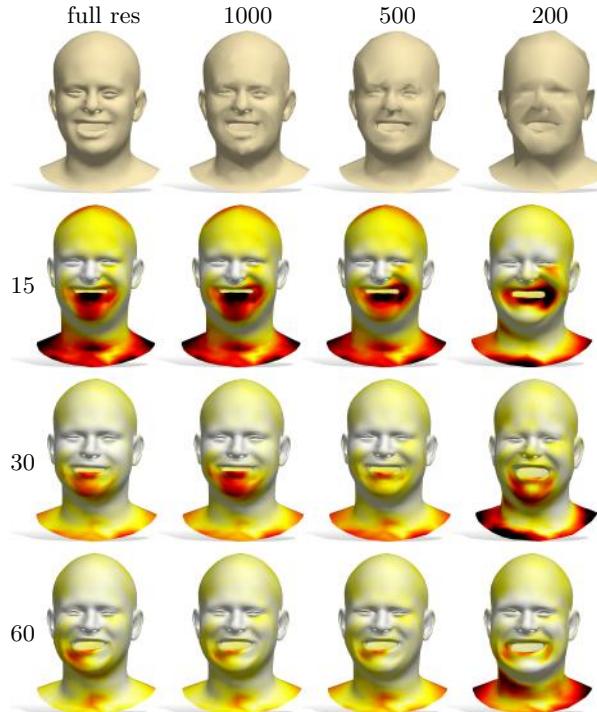
## 2 Analysis of increasing bandwidth ( $k$ )

We performed an experimental analysis for different values of  $k$  (the number of Laplacian eigenvalues given as input). This parameter affects both the amount of information used by the network (*i.e.*, more eigenvalues) and the capability of the learned representation (*i.e.*, higher-dimensional latent space). We evaluate our model with  $k = 15$ ,  $k = 30$  and  $k = 60$  on the shape-from-spectrum reconstruction task, using exactly the same dataset and experimental setup as in Table 1 of the main manuscript.

The quantitative results are reported in Table 1. While cutting  $k$  by half produces a sensible performance degradation, we observed that doubling it achieves just slightly better results. For this reason, in our experiments we settled for a value of  $k = 30$ . From a qualitative perspective, our results suggest that without changing other architecture details, a larger value for  $k$  can achieve better precision on the high-frequency details (*e.g.*, the pose of the mouth); as shown in Fig. 1.

	full res	1000	500	200
$k = 30$	1.61	1.62	<b>1.71</b>	2.13
$k = 15$	3.74	3.78	3.72	3.59
$k = 60$	<b>1.60</b>	<b>1.52</b>	1.79	<b>2.05</b>

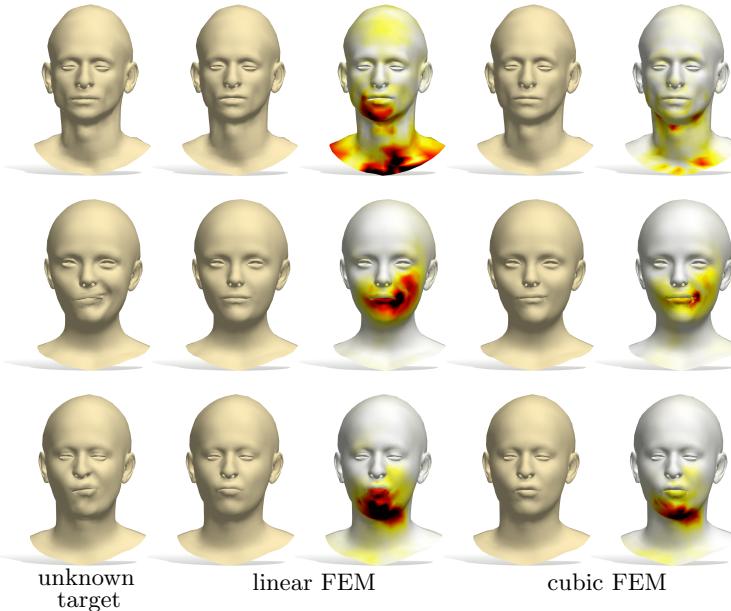
**Table 1.** Shape-from-spectrum reconstruction comparisons with different  $k$  trained for the same number of epochs; we report average error over 100 shapes of an unseen subject from the COMA dataset [42]. All errors must be rescaled by  $10^{-5}$ .



**Fig. 1.** Examples of shape-from-spectrum reconstructions at different values for the spectral bandwidth  $k$ ; see Table 1 for a quantitative evaluation.

