# iSeg: Interactive 3D Segmentation via Interactive Attention

ITAI LANG, University of Chicago, United States of America
FEI XU, University of Chicago, United States of America
DALE DECATUR, University of Chicago, United States of America
SUDARSHAN BABU, Toyota Technological Institute at Chicago, United States of America
RANA HANOCKA, University of Chicago, United States of America

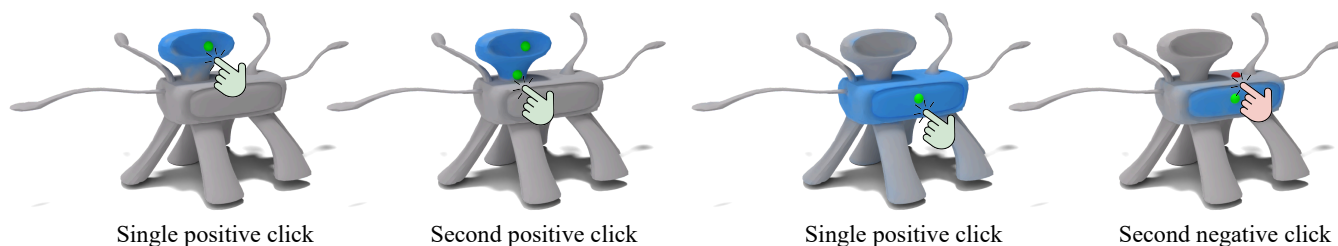| Single positive click | Second positive click | Single positive click | Second negative click |

Fig. 1. iSeg computes customized fine-grained segmentations on shapes interactively specified by user clicks. The clicks, denoted by a green or a red dot, indicate whether to include or exclude regions, respectively. Our method is capable of segmenting regions that are not accurately specified by text.

We present iSeg, a new interactive technique for segmenting 3D shapes. Previous works have focused mainly on leveraging pre-trained 2D foundation models for 3D segmentation based on text. However, text may be insufficient for accurately describing fine-grained spatial segmentations. Moreover, achieving a consistent 3D segmentation using a 2D model is highly challenging, since occluded areas of the same semantic region may not be visible together from any 2D view. Thus, we design a segmentation method conditioned on fine user clicks, which operates entirely in 3D. Our system accepts user clicks directly on the shape's surface, indicating the inclusion or exclusion of regions from the desired shape partition. To accommodate various click settings, we propose a novel interactive attention module capable of processing different numbers and types of clicks, enabling the training of a single unified interactive segmentation model. We apply iSeg to a myriad of shapes from different domains, demonstrating its versatility and faithfulness to the user's specifications. Our project page is at https://threedle.github.io/iSeg/.

CCS Concepts: • **Computing methodologies → Shape representations**; *Image segmentation*; • **Human-centered computing → Interactive systems and tools**.

Additional Key Words and Phrases: 3D mesh representation, Interactive segmentation, Differentiable rendering, Deep Learning, Knowledge distillation

Authors' Contact Information: Itai Lang, University of Chicago, United States of America, itailang@uchicago.edu; Fei Xu, University of Chicago, United States of America, feixu@uchicago.edu; Dale Decatur, University of Chicago, United States of America, ddecatur@uchicago.edu; Sudarshan Babu, Toyota Technological Institute at Chicago, United States of America, sudarshan@ttic.edu; Rana Hanocka, University of Chicago, United States of America, ranahanocka@uchicago.edu.

## 1 Introduction

Interactive 3D segmentation, the ability to select fine-grained segments from a 3D shape based on user inputs like clicks, is a fundamental problem in computer graphics with broad implications. In fields such as computer-aided design and 3D modeling, precise segment selection facilitates detailed model refinement. Moreover, in engineering, architecture, and medicine, fine-grained selection is indispensable for simulation and analysis, allowing for accurate assessment of structural integrity and behavior. While important, this problem poses significant challenges. How can we decipher the user intentions from such a minimal input as clicks? How do we handle diverse shapes with varying geometries and select specific and unique shape parts? In this work, we propose a method tailored to the shape at hand that selects regions adhering to the user clicks.

Traditional segmentation techniques do not utilize user inputs and instead rely on geometric features to delineate semantic parts [Cornea et al. 2007; Dey and Zhao 2004; Hoffman and Richards 1984; Lien and Amato 2007; Shamir 2008; Zheng et al. 2015]. Recent data-driven techniques have further leveraged fully annotated 3D datasets and achieved ipressive 3D segmentation results [Chen et al. 2019; Deng et al. 2020; Hanocka et al. 2019; Hu et al. 2022; Milano et al. 2020; Milletari et al. 2016; Qi et al. 2017; Sharp et al. 2022; Sun et al. 2021; Yi et al. 2017; Zhu et al. 2020]. However, the reliance on a dataset and the scarcity of large-scale 3D datasets limits the network to a specific shape domain with a predefined set of parts.
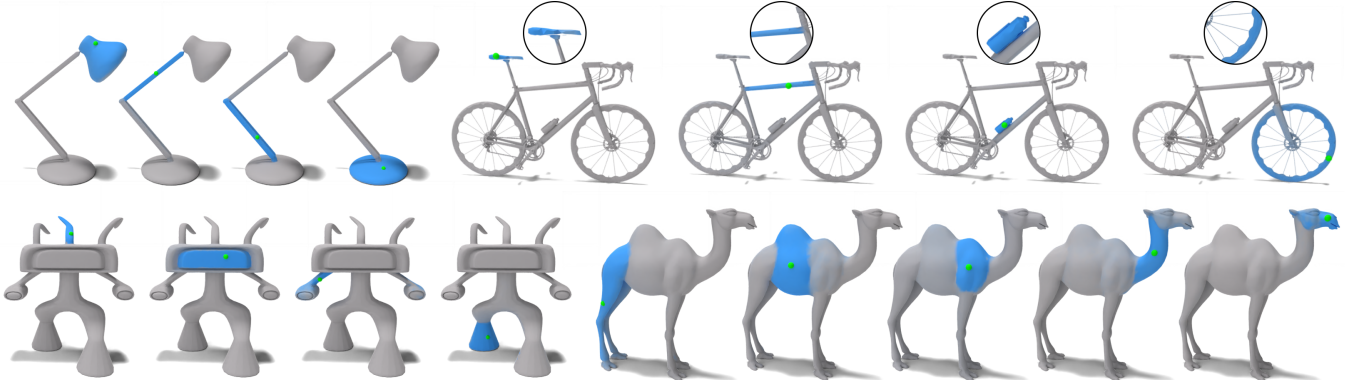
Fig. 2. **Fine-grained segmentation from a single positive click.** iSeg is capable of generating granular segmentations (visualized in blue) given a single click as input (depicted with a green dot). Our method is highly flexible and can select parts that vary in size, geometry, and semantic meaning.

Current 3D segmentation methods have circumvented the dependency on 3D data and pre-determined part definition by utilizing pre-trained 2D foundation models to learn semantic co-segmentation [Ye et al. 2023] or text-driven segmentation [Abdelreheem et al. 2023a,b; Decatur et al. 2024, 2023; Ha and Song 2022; Kim and Sung 2024; Liu et al. 2023]. Nonetheless, text may not be able to accurately describe all fine-grained segmentations, such as the fourth leg of an octopus or a region corresponding to a particular point on the shape.

In this paper, we present *iSeg*, a new data-driven interactive technique for 3D shape segmentation that generates customized partitions of the shape according to user clicks. Given a shape represented as a triangular mesh, the user selects points on the mesh interactively to indicate a desired segmentation and iSeg predicts a region over the mesh surface that adheres to the clicked points. Our interactive interface can utilize positive and negative clicks, enabling additions and exclusions of areas from the segmented region, respectively (see Fig. 1).

We harness the power of a pretrained 2D foundation segmentation model [Kirillov et al. 2023] and distill its knowledge to 3D. However, segmenting a meaningful 3D region using a 2D model is very challenging, since occluded shape regions cannot be seen together from a single 2D view. Accordingly, we design an interactive segmentation system that operates *entirely in 3D*, where the user clicks and the inferred corresponding region are applied over the shape surface *directly*, ensuring 3D consistency *by construction*. During training only, we project the 3D user clicks and the predicted segmentation to multiple 2D views to enable supervision from the powerful pretrained foundation model [Kirillov et al. 2023].

For interactiveness, we want our system to accommodate different user inputs, meaning, point clicks that can vary in number and type. Instead of training a separate segmentation model for each user click configuration, we propose a novel interactive attention mechanism, which learns the representation of positive and negative clicks and computes their interaction with the other points of the mesh. This attention layer consolidates variable-size guidance into a fixed-size representation, resulting in a unified flexible segmentation model capable of predicting shape regions for various click settings.

iSeg is optimized per mesh to capture its unique segments, without any ground-truth annotations. We train the model with only a

small fraction of the mesh vertices, while the model successfully infers segmentations for other vertices not used during training. iSeg further generalizes beyond its training data and computes complete segments in 3D for clicks and regions occluded from each other.

