




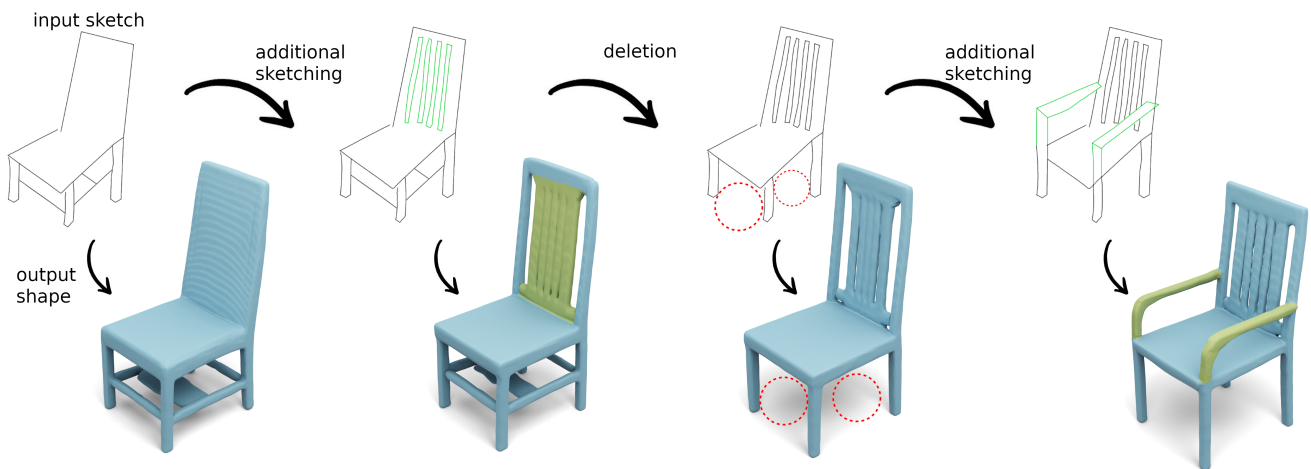


# SENS: Part-Aware Sketch-based Implicit Neural Shape Modeling

Alexandre Binninger<sup>1</sup> , Amir Hertz<sup>2</sup> , Olga Sorkine-Hornung<sup>1</sup> , Daniel Cohen-Or<sup>2</sup> , Raja Giryes<sup>2</sup> <sup>1</sup>ETH Zurich, Switzerland<sup>2</sup>Tel Aviv University, Israel

**Figure 1:** SENS generates shapes and enables ongoing edits via sketching. Adding details or removing parts from the sketch is reflected in the output shape.

## Abstract

We present SENS, a novel method for generating and editing 3D models from hand-drawn sketches, including those of abstract nature. Our method allows users to quickly and easily sketch a shape, and then maps the sketch into the latent space of a part-aware neural implicit shape architecture. SENS analyzes the sketch and encodes its parts into ViT patch encoding, subsequently feeding them into a transformer decoder that converts them to shape embeddings suitable for editing 3D neural implicit shapes. SENS provides intuitive sketch-based generation and editing, and also succeeds in capturing the intent of the user's sketch to generate a variety of novel and expressive 3D shapes, even from abstract and imprecise sketches. Additionally, SENS supports refinement via part reconstruction, allowing for nuanced adjustments and artifact removal. It also offers part-based modeling capabilities, enabling the combination of features from multiple sketches to create more complex and customized 3D shapes. We demonstrate the effectiveness of our model compared to the state-of-the-art using objective metric evaluation criteria and a user study, both indicating strong performance on sketches with a medium level of abstraction. Furthermore, we showcase our method's intuitive sketch-based shape editing capabilities, and validate it through a usability study.

## CCS Concepts

• **Computing methodologies** → Volumetric models; Neural networks;

## 1. Introduction

Data-driven techniques have become the de facto state-of-the-art for recovering a shape from a partial representation in computer graphics. Training neural networks can leverage prior domain knowledge of the data to deal with the innate ambiguity of the input. Neu-

ral implicit fields are currently widely used as a generative model because of their ability to represent arbitrary shapes at arbitrary resolutions [CZ19, PFS\*19, AHY\*19, OELS\*22, TTM\*22]. However, generative models either allow one to randomly sample from the latent space or interpolate between known latent representations,

and hence offer only very limited control over the output shape, which hinders creativity. Thus, editing implicit representations for creative processes is not straightforward [HPG\*22, HASB20].

In this paper, we approach the generation and editing of neural implicit shapes based on free-form sketching. Sketching is an intuitive and effective way to visually communicate shape information. Moreover, sketch-based modeling and editing can be particularly impactful in fields such as architecture, game development and product design, where 3D models are an essential part of the workflow. Despite vigorous efforts in sketch-based 3D modeling, it remains a challenging problem: First, the reconstruction of a 3D shape from an image is inherently ill-posed, since a raw image without annotation is generally a representation of a 3D object merely from a single viewpoint. Second, sketches can vary significantly in style and abstraction level, ranging from fast, casual or even sloppy styles to professional, rigorous sketches. In this paper, we define *abstract sketches* as hand-drawn representations that may lack geometric accuracy and focus more on capturing the essence or key features of the intended 3D shape rather than its exact specifications. When assuming near-perfect correspondence between the sketched silhouettes or other shape features and the output shape, high quality results can be achieved, see e.g. [LGK\*17, LPL\*18, DSC\*20, ZLY\*23]. Similarly, exceptional 3D results can be extracted from high quality input technical drawings that include 3D clues, such as hidden lines [LPBM20] or symmetric strokes [HGSB22]. However, designing a sketch-based 3D modeling system that is agnostic to the level of sketch abstraction of the input and the personal style of the user, accommodating inexact or unskilled drawings, is challenging.

Aside from using sketches to retrieve scenes for modeling [ERB\*12], data-driven generating techniques have always been susceptible to being mere retrievals of the datasets [TRR\*19, SSG\*22]. Providing guarantees that shape-generating systems create novel shapes is thus imperative. We therefore approach the problem using a part-aware generative model to avoid this retrieval pitfall. Part-aware modeling can mitigate the issue, since the generation first detaches the different parts, before assembling the whole shape coherently. This motivates us to use SPAGHETTI [HPG\*22], a part-aware neural implicit shape representation model, as our backbone.

We present SENS, a method that leverages part-aware neural implicit representation to output novel shapes out of an input sketch. Our framework decomposes the input sketch into patches that are fed into a Vision Transformer [DBK\*20]. A transformer decoder then outputs the latent code into the latent space used by SPAGHETTI [HPG\*22]. Using this space, editing can be applied to specific isolated parts of the shapes. For example, the user can manually select a part of the generated shape, such as the back of a chair, and redraw it by restricting the modification to the selected part only. Furthermore, our method offers the ability to systematically replace selected parts of a generated shape, providing an effective means of refining the model and removing any undesired artifacts. SENS also offers the possibility to outline the obtained shape while modeling, granting the user the possibility to modify the sketch directly and lowering the sketching skill cap.

We compare SENS with state-of-the-art sketch-to-shape techniques, encompassing both empirical and quantitative analyses. To illustrate that our method goes beyond simple shape retrieval, we

**Table 1:** Comparison of sketch-based shape generation methods.

	Single view	Editing	Abstract sketches
ShapeMVD [LGK*17]	✗	✗	✗
Pixel2Mesh [WZL*18]	✓	✗	✗
ProSketch [ZQG*21]	✗	✗	✗
Sketch2Mesh [GRYF21]	✓	✓	✗
DeepSketch [ZGZS22]	✓	✗	✗
Ours	✓	✓	✓

present the top-4 shapes retrieved from the shapes generated by our approach. We further validate the quality of SENS's generation ability via a comparative perceptual user study. We also showcase the editing possibilities of our method in an interactive environment. Our key contributions are:

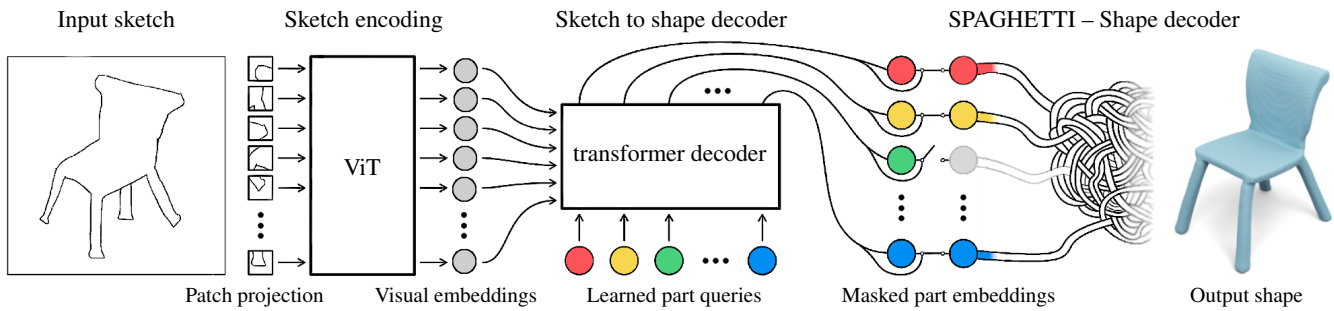
- Sketch-based modeling based on single-view sketches of diverse levels of abstraction.
- State-of-the-art results for shape generation with limited retrieval.
- New editing capabilities that allow for part-based shape refinement and localized sketch-based reshaping and combinations.