### 3 Laplace-Beltrami operator discretization

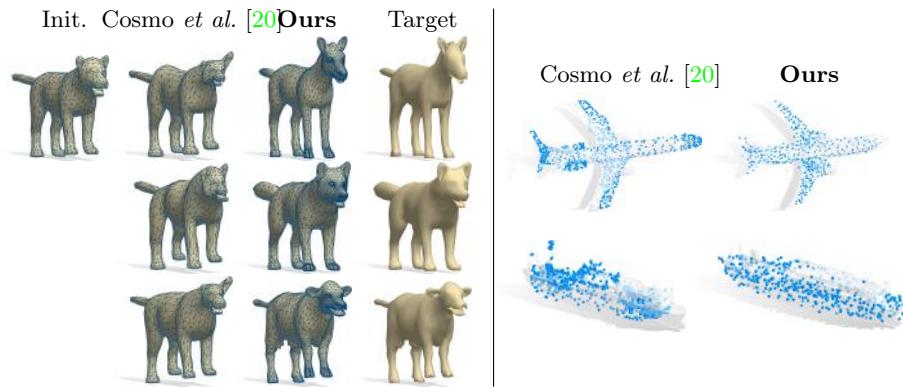
One of the advantages of our pipeline is that we can use the *cubic* FEM (*e.g.*[43, Sec. 4.1]) to discretize the Laplace-Beltrami operator, virtually without any additional cost if compared with the linear finite elements. The cubic FEM yields a more accurate discretization and improves the quality of the results produced by our pipeline. In Fig. 2 we collect additional comparisons between the two different discretizations.



**Fig. 2.** Additional comparisons (one per row) between the use of eigenvalues of the Laplacian discretized with linear FEM or with cubic FEM in our shape-from-spectrum pipeline. The heatmap encodes point-wise reconstruction error, growing from white to dark red. “Unknown target” in the left-most column refers to the source shape from which the eigenvalues are computed.

## 4 Shape from spectrum

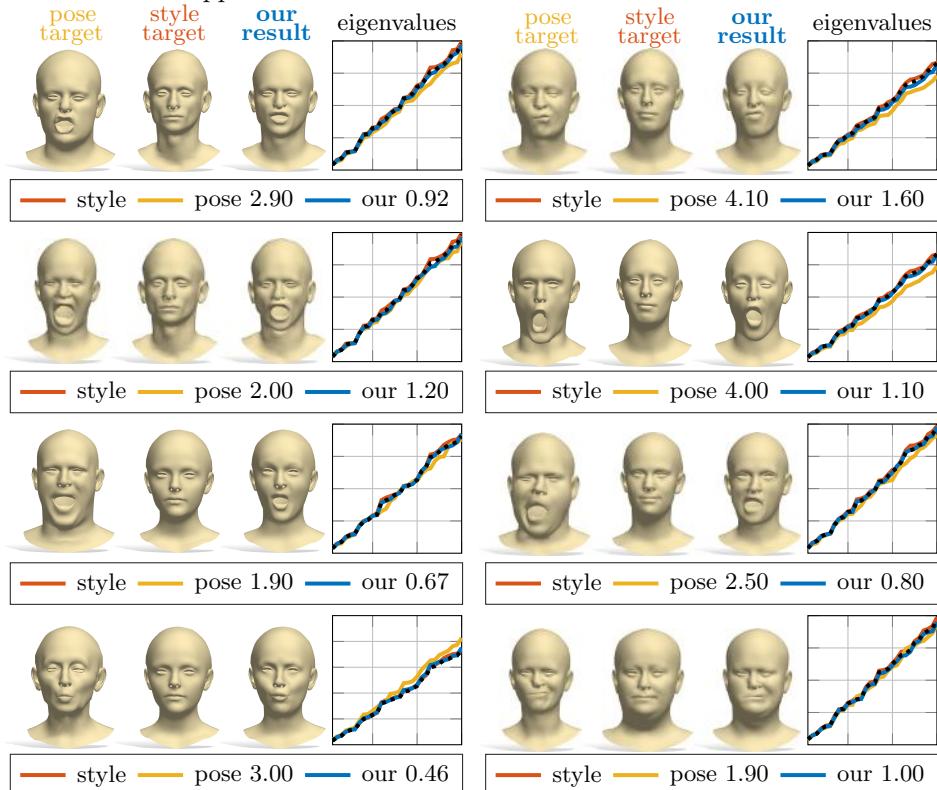
In the task of recovering shapes from Laplacian spectra we compared with the *isospectralization* approach introduced in [20]. In Fig. 2, we report some qualitative examples from SMAL dataset [54] and ShapeNet dataset [24]. For SMAL, both the unknown shape to be recovered and our reconstruction are meshes (left panel of Fig. 2); for objects from ShapeNet, our network has been trained on unorganized sparse point clouds, so the reconstruction in this case is a point cloud as well (right panel of Fig. 2). Our method recovers shape from spectrum in a *single forward pass*, while isospectralization *deforms* an initial shape such that its spectrum align the one of a target shape using geometric regularizers. As initialization for isospectralization we used the template of SMAL [54] for animals, and the closest (in terms of distance between the eigenvalues) shape from our training set for objects of ShapeNet. Even so, isospectralization produces unrealistic instances or fails to capture the geometry of the target compared to our method.