In summary, this paper presents iSeg, an interactive method for selecting customized fine-grained regions on a 3D shape. We distill inconsistent feature embeddings of a 2D foundation model into a coherent feature field over the mesh surface and decode it along with user inputs to segment the mesh on the fly. Our interactive attention mechanism handles a variable number of user clicks that can signify both the inclusion and exclusion of regions. We showcase the effectiveness of iSeg on a variety of meshes from different domains, including humanoids, animals, and man-made objects, and show its flexibility for various segmentation specifications.

## 2 Related Work

### 2.1 Non-Interactive 3D Segmentation

A large body of research has been focused on 3D segmentation using annotated datasets [Armeni et al. 2017; Hanocka et al. 2019; Hu et al. 2022; Kalogerakis et al. 2017; Milano et al. 2020; Sharp et al. 2022; Yi et al. 2017]. Such models demonstrate impressive performance at the cost of being restricted to the domain of the training data and the set of manually defined semantic labels. A partial solution to this limitation is utilizing unlabeled data, where common semantic elements are discovered by unsupervised learning [Chen et al. 2019; Deng et al. 2020; Hong et al. 2022; Sun et al. 2021; Zhu et al. 2020]. Still, the segmentation is confined to the learned parts and is not easy to alter.

In contrast, our segmentation approach is highly versatile and flexible. It is applied to various shapes from different domains. Our model is trained without any segment labels, and instead, it is optimized to the shape at hand to discover its unique partitions. Moreover, iSeg is interactive – its segmentation result can be updated simply with an intuitive user-click interface.

### 2.2 Lifting 2D Foundation Models to 3D

The emergence of powerful 2D foundation models with a broad semantic understanding has propelled a surge of interest in distilling
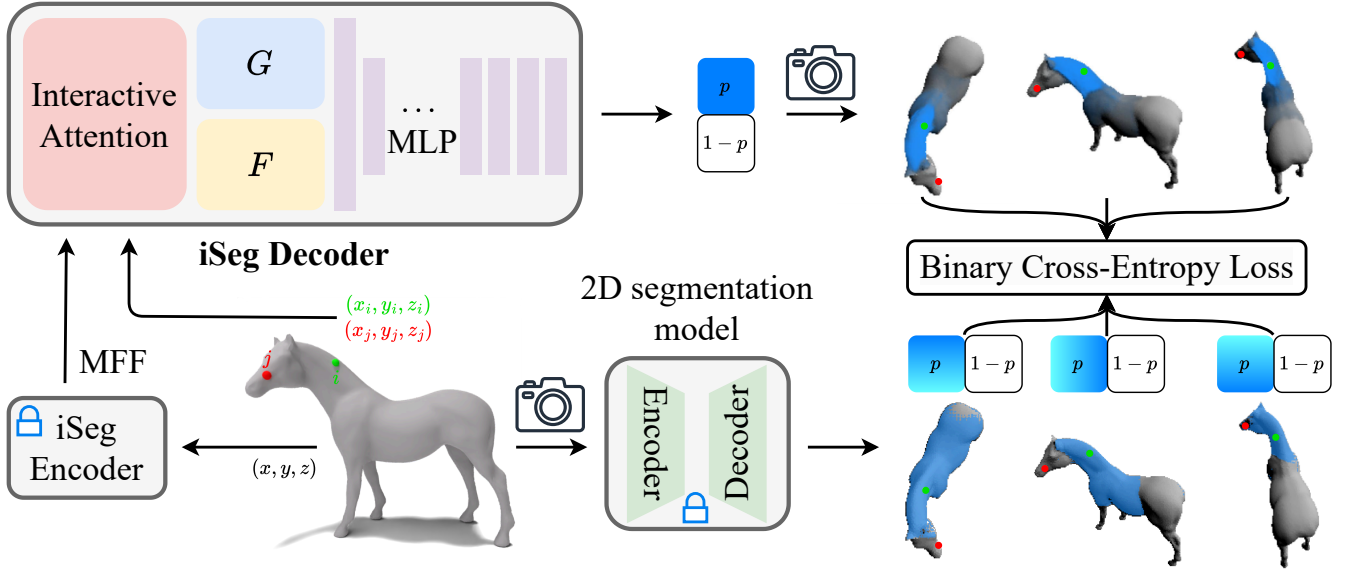
Fig. 3. **Training of the iSeg decoder.** Our decoder takes the Mesh Feature Field (MFF) computed by the iSeg encoder, along with the user input clicks, and generates a 3D segmentation map visualized in blue. We leverage a pre-trained 2D segmentation model [Kirillov et al. 2023] to supervise our training with 2D segmentation masks using rendered images of the shape and the 2D projection of the 3D clicks. Although iSeg is trained using noisy and inconsistent 2D segmentations, it is view-consistent by construction.

their knowledge and lifting it to a 3D representation [Abdelreheem et al. 2023a,b; Chen et al. 2023a; Decatur et al. 2024, 2023; Fan et al. 2023; Kundu et al. 2020; Peng et al. 2023; Umam et al. 2024; Yin et al. 2024; Zhang et al. 2022]. Notably, several researchers [Kerr et al. 2023; Kobayashi et al. 2022; Tschernezki et al. 2022] augmented the neural radiance scene representation (NeRF) [Mildenhall et al. 2020] with a volumetric feature filed. This approach enabled text-driven segmentation of objects within the scene, alleviating the need for a training dataset.

Similarly, we lift the features of a 2D foundation model [Kirillov et al. 2023] into 3D. However, instead of using the implicit NeRF representation [Kerr et al. 2023; Ye et al. 2023], our model operates directly on explicit 3D meshes, making it readily adaptable to 3D modeling workflows. Moreover, rather than decoding the feature field by a simple correlation with the embedding of the semantic prompt [Fan et al. 2023; Peng et al. 2023], we learn a dedicated decoder in 3D to exploit the semantic information embodied within our mesh feature field better.

## 2.3 Interactive 3D Segmentation

Traditional interactive techniques have utilized heuristic smoothness priors and formulated the problem with a graph cut optimization objective [Boykov and Jolly 2001; Rother et al. 2004; Sormann et al. 2006]. More recently, several learning-based methods have been proposed for interactive segmentation [Goel et al. 2023; Kontogianni et al. 2023; Mirzaei et al. 2023; Ren et al. 2022; Ying et al. 2024; Yue et al. 2024]. For example, [Kontogianni et al. 2023] segmented 3D point clouds based on user clicks. Unlike our work, they constructed a dataset for training their model, which limited its utility to parsing objects from a scene.

Very recently, [Kirillov et al. 2023] presented a foundation model for 2D interactive segmentation termed SAM, which triggered a line of follow-up works aiming at harnessing SAM's impressive capabilities to the 3D domain [Cen et al. 2023; Chen et al. 2023b; Yang et al. 2023; Zhang et al. 2023]. One approach is to segment 2D projections of the 3D data and fuse them in 3D. However, such an approach requires high user guidance, as the segmentation is performed in 2D, and the user's input is required for different views.

Another approach taken by [Chen et al. 2023b] is to lift SAM's features to a NeRF representation and use SAM's decoder to obtain the segmentation masks. As in the first approach, applying the segmentation in 2D limits the method's capabilities, since it cannot segment together occluded regions in 3D that are not visible concurrently in any 2D view. In contrast to existing works, our model and the user clicks are applied directly in 3D, simplifying the segmentation process and enabling the native segmentation of meaningful regions in 3D (as demonstrated in Fig. 5).

## 3 Method

Given a 3D shape depicted as a mesh $\mathcal{M}$ with vertices $V = \{v_i\}_{i=1}^n$, $v_i = (x_i, y_i, z_i) \in \mathbb{R}^3$, and a set of selected vertices by the user representing positive or negative clicks, our goal is to predict the per-vertex probability $P = \{p_i\}_{i=1}^n$, $p_i \in [0, 1]$ of belonging to a region adhering to the user inputs. Our system offers an interactive user interface, such that the number of clicks and their type can be varied and the segmented region of the shape is updated accordingly.