## 2. Related work

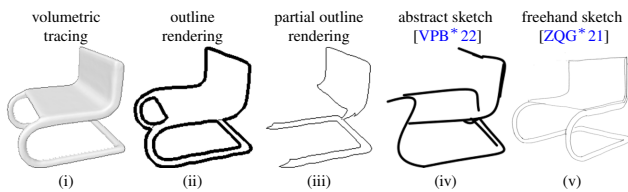
*Sketch-based modeling.* Sketch-based modeling was extensively researched before the recent burst of data-driven techniques. As we focus on the latter, we only present a fragment of this domain and refer the reader to [CIW08, BAC\*19] for a more complete survey. Teddy [IMT99] was one of the first modeling systems introduced for casual modeling, and has inspired many works since [TZF04, NISA07, SS08, BPCB08, GIZ09, DSC\*20, ZYC\*22]. Some methods offer sketch-based creation by targeting a specific class of shapes, such as garment modeling systems [TCH04, FRH\*21]. Virtual reality provides an environment in which sketches are three-dimensional, resolving partial ambiguities for shape modeling [VSH19, YAS\*21, YAB\*22]. Using inputs with additional information such as concept sketches [GHL\*20, HGSB22] or manual annotations [XCS\*14] can facilitate reconstruction but requires higher sketching skills.

*Neural networks shape representation types.* The rise of deep learning for 3D geometry inspired the use of many shape representations. Explicit representations are popular for their expressiveness and editing possibilities. However, mesh representations require using graph neural networks [HHF\*18, WZL\*18, FFY\*19], which are computationally harder to process due to the inherent lack of regularity. Parametric representations offer mathematical accuracy but are hard to acquire and often rely on other representations for learning, such as meshes [PUG19], point clouds [SLK\*20] or distance fields [SFK\*20]. Voxel representations leverage the regularity of the grid to ease the design of effective networks [ZZZ\*18, WZZ\*18], but they are resolution dependent and lead to poor representations of details. Point clouds are easy to acquire and process but do not embed geometrical structures [FSG16, YHCOZ18, YHH\*19]. We refer the reader to [MKKv22] for a comprehensive survey on neural shape representations.

*Neural implicit shape generation and modeling.* Neural implicit representations emerged as an alternative representation.



**Figure 2:** SENS takes as input a  $256 \times 256$  normalized grayscale sketch. It is partitioned into  $16 \times 16$  patches, and then passed through a Vision Transformer. A transformer decoder is then used to generate the latent variable  $z \in \mathbb{R}^{m \times d_{model}}$ , which is a part-aware latent space with  $m$  parts represented by latent vectors of dimension  $d_{model}$  that conditions the neural implicit representation given by SPAGHETTI, which is used to generate the output shape. By the part-aware latent space we get a mapping between sketch and shape parts.



**Figure 3:** We used a variety of sketch styles as inputs to our method. The target shape is an implicit shape rendered via volume rendering, outline rendering and partial outline rendering. The abstract sketch is produced using CLIPasso [VPB\*22] on the volume rendering. Expert freehand sketches come from ProSketch [ZQG\*21].

DeepSDF learns the truncated sign distance function [PFS\*19], while other methods are based on a binary outside/inside classification [MON\*19, CZ19]. In the realm of occupancy networks, some approaches have been developed to learn latent codes on regular grids [PNM\*20]. However, recent advancements propose employing irregular grids for latent vector distribution [ZNW22], or even utilizing sets of latent vectors [ZTNW23]. Since implicit representations are level-sets of a function, they are often restricted to closed meshes, though this can be avoided by learning unsigned distance functions [CMPM20, GSF22]. They are also restricted to watertight surfaces and are difficult to modify directly. To solve these issues, mixed representations have also emerged. Deepcurves handles boundaries using an explicit representation [PSW\*21]. Since implicit representations are hard to edit, DualSDF [HASB20] proposes a combined explicit representation that the user can edit. SPAGHETTI is a part-aware generative network [HPG\*22] which relies on Gaussian mixture models to represent each part of the shape and provides editing via an affine transform of each Gaussian cluster. Part-aware representations can also help avoid the caveat of falling into mere retrieval [SKZC18, SSG\*22], which is a property we use in this work.

*Neural sketch-to-mesh methods.* Using neural networks for sketch-based modeling is an active area of research in computer graphics, and there has been notable progress in recent years in developing neural network-based approaches for generating 3D models from 2D sketches. ShapeMVD [LGK\*17] and SketchCNN [LPL\*18] reconstruct shapes from 2D sketches using a convolutional neural network, but require multiple views and do not support abstract sketches.

ProSketch [ZQG\*21] and DeepSketch [ZGZS22] are trained on a mix of synthetic and professional sketches. Some view-aware modeling systems exist: Sketch2Mesh [GRYF21] proposes an encoder/decoder architecture to reconstruct 3D shapes that can be refined via a user interface; Garment Ideation [CWC\*22] is a feature aggregation-based iterative method targeted towards garment ideation that predicts a winding number to generate 3D shapes; concurrently to our work, LAS-Diffusion [ZPW\*23] proposes a multi-class diffusion method based on an attention mechanism and GA-Sketching [ZLY\*23] proposes a multi-view method with modeling options via iterative refinement, but they fall short in effectively processing abstract sketches. Edit3D [CCR\*22] employs a unified latent space to generate 3D shapes, sketches, and RGB images, thereby establishing a correspondence between these three types of representations that enables shape and color editing. Delanoy et al. [DBA\*17] propose a method to recover a volumetric shape from an input sketch. Parametric representations are also used for sketches [SBS19]. Sketch2CAD [LPBM20] is based on the generation of primitives, Free2CAD [LPBM22] decomposes an input sketch into a sequence of strokes that are mapped to a sequence of CAD instructions, and GeoCode [PLH\*22] offers sketch-based modeling of parametric shapes with additional part-aware control of the relevant parameters. Note that neural methods can also recover shapes from non-sketch images. Pixel2Mesh [WZL\*18] recovers a mesh from an image while 3D-R2N2 [CXG\*16] and NeRFs [MST\*20] can reconstruct a shape from multiple views. SKED [MPS\*23] is a NeRF-based method which provides a sketch-guided text-based shape editing method. Table 1 presents the strengths and weaknesses of the works most related to ours.

### 3. Method

SENS generates a neural implicit shape from a *single-view* input sketch. More specifically, it associates to a sketch a latent code that can be interpreted by a neural implicit shape decoder. To this end, we design a neural network that learns to match a sketch to its corresponding shape’s latent code in the latent space of SPAGHETTI [HPG\*22]. SPAGHETTI is designed to convert a latent vector into a collection of  $m$  Gaussians, where each Gaussian represents a part of the object. Subsequently, each part goes through a “mixing network”, a transformer encoder that ensures global consistency

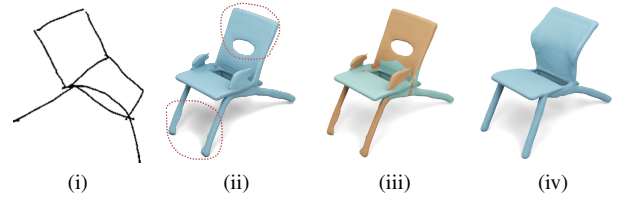
across the shape. An “occupancy network” follows for decoding the final shape, as it returns the signed distance function from a query point. The main property of SPAGHETTI that we use lies in the fact that it is a *part-aware* implicit shape decoder, which means that its latent space is divided in several parts, and each part of the latent space encodes for a corresponding part in the resulting shape. This feature enables to train our network on partial inputs to mitigate shape retrieval, to train a refinement network that can regenerate selected latent parts, and to restrict the shape generation to specific parts during the modeling process.

### 3.1. Data generation and input normalization

To improve our network robustness with respect to the style and the level of abstraction of the input, we use a dataset with a variation of designs. Our dataset is based on a subset of the ShapeNet dataset [CFG\*15]: the chair dataset with 6755 shapes, the lamp dataset with 833 shapes, and the airplane dataset with 1775 shapes. Each was rendered with six different views in three different manners: (i) volumetric rendering that relies on ray marching; (ii) outline rendering based on the depth map; and (iii) partial outline rendering, which are renders of SPAGHETTI’s shapes after masking out parts of their latent code. In addition, (iv) abstract sketches of eight strokes were computed based on each view of the volumetric renderings by using CLIPasso [VPB\*22] with 2000 iterations. For chairs, we used an additional dataset, (v) ProSketch [ZQG\*21], to add free-hand sketches drawn by experts. We display examples from our sketch dataset in Fig. 3. The data is augmented with random perspective transformation and horizontal symmetry. Using the fact that CLIPasso provides vector graphics outputs, we applied data augmentation to its abstract sketches by modifying the stroke width before rendering it as an image. We normalize the input by centering the sketch, cropping the empty borders and resizing it to a  $256 \times 256$  image. Partial outline renderings are normalized and cropped in alignment with their respective full renders.

### 3.2. Sketch-to-latent representation

Our network maps a sketch to the latent representation of a neural implicit shape generator, namely SPAGHETTI [HPG\*22]. SPAGHETTI receives as input a latent representation that is mapped to a collection of  $m$  vectors of dimension  $d_{\text{model}}$  that represents a Gaussian mixture model (GMM), i.e. each of these  $m$  vectors corresponds to a 3D Gaussian. SPAGHETTI outputs a 3D implicit shape by mapping each Gaussian to a part of the represented shape, and mixes these parts to produce a globally coherent shape. In this work, we make use of this intermediate GMM-based latent space and map the input sketch directly to it. For each given shape, we precompute its collection of latent vectors  $\{z_i\}_{i=1}^m$  using shape inversion [HPG\*22]. An overview of our network architecture is displayed in Fig. 2. Inspired by the DETR object detection model [CMS\*20], our network is composed of an image encoder that takes an input sketch and outputs visual embeddings. A transformer decoder maps a set of learned part queries together with these visual embeddings to SPAGHETTI’s multi-part latent space. The image encoder (Fig. 2 left) is a Vision Transformer network [DBK\*20]. It divides  $256 \times 256$  sketch images into  $16 \times 16$  patches. Each patch is mapped to a single visual embedding via a transformer encoder.



**Figure 4:** A sketch of poor quality (i) may yield inadequate results (ii). Users can select unsatisfactory parts of the output (ii, lasso selection on shape in red; iii, selected parts in orange). Our refinement network can predict a refined shape (iv) by regenerating the selected parts of the latent space based on the non-selected parts.

The transformer decoder (Fig. 2 middle) takes as input these visual embeddings and a set of  $m$  part queries, and processes them using its self-attention and cross-attention layers. The part queries are *learnable* vectors, i.e. they are optimized at the same time as the network. Finally, each output vector of the decoder is mapped to a latent part vector  $\{\tilde{z}_i\}_{i=1}^m$  of the neural implicit shape decoder, SPAGHETTI, which uses them to generate the output shape (Fig. 2 right). The training loss we use is

$$\mathcal{L}_{\text{full}} = \frac{1}{m} \sum_{i=1}^m \|\tilde{z}_i - z_i\|_1,$$

where  $z_i$  is the ground truth  $i$ th part vector of the 3D shape that corresponds to the input sketch, and  $\tilde{z}_i$  is the prediction of SENS.

### 3.3. Partial shape

SENS is trained to perform reconstruction also by additional outline renders of *partial* 3D shapes. The goal is to reinforce the uncoupling between parts of the restored shape as demonstrated in Sec. 4.6.

The *partial outline rendering* supervision for this task is obtained by randomly selecting a subset of part vectors  $\{z_i[c_i]\}_{i=1}^m$  where the binary assignment  $c_i$  indicates the presence of part  $i$  in the subset. Then the subset of vectors is given to SPAGHETTI which generates the corresponding partial 3D implicit shape. Finally, we render the partial shape. See Fig. 3(iii) for an example.

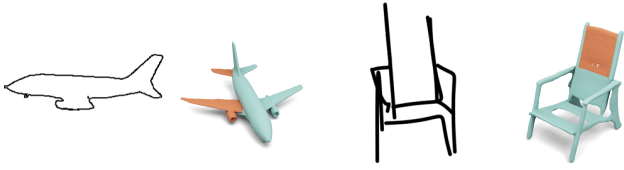
When feeding partial outline renders into SENS, we use different loss functions. In this case, the output of the transformer decoder is passed through an MLP to an additional classification score  $\tilde{c}_i \in [0, 1]$  which indicates the presence of part  $i$  in the input outline render. We optimize it by the binary cross entropy loss,

$$\mathcal{L}_{\text{cls}} = \frac{1}{m} \sum_{i=1}^m \text{BCE}(\tilde{c}_i, c_i),$$

where  $c_i$  is the ground truth indicator of part  $i$  in the input render. Moreover, the loss for the latent vector prediction of our network is

$$\mathcal{L}_{\text{part}} = \frac{1}{\|\mathbf{c}\|_0} \sum_{i=1}^m c_i \|\tilde{z}_i - z_i\|_1,$$

where  $c_i$  are used to ignore latents of parts not present at the input and the normalizing factor  $\|\mathbf{c}\|_0$  counts the number of non-zero entries in  $\mathbf{c} = [c_1, \dots, c_m]$ .



**Figure 5:** We exemplify how our network performs shape completion from single-view sketches. If input sketch does not display the full shape, the network is still able to reconstruct it, notably taking advantage of the symmetry of the class of shapes in the dataset.

### 3.4. Refinement network

The refinement network allows to regenerate parts of a given shape, and is illustrated in Fig. 4. In some cases, poor quality or ambiguities in the input sketch (i) may lead to artifacts in the generated shape (ii). The user can select unsatisfactory parts from the output shape (iii, marked in orange). A selected part on the shape has a corresponding latent vector part in the GMM latent space. Our refinement network, which is conditioned on the latent vector parts of the non-selected parts, outputs a set of vectors parts that replace the selected ones. Finally, the shape decoder regenerates the refined shape using the new latent vector parts (iv).

The refinement network is a bidirectional transformer encoder network that receives the set of latent vectors  $\tilde{z} \in \mathbb{R}^{m \times d_{\text{model}}}$  such that the corresponding vectors of the selected parts are masked (i.e., zeroed). It outputs  $\hat{z} \in \mathbb{R}^{m \times d_{\text{model}}}$ , which contains the refined vectors in the entries corresponding to the selected parts.

The network uses a masking objective [DCLT18], where 5 – 40% of the input vectors are masked, and the network has to predict their content based on the unmasked context. The loss is

$$\mathcal{L}_{\text{refine}} = \frac{1}{\|\mathbb{1}\|_1} \sum_{i=1}^m \mathbb{1}_i \|\hat{z}_i - z_i\|_1,$$

where the indicator  $\mathbb{1}_i$  equals one if and only if the input vector  $\tilde{z}_i$  was masked and  $\|\mathbb{1}\|_1 = \sum_{i=1}^m \mathbb{1}_i$  is a normalizing factor.

## 4. Results

We show our shape generation and editing results, with quantitative evaluation and insights into retrieval, completion, ablation, and limitations. We trained two single-class SENS networks over chairs and airplanes and a multi-class network that was trained jointly over chairs, airplanes and lamps. The latent space of the pre-trained SPAGHETTI model consists of  $m = 16$  and  $m = 32$  parts with dimensions  $d_{\text{model}} = 512$  and  $d_{\text{model}} = 768$  for the single and multi-class networks respectively. We will publish our sketches dataset, code, pre-trained models and user interface upon acceptance.

### 4.1. Generation comparison

SENS can generate a shape from a single input sketch. As we trained our neural network on a combination of outline renderings, abstract sketches and expert freehand sketches (Fig. 3), we are able to produce sensible outputs from sketches of diverse styles. Fig. 13, Fig. 14, and our supplementary material show some examples of our sketch-based generation.

We compare SENS in Fig. 6 with three single-view image to shape methods, namely Pixel2Mesh [WZL\*18], Sketch2Mesh [GRYF21] and DeepSketch [ZGZS22]. Pixel2Mesh is a generic, non-sketch-specific, image-to-shape method. Though able to reconstruct a shape that maps the outline of the input, the result is less aesthetically pleasing. While DeepSketch and Sketch2Mesh are targeted towards sketch-to-shape applications, their methods struggle to produce reasonable output from abstract sketches. DeepSketch is trained on synthetic shapes [ZGZS20] and expert freehand sketches [ZQG\*21], and even though Sketch2Mesh is trained on several sketch styles, the external contours of the input sketches remain the same. We do not use its refinement because it requires additional camera view parameters.

We also compare with ShapeMVD [LGK\*17], a multi-view reconstruction method in Fig. 7, using input sketches from their own test dataset. The inputs to ShapeMVD are two orthogonal views that are precisely aligned. Their method predicts the depth map and normal map to output a point cloud from which a mesh is extracted using screened Poisson Surface Reconstruction [KH13]. Because the additional view reduces the ambiguity, their method is able to generate shapes that are more accurate to the input, but which seem to be more prone to artefacts. We noticed that ShapeMVD failed at shape generation from abstract sketches, thereby raising the level of skill required to use it.

