UNVEILING LIMITATIONS IN SINGLE-VIEW 3D OBJECT RECONSTRUCTION MODELS

by

Merve Gül Kantarcı

B.S., Computer Engineering, Hacettepe University, 2020

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2024

ACKNOWLEDGEMENTS

I would like to thank my advisors, Prof. Lale Akarun and Dr. Berk Gökberk, for their understanding, continuous support, and guidance throughout my studies. I also would like to thank the jury members, Dr. İnci Baytaş and Dr. Helin Dutağacı, for their valuable feedback and contributions.

I am thankful to TÜBİTAK for granting me support as part of the 2211 Graduate Scholarship Program.

I am grateful to my dear friends Büşra Demirtaş and Nil Melike Pala for always being by my side and standing through the challenges with me, Hatice Şeyma Gurbetoğlu for her constant support and being an inspiration in many ways, Ümmühan Derelioğlu for never doubting my capabilities and being the best at reminding me the positive sides.

I appreciate having the best sisters, Sena and Zümra Kantarcı, who made all the difference with their endless love, support, and patience. I am grateful to Mustafa Bekiroğlu for being there when I need him, like a brother. I thank my mom and dad for their guidance at critical points and lifelong support.

Finally, I am indebted to God Almighty for giving me the ability and patience to complete my thesis.

ABSTRACT

UNVEILING LIMITATIONS IN SINGLE-VIEW 3D OBJECT RECONSTRUCTION MODELS

3D object reconstruction models typically learn from a single dataset, ShapeNetCore [1], and are evaluated against similar datasets that measure aspects closely related to ShapeNetCore. We tackled the problem of limited performance assessment by proposing novel benchmarks to reveal their robustness to new challenges. To demonstrate our benchmark's effectiveness, we selected three state-of-the-art models for comparison: 3D-C2FT [2], Pix2Vox [3], and Occupancy Networks [4]. This selection covers two well-known 3D shape representations: voxel and occupancy function as an implicit representation. We first investigated the effect of changing background color on performance. We found that this seemingly simple variable causes a drastic decrease in performance. We observed that models perform sufficiently close to the original scenario with changing input object sizes in 2D. Further, we adapted a novel dataset 3DCoMPaT++ [5, 6] for 3D reconstruction evaluation. 3DCoMPaT++ offers rich material and part annotations. We assessed reconstruction performance by slightly changing viewpoints and varying styles in 2D input images. The results show that models struggle to adapt to novel settings. Performance degrades drastically in the best-case scenario, and surprisingly, the model performing the best in the standard ShapeNetCore experiment, Pix2Vox, scores the worst across all novel dataset experiments. We also evaluated models at the part level to identify the most challenging parts. We transferred part-level point clouds to part-annotated surface points from 3DCoMPaT++ using point cloud registration with L2 distance. We then utilized our version of F-Score@0.01, Part F-Score@0.01, for evaluation. This experiment quantitatively confirmed the known issue of poor performance in finer details and thin parts, unlike previous works that only made qualitative observations.

