DIFFUSION 3D FEATURES (DIFF3F)
Decorating Untextured Shapes with Distilled Semantic Features

Niladri Shekhar Dutt[1,2] Sanjeev Muralikrishnan[1] Niloy J. Mitra[1,3]
[1]University College London [2]Ready Player Me [3]Adobe Research

https://diff3f.github.io/

Figure 1. **Correspondence in-the-wild.** We introduce DIFF3F, a novel feature distiller that harnesses the expressive power of inpainting diffusion features and distills them to points on 3D surfaces. Here, the proposed features are employed for point-to-point shape correspondence between assets varying in shape, pose, species, and topology. We achieve this without any fine-tuning of the underlying diffusion models, and demonstrate results on untextured meshes, point clouds, and raw scans. The leftmost mesh is the source, and all the remaining 3D shapes are targets. Note that we show raw point-to-point correspondence, without any regularization or smoothing. Inputs are point clouds, non-manifold meshes, or 2-manifold meshes. Corresponding points are similarly colored across the shapes.

Abstract

We present DIFF3F *as a simple, robust, and class-agnostic feature descriptor that can be computed for untextured input shapes (meshes or point clouds). Our method distills diffusion features from image foundational models onto input shapes. Specifically, we use the input shapes to produce depth and normal maps as guidance for conditional image synthesis. In the process, we produce (diffusion) features in 2D that we subsequently lift and aggregate on the original surface. Our key observation is that even if the conditional image generations obtained from multi-view rendering of the input shapes are inconsistent, the associated image features are robust and, hence, can be directly aggregated across views. This produces semantic features on the input shapes, without requiring additional data or training. We perform extensive experiments on multiple benchmarks (SHREC'19, SHREC'20, FAUST, and TOSCA) and demonstrate that our features, being semantic instead of geometric, produce reliable correspondence across both isometric and non-isometrically related shape families. Code is available at https://github.com/niladridutt/Diffusion-3D-Features.*

1. Introduction

Feature descriptors are crucial building blocks in most shape analysis tasks. The ability to extract reliable features from input meshes or point clouds paves the way for establishing shape correspondence, extracting low-dimensional shape spaces, and learning 3D generative models, to name a few

applications. Classical geometry processing algorithms, extensively explored over recent decades [9, 18, 35, 41, 55, 60], concentrate on identifying geometric invariants and, at best, coincidentally align with semantic features. Recent learning-based approaches [10, 14, 30], trained on a limited amount of data, learn the correlation between geometric and semantic features but struggle to generalize to unseen categories. In contrast, in image analysis, a recent winner has emerged — foundational models [11, 40, 45, 47] trained on massive image datasets have been repurposed to yield general-purpose feature descriptors [5, 36, 54, 65]. Remarkably, such large-scale models have implicitly learned robust semantic features that match and often surpass classical image feature descriptors. For instance, DINO features [40] and diffusion features [54] extract dense semantic image features from photo-realistic images *without* additional training. In this paper, we investigate the adaptation of this success to the realm of 3D shapes.

A significant challenge is to address the absence of textures on most 3D models. This hinders immediate rendering to produce photo-realistic images required by image-based feature detectors mentioned earlier. Additionally, when shapes are represented as meshes, they may have non-manifold faces, making it challenging to extract UV parameterizations; when shapes are represented as point clouds, they lack connectivity information, making rendering ambiguous. One potential solution for input meshes involves utilizing recent approaches [12, 46] to first generate seamlessly textured meshes through image-guided techniques and subsequently extract image feature descriptors. For point cloud representation, one can first produce surface reconstructions [24], then employ the aforementioned mesh-based approach. However, these methods are cumbersome, optimization-based, and unsuitable for seamless end-to-end workflows. We propose a simple and robust solution.

We present DIFFUSION 3D FEATURES (DIFF3F), a simple and practical framework for extracting semantic features that eliminates the need for additional training or optimization. DIFF3F renders input shapes from a sampling of camera views to produce respective depth/normal maps. These maps are used as guidance in ControlNet [66] to condition Stable Diffusion [47] to produce photo-realistic images. We directly use the features on these images produced during diffusion and aggregate them back on the input surfaces. Our main insight is that we do *not* require consistent mesh texturing to produce reliable shape features as we 'denoise' smaller inconsistencies in the feature aggregation step. Since we use diffusion features, our approach reuses the intermediate information generated in the depth-guided image generation step, thus avoiding any additional training. The extracted features produce accurate semantic correspondence across diverse input shapes (see Figure 1), even under significant shape variations.

Table 1. **Comparison of DIFFUSION 3D FEATURES to related methods.** Unlike traditional geometric feature detectors (e.g., WKS), modern learning-based approaches require training and can struggle to generalize to novel settings. We leverage strong image priors in the form of image diffusion models to directly decorate input shapes with distilled semantic features.

	3D-CODED [20]	DPC [30]	SE-ORNet [14]	FM+WKS [41]	Ours
No 3D training data?	✗	✗	✗	✓	✓
Unsupervised?	✗	✓	✓	✓	✓
Class agnostic?	✗	✗	✗	✓	✓
Handles meshes?	✓	✓	✓	✓	✓
Handles point cloud?	✓	✓	✓	✗	✓

We evaluate our algorithm on a range of input shapes (meshes and point clouds) and compare the quality of the extracted features on a set of established correspondence benchmarks. We study the importance of feature aggregation versus consistency of multi-view image generations, choice of the (image) features used, and robustness to input specification. We report comparable performance to state-of-the-art algorithms on multiple benchmarks, containing isometric and non-isometric variations, and outperform existing approaches in generalizability. Our main contribution is a simple and surprisingly effective semantic feature detection algorithm on 3D shapes that can be readily integrated into existing shape analysis workflows without requiring extra data or training. DIFF3F, being semantic, is complementary to existing geometric features.

2. Related Work

Point-to-point based shape correspondence. These methods, either by formulation or by explicit supervision, train algorithms to map points to points between surfaces. In other words, they establish a discrete point-to-point map instead of a continuous surface map. 3D-Coded [20] finds the correspondence between shape pairs by estimating a transformation between two point clouds. This transformation is learned by deforming a template shape to learn its reconstruction on different shapes. Elementary [15] extends this concept further by trying to find an ideal set of primitives to represent a shape collection. Many such algorithms require ground truth for training such as DCP [58], RPMNet [62], GeomFMap [16], 3D-Coded [20], Elementary [15], and/or mesh connectivity such as GeomFMap [16] and SURFMNet [48], both of which are hard to acquire in the real world.

Recent efforts have focused on unsupervised methods for learning on point clouds. CorrNet3D [64] first learns the feature embeddings using a shared DGCNN [60] and then utilizes a symmetry deformation module to learn the reconstruction and compute correspondence. DGCNN [60] uses a graph with multiple layers of EdgeConv [60] to learn feature

embeddings by incorporating information from local neighborhoods. DPC [30] employs self and cross-reconstruction modules to learn discriminative and smooth representations and uses DGCNN for learning the per-point feature embeddings. To improve predictions for symmetrical parts, SE-ORNet [14] first aligns the source and target point clouds with an orientation estimation module before using a teacher-student model and a DGCNN backbone to find the correspondence. These methods operate directly at the geometry level and fail to understand semantic features that may not be represented directly as geometric features.

Surface maps based shape correspondence. Surface map methods learn a continuous map between two arbitrary 2-manifold surfaces. The learned map can then be sampled for point-to-point correspondence if needed. Classically, these works map eigenfunctions defined on surfaces leading to a functional mapping [16, 41] or they construct an atlas of maps (charts) from $\mathbb{R}^2 \mapsto \mathbb{R}^n$ $(n = 2, 3)$ [4, 37, 39]. Usually, algorithms compute a specific type of surface map aiming to preserve specific geometric properties - preservation of angles [32, 34] for conformal maps, preservation of geodesic distances [44, 52] for isometric maps, etc. The unifying idea is to map both surfaces to a base domain, which can be a mesh [28, 31, 49] or a planar region [2, 3] thus mapping via the shared domain. For example, SURFMNet [48] extends FMNet [35] to an unsupervised setting by enforcing pre-desired structural properties on estimated functional maps. These methods require meshes, often 2-manifold; our work by design can find correspondences between poorly reconstructed meshes with artifacts or directly between point clouds. Functional maps rely on geometric descriptors such as WKS [6] to compute the mapping.

Multi-view rendering based learning. Projective analysis [59] encodes shapes as collections of 2D projections, performs image space analysis, and projects the results back to the 3D. This is a powerful idea. This family of multi-view rendering-based methods has shown remarkable performance on a variety of 3D tasks including shape/object recognition [53, 56, 57, 61, 63, 63], human pose estimation [33], part correspondence [23], reconstruction[43], segmentation [27, 50, 56, 59] and many more. The approach involves rendering a 3D shape from multiple views and extracting information by employing visual descriptors per view, usually obtained by training a CNN in a supervised setting. Various approaches have been suggested to aggregate the features from different views like averaging, max pooling [23, 56], concatenating the image features, using another CNN to fuse the intermediate latent representations pooled from different views [53], etc.

We are inspired by Huang et al. [23], who learn to aggregate descriptors by fine-tuning AlexNet [29] on multi-scale renders of shapes from different viewing directions. The entire network is trained in a supervised setting using contrastive loss [22] to group semantically and geometrically similar points close in the descriptor space. The method, however, suffers from limited correspondence accuracy for lower tolerance levels (due to its noisy dataset) compared to state-of-the-art geometric methods available today.

A more recent work [1] explored the usage of foundational image models for generating zero-shot correspondence. They use LLMs and vision models to first generate a set of segmentation maps for each object and a semantic mapping between each set. This is followed by a 3D semantic segmentation model based on SAM [25] to segment the shape according to the generated set. Based on the segmented areas, geometric descriptors are initialized to compute a functional map. Finally, they apply iterative refinement to produce a final point-to-point correspondence. Instead, we distill an image foundational model to produce descriptors with rich semantic features that can be directly used to compute correspondence.

Aggregating 3D features from 2D foundational models. 3D Highlighter [13] renders a mesh from multiple views and calculates their CLIP [45] embedding. Distilled Feature Field [26] distills CLIP or DINO embeddings to 3D feature fields to enable zero-shot segmentation of Neural Radiance Fields. NeRF Analogies [19] further shows that DINO features on multi-view rendered images can be used to calculate correspondence for semantically transferring the appearance of a source NeRF to a target 3D geometry. We utilize ControlNet [66] in-the-loop to generate multi-view inconsistent textured renderings and, in that process, generate diffusion features that are semantically coherent to compute accurate correspondence.

3. Method

We aim to decorate 3D points of a given shape in any modality – point clouds or meshes – with rich semantic descriptors. Given the scarcity of 3D geometry data from which to learn these meaningful descriptors, we leverage foundational vision models trained on very large datasets to obtain these features. This enables DIFF3F to produce semantic descriptors in a zero-shot way. Our code can be found here.

3.1. Semantic Diffusion Features

Given a shape S with vertices $V \in \mathbb{R}^3$, we want to project it to the image space to distill per-point semantic 3D features from images. We define an image projector P as

$$P(\cdot|C_j) := S \mapsto I_j^S \in \mathbb{R}^{H \times W}, \tag{1}$$

where H, W denote the height and width of the image rendered by P, with C_j representing the j^{th} camera producing

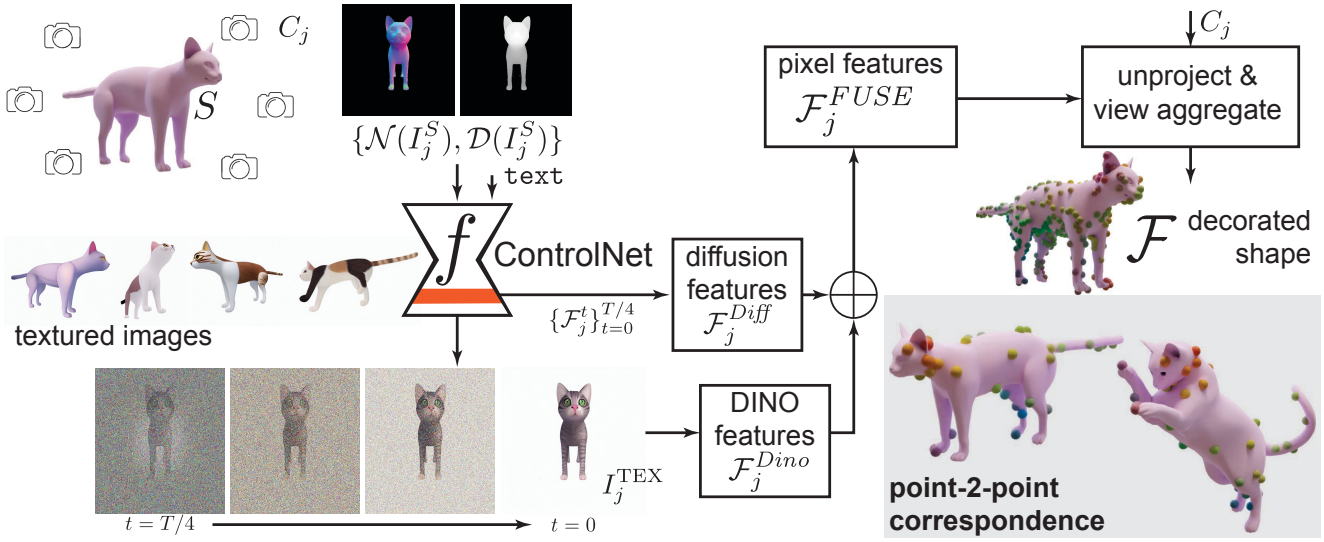

Figure 2. **Method overview.** DIFF3F is a feature distiller to map semantic diffusion features to 3D surface points. We render the given shape without textures from multiple views, and the resulting renderings are in-painted by guiding ControlNet with geometric conditions; the generative features from ControlNet are fused with features obtained from the textured rendering, followed by unprojecting to the 3D surface. Note that the textured images obtained by conditioning ControlNet from different views can be inconsistent.

the image I_j^S. P can be a renderer with shading or simply a rasterizer that returns the depth from the camera.

As an emergent behaviour, pre-trained foundational vision models have been found to assign distinctive semantic features [54] to pixels in the input image, to be able to distinguish between nearby pixels to perform core tasks like object detection or segmentation. Our core idea to get such features is, therefore, to drive a pretrained foundational model to perform a challenging task that requires generating semantic per-pixel features during the process, so that we can extract these features into 3D. Since we aim for per-point features, instead of regional descriptors, we add textures to rendered images conditioned on text prompts. Creating a realistic textured image from an untextured image requires the model to distinguish between nearby pixels in their semantic sense such that the visual result satisfies the text prompt. For example, a shading model may color a drawn cube completely gray. Still, when conditioned on the text "iron box", it would be driven to add specific characteristics, such as metallic textures that clearly allow it to be identified as "iron".

Given a point cloud or raw mesh, projection P produces an untextured image with silhouette or shading, respectively. We drive a diffusion model [47] f to color textureless I_j^S (from camera view C_j) realistically and take it to the RGB space:

$$f(\cdot|G, \texttt{text}) := I_j^S \in \mathbb{R}^{H\times W} \mapsto I_j^{TEX} \in \mathbb{R}^{H\times W\times 3}, \quad (2)$$

where G is a set of functions describing geometric constraints that guide the texturing model and `text` defines the text prompt defining the subjects. We guide the texturing by providing constraints G to ControlNet [66]. In effect, f projects shape S to an RGB image based on camera C_j.

3.2. Semantics through Painting

Realistically texturing a silhouette image is an open problem and a challenging task. Given untextured images $\{I_j^S\}$, a naive approach of assigning a constant color does not require inferring the semantics of the given geometry. We, therefore, condition our painting module f with geometric constraints that describe the latent 3D object.

We define G as a set of geometric maps that can be applied as conditional image constraints,

$$G := \{\mathcal{N}(I_j^S), \mathcal{D}(I_j^S)\}, \quad (3)$$

where $\mathcal{N}$ is a normal map and $\mathcal{D}$ is a continuous depth map from the camera describing the input shape. When combined with a text prompt, we expand Equation 3 as

$$f(\cdot|\mathcal{N}(I_j^S), \mathcal{D}(I_j^S), \texttt{text}) := I_j^S \mapsto I_j^{TEX}. \quad (4)$$

During this texturing forward pass, we extract features $\mathcal{F}_L^t$ from an intermediate layer L of Stable Diffusion's UNet decoder at diffusion time step t with $t \in [0, T]$. We use DDIM [51] to accelerate the sampling process for Stable Diffusion [47] and use 30 inference steps. For notational simplicity, in the following, we drop the layer index L, i.e., we use $\mathcal{F}_j^t$ to indicate $\mathcal{F}_{L,j}^t$. We thus directly get our feature renderer as,

$$f(\cdot|\mathcal{N}(I_j^S), \mathcal{D}(I_j^S), \texttt{text}, L, t) := I_j^S \mapsto \mathcal{F}_j^t, \quad (5)$$

These features are normalized to have unit norm:

$$\mathcal{F}_j^t := \mathcal{F}_j^t / \|\mathcal{F}_j^t\|_2. \quad (6)$$

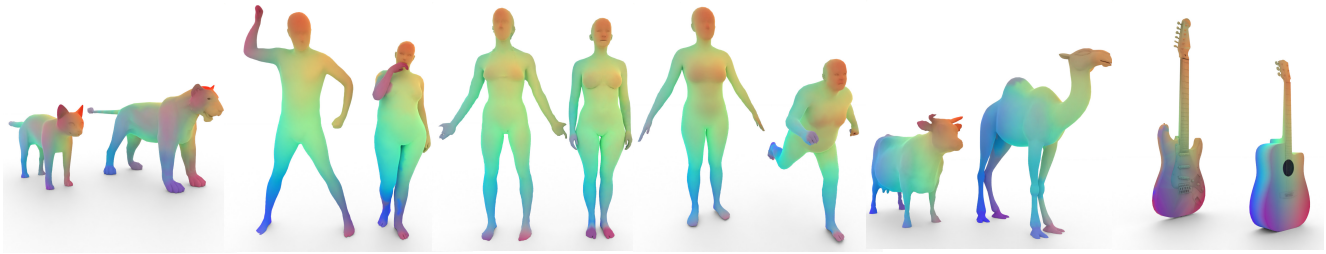

Figure 3. **Results gallery.** DIFF3F's performance on various point correspondence challenges. Corresponding points are similarly colored. Note that DIFF3F can successfully distinguish between symmetric parts and remains fairly robust under pose and shape variations. For each shape pair, the source is on the left and the target is on the right.

We aggregate the normalized features using a weighted approach starting from T/4 to the final denoising step 0. The weights w_t are linearly spaced from 0.1 to 1 depending on the number of time steps (t) to assign higher weights to the embeddings from less noisy images (i.e., higher t). Each pixel gets a 1280 dimensional feature from the diffusion UNet, aggregated over diffusion time steps. Specifically,

$$\mathcal{F}_j^{Diff} := \sum_{t=0}^{T/4} w_t \cdot \mathcal{F}_j^t \in \mathbb{R}^{H \times W \times 1280}. \tag{7}$$

We further fuse the diffusion features $\mathcal{F}_j$ with DINOv2 [40] features $\mathcal{F}_j^{Dino}$ extracted from the textured renderings I_j^{TEX}. We also normalize these image features as in Equation 6. It has been noted that DINO features contain strong complementary semantic signals but are weaker regarding spatial understanding [65]. Hence, combining the two features gives stronger semantic descriptors while retaining spatial information. We employ a feature fusion strategy proposed by [65], where we first normalize the features and then concatenate them as,

$$\mathcal{F}_j^{FUSE} := (\alpha \mathcal{F}_j^{Diff}, (1-\alpha)\mathcal{F}_j^{Dino}) \tag{8}$$

where α is a tunable parameter; we use $\alpha = 0.5$ in all our experiments. $\mathcal{F}_j^{FUSE}$ is also unit-normalized as in Equation 6.

3.3. Distilling 2D Features to 3D

We leverage known camera parameters to unproject features from the image space back to the points on the 3D input, i.e., $\mathcal{F}_j^{FUSE} \xrightarrow{P^{-1}} \mathcal{F}_j^{3D}$, where P is the projection function from equation 1. We also employ a ball query $B_r(x)$, introduced in [42], to facilitate feature sharing within local neighborhoods of radius r around any surface point $x \in S$ and promote local consensus. We use $r = 1\%$ of the object's bounding box diagonal length. This is particularly useful for shape correspondence where points in a local neighborhood of the source should match with points in a local neighborhood of the target.

To aggregate features from multiple views per point, we compute the mean of the normalized feature vectors. We also experimented with max pooling, but the results were inferior. Our rendering setup and unprojection with ball querying make the per-point accumulated features coherent, enabling a simple aggregation. Moreover, we render the 3D shape from several views ($n = 100$), which further stabilizes the aggregation, resulting in descriptors that mostly capture semantic meaning:

$$\mathcal{F} := \frac{1}{n} \sum_{j=1}^{n} \mathcal{F}_j^{3D}. \tag{9}$$

The above step aggregates descriptors per vertex across n views, spread uniformly around a sphere around the object, to compute our final semantic 3D point descriptors. Next, we describe how to use these descriptors to compute correspondences between pairs of shapes.

3.4. Computing Correspondence

Given a source point cloud $\mathcal{S}$ and a target point cloud $\mathcal{T}$, we want to find a mapping $m : \mathcal{S} \mapsto \mathcal{T}$, such that we compute a corresponding point $\mathcal{T}_k \in \mathcal{T}$ for each point $\mathcal{S}_i \in \mathcal{S}$ where $1 \leq i, k \leq N$.

Point-to-Point: To find correspondence between points of two shapes, we compute their point descriptors independently and match them by cosine similarity in the shared feature space. We calculate similarity as the cosine of the angle between the source ($\mathcal{S}$) and target ($\mathcal{T}$) feature vectors, $\mathcal{F}_\mathcal{S}$ and $\mathcal{F}_\mathcal{T}$, respectively. Specifically,

$$s_{ik} := \frac{\langle \mathcal{F}_{\mathcal{S}_i}, \mathcal{F}_{\mathcal{T}_k} \rangle}{\|\mathcal{F}_{\mathcal{S}_i}\|_2 \|\mathcal{F}_{\mathcal{T}_k}\|_2} \tag{10}$$

where $\mathcal{F}_\mathcal{S}^i$ and $\mathcal{F}_\mathcal{T}^j$ are the i^{th} and k^{th} rows of $\mathcal{F}_\mathcal{S}$ and $\mathcal{F}_\mathcal{T}$ respectively, $\langle \cdot \rangle$ denotes the dot product operation. We choose a corresponding point $\mathcal{T}_k$ in $\mathcal{T}$ for every point $\mathcal{S}_i$ in $\mathcal{S}$ where s_{ik} is highest. Note that, in order to assess the quality of our decorated features, we assign correspondence based on the highest per-vertex similarity and do not regularize the solution with any other energy terms.

Surface-to-Surface: While point-to-point correspondence is important for certain applications like non-rigid registration,

Table 2. **Comparison.** We report correspondence accuracy within 1% error tolerance, with our method against competing works. The Laplace Beltrami Operator (LBO) computation for Functional Maps is unstable on TOSCA since the inputs contain non-manifold meshes. By '*' we denote results reported by SE-ORNet [14].

Method ↦	DPC [30]		SE-ORNet [14]		3DCODED [20]		FM [41]+WKS [6]		DIFF3F (ours)		DIFF3F (ours)+FM [41]	
↓ **Dataset**	$acc \uparrow$	$err \downarrow$	$acc \uparrow$	$err \downarrow$	$acc \uparrow$	$err \downarrow$	$acc \uparrow$	$err \downarrow$	$acc \uparrow$	$err \downarrow$	$acc \uparrow$	$err \downarrow$
TOSCA	*30.79*	**3.74**	**33.25**	*4.32*	0.5*	19.2*	✗	✗	20.27	5.69	✗	✗
SHREC'19	17.40	6.26	21.41	4.56	2.10	8.10	4.37	3.26	**26.41**	*1.69*	*21.55*	**1.49**
SHREC'20	31.08	2.13	31.70	1.00	✗	✗	4.13	7.29	**72.60**	*0.93*	*62.34*	**0.71**

we note that it might also be desirable in certain cases to compute a continuous surface-to-surface map, rather than matching discrete points. To enable this, we pass our computed descriptors to a vanilla Functional Map [41] implementation, which returns a continuous surface-to-surface map that can then be used for direct correspondences.

4. Evaluation

4.1. Datasets and Benchmarks

We evaluate our method on datasets involving both human and animal subjects to showcase the efficacy and applicability of our approach. To make a fair comparison with existing works, we follow a similar experiment setup described in DPC [30] and SE-ORNet [14].

Human shapes: We test our method on SHREC'19 [38] comprising of 44 actual human scans, which are organized into 430 annotated test pairs with considerable variation. We choose the more challenging remeshed version from [16]. We also evaluate our method on the FAUST benchmark [8].

Animal shapes: For testing our method on animal shapes, we evaluate our method on the SHREC'20 [17] and TOSCA [9] datasets. SHREC'20 contains various animals in different poses with non-isometric correspondence annotated by experts. We compute correspondence on the annotated correspondences per pair (approximately 50). TOSCA comprises of 80 objects representing a mixture of animals and humans, formed by deforming template meshes. We ignore the human figures and select all the 41 animal figures within the test set and pair shapes from the same category to create a testing set of 286 paired samples.

Table 3. **Generalization.** We compare generalization capabilities of DIFF3F vs others by training on one dataset and testing on a different set. For DPC and SE-ORNet, we choose SURREAL and SMAL as the training sets for human and animal shapes, respectively – these larger datasets lead to improved generalization scores. By '*' we denote results reported by SE-ORNet [14].

Train	**Method**	TOSCA		SHREC'19		SHREC'20	
		$acc \uparrow$	$err \downarrow$	$acc \uparrow$	$err \downarrow$	$acc \uparrow$	$err \downarrow$
SURREAL	DPC [30]	29.30	*5.25*	17.40	6.26	31.08	2.13
	SE-ORNET [14]	16.71	9.19	*21.41*	*4.56*	*31.70*	*1.00*
SMAL	DPC [30]	*30.28*	6.43	12.34	8.01	24.5*	7.5*
	SE-ORNET [14]	**31.59**	**4.76**	12.49	9.87	25.4*	2.9*
Pretrained	DIFF3F (ours)	20.27	5.69	**26.41**	**1.69**	**72.60**	**0.93**

4.2. Evaluation Metrics

We use the average correspondence error and the correspondence accuracy as our evaluation criteria. Since our method operates in the domain of point clouds, we use a Euclidean-based measure, as used in previous works [14, 30, 64].

The average correspondence error for a pair of shapes, source $\mathcal{S}$ and target $\mathcal{T}$ is defined as:

$$err = \frac{1}{n} \sum_{\mathcal{S}_i \in \mathcal{S}} \|f(\mathcal{S}_i) - t_{gt}\|_2 \tag{11}$$

where $f(\mathcal{S}_i)$ is the computed correspondence for $\mathcal{S}_i \in \mathcal{S}$ in $\mathcal{T}$ and $t_{gt} \in \mathcal{T}$ is the ground truth correspondence for a set of n samples.

The correspondence accuracy is measured as the fraction of correct correspondences within a threshold tolerance distance:

$$acc(\epsilon) = \frac{1}{n} \sum_{\mathcal{S}_i \in \mathcal{S}} \mathbb{I}(\|f(\mathcal{S}_i) - t_{gt}\|_2 < \gamma d) \tag{12}$$

where $\mathbb{I}(\cdot)$ is the indicator function , $\gamma \in [0, 1]$ is the error tolerance, and d is the maximal Euclidean distance between points in T.

4.3. Baseline Methods

We present our results on input pairs of meshes and pairs of pointclouds. We compare our method to recent state-of-the-art methods in shape correspondence namely DPC [30], SE-ORNet [14], and 3D-CODED [20]. While 3D-CODED requires an extensively annotated dataset for training, DPC and SE-ORNet are unsupervised methods and have been trained on human datasets- SURREAL [21] and SHREC'19 [38], as well as animal datasets- SMAL [67] and TOSCA [9]. Note that we do not have access to pretrained 3D-CODED models for animal models. In comparison, our method requires no training, enabling zero-shot feature extraction. Our method computes semantic descriptors, therefore, for a complete comparison, we also evaluate against the Wave Kernel Signature (WKS) [7] geometric descriptors combined with Functional Maps [41].

4.4. Evaluation on Human Shapes

We present results on the SHREC'19 dataset. Table 2 shows correspondence accuracy at 1% error tolerance which represents a near-perfect hit. Our method achieves a state-of-the-art correspondence accuracy of 26.41% at 1% error tolerance, an improvement of 5%.

Many tasks, including alignment and texture transfer, require a certain number of precise correspondences rather than average correspondence quality to work well. We choose baseline methods trained on SURREAL as it is a significantly larger dataset (consisting of human shapes) than SHREC'19, leading to improved performance. Our method achieves the highest correspondence accuracy compared to existing works and the lowest average correspondence error compared to baseline methods, as seen in Table 2. We show qualitative results for comparison in Figure 5 using our method with point cloud rendering. While DPC and SE-ORNet both get confused by the different alignments of the human pair resulting in a flipped prediction, ours, being a multi-view rendering-based method, it is robust to rotation. Hence, it can reliably solve the correspondence. We show additional visual results from a human source to multiple targets spanning modality, class and pose in Figure 1. Most

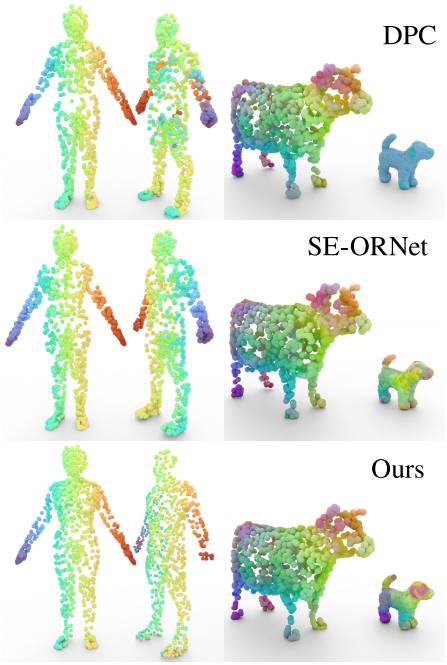

Figure 4. **Comparisons.** We compare our DIFF3F (bottom) against SOTA methods (i.e., DPC [30] and SE-ORNet [14]) for the task of point-to-point shape correspondence. Corresponding points, computed as described in Section 3.4, are similarly colored. We show results using point cloud rendering of our method for the human pair (left) and results with mesh rendering for the animal pair (right). Table 2 shows qualitative evaluation on benchmarks.

correspondences including highly non-isometric deformations are accurate but we see misaligned correspondence for the human to alien pair as the legs get flipped. This is because the front and back sides are less clear as the mesh has no dominant front feature. Moreover, we do not perform any processing on the raw meshes and use random coordinate system. Please refer to the supplemental for similarity heatmaps for selected points on examples.

Evaluation on FAUST scans. We further evaluate DIFF3F on the FAUST [8] intra-subject challenge, which consists of high-resolution human scans (100k+ vertices). DIFF3F achieves an average geodesic error of 5.29cm. Error profiles and visual results can be found on the FAUST website.

4.5. Evaluation on Animal Shapes

We evaluate baseline methods trained on TOSCA and SMAL, and select the best performing configuration- for DPC and SE-ORNet, we choose TOSCA, whereas for 3D-CODED we choose SMAL. Our method achieves comparable accuracy and error to the baseline methods on the TOSCA dataset, as seen in Table 2, and generalizes better than baseline methods trained on human shapes, as seen in Table 3. Results using 3D-CODED are particularly poor on TOSCA mainly for two reasons: (i) It needs a much larger dataset with ground truth annotations, which is not available for animal shapes; and (ii) it computes correspondence through template deformation, which fails on TOSCA due to the varied shapes and poses of different animals. TOSCA consists of highly isomteric shapes, therefore we also evaluate our method on SHREC'20, which consists of highly non-isometric pairs of animals. We outperform baseline methods by a large margin for non-isometric shapes thanks to the semantic nature of DIFF3F. While the evaluation is on a subset of only about 50 points as the number of annotated points is very limited, we show dense correspondence in Figure 5. The visual results showing dense correspondence for non-isometric pairs highlight the efficacy of our semantic descriptors compared to competing methods. DPC can get confused by the radical change in structure, while SE-ORNet has largely misaligned correspondences. We present additional visual results of animal pairs and a guitar pair in Figure 3.

4.6. Ablations

We ablate different components of our method and report their performance. Table 4 shows our findings. We find that adding realistic texture, as opposed to only shading, results in a significant improvement in terms of accuracy and reducing errors. We also explore a baseline method using DINO features on consistent textures obtained using TEXTure [46]. Ours is better, particularly at an *accuracy* of 1% *err* tolerance, as diffusion features capture more geometric information than DINO. Additionally, varied textured renderings enable a more robust feature aggregation due to the implicit denoising of unnecessary feature dimensions

Table 4. **Ablation.** We ablate different components of our method and compare accuracy at 1% tolerance on SHREC'19 and SHREC'20, against our full method (last row).

Ablation	SHREC'19		SHREC'20	
	acc ↑	*err* ↓	*acc* ↑	*err* ↓
w/o ControlNet (untextured)	17.20	2.04	65.48	**0.69**
TEXTure[46]+DINO	17.20	2.04	65.48	**0.69**
w/o Fusion with DINO	**26.53**	2.06	64.89	1.60
w/o Normal Maps	25.68	**1.67**	69.71	1.17
w/o Time Aggregation	25.73	1.71	68.95	*0.87*
w/o Ball query	25.72	1.73	**74.10**	0.99
DIFF3F (full method)	*26.41*	*1.69*	*72.60*	0.93

such as color. We note that TEXTure yielded poor results for humans. As meshes are not aligned and rely on iterative inpainting, if the first texture is poor, subsequent textures are poor, too. In contrast, ours aggregates over multiple views. Although our complete approach produces the second-best score in every category, incorporating all of our parts together (including fusion with DINO) resulted in the best overall balance of high accuracy and low average error.

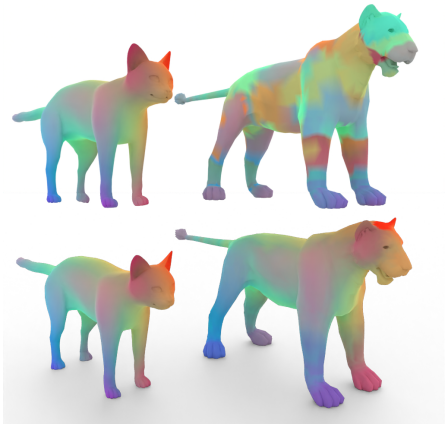

Figure 5. **Regularizing point-2-point maps.** We compare the effectiveness of vanilla functional maps with the Wave Kernel Signature as descriptors (top) vs our descriptors DIFF3F (bottom). Ours being semantic enables Functional Maps to work with non-isometric deformations even though FMs typically struggle with such cases when using traditional geometric descriptors. Our descriptors yield accurate correspondence in most cases, thus eliminating the need for further refinement algorithms typically used in related works.

5. Part Segmentation

We directly apply k-means clustering to our features to extract part segments. Interestingly, we discover that the k-means centroids, extracted from one shape (e.g., human), can be used to segment another (e.g., cat), thanks to the semantic nature of our descriptors. This leads to corresponding part segmentation (arms of the human map to the front legs of the cat, head maps to head, etc.) as seen in Figure 6. One possible method to automatically identify the number of segments (k) is to query an LLM, as explored in [1].

Figure 6. **Segmentation.** We apply k-means on DIFF3F features with $k = 6$ on human and elephant; and $k = 3$ on chair and dolphin. The cat is segmented using k-means centroids of the human leading to corresponding part segmentation (arms map to front legs, etc.).

6. Conclusion

We introduced DIFF3F as a robust semantic descriptor for textureless input shapes like meshes or point clouds. Distilled through image diffusion models, these descriptors, without the need for extra data or training, complement existing geometric features and generalize well across diverse inputs. Our thorough evaluation on benchmark datasets, including isometric and non-isometric shapes, positions DIFF3F as a state-of-the-art performer. We outperform recent learning-based methods on multiple datasets, demonstrating superior performance even on shapes beyond the training sets.

Limitations. Since our method relies on multi-view images, DIFF3F fails to produce features on parts of the shapes that are invisible from all the sampled views (self-occlusion). Further, since we aggregate (diffusion) features from image diffusion models, we inherit their limitations in terms of suffering from bias in the dataset and/or view bias for objects. For example, the features aggregated on a horse model are worse in less seen regions, like the underneath of its belly.

Future work. The next step involves combining semantic features with geometric ones, aiming for enhanced performance. Addressing potential noise in less visible areas of the distilled features is crucial, and we plan to explore the impact of refining features through geometric smoothness energies, such as local conformality or isometry. Overcoming challenges related to point clouds and non-manifold meshes is essential, given that many traditional geometry processing approaches assume manifold meshes. Additionally, there is an intriguing prospect of extending these descriptors to accommodate volumetric inputs like NeRFs or distance fields.

Acknowledgments. We thank Romy Williamson for her comments and suggestions. NM was supported by Marie Skłodowska-Curie grant agreement No. 956585, gifts from Adobe, and UCL AI Centre.

References

[1] Ahmed Abdelreheem, Abdelrahman Eldesokey, Maks Ovsjanikov, and Peter Wonka. Zero-shot 3d shape correspondence. *arXiv preprint arXiv:2306.03253*, 2023. 3, 8

[2] Noam Aigerman, Roi Poranne, and Yaron Lipman. Lifted bijections for low distortion surface mappings. *ACM Transactions on Graphics (TOG)*, 33(4):1–12, 2014. 3

[3] Noam Aigerman, Roi Poranne, and Yaron Lipman. Seamless surface mappings. *ACM Transactions on Graphics (TOG)*, 34 (4):1–13, 2015. 3

[4] Noam Aigerman, Kunal Gupta, Vladimir G Kim, Siddhartha Chaudhuri, Jun Saito, and Thibault Groueix. Neural jacobian fields: Learning intrinsic mappings of arbitrary meshes. *SIGGRAPH*, 2022. 3

[5] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *ECCVW What is Motion For?*, 2022. 2

[6] Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *2011 IEEE international conference on computer vision workshops (ICCV workshops)*, pages 1626–1633. IEEE, 2011. 3, 6

[7] Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 1626–1633, 2011. 6

[8] Federica Bogo, Javier Romero, Matthew Loper, and Michael J Black. Faust: Dataset and evaluation for 3d mesh registration. In *Proc. CVPR*, pages 3794–3801, 2014. 6, 7

[9] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. *Numerical geometry of non-rigid shapes*. Springer Science & Business Media, 2008. 2, 6

[10] Dongliang Cao and Florian Bernard. Self-supervised learning for multimodal non-rigid 3d shape matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17735–17744, 2023. 2

[11] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 2

[12] Dave Zhenyu Chen, Yawar Siddiqui, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. Text2tex: Text-driven texture synthesis via diffusion models. *arXiv preprint arXiv:2303.11396*, 2023. 2

[13] Dale Decatur, Itai Lang, and Rana Hanocka. 3d highlighter: Localizing regions on 3d shapes via text descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20930–20939, 2023. 3

[14] Jiacheng Deng, Chuxin Wang, Jiahao Lu, Jianfeng He, Tianzhu Zhang, Jiyang Yu, and Zhe Zhang. Se-ornet: Self-ensembling orientation-aware network for unsupervised point cloud shape correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5364–5373, 2023. 2, 3, 6, 7

[15] Theo Deprelle, Thibault Groueix, Matthew Fisher, Vladimir Kim, Bryan Russell, and Mathieu Aubry. Learning elementary structures for 3d shape generation and matching. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[16] Nicolas Donati, Abhishek Sharma, and Maks Ovsjanikov. Deep geometric functional maps: Robust feature learning for shape correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 3, 6

[17] Roberto M Dyke, Yu-Kun Lai, Paul L Rosin, Stefano Zappalà, Seana Dykes, Daoliang Guo, Kun Li, Riccardo Marin, Simone Melzi, and Jingyu Yang. Shrec'20: Shape correspondence with non-isometric deformations. *Computers & Graphics*, 92:28–43, 2020. 6

[18] Marvin Eisenberger, Zorah Lähner, and Daniel Cremers. Divergence-free shape correspondence by deformation. In *Computer Graphics Forum*, pages 1–12. Wiley Online Library, 2019. 2

[19] Michael Fischer, Zhengqin Li, Thu Nguyen-Phuoc, Aljaz Bozic, Zhao Dong, Carl Marshall, and Tobias Ritschel. Nerf analogies: Example-based visual attribute transfer for nerfs. *arXiv preprint arXiv:2402.08622*, 2024. 3

[20] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. 3d-coded: 3d correspondences by deep deformation. In *Proceedings of the european conference on computer vision (ECCV)*, pages 230–246, 2018. 2, 6

[21] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. 3d-coded: 3d correspondences by deep deformation. In *Proceedings of the european conference on computer vision (ECCV)*, pages 230–246, 2018. 6

[22] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, pages 1735–1742. IEEE, 2006. 3

[23] Haibin Huang, Evangelos Kalogerakis, Siddhartha Chaudhuri, Duygu Ceylan, Vladimir G. Kim, and Ersin Yumer. Learning local shape descriptors from part correspondences with multiview convolutional networks. *ACM Trans. Graph.*, 37 (1), 2017. 3

[24] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):1–13, 2013. 2

[25] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 3

[26] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. *Advances in Neural Information Processing Systems*, 35:23311–23330, 2022. 3

[27] Artem Komarichev, Zichun Zhong, and Jing Hua. A-cnn: Annularly convolutional neural networks on point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7421–7430, 2019. 3

[28] Vladislav Kraevoy and Alla Sheffer. Cross-parameterization and compatible remeshing of 3d models. *ACM Transactions on Graphics (ToG)*, 23(3):861–869, 2004. 3

[29] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 3

[30] Itai Lang, Dvir Ginzburg, Shai Avidan, and Dan Raviv. Dpc: Unsupervised deep point correspondence via cross and self construction. In *2021 International Conference on 3D Vision (3DV)*, pages 1442–1451. IEEE, 2021. 2, 3, 6, 7

[31] Aaron WF Lee, David Dobkin, Wim Sweldens, and Peter Schröder. Multiresolution mesh morphing. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 343–350, 1999. 3

[32] Bruno Lévy, Sylvain Petitjean, Nicolas Ray, and Jérome Maillot. Least squares conformal maps for automatic texture atlas generation. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 193–202. 2023. 3

[33] Jiahao Lin and Gim Hee Lee. Multi-view multi-person 3d pose estimation with plane sweep stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11886–11895, 2021. 3

[34] Yaron Lipman. Bounded distortion mapping spaces for triangular meshes. *ACM Transactions on Graphics (TOG)*, 31(4): 1–13, 2012. 3

[35] Or Litany, Tal Remez, Emanuele Rodola, Alex Bronstein, and Michael Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. In *Proceedings of the IEEE international conference on computer vision*, pages 5659–5667, 2017. 2, 3

[36] Grace Luo, Lisa Dunlap, Dong Huk Park, Aleksander Holynski, and Trevor Darrell. Diffusion hyperfeatures: Searching through time and space for semantic correspondence. *arXiv preprint arXiv:2305.14334*, 2023. 2

[37] Manish Mandad, David Cohen-Steiner, Leif Kobbelt, Pierre Alliez, and Mathieu Desbrun. Variance-minimizing transport plans for inter-surface mapping. *ACM Transactions on Graphics (ToG)*, 36(4):1–14, 2017. 3

[38] Simone Melzi, Riccardo Marin, Emanuele Rodolà, Umberto Castellani, Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. Shrec 2019: Matching humans with different connectivity. In *Eurographics Workshop on 3D Object Retrieval*, page 3. The Eurographics Association, 2019. 6

[39] Luca Morreale, Noam Aigerman, Vladimir G Kim, and Niloy J Mitra. Neural surface maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4639–4648, 2021. 3

[40] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 2, 5

[41] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (ToG)*, 31(4):1–11, 2012. 2, 3, 6

[42] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 5

[43] Yvain Quéau, Jean Mélou, Jean-Denis Durou, and Daniel Cremers. Dense multi-view 3d-reconstruction without dense correspondences. 2017. 3

[44] Michael Rabinovich, Roi Poranne, Daniele Panozzo, and Olga Sorkine-Hornung. Scalable locally injective mappings. *ACM Transactions on Graphics (TOG)*, 36(4):1, 2017. 3

[45] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2, 3

[46] Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3d shapes. *arXiv preprint arXiv:2302.01721*, 2023. 2, 7, 8

[47] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2, 4

[48] Jean-Michel Roufosse, Abhishek Sharma, and Maks Ovsjanikov. Unsupervised deep learning for structured shape matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1617–1627, 2019. 2, 3

[49] John Schreiner, Arul Asirvatham, Emil Praun, and Hugues Hoppe. Inter-surface mapping. In *ACM SIGGRAPH 2004 Papers*, pages 870–877. 2004. 3

[50] Gopal Sharma, Kangxue Yin, Subhransu Maji, Evangelos Kalogerakis, Or Litany, and Sanja Fidler. Mvdecor: Multi-view dense correspondence learning for fine-grained 3d segmentation. In *European Conference on Computer Vision*, pages 550–567. Springer, 2022. 3

[51] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 4

[52] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, pages 109–116. Citeseer, 2007. 3

[53] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015. 3

[54] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. *arXiv preprint arXiv:2306.03881*, 2023. 2, 4

[55] Federico Tombari, Samuele Salti, and Luigi Di Stefano. 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*, pages 356–369. Springer, 2010. 2

[56] Bach Tran, Binh-Son Hua, Anh Tuan Tran, and Minh Hoai. Self-supervised learning with multi-view rendering for 3d point cloud analysis. In *Proceedings of the Asian Conference on Computer Vision*, pages 3086–3103, 2022. 3

[57] Wenju Wang, Yu Cai, and Tao Wang. Multi-view dual attention network for 3d object recognition. *Neural Computing and Applications*, 34(4):3201–3212, 2022. 3

[58] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3523,3526,3527, 2019. 2

[59] Yunhai Wang, Minglun Gong, Tianhua Wang, Daniel Cohen-Or, Hao Zhang, and Baoquan Chen. Projective analysis for 3d shape segmentation. *ACM TOG*, 32(6), 2013. 3

[60] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5):1–12, 2019. 2

[61] Yong Xu, Chaoda Zheng, Ruotao Xu, Yuhui Quan, and Haibin Ling. Multi-view 3d shape recognition via correspondence-aware deep learning. *IEEE Transactions on Image Processing*, 30:5299–5312, 2021. 3

[62] Zi Jian Yew and Gim Hee Lee. Rpm-net: Robust point matching using learned features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11824,11827, 2020. 2

[63] Qian Yu, Chengzhuan Yang, Honghui Fan, and Hui Wei. Latent-mvcnn: 3d shape recognition using multiple views from pre-defined or random viewpoints. *Neural Processing Letters*, 52:581–602, 2020. 3

[64] Yiming Zeng, Yue Qian, Zhiyu Zhu, Junhui Hou, Hui Yuan, and Ying He. Corrnet3d: Unsupervised end-to-end learning of dense correspondence for 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6052–6061, 2021. 2, 6

[65] Junyi Zhang, Charles Herrmann, Junhwa Hur, Luisa Polania Cabrera, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang. A tale of two features: Stable diffusion complements dino for zero-shot semantic correspondence. *arXiv preprint arXiv:2305.15347*, 2023. 2, 5

[66] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023. 2, 3, 4

[67] Silvia Zuffi, Angjoo Kanazawa, David W Jacobs, and Michael J Black. 3d menagerie: Modeling the 3d shape and pose of animals. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6365–6373, 2017. 6