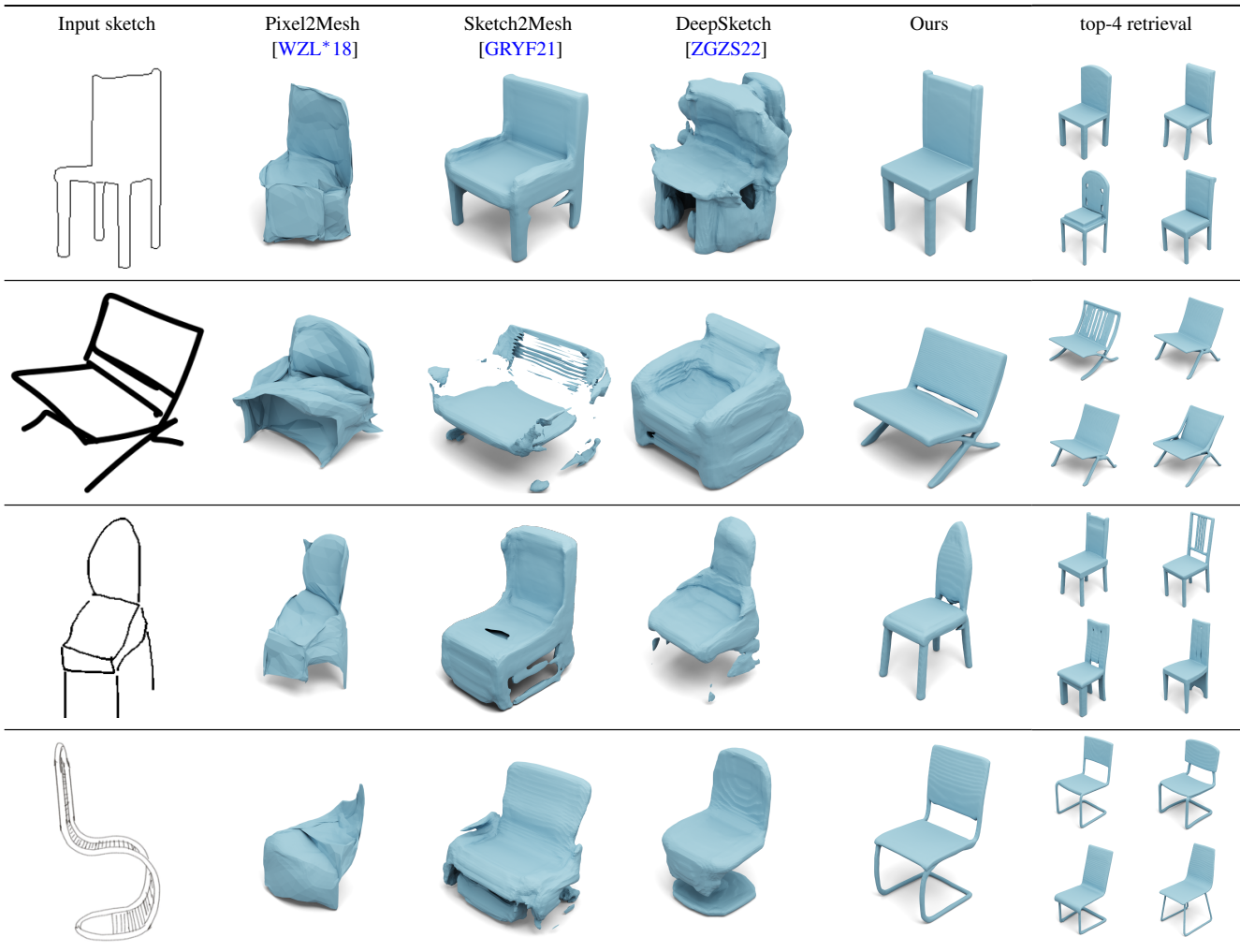
### 4.2. Evaluation

For an objective evaluation, we ran Pixel2Mesh, Sketch2Mesh, DeepSketch and SENS on the AmateurSketch dataset [QGS\*21], which contains 3000 freehand sketches of ShapeNet chairs of medium abstraction level. We then computed the chamfer distance (CD), the Earth Mover’s distance (EMD) and the shading-image-based Fréchet Inception distance (FID) [HRU\*18, PZZ22, ZLWT22]. Our results are reported in Table 2, and we refer the reader to our supplementary material for more details about the used metrics. Note that SENS performs better in all the metrics referenced here.

As an additional perceptual evaluation, we conducted a user study. We randomly sampled 24 sketches from the AmateurSketch [QGS\*21] dataset on which we applied the methods we compare with. Users were asked to rank the four chairs for how realistic and how similar to the input sketch they are. Table 3 shows the results for both questions in separate columns. 54 people took part in our user study. Note that SENS consistently ranks highest both in terms of realism and similarity. More details are to be found in the supplementary material.

### 4.3. Shape completion

As explained in Sec. 3.3, our network predicts latent codes  $\tilde{z} \in \mathbb{R}^{m \times d_{\text{model}}}$  and a continuous score  $\tilde{c} \in [0, 1]^m$ , where  $\tilde{c}_i$  indicates the probability that the  $i$ th component of  $\tilde{z}$  is represented in the sketch. While the use of partial outline rendering allows our training to disentangle the different parts of the input sketch, the prediction of the mask  $\tilde{c}$  is useful to determine the confidence of the network in the reconstruction of each part. Because SENS reconstructs a shape from a single viewpoint, it often has to reconstruct parts of the shape that are not depicted in the input sketch. We show in Fig.



**Figure 6:** We compare our method with state-of-the-art single-view reconstruction methods on sketches of various styles such as an outline, an abstract sketch, a non-expert handmade sketch and an expert freehand sketch from the ProSketch dataset [ZQG\*21]. Note that these sketches were not part of the dataset. We also show the top-4 retrieval: we first remesh the output shapes of SPAGHETTI [HSG18] that were used for training SENS, and compute the Chamfer Distance over 100,000 sampled points over the surface. We display the output shape of SPAGHETTI. The order is left to right, top to bottom.

5 several examples of completion. The part  $i$  of a shape is said to be *completed* if the mask probability  $c_i$  is below a certain threshold, here set to 0.01. Completed parts are displayed in orange.

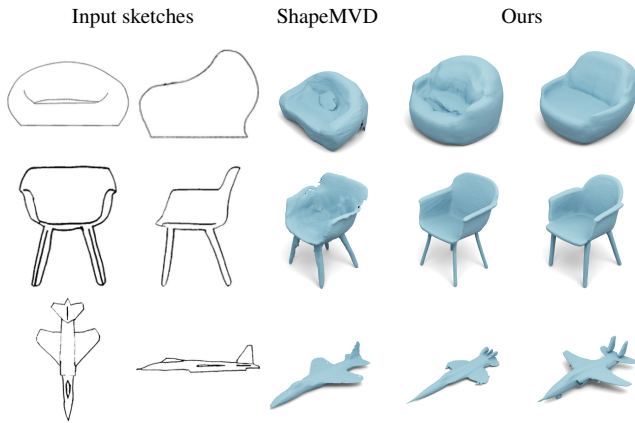
#### 4.4. Shape retrieval

It is crucial for shape-generation techniques to address the retrieval problem. This means that a method should be able to generate a desired shape based on a given sketch, and not just retrieve a shape from the training dataset that approximates a reasonable result. In Fig. 6, we provide evidence that SENS does not merely retrieve shapes. The main enabler for this is the part-aware property of SENS as it is trained to produce disentangled part vectors that are combined to generate the whole shape. For instance, while the first and second output shapes share similar legs as their respective first retrievals, they exhibit significant differences in the back area. The

rounded back of the third chair is not present in the top-4 shape retrieval results. While the fourth shape has an identical structure to its top retrieval, the back, seat, and legs' lengths vary.

#### 4.5. Editing

The ability to generate 3D shapes from sketches can simplify 3D modeling. Yet, a user may desire to edit the generated shape, which is a complex task. One major advantage of SENS is the ability to easily edit shapes through sketching (Fig. 1). We implemented a user interface using the Visualization Toolkit (VTK) [SMLK06] featuring a drawing canvas and a viewer that displayed the generated shape after its conversion to a mesh via marching cube [LC87]. We present a live demonstration of the editing possibilities in a video attached to the supplementary material.



**Figure 7:** We compare SENS with ShapeMVD [LGK\*17], a sketch-to-shape method requiring multi-view input sketches. The pairs of input sketches belong to ShapeMVD test set. Since SENS relies on a single-view input, we show the results for both input sketches.

**Table 2:** Performance comparison of shape reconstruction methods on the AmateurSketch dataset [QGS\*21] using chamfer distance (CD), earth mover’s distance (EMD), and Fréchet inception distance (FID). Lower values indicate better performance. Comparison is done with Pixel2Mesh [WZL\*18], Sketch2Mesh [GRYF21], and DeepSketch [ZGZS22]. The supplementary material contains additional comparisons.

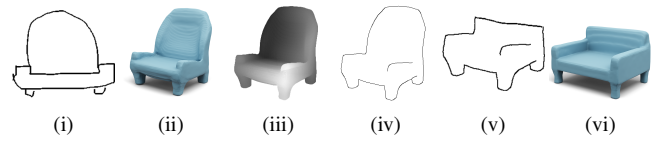
Method	CD↓	EMD↓	FID↓
Pixel2Mesh	0.2191	0.1658	401.7
Sketch2Mesh	0.2113	0.1573	368.4
DeepSketch	0.1520	0.1142	292.2
SENS	<b>0.1186</b>	<b>0.0946</b>	<b>171.3</b>

#### 4.5.1. Outline rendering

Our interface proposes an outline rendering method of the displayed shape, enabling users to perform direct modifications on the drawing canvas. The pipeline is illustrated in Fig. 8: after an initial drawing (i) generates a starting shape (ii), the shape is rendered as a depth map (iii), which is then smoothed via a Gaussian filter. Edges are then extracted using the Canny edge detection method [Can86]. Consequently, the outline aligns with the shape’s orientation on the screen (iv). As a result, our interface allows users to first create an

**Table 3:** Perceptual evaluation through a user study, highlighting the performance of our method in comparison to Pixel2Mesh [WZL\*18], Sketch2Mesh [GRYF21], and retrained DeepSketch [ZGZS22] in terms of realism and similarity to input sketches (1 is best rank).

Question Rank	Realistic				Similar			
	1	2	3	4	1	2	3	4
Pixel2Mesh	0.1	1.1	13.1	<b>85.7</b>	0.6	16.0	24.5	<b>59.0</b>
Sketch2Mesh	10.5	<b>52.0</b>	34.1	3.4	1.8	28.6	<b>44.8</b>	24.8
DeepSketch	2.0	37.4	<b>49.9</b>	10.6	4.1	<b>49.5</b>	30.3	16.1
SENS	<b>87.4</b>	9.5	2.9	0.2	<b>93.5</b>	5.9	0.5	0.1



**Figure 8:** Our outline rendering pipeline. An initial drawing (i) serves for shape generation (ii). We render its depth map (iii), which is in turn used for edge extraction (iv). The outline can be modified (v) and used as an input for further shape generation (vi).

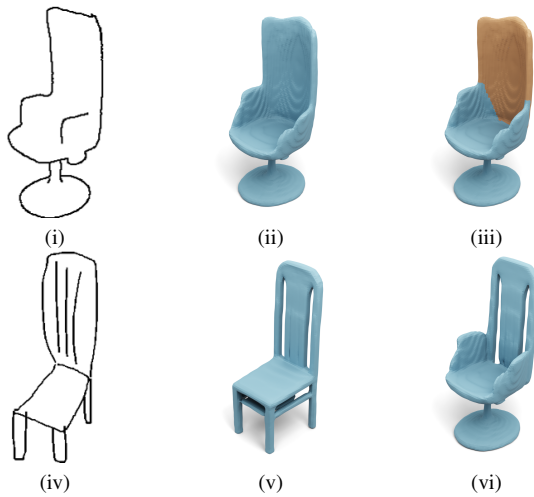
abstract sketch of a chair, generate its outline, and then directly edit the outline (v) for further shape generation (vi). This simplification of the 3D modeling process greatly reduces the demand for advanced sketching skills.

#### 4.5.2. Refinement via part reconstruction

Because SPAGHETTI is a part-aware shape decoder, it is possible to select parts of the latent code and use a refinement network to regenerate them based on the unselected parts, as described in Sec. 3.4. The selection is illustrated in Fig. 4 and operates as follows: first, the user employs a freehand lasso selection on the screen (ii). Then, our interface detects which faces of the mesh are picked by the lasso selection. The parts of the latent code that encode for the generation of these picked faces are then labeled as "selected". Once a part is selected, we display in orange all the faces that are generated by this part (iii), not only the originally picked faces. Our interface will mask the selected parts and feeds the latent code to the refinement network, which generates new parts of the latent code to replace the selected one (iv). While the refinement network was initially trained to reconstruct 5% – 40% of masked latent vectors, there are no practical constraints on the number of vector components that can be masked for refinement. The refinement strategy can be particularly useful for removing artifacts from the generated shape, as exemplified in Fig. 4 and in the supplementary video.

#### 4.5.3. Part-based modeling

The use of a part-aware shape decoder also enables local modifications to the generated shapes. Indeed, SENS accepts a sketch as input and produces a corresponding latent code that can be broken down into several parts. However, these latent parts can originate from different input sketches, hence allowing the fusion of features from distinct shapes. We provide an illustration of part-based modeling in Fig. 9. The initial drawing (i) generates a latent code that, when decoded, yields a shape (ii). The user can select parts of the latent code, illustrated in orange on the output shape (iii). Drawing another sketch (iv) generates a new latent code, that if decoded by SPAGHETTI, would yield a completely different shape (v). Instead of replacing the entire latent code, only the selected parts are replaced, hence producing a new shape that blends features from both original shapes (vi). In this example, the resulting chair combines the base of the first chair with the backrest of the second chair. This technique represents a substantial improvement over traditional sketch-to-shape methods in significantly extending the modeling flexibility and generation capabilities, going beyond the dataset’s inherent limitations. Note that our part-based modeling method can be used with sketches of different abstraction levels, which strengthens its flexibility.



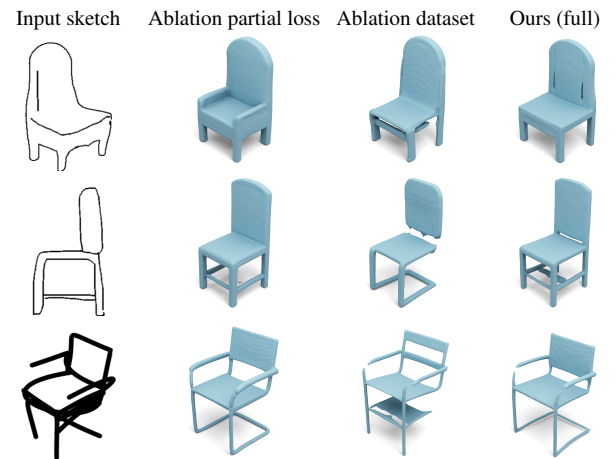
**Figure 9:** Part-based modeling example. The input sketch (i) is fed to our network to generate a shape (ii). The user can select parts of the resulting shape (iii). Given another sketch (iv), SENS would generate a completely different shape (v). But using part-based modeling, our interface will only replace the selected parts (vi).

#### 4.5.4. Evaluation

To evaluate the usability of our method’s editing capabilities, we carried out a user study with 8 participants from diverse backgrounds, possessing varying levels of modeling and sketching expertise. During this session, participants were tasked with two assignments: firstly, creating any chair design, ensuring they utilized all available editing tools to familiarize themselves with our system; and secondly, modeling three distinct shapes based on provided images. After the modeling session, participants completed two questionnaires to gauge the system usability and the efforts required to use it. We show the results in our supplementary material, where we detail the questionnaire outcomes and showcase a range of shapes crafted during the study. Feedback from participants was largely positive; they found the system intuitive and user-friendly, expressing satisfaction with their outputs.

#### 4.6. Ablation studies

To analyze the relevance of different components of SENS, we provide an ablation study. Visual results are displayed in Fig. 10 on inputs presented with increasing levels of abstraction from top to bottom. For quantitative evaluations, refer to Table 4. To provide a fair comparison, no model in our ablation study has been trained on the ProSketch dataset, and all networks were trained for 40 hours. First, we trained the same network by removing the mask loss  $\mathcal{L}_{cls}$  and the partial loss  $\mathcal{L}_{part}$ , both explained in Sec. 3.3 and referred to as “ablation partial loss”. We claim that these losses improve the part disentanglement, hence allowing SENS to produce shapes that are less prone to mere shape retrieval. This is particularly visible in the chairs’ handles that are not present or not connected to the seat in the original drawing. Yet, they are visible in the output shape. The quantitative comparison supports our analysis. The metrics indicate that eliminating the partial loss significantly decreases the distance between the shapes in the dataset and those generated, indicating



**Figure 10:** We present our ablation study on three different input sketch styles, namely a shape outline, a drawing and an abstract sketch. “ablation partial loss” means that the network did not train with partial loss; “ablation dataset” means that the network did not train with abstract sketches.

a tendency towards retrieval. Second, we trained SENS without using abstract sketches, referred to as “ablation dataset”. It clearly appears that the more abstract the input sketch, the more the obtained result decreases in quality, notably with some parts being absent from the output shape. The quantitative metrics further demonstrate that incorporating sketches of varying abstraction levels enhances our method’s adaptability to different input sketch styles. This is evidenced by the weaker performance on our metrics by the version of our method with dataset ablation.

#### 4.7. Multi-class reconstruction

Until now, we had conditioned SENS on a specific class of shapes. We demonstrate here that it is possible to condition SENS on multiple classes at the same time. To account for the greater shape diversity, our multi-class shape generator relies on a higher number of Gaussians and the latent representation has a higher dimension. Fig. 11 compares the results of the multi-class network to the single-class network of the respective category. We observe that the multi-class version produces successfully shapes that correspond to the right category. Compared to the single-class version, the output shapes are slightly less accurate, especially with sharp features. This can

**Table 4:** Performance comparison of ablated methods on the AmateurSketch dataset [QGS\* 21] using chamfer distance (CD), earth mover’s distance (EMD), and Fréchet inception distance (FID). The metrics indicate that our mask loss and partial loss enhance our method’s ability to resist shape retrieval issues, and the use of abstract sketches increases its resilience to sketch abstraction.

Method	CD	EMD	FID
Ablation partial loss	0.117	0.0940	170.6
Ablation dataset	0.1300	0.1023	181.9
Ours (full)	0.1235	0.0981	174.3



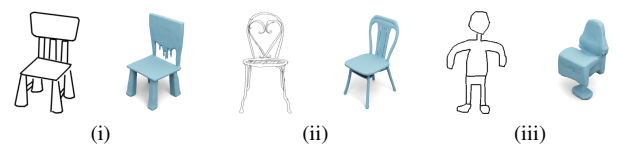


**Figure 11:** We compare our multi-class and single-class sketch-to-shape models. The input of the last column comes from *AmateurSketch*. The other sketches are produced by us.

be observed in the chair and airplane sketches. Note though that for lamps, we get better results with the multi-class network. As lamps have a smaller training set, the multi-class network exhibits better generalization than the single class as it has access to more data.

#### 4.8. Limitations

Single-view sketch-to-shape reconstruction is a challenging problem as it requires overcoming necessary ambiguities. SENS tackles this by conditioning the network on a limited number of classes. Yet, it might struggle to produce a shape that corresponds to the input sketch if it cannot resolve these ambiguities. Fig. 12 shows such limitations. Fig. 12(i) exhibits that SENS might omit, deform, or add additional details that were not required by the user. This problem also appears in the airplane’s tail in Fig. 7. Yet, Fig. 5 demonstrates that tackling this ambiguity can benefit the consistency of the result. This often comes down to a trade-off between being close to the input or producing a coherent shape. Fig. 12(ii) shows that although the sketch may be drawn with precision, the final shape may not include the high-frequency details or patterns depicted in the sketch. Such challenges can be attributed to our method’s handling of sketches with varying abstraction levels, inherent limitations in the SPAGHETTI shape decoder’s detail rendering capabilities, and the absence of view parameters to guide the generation process; factors that collectively impact the method’s ability to deal with intricate details. As we condition SENS to limited classes of shapes, the output is restricted to an object of such a class, even when the input sketch is unrelated. Fig. 12(iii) presents such direct example. Also, the stool sketch in the middle row of Fig. 13 is not correctly mapped. Multi-class SENS is subject to misinterpretation of the shape category, as exemplified in the top right corner of Fig. 14. Finally, SENS inherits some of SPAGHETTI’s limitations, such as the necessity of training on a limited number of shape classes with similar structures, similar artifacts and lack of fine detail in the generated shapes, and potential under or over-clustering of parts within the same Gaussian, which restricts the desired level of control permitted by our selection tool for refinement or part-based modeling.



**Figure 12:** While our method quickly allows to obtain a shape from a drawing, it struggles in certain cases. (ii) comes from *ProSketch* [ZQG\*21] but was not included in the training data.

#### 5. Conclusion

In this paper, we present SENS, a method for generating neural implicit shapes through sketching. The key concept of our approach is mapping different parts of the input sketch to a part-aware latent space. Each latent code’s part is consistently mapped to a different part of the generated shape. Our part-aware reconstruction approach allows the network to integrate the relationships between different parts of the object, resulting in 3D models that are less prone to mere shape retrieval from the training dataset. In addition, we also offer part-based shape modeling, where users can select a part of a shape and redraw its corresponding sketch. This allows for even more precise model editing, and enables users to combine features from different shapes, thus expanding the scope of what can be modeled beyond the dataset’s inherent limitations. Another implication of a part-aware latent space is the possibility to refine specific parts of the shape, hence allowing systematic artifacts removal in the final model. Recent developments in generative diffusion-based models have shown promise for sketch-to-shape modeling, as highlighted in works like [ZPW\*23]. These models, when combined with part-aware shape decoders [BKD\*23], offer new potential for advancing the field. This integration not only enhances current methodologies but also paves the way for innovative research directions in sketch-based shape generation.

Among the key contributions of our method also lies the ability to generate shapes via a single sketch at various levels of abstraction. Moreover, we can edit their outline directly through sketching, reducing the need for advanced artistic skills in the modeling process. We have shown through our experiments and comparisons with prior shape generation methods that SENS generates models with a higher level of detail and realism while requiring less drawing expertise. We believe that our method provides a powerful tool for creating 3D models, offering both ease of use and high-quality results.

#### Acknowledgements

We thank the reviewers for their insightful and constructive comments. We use Silvia Sellán’s Blender template for rendering. This work was supported in part by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation program (grant agreement No. 101003104, ERC CoG MYCLOTH). Open access funding provided by Eidgenössische Technische Hochschule Zurich.

#### References

- [AHY\*19] ATZMON M., HAIM N., YARIV L., ISRAELOV O., MARON H., LIPMAN Y.: Controlling neural level sets, 2019. doi:10.48550/ARXIV.1905.11911. 1

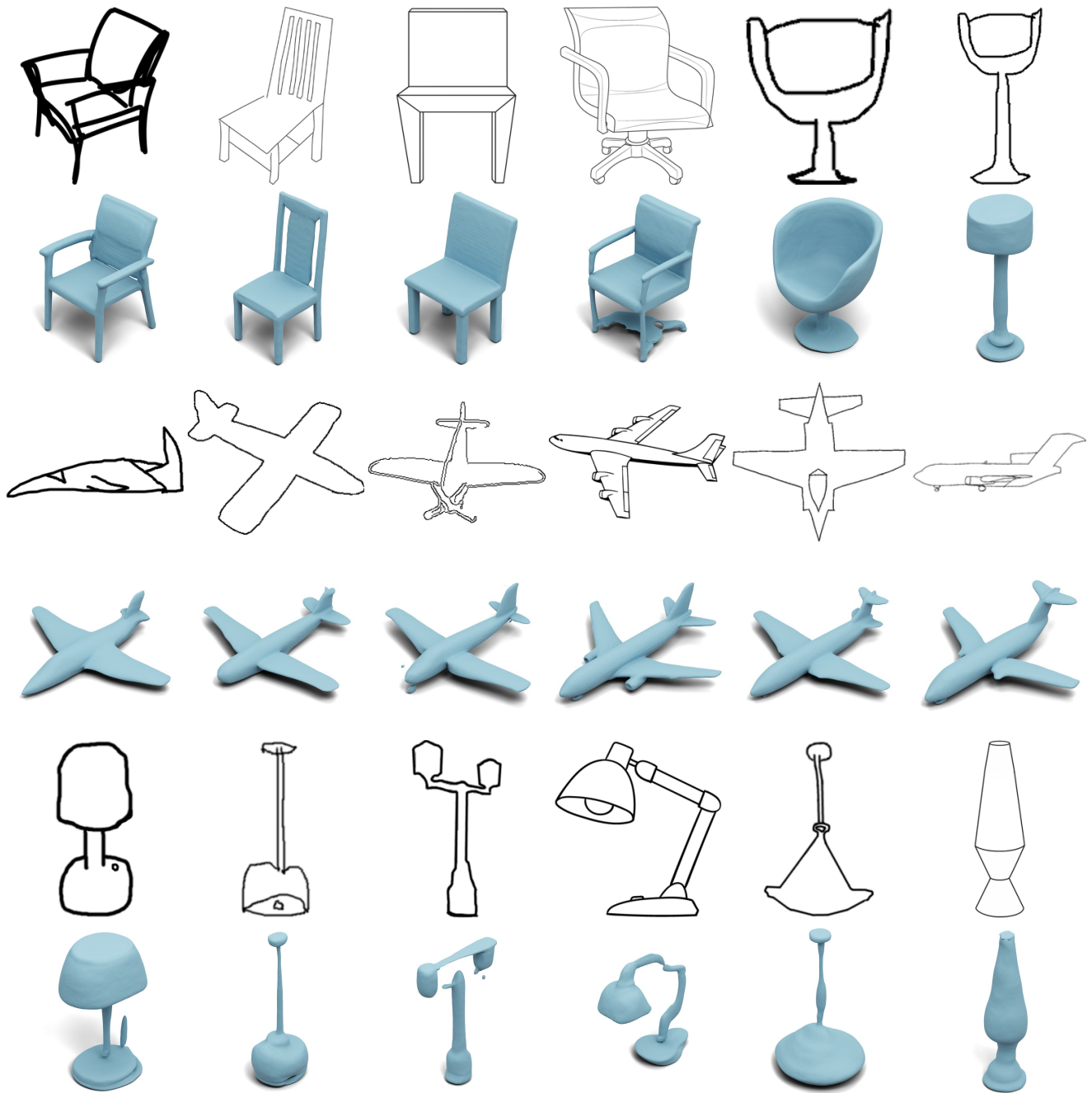
- [BAC\*19] BONNICI A., AKMAN A., CALLEJA G., CAMILLERI K., FEHLING P., FERREIRA A., HERMUTH F., ISRAEL J., LANDWEHR T., LIU J., PADFIELD N., SEZGIN T., ROSIN P.: Sketch-based interaction and modeling: where do we stand? *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 33 (11 2019), 1–19. doi:10.1017/S0890060419000349. 2
- [BKD\*23] BANDYOPADHYAY H., KOLEY S., DAS A., SAIN A., CHOWDHURY P. N., XIANG T., BHUNIA A. K., SONG Y.-Z.: Doodle your 3d: From abstract freehand sketches to precise 3d shapes. *arXiv preprint arXiv:2312.04043* (2023). 9
- [BPCB08] BERNHARDT A., PIHUIT A., CANI M.-P., BARTHE L.: Matisse : Painting 2D regions for Modeling Free-Form Shapes. In *SBM'08 - Eurographics Workshop on Sketch-Based Interfaces and Modeling* (Annecy, France, June 2008), Alvarado C., Cani M.-P., (Eds.), SBM'08 Proceedings of the Fifth Eurographics conference on Sketch-Based Interfaces and Modeling, Eurographics Association, pp. 57–64. doi:10.2312/SBM/SBM08/057-064. 2
- [Can86] CANNY J.: A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8*, 6 (1986), 679–698. doi:10.1109/TPAMI.1986.4767851. 7
- [CCR\*22] CHENG Z., CHAI M., REN J., LEE H.-Y., OLSZEWSKI K., HUANG Z., MAJI S., TULYAKOV S.: Cross-modal 3d shape generation and manipulation. In *European Conference on Computer Vision (ECCV)* (2022). 3
- [CFG\*15] CHANG A. X., FUNKHOUSER T., GUIBAS L., HANRAHAN P., HUANG Q., LI Z., SAVARESE S., SAVVA M., SONG S., SU H., XIAO J., YI L., YU F.: Shapenet: An information-rich 3d model repository, 2015. doi:10.48550/ARXIV.1512.03012. 4
- [CIW08] CANI M.-P., IGARASHI T., WYVILL G.: *Interactive Shape Design*. Synthesis Lectures on Computer Graphics and Animation. Morgan & Claypool Publishers, ISSN:1933-8996, July 2008. doi:10.2200/S00122ED1V01Y200806CGR006. 2
- [CMPM20] CHIBANE J., MIR A., PONS-MOLL G.: Neural unsigned distance fields for implicit function learning. In *Advances in Neural Information Processing Systems (NeurIPS)* (December 2020). 3
- [CMS\*20] CARION N., MASSA F., SYNNAEVE G., USUNIER N., KIRILLOV A., ZAGORUYKO S.: End-to-end object detection with transformers. In *European conference on computer vision* (2020), Springer, pp. 213–229. 4
- [CWC\*22] CHOWDHURY P. N., WANG T., CEYLAN D., SONG Y.-Z., GRYADITSKAYA Y.: Garment ideation: Iterative view-aware sketch-based garment modeling. In *2022 International Conference on 3D Vision (3DV)* (2022), pp. 22–31. doi:10.1109/3DV57658.2022.00015. 3
- [CXG\*16] CHOY C. B., XU D., GWAK J., CHEN K., SAVARESE S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction, 2016. doi:10.48550/ARXIV.1604.00449. 3
- [CZ19] CHEN Z., ZHANG H.: Learning implicit fields for generative shape modeling. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 5932–5941. doi:10.1109/CVPR.2019.00609. 1, 3
- [DBA\*17] DELANOY J., BOUSSEAU A., AUBRY M., ISOLA P., EFROS A. A.: What you sketch is what you get: 3d sketching using multi-view deep volumetric prediction. *CoRR abs/1707.08390* (2017). arXiv:1707.08390. 3
- [DBK\*20] DOSOVITSKIY A., BEYER L., KOLESNIKOV A., WEISENBORN D., ZHAI X., UNTERTHINER T., DEGHANI M., MINDERER M., HEIGOLD G., GELLY S., USZKOREIT J., HOULSBY N.: An image is worth 16x16 words: Transformers for image recognition at scale, 2020. doi:10.48550/ARXIV.2010.11929. 2, 4
- [DCLT18] DEVLIN J., CHANG M.-W., LEE K., TOUTANOVA K.: Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018). 5
- [DSC\*20] DVOROŽNÁK M., ŠÝKORA D., CURTIS C., CURLESS B., SORKINE-HORNUNG O., SALESIN D.: Monster Mash: A single-view approach to casual 3D modeling and animation. *ACM Transactions on Graphics (proceedings of SIGGRAPH ASIA)* 39, 6 (2020). 2
- [ERB\*12] EITZ M., RICHTER R., BOUBEKEUR T., HILDEBRAND K., ALEXA M.: Sketch-based shape retrieval. *ACM Trans. Graph. (Proc. SIGGRAPH)* 31, 4 (2012), 31:1–31:10. 2
- [FFY\*19] FENG Y., FENG Y., YOU H., ZHAO X., GAO Y.: Meshnet: Mesh neural network for 3d shape representation. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2019), vol. 33, pp. 8279–8286. 2
- [FRH\*21] FONDEVILLA A., ROHMER D., HAHMANN S., BOUSSEAU A., CANI M.-P.: Fashion Transfer: Dressing 3D Characters from Stylized Fashion Sketches. *Computer Graphics Forum* 40, 6 (2021), 466–483. doi:10.1111/cgf.14390. 2
- [FSG16] FAN H., SU H., GUIBAS L.: A point set generation network for 3d object reconstruction from a single image. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (12 2016). 2
- [GHL\*20] GRYADITSKAYA Y., HÄHNLEIN F., LIU C., SHEFFER A., BOUSSEAU A.: Lifting freehand concept sketches into 3d. *ACM Transactions on Graphics (SIGGRAPH Asia Conference Proceedings)* (2020). URL: <http://www-sop.inria.fr/reves/Basilic/2020/GHLSB20>. 2
- [GIZ09] GINGOLD Y., IGARASHI T., ZORIN D.: Structured annotations for 2d-to-3d modeling. *ACM Trans. Graph.* 28 (12 2009). doi:10.1145/1618452.1618494. 2
- [GRYF21] GUILLARD B., REMELLI E., YVERNAY P., FUA P.: Sketch2mesh: Reconstructing and editing 3d shapes from sketches. *CoRR abs/2104.00482* (2021). arXiv:2104.00482. 2, 3, 5, 6, 7
- [GSF22] GUILLARD B., STELLA F., FUA P.: Meshudf: Fast and differentiable meshing of unsigned distance field networks. In *European Conference on Computer Vision* (2022). 3
- [HASB20] HAO Z., AVERBUCH-ELOR H., SNAVELY N., BELONGIE S. J.: Dualsdf: Semantic shape manipulation using a two-level representation. *CoRR abs/2004.02869* (2020). arXiv:2004.02869. 2, 3
- [HGSB22] HÄHNLEIN F., GRYADITSKAYA Y., SHEFFER A., BOUSSEAU A.: Symmetry-driven 3d reconstruction from concept sketches. In *ACM SIGGRAPH 2022 Conference Proceedings* (New York, NY, USA, 2022), SIGGRAPH '22, Association for Computing Machinery. doi:10.1145/3528233.3530723. 2
- [HHF\*18] HANOCKA R., HERTZ A., FISH N., GIRYES R., FLEISHMAN S., COHEN-OR D.: Meshcnn: A network with an edge. *CoRR abs/1809.05910* (2018). arXiv:1809.05910. 2
- [HPG\*22] HERTZ A., PEREL O., GIRYES R., SORKINE-HORNUNG O., COHEN-OR D.: Spaghetti: Editing implicit shapes through part aware generation. *arXiv preprint arXiv:2201.13168* (2022). 2, 3, 4
- [HRU\*18] HEUSEL M., RAMSAUER H., UNTERTHINER T., NESSLER B., HOCHREITER S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018. arXiv:1706.08500. 5
- [HSG18] HUANG J., SU H., GUIBAS L.: Robust watertight manifold surface generation method for shapenet models, 2018. doi:10.48550/ARXIV.1802.01698. 6
- [IMT99] IGARASHI T., MATSUOKA S., TANAKA H.: Teddy: A sketching interface for 3d freeform design. *SIGGRAPH 99 Conference Proceedings, 109-126*. ACM 99 (01 1999), 409–416. doi:10.1145/311535.311602. 2
- [KH13] KAZHDAN M., HOPPE H.: Screened poisson surface reconstruction. *ACM Trans. Graph.* 32, 3 (jul 2013). doi:10.1145/2487228.2487237. 5
- [LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.* 21, 4 (aug 1987), 163–169. doi:10.1145/37402.37422. 6
- [LGK\*17] LUN Z., GADELHA M., KALOGERAKIS E., MAJI S., WANG R.: 3d shape reconstruction from sketches via multi-view convolutional

- networks, 2017. doi:10.48550/ARXIV.1707.06375. 2, 3, 5, 7, 13
- [LPBM20] LI C., PAN H., BOUSSEAU A., MITRA N. J.: Sketch2cad: Sequential cad modeling by sketching in context. *ACM Trans. Graph. (Proceedings of SIGGRAPH Asia 2020)* 39, 6 (2020), 164:1–164:14. doi:https://doi.org/10.1145/3414685.3417807. 2, 3
- [LPBM22] LI C., PAN H., BOUSSEAU A., MITRA N. J.: Free2cad: Parsing freehand drawings into cad commands. *ACM Trans. Graph. (Proceedings of SIGGRAPH 2022)* 41, 4 (2022), 93:1–93:16. doi:https://doi.org/10.1145/3528223.3530133. 3
- [LPL\*18] LI C., PAN H., LIU Y., SHEFFER A., WANG W.: Robust flow-guided neural prediction for sketch-based freeform surface modeling. *ACM Trans. Graph. (SIGGRAPH ASIA)* 37, 6 (2018), 238:1–238:12. doi:10.1145/3272127.3275051. 2, 3
- [MKKv22] MIRBAUER M., KRABEC M., KRIVÁNEK J., ŠIKUDOVÁ E.: Survey and evaluation of neural 3d shape classification approaches. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 11 (2022), 8635–8656. doi:10.1109/TPAMI.2021.3102676. 2
- [MON\*19] MESCHEDER L., OECHSLE M., NIEMEYER M., NOWOZIN S., GEIGER A.: Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (2019). 3
- [MPS\*23] MIKAEILI A., PEREL O., SAFAEE M., COHEN-OR D., MAHDAVI-AMIRI A.: Sked: Sketch-guided text-based 3d editing, 2023. arXiv:2303.10735. 3
- [MST\*20] MILDENHALL B., SRINIVASAN P. P., TANCIK M., BARRON J. T., RAMAMOORTHY R., NG R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV* (2020). 3
- [NISA07] NEALEN A., IGARASHI T., SORKINE O., ALEXA M.: Fiber-Mesh: Designing freeform surfaces with 3D curves. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)* 26, 3 (2007), article no. 41. 2
- [OELS\*22] OR-EL R., LUO X., SHAN M., SHECHTMAN E., PARK J. J., KEMELMACHER-SHLIZERMAN I.: Stylesdf: High-resolution 3d-consistent image and geometry generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2022), pp. 13503–13513. 1
- [PFS\*19] PARK J. J., FLORENCE P., STRAUB J., NEWCOMBE R., LOVE-GROVE S.: DeepSDF: Learning continuous signed distance functions for shape representation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 165–174. 1, 3
- [PLH\*22] PEARL O., LANG I., HU Y., YEH R. A., HANOCKA R.: Geocode: Interpretable shape programs, 2022. doi:10.48550/ARXIV.2212.11715. 3
- [PNM\*20] PENG S., NIEMEYER M., MESCHEDER L., POLLEFEYS M., GEIGER A.: Convolutional occupancy networks, 2020. arXiv:2003.04618. 3
- [PSW\*21] PALMER D., SMIRNOV D., WANG S., CHERN A., SOLOMON J.: Deepcurrents: Learning implicit representations of shapes with boundaries, 2021. doi:10.48550/ARXIV.2111.09383. 3
- [PUG19] PASCHALIDOU D., ULUSOY A. O., GEIGER A.: Superquadrics revisited: Learning 3d shape parsing beyond cuboids. *CoRR abs/1904.09970* (2019). arXiv:1904.09970. 2
- [PZZ22] PARMAR G., ZHANG R., ZHU J.-Y.: On aliased resizing and surprising subtleties in gan evaluation. In *CVPR* (2022). 5
- [QGS\*21] QI A., GRYADITSKAYA Y., SONG J., YANG Y., QI Y., HOSPEDALES T. M., XIANG T., SONG Y.-Z.: Toward fine-grained sketch-based 3d shape retrieval. *Trans. Img. Proc.* 30 (jan 2021), 8595–8606. doi:10.1109/TIP.2021.3118975. 5, 7, 8
- [SBS19] SMIRNOV D., BESSMELTSEV M., SOLOMON J.: Learning manifold patch-based representations of man-made shapes, 2019. doi:10.48550/ARXIV.1906.12337. 3
- [SFk\*20] SMIRNOV D., FISHER M., KIM V. G., ZHANG R., SOLOMON J.: Deep parametric shape predictions using distance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020). 2
- [SKZC18] SCHOR N., KATZIR O., ZHANG H., COHEN-OR D.: Learning to generate the "unseen" via part synthesis and composition. *CoRR abs/1811.07441* (2018). arXiv:1811.07441. 3
- [SLK\*20] SHARMA G., LIU D., KALOGERAKIS E., MAJI S., CHAUDHURI S., MECH R.: Parsenet: A parametric surface fitting network for 3d point clouds. *CoRR abs/2003.12181* (2020). arXiv:2003.12181. 2
- [SMLK06] SCHROEDER W., MARTIN K., LORENSEN B., KITWARE I.: *The Visualization Toolkit: An Object-oriented Approach to 3D Graphics*. Kitware, 2006. URL: https://books.google.ch/books?id=rx4vPwAACAAJ. 6
- [SS08] SCHMIDT R., SINGH K.: Sketch-based procedural surface modeling and compositing using surface trees. *Comput. Graph. Forum* 27 (04 2008), 321–330. doi:10.1111/j.1467-8659.2008.01129.x. 2
- [SSG\*22] SOMEPELLI G., SINGLA V., GOLDBLUM M., GEIPING J., GOLDSTEIN T.: Diffusion art or digital forgery? investigating data replication in diffusion models, 2022. doi:10.48550/ARXIV.2212.03860. 2, 3
- [TCH04] TURQUIN E., CANI M.-P., HUGHES J.: Sketching garments for virtual characters. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling* (Grenoble, France, 2004), Hughes J. F., Jorge J. A., (Eds.), Eurographics. URL: https://hal.inria.fr/inria-00510171. 2
- [TRR\*19] TATARCHENKO M., RICHTER S. R., RANFTL R., LI Z., KOLTUN V., BROX T.: What do single-view 3d reconstruction networks learn?, 2019. arXiv:1905.03678. 2
- [TTM\*22] TEWARI A., THIES J., MILDENHALL B., SRINIVASAN P., TRETSCHK E., YIFAN W., LASSNER C., SITZMANN V., MARTIN-BRUALLA R., LOMBARDI S., SIMON T., THEOBALT C., NIESSNER M., BARRON J. T., WETZSTEIN G., ZOLLHÖFER M., GOLYANIK V.: Advances in neural rendering. *Computer Graphics Forum* 41, 2 (2022), 703–735. 1
- [TZF04] TAI C.-L., ZHANG H., FONG J.: Prototype modeling from sketched silhouettes based on convolution surfaces. *Comput. Graph. Forum* 23 (03 2004), 71–84. doi:10.1111/j.1467-8659.2004.00006.x. 2
- [VPB\*22] VINKER Y., PAJOUHESHGAR E., BO J. Y., BACHMANN R. C., BERMANO A. H., COHEN-OR D., ZAMIR A., SHAMIR A.: Clipasso: Semantically-aware object sketching. *ACM Trans. Graph.* 41, 4 (jul 2022). doi:10.1145/3528223.3530068. 3, 4
- [VSH19] VERHOEVEN F., SORKINE-HORNUNG O.: Rodmesh: Two-handed 3d surface modeling in virtual reality. In *Proceedings of the Symposium on Vision, Modeling and Visualization (VMV)* (2019), Eurographics Association. 2
- [WZL\*18] WANG N., ZHANG Y., LI Z., FU Y., LIU W., JIANG Y.-G.: Pixel2mesh: Generating 3d mesh models from single rgb images. In *ECCV* (2018). 2, 3, 5, 6, 7
- [WZZ\*18] WU J., ZHANG C., ZHANG X., ZHANG Z., FREEMAN W. T., TENENBAUM J. B.: Learning shape priors for single-view 3d completion and reconstruction. *CoRR abs/1809.05068* (2018). arXiv:1809.05068. 2
- [XCS\*14] XU B., CHANG W., SHEFFER A., BOUSSEAU A., MCCRAE J., SINGH K.: True2Form: 3D Curve Networks from 2D Sketches via Selective Regularization. *ACM Transactions on Graphics* 33, 4 (2014). doi:10.1145/2601097.2601128. 2
- [YAB\*22] YU E., ARORA R., BAERENTZEN J. A., SINGH K., BOUSSEAU A.: Piecewise-Smooth Surface Fitting onto Unstructured 3D Sketches. In *Siggraph 2022 - ACM conference on computer graphics and interactive techniques* (Vancouver, Canada, Aug. 2022). doi:10.1145/3528223.3530100. 2

- [YAS\*21] YU E., ARORA R., STANKO T., BÆRENTZEN J. A., SINGH K., BOUSSEAU A.: Cassie: Curve and surface sketching in immersive environments. In *ACM Conference on Human Factors in Computing Systems (CHI)* (2021). URL: <http://www-sop.inria.fr/revues/Basilic/2021/YASBS21>. 2
- [YHCOZ18] YIN K., HUANG H., COHEN-OR D., ZHANG H.: P2p-NET. *ACM Transactions on Graphics* 37, 4 (aug 2018), 1–13. doi:10.1145/3197517.3201288. 2
- [YHH\*19] YANG G., HUANG X., HAO Z., LIU M.-Y., BELONGIE S., HARIHARAN B.: Pointflow: 3d point cloud generation with continuous normalizing flows, 2019. doi:10.48550/ARXIV.1906.12320. 2
- [ZGZS20] ZHONG Y., GRYADITSKAYA Y., ZHANG H., SONG Y.-Z.: Deep sketch-based modeling: Tips and tricks. In *2020 International Conference on 3D Vision (3DV)* (2020), pp. 543–552. doi:10.1109/3DV50981.2020.00064. 5
- [ZGZS22] ZHONG Y., GRYADITSKAYA Y., ZHANG H., SONG Y.-Z.: A study of deep single sketch-based modeling: View/style invariance, sparsity and latent space disentanglement. *Comput. Graph.* 106, C (aug 2022), 237–247. doi:10.1016/j.cag.2022.06.005. 2, 3, 5, 6, 7
- [ZLWT22] ZHENG X.-Y., LIU Y., WANG P.-S., TONG X.: Sdf-stylegan: Implicit sdf-based stylegan for 3d shape generation. In *Comput. Graph. Forum (SGP)* (2022). 5
- [ZLY\*23] ZHOU J., LUO Z., YU Q., HAN X., FU H.: Ga-sketching: Shape modeling from multi-view sketching with geometry-aligned deep implicit functions, 2023. arXiv:2309.05946. 2, 3
- [ZNW22] ZHANG B., NIESSNER M., WONKA P.: 3DILG: Irregular latent grids for 3d generative modeling. In *Advances in Neural Information Processing Systems* (2022), Oh A. H., Agarwal A., Belgrave D., Cho K., (Eds.). URL: <https://openreview.net/forum?id=R00wSr3R7y->. 3
- [ZPW\*23] ZHENG X.-Y., PAN H., WANG P.-S., TONG X., LIU Y., SHUM H.-Y.: Locally attentional sdf diffusion for controllable 3d shape generation. *ACM Transactions on Graphics (SIGGRAPH)* 42, 4 (2023). 3, 9
- [ZQG\*21] ZHONG Y., QI Y., GRYADITSKAYA Y., ZHANG H., SONG Y.-Z.: Towards practical sketch-based 3d shape generation: The role of professional sketches. *IEEE Transactions on Circuits and Systems for Video Technology* 31, 9 (2021), 3518–3528. doi:10.1109/TCSVT.2020.3040900. 2, 3, 4, 5, 6, 9
- [ZTNW23] ZHANG B., TANG J., NIESSNER M., WONKA P.: 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Trans. Graph.* 42, 4 (jul 2023). URL: <https://doi.org/10.1145/3592442>, doi:10.1145/3592442. 3
- [ZYC\*22] ZHANG C., YANG L., CHEN N., VINING N., SHEFFER A., LAU F. C., WANG G., WANG W.: Creatureshop: Interactive 3d character modeling and texturing from a single color drawing. *IEEE Transactions on Visualization and Computer Graphics* (2022), 1–18. doi:10.1109/TVCG.2022.3197560. 2
- [ZZZ\*18] ZHANG X., ZHANG Z., ZHANG C., TENENBAUM J. B., FREEMAN W. T., WU J.: Learning to reconstruct shapes from unseen classes. *CoRR abs/1812.11166* (2018). arXiv:1812.11166. 2



**Figure 13:** We showcase our method using sketches of various styles and levels of abstraction. Chairs in the first row are casually drawn or produced via image processing techniques. The second row shows that our method works with sketches drawn by professionals. Images from the last row are front and side views of chairs originating from ShapeMVD [LGK\*17]. We include them here to facilitate comparisons with further works.



**Figure 14:** Multi-class SENS can produce chairs, planes and lamps out of sketches at diverse abstraction levels. Note that we do not indicate to the network at inference the kind of object we draw. In some cases, this can lead SENS to misinterpret the class of the drawn shape (see top-right).