**Fig. 3.** Comparison in shape from spectrum estimation between isospectralization [20] and our method in the case of meshes (left) and point clouds (right). The *Init.* shape on the left is the neutral shape of the SMAL[54] generative model used as initialization of isospectralization algorithm for animals. The target shapes in the point cloud case are shown under the point clouds.

## 5 Style transfer

As described in the main manuscript, we apply our trained network for the style transfer task. In Fig. 4 we show some additional examples. We emphasize here that our method is correspondence-free, since the input eigenvalues completely encode the target style. Our method also does not rely on the presence of the undeformed source shape, used in previous analogies applications, e.g. [9] that phrased the problem of finding  $X$  such that  $(A \rightarrow B, C \rightarrow X)$  for some known  $A, B, C$ . These examples confirm the robustness and accuracy of the proposed method in this application.



**Fig. 4.** Examples of style transfer organized in two columns. In each column, the target style (middle) is applied to the target pose (left). The result shown on the right is obtained by decoding the resulting latent vector. For each example we also report the corresponding eigenvalue alignment (rightmost plots). The black dotted line is the image of  $\rho$ . The numbers in the legend denote the distance from the target “style” spectrum to the source pose and to our generated shape; a small number suggests that the generated shape is a near-isometry of the style target.

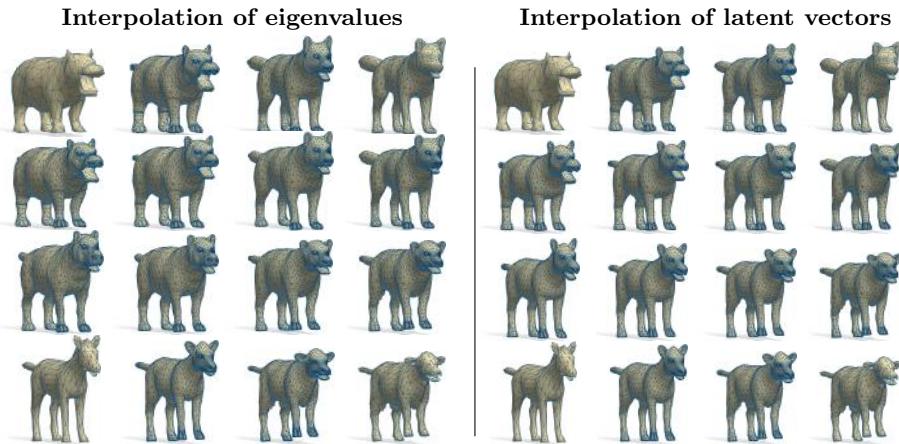
## 6 Spectral shape exploration

In the main manuscript, we show two different sub-tasks of shape exploration: shape interpolation and interactive spectrum-driven exploration.

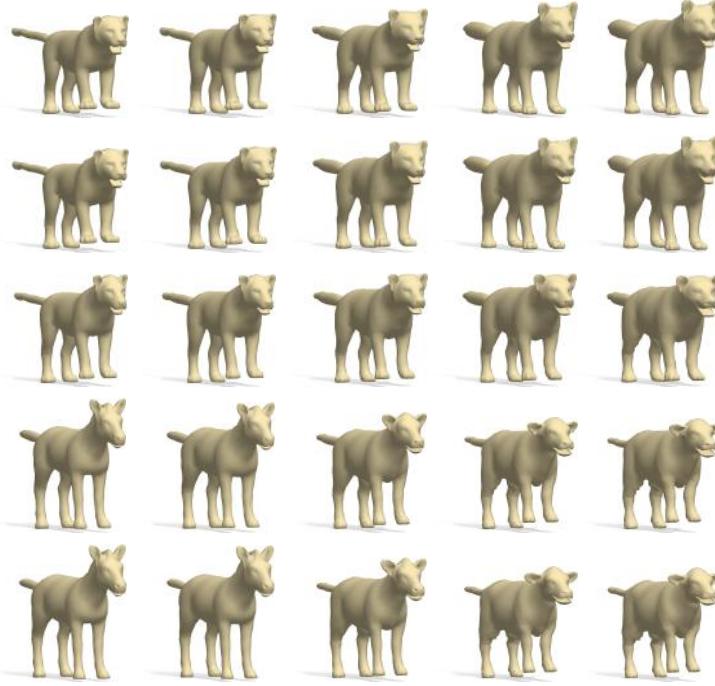
As we claim in the main manuscript, an alternative for shape interpolation is to regard the eigenvalues themselves as a space that we can explore, as is typically done with latent spaces. In Fig. 5, on the same four low-resolution shapes, we compare interpolation in the latent space (on the right) with interpolation in the space of the eigenvalues (on the left). The former evaluation is also included in the main manuscript. For the latter, given the four shapes, we first compute their spectra, directly perform a bilinear interpolation among these spectra, and finally reconstruct the corresponding shapes. Also in this case, the entire procedure can be performed for shapes with different connectivity. The linear interpolation among spectra gives rise to meaningful results. However, observe that the space of spectra (where each point is a different sequence of eigenvalues) is not a proper vector space, and taking linear steps is not completely appropriate in this domain.

We support this with two additional figures. In Fig. 6, for four shapes in the training set, we provide a denser linear interpolation in the space of the eigenvalues. In Fig. 7, we show the same linear interpolation between the spectra of a pair of shapes that come from two different datasets, and which were never seen in the training phase. Both cases confirm that interpolation can indeed be performed in the space of eigenvalues.

Finally, in a video attached to these supplementary materials, we show a demo of interactive spectrum-driven shape exploration. In the video, we modify the eigenvalues of a given shape and we show how these modifications allow us to interactively navigate the space of shapes.



**Fig. 5.** Comparison between interpolation in the space of eigenvalues (left) and the interpolation in the latent space (right). The input shapes are four low-resolution versions of as many shapes from the training set.



**Fig. 6.** Interpolation *between spectra* (as opposed to latent vectors) between four shapes in the training set.



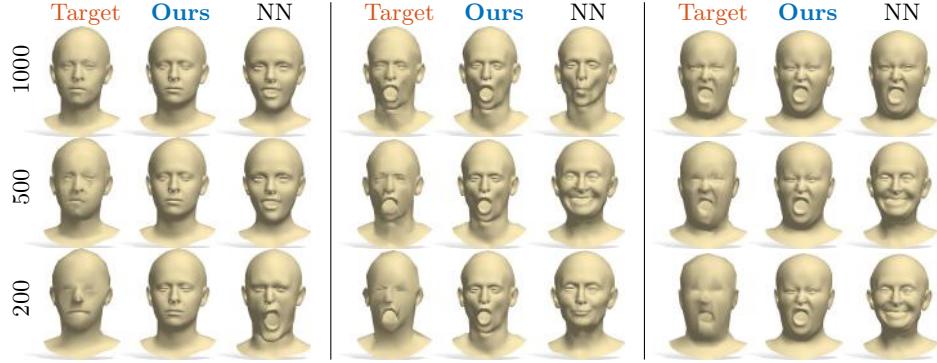
**Fig. 7.** Interpolation between the spectra of two low-poly shapes that belong to two different datasets, never seen at training time.

## 7 Super-resolution

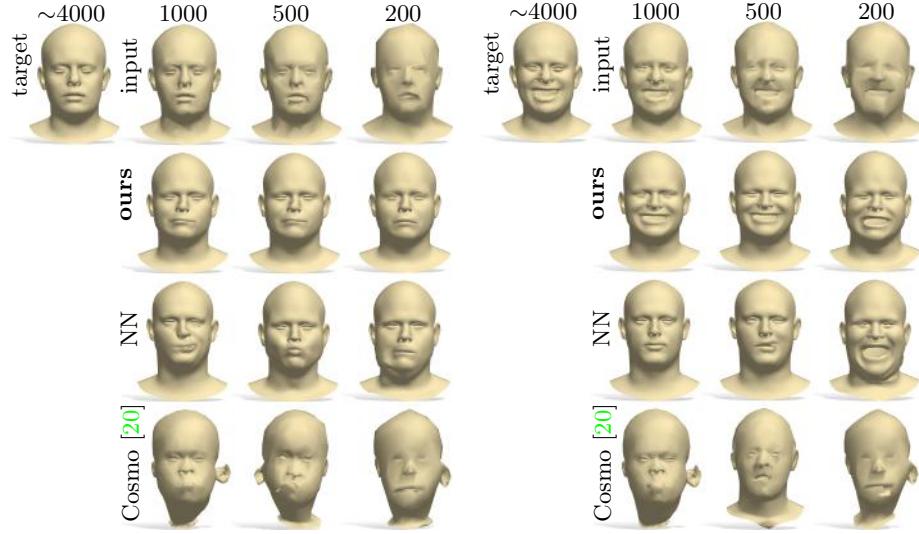
Our trained network model is largely insensitive to mesh resolution and sampling as we show in our experiments in the main manuscript. This property makes our network appropriate for the task of mesh super-resolution. Given a low-resolution mesh as input, we can recover a higher resolution counterpart of the mesh that is also in dense point-to-point correspondence with models from the training set. We can do that in a single shot starting only from the spectrum of the input low-resolution mesh. In Fig. 8, we show a comparison with nearest-neighbors (NN) between eigenvalues (with shapes within the training set), and

