

FRIDU: Functional Map Refinement with Guided Image Diffusion

Avigail Cohen Rimon¹, Mirela Ben-Chen¹ and Or Litany¹

Technion - Israel Institute of Technology

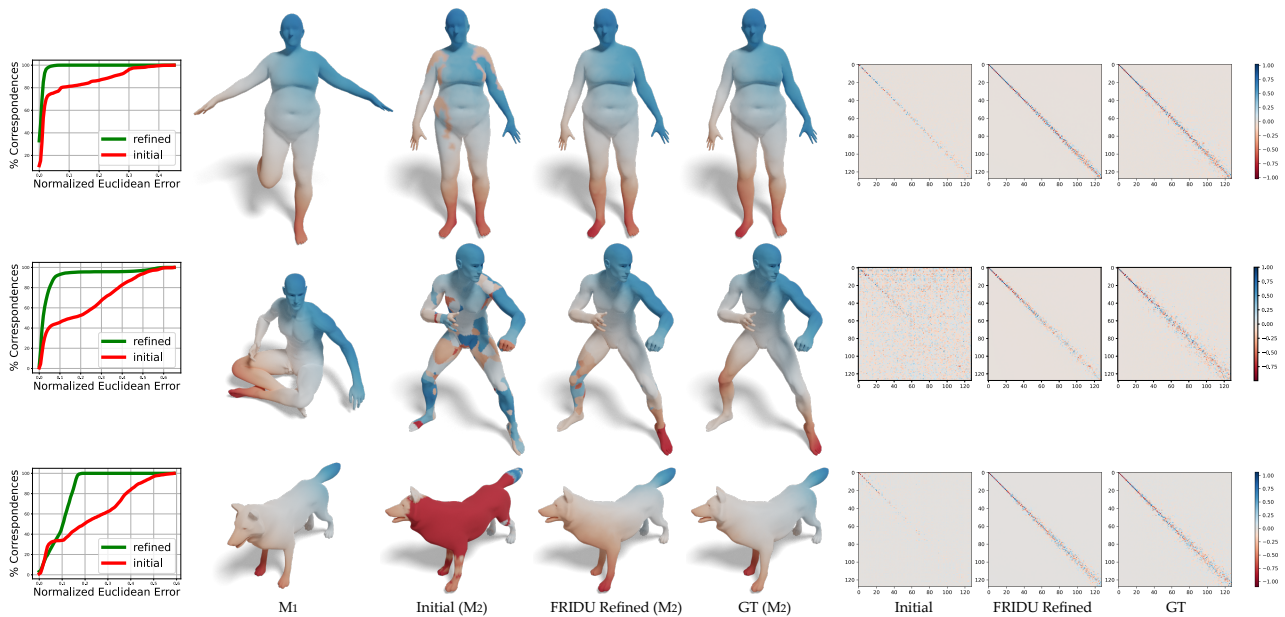


Figure 1: FRIDU refines computed functional maps (Initial) via a guided image diffusion process. We show refinement results across different datasets and descriptor types: (top) FAUST with WKS, (middle) human TACO with SHOT, and (bottom) non-human TACO with WKS. Left: Error curves demonstrating consistent and significant improvement in correspondence quality. Middle: representative mappings from source to target shapes using the initial, FRIDU refined, and ground truth maps. Right: the corresponding functional map matrices.

Abstract

We propose a novel approach for refining a given correspondence map between two shapes. A correspondence map represented as a functional map, namely a change of basis matrix, can be additionally treated as a 2D image. With this perspective, we train an image diffusion model directly in the space of functional maps, enabling it to generate accurate maps conditioned on an inaccurate initial map. The training is done purely in the functional space, and thus is highly efficient. At inference time, we use the pointwise map corresponding to the current functional map as guidance during the diffusion process. The guidance can additionally encourage different functional map objectives, such as orthogonality and commutativity with the Laplace-Beltrami operator. We show that our approach is competitive with state-of-the-art methods of map refinement and that guided diffusion models provide a promising pathway to functional map processing.

1. Introduction

Functional maps represent correspondences between shapes as a change of basis matrix [OBCS*12, OCB*16]. This point of view has a myriad of benefits, such as efficiency and generality. Since

their inception, multiple aspects of the functional maps computational pipeline have been extended and improved, for example by using different functional bases [WBCPS18, HSA*23, BXNL24], by incorporating them in a deep learning framework [LRR*17, HLR*19a, SMJ*23a], and by addressing issues such as partial-

ity [RCB*17, LRB*16, LRBB17, APO21, BDK24] and smoothness [RPWO18, MRSO22], to mention just a few.

However, not all functional maps represent a valid point to point (P2P) map. Hence, a major challenge is converting the functional map to a high quality P2P map for downstream applications. This task is known as functional map *refinement*, and has been widely addressed in the literature, using both classical [EBC17, NMR*18, ESBC19, MRR*19] and data-driven methods [MO24].

Reconstructing a P2P map while refining the functional map is a strong regularizer which has been used in previous works, e.g. ZoomOut [MRR*19]. However, incorporating this step within a deep learning network is difficult, due to the large dimensions of the data involved, which depends on the dimensions of the input meshes, and the difficult non-linear constraints, essentially seeking a permutation.

Image diffusion models [HJA20, SDWGM15, SE19, RBL*22] have been hugely popular for image generation due to their high fidelity and flexibility. Initially primarily conditioned on text, these models have been adapted to support image-conditioned generation, allowing control via visual inputs for tasks such as style transfer, image-to-image translation, or super-resolution. In addition, guidance techniques [CKM*22, SZY*23, BCS*23, YWZ*23, HML*23] have been developed to steer the generative process at inference time using additional objectives – enabling control over semantics, geometry, or alignment with external signals.

We propose that a *conditioned image diffusion model with guidance* is a tool which is highly beneficial for functional map refinement. First, functional map matrices are treated as images, mapping the values to pixel values. Second, in the learning phase we work solely on the functional maps, learning to generate an "image" which represents a refined functional map conditioned on an "image" which represents a noisy initial functional map. The initial maps can be computed using different means, e.g., using descriptor correspondence. Finally, in the inference step, we use the input initial functional map as a condition and generate using the learned model a refined functional map.

This is highly efficient, as no P2P maps are required in the learning stage, since the ground truth functional maps are computed from the ground truth P2P maps in a preprocessing step. However, as is widely known [MRR*19], restricting the output functional map to correspond to a P2P map is highly beneficial for the output map quality. Thus, in the inference step, we use a *guidance objective* inspired by P2P reconstruction techniques to guide our output map, leading to refined functional maps that correspond to high quality P2P maps.

We demonstrate that our approach is flexible, accepting as input functional maps computed through various means (e.g., different descriptors, deep feature extractors). Furthermore, we show that additional guidance objectives (e.g., orthogonality, or other functional map regularizations) can effectively improve the output map. Notably, even when trained on a very small dataset our method successfully improves maps for shapes beyond the training distribution, as illustrated in Figure 1 by an example involving a dataset distinctly different from the training set. Finally, we show that our

method compares favorably to the state of the art of functional map refinement approaches, both classical and data-driven.

1.1. Related Work

Functional Map Refinement. Functional maps have been introduced by Ovsjanikov et al. [OBCS*12], and since then have been generalized in many ways [OCB*16, DYDZ22]. Here, we focus on map *refinement*, namely improving an initial functional map, computed by some non-accurate means, e.g. from shape descriptors, such that an accurate point-to-point (P2P) map can be extracted from it.

Already when introduced [OBCS*12], an iterative ICP algorithm in the spectral embedding space was proposed for extracting a P2P map. Later improvements included considering a smoothness assumption [EBC17], namely that the pulled back Laplace-Beltrami eigenfunctions of one shape are *in the span* of the Laplace-Beltrami eigenfunctions of the second shape. We also use this prior, though we leverage it as a *guidance* term during the inference process. Thus we guide the functional map improvement using the P2P alignment prior. Among the many later refinement schemes, ZoomOut [MRR*19] and IMA [PRM*21] stand out. ZoomOut uses the insight that alternatively upsampling the spectral dimension and projecting on the space of pointwise maps leads to better map recovery. IMA uses a connection to optimal transport to directly improve the functional map matrix. Recently, a differential version of ZoomOut was proposed [MO24], which is incorporated within a network for learning shape correspondences. The refinement component there, however, has no learnable parameters.

As opposed to other map refinement methods, our approach has a few unique properties. First, to the best of our knowledge, it is the first method that *learns* to refine functional maps from initial and ground truth maps, as opposed to computing refined maps from learned features. Second, the training phase is done fully on the functional map matrix (after a pre-computation of the initial and ground truth maps). Finally, we treat the functional maps as *images* and leverage the powerful image diffusion models.

Diffusion Models for Geometric Data. Diffusion models were originally introduced for high-fidelity 2D image generation [HJA20, SE19, DN21, RBL*22], and have since been extended to a wide range of modalities. Their flexibility, stability, and controllable generation capabilities have recently made them attractive for 3D geometric data. This includes applications in 3D shape and scene synthesis, novel view synthesis, avatar modeling, and structure-aware feature extraction.

For geometric generation, diffusion models have been applied across a broad spectrum of representations. Early works explored point-based generation [LH21, NJD*22, MKRV23, WWF*23, ZHM*24], while voxel and volumetric formulations enabled more structured geometry synthesis [HLHF22, MSP*23, TGW*23]. Triplane-based models [CGC*23, SCP*23, WZZ*23, ZCW*24, GXN*23, LGL*24] have proven effective for neural field generation and textured mesh synthesis. Direct mesh-based diffusion has also been explored [LFB*23]. More recently, 3D Gaussian Splatting (3DGS) has become a dominant representation. Several methods are trained directly on explicit 3D splat

data [HCP*24, ZCY*24, LTQ*23, LWT23], while a number of works have demonstrated that high-quality 3DGS can also be generated using only 2D supervision by leveraging differentiable rasterization and novel view consistency [PST*24, SRV23, LXC*24, CZX*24]. In parallel, 2D diffusion models have also been leveraged to create 3D assets via optimization-based lifting [PBJM22, TRZ*23, LSdOBG*24]. Other approaches target novel view synthesis [WCMB*22, LWVH*23, SXL24] and multimodal 3D outputs [XLX*24, WZG*25], expanding the generative space to include conditional, occluded, or multi-view scenarios.

While diffusion models have been widely applied to 3D generation, their use for geometric understanding and shape correspondence remains limited. Some recent works have focused on extracting features from 2D renderings of meshes using diffusion backbones [DMM24], to decorate surfaces with semantic features. Others have repurposed 2D diffusion models fine-tuned for view synthesis to extract geometry-aware features [XLFL24, TJW*23], which show strong view correspondence but are not used for shape matching or refinement. Je et al. [JLY*24] apply generative sampling via Riemannian Langevin Dynamics for robust symmetry detection, illustrating the broader use of probabilistic methods in geometric reasoning.

The most closely related and concurrent work is by Zhuravlev et al. [ZLG25], who also use diffusion models to learn functional maps. Their approach predicts correspondences to a fixed template using a denoising pipeline trained on synthetic human shapes. In contrast, our method assumes a given (possibly noisy) functional map as input and *refines* it between arbitrary shape pairs, with flexible plug-and-play guidance during inference.