We tackle the problem by proposing an interactive segmentation technique consisting of two parts: an encoder that maps vertex coordinates to a deep semantic vector and a decoder that takes the vertex features and the user clicks and predicts the corresponding
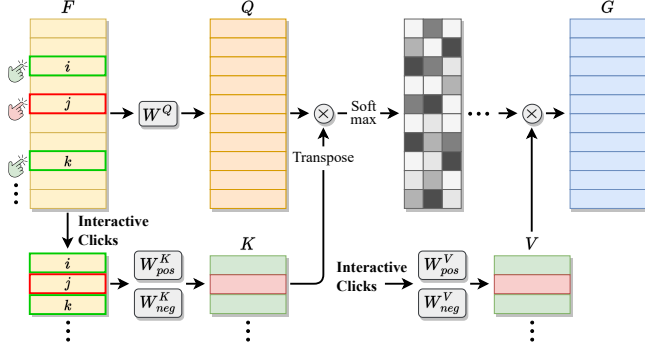
Fig. 4. **Interactive Attention.** Our interactive attention layer can handle a variable number of user clicks. The clicks may be positive or negative to indicate region inclusion or exclusion, respectively.
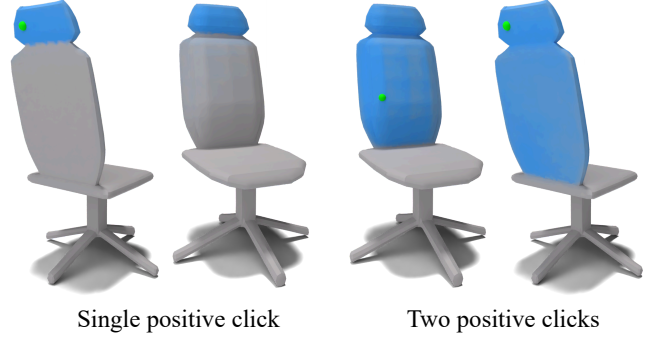


Single positive click      Two positive clicks

Fig. 5. **Native 3D segmentation.** iSeg segments parts in a 3D-consistent manner, regardless of whether the surface is occluded from the point click. A point is selected on the back of the chair (left), which is not visible from the front view. Still, our method delineates the occluded surface even though the 2D training data cannot contain this information. Furthermore, we may input two point clicks occluded from each other, one on the back of the chair and one on the front (right). These points cannot be simultaneously input to any 2D decoder, as they are not visible concurrently from any single viewpoint. Nonetheless, iSeg faithfully segments the whole backrest part.

mesh segment. The decoder contains an interactive attention layer supporting a variable number of clicks, which can be positive or negative, to increase or decrease the segmented region. Fig. 3 presents an overview of the method.

### 3.1 Mesh Feature Field

Our encoder learns a function $\phi \colon \mathbb{R}^3 \to \mathbb{R}^d$ that embeds each mesh vertex into a deep feature vector $\phi(v_i) = f_i$, where $d$ is the feature dimension. The collection of mesh vertex features is denoted as $F \in \mathbb{R}^{n \times d}$ and regarded as the Mesh Feature Field (MFF).

The encoder distills the semantic information from a pretrained 2D foundation model for image segmentation [Kirillov et al. 2023] and facilitates a 3D consistent feature representation for interactive segmentation of the mesh. To train the encoder, we render the high-dimensional vertex attributes differentiably and obtain the 2D projected features:

$$I_f^\theta = \mathcal{R}(\mathcal{M}, f, \theta) \in \mathbb{R}^{w \times h \times d}, \tag{1}$$

where $\mathcal{R}$ is a differentiable renderer, $\theta$ is the viewing direction, $f$ represents the visible vertices in the view, and $w \times h$ are the spatial dimensions of the rasterized image.

The encoder is implemented as a multi-layer perceptron network. To supervise its training, we render the mesh into a color image $I_c^\theta$ and pass it through the encoder $E_{2D}$ of the 2D foundation model [Kirillov et al. 2023] to obtain a reference feature map:

$$I_e^\theta = E_{2D}(I_c^\theta) \in \mathbb{R}^{w \times h \times d}. \tag{2}$$

This process is repeated for multiple random viewing angles $\Theta$, and our encoder is trained to minimize the discrepancy between the rendered MFF and the reference 2D features:

$$\mathcal{L}_{enc} = \frac{1}{|\Theta|} \sum_{\theta \in \Theta} ||I_f^\theta - I_e^\theta||^2. \tag{3}$$

The 2D model operates on each image separately and might produce inconsistent features for different views of shape. In contrast, our MFF is defined in 3D and is view-consistent by construction. It consolidates the information from the multiple views and lifts the 2D embeddings to a coherent field over the mesh surface. Additionally, we emphasize that the MFF is optimized independently

of the user inputs. This is a key consideration in our method, resulting in a condition-agnostic representation, describing inherent semantic properties of the mesh. We validate this design choice with an ablation experiment demonstrated in Fig. 16 and explained in the supplementary. The MFF is optimized until convergence and then utilized together with the user click prompts to compute the interactive mesh partition.

### 3.2 Interactive Attention Layer

The interactive attention layer is part of the decoding component of our system (Fig. 3). Its structure is illustrated in Fig. 4. The layer computes the interaction between the user input clicks and the mesh vertices, accommodating variable numbers and types of clicks, positive and negative. This key element in our method enables a unified decoder architecture supporting various user click settings.

Our interactive attention extends the scaled dot-product attention mechanism [Vaswani et al. 2017]. The features of the positively and negatively clicked vertices are marked as $F_{pos} \in \mathbb{R}^{n_{pos} \times d}$ and $F_{neg} \in \mathbb{R}^{n_{neg} \times d}$, respectively, where $n_{pos}$ and $n_{neg}$ are the number of clicks of each type. The interactive attention layer projects the mesh features $F$ to Queries and the features of the clicked points to Keys and Values:

$$Q = FW^Q \in \mathbb{R}^{n \times d}$$
$$K_{\{pos,neg\}} = F_{\{pos,neg\}} W^K_{\{pos,neg\}} \in \mathbb{R}^{n_{\{pos,neg\}} \times d} \tag{4}$$
$$V_{\{pos,neg\}} = F_{\{pos,neg\}} W^V_{\{pos,neg\}} \in \mathbb{R}^{n_{\{pos,neg\}} \times d},$$

where $W^Q, W^K_{\{pos,neg\}}, W^V_{\{pos,neg\}} \in \mathbb{R}^{d \times d}$ are learnable weight matrices. Then, the mesh vertices are attended to the user clicks to obtain the conditioned mesh features:

$$G = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right) V \in \mathbb{R}^{n \times d}, \tag{5}$$
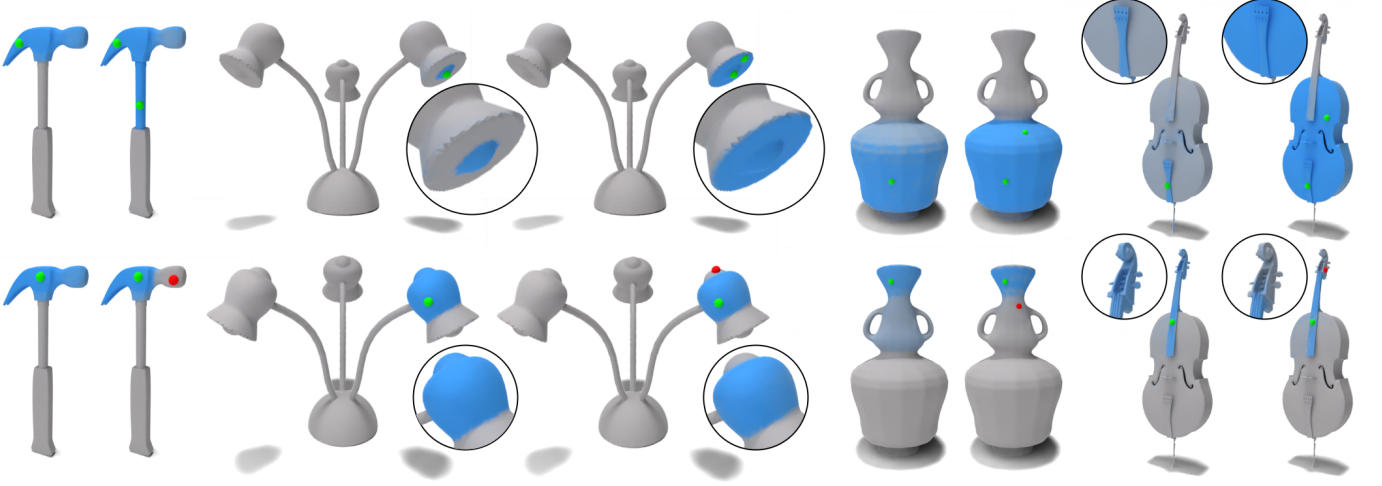
Fig. 6. **Couple of clicks results.** iSeg produces fine-grained segmentations from a couple of clicks (both positive and negative) as input. Each pair of shapes starts with a single positive click (left), which can be further customized using an additional click (right).

where $K, V \in \mathbb{R}^{(n_{pos}+n_{neg}) \times d}$ are the concatenation of $K_{pos}, K_{neg}$ and $V_{pos}, V_{neg}$, respectively.