ÖZET

TEK İMGEDEN 3B NESNE GERİÇATMA MODELLERİNİN SINIRLARININ AÇIKLANMASI

3B nesne geriçatma modelleri tipik olarak tek bir veri setinden, ShapeNetCore [1], öğrenir ve ShapeNetCore ile yüksek benzerlikli özellikleri ölçen veri setlerine karşı değerlendirilir. Bu çalışmada bu sınırlı değerlendirme sorununu, yeni zorluklara karşı modellerin dayanıklılıklarını ölçmek için testler önererek ele aldık. Önerdiğimiz testlerin etkinliğini göstermek için karşılaştırmada üç yaygın kullanılan modeli seçtik: 3D-C2FT [2], Pix2Vox [3], ve Occupancy Networks [4]. Bu seçim, iki iyi bilinen 3B şekil temsilini kapsar: voksel ve örtük gösterim için doluluk fonksiyonu. İlk olarak, performans üzerinde arka plan renginin değişiminin etkisini araştırdık. Bu görünüşte basit değişkenin performansta önemli bir düşüşe sebep olduğunu bulduk. Modellerin, 2B girdilerdeki nesne boyutları değişse bile, orijinal senaryoya oldukça yakın bir performans sergilediklerini gözlemledik. Daha sonra, yeni tanıtılan 3DCoMPaT++ [5, 6] verisetini 3B geriçatma değerlendirmesi için uyarladık. 3DCoMPaT++ şekiller için malzeme ve parça etiketleri açısından zengindir. Bu setle, 2B imgelerde hafifçe değişen bakış açıları ve stillerle geriçatma performansını değerlendirdik. Sonuçlar, modellerin yeni durumlara uyum sağlamakta zorlandığını gösteriyor. En iyi senaryoda performansta belirgin bir düşüş gözlemlendi ve beklenmedik bir şekilde, standart ShapeNetCore deneylerinin en iyisi Pix2Vox, tüm yeni veri seti deneylerinde en düşük skoru elde etti. Ek olarak, geriçatması en zor nesne kısımlarını belirlemek için modeller parça seviyesinde değerlendirildi. 3DCoMPaT++ setindeki parça nokta bulutlarının bilgisini yüzey nokta bulutlarına L2 mesafesiyle aktardık. Değerlendirme için önerdiğimiz Parça F-Skor@0.01 metriğini kullandık. Böylece, önceki çalışmalarda sadece nitel gözlemlerle aktarılan detay ve ince kısımlarda düşük geriçatma performansı nicel olarak gösterildi.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii
ABSTRACT . . . iv
ÖZET . . . v
LIST OF FIGURES . . . viii
LIST OF TABLES . . . x
LIST OF SYMBOLS . . . xi
LIST OF ACRONYMS/ABBREVIATIONS . . . xii
1. INTRODUCTION . . . 1
2. LITERATURE REVIEW . . . 6
2.1. 3D Shape Representation Types . . . 6
2.2. Voxel Respresentation . . . 7
2.3. Mesh Representation . . . 9
2.4. Point Cloud Representation . . . 10
2.5. Implicit Functions . . . 12
2.6. 3D Shape Datasets . . . 14
3. BASELINE 3D RECONSTRUCTION METHODS . . . 22
3.1. Background on Occupancy Networks . . . 23
3.2. Background on Pix2Vox . . . 25
3.3. Background on 3D-C2FT . . . 26
4. EXPERIMENTS . . . 28
4.1. Datasets . . . 28
4.1.1. ShapeNet . . . 28
4.1.2. 3DCoMPaT++ . . . 30
4.2. Experimental Setup . . . 32
4.2.1. Input . . . 34
4.2.2. Voxels . . . 35
4.2.3. Watertight Shapes . . . 35
4.2.4. Transferring Part Annotation . . . 36

4.2.5. Baselines 37
4.3. Evaluation Metrics 39
4.3.1. Voxel Metrics 39
4.3.2. Point Cloud Metrics 40
4.3.2.1. F-Score and Part F-Score 40
4.3.2.2. Chamfer Distance 41
4.4. ShapeNet Experiments 41
4.4.1. Original ShapeNet Test 41
4.4.2. Adding Background Color 42
4.4.3. Changing Object Size in Rendering 46
4.5. 3DCoMPaT++ Experiments 51
4.5.1. Changing Input Viewpoint 51
4.5.2. Changing Input Texture 56
4.5.3. Part-Based Evaluation 60
4.6. Benchmark Results 63
5. CONCLUSION AND FUTURE WORK 65
REFERENCES 68

LIST OF FIGURES

Figure 2.1. An example shape in different 3D representation types: voxel, mesh, point cloud, implicit function. 7

Figure 3.1. A visual summary of selected models' architectures. 22

Figure 4.1. An example of GT mesh in various formats. The difference between 1 and 4 shows an example of voxelization distortion on a mesh. The difference between 1 and 6 shows an example of the effect of simplification on a mesh. 31

Figure 4.2. Example renderings from ShapeNet and 3DCoMPaT++. 32

Figure 4.3. Preprocessing pipeline of GT shapes in ShapeNet. 33

Figure 4.4. Preprocessing pipeline of GT shapes in 3DCoMPaT++. 34

Figure 4.5. Input types for ShapeNet experiments. 34

Figure 4.6. Input types for 3DCoMPaT++ experiments. 35

Figure 4.7. Mesh simplification example for categories: speaker and lamp. . . 36

Figure 4.8. Updates in the preprocessing pipeline for part annotation transfer. 37