the *isospectralization* method of Cosmo *et al.* [20] for two different shapes from a subject that is not involved in the training of our network model.

In Fig. 9, we perform the same experiment on three different shapes never seen during the training, and belonging to a subject whose other poses are in the training set. This comparison highlights that our method generalizes better than NN also for shapes that should be well described by the training set. This allows us to state that navigation driven by the eigenvalues is more accurate than NN navigation of the latent space.



**Fig. 8.** Mesh super-resolution for input shapes at decreasing resolution (top to bottom, target shapes have respectively 1000, 500 and 200 vertices). Our solutions match closely with the original high-resolution version of the input shapes, while the nearest neighbor baseline (NN) predicts the wrong identity or pose.



**Fig. 9.** Two mesh super-resolution examples for input shapes at decreasing resolution (top row of each of the two images, left to right). Our solutions match closely with the original high-resolution version of the input shapes (top left), while other approaches either predict the wrong pose (NN baseline) or generate an unrealistic shape (Cosmo *et al.*).

## 8 Training and test details

Here we summarize the data involved in the trainings and in the experiments. In the code that we will release upon acceptance, we will attach pre-processing scripts and when possible the data used to permit replication.

Overall, we perform five different trainings: for CoMa and SMAL datasets we train the dense and the PointNet version of our Network on the same set of data. For the ShapeNet dataset, only the PointNet version is trained. All the PointNet trainings are performed using 500 points sampled sparsely over the original mesh. The applications use these five models, and we do not do any ad-hoc optimization for different tasks.

Concerning the involved training sets:

**CoMa.** We use 1,853 shapes randomly picked between 11 subjects from the CoMa dataset [42] (168 shapes for each subject), leaving out the 12th subject for testing purpose.

**SMAL.** We use 4,430 animal meshes generated with the SMAL [54] generative model, among all the five possible classes. Thanks to the method explained in the original paper, we can cluster them in the correct races while we explore the inter-class variety.

**ShapeNet.** We would remark that ShapeNetCore dataset [13] is not practical for standard spectral analysis since the models consist of non-manifold disconnected pieces. For this reason, we used a processed watertight version of it [24]. From it, we pruned the meshes that still have artifacts (e.g. disconnected components). Finally, we remeshed the remaining ones to have at maximum 10K vertices. For training, we relies on four different classes: airplanes, boats, screens, and chairs. We pick 2,047 meshes from these classes, among the ones with a more reliable spectrum (i.e. removing outliers case). This last property was not considered for the test set, to challenge ourselves also against non-standard models (as you can see in Figure 11 of the main manuscript, where the target airplane has a tail's part missing).

Concerning the quantitative experiments:

**Shape from spectrum.** This experiment was performed on a total of 400 shapes; 100 shapes of the unseen subject from the CoMa dataset at four different resolutions.

**Estimating point cloud spectra.** Here we consider a total of 1,186 models for the test: 393 shapes of different subjects from FLAME [32] dataset at two different samplings, and 400 shapes from ShapeNet from the four classes.

**Shape matching.** Finally, for the shape matching, we consider 100 couples of random animals of different species. These animals are generated with the same method of the above training set, but with higher variance in the generative space of SMAL. In this way, the testset presents more extreme characteristics of the ones seen at training time. For matching, we mainly consider extreme cases where high non-isometric deformation occurs (e.g. as shown in Figure 11 between hippos and horses).

As a final comment, we would emphasize that our results against Nearest-Neighbor suggest our generalization capability and the real challenge of our testsets.