Our attention mechanism condenses variable interactive user inputs into a fixed-length output. It learns the representation of positive and negative clicks, correlates the mesh vertices with them, and yields updated vertex features to enable the on-the-fly segmentation of the shape. Another benefit of our attention layer is that it is permutation invariant w.r.t. the user clicks. In other words, it is independent of the sequential order of the point clicks and consistent in their joint influence on the shape partition. Moreover, the attention's output $G$ is permutation equivariant w.r.t. the vertex order in $F$, a desired property for the mesh data structure.

### 3.3 Segmentation Prediction

The output of our model is a segmentation of the mesh that adheres to the user clicks, represented as the per-vertex probability of belonging to the desired region. To do so, we learn to decode the a posteriori condition dependent vertex features $g_i = [G]_i$ and the a priori inherent embedding $f_i = [F]_i$ into the partition probability:

$$p_i = \psi(f_i, g_i) \in [0, 1]. \tag{6}$$

$\psi$ is implemented as a multi-layer perceptron network, where $f_i$ and $g_i$ are concatenated along the feature dimension at the network's input. The remaining question is – how to supervise the training of such an obscure and ill-defined problem?

Similar to our encoder's training, we translate the problem to the 2D domain and harness the power of the 2D foundation model [Kirillov et al. 2023] for our 3D decoder learning (Fig. 3). We project the mesh probability map with a differentiable rasterizer to a probability image $I_p^{\theta'} = \mathcal{R}(\mathcal{M}, p, \theta') \in [0, 1]^{w' \times h' \times 2}$, where $\theta'$ is the viewing angle, and the image channels represent the segment and background probabilities. Then, we project the 3D clicks to their corresponding 2D pixels and use them as prompts to segment the rendered color mesh image with the 2D segmentation model, resulting in the supervising probability mask $I_m^{\theta'} \in \{0, 1\}^{w' \times h' \times 2}$. We

randomize the viewing direction $\theta'$ and train our decoder subject to the optimization objective:

$$\mathcal{L}_{dec} = \frac{1}{|\Theta'|} \sum_{\theta' \in \Theta'} \text{CE}(I_p^{\theta'}, I_m^{\theta'}), \tag{7}$$

where CE denotes the binary cross-entropy loss.

To prepare data for our 3D decoder training, we simulate user clicks and generate masks from the 2D model [Kirillov et al. 2023]. The data generation process includes two phases. First, we pick a small training subset of 3% of the mesh vertices well distributed over the shape using Farthest Point Sampling [Eldar et al. 1997], where each vertex is regarded as a single positive click. For each vertex, we generate random views, feed each one through the 2D foundation model, and get the reference segmentation mask. Then, for each view, we sample another training vertex visible within the viewing angle, which is set to be a positive or a negative click, and compute the updated segmentation by the 2D model. According to its type, we require the second click to increase or decrease the previous segmentation mask to obtain rich and diverse training data.

As seen in Fig. 3, the supervision signal is highly inconsistent. The 3D shape and the same clicks are interpreted differently by the 2D model, yielding strong variations in the 2D segmentation masks. Nevertheless, our method reveals a coherent underlying 3D segmentation function out of the noisy 2D measurements. Our decoder utilizes the robust distilled 3D vertex features, applies their interaction with the clicked points, and computes the region probability map directly in 3D. iSeg segmentaitons are view-consistent *by construction*, improving substantially over its training data. Furthermore, although trained with only 2D supervision, iSeg delineates meaningful regions in 3D that are not entirely visible in a single 2D projection (Fig. 5).

### 4 Experiments

We evaluate iSeg in a variety of aspects. First, we demonstrate the generality and fidelity of our method in Secs. 4.1 and 4.2, respectively.

Table 1. **Quantitative evaluation on PartNet.** We compare the segmentation performance of different clicked-based interactive techniques on shapes from the PartNet dataset [Mo et al. 2019]. IoU stands for Intersection over Union. iSeg's scores are substantially higher than those of the alternatives.

| Method | InterObject3D | SAM Baseline | iSeg (ours) |
|---|---|---|---|
| Accuracy ↑ | 0.54 | 0.76 | **0.95** |
| IoU ↑ | 0.38 | 0.51 | **0.90** |

Table 2. **Perceptual user study.** We evaluate the 3D segmentation effectiveness on a scale of 1 to 5, corresponding to completely ineffective and completely effective segmentation. Our method is considered much more effective than the competitors.

| Method | InterObject3D | SAM Baseline | iSeg (ours) |
|---|---|---|---|
| Effectiveness ↑ | 2.54 | 3.02 | **4.55** |

Then, in Sec. 4.3, we showcase the generic feature information captured by iSeg. Finally, Sec. 4.4 presents the strong generalization power of iSeg in terms of the selected point, views of the click, and the number of clicks.

We apply our method to diverse meshes from different sources: COSEG [van Kaick et al. 2011], Turbo Squid [TurboSquid 2021], Thingi10K [Zhou and Jacobson 2016], Toys4k [Rehg 2022], SCAPE [Anguelov et al. 2005], SHREC '19 [Melzi et al. 2019], ModelNet [Wu et al. 2015], ShapeNet [Chang et al. 2015], and PartNet [Mo et al. 2019]. iSeg is highly robust to the shape properties. It operates on meshes with different numbers of vertices and various geometries, including thin, flat, and high-curvature surfaces.

iSeg is implemented in PyTorch [Paszke et al. 2017] and its training time varies according to the number of mesh vertices. For a mesh with 3000 vertices, the optimization takes about 3 hours on a single Nvidia A40 GPU. Training the model is a one-time offline phase. Once trained, querying the model with input clicks takes only about 0.7 seconds, which allows fast interactive interaction with the shape. In our experiments, we used SAM ViT-H with an image size of $224 \times 224$. To obtain fine-grained segmentations, we utilized the smallest scale mask from SAM for the projected clicked points. Additional details are provided in the supplementary.

### 4.1 Generality of iSeg

iSeg is highly versatile and works on a variety of shapes and geometries. It is not limited to any specific shape category nor a pre-defined set of parts and can be applied to meshes from various domains, including humanoids, animals, musical instruments, household objects, and more. Our method is also applicable to shapes with complex geometric structures and is optimized to capture the elements of the given object.

Fig. 2 presents different single-click results. iSeg successfully segments regions with sharp edges, such as the neck of the lamp and the thin spokes of the bike. It also captures accurately the flat surface of the alien's head and the curved lower part of its leg. Moreover, iSeg can segment small parts of the shape - the bike's seat and the water bottle, or larger portions, such as the body parts of the camel.

### 4.2 Fidelity of iSeg

Our method's training is supervised by segmentation masks generated from SAM [Kirillov et al. 2023] for 2D renderings of the shape. As we show in Fig. 3, SAM's masks differ substantially between views. In contrast, iSeg manages to fuse the noisy training examples into a coherent 3D segmentation model that corresponds to the

clicked vertices. Examples are presented in Figs. 2 and 6. In the supplementary, we further demonstrate the method's 3D consistency.

iSeg is adapted to the granularity of the given mesh, which enables it to adhere to the user's clicks and segment the region of the shape related to the user's inputs. For example, in Fig. 2, for a click on the alien's middle antenna, the entire antenna and *only that particular antenna* is selected. We see similar behavior for other clicks, such as the one on the lamp's base and the camel's neck.

In Fig. 6, we further show the results of iSeg for a couple of clicks. We incorporate either a second positive click that extends the segmented part or a second negative click that retracts the region. For example, with the first click, the fine-grained bulb area of the lamp is segmented. Then, the second positive click enables to include the flat surface surrounding the bulb. The negative click, on the other hand, offers the control to reduce and refine the segmentation region. As shown in Fig. 6, the first click on the hammer's head segments the entire head. The front part can be easily and intuitively removed by a second negative click on it.

*Quantitative evaluation.* As far as we can ascertain, there are no annotated datasets for click-based interactive segmentation of 3D shapes. Thus, we adapted the part segmentation dataset PartNet [Mo et al. 2019] for our setting. The evaluation included 170 meshes sourced from all the categories in the dataset. For each mesh, we selected five test vertices from a part at random, where each vertex was regarded as a single click, and measured how well the part was segmented. We used two evaluation metrics: Accuracy (ACC) and Intersection over Union (IoU). Further details about the evaluation are provided in the supplemental material.