Figure 4.9. Part annotation transfer example for voxel surface and watertight mesh. 38

Figure 4.10. Reconstruction examples of images with the black background. . . 44

Figure 4.11. Reconstruction examples of images with the black background. . . 45

Figure 4.12. Reconstruction examples from the chairs category at the selected scales. 48

Figure 4.13. Reconstruction examples of original input and scaled by 0.75. . . . 49

Figure 4.14. Reconstruction examples of original input and scaled by 0.75. . . . 50

Figure 4.15. Examples of the selected viewpoints from two categories. 51

Figure 4.16. Reconstruction examples of images from 3 viewpoints. 53

Figure 4.17. Reconstruction examples of images from 3 viewpoints. 54

Figure 4.18. Reconstruction examples of images from 3 viewpoints. 55

Figure 4.19. Reconstruction examples of images with 3 different object textures. 57

Figure 4.20. Reconstruction examples of images with 3 different object textures. 58

Figure 4.21. Reconstruction examples of images with 3 different object textures. 59

Figure 4.22. Example reconstructions at part level. The closest points between shapes in each row are painted in the same color. 61

LIST OF TABLES

Table 2.1. Summary of reviewed models. 19

Table 2.2. Summary of reviewed models. 20

Table 2.3. Comparison of reviewed datasets. 21

Table 4.1. Categories in ShapeNet and 3DCoMPaT++. 29

Table 4.2. ShapeNet test results of original input images. 42

Table 4.3. ShapeNet test results of images with black background. 43

Table 4.4. ShapeNet test results of input images scaled by 90%. 46

Table 4.5. ShapeNet test results of input images scaled by 75%. 47

Table 4.6. ShapeNet test results of input images scaled by 50%. 48

Table 4.7. 3DCoMPaT++ F-Score results for 3 different viewpoints. 52

Table 4.8. 3DCoMPaT++ F-Score for 3 different object textures. 56

Table 4.9. 3DCoMPaT++ Part F-Score results. 62

Table 4.10. Summary of benchmark results. 64

LIST OF SYMBOLS

G	Ground truth voxel volume
$\mathcal{L}_{\text{bce}}()$	Binary cross-entropy loss
$\mathcal{L}_{\text{mse}}()$	Mean-square-error loss
$\mathcal{L}_{\text{total}}()$	Combined loss for mean-square-error and structural similarity
M	Total number of voxel cells in a volume
N	Total number of points for occupancy function learning
O^G	Ground truth occupancy probabilities
O^P	Predicted occupancy probabilities
P	Predicted voxel volume
μ_G	Mean of ground truth volume
μ_P	Mean of predicted volume
σ_G^2	Variance of ground truth volume
σ_{GP}^2	Covariance between ground truth and predicted volumes
σ_P^2	Variance of predicted volume

LIST OF ACRONYMS/ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
3D-C2FT	3D Coarse-to-Fine Transformer
C2F	Coarse-to-Fine
CAD	Computer-Aided Design
CNN	Convolutional Neural Network
CO3D	Common Objects in 3D
FCN	Fully Convolutional Network
FLANN	Fast Library for Approximate Nearest Neighbors
GAN	Generative Adversarial Network
GCN	Graph Convolutional Network
GCR	Grounded Compositions of Materials on Parts Recognition
GRU	Gated Recurrent Units
GT	Ground Truth
IoU	Intersection Over Union
MISE	Multiresolution IsoSurface Extraction
MLP	Multilayer Perceptron
MSA	Multi-head Self Attention
MVS	Multi-View Stereo
NERF	Neural Radiance Fields
OccNet	Occupancy Networks
SDF	Signed Distance Function
T-SDF	Truncated Signed Distance Function

1. INTRODUCTION

Three-dimensional (3D) reconstruction involves creating 3D digital models from various input sources such as two-dimensional (2D) images, video, and depth camera captures. Researchers have been working in this field for a long time but the field has recently seen remarkable advancements due to advances in deep learning. Such progress has had remarkable influence in various fields, including medical imaging, autonomous driving, robotics, 3D design, and gaming [7–9].

In medical imaging, for instance, 3D reconstruction is transforming 2D scans into anatomical models, invaluable for surgical preparation, education, and tailored treatments [10]. The automotive industry is leveraging this technology to enhance driver assistance systems and further develop self-driving vehicles, improving navigation and safety by interpreting the surrounding environment in 3D [11]. In urban development, digital modeling aids architects and planners in their design processes [12]. The entertainment industry has also utilized 3D reconstruction to create visually pleasing environments in movies and games, and the generation of personalized avatars has become increasingly popular [13].

Despite its potential, widespread and cost-effective implementation of 3D reconstruction depends on automation, as manual creation is costly and requires expertise. Automated solutions provide quick and accessible options for non-experts as well. Deep learning, in particular, allows a significant shift from traditional multi-view stereo (MVS) methods [14–16] that require numerous input images and often manual post-processing. Deep learning models strive to faithfully reconstruct a shape using just one image, without requiring human intervention.

However, the field is still growing and faces many challenges to achieving this on a large scale and under realistic scenarios. Considering the high volume of applications, 3D reconstruction has branched into more specialized areas such as human body re-

construction, facial modeling, scene creation, and related tasks like object completion and enhancement. We select object reconstruction from its fields. Another concern in categorizing reconstruction tasks is the input types. Input types vary largely, such as videos, depth images, coarse point clouds, and 2D images. Among these, we favor 2D images as input. Despite the development of specialized techniques for different areas, realistic scenarios continue to pose significant challenges for deep learning models. In this thesis, we highlight the challenges of 3D object reconstruction by carefully designing test benchmarks to unveil the current limitations.

The classification of 3D reconstruction methods based on the number of input images is a crucial aspect as well. Single-view reconstruction is limited to only one input image hence it is particularly challenging due to the limited information available to accurately define the object's 3D form. This ill-posed nature is a well-known problem. Several studies [17, 18] demonstrated this through an experiment where reconstructed shapes were compared against a selection from a learned database, showing that shape templates drawn from learned priors would be competitively effective. Using multiple images allows for a more comprehensive description of an object's geometry. The benefit of this multi-view approach is evident when multiple perspectives provide a more complete understanding of the object, as opposed to a single image. However, it also introduces the complexity of fusing information into the final shape. Despite the potential for higher accuracy with multi-view reconstruction, this thesis focuses on single-view reconstruction. This focus is chosen because it represents a more commonly needed and practical scenario. It offers significant benefits for real-world applications where multiple images may not be available.

Many studies have explored single-view 3D object reconstruction, with models gradually improving over time. However, evaluations of these models are often based on outdated datasets and do not fully measure the robustness, which limits the ability to determine the most suitable architecture for specific needs. Some research addresses more challenging situations, such as occluded objects or those within real-world contexts, but these studies do not fully capture the effectiveness of different methods.

In this thesis, we focus on the reconstruction of a single object from a single 2D image. There are many solutions for this task. However, they all have shortcomings. After reviewing the literature, we choose three major approaches, and then evaluate and compare them using a novel methodology, in order to guide further research. We achieve this purpose by proposing a structured benchmark for three carefully selected state-of-the-art deep learning models: Occupancy Networks (OccNet) [4], Pix2Vox [3], and 3D-C2FT [2]. They all are trained on the same 13 categories of ShapeNetCore [1]. Despite OccNet and Pix2Vox being older models from 2019, they remain widely referenced and are open-source. Our evaluations show their performance to be competitive with newer models. 3D-C2FT is a model proposed in 2022 and utilizes a transformer architecture [19]. We especially selected this model since transformers have recently become popular deep learning models in 3D computer vision. These models also differ in the way that they represent the reconstructed shapes.

3D data representation is typically divided into four main categories: voxels, meshes, point clouds, and implicit functions. Voxels extend 2D pixels into 3D but can be memory-heavy at high resolutions. Meshes, made of vertices, edges, and faces, offer detailed representations but are complex for deep learning. Point clouds are collections of 3D points but lack order and smooth surfaces. Implicit functions use mathematical equations to define shapes, providing detail and flexibility but are complex in design. Our selection includes the popular implicit function occupancy function in Occupancy Networks and voxels in Pix2Vox and 3D-C2FT. In our experiments, we predominantly convert to mesh representation by surface fitting to occupancy volumes and voxel volumes, and sample point clouds from these surfaces for model evaluation in a joint representation. While we highlight the strengths and weaknesses of voxels and this implicit representation, we ensure the integrity of the original representation types during conversions to maintain the sterility of our experiments.

Our proposed experiments introduce enhanced test scenarios. The most common vanilla experiment is that models predict 3D shapes from images with a plain white background. For this set of experiments, we use the known dataset by the models,

ShapeNetCore. It is a synthetic dataset which is the most used in 3D reconstruction. It has 3D shapes in mesh and voxel formats. 2D input images are given from random 24 perspectives around the 360 degrees of the object. We expand these by using input test images with black backgrounds and with the objects in input scaled to 90%, 75%, and 50% of their original size. These variations test the models' ability to generalize from standard input images and assess whether they are learning structural features or are overly sensitive to dataset-specific characteristics. We found that changing the background color significantly affects the performance of the two models, while 3D-C2FT maintains reasonable reconstruction quality. The models are reasonably robust to changes in input object size.

Further, we test the models on a new dataset, 3DCoMPaT++ [5,6]. This dataset is originally collected for compositional material recognition in 3D. We find it well-suited for our experiments in 3D reconstruction as it contains 3D shapes similar to those in ShapeNet for 16 categories. It also features richer renderings, new viewpoints, and novel object textures, which are scarce in ShapeNet where models typically learn from textureless images. Most importantly, part segmentations are annotated in 3D shapes.

In our proposed tests with 3DCoMPaT++, the performance of the models drops significantly compared to what was achieved with ShapeNet even in the best scenario. More specifically it drops to 0.16 - 0.11 range from approximately 0.40 in F-Score@0.01 compared to the vanilla test case with ShapeNet. This indicates challenges in adapting to new data even when the categories are consistent between the datasets. Due to this performance decrease, we do not evaluate the models on unseen categories. Our proposed experiments also included changes in viewpoints and object textures in 2D. While none of the models perform reasonably close to ShapeNet-level performance, OccNet and 3D-C2FT were more adaptable to these new scenarios. However, the results are sufficient to prove that our experimental approach offers a valuable new benchmark, yielding insights distinct from those provided by standard datasets like ShapeNet and Pix3D [20].

Finally and most importantly, we address the common challenge of reconstructing thin parts or fine details, which are only noted qualitatively in previous studies. Using the part segment labels available in 3DCoMPaT++ and our version of the well-known F-Score metric for part-level evaluation, we quantify the reconstruction performance in finer detail and with part labels. This allows us to move beyond subjective observations to identify which parts are most challenging for the models. The contributions of this thesis can be listed as:

- Sensitivity Analysis: We highlight the over-sensitivity of current models to minor variations in the training dataset.
- New Dataset Adaptation: We introduce a new dataset from a related field and outline the adaptation process for testing 3D reconstruction models.
  (i) We create a benchmark for invariance to viewpoint changes, which is crucial in single-view reconstruction.
  (ii) We establish a benchmark for invariance to input object texture changes. This aspect is well-represented in the new dataset but rare in ShapeNet.
  (iii) We successfully transfer labels across multiple mesh formats, setting a baseline for part-level reconstruction with a comprehensive dataset.
- Part-level Evaluation: We propose a novel benchmark methodology and adapt a well-known F-score to quantitatively assess reconstructions, focusing on part-level and finer-detail challenges. To the best of our knowledge, we are the first to evaluate holistic methods at this finer level in 3D object reconstruction.

Chapter 2 gives a review of 3D reconstruction methods. Chapter 3 explains the compared baseline methods in detail. Chapter 4 explains our proposed benchmark, evaluation criteria, and the results. Chapter 5 concludes the thesis.

2. LITERATURE REVIEW

In this section, we conduct a thorough investigation to identify baseline models for our experiment framework. Given the high volume of methods within the field, we organize this section according to output representation. The selection of output representation stands as a critical criterion since it has the most influence on architecture choice. Consequently, we categorize the literature review based on representation types: voxel, mesh, point clouds, and implicit functions. We first explain these representation forms and give a review of the methods. Table 2.1 and Table 2.2 summarize some of the reviewed models for a better comparison. Consequently, we explore 3D datasets within this field and close fields with goals similar to our research focus.

2.1. 3D Shape Representation Types

3D data can be represented in various ways, and often they are categorized into four main groups: voxels, meshes, point clouds, and implicit functions [8]. Voxels are a direct extension of pixel-based representation in 2D images, offering a discrete, volume-based way to represent 3D data. Despite being memory-intensive at high resolutions, voxels remain popular due to their straightforward nature. On the other hand, meshes, which utilize vertices, edges, and faces, provide a more efficient and smooth representation. However, learning this structure can be challenging for deep learning models. This form is most commonly preferred as a post-processing step. For example, voxels can be obtained in a more visually appealing mesh form by fitting a surface. Point clouds are collections of points in 3D. They are simple, lightweight, and easy to obtain with technologies like LIDAR. Yet, their unordered nature makes them difficult to manipulate, and the sparsity decreases the representation power. Implicit functions, defining shapes through mathematical functions that identify points inside, outside, or on an object's surface, offer high flexibility and detail but come with design complexity. The choice of method depends on the application's specific needs, considering the balance between detail, computational load, and data nature. Each representation has

its unique advantages: voxels for being straightforward, meshes for being smooth and de facto in design, point clouds for simplicity, and implicit functions for high quality at reduced costs. Our extensive review reveals that voxels and implicit functions are increasingly popular in image reconstruction. To illustrate these concepts, an example shape represented in each of these formats is depicted in Figure 2.1.



Figure 2.1. An example shape in different 3D representation types: voxel, mesh, point cloud, implicit function.

2.2. Voxel Respresentation

Voxels are the direct equivalent of pixels in 3D. Choy et al. [21] utilized voxels in their study. They introduced a convolutional LSTM [22] architecture for 3D integration. This architecture deploys 3D-LSTM units to encode features across regions of the shape. The authors noted the superior performance of the Gated recurrent units (GRU) [23]. Convolutional neural networks (CNNs) were used for encoding and decoding. Being the earliest model into reconstruction with deep learning, this study holds significant influence in the history. Subsequent studies addressed the drawbacks of using recurrent neural networks, such as high time complexity, sensitivity to initialization, and long-term memory loss. For instance, Yang et al. [24] improved a fusion module in the AttSets model by incorporating attentional aggregation. This module is compatible with other encoder-decoder architectures, ensuring high integration.

Xie et al. [3], [25] proposed Pix2Vox, a unified system for single and multiple image reconstruction and an enhanced version specifically in multi-view scenario Pix2Vox++. In their system, 3D coarse representations were obtained for each image using CNN encoder-decoder modules from 2D images. The multi-scale context-aware fusion module presented in the method produces a score map from coarse representations, combining the best-predicted region from each representation in the final output. The authors demonstrated that using multi-scale features in the fusion module improved results in multi-view reconstruction settings.

Yang et al. [26] proposed using shape priors to alleviate the ill-posed nature of the problem. They created a memory dictionary during training with feature vectors obtained from a 2D encoder. Instead of selecting a fixed number of shapes from the dictionary for each image, they chose a variable number of shape priors based on similarity scores, enhancing the quality of prior information. The study showed that using shape priors significantly improved reconstruction performance in the presence of heavy occlusion. Similarly, MPCN [27] proposes a shape prior module to guide decoding explicitly. They use an attention mechanism to obtain a fused prior embedding from multiple candidate 3D shape priors. Also, the authors successfully use contrastive loss in a 3D context, aiming to maximize the similarity of similar 3D shapes by computing a similarity score between ground truth (GT) 3D shapes.

Wang et al. [28], on the other hand, emphasized the importance of integrating fusion into the entire model rather than treating it separately. They showcased the significance of their 3D Transformer model, where the encoder learns the relationship between 2D images, and the decoder focuses on the relationship between 2D and 3D representations and the holistic relationship between 3D volumes. The unified architecture was reported to significantly improve reconstruction quality with an increasing number of views compared to the compared methods. Concurrently, LegoFormer [29] proposes a similar transformer architecture with a non-autoregressive decoding scheme. Instead of voxel-based prediction, their transformer model decomposes shapes into meaningful parts. Similar to the work of Wang et al., LegoFormer also reports the efficiency of

integrated aggregation. Most recently, 3D-C2FT [2] proposes a coarse-to-fine patch attention mechanism and reconstructs at least comparably to both CNN-based and transformer-based models. Similar to Pix2Vox, they propose a 3D refiner module to enhance further the output shapes.

While voxel representation is the most straightforward, it suffers from memