Back to 3D: Few-Shot 3D Keypoint Detection with Back-Projected 2D Features

Thomas Wimmer[1,2] Peter Wonka[3] Maks Ovsjanikov[1]
[1]LIX, École Polytechnique [2]Technical University of Munich [3]KAUST
thomas.m.wimmer@tum.de, pwonka@gmail.com, maks@lix.polytechnique.fr

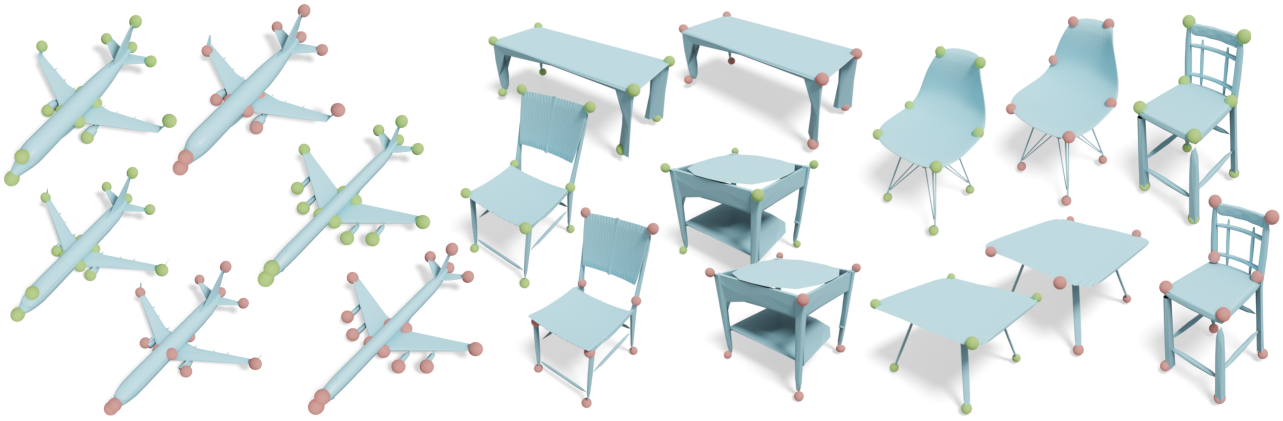


Figure 1. Qualitative results of our proposed method B2-3D for few-shot keypoint detection using back-projected features (red) with ground truth keypoint annotations (green).

Abstract

With the immense growth of dataset sizes and computing resources in recent years, so-called foundation models have become popular in NLP and vision tasks. In this work, we propose to explore foundation models for the task of keypoint detection on 3D shapes. A unique characteristic of keypoint detection is that it requires semantic and geometric awareness while demanding high localization accuracy. To address this problem, we propose, first, to back-project features from large pre-trained 2D vision models onto 3D shapes and employ them for this task. We show that we obtain robust 3D features that contain rich semantic information and analyze multiple candidate features stemming from different 2D foundation models. Second, we employ a keypoint candidate optimization module which aims to match the average observed distribution of keypoints on the shape and is guided by the back-projected features. The resulting approach achieves a new state of the art for few-shot keypoint detection on the KeyPointNet dataset, almost doubling the performance of the previous best methods.

1. Introduction

Foundation models are finding their way into an increasing number of downstream applications. They show strong generalization to tasks they were not explicitly trained for and exhibit surprising zero- or few-shot capabilities. Successful approaches have been presented for processing text or 2D images [8, 28, 33], but there are no prominent 3D methods yet available that are aimed at *local* details. The main reasons for this are the diversity of 3D representations (i.e., meshes, point clouds, volumetric or implicit representations) and the comparably low availability of high-quality 3D data.

Recent works using 2D foundation models for shape analysis, like [19, 25, 49], have focused mainly on global tasks like shape classification, or segmentation which represents a middle ground between global and local analyses [2]. This paper builds on this continuum, advancing from classification to segmentation and now to keypoints, each stage requiring a more localized understanding of geometric details.

In this work, we focus on the task of few-shot keypoint detection on 3D meshes. Repeatable keypoints enable a wide range of downstream applications, including rigid and non-rigid shape matching, object tracking, shape reconstruction, and shape manipulation [9, 12, 21, 29, 43] to name a few. Furthermore, the complex nature of the keypoint detection problem, which requires both semantic and geometric shape understanding, has been used in the past as a fruitful testbed for both exploiting and evaluating var-

ious local shape descriptors [7, 38] as well as consistency relations between 3D shapes and their views [45].

A natural approach for keypoint detection is to leverage dense 3D geometric features or descriptors. Ideally, such features should capture both global (semantic) and local (geometric) information at the same time. Previous works in this domain either focus on axiomatic features or propose to pre-train models on 3D datasets [3, 5]. Such methods, however, are limited by both robustness issues and the lack of diversity and amount of 3D training data. As a result, existing approaches tend to have limited accuracy on complex 3D models. In contrast, we propose to retrieve informative features from powerful *pre-trained* 2D vision encoders that can be used directly without any further training.

We show, for the first time, that such back-projected features enable a localized understanding of shape geometry while at the same time being able to capture rich global semantic information. Specifically, we demonstrate that these features are robust against rotations and scaling changes and can be computed for any point on the shape's surface and with any texture type. Through assembling information from multiple rendered views around the shape, we show that even coarse 2D features can lead to dense 3D descriptors that vary continuously on the surface.

In addition to the lack of informative features, a common challenge in keypoint detection is the presence of symmetries. A successful method should detect *all* keypoints across various symmetries (e.g., different legs of a table) while avoiding superfluous detections. To address this issue, we introduce a simple yet effective optimization strategy that leverages the back-projected features and ensures that the detected keypoints have a similar distribution on the target shapes as on the given few-shot examples.

Our main contributions are (1) the formalization of feature back-projection, including Gaussian geodesic reweighting for handling noisy point visibility information, (2) the analysis of back-projected features from different recent foundation models and their properties, and (3) an optimization module which aims to match the average observed distribution of keypoints on the target shape. Our resulting method achieves state-of-the-art results on the KeypointNet benchmark, improving over the best-competing method by over **93%** IoU on average over all evaluation distances. We compare different 2D feature extractors as well as axiomatic 3D descriptors and evaluate further components of our optimization module in an extensive ablation study. As an additional validation, we demonstrate the strong generalization of the features to other tasks and achieve state-of-the-art results in the task of part segmentation transfer.

2. Related Work

The flexibility of the transformer architecture and the increase in data and computing resources have led to the creation of powerful foundation models. This section gives an overview of relevant models and their transfer for the analysis of 3D data, as well as an overview of classic shape descriptors and previous methods for keypoint detection on 3D shapes.

Foundation Models Large pre-trained models that were trained on vast quantities of data and exhibit strong generalization to new tasks with no or only little fine-tuning are referred to as foundation models. Prominent examples include large language models that exhibit strong zero-shot generalization through prompting. The success in the textual domain, as well as the high availability of image data on the internet, has led to recent advances in vision and multi-modal models. Models like DINO [10, 28] or masked autoencoders [18] are, in their essence, just feature extractors for image data as any other neural network. However, their self-supervised training on large datasets has led to a great semantic understanding of scenes that can be utilized using simple linear or nearest-neighbor-based models to solve downstream tasks. Multi-modal foundation models like the vision-language model (VLM) CLIP [33] consist of separate encoder models that aim to map data from different modalities, like text and images, to the same meaningful embedding space. By comparing the computed embeddings of candidate text prompts with an image embedding, one can thus perform, e.g., open-vocabulary zero-shot classification of images. Finally, Kirillov et al. [22] proposed SAM, a foundation model specifically aimed at performing various segmentation tasks.

Lifting Knowledge from 2D to 3D Several works have aimed at replicating the success of self-supervised pre-training in the 3D domain [30, 48]. However, these approaches are limited by the lack of high-quality training data in 3D. Other methods thus try to transfer the meaningful embeddings from pre-trained 2D models like CLIP to 3D. Several works focus on multi-modal contrastive pre-training, where they train a new 3D model that should be aligned with the embeddings computed from the frozen 2D CLIP encoder [16, 19, 31, 36, 47]. While these methods could exhibit similar zero-shot properties as VLMs, for example, for shape classification, they are limited by the lack of quantity and variety of 3D data for training. Another approach is to bypass the training of 3D-specific models and apply 2D encoders directly to rendered views of the 3D object or scenes. Several works propose to render shapes from different viewpoints, process the views with 2D CLIP encoders, average the resulting image embeddings over all views, and directly compare them with text prompt embeddings for tasks like zero-shot shape recognition [19, 25, 49]. Other recent works propose to use pre-trained 2D segmentation or object detection models for shape part segmentation [2] or subsequently for shape matching [1]. Paral-

lel to this work, Morreale et al. [27] proposed to use pre-trained 2D feature extractors to obtain fuzzy matches between shapes that are then projected onto the 3D shape and subsequently refined. While their approach is the most similar to our work, the core difference is that they propose to process the features in the 2D domain to obtain semantic correspondences and project these back onto the shapes, while we propose to back-project and aggregate the meaningful extracted features to the 3D shapes. Our approach is thus more versatile and allows for various downstream tasks instead of solely shape matching. A third way of using pre-trained VLMs is to optimize a separate network on a single instance, guided by the CLIP model (with frozen weights), while using differentiable rendering. This approach has been used for shape stylization [26], localization of semantic regions on shapes [14] or mesh deformation [17].

We render the 3D shape from multiple views and project features back from 2D encoders onto the surface. As we demonstrate below, by doing so, we obtain high-quality pointwise feature descriptors that enable tasks that require high accuracy, like keypoint detection.

3D Shape Descriptors A significant amount of hand-crafted shape descriptors is based on spectral methods, utilizing information from Laplacian eigenvalues and eigenfunctions of the shape. The Laplace-Beltrami operator, as one of the fundamental tools in geometry processing, gave rise to early, well-established shape descriptors like HKS [40] and WKS [4]. Another class of hand-crafted features is based on local reference frames (LRF). A prominent example is the SHOT descriptor [37]. While spectral descriptors are often used for shape matching, LRF-based methods are mainly employed for 3D recognition or registration tasks. With the rise of learning-based methods, recent attempts aim at using neural networks to learn shape descriptors [5]. While all neural methods naturally provide embeddings that can be extracted from their hidden layers, the difficulties have already been described before; as 3D data exists in various formats and there are only relatively small datasets available, such shape descriptors are usually limited in their quality and steered towards specific tasks, such as shape matching.

A general weakness of common shape descriptors is their dependence on mesh triangulation and quality (e.g., to compute robust Laplacian information), as well as missing semantic understanding. Our proposed back-projected features can be computed for any point on the surface of a mesh. They provide high-quality semantic information and can include texture information if available while being robust to low mesh quality, surface holes, and other common problems for classic shape descriptors.

3D Keypoint Detection The choice of method for 3D keypoint detection often depends on the desired downstream use. Classic keypoint detection methods are often hand-engineered methods that mark a (usually high number) of geometrically salient points on a shape regardless of their semantic meaning. In a subsequent step, these points can then be, e.g., matched with a scan at a different point in time and thus used for tasks such as shape matching or registration. Tombari et al. [42] provides an overview over classic 3D keypoint detection methods.

A more challenging task is that of finding a specific set of keypoints on 3D shapes. KeypointNet [45] is a subset of the ShapeNet [11] dataset with annotated keypoints for 16 shape categories, where for each model, there are between five and 23 semantic keypoints. One can employ plain 3D models, such as PointNet [32], to detect keypoints from shapes in a supervised manner. However, such approaches usually require large amounts of training data, which implies high costs for capturing and labeling 3D models when trying to apply these techniques in practice.

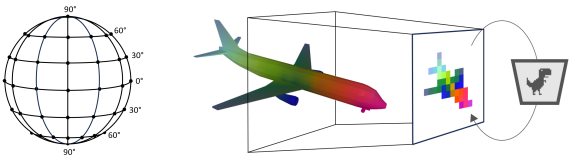
For these reasons, recent works proposed to instead focus on few-shot keypoint detection. Approaches range from learning self-supervised 3D features and fitting custom detection modules on them [3, 5] to training unsupervised keypoint detection models on large unlabeled datasets followed by a few-shot selection of the detected keypoints [46].

3. Method

As mentioned above, we consider the few-shot keypoint detection problem. Thus, we are given a small set of shapes with known keypoints, and our aim is to detect keypoints on some new target shape. For this, our overall strategy consists of transferring the keypoints from the labeled (source) shapes onto the unlabeled (target) one. Our solution comprises two main components: a point similarity component which measures the similarity between vertices of the source and target shapes, and an optimization block which aims to preserve the overall distribution of keypoints and prevent collapses, e.g., due to symmetries. Crucially, for point similarity, we propose to back-project features given by 2D foundation models onto the 3D shapes. Below, we describe first our feature extraction strategy (Sec. 3.1) and then the optimization module (Sec. 3.2) before presenting the analysis of computed features and results in the following sections. We call our method B2-3D.

3.1. 3D Feature Extraction

Our proposed pipeline for feature extraction consists of first rendering the object from multiple viewpoints around the object, processing the rendered views with a pre-trained 2D encoder, and back-projecting the computed features onto the 3D shape (see Fig. 2). While we propose to use the DINO model [28], we investigate different 2D feature extractors for shape analysis tasks in our experiments.



(a) View points used for rendering.

(b) Back-projection of features computed from rendered views.

Figure 2. After processing the rendered views of an object with a large pre-trained vision encoder (e.g., DINO), we back-project the features onto the 3D shape and aggregate the information from all views (a) to obtain rich semantic 3D features (b).

As we know the intrinsic camera parameters K, as well as the exact displacement T and rotation R of the cameras for all rendered views, we can compute the exact pixel location (x, y) of any 3D point (X_0 in homogeneous coordinates) in the rendered images as

$$\lambda(x, y, 1)^T = KC_0\boldsymbol{g}X_0 \quad \text{with} \quad \boldsymbol{g} = \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix}, \tag{1}$$

where we use a normal perspective camera with the standard projection matrix C_0 and the scalar factor λ.

Using PyTorch3D [34] for rendering, we can additionally determine whether a 3D point is visible in a rendered image. If it is, we assign the feature at the corresponding computed pixel location (x, y) to it. To aggregate the back-projected features from all views, we simply average them.

When projecting features only to the points visible in the rendered views, we observe that for more complex meshes, the point visibility information is not of sufficient quality to get a noise-free signal. To bypass this problem, we propose Gaussian geodesic re-weighting of the back-projected features which is essentially a Gaussian smoothing of the features along the surface. The feature f_i for a point i on the surface can then be computed as

$$\begin{aligned} f_i &= \frac{1}{\sum_j w(d_{ij})} \sum_j w(d_{ij}) f_j^{(b)}, \\ \text{with } w(d_{ij}) &= \exp(-d_{ij}^2 / 2\sigma^2), \end{aligned} \tag{2}$$

using the geodesic distance information $d_{i,j}$ between pairs of points (i, j), the back-projected features for visible points $f^{(b)}$, and a standard deviation σ that is left as a hyperparameter.

3.2. Few-Shot Keypoint Detection

Our goal is to capitalize on the rich semantic features won through back-projection for the task of keypoint detection. We aim at doing so in a few-shot setting, where we are given only a few shapes for a given class with an annotated set of keypoints and predict corresponding keypoint locations on a new shape from the same class.

When simply matching features of given keypoints with features for candidate positions on a new shape, the results for symmetric keypoints (e.g., the ends of table legs) may suffer from finding only a subset of the relevant points (e.g., all keypoints get matched to the same table leg on the new shape). To properly handle such symmetries, we propose an optimization of the keypoint locations that aims to match the global distribution of keypoints on the new shape to prevent a collapse.

Given the features $F_{\text{kp}} \in \mathbb{R}^{k \times d_{\text{emb}}}$ computed for k keypoints and the pairwise relative geodesic distances between these keypoints $D_{\text{kp}} \in [0, 1]^{k \times k}$ on a given shape, we can find the corresponding keypoints on a new shape with n candidate locations, for which we compute the features $F_{\text{cand}} \in \mathbb{R}^{n \times d_{\text{emb}}}$ and the pairwise relative geodesic distances $D_{\text{cand}} \in [0, 1]^{n \times n}$. Usually, the dimensions are $k \approx 10$ and $n = 2048$ in our experiments, where the keypoint candidate locations can be simply obtained through farthest-point sampling from the surface.

We can now define an optimization to find the best locations among the n keypoint candidates on the new shape. To do so, we define a right-stochastic selection matrix $S \in [0, 1]^{n \times (k+1)}$, where for each candidate point i, the probability that this point corresponds to the keypoint j is given by the value S_{ij}. The last, additional column shows the probability of a point not corresponding to any of the keypoints. We impose the right-stochastic character on S by applying a softmax per row at every optimization step. For convenience, we define $\hat{S} \in [0, 1]^{n \times k} = S_{[1,..,n;1,..,k]}$ as the first k columns of the matrix S. We then formulate the optimization objective $L = L_{\text{feature}} + \alpha L_{\text{distance}}$ with

$$\begin{aligned} L_{\text{feature}} &= \left\| \hat{S}^T F_{\text{cand}} - F_{\text{kp}} \right\|_2, \\ L_{\text{distance}} &= \left\| \hat{S}^T D_{\text{cand}} \hat{S} - D_{\text{kp}} \right\|_2. \end{aligned} \tag{3}$$

To extract the corresponding keypoints, we simply take the argmax over the columns of $\hat{S}$ after our optimization has finished.

By formalizing the keypoint search as an optimization problem, we can integrate objectives like matching the relative geodesic distances between keypoints, besides simply matching the computed features of given keypoints with the candidate points. By doing so, we also move from a per-keypoint solution to a global optimization over all keypoints at the same time, which we show to be useful in our experiments. Note that our proposed keypoint optimization module is agnostic to the used shape descriptor. We ablate the choice of DINO features in our experiments (Sec. 5.1.2).

We normalize the geodesic distances between different points to $[0, 1]$ by division through the maximal observed geodesic distance on the given shape. The use of such pairwise relative geodesic distances is more robust than simply matching average 3D positions of observed key points or

non-normalized geodesic distances, which are sensitive to shape alignment and are not rotation- or scale-invariant.

When given multiple labeled samples, one can simply average the distance matrices and features per keypoint class over all samples. Additionally, we experiment with the use of a retrieval module, where we first retrieve the closest match in the labeled shapes and then only use the distance and keypoint information from the retrieved shape. For retrieval, we additionally use the class token of the DINO model, average it over all viewpoints, and perform a nearest-neighbor search in the feature space to find the best match with the unseen shape. We analyze the performance of this retrieval module in our experiments.

4. Analysis of Back-Projected Features

Before diving deeper into the task of keypoint detection, we investigate several important properties of back-projected features. In this section, we focus on the features back-projected from the DINO model [28], which produces the best results in our keypoint detection experiments (Sec. 5).

4.1. Feature Stability

We first analyze how stable the found elements behave under changes in certain parameters of the extraction process. This analysis is of great importance as one of the key limitations of the rendering- and back-projection-based method is its sensitivity to the quality of rendered images and other scene and capture parameters.

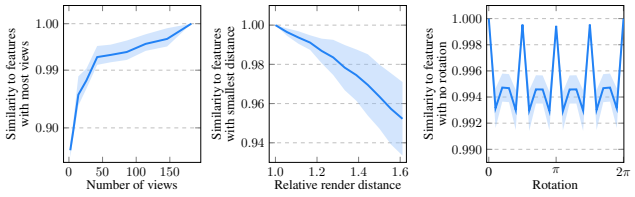


Figure 3. Feature stability analysis measuring the mean cosine similarity (with standard deviation in light blue) of extracted point features when applying modifications to the rendering process.

One of these crucial parameters is the number of different viewpoints from which we render the object. In our experiments, we use a sampling strategy for the camera positions that partitions the unit sphere around the object in n equidistant horizontal slices and spreads $2(n+1)$ viewpoints equiangular on the circumference of each slice, as well as one view from the top and one from the bottom of the object. When varying this sample parameter n, we can observe that the features seem to converge at around a total number of 50 views (see Fig. 3). In our further experiments, we thus use $n = 5$, resulting in 62 viewpoints around the object, as shown in Fig. 2a.

Generally, we observe that although the features back-projected from a single 2D image are coarse, features assembled from multiple views around the object are smoother and with a higher detail level, as shown in Fig. 2b.

We visualize the effect of increasing the number of rendering viewpoints in Fig. 4. As can be observed, the features get more distinctive and detailed with more rendered views, while with only a few views, one can clearly identify the patch-based architecture of the underlying vision transformer.

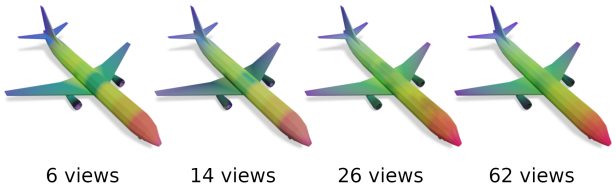


Figure 4. Increase in feature quality and distinctiveness with increasing number of rendering viewpoints. Visualization using a PCA, as described in Sec. 4.2.

We further analyze the effects of changing the camera distance during rendering on the features. We find that there is an almost linear relationship between the increase in distance and the decrease in similarity to features captured at the minimal distance where all views capture the full object. The features are likely to collapse with too large render distances as more points of the shape are back-projected to the same patch in the 2D image. In practice, this does not pose a problem to our method, as shapes can be normalized to a certain scale in a pre-processing step.

For rendering, we employ PyTorch3D using one light source at the camera position for each rendering view. Through doing so, we ensure an even lighting of the shape for feature extraction, as well as robustness to rotations around the up-axis of the object, which we assume as given. In our analysis, we observe that through the high number of viewpoints spread around the shape, the cosine similarity of features computed for rotated versions of shapes always stays above 0.99, thus proving the robustness of our capturing strategy. The three peaks in similarity when rotating the object can be explained by the perfect alignment of rotation with the viewpoints, with rotations of 90, 180, and 270 degrees, respectively.

4.2. Semantic Awareness

In order to visualize the retrieved features, we can compute them for all vertices of a given mesh and perform a dimensionality reduction using a PCA to get an interpretable 3-dimensional color vector for each point. An example is shown in Fig. 5.

The visualization of the principal components of the features suggests that the proposed method of back-projecting features on the 3D shape produces semantic features as similar, e.g., symmetric parts of the objects also get assigned

Figure 5. Back-projected ViT features on a shape can be visualized after performing a PCA to just three values per vertex. The extracted features contain rich semantic information and clearly assign different values to different semantic parts of the object.

similar values. This visualization further highlights that through this simple back-projection technique, we are able to lift the powerful features from the 2D encoder to the three-dimensional shape.

4.3. Geometric Properties

Back-projected features comprise semantic information about scenes, which can be helpful for various downstream applications in shape analysis, as this is often a missing component of geometry-based methods. However, an additional important property of shape descriptors is the understanding of pure *geometric* information.

To investigate these properties, we analyze the extracted features for two isometric cubes, where one has an inwards-dented side which is dented outwards on the other cube (see Fig. 6). While the two shapes are isometric, they are obviously not the same, and intuitively, a good shape descriptor should reflect the local geometry change with a change in the respective local features.

Classic shape descriptors that are based on the shape Laplacian, like the HKS [40] and WKS [4], fail to distinguish the two cubes and assign the exact same features for all points on the shape, as the geodesic pairwise distances between any pair of points on the surface stay the same. The SHOT descriptor [37] is sensitive to triangulation changes and its support radius parameter, resulting in the noisy behavior shown in Fig. 6. In contrast, our back-projected DINO features show a strong, localized reaction to the change in geometry while remaining similar in the unmodified parts of the shape.

Generally, we observe that the uptake to small local modifications of the shape is usually also local, with features for unrelated parts on the shape not being affected. Considering the global attention mechanism over the whole image in a ViT, this is an interesting and encouraging finding. Further examples of this behavior are shown in the supplementary materials.

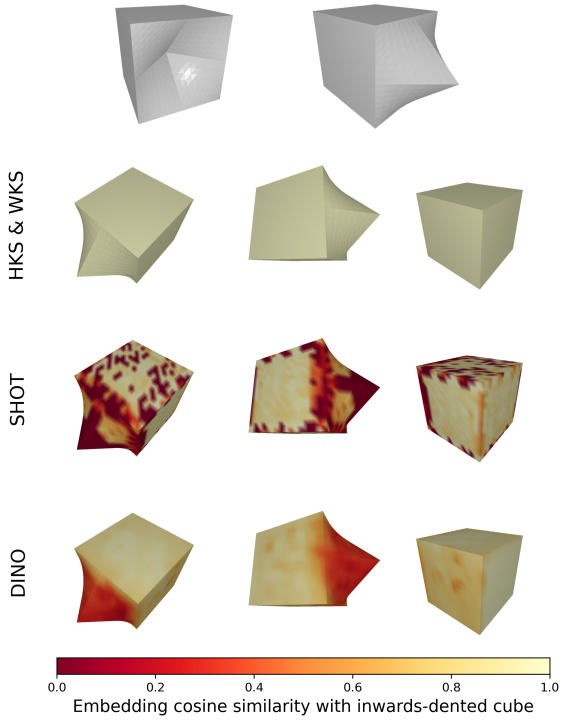


Figure 6. We compute different shape descriptors on two isometric cubes, once with an inward dent and once with an outward dent. Since the HKS and WKS signatures are based on geodesic information, the computed features for the different shapes do not change. The SHOT descriptor is sensitive to changes in triangulation, as well as to its support radius parameter, and results in noisy, changing features across the cube. Back-projected DINO features show significantly better performance as they change in the modified parts of the cube, while the features remain the same for the unmodified parts of the shape.

5. Experiments

5.1. Few-Shot Keypoint Detection

Setup We evaluate our method B2-3D on the KeypointNet dataset [45]. For each class, we select three random models from the KeypointNet dataset and use them as our few-shot samples. We compare our results to several baseline methods for few-shot keypoint detection: SIFT-3D [35], HARRIS-3D [39], ISS [50], D3Feat [5], USIP [23], UKPGAN [46], and FSKD [3].

We use the same evaluation strategy as You et al. [46], which computes the IoU of predicted and ground-truth keypoints from KeypointNet with a varying distance threshold for the evaluation. An intersection is counted if the geodesic distance between a ground-truth keypoint and a predicted keypoint is smaller than this distance threshold. We also stick to the same three classes of the dataset for evaluation: airplane, chair, and table.

5.1.1 Quantitative and Qualitative Analysis

On KeypointNet, B2-3D outperforms the previous state-of-the-art by a very significant margin on all distance thresholds (Fig. 7). It reaches similar IoU levels as FSKD at a distance threshold of 0.1, already at a threshold of around 0.045. The mean relative improvement over FSKD lies at 93% with over 200% improvement at a distance threshold of 0.02. Qualitative results are shown in Figure 1.

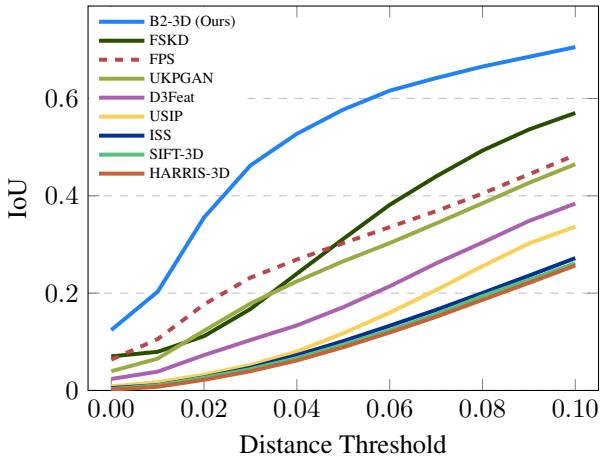


Figure 7. Our algorithm reaches a new state-of-the-art in few-shot keypoint detection by a large margin.

We also include a comparison to farthest-point sampling with the average number of keypoints observed in the few-shot samples. Our implementation also uses the maximum of the average geodesic distance to seed the first point. Surprisingly, this simple strategy, without any understanding of the underlying semantics or geometry, outperforms many of the previous works. This can be explained by how we normally select keypoints: In order to have keypoints that describe the objects as well as possible, we tend to choose extreme points that are well distributed over the shape as keypoints. Nevertheless, such a simple baseline is insufficient for accurate keypoint detection.

5.1.2 Ablation Studies

We examine the reasons for the success of our method with several ablation studies on different parts of the proposed pipeline. Our findings are illustrated in Fig. 8.

We find that changing the large DINO model for a distilled version [ViT-S] does not significantly harm the performance, but using the given textures of ShapeNet meshes does [Texture]. This is consistent with the observations made by Morreale et al. [27]; low-quality texture information is more likely to impair performance than to favor it.

We motivated our keypoint candidate optimization module with the problem of handling symmetries in an object.

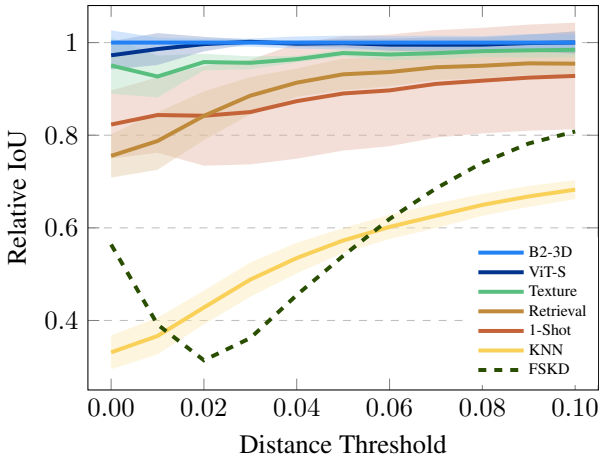


Figure 8. Relative performance (with standard deviation) compared to our optimization with features extracted from 3 shapes without textures with the DINO ViT-G model. We observe that features extracted from the smaller ViT result in almost equal performance, using (low quality) texture information slightly distracts the model, and the 1-shot performance has a higher variance but still always outperforms the previous state-of-the-art 3-shot method FSKD by a large margin. The reduced performance with simple nearest-neighbor-based keypoint detection [KNN] stresses the effectiveness of our optimization module in matching the keypoint distributions.

As we can observe in Fig. 8, the optimization successfully avoids a collapse in the prediction, which is the reason for the lower performance of a simple nearest-neighbor-based selection [KNN]. The retrieval of the most similar shape from the labeled samples before optimization does not prove to be effective with this low number of shots [Retrieval].

Finally, we also evaluate our proposed method when using only one labeled sample [1-Shot]. While the performance is worse than with three labeled samples and has a higher variance, it still outperforms the previous state-of-the-art for keypoint detection with three given labeled samples by a large margin, thus demonstrating the strength of our framework. In the supplementary materials, we additionally give insights into the hyperparameter search for α, β and σ.

Other Shape Descriptors We want to evaluate the contribution of the back-projected features and compare our framework with the same keypoint candidate optimization but with other shape descriptors. We first compare against traditional, geometry-based shape descriptors HKS [40], WKS [4] and SHOT [37]. As these geometry-based descriptors are sensitive to low-quality meshes, we pre-process the given ShapeNet models to obtain clean watertight manifolds [20]. After optimization of hyperparameters, like the diffusion time for HKS and WKS, we report the results of

using our keypoint optimization module together with the geometry-based descriptors in Fig. 9.

In addition, we compare deep features back-projected from CLIP [33] and SAM [22]. We extract the features from the last layer of the respective image-encoding vision transformers, discarding subsequent layers that no longer provide significant spatial information. For additional comparison, we also extract features from the CNN EfficientNet [41]. While all deep methods' features perform better than the traditional descriptors, they fall short of the features extracted from the DINO model. Our method with CLIP and EfficientNet features also outperforms the previous SOTA FSKD.

To our surprise, features extracted from the smaller and older EfficientNet outperform the CLIP and SAM models. We attribute this performance potentially to the fact that CLIP and SAM have specific properties (i.e., the joint image-text embedding for CLIP and the segmentation decoder for SAM) that may not necessarily be relevant in our setting. However, a more thorough investigation is necessary to fully understand the exact properties of the different features.

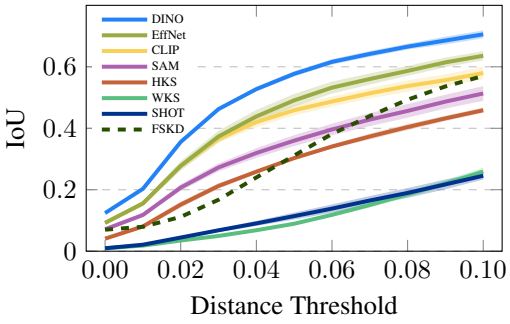


Figure 9. Experimental results when using different features with our proposed keypoint optimization module. We find that traditional shape descriptors cannot reach the performance of back-projected features of 2D foundation models, while other back-projected features cannot handle local geometry as well as DINO.

While the HKS performs the best out of the three traditional shape descriptors, the results are only as good as simple farthest-point sampling in our experiments. Since such hand-crafted methods are not aware of semantic information, they struggle to provide similar features for similar points on shapes of the same object class that have a different geometry. To validate this conjecture, we compare the features computed across multiple shapes of the same class in the supplementary materials.

5.2. Part Segmentation Transfer

As an additional validation of our results, we investigate the performance of back-projected features for transferring part segmentation labels between a pair of shapes. Using the back-projected DINO features and a simple nearest-neighbor-based classification, we obtain an average IoU of 71.0% on the ShapeNet part dataset [44], improving by nearly 2% over the previous state-of-the-art NCP [3] (69.2% IoU). Further details and results with features back-projected from other 2D models can be found in the supplementary materials.

6. Conclusions

We presented B2-3D, a novel method for few-shot keypoint detection on 3D meshes. Our method consists of back-projecting features from powerful pre-trained 2D vision encoders to the 3D shape, which carry strong semantic and geometric information, as we were able to show in a comprehensive feature analysis. In order to transfer keypoints between shapes, we match the observed keypoint distributions on the shape with a simple yet effective optimization strategy that is agnostic of the specific shape descriptor used. We demonstrated the effectiveness of our formulation by achieving state-of-the-art performance on the KeypointNet dataset by a large margin in combination with back-projected DINO features, even outperforming the previous SOTA with back-projected CNN features. We further achieve a new state-of-the-art for the task of part segmentation transfer.

While the proposed feature extraction is learning-free, the computation cost of features for multiple views around the object is slightly higher than with pure 3D-based methods. A remaining difficulty is filtering out non-existent keypoints on new shapes and detecting unseen keypoint classes. While we propose to solve the first problem using shape retrieval, the diversity and representativeness of the given labeled shapes are still essential for these problems.

Back-projected features can serve as a powerful prior for various other shape analysis tasks where pure geometric methods currently still fail, which is an exciting avenue for future research. The back-projection of different 2D features for such downstream tasks can also serve as a powerful testbed for comparing the quality of learned features, especially on photo-realistic datasets [6].

Acknowledgements. Thomas Wimmer is supported by the Konrad Zuse School of Excellence in Learning and Intelligent Systems (ELIZA) through the DAAD programme Konrad Zuse Schools of Excellence in Artificial Intelligence, sponsored by the German Federal Ministry of Education and Research. Parts of this work were supported by the ERC Starting Grant 758800 (EXPROTEA), ERC Consolidator Grant 101087347 (VEGA), ANR AI Chair AIGRETTE, and gifts from Adobe Inc. and Ansys Inc.

References

[1] Ahmed Abdelreheem, Abdelrahman Eldesokey, Maks Ovsjanikov, and Peter Wonka. Zero-shot 3d shape correspondence. *arXiv preprint arXiv:2306.03253*, 2023. 2

[2] Ahmed Abdelreheem, Ivan Skorokhodov, Maks Ovsjanikov, and Peter Wonka. Satr: Zero-shot semantic segmentation of 3d shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15166–15179, 2023. 1, 2

[3] Souhaib Attaiki and Maks Ovsjanikov. Ncp: Neural correspondence prior for effective unsupervised shape matching. *Advances in Neural Information Processing Systems*, 35:28842–28857, 2022. 2, 3, 6, 8

[4] 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

[5] Xuyang Bai, Zixin Luo, Lei Zhou, Hongbo Fu, Long Quan, and Chiew-Lan Tai. D3feat: Joint learning of dense detection and description of 3d local features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6359–6367, 2020. 2, 3, 6

[6] Florian Bordes, Shashank Shekhar, Mark Ibrahim, Diane Bouchacourt, Pascal Vincent, and Ari S Morcos. Pug: Photorealistic and semantically controllable synthetic data for representation learning. *arXiv preprint arXiv:2308.03977*, 2023. 8

[7] Edmond Boyer, Alexander M Bronstein, Michael M Bronstein, Benjamin Bustos, Tal Darom, Radu Horaud, Ingrid Hotz, Yosi Keller, Johannes Keustermans, Artiom Kovnatsky, et al. Shrec 2011: robust feature detection and description benchmark. In *4th Eurographics workshop on 3D object retrieval-3DOR 2011*, 2011. 2

[8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1

[9] M Bueno, J Martínez-Sánchez, H González-Jorge, and H Lorenzo. Detection of geometric keypoints and its application to point cloud coarse registration. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 41:187–194, 2016. 1

[10] 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

[11] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 3

[12] Boyuan Chen, Pieter Abbeel, and Deepak Pathak. Unsupervised learning of visual 3d keypoints for control. In *International Conference on Machine Learning*, pages 1539–1549. PMLR, 2021. 1

[13] An-Chieh Cheng, Xueting Li, Min Sun, Ming-Hsuan Yang, and Sifei Liu. Learning 3d dense correspondence via canonical point autoencoder. *Advances in Neural Information Processing Systems*, 34:6608–6620, 2021. 3

[14] 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

[15] Matt Deitke et al. Objaverse: A universe of annotated 3d objects. In *CVPR*, 2023. 2

[16] Mingfei Gao, Chen Xing, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Le Xue, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip: Learning a unified representation of language, images, and point clouds for 3d understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1179–1189, 2023. 2

[17] William Gao, Noam Aigerman, Thibault Groueix, Vova Kim, and Rana Hanocka. Textdeformer: Geometry manipulation using text guidance. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023. 3

[18] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 2

[19] Deepti Hegde, Jeya Maria Jose Valanarasu, and Vishal Patel. Clip goes 3d: Leveraging prompt tuning for language grounded 3d recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2028–2038, 2023. 1, 2

[20] Jingwei Huang, Yichao Zhou, and Leonidas Guibas. Manifoldplus: A robust and scalable watertight manifold surface generation method for triangle soups. *arXiv preprint arXiv:2005.11621*, 2020. 7

[21] Tomas Jakab, Richard Tucker, Ameesh Makadia, Jiajun Wu, Noah Snavely, and Angjoo Kanazawa. Keypointdeformer: Unsupervised 3d keypoint discovery for shape control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12783–12792, 2021. 1

[22] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, 2023. 2, 8

[23] Jiaxin Li and Gim Hee Lee. Usip: Unsupervised stable interest point detection from 3d point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 361–370, 2019. 6

[24] Feng Liu and Xiaoming Liu. Learning implicit functions for topology-varying dense 3d shape correspondence. *Advances in Neural Information Processing Systems*, 33:4823–4834, 2020. 3

[25] Minghua Liu, Yinhao Zhu, Hong Cai, Shizhong Han, Zhan Ling, Fatih Porikli, and Hao Su. Partslip: Low-shot part segmentation for 3d point clouds via pretrained image-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21736–21746, 2023. 1, 2

[26] Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13492–13502, 2022. 3

[27] Luca Morreale, Noam Aigerman, Vladimir G Kim, and Niloy J Mitra. Neural semantic surface maps. *arXiv preprint arXiv:2309.04836*, 2023. 3, 7

[28] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 1, 2, 3, 5

[29] Maks Ovsjanikov, Quentin Mérigot, Facundo Mémoli, and Leonidas Guibas. One point isometric matching with the heat kernel. In *Computer Graphics Forum*, pages 1555–1564. Wiley Online Library, 2010. 1

[30] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II*, pages 604–621. Springer, 2022. 2

[31] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 815–824, 2023. 2

[32] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 3

[33] 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. 1, 2, 8

[34] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020. 4, 1

[35] Blaine Rister, Mark A Horowitz, and Daniel L Rubin. Volumetric image registration from invariant keypoints. *IEEE Transactions on Image Processing*, 26(10):4900–4910, 2017. 6

[36] David Rozenberszki, Or Litany, and Angela Dai. Language-grounded indoor 3d semantic segmentation in the wild. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*, pages 125–141. Springer, 2022. 2

[37] Samuele Salti, Federico Tombari, and Luigi Di Stefano. Shot: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 125:251–264, 2014. 3, 6, 7

[38] Samuele Salti, Federico Tombari, Riccardo Spezialetti, and Luigi Di Stefano. Learning a descriptor-specific 3d keypoint detector. In *Proceedings of the IEEE international conference on computer vision*, pages 2318–2326, 2015. 2

[39] Ivan Sipiran and Benjamin Bustos. Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes. *The Visual Computer*, 27:963–976, 2011. 6

[40] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Computer graphics forum*, pages 1383–1392. Wiley Online Library, 2009. 3, 6, 7

[41] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 8

[42] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Performance evaluation of 3d keypoint detectors. *International Journal of Computer Vision*, 102(1-3):198–220, 2013. 3

[43] Hanyu Wang, Jianwei Guo, Dong-Ming Yan, Weize Quan, and Xiaopeng Zhang. Learning 3d keypoint descriptors for non-rigid shape matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018. 1

[44] Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG)*, 35(6):1–12, 2016. 8, 3

[45] Yang You, Yujing Lou, Chengkun Li, Zhoujun Cheng, Liangwei Li, Lizhuang Ma, Cewu Lu, and Weiming Wang. Keypointnet: A large-scale 3d keypoint dataset aggregated from numerous human annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13647–13656, 2020. 2, 3, 6

[46] Yang You, Wenhai Liu, Yanjie Ze, Yong-Lu Li, Weiming Wang, and Cewu Lu. Ukpgan: A general self-supervised keypoint detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17042–17051, 2022. 3, 6

[47] Yihan Zeng, Chenhan Jiang, Jiageng Mao, Jianhua Han, Chaoqiang Ye, Qingqiu Huang, Dit-Yan Yeung, Zhen Yang, Xiaodan Liang, and Hang Xu. Clip2: Contrastive language-image-point pretraining from real-world point cloud data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15244–15253, 2023. 2

[48] Renrui Zhang, Ziyu Guo, Peng Gao, Rongyao Fang, Bin Zhao, Dong Wang, Yu Qiao, and Hongsheng Li. Point-m2ae: Multi-scale masked autoencoders for hierarchical point cloud pre-training. In *Advances in Neural Information Processing Systems*, 2022. 2

[49] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by clip. In *Pro-*

ceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8552–8562, 2022. 1, 2

[50] Yu Zhong. Intrinsic shape signatures: A shape descriptor for 3d object recognition. In *2009 IEEE 12th international conference on computer vision workshops, ICCV Workshops*, pages 689–696. IEEE, 2009. 6