Guidance in Diffusion Models Recent work has shown that diffusion models can be steered at inference time using guidance mechanisms that incorporate semantic or structural objectives. Classifier-based [DN21], classifier-free [HS22], and gradient-based plug-and-play methods [BCS*23, KEES22, WYZ22, CKM*22, LDR*22, CSRY22, GMJS22, HML*23, CLY23] allow external objectives to influence the generation process without retraining. These approaches have been used for semantic control, view consistency, and geometric constraints [PST*24, SRV23, RLP*23].

Our method follows this paradigm but applies guidance to functional map refinement, directing the diffusion process with geometric losses such as pointwise consistency, orthogonality, and Laplacian commutativity—objectives specific to spectral correspondence problems.

1.2. Contributions

Our main contribution is a reformulation of the functional map refinement problem as a conditional image generation problem. Specifically, we:

- Show that conditional image diffusion with guidance can refine functional maps generated by various sources.
- Provide a P2P guidance that efficiently improves the map at test time, as well as a variety of guidance mechanisms appropriate for different shape classes.

- Demonstrate that our approach compares favorably to state of the art functional map refinement algorithms, both classical and deep.

2. FRIDU

2.1. Notation

We denote a pair of meshes by $\mathcal{M}_j = (\mathcal{V}_j, \mathcal{F}_j)$, where $j = 1..2$, and $\mathcal{V}_j, \mathcal{F}_j$ are the vertices and faces respectively, with $n_j = |\mathcal{V}_j|, m_j = |\mathcal{F}_j|$. The first k_j eigenfunctions of the Laplace-Beltrami operator of the mesh \mathcal{M}_j are denoted by a matrix $\Phi_j \in \mathbb{R}^{n_j \times k_j}$, and the corresponding diagonal matrix of k_j eigenvalues is denoted by Λ_j . A point to point (P2P) map between the meshes is denoted by $T_{21} : \mathcal{M}_2 \rightarrow \mathcal{M}_1$, and maps vertices on \mathcal{M}_2 to vertices on \mathcal{M}_1 .

The P2P map can also be represented as a binary row stochastic matrix $\Pi_{21} \in \{0, 1\}^{n_2 \times n_1}$, where $\Pi_{21}(i, j) = 1$ if and only if $v_i \in \mathcal{M}_2$ is mapped to $v_j \in \mathcal{M}_1$. Π_{21} maps *indicator* functions defined on \mathcal{V}_1 to *indicator* (or zero) functions defined on \mathcal{V}_2 . A *soft* P2P map $P_{21} \in \mathbb{R}^{n_2 \times n_1}$ maps *real* functions defined on \mathcal{V}_1 to *real* functions on \mathcal{V}_2 . A functional map $C_{21} \in \mathbb{R}^{k_2 \times k_1}$ maps functions given in the spectral basis Φ_1 to functions defined in the spectral basis Φ_2 .

2.2. Overview

At inference time we are given an initial functional map \tilde{C}_{21} between two meshes $\mathcal{M}_1, \mathcal{M}_2$, as well as the corresponding functional bases Φ_1, Φ_2 . We need to provide as output a refined functional map C_{21} , and its corresponding refined P2P map Π_{21} .

Towards that end we train a conditioned image diffusion model d_θ which generates our refined map C_{21} , conditioned on the initial map \tilde{C}_{21} . We train on maps \tilde{C}_{ij} between models $\mathcal{M}_i, \mathcal{M}_j$, from a dataset of models $\mathcal{D}_{\text{shapes}}$. These initial maps are given together with the ground truth P2P maps Π_{ij}^* , from which we generate during pre-processing the corresponding ground truth functional maps $C_{ij}^* = \Phi_i^\dagger \Pi_{ij}^* \Phi_j$. The training is done solely in the functional space, namely only C_{ij}^* and \tilde{C}_{ij} are used during training.

The training and inference procedures are illustrated in Figure 2.

2.3. Training

To simulate real-world scenarios where only noisy initial maps are available, we compute initial functional maps \tilde{C}_{ij} using descriptor-based techniques (e.g., WKS [ASC11] or SHOT [TSDS10] descriptors). Thus, each training example consists of a pair $(\tilde{C}_{ij}, C_{ij}^*)$, where \tilde{C}_{ij} serves as the conditioning input and C_{ij}^* is the supervision target. We follow the denoising diffusion framework [HJA20], and adopt the EDM-DDPM++ formulation proposed by Karras et al. [KAAL22]. In this formulation, the ground-truth functional map C_{ij}^* is corrupted using a diffusion process by adding Gaussian noise with a *continuous* noise level σ :

$$C_{ij}^N = C_{ij}^* + \sigma \cdot \epsilon, \quad \log \sigma \sim \mathcal{N}(\mu_p, \sigma_p^2) \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad (1)$$

where μ_p and σ_p are scalar parameters controlling the center and spread of the noise levels, and $\epsilon \in \mathbb{R}^{k_2 \times k_1}$ is a matrix of i.i.d. Gaussian noise with entries $\epsilon_{a,b} \sim \mathcal{N}(0, 1)$, matching the dimensions of the functional map C_{ij}^* .

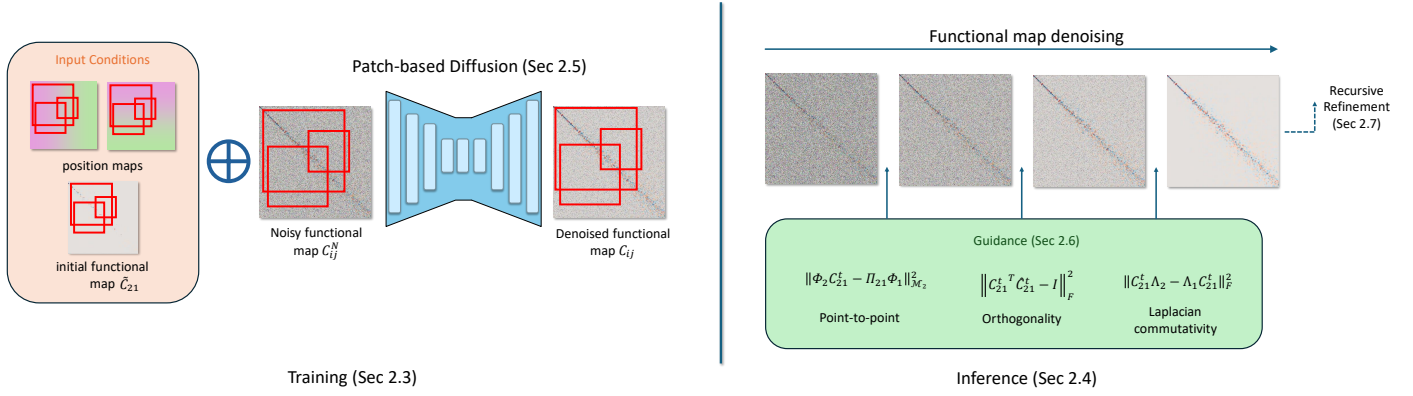


Figure 2: An illustration of our (left) training and (right) inference procedures. During training, our pipeline takes as input a random patch of a noisy functional map C_{ij}^N , conditioned on a corresponding patch of an initial functional map C_{ij} and position maps, and outputs the denoised patch of the functional map C_{ij} . At inference time, the patch covers the full-sized image, and we incorporate guidance at each denoising step, including point-to-point guidance and potentially additional regularizers, such as orthogonality and Laplacian commutativity.

Our model d_θ is trained to directly recover the clean map from the noisy sample C_{ij}^N , conditioned on the initial map and noise scale σ . We minimize the denoising objective:

$$\mathbb{E}_{(\mathcal{M}_j, \mathcal{M}_i) \sim \mathcal{D}_{\text{shapes}}, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \log \sigma \sim \mathcal{N}(\mu_p, \sigma_p^2)} \left[\lambda(\sigma) \left\| C_{ij}^* - d_\theta(C_{ij}^N | \tilde{C}_{ij}, \sigma) \right\|_2^2 \right] \quad (2)$$

where $\lambda(\sigma)$ is a noise-dependent scalar weighting function. See [KAAL22] for further details.

Training is performed entirely in the spectral (functional) domain, without requiring access to explicit P2P supervision. Once trained, the model can generate refined functional maps C_{ij} from noise, guided by an initial estimate \tilde{C}_{ij} , which can then be decoded into dense correspondences using existing recovery techniques.

2.4. Inference

At inference time, we are given a new pair of shapes, their corresponding spectral bases Φ_1, Φ_2 , and an initial functional map \tilde{C}_{21} . Our goal is to generate a refined functional map C_{21} , and recover accurate point-to-point correspondences Π_{21} .

We initialize a noisy sample $C_{21}^{t=T} \sim \mathcal{N}(0, \sigma_T^2 \mathbf{I})$, and iteratively denoise it using our EDM-based model d_θ . At each step $t = T, \dots, 1$, the model predicts a cleaner version of the functional map corresponding to the current noise level σ_t , and the sample is updated accordingly:

$$C_{21}^{t-1} = C_{21}^t + (\sigma_{t-1}^2 - \sigma_t^2) \frac{d_\theta(C_{21}^t | \tilde{C}_{21}, \sigma_t) - C_{21}^t}{\sigma_t^2} \quad (3)$$

This update rule follows the deterministic sampling procedure proposed in [KAAL22]. In practice, we use the full EDM sampler with noise perturbation and second-order correction for improved stability. The final output C_{21}^0 is the refined functional map C_{21} .

2.5. Efficient Training via Patch-based Diffusion

Diffusion models typically require large amounts of training data, but in the case of functional maps, available datasets are relatively small. To improve training efficiency and data effectiveness, we adopt a *patch-wise training strategy* [WJZ*23].

Rather than denoising full functional maps, we train on smaller patches $C_{ij}^{(p)} \in \mathbb{R}^{s \times s}$ randomly cropped from the ground-truth functional map C_{ij}^* , with patch size s sampled stochastically or progres-

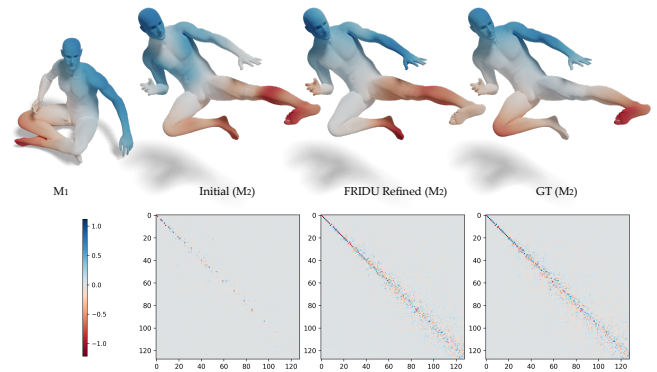


Figure 3: Refinement without Guidance (WKS). We map a function $f_1 \in \mathbb{R}^{n_1}$ defined on \mathcal{M}_1 to \mathcal{M}_2 , using the initial, FRIDU refined, and ground-truth functional maps. Here, we map the function $\Phi_1^\dagger f_1$ using the functional map matrices to get \tilde{f}_2 , and show $\Phi_2 \tilde{f}_2$ on \mathcal{M}_2 . Note that in this figure only, our refined map does not include guidance in inference, in order to isolate and illustrate the refinement ability of the base model. The top row visualizes the source and mapped functions, and the bottom row shows the corresponding functional map matrices. We observe improvement in our refined map in both the mapped function and the functional map matrix.

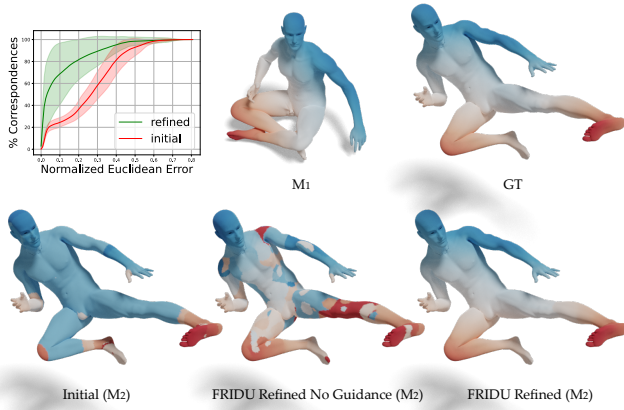


Figure 4: Refinement with P2P Guidance (WKS). We show the *pointwise* mapping Π_{21} extracted from the functional maps in Figure 3, mapping a function $f_1 \in \mathbb{R}^{n_1}$ from \mathcal{M}_1 to \mathcal{M}_2 using $\Pi_{21}f_1$. We show the pointwise maps obtained without guidance (center) and with P2P guidance (right). Note the improvement in the pointwise map extracted from the FRIDU refined functional map compared to the initial map, and the significant improvement when adding guidance. The plot shows the average normalized Euclidean error over the Michael dataset, for the initial and refined maps. The shaded region shows the standard deviation around the average.

sively. Each patch is denoised independently, conditioned on the corresponding region from the initial map \tilde{C}_{ij} , its spatial location (a, b) , and the noise level. We concatenate two additional channels encoding the normalized position of each pixel within the full map to the patch input.

Patch locations (a, b) are sampled uniformly from the valid crop positions within the full functional map domain, denoted $(a, b) \sim \mathcal{U}(\mathcal{R})$. Patch sizes s are drawn from a predefined schedule $p(s)$.

The model is trained to minimize the patch-level denoising loss:

$$\mathbb{E}_{\substack{(a,b) \sim \mathcal{U}(\mathcal{R}) \\ s \sim p(s) \\ \log \sigma \sim \mathcal{N}(\mu_p, \sigma_p^2) \\ \epsilon \sim \mathcal{N}(0,1)^{s \times s}}} \left\| d_{\theta} \left(C_{ij}^{(p)} + \sigma \cdot \epsilon \mid \tilde{C}_{ij}^{(p)}, a, b, s, \sigma \right) - C_{ij}^{(p)} \right\|_2^2 \quad (4)$$

To encourage the model to learn both local details and global structure, we vary the patch size during training and occasionally include full functional maps. Inference is performed on full maps without modification to the model architecture.

2.6. Guidance

Point to Point Map Extraction. The conditional diffusion model produces plausible functional maps C_{21} that resemble the training distribution (see Figure 3). However, we also need to compute the P2P map Π_{21} . In general, since the functional maps are given in a reduced spectral basis (and thus are *smoothed* versions of the P2P maps), there are many possible corresponding P2P maps (see e.g. the discussion in Ezuz et al. [EBC17]). Hence, some regularization

is required to solve this ambiguity. A minimal additional regularizing requirement, is that smooth functions on \mathcal{M}_1 are smooth on \mathcal{M}_2 after the map, which is formalized using $\Pi_{21}\Phi_1 \in \text{span}(\Phi_2)$. This combined with the requirement that $\|C_{21} - \Phi_2^\dagger \Pi_{21}\Phi_1\|$, leads to the optimization problem [EBC17]:

$$\Pi_{21}(C_{21}) = \arg \min_{\Pi_{21} \in \mathcal{P}_{21}} \|\Phi_2 C_{21} - \Pi_{21}\Phi_1\|_{\mathcal{M}_2}^2, \quad (5)$$

where \mathcal{P}_{21} is the set of valid maps, i.e. binary row stochastic matrices of dimension $n_2 \times n_1$. This is easily optimized using a nearest neighbor search between the rows of $\Phi_2 C_{21}$ and the rows of Φ_1 [EBC17, Sec 4.2].

Point to Point Guidance. Extracting the P2P map that corresponds to a refined functional map does not, however, result in a high quality map (see Figure 4). Thus, at inference, we would like to incorporate a structural constraint that the computed functional map corresponds to a P2P map.

Diffusion models allow such structure to be injected via *guidance* at inference time [CKM*22, SZY*23, BCS*23], without requiring retraining. In our case, we wish to guide the model toward refined maps that produce accurate point-to-point correspondences, hence a natural guidance loss would be to use Eq. (5):

$$\|\Phi_2 C_{21}^t - \Pi_{21}(C_{21}^t)\Phi_1\|_{\mathcal{M}_2}^2. \quad (6)$$

However, directly differentiating through the solution of this optimization is computationally expensive due to the nearest neighbor computation for n_2 points in \mathbb{R}^{k_1} which is required for computing Π_{21} .

Instead, at each step, we solve for Π_{21} but stop gradients through it — treating it as a fixed prediction. Hence, our P2P loss for guidance is:

$$\mathcal{L}_{\text{P2Pg}}(C_{21}^t) = \|\Phi_2 C_{21}^t - \Pi_{21}\Phi_1\|_{\mathcal{M}_2}^2. \quad (7)$$

Although Π_{21} is treated as constant during backpropagation, the iterative nature of the denoising process ensures that updates to C_{21} affect future Π_{21} values. This indirect influence allows the guidance signal to propagate across timesteps, effectively steering the generation toward spectrally and geometrically consistent solutions. Our final P2P map is $\Pi_{21}(C_{21}^0)$. Figure 4 shows the resulting P2P map when using this guidance loss. Note the considerable improvement compared to inference without P2P guidance.

Task-dependent Guidance. This approach to guidance is general and can accommodate a wide variety of loss functions, including enabling adaptation to new tasks or assumptions at inference time. Such flexibility is a key advantage of diffusion-based refinement, enabling controlled generation without compromising generality. In the experimental section (Table 1), we report results where two classical functional map regularizers are added to the guidance: the orthogonality constraint $L_{\text{orth}}(C_{21}) = \|C_{21}^T C_{21} - I\|_F^2$ (promoting area preserving maps) and the Laplacian commutativity term $L_{\Delta}(C_{21}) = \|C_{21}\Lambda_1 - \Lambda_2 C_{21}\|_F^2$ (promoting isometric maps), see e.g. [CB22, RPWO19].

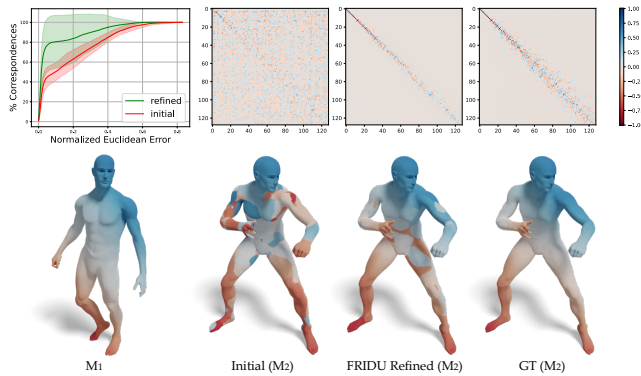


Figure 5: Functional Map Refinement (SHOT). We show the performance of the pointwise mapping extracted from our refined map alongside the initial and ground-truth mappings. We additionally show the corresponding functional map matrices. Note the noisy appearance of the SHOT-based initial map. The plot shows the normalized Euclidean error over the Michael dataset, where our refined maps consistently outperform the initial maps.

2.7. Recursive Refinement

Following the logic of refining a given noisy functional map, we conduct an experiment in which the model is recursively fed its own denoised output during *inference* to assess whether performance continues to improve. In our experiments, we observe that a single recursive iteration typically enhances accuracy, but applying more than one iteration leads to degradation. We report results for adding one recursive iteration during inference in Table 1. Note that this additional refinement entails a computational trade-off.

3. Experimental Results

The following section is organized as follows: In Section 3.1, we evaluate our model’s performance when conditioned on initial functional maps computed different descriptors, and demonstrate its dataset generalization capabilities. In Section 3.2, we compare our method to existing shape matching approaches, and in Section 3.3, we present an ablation study exploring zero-shot condition generalization and guidance parameters.

3.1. Functional Map Refinement

To demonstrate the effectiveness of our refinement model, we apply it to initial functional maps computed from both WKS [ASC11] and SHOT [STDS14] descriptors in a classical way; see details in Appendix B.

Data. We use the *Michael* meshes from the TACO [PMM24] dataset, which provides remeshed versions of different figures in various poses, along with ground-truth correspondences within each figure. Each mesh contains approximately 50K vertices. The dataset includes 190 ground-truth correspondences between the 20 *Michael* shapes. We split the dataset into training and testing sets using a 90:10 ratio.

Initial map from WKS. Figure 3 shows an example of mapping a function $f_1 \in \mathbb{R}^{n_1}$ defined on \mathcal{M}_1 to function on \mathcal{M}_2 . The upper row shows the function on \mathcal{M}_1 and its mapping to \mathcal{M}_2 using (i) the initial map \tilde{C}_{21} , (ii) our refined map C_{21} , and (iii) the ground-truth map C_{21}^* . Here we map $\Phi_1^\dagger f_1$ using the functional map matrix to get \tilde{f}_2 , and show $\Phi_2 \tilde{f}_2$ on \mathcal{M}_2 . Note that only in this figure our refined version doesn’t include guidance during inference, to demonstrate the ability of the base-model refinement abilities beside guidance. The bottom row presents the corresponding functional map matrices. We observe that our refined map is closer to the ground truth than the initial one, successfully resolving artifacts. Some symmetry flips remain, which we discuss later in this subsection.

Figure 4 illustrates the pointwise map extracted from the corresponding functional maps, comparing our refined mapping obtained with and without incorporating P2P guidance during inference. The bottom row shows the pointwise mapping of f_1 to \mathcal{M}_2 using (i) the initial extracted map $\tilde{\Pi}_{21}$, (ii) our refined extracted map *without* guidance, and (iii) our refined extracted map *with* guidance Π_{21} . The top row shows f_1 on \mathcal{M}_1 and its ground-truth mapping on \mathcal{M}_2 , obtained using Π_{21}^* . Here all the pointwise maps are in $\mathbb{R}^{n_2 \times n_1}$, and thus the mapping is done using, e.g., $\Pi_{21} f_1$. We additionally show the normalized Euclidean error over the test set for the initial and refined (*with* guidance) maps. We observe that our approach improves performance both qualitatively and quantitatively.

From now on, whenever we refer to our refined mapping, we mean the version refined *with P2P guidance* during inference.

Initial Map from SHOT. Figure 5 presents the performance of our model when conditioned on SHOT-based functional maps. The bottom row shows f_1 on \mathcal{M}_1 and the (i) initial, (ii) our refined, and (iii) ground-truth pointwise mappings. The top row shows the corresponding functional map matrices and the normalized Euclidean error over the test dataset for both the initial and our refined maps. Note that both the initial pointwise mapping and functional map appear quite noisy, while our refined version resolves many of the artifacts, despite still having some. We observe that although our refined maps exhibit higher error and variance compared to the WKS-based maps experiment, our refined maps still outperforms the initial ones. See an additional example of SHOT-based map denoising in Figure 1.

Landmark Constraints. If a sparse set of corresponding landmarks is available, it can be easily incorporated in our framework, by adding functional constraints based on these landmarks (e.g. as implemented in PyFM [Mag21]) when computing the functional map. Landmarks constraints are useful for improving the quality of the correspondence, especially in the presence of intrinsic symmetries, which lead to non-unique eigendecomposition of the Laplace-Beltrami operator. When only symmetry-invariant descriptors (such as the WKS) are used, symmetry cannot be disambiguated without an additional step of aligning the LB eigenfunctions (see e.g. [ZLG25]), or adding information that disambiguates the symmetry, such as landmarks.

To demonstrate this, we mark 5 landmarks (head, two hands, and two legs) on both source and target shapes and incorporate

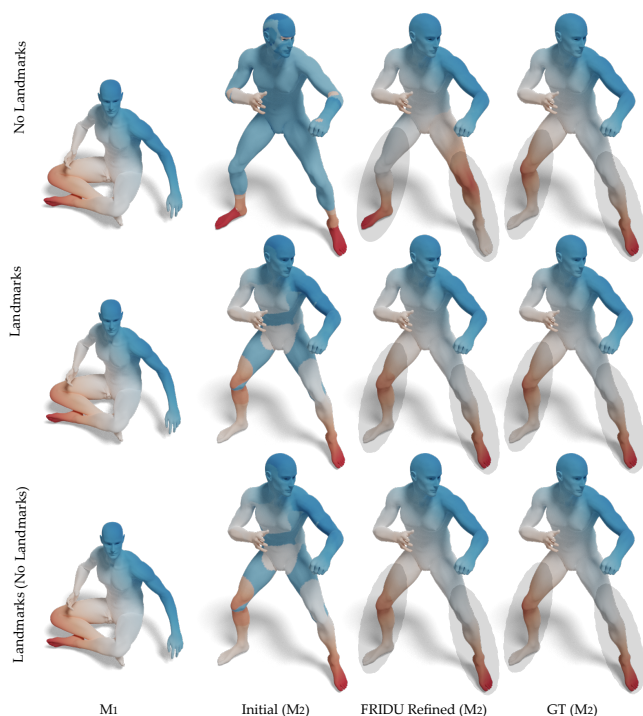


Figure 6: Landmarks. We show results for three settings: (top) training and evaluating on WKS-based initial maps, (center) training and evaluating on WKS-based initial maps with landmarks, and (bottom) training on WKS-based initial maps and evaluating on WKS-based initial maps with landmarks. We note that symmetry flips are resolved in cases where the model is conditioned on an initial map with landmarks, even without training on such maps (see the shaded regions).

these landmark constraints into the initial functional map computation. Figure 6 compares three settings: training and evaluating using the WKS-only initial maps (top row), training and evaluating using initial maps computed with landmarks (middle row), and training using the WKS-only initial maps and evaluating using the initial maps with landmarks (bottom row). From left to right, for each setting we show f_1 on \mathcal{M}_1 , and its pointwise mapping extracted from the (i) initial, (ii) our refined, and (iii) ground-truth mappings. We observe that the symmetric flip of the legs (see the shaded oval regions) present in the top row is resolved whenever conditioning on initial maps with landmarks (middle and bottom rows).

Figure 7 presents the normalized Euclidean error over the test dataset for both the regular and landmark-constrained settings. For each setting, we show the error for the initial and refined extracted pointwise maps. We observe that the landmark-constrained setting improves both the initial and refined map errors compared to the regular setting, with our refined maps outperforming the initial maps in both cases.

Dataset Generalization. To evaluate dataset generalization, we apply the model trained on the *Michael* dataset, conditioned on a WKS-based initial functional map, to shape pairs of human fig-

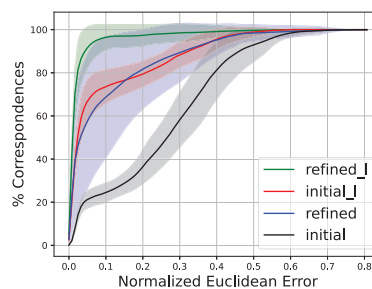


Figure 7: Landmarks. The normalized Euclidean error over the *Michael* dataset for the regular ("refined") and landmarks-constrained ("refined_l") settings. For each setting, we also report the error of the initial maps ("initial" and "initial_l", respectively). Note that incorporating landmark constraints improves the accuracy of both the initial and refined maps, with our refined maps consistently outperforming the initial maps in both settings.

ures outside this dataset, as well as to a non-human figure from the TACO dataset. The top row of Figure 1 shows results for mapping a function between a pair from the FAUST dataset, while the bottom row shows mapping results for a pair of *wolf* shapes from TACO. Note we use the remeshed version of FAUST. From left to right, each row shows (i) the normalized Euclidean error graph of the initial and FRIDU refined maps, (ii) the source and pointwise mapping of f_1 using the initial, FRIDU refined, and ground-truth mappings, and (iii) the corresponding functional map matrices. The average initial and final errors for the top row (human) are 0.19 and 0.05, respectively, and for the third row (*wolf*) are 0.21 and 0.03, respectively. Note that despite being trained on the small *Michael* dataset, our model generalizes well and significantly improves the quality of the initial maps—both quantitatively and qualitatively—on these unseen shape categories.

3.2. Shape Matching Comparisons

Since deep learning-based functional map *refinement* is less established, we position our method within the broader context of deep learning for shape *matching*. Most existing approaches consist of a learned feature extractor followed by functional map computation, typically optimized with geometric regularizers. Our method is compatible with both supervised and unsupervised pipelines, as it operates independently of how the initial functional map is obtained. To isolate the effect of the refinement step, we use as input the functional map produced by the feature extractor of a state-of-the-art method, DiffZO. Notably, DiffZO includes a built-in refinement stage, allowing for a direct comparison between their refinement and ours under the same initial conditions. This setup also demonstrates our model's ability to refine functional maps originating from different sources.

We train two models — one on FAUST [BRLB14] and the other on FAUST + SCAPE [ASK*05] — and evaluate them on FAUST, SCAPE and SHREC'19 [MMR*19] datasets, using the remeshed version [RPWO18] of each dataset. We used the same train/test split as in the DiffZo experiments. Specifically, for the FAUST remeshed dataset we used 80 pairs for training and 20 for testing, and for the SCAPE remeshed dataset we used 51 for training and 20 for testing. The SHREC19 dataset was only used for testing.

Table 1 reports the geodesic error $\times 100$ for axiomatic, supervised, and unsupervised methods. Note that the table presents results for our base

pipeline, as well as for three variants: one using a recursive refinement operator (as described in Sec. 2.7), and another incorporating two common functional map regularizers into the guidance. We also report the error of the initial functional map given as input to our model. Our method consistently improves the initial mapping and outperforms DiffZO in intra-dataset evaluations, while maintaining comparable results in cross-dataset evaluations relative to supervised methods. We emphasize that our primary point of comparison is DiffZo, as our model is conditioned on maps generated using its feature extractor.

Among the axiomatic refinement methods listed, SmoothShells [ELC20] requires access to the full shape geometry during the refinement process — a requirement our method avoids. The DiscreteOp approach [RMWO21] is tailored to minimizing a given functional map energy while keeping the maps *proper*. In contrast, our method does not rely on functional map objectives and except the spectral guidance operates work purely in image space. However, we show that regularization can be incorporated into the guidance when needed, without additional training. The method most closely related to ours is ZoomOut [MRR*19], which has been shown to outperform BCICP [RPWO18] in both accuracy and runtime.

In Figure 8, we compare our method to ZoomOut on the three network models presented in Sect. 3.1 (i.e. initial map computation using WKS, WKS+landmarks and SHOT), trained on the *michael* shapes. Leveraging the flexibility of our guidance framework, we experiment with incorporating spectral upsampling during inference – that is, progressively increasing the dimension of the functional map used to compute the pointwise correspondence throughout the denoising process. With upsampling our method performs comparably to ZoomOut in terms of accuracy. Furthermore, since the *michael* meshes consist of approximately 50K vertices, the computational cost of ZoomOut is quite high (~ 600 seconds per pair), while our inference procedure takes only ~ 60 seconds.

Method	Train: F			Train: F+S		
	F	S	S19	F	S	S19
BCICP [RPWO18]	6.4	-	-	-	-	-
ZoomOut [MRR*19]	6.1	-	-	-	-	-
SmoothShells [ELC20]	2.5	-	-	-	-	-
DiscreteOp [RMWO21]	5.6	-	-	-	-	-
ACSCNN [LLHL20]	2.7	8.4	-	-	-	-
TransMatch [TCM*21]	1.7	30.4	14.5	1.6	11.7	10.9
GeomFmaps [DSO20]	3.5	4.8	8.5	3.5	4.4	7.1
Deep Shells [ETLTC20]	1.7	5.4	27.4	1.6	2.4	21.1
NeuroMorph [ENK*21]	8.5	28.5	26.3	9.1	27.3	25.3
DUO-FMNet [DCO22]	2.5	4.2	6.4	2.5	4.3	6.4
UDMSM [CB22]	1.5	7.3	21.5	1.7	3.2	17.8
ULRSSM [CRB23]	1.6	6.4	14.5	1.5	2.0	7.9
ULRSSM + fine-tune [CRB23]	1.6	2.2	5.7	1.6	2.1	4.6
AttentiveFMaps [LDO22]	1.9	2.6	5.8	1.9	2.3	6.3
ConsistentFMaps [SMJ*23b]	2.3	2.6	3.8	2.2	2.3	4.3
DiffZO [MO24]	1.9	2.9	4.2	1.9	2.3	3.6
Initial	2.5	6.9	12.4	2.1	2.7	8.6
Ours	1.7	4.1	9.4	1.5	2.1	7.1
+ Recursive Refinement (Sec. 2.7)	1.6	4.1	9.3	1.2	1.9	6.6
+ Orthogonality Guidance	1.5	2.7	7.3	1.4	2.0	6.5
+ Laplacian Commutativity Guidance	1.7	4.4	9.8	1.5	2.0	7.3

Table 1: Mean geodesic errors ($\times 100$) when training and testing on the FAUST, SCAPE and SHREC19 datasets. Best result within each method category (axiomatic, supervised, and unsupervised methods) is shown in bold. We consider DiffZO as part of the unsupervised category. The axiomatic and supervised methods are from [SMJ*23b] and the unsupervised methods are from [MO24].

Figure 9 shows a qualitative comparison between our approach and DiffZo when training on Faust and generalizing to SHREC19. We show texture transfer by transferring the texture coordinates generated on M_1 to the other meshes using the corresponding point-to-point map. Since all the maps are vertex to vertex, which leads to highly noisy texture transfer, we first apply a single iteration of RHM [EBC17] to all the maps. We note that the initial map that both methods start from is quite noisy, and our approach (with upsampling) and DiffZo achieve comparable results. It is possible that a better initialization method may lead to improved results.

3.3. Ablation

Zero-Shot Condition Generalization We test our model’s generalization ability to different sources of initial maps by refining WKS-based initial maps using the model trained on SHOT-based mappings, and vice versa. Figure 10 shows two examples of WKS-based maps refined using the SHOT-based trained model (Section 3.1), and two examples of SHOT-based maps refined using the WKS-based trained model (Section 3.1). For each example, we show the pointwise mapping of f_1 and the corresponding functional map matrices. We note that in both cases, our model manages to refine the initial map, but the results of refining SHOT-based maps using the WKS-based model are significantly better.

Guidance Parameters We perform an ablation study on the parameters of the guidance algorithm proposed in [BCS*23]. These are: the number of gradient steps for backward guidance m , the number of recurrent steps k , and the guidance strength parameter s . We refer to the paper for more details. In all our experiments, we set $m = 2$, $k = 5$, and $s = 500$. For cross-dataset evaluations in Section 3.2, we find that increasing s to 2000 improves performance.

Figure 11 shows the performance of the model when inference is performed with different values of these guidance parameters. We use the model from Section 3.1, trained on WKS-based functional maps between *michael* shape pairs, and evaluate it on WKS-based functional map between *horse* shape pairs from TACO. The top row shows f_1 , and the initial and our refined pointwise mapping using the default parameters $m = 2, k = 5, s = 500$. The next three rows show ablations for m, k , and s , each with the other two parameters fixed. Table 2 provides the inference time in seconds and the normalized Euclidean error for each set of parameters. In general, we observe that increasing m and k increases inference

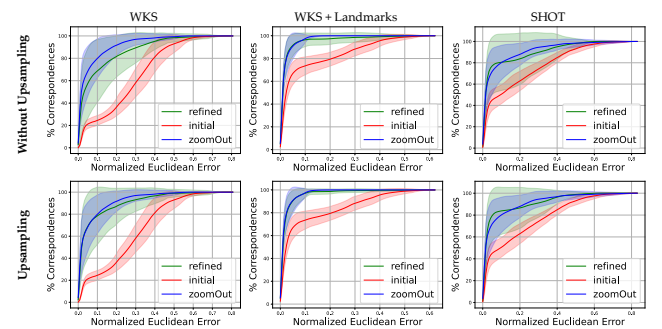


Figure 8: Comparison to ZoomOut. For the three initial map computation methods presented in Section 3.1, we show a comparison to the ZoomOut axiomatic refinement algorithm [MRR*19] on the Michael dataset. We show results with and without incorporating spectral upsampling during inference of our method. We observe that when incorporating upsampling, our method is comparable to ZoomOut.

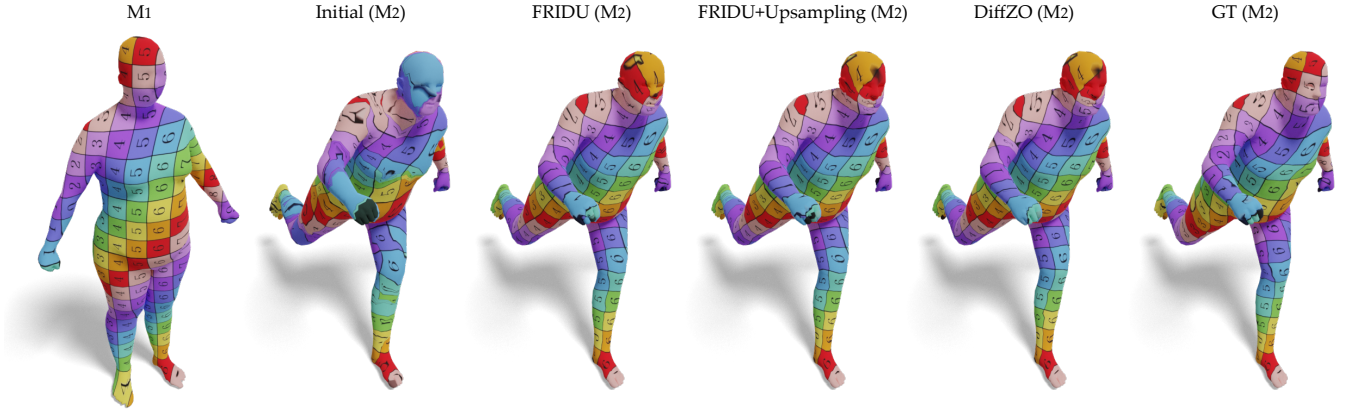


Figure 9: A qualitative example comparing the map improvement obtained by our approach and by DiffZo [MO24]. For a pair of shapes from SHREC19, we show the resulting maps when both methods were trained on Faust (third column in Table 1) We show, from left to right: M_1 , the initial map, our result with orthogonality guidance (FRIDU), our result with with orthogonality guidance and upsampling, the DiffZo result and the ground truth. We note that the initial map is very noisy, and FRIDU with upsampling achieves a very similar result to DiffZo despite the difficult initialization.

time but improves performance only up to a point. The parameter s exhibits a balance point, where values that are too low or too high degrade performance. Overall, we find that our chosen parameters achieve a good trade-off between inference time and error in this example.

4. Limitations

As a refinement method, our approach is somewhat sensitive to the initial map. For example, as we show in Figure 9, given a very bad initialization our approach, while improving the map, does not fully reach the optimal solution. This is further exacerbated by training on FAUST and testing on SHREC19, which leads to additional generalization challenges, as can be seen in Table 1. In addition, we achieve only comparable performance to ZoomOut, when starting from the same initial map. However, in a setup where *many* maps between meshes from the same class need to be improved (e.g., if we want to compute all the pairwise maps for a large dataset of meshes), our approach can offer a considerable speedup by first training on a subset of the dataset, and then efficiently testing on the rest of the shapes. Finally, we have only experimented with *square* functional maps, whereas in practice one may prefer to use rectangular maps (e.g., using more basis functions on the target than on the source). We note that our

m	k	s	↓ Inference (s)	↓ Mean Error
2	5	500	16.20	0.10
2	10	500	31.72	0.10
10	5	500	45.70	0.10
1	5	500	9.18	0.12
2	5	50	16.29	0.15
2	5	5000	16.19	0.23
2	1	500	3.68	0.28

Table 2: Guidance Parameters. Inference time and mean error for different values of the guidance parameters m , k , and s from [BCS*23]. The best result in each column is shown in bold. We note that our chosen parameters (first row) provide a good balance between inference time and accuracy.

algorithm extends naturally to this case, and all the guidance objectives can be used as-is. Yet we leave further validation to future work.

5. Conclusion and Future Work

We introduced FRIDU, a novel framework for refining functional maps via guided diffusion in the spectral domain. By interpreting functional maps as images, FRIDU leverages the generative capacity of diffusion models while enabling inference-time guidance to steer generation toward task-specific geometric properties. These objectives are applied modularly at test time, without requiring supervision or retraining, leading to high-quality correspondence refinement across datasets and diverse sources of initial maps.

We believe that diffusion models can be useful in additional aspects of the functional framework. For example, shape difference operators [ROA*13] and functional vector fields [ABCCO13] are also operators which are potentially amenable to similar treatment using image diffusion models. Further, a composition of multiple maps (or of self-maps, generated by a tangent vector field) could potentially be analyzed using tools for videos.

Looking ahead, a major direction is to move beyond training a separate model for each type of input map. Instead, we envision *foundation models* [BHA*21] for functional maps: diffusion-based priors trained on broad distributions of noisy correspondences that generalize across shape categories, datasets, and correspondence pipelines. Such models would support *zero-shot refinement* via guidance alone, inspired by plug-and-play techniques developed in the diffusion literature, enabling modular adaptation without retraining. FRIDU lays the groundwork for this direction by demonstrating that flexible, loss-based guidance in the spectral domain is highly effective. To realize this vision, new forms of guidance may be required. One promising direction is to move beyond hand-crafted losses and explore *learned guidance* techniques such as neural critics or reward models [FWD*23], which could facilitate high-level constraints.

By connecting spectral geometry with modern generative modeling, FRIDU establishes a foundation for such cross-domain exploration. We hope this bridge between two previously unrelated paradigms will spur future work at the intersection of geometric analysis and diffusion-based generative modeling.

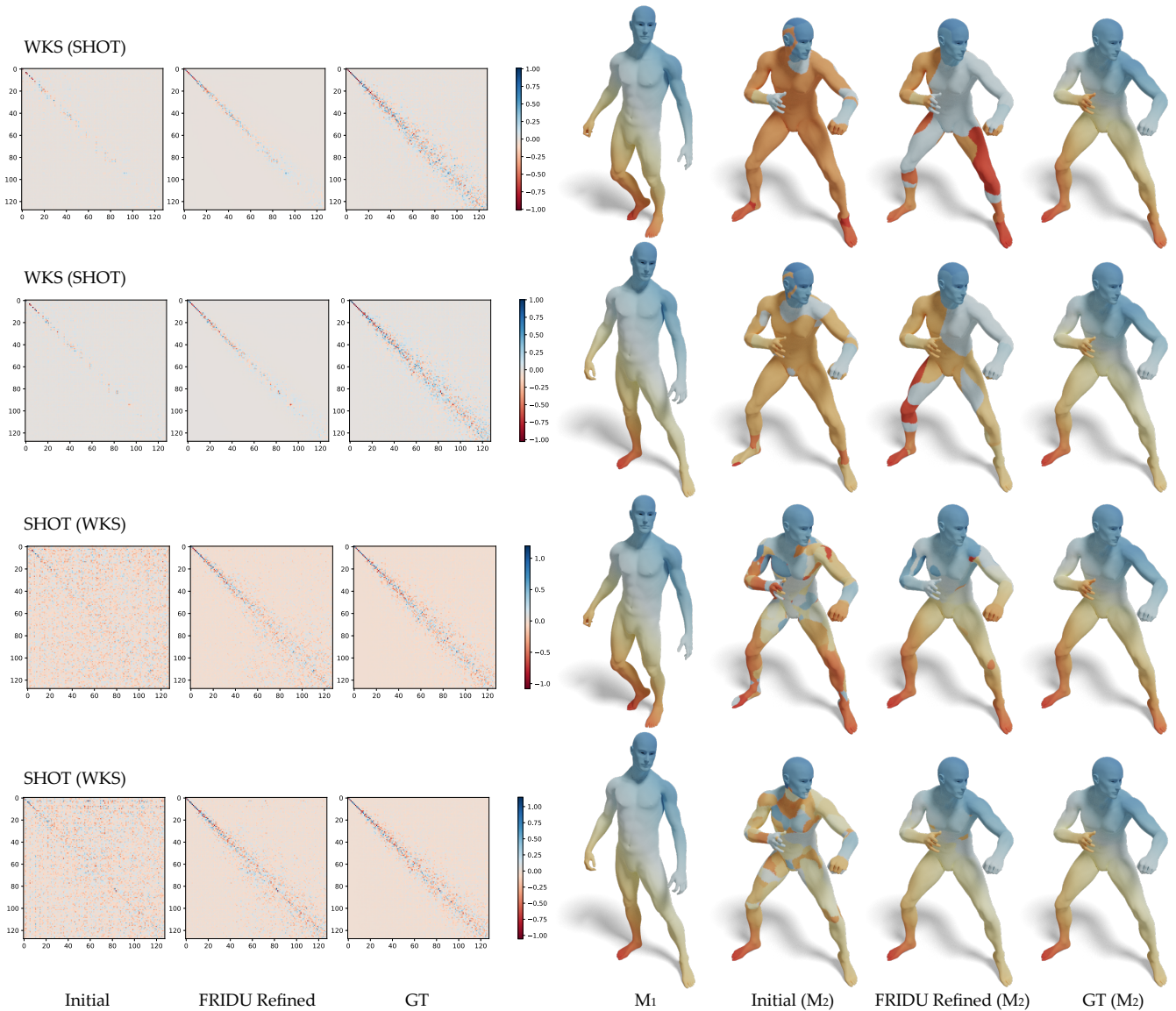


Figure 10: Zero-Shot Condition Generalization We test zero-shot condition generalization of our model by refining *SHOT*-based maps using a model trained on *WKS*-based maps, and vice versa, without additional training. The top two rows show the refinement of *WKS*-based maps using the *SHOT*-trained model, denoted as *WKS (SHOT)*, while the bottom two rows show the refinement of *SHOT*-based maps using the *WKS*-trained model, denoted as *SHOT (WKS)*. We observe that our model refines in both cases, with *SHOT (WKS)* achieving better performance than *WKS (SHOT)*.

Acknowledgments. Avigail Cohen Rimon and Mirela Ben Chen acknowledge the support of the Israel Science Foundation (grant No. 1073/21). Or Litany is a Taub fellow and is supported by the Azrieli Foundation Early Career Faculty Fellowship.

References

- [ABCCO13] AZENCOT O., BEN-CHEN M., CHAZAL F., OVSJANIKOV M.: An operator approach to tangent vector field processing. In *Computer Graphics Forum* (2013), vol. 32, Wiley Online Library, pp. 73–82. [9](#)
- [APO21] ATTAIKI S., PAI G., OVSJANIKOV M.: Dpfm: Deep partial functional maps. In *2021 International Conference on 3D Vision (3DV)* (2021), IEEE, pp. 175–185. [2](#)
- [ASC11] AUBRY M., SCHLICKWEI U., CREMERS D.: The wave kernel signature: A quantum mechanical approach to shape analysis. In *2011 IEEE international conference on computer vision workshops (ICCV workshops)* (2011), IEEE, pp. 1626–1633. [3, 6](#)
- [ASK*05] ANGUELOV D., SRINIVASAN P., KOLLER D., THRUN S., RODGERS J., DAVIS J.: Scape: shape completion and animation of people. In *ACM Siggraph 2005 Papers*. 2005, pp. 408–416. [7](#)
- [BCS*23] BANSAL A., CHU H.-M., SCHWARZSCHILD A., SENGUPTA

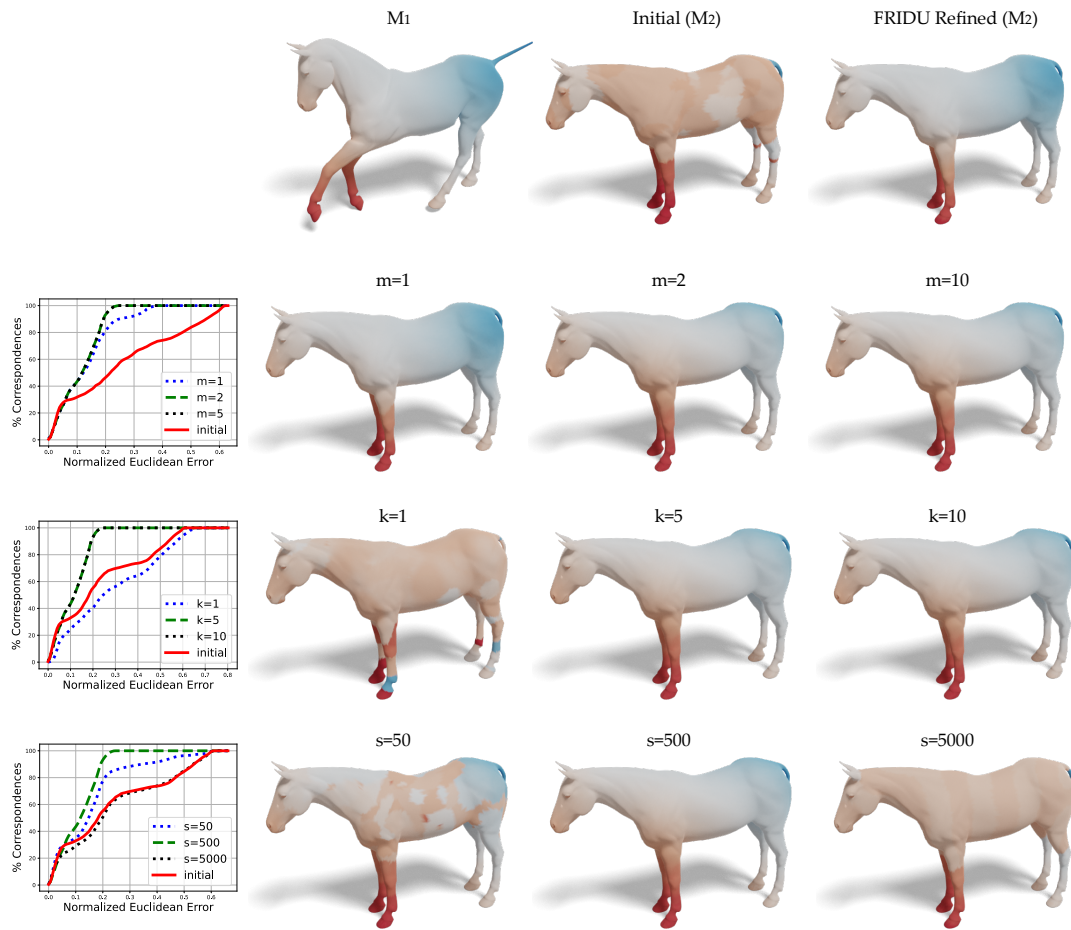


Figure 11: Guidance Parameters. Ablation study of the guidance parameters m , k , and s . Top row: the source function on \mathcal{M}_1 and its mapping using the initial map and our refined map with our chosen parameters: $m = 2$, $k = 5$, $s = 500$. In the three subsequent rows, we vary each parameter independently while keeping the other two fixed. The plots on the left show the corresponding normalized Euclidean error:

- S., GOLDBLUM M., GEIPING J., GOLDSTEIN T.: Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 843–852. 2, 3, 5, 8, 9, 14
- [BDK24] BRACHA A., DAGÈS T., KIMMEL R.: On unsupervised partial shape correspondence. In *Proceedings of the Asian Conference on Computer Vision* (2024), pp. 4488–4504. 2
- [BHA*21] BOMMASANI R., HUDSON D. A., ADELI E., ALTMAN R., ARORA S., VON ARX S., BERNSTEIN M. S., BOHG J., BOSSELUT A., BRUNSKILL E., BRYNJOLFSSON E., BUCH S., CARD D., CASTELLON R., CHATTERJI N. S., CHEN A. S., CREEL K. A., DAVIS J., DEMSZKY D., DONAHUE C., DOUMBOUYA M., DURMUS E., ERMON S., ÉTCHEMENDY J., ETHAYARAJH K., FEI-FEI L., FINN C., GALE T., GILLESPIE L. E., GOEL K., GOODMAN N. D., GROSSMAN S., GUHA N., HASHIMOTO T., HENDERSON P., HEWITT J., HO D. E., HONG J., HSU K., HUANG J., ICARD T. F., JAIN S., JURAFSKY D., KALLURI P., KARAMCHETI S., KEELING G., KHANI F., KHATTAB O., KOH P. W., KRASS M. S., KRISHNA R., KUDITIPUDI R., KUMAR A., LADHAK F., LEE M., LEE T., LESKOVEC J., LEVENT I., LI X. L., LI X., MA T., MALIK A., MANNING C. D., MIRCHANDANI S. P., MITCHELL E., MUNYIKWA Z., NAIR S., NARAYAN A., NARAYANAN D., NEWMAN B., NIE A., NIEBLES J. C., NILFOROSHAN H., NYARKO J. F., OGUT G., ORR L., PAPADIMITRIOU I., PARK J. S., PIECH C., PORTELANCE E., POTTS C., RAGHUNATHAN A., REICH R., REN H., RONG F., ROOHANI Y. H., RUIZ C., RYAN J., R’E C., SADIGH D., SAGAWA S., SANTHANAM K., SHIH A., SRINIVASAN K. P., TAMKIN A., TAORI R., THOMAS A. W., TRAMÈR F., WANG R. E., WANG W., WU B., WU J., WU Y., XIE S. M., YASUNAGA M., YOU J., ZAHARIA M. A., ZHANG M., ZHANG T., ZHANG X., ZHANG Y., ZHENG L., ZHOU K., LIANG P.: On the opportunities and risks of foundation models. *ArXiv* (2021). URL: <https://crfm.stanford.edu/assets/report.pdf>. 9
- [BRLB14] BOGO F., ROMERO J., LOPER M., BLACK M. J.: Faust: Dataset and evaluation for 3d mesh registration. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2014), pp. 3794–3801. 7
- [BXNL24] BASTIAN L., XIE Y., NAVAB N., LÄHNER Z.: Hybrid functional maps for crease-aware non-isometric shape matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), pp. 3313–3323. 1
- [CB22] CAO D., BERNARD F.: Unsupervised deep multi-shape matching. In *European Conference on Computer Vision* (2022), Springer, pp. 55–71. 5, 8
- [CGC*23] CHEN H., GU J., CHEN A., TIAN W., TU Z., LIU L., SU H.: Single-stage diffusion nerf: A unified approach to 3d generation and re-

- construction. In *Proceedings of the IEEE/CVF international conference on computer vision* (2023), pp. 2416–2425. [2](#)
- [CKM*22] CHUNG H., KIM J., MCCANN M. T., KLASKY M. L., YE J. C.: Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687* (2022). [2](#), [3](#), [5](#)
- [CLY23] CHUNG H., LEE S., YE J. C.: Decomposed diffusion sampler for accelerating large-scale inverse problems. *arXiv preprint arXiv:2303.05754* (2023). [3](#)
- [CRB23] CAO D., ROETZER P., BERNARD F.: Unsupervised learning of robust spectral shape matching. *arXiv preprint arXiv:2304.14419* (2023). [8](#)
- [CSRY22] CHUNG H., SIM B., RYU D., YE J. C.: Improving diffusion models for inverse problems using manifold constraints. *Advances in Neural Information Processing Systems 35* (2022), 25683–25696. [3](#)
- [CZX*24] CHEN Y., ZHENG C., XU H., ZHUANG B., VEDALDI A., CHAM T.-J., CAI J.: Mvsplat360: Feed-forward 360 scene synthesis from sparse views. *arXiv preprint arXiv:2411.04924* (2024). [3](#)
- [DCO22] DONATI N., CORMAN E., OVSJANIKOV M.: Deep orientation-aware functional maps: Tackling symmetry issues in shape matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 742–751. [8](#)
- [DMM24] DUTT N. S., MURALIKRISHNAN S., MITRA N. J.: Diffusion 3d features (diff3f): Decorating untextured shapes with distilled semantic features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), pp. 4494–4504. [3](#)
- [DN21] DHARIWAL P., NICHOL A.: Diffusion models beat gans on image synthesis. *Advances in neural information processing systems 34* (2021), 8780–8794. [2](#), [3](#)
- [DSO20] DONATI N., SHARMA A., OVSJANIKOV M.: Deep geometric functional maps: Robust feature learning for shape correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 8592–8601. [8](#)
- [DYDZ22] DENG B., YAO Y., DYKE R. M., ZHANG J.: A survey of non-rigid 3d registration. In *Computer Graphics Forum* (2022), vol. 41, Wiley Online Library, pp. 559–589. [2](#)
- [EBC17] EZUZ D., BEN-CHEN M.: Deblurring and denoising of maps between shapes. In *Computer Graphics Forum* (2017), vol. 36, Wiley Online Library, pp. 165–174. [2](#), [5](#), [8](#)
- [ELC20] EISENBERGER M., LAHNER Z., CREMERS D.: Smooth shells: Multi-scale shape registration with functional maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 12265–12274. [8](#)
- [ENK*21] EISENBERGER M., NOVOTNY D., KERCHENBAUM G., LABATUT P., NEVEROVA N., CREMERS D., VEDALDI A.: Neuromorph: Unsupervised shape interpolation and correspondence in one go. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 7473–7483. [8](#)
- [ESBC19] EZUZ D., SOLOMON J., BEN-CHEN M.: Reversible harmonic maps between discrete surfaces. *ACM Transactions on Graphics (ToG)* 38, 2 (2019), 1–12. [2](#)
- [ETLTC20] EISENBERGER M., TOKER A., LEAL-TAIXÉ L., CREMERS D.: Deep shells: Unsupervised shape correspondence with optimal transport. *Advances in Neural Information Processing Systems 33* (2020), 10491–10502. [8](#)
- [FWD*23] FAN Y., WATKINS O., DU Y., LIU H., RYU M., BOUTILIER C., ABBEEL P., GHAVAMZADEH M., LEE K., LEE K.: Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems 36* (2023), 79858–79885. [9](#)
- [GMJS22] GRAIKOS A., MALKIN N., JOJIC N., SAMARAS D.: Diffusion models as plug-and-play priors. *Advances in Neural Information Processing Systems 35* (2022), 14715–14728. [3](#)
- [GXN*23] GUPTA A., XIONG W., NIE Y., JONES I., OĞUZ B.: 3dgen: Triplane latent diffusion for textured mesh generation. *arXiv preprint arXiv:2303.05371* (2023). [2](#)
- [HCP*24] HE X., CHEN J., PENG S., HUANG D., LI Y., HUANG X., YUAN C., OUYANG W., HE T.: Gvgen: Text-to-3d generation with volumetric representation. In *European Conference on Computer Vision* (2024), Springer, pp. 463–479. [3](#)
- [HJA20] HO J., JAIN A., ABBEEL P.: Denoising diffusion probabilistic models. *Advances in neural information processing systems 33* (2020), 6840–6851. [2](#), [3](#)
- [HLHF22] HUI K.-H., LI R., HU J., FU C.-W.: Neural wavelet-domain diffusion for 3d shape generation. In *SIGGRAPH Asia 2022 Conference Papers* (2022), pp. 1–9. [2](#)
- [HLR*19a] HALIMI O., LITANY O., RODOLA E., BRONSTEIN A. M., KIMMEL R.: Unsupervised learning of dense shape correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019). [1](#)
- [HLR*19b] HALIMI O., LITANY O., RODOLA E., BRONSTEIN A. M., KIMMEL R.: Unsupervised learning of dense shape correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 4370–4379. [15](#)
- [HML*23] HE Y., MURATA N., LAI C.-H., TAKIDA Y., UESAKA T., KIM D., LIAO W.-H., MITSUFUJI Y., KOLTER J. Z., SALAKHUTDINOV R., ET AL.: Manifold preserving guided diffusion. *arXiv preprint arXiv:2311.16424* (2023). [2](#), [3](#)
- [HS22] HO J., SALIMANS T.: Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598* (2022). [3](#)
- [HSA*23] HARTWIG F., SASSEN J., AZENCOT O., RUMPF M., BEN-CHEN M.: An elastic basis for spectral shape correspondence. In *ACM SIGGRAPH 2023 conference proceedings* (2023), pp. 1–11. [1](#)
- [JLY*24] JE J., LIU J., YANG G., DENG B., CAI S., WETZSTEIN G., LITANY O., GUIBAS L.: Robust symmetry detection via riemannian langevin dynamics. In *SIGGRAPH Asia 2024 Conference Papers* (2024), pp. 1–11. [3](#)
- [KAAL22] KARRAS T., AITTALA M., AILA T., LAINE S.: Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems 35* (2022), 26565–26577. [3](#), [4](#), [14](#)
- [KEES22] KAWAR B., ELAD M., ERMON S., SONG J.: Denoising diffusion restoration models. *Advances in Neural Information Processing Systems 35* (2022), 23593–23606. [3](#)
- [LDO22] LI L., DONATI N., OVSJANIKOV M.: Learning multi-resolution functional maps with spectral attention for robust shape matching. *Advances in Neural Information Processing Systems 35* (2022), 29336–29349. [8](#)
- [LDR*22] LUGMAYR A., DANELLJAN M., ROMERO A., YU F., TIMOFTE R., VAN GOOL L.: Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2022), pp. 11461–11471. [3](#)
- [LFB*23] LIU Z., FENG Y., BLACK M. J., NOWROUZSAHRAI D., PAULL L., LIU W.: Meshdiffusion: Score-based generative 3d mesh modeling. *arXiv preprint arXiv:2303.08133* (2023). [2](#)
- [LGL*24] LONG X., GUO Y.-C., LIN C., LIU Y., DOU Z., LIU L., MA Y., ZHANG S.-H., HABERMANN M., THEOBALT C., ET AL.: Wonder3d: Single image to 3d using cross-domain diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2024), pp. 9970–9980. [2](#)
- [LH21] LUO S., HU W.: Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2021), pp. 2837–2845. [2](#)
- [Liu18] LIU H.-T. D.: Blender Toolbox, 12 2018. URL: <https://github.com/HTDerekLiu/BlenderToolbox>. [15](#)
- [LJO19] LIU H.-T. D., JACOBSON A., OVSJANIKOV M.: Spectral coarsening of geometric operators. *arXiv preprint arXiv:1905.05161* (2019). [15](#)

- [LLHL20] LI Q., LIU S., HU L., LIU X.: Shape correspondence using anisotropic chebyshev spectral cnns. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition* (2020), pp. 14658–14667. 8
- [LRB*16] LITANY O., RODOLÀ E., BRONSTEIN A., BRONSTEIN M., CREMERS D.: Non-rigid puzzles. In *Computer Graphics Forum* (2016), vol. 35, The Eurographics Association and John Wiley & Sons Ltd., pp. 135–143. 2
- [LRBB17] LITANY O., RODOLÀ E., BRONSTEIN A. M., BRONSTEIN M. M.: Fully spectral partial shape matching. In *Computer Graphics Forum* (2017), vol. 36, The Eurographics Association and John Wiley & Sons Ltd. 2
- [LRR*17] LITANY O., REMEZ T., RODOLA E., BRONSTEIN A., BRONSTEIN M.: Deep functional maps: Structured prediction for dense shape correspondence. In *Proceedings of the IEEE international conference on computer vision* (2017), pp. 5659–5667. 1
- [LSdOBG*24] LUKOIANOV A., SÁEZ DE OCÁRIZ BORDE H., GREENEWALD K., GUZILINI V., BAGAUTDINOV T., SITZMANN V., SOLOMON J. M.: Score distillation via reparametrized ddim. *Advances in Neural Information Processing Systems 37* (2024), 26011–26044. 3
- [LTQ*23] LAN Y., TAN F., QIU D., XU Q., GENOVA K., HUANG Z., FANELLO S., PANDEY R., FUNKHOUSER T., LOY C. C., ET AL.: Gaussian3diff: 3d gaussian diffusion for 3d full head synthesis and editing. *arXiv preprint arXiv:2312.03763* (2023). 3
- [LWT23] LI X., WANG H., TSENG K.-K.: Gaussiandiffusion: 3d gaussian splatting for denoising diffusion probabilistic models with structured noise. *arXiv preprint arXiv:2311.11221* (2023). 3
- [LWVH*23] LIU R., WU R., VAN HOORICK B., TOKMAKOV P., ZAKHAROV S., VONDRICK C.: Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision* (2023), pp. 9298–9309. 3
- [LXC*24] LIU J., XU J., CHENG W., GAO Y., WANG X., SHAN Y., TANG Y.: Novelgs: Consistent novel-view denoising via large gaussian reconstruction model, 2024. URL: <https://arxiv.org/abs/2411.16779>, [arXiv:2411.16779](https://arxiv.org/abs/2411.16779). 3
- [Mag21] MAGNET R.: pyfm: Functional maps in python. <https://github.com/RobinMagnet/pyFM>, 2021. Accessed: 2024-04-10. 6, 14
- [MKRV23] MELAS-KYRIAZI L., RUPPRECHT C., VEDALDI A.: Pc2: Projection-conditioned point cloud diffusion for single-image 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 12923–12932. 2
- [MMR*19] MELZI S., MARIN R., RODOLÀ E., CASTELLANI U., REN J., POULENARD A., OVSJANIKOV P., ET AL.: Shrec’19: matching humans with different connectivity. In *Eurographics Workshop on 3D Object Retrieval* (2019), The Eurographics Association, pp. 1–8. 7
- [MO24] MAGNET R., OVSJANIKOV M.: Memory-scalable and simplified functional map learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), pp. 4041–4050. 2, 8, 9, 15
- [MRR*19] MELZI S., REN J., RODOLA E., SHARMA A., WONKA P., OVSJANIKOV M.: Zoomout: Spectral upsampling for efficient shape correspondence. *arXiv preprint arXiv:1904.07865* (2019). 2, 8
- [MRSHO22] MAGNET R., REN J., SORKINE-HORNUNG O., OVSJANIKOV M.: Smooth non-rigid shape matching via effective dirichlet energy optimization. In *2022 International Conference on 3D Vision (3DV)* (2022), IEEE, pp. 495–504. 2
- [MSP*23] MÜLLER N., SIDDIQUI Y., PORZI L., BULO S. R., KONTSCHIEDER P., NIESSNER M.: Diffrr: Rendering-guided 3d radiance field diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 4328–4338. 2
- [NJD*22] NICHOL A., JUN H., DHARIWAL P., MISHKIN P., CHEN M.: Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751* (2022). 2
- [NMR*18] NOGNENG D., MELZI S., RODOLA E., CASTELLANI U., BRONSTEIN M., OVSJANIKOV M.: Improved functional mappings via product preservation. In *Computer Graphics Forum* (2018), vol. 37, Wiley Online Library, pp. 179–190. 2
- [OBGS*12] 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 (2012), 1–11. 1, 2
- [OCB*16] OVSJANIKOV M., CORMAN E., BRONSTEIN M., RODOLÀ E., BEN-CHEN M., GUIBAS L., CHAZAL F., BRONSTEIN A.: Computing and processing correspondences with functional maps. In *SIG-GRAPH ASIA 2016 Courses*. 2016, pp. 1–60. 1, 2
- [PBJM22] POOLE B., JAIN A., BARRON J. T., MILDENHALL B.: Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988* (2022). 3
- [PMM24] PEDICO S., MELZI S., MAGGIOLI F.: Taco: a benchmark for connectivity-invariance in shape correspondence. In *Smart Tools and Applications in Graphics-Eurographics Italian Chapter Conference* (2024). 6
- [PRM*21] PAI G., REN J., MELZI S., WONKA P., OVSJANIKOV M.: Fast sinkhorn filters: Using matrix scaling for non-rigid shape correspondence with functional maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 384–393. 2
- [PST*24] PENG C., SOBOL I., TOMIZUKA M., KEUTZER K., XU C., LITANY O.: A lesson in splats: Teacher-guided diffusion for 3d gaussian splats generation with 2d supervision, 2024. URL: <https://arxiv.org/abs/2412.00623>, [arXiv:2412.00623](https://arxiv.org/abs/2412.00623). 3
- [RBL*22] ROMBACH R., BLATTMANN A., LORENZ D., ESSER P., OMMER B.: High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2022), pp. 10684–10695. 2
- [RCB*17] RODOLÀ E., COSMO L., BRONSTEIN M. M., TORSSELLO A., CREMERS D.: Partial functional correspondence. In *Computer graphics forum* (2017), vol. 36, Wiley Online Library, pp. 222–236. 2
- [RLP*23] REMPE D., LUO Z., PENG X. B., YUAN Y., KITANI K., KREIS K., FIDLER S., LITANY O.: Trace and pace: Controllable pedestrian animation via guided trajectory diffusion. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2023). 3
- [RMWO21] REN J., MELZI S., WONKA P., OVSJANIKOV M.: Discrete optimization for shape matching. In *Computer Graphics Forum* (2021), vol. 40, Wiley Online Library, pp. 81–96. 8
- [ROA*13] 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), 1–12. 9
- [RPWO18] REN J., POULENARD A., WONKA P., OVSJANIKOV M.: Continuous and orientation-preserving correspondences via functional maps. *ACM Transactions on Graphics (ToG)* 37, 6 (2018), 1–16. 2, 7, 8
- [RPWO19] REN J., PANINE M., WONKA P., OVSJANIKOV M.: Structured regularization of functional map computations. In *Computer Graphics Forum* (2019), vol. 38, Wiley Online Library, pp. 39–53. 5
- [SACO22] SHARP N., ATTAIKI S., CRANE K., OVSJANIKOV M.: Diffusionnet: Discretization agnostic learning on surfaces. *ACM Transactions on Graphics (TOG)* 41, 3 (2022), 1–16. 15
- [SCP*23] SHUE J. R., CHAN E. R., PO R., ANKNER Z., WU J., WETZSTEIN G.: 3d neural field generation using triplane diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 20875–20886. 2
- [SDWMG15] SOHL-DICKSTEIN J., WEISS E., MAHESWARANATHAN N., GANGULI S.: Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning* (2015), pmlr, pp. 2256–2265. 2

- [SE19] SONG Y., ERMON S.: Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems* 32 (2019). 2
- [SMJ*23a] SUN M., MAO S., JIANG P., OVSJANIKOV M., HUANG R.: Spatially and spectrally consistent deep functional maps, 2023. URL: <https://arxiv.org/abs/2308.08871>, arXiv:2308.08871. 1
- [SMJ*23b] SUN M., MAO S., JIANG P., OVSJANIKOV M., HUANG R.: Spatially and spectrally consistent deep functional maps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), pp. 14497–14507. 8
- [SRV23] SZYMANOWICZ S., RUPPRECHT C., VEDALDI A.: Viewset diffusion: (0-)image-conditioned 3d generative models from 2d data. *International Conference on Computer Vision* (2023). 3
- [STDS14] SALTI S., TOMBARI F., DI STEFANO L.: Shot: Unique signatures of histograms for surface and texture description. *Computer vision and image understanding* 125 (2014), 251–264. 6
- [SXL24] SOBOL I., XU C., LITANY O.: Zero-to-hero: Enhancing zero-shot novel view synthesis via attention map filtering. *Advances in Neural Information Processing Systems* 37 (2024), 30522–30553. 3
- [SZY*23] SONG J., ZHANG Q., YIN H., MARDANI M., LIU M.-Y., KAUTZ J., CHEN Y., VAHDAT A.: Loss-guided diffusion models for plug-and-play controllable generation. In *International Conference on Machine Learning* (2023), PMLR, pp. 32483–32498. 2, 5
- [TCM*21] TRAPPOLINI G., COSMO L., MOSCHELLA L., MARIN R., MELZI S., RODOLÀ E.: Shape registration in the time of transformers. *Advances in Neural Information Processing Systems* 34 (2021), 5731–5744. 8
- [TGW*23] TANG Z., GU S., WANG C., ZHANG T., BAO J., CHEN D., GUO B.: Volumediffusion: Flexible text-to-3d generation with efficient volumetric encoder. *arXiv preprint arXiv:2312.11459* (2023). 2
- [TJW*23] TANG L., JIA M., WANG Q., PHOO C. P., HARIHARAN B.: Emergent correspondence from image diffusion. *Advances in Neural Information Processing Systems* 36 (2023), 1363–1389. 3
- [TRZ*23] TANG J., REN J., ZHOU H., LIU Z., ZENG G.: Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653* (2023). 3
- [TSDS10] TOMBARI F., SALTI S., DI STEFANO L.: Unique signatures of histograms for local surface description. In *Computer Vision—ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5–11, 2010, Proceedings, Part III 11* (2010), Springer, pp. 356–369. 3
- [WBCPS18] WANG Y., BEN-CHEN M., POLTEROVICH I., SOLOMON J.: Steklov spectral geometry for extrinsic shape analysis. *ACM Transactions on Graphics (TOG)* 38, 1 (2018), 1–21. 1
- [WCMB*22] WATSON D., CHAN W., MARTIN-BRUALLA R., HO J., TAGLIASACCHI A., NOROUZI M.: Novel view synthesis with diffusion models, 2022. URL: <https://arxiv.org/abs/2210.04628>, arXiv:2210.04628. 3
- [WJZ*23] WANG Z., JIANG Y., ZHENG H., WANG P., HE P., WANG Z., CHEN W., ZHOU M., ET AL.: Patch diffusion: Faster and more data-efficient training of diffusion models. *Advances in neural information processing systems* 36 (2023), 72137–72154. 4, 14
- [WWF*23] WU Z., WANG Y., FENG M., XIE H., MIAN A.: Sketch and text guided diffusion model for colored point cloud generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), pp. 8929–8939. 2
- [WYZ22] WANG Y., YU J., ZHANG J.: Zero-shot image restoration using denoising diffusion null-space model. *arXiv preprint arXiv:2212.00490* (2022). 3
- [WZG*25] WU T., ZHENG C., GUAN F., VEDALDI A., CHAM T.-J.: Amodal3r: Amodal 3d reconstruction from occluded 2d images. *arXiv preprint arXiv:2503.13439* (2025). 3
- [WZZ*23] WANG T., ZHANG B., ZHANG T., GU S., BAO J., BALTRUSAITIS T., SHEN J., CHEN D., WEN F., CHEN Q., ET AL.: Rodin: A generative model for sculpting 3d digital avatars using diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2023), pp. 4563–4573. 2
- [XLFL24] XU C., LING H., FIDLER S., LITANY O.: 3diffaction: 3d object detection with geometry-aware diffusion features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), pp. 10617–10627. 3
- [XLX*24] XIANG J., LV Z., XU S., DENG Y., WANG R., ZHANG B., CHEN D., TONG X., YANG J.: Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506* (2024). 3
- [YWZ*23] YU J., WANG Y., ZHAO C., GHANEM B., ZHANG J.: Freedom: Training-free energy-guided conditional diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), pp. 23174–23184. 2
- [ZCW*24] ZHANG B., CHENG Y., WANG C., ZHANG T., YANG J., TANG Y., ZHAO F., CHEN D., GUO B.: Rodinhd: High-fidelity 3d avatar generation with diffusion models. In *European Conference on Computer Vision* (2024), Springer, pp. 465–483. 2
- [ZCY*24] ZHANG B., CHENG Y., YANG J., WANG C., ZHAO F., TANG Y., CHEN D., GUO B.: Gaussiancube: A structured and explicit radiance representation for 3d generative modeling. *arXiv preprint arXiv:2403.19655* (2024). 3
- [ZHM*24] ZHENG X., HUANG X., MEI G., HOU Y., LYU Z., DAI B., OUYANG W., GONG Y.: Point cloud pre-training with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), pp. 22935–22945. 2
- [ZLG25] ZHURAVLEV A., LÄHNER Z., GOLYANIK V.: Denoising functional maps: Diffusion models for shape correspondence. *arXiv preprint arXiv:2503.01845* (2025). 3, 6

Appendix A: Implementation Details and Network Parameters

To enable training a diffusion model on a very limited dataset, we adopt the Patch-Diffusion paradigm from [WJZ*23] and build upon the code they provide. Specifically, we use their implementation based on the UNet-based diffusion model EDM-DDPM++ [KAAL22]. Following [WJZ*23], we employ the EDM sampling strategy with 50 deterministic reverse steps during inference.

In addition to default settings, we use the following hyperparameters for the Patch-Diffusion model: `num_blocks = 2`, `model_channels = 16`, `channel_mult = [2, 4]`, and `channel_mult_emb = 0.1`. We train using 3 patch resolutions, where the functional maps have dimension 128×128 in all experiments—except in Section 4.2, where we match the resolution used in DiffZO which is 130×130 . The parameter `duration = 2` is used across all experiments, except in Section 4.2, where we set `duration = 10` for the model trained on FAUST, and `duration = 15` for the model trained on FAUST+SCAPE.

We use a batch size of 32 for all experiments, except for training on FAUST+SCAPE, where we use a batch size of 64. We run all experiments on a single NVIDIA A40 GPU, except for the FAUST+SCAPE training, which we run on NVIDIA L40S GPU.

For guidance, we implement the algorithm described in [BCS*23] with adaptation to the EDM.

Appendix B: Initial Functional Map Computation

We use the `pyFM` Python package [Mag21] to generate initial functional maps for the experiments in Section 4.1, which also includes the computation of WKS descriptors. For the WKS we use the parameters `n_descr =`

100, $subsample_step = 5$ based on 150 eigenvectors. For SHOT descriptors, we compute them in advance using the code from [HLR*19b], with the parameters $num_evecs = 150$, $num_bins = 10$, and $radius = 15$.

Appendix C: DiffZO Comparison

For comparison with the DiffZo method [MO24], we extract the weights of their pretrained Feature Extractor model (DiffusionNet [SACO22]) for models trained on FAUST and FAUST+SCAPE. To compute the initial functional map used in our model, we first compute the pointwise map Π_{init} from the extracted features and then derive the corresponding functional map using the corresponding Laplace-Beltrami eigenbases.

Appendix D: Visualizations

For visualizing functional map matrices, we use the colormap defined in [LJO19], and for rendering in Blender, we utilize BlenderToolbox [Liu18].