We considered two alternatives for comparison, InterObject3D [Kontogianni et al. 2023] and a baseline we constructed based on SAM's 2D segmentations. InterObject3D is a recent work on interactive segmentation of 3D objects. In our experiments, we employed the publicly available pretrained model released by the authors.

For the SAM baseline, we rendered the shape and projected the clicked point to 2D from 100 random views, computed SAM's mask, and re-projected the result back to 3D for each visible vertex. Then, we averaged the predictions according to the number of times each vertex was seen. In the supplementary material, we discuss additional baselines we devised using SAM. Other interactive segmentation techniques use an implicit 3D representation [Chen et al. 2023b] or perform a different task (object detection) [Zhang et al. 2023], and thus, are not directly comparable to our method.

Tab. 1 presents the ACC and IoU averaged over the test clicks and shapes, and Fig. 7 shows visual examples. InterObject3D does not select the part properly and yields a partial segmentation. SAM Baseline segments the region of the part where the click is visible.

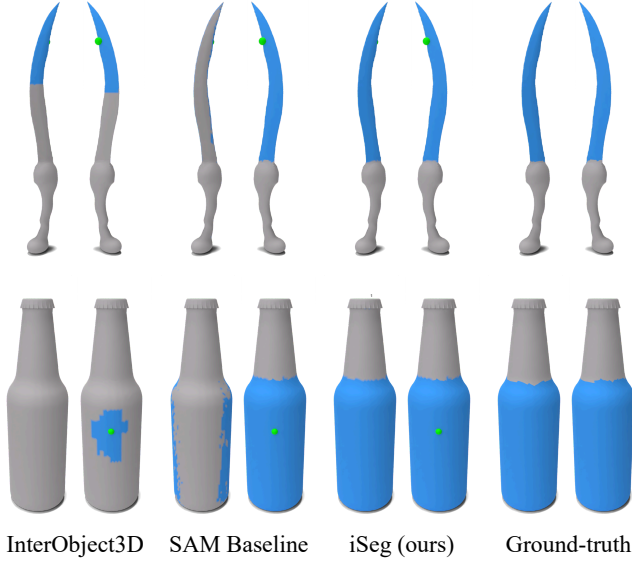InterObject3D · · · · · · SAM Baseline · · · · · · iSeg (ours) · · · · · · Ground-truth

Fig. 7. **Segmentation comparison on PartNet.** Each pair shows different views of the segmentation result for different interactive techniques. The other methods produce a partial segmentation of the part containing the click. In contrast, iSeg obtains an accurate result, which is similar to the ground-truth annotation of the part.

Clicked point · · · · · · Self-segmentation · · · · · · Cross-segmentation
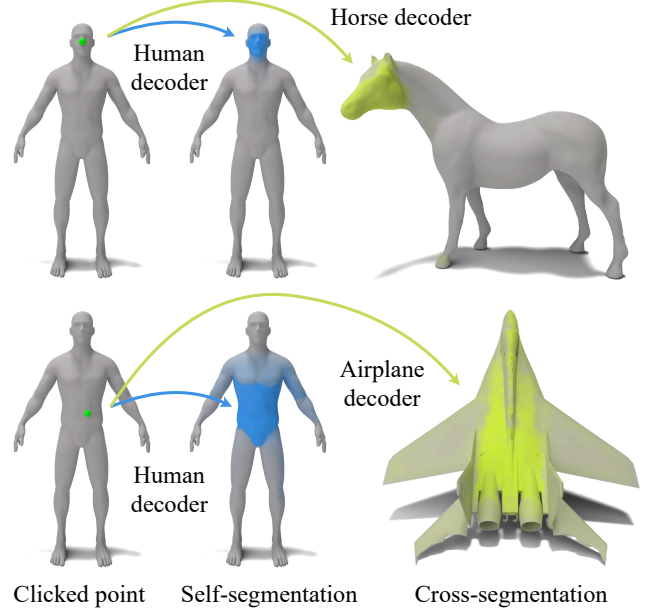
Fig. 8. **Cross-domain segmentation.** iSeg optimizes a condition-agnostic feature field, which is capable of transferring between shapes. The feature vector of a point click of one mesh (left) is used to segment the same shape (middle) as well as *another* shape from a *different* domain (right).

However, occluded regions that belong to the part are not marked. In contrast, iSeg adheres to the clicked point. It segments a coherent region in 3D, which is similar to the ground-truth part label, and achieves much higher ACC and IoU than the baselines.

*Perceptual user study.* iSeg is not limited to a particular shape type from a dataset or specific parts defined in the dataset. In such cases, ground-truth labels are unavailable. Thus, to evaluate the effectiveness of the flexible and diverse segmentations offered by our method, we opt to perform a perceptual user study. We used 20 meshes from different categories, such as humanoids, animals, and man-made objects, and included 40 participants in our study.

For each mesh, we showed the 3D segmentation for a clicked point from multiple viewing angles and asked the participants to rate the effectiveness of the result on a scale of 1 to 5. The score 5 refers to a completely effective segmentation, where the entire 3D region corresponding to the clicked point is selected. When part of the 3D region is marked, the segmentation is considered partially effective. The score 1 is defined as a completely ineffective segmentation and refers to no region selection. Examples are presented in Fig. 14.

Tab. 2 summarizes the effectiveness score averaged over all the meshes and participants. Fig. 15 further compares our method with the SAM baseline. As seen in the figure and reflected by the table, the participants rated the effectiveness of iSeg much higher than the other methods, indicating its fidelity to the clicked point.

*Applications.* iSeg computes contiguous and localized shape partitions. These segmentations can be used to extract shape parts easily and enable applications such as full-shape segmentation and local geometric editing of the mesh. These results are demonstrated

in Figs. 9, 10 and 13. Further details and discussion appear in the supplementary material.

### 4.3 Generic Feature Information

Our mesh feature field is distilled directly from SAM's encoder and is independent of the user's clicks for segmentation. Thus, although iSeg is optimized per mesh, the semantic feature representation is shared across shapes. We demonstrate this property by cross-domain segmentation. In this experiment, we use the point encoded features from one shape and predict region probability with a *different* shape's decoder. Fig. 8 shows examples. The transferable features enable the creation of cross-domain shape analogies, such as how the belly of a human corresponds to the *"belly"* of an airplane.

### 4.4 Generalization Capabilities

*Unseen mesh vertices.* Our method exhibits strong generalization power. First, we emphasize that we train just on a small fraction of 3% of the mesh vertices. Still, iSeg is successfully applied to other mesh vertices unseen during training and properly respects the clicked points, as shown in Figs. 2 and 6 and discussed in Sec. 4.2. We note that all the results shown in the paper are for test vertices.

*Unseen views.* Although iSeg was trained with 2D supervision only, its predictions are 3D in nature. Fig. 5 exemplifies this phenomenon. A click on the back side of the backrest segments the front side as well. Such supervision does not exist for our model's training, since the clicked vertex cannot be seen from the front side. Similarly, two clicks at opposite sides of the backrest segment the entire part, although they cannot be seen together from *any*

*single 2D view.* This result suggests that iSeg learned 3D-consistent semantic vertex information, enabling it to generalize beyond its 2D supervision.

*Unseen number of clicks.* For resource efficient training, we trained iSeg on up to two clicks: single click, second positive click, and second negative click. Nonetheless, our model offers customized segmentation with more than two clicks, as demonstrated in Fig. 11. We attribute this capability to the interactive attention mechanism. The interactive attention layer seemed to learn the representation of a positive and a negative click, and how to attend to each click for a meaningful multi-click segmentation.

*Limitations.* Our method may not follow the symmetry of the mesh exactly, as exemplified in Fig. 12. For a click on the goat's head, the segmented regions from the sides of the head differ somewhat from each other, since iSeg is not trained to segment those regions the same. In the supplementary, we discuss the potential limitation of processing 3D shapes using MLPs operating in the Euclidean space.

## 5 Conclusion

In this work, we presented iSeg, a technique for interactively generating fine-grained tailored segmentations of 3D meshes. We opt to lift features from a powerful pre-trained 2D segmentation model onto a 3D mesh, which can be used to create customized user-specified segmentations. Our mesh feature field is general and may be used for additional tasks such as cross-segmentation across shapes of different categories (*e.g.* Fig. 8). Key to our method is an interactive attention mechanism that learns a unified representation for a varied number of positive or negative point clicks. Our 3D-consistent segmentation enables selecting points across occluded surfaces and segmenting meaningful regions directly in 3D (*e.g.* Fig. 5).

In the future, we are interested in exploring additional applications of iSeg beyond segmentation. We have demonstrated that it can potentially be used for cross-domain segmentation, and there may be other exciting applications, such as key-point correspondence, texture transfer, and more.

## Acknowledgments

## References

Ahmed Abdelreheem, Abdelrahman Eldesokey, Maks Ovsjanikov, and Peter Wonka. 2023a. Zero-Shot 3D Shape Correspondence. *SIGGRAPH Asia 2023 Conference Papers* (2023).

Ahmed Abdelreheem, Ivan Skorokhodov, Maks Ovsjanikov, and Peter Wonka. 2023b. SATR: Zero-Shot Semantic Segmentation of 3D Shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. 2005. SCAPE: Shape Completion and Animation of People. In *ACM SIGGRAPH 2005 Papers*. 408–416.

I. Armeni, A. Sax, A. R. Zamir, and S. Savarese. 2017. Joint 2D-3D-Semantic Data for Indoor Scene Understanding. *ArXiv e-prints* (Feb. 2017). arXiv:1702.01105 [cs.CV]

Yuri Y Boykov and M-P Jolly. 2001. Interactive graph cuts for optimal boundary & region segmentation of objects in ND images. In *Proceedings eighth IEEE international conference on computer vision. ICCV 2001*, Vol. 1. IEEE, 105–112.

Jiazhong Cen, Zanwei Zhou, Jiemin Fang, Chen Yang, Wei Shen, Lingxi Xie, Dongsheng Jiang, Xiaopeng Zhang, and Qi Tian. 2023. Segment Anything in 3D with NeRFs. *Advances in Neural Information Processing Systems* 36 (2023), 25971–25990.

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. 2015. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012* (2015).

Xiaokang Chen, Jiaxiang Tang, Diwen Wan, Jingbo Wang, and Gang Zeng. 2023b. Interactive Segment Anything NeRF with Feature Imitation. arXiv:2305.16233 [cs.CV]

Zhimin Chen, Longlong Jing, Yingwei Li, and Bing Li. 2023a. Bridging the Domain Gap: Self-Supervised 3D Scene Understanding with Foundation Models. arXiv:2305.08776 [cs.CV]

Zhiqin Chen, Kangxue Yin, Matthew Fisher, Siddhartha Chaudhuri, and Hao Zhang. 2019. Bae-net: Branched autoencoder for shape co-segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 8490–8499.

Nicu D Cornea, Deborah Silver, and Patrick Min. 2007. Curve-skeleton properties, applications, and algorithms. *IEEE Transactions on visualization and computer graphics* 13, 3 (2007), 530.

Dale Decatur, Itai Lang, Kfir Aberman, and Rana Hanocka. 2024. 3D Paintbrush: Local Stylization of 3D Shapes with Cascaded Score Distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 4473–4483.

Dale Decatur, Itai Lang, and Rana Hanocka. 2023. 3D Highlighter: Localizing Regions on 3D Shapes via Text Descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 20930–20939.

Boyang Deng, Kyle Genova, Soroosh Yazdani, Sofien Bouaziz, Geoffrey Hinton, and Andrea Tagliasacchi. 2020. Cvxnet: Learnable convex decomposition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 31–44.

Tamal K Dey and Wulue Zhao. 2004. Approximating the medial axis from the Voronoi diagram with a convergence guarantee. *Algorithmica* 38, 1 (2004), 179–200.

Yuval Eldar, Michael Lindenbaum, Moshe Porat, and Y. Yehoshua Zeevi. 1997. The Farthest Point Strategy for Progressive Image Sampling. *IEEE Transactions on Image Processing* 6 (1997), 1305–1315.

Zhiwen Fan, Peihao Wang, Yifan Jiang, Xinyu Gong, Dejia Xu, and Zhangyang Wang. 2023. NeRF-SOS: Any-View Self-supervised Object Segmentation on Complex Scenes. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Rahul Goel, Dhawal Sirikonda, Saurabh Saini, and P. J. Narayanan. 2023. Interactive Segmentation of Radiance Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 4201–4211.

Huy Ha and Shuran Song. 2022. Semantic Abstraction: Open-World 3D Scene Understanding from 2D Vision-Language Models. In *Proceedings of the 2022 Conference on Robot Learning*.

Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. 2019. MeshCNN: A Network with an Edge. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 90:1–90:12.

Donald D Hoffman and Whitman A Richards. 1984. Parts of recognition. *Cognition* 18, 1-3 (1984), 65–96.

Yining Hong, Yilun Du, Chunru Lin, Josh Tenenbaum, and Chuang Gan. 2022. 3D Concept Grounding on Neural Fields. In *Annual Conference on Neural Information Processing Systems*.

Shi-Min Hu, Zheng-Ning Liu, Meng-Hao Guo, Junxiong Cai, Jiahui Huang, Tai-Jiang Mu, and Ralph R. Martin. 2022. Subdivision-based Mesh Convolution Networks. *ACM Trans. Graph.* 41, 3 (2022), 25:1–25:16. https://doi.org/10.1145/3506694

Evangelos Kalogerakis, Melinos Averkiou, Subhransu Maji, and Siddhartha Chaudhuri. 2017. 3D Shape Segmentation With Projective Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3779–3788.

Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. 2023. LERF: Language Embedded Radiance Fields. arXiv:2303.09553 [cs.CV]

Hyunjin Kim and Minhyuk Sung. 2024. PartSTAD: 2D-to-3D Part Segmentation Task Adaptation.

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. 2023. Segment Anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 4015–4026.

Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. 2022. Decomposing NeRF for Editing via Feature Field Distillation. *Advances in Neural Information Processing Systems* 35 (2022), 23311–23330.

Theodora Kontogianni, Ekin Celikkan, Siyu Tang, and Konrad Schindler. 2023. Interactive object segmentation in 3d point clouds. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2891–2897.

Abhijit Kundu, Xiaoqi Yin, Alireza Fathi, David Ross, Brian Brewington, Thomas Funkhouser, and Caroline Pantofaru. 2020. Virtual Multi-view Fusion for 3D Semantic Segmentation. *arXiv e-prints*, Article arXiv:2007.13138 (July 2020), arXiv:2007.13138 pages. https://doi.org/10.48550/arXiv.2007.13138 arXiv:2007.13138 [cs.CV]

Jyh-Ming Lien and Nancy M Amato. 2007. Approximate convex decomposition of polyhedra. In *Proceedings of the 2007 ACM symposium on Solid and physical modeling*. 121–131.

Nobuyuki Ostu. 1979. A threshold selection method from gray-level histograms. *IEEE Trans SMC* 9 (1979), 62.

Minghua Liu, Yinhao Zhu, Hong Cai, Shizhong Han, Zhan Ling, Fatih Porikli, and Hao Su. 2023. 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 (CVPR)*. 21736–21746.

Simone Melzi, Riccardo Marin, Emanuele Rodolà, Umberto Castellani, Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. 2019. SHREC 2019: Matching Humans with Different Connectivity. In *Eurographics Workshop on 3D Object Retrieval*, Vol. 7. The Eurographics Association, 3.

Francesco Milano, Antonio Loquercio, Antoni Rosinol, Davide Scaramuzza, and Luca Carlone. 2020. Primal-dual mesh convolutional neural networks. *Advances in Neural Information Processing Systems* 33 (2020), 952–963.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 405–421.

Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. 2016. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. arXiv:1606.04797 [cs.CV]

Ashkan Mirzaei, Tristan Aumentado-Armstrong, Konstantinos G. Derpanis, Jonathan Kelly, Marcus A. Brubaker, Igor Gilitschenski, and Alex Levinshtein. 2023. SPInNeRF: Multiview Segmentation and Perceptual Inpainting With Neural Radiance Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 20669–20679.

Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. 2019. PartNet: A Large-Scale Benchmark for Fine-Grained and Hierarchical Part-Level 3D Object Understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 909–918.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In *NIPS-W*.

Songyou Peng, Kyle Genova, Chiyu "Max" Jiang, Andrea Tagliasacchi, Marc Pollefeys, and Thomas Funkhouser. 2023. OpenScene: 3D Scene Understanding With Open Vocabularies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 815–824.

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. arXiv:1612.00593 [cs.CV]

James Matthew Rehg. 2022. Toys4K 3D Object Dataset. https://github.com/rehg-lab/lowshot-shapebias/tree/main/toys4k.

Zhongzheng Ren, Aseem Agarwala, Bryan Russell, Alexander G. Schwing, and Oliver Wang. 2022. Neural Volumetric Object Selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 6133–6142.

Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. 2004. GrabCut - Interactive Foreground Extraction using Iterated Graph Cuts. *ACM Transactions on Graphics (SIGGRAPH)* (August 2004). https://www.microsoft.com/en-us/research/publication/grabcut-interactive-foreground-extraction-using-iterated-graph-cuts/

Ariel Shamir. 2008. A survey on mesh segmentation techniques. *Computer graphics forum* 27, 6 (2008), 1539–1556.

Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov. 2022. Diffusionnet: Discretization agnostic learning on surfaces. *ACM Transactions on Graphics (TOG)* 41, 3 (2022), 1–16.

Mario Sormann, Christopher Zach, and Konrad Karner. 2006. Graph Cut Based Multiple View Segmentation for 3D Reconstruction. In *Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*. 1085–1092. https://doi.org/10.1109/3DPVT.2006.70

Weiwei Sun, Andrea Tagliasacchi, Boyang Deng, Sara Sabour, Soroosh Yazdani, Geoffrey E Hinton, and Kwang Moo Yi. 2021. Canonical Capsules: Self-Supervised Capsules in Canonical Pose. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., 24993–25005. https://proceedings.neurips.cc/paper/2021/file/d1ee59e20ad01cedc15f5118a7626099-Paper.pdf

Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi. 2022. Neural Feature Fusion Fields: 3D Distillation of Self-Supervised 2D Image Representations. In *Proceedings of the International Conference on 3D Vision (3DV)*.

TurboSquid. 2021. TurboSquid 3D Model Repository. https://www.turbosquid.com/.

Ardian Umam, Cheng-Kun Yang, Min-Hung Chen, Jen-Hui Chuang, and Yen-Yu Lin. 2024. PartDistill: 3D Shape Part Segmentation by Vision-Language Model Distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 3470–3479.

Oliver van Kaick, Andrea Tagliasacchi, Oana Sidi, Hao Zhang, Daniel Cohen-Or, Lior Wolf, and Ghassan Hamarneh. 2011. Prior Knowledge for Part Correspondence. *Computer Graphics Forum* 30, 2 (2011), 553–562. https://doi.org/10.1111/j.1467-8659.2011.01893.x

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *Advances in neural information processing systems* 30 (2017).

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 2015. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1912–1920.

Yunhan Yang, Xiaoyang Wu, Tong He, Hengshuang Zhao, and Xihui Liu. 2023. SAM3D: Segment Anything in 3D Scenes. *arXiv e-prints*, Article arXiv:2306.03908 (June 2023), arXiv:2306.03908 pages. https://doi.org/10.48550/arXiv.2306.03908 arXiv:2306.03908 [cs.CV]

Jianglong Ye, Naiyan Wang, and Xiaolong Wang. 2023. FeatureNeRF: Learning Generalizable NeRFs by Distilling Foundation Models. arXiv:2303.12786 [cs.CV]

Li Yi, Hao Su, Xingwen Guo, and Leonidas J Guibas. 2017. Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. 2282–2290.

Yingda Yin, Yuzheng Liu, Yang Xiao, Daniel Cohen-Or, Jingwei Huang, and Baoquan Chen. 2024. SAI3D: Segment Any Instance in 3D Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 3292–3302.

Haiyang Ying, Yixuan Yin, Jinzhi Zhang, Fan Wang, Tao Yu, Ruqi Huang, and Lu Fang. 2024. OmniSeg3D: Omniversal 3D Segmentation via Hierarchical Contrastive Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 20612–20622.

Yuanwen Yue, Sabarinath Mahadevan, Jonas Schult, Francis Engelmann, Bastian Leibe, Konrad Schindler, and Theodora Kontogianni. 2024. AGILE3D: Attention Guided Interactive Multi-object 3D Segmentation. In *International Conference on Learning Representations (ICLR)*.

Dingyuan Zhang, Dingkang Liang, Hongcheng Yang, Zhikang Zou, Xiaoqing Ye, Zhe Liu, and Xiang Bai. 2023. SAM3D: Zero-Shot 3D Object Detection via Segment Anything Model. *arXiv e-prints*, Article arXiv:2306.02245 (June 2023), arXiv:2306.02245 pages. https://doi.org/10.48550/arXiv.2306.02245 arXiv:2306.02245 [cs.CV]

Renrui Zhang, Liuhui Wang, Yu Qiao, Peng Gao, and Hongsheng Li. 2022. Learning 3D Representations from 2D Pre-trained Models via Image-to-Point Masked Autoencoders. arXiv:2212.06785 [cs.CV]

Qian Zheng, Zhuming Hao, Hui Huang, Kai Xu, Hao Zhang, Daniel Cohen-Or, and Baoquan Chen. 2015. Skeleton-Intrinsic Symmetrization of Shapes. *Computer Graphics Forum* 34, 2 (2015), 275–286.

Qingnan Zhou and Alec Jacobson. 2016. Thingi10K: A Dataset of 10,000 3D-Printing Models. *arXiv preprint arXiv:1605.04797* (2016).

Chenyang Zhu, Kai Xu, Siddhartha Chaudhuri, Li Yi, Leonidas J Guibas, and Hao Zhang. 2020. AdaCoSeg: Adaptive Shape Co-Segmentation with Group Consistency Loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8543–8552.
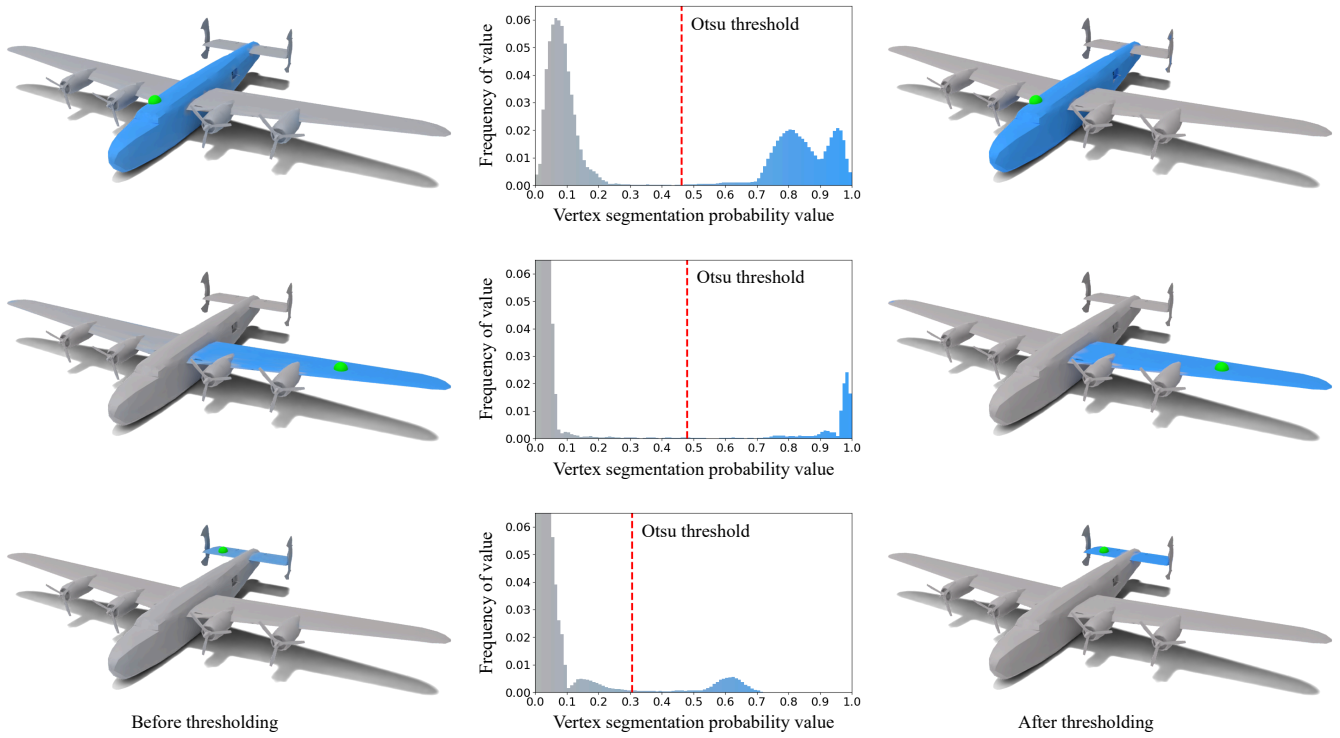
Fig. 9. **Segmentation separability.** The per-vertex segmentation probability can be separated distinctively into low and high value populations, which enables the hard selection of the segmented part by simple thresholding, namely, the Otsu threshold [Ostu 1979].
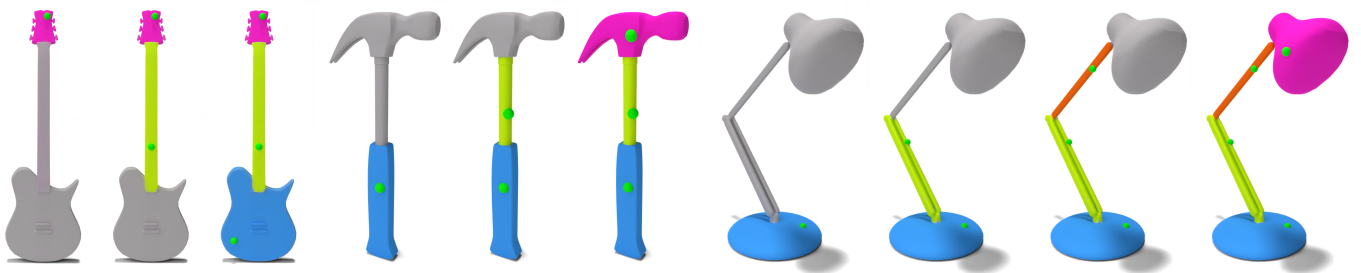


Fig. 10. **Full-shape segmentation.** iSeg can be used to segment the entire shape with a sequence of interactive clicks. We visualize the segmented parts with different colors and the background region is shown in gray.
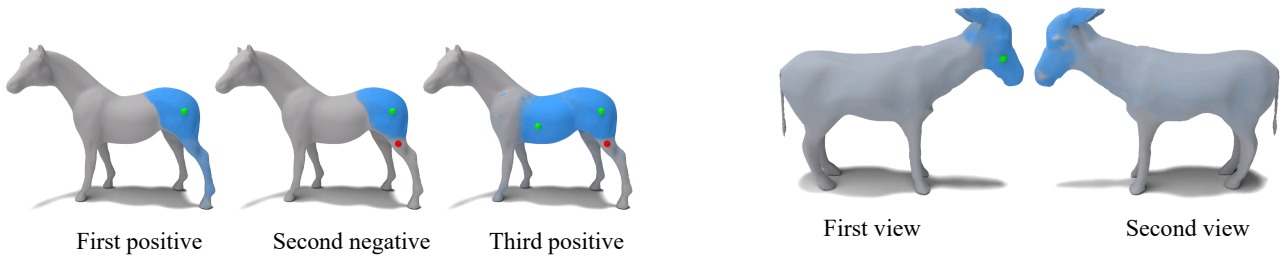


Fig. 11. **Customized segmentations.** iSeg is capable of creating customized segmentations specified by several input clicks.



Fig. 12. **Limitation.** iSeg may not produce a symmetric segmentation result for a symmetric shape. In this case, the segmented region of one side of the shape is different than the other.
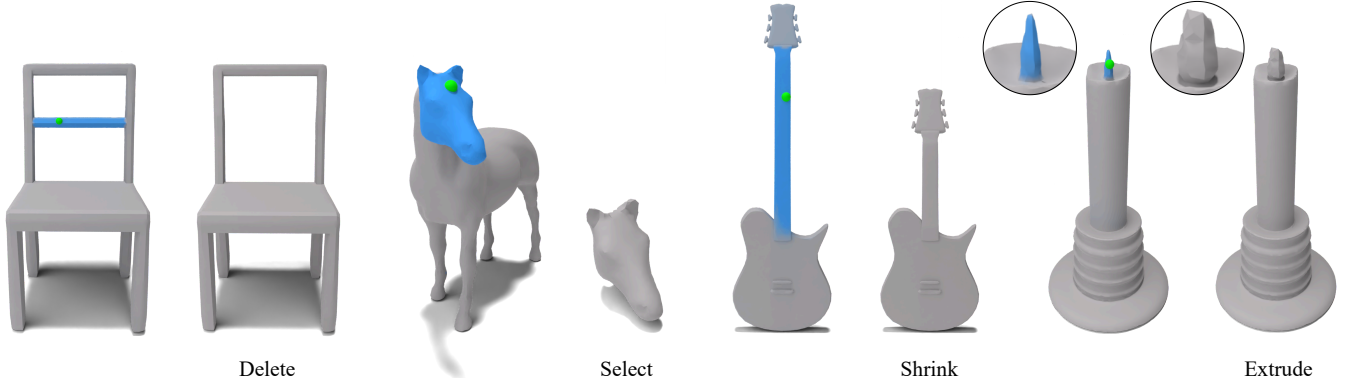
Fig. 13. **Local geometric edits.** Our localized and contiguous segmentations enable various shape edits, such as deleting or selecting the segmented region, shrinking it, or extruding it along the surface normal.
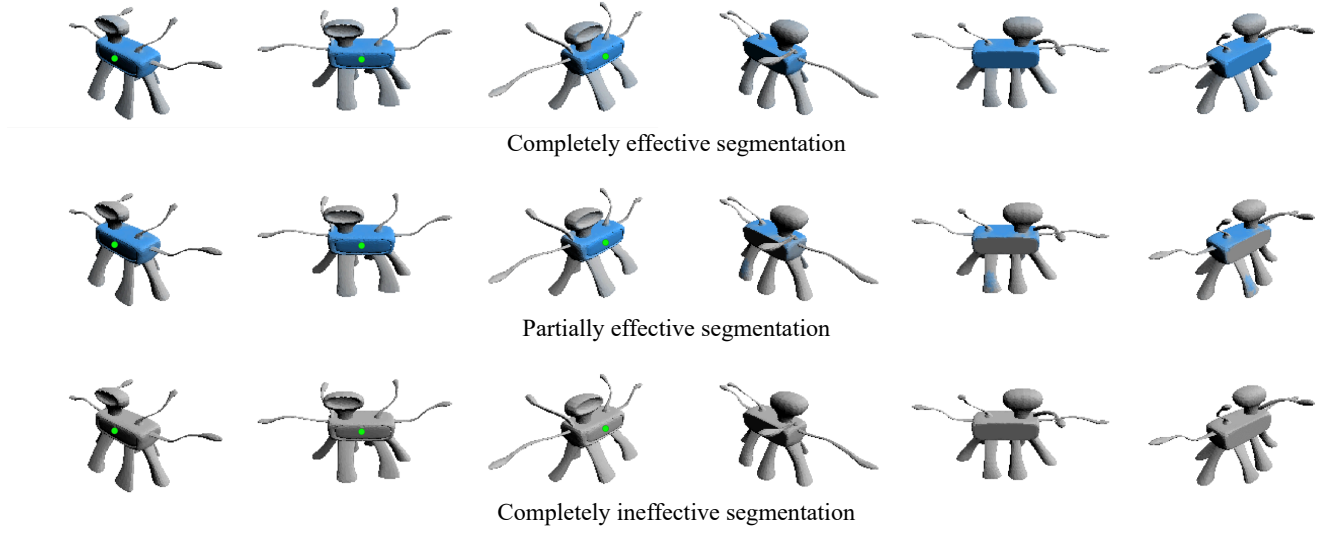


Fig. 14. **Segmentation effectiveness.** We visualize results with a varying level of effectiveness, as presented in our perceptual user study. The segmentations from top to bottom rows are considered completely effective, partially effective, and completely ineffective, respectively.
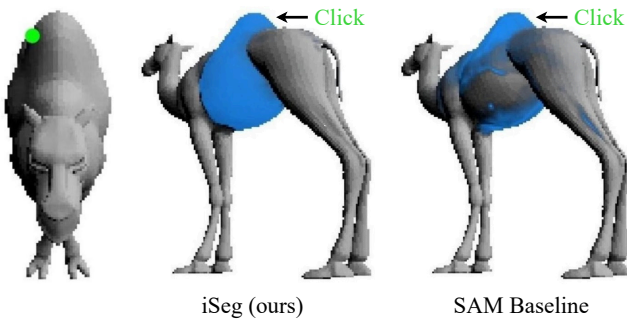


Fig. 15. **The power of iSeg for occluded point click.** When the point click (green dot in the leftmost image) is occluded, iSeg can produce an effective 3D segmentation (highlighted in blue), whereas the SAM baseline is unable to do so.
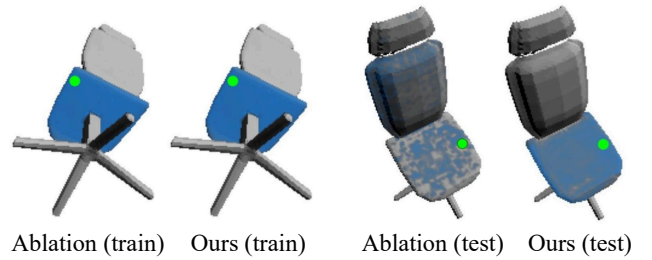
Fig. 16. **Ablation test.** We compare an iSeg model from separate training of the encoder and decoder with an ablation model from joint training of both components. Our proposed separate training scheme results in better generalization for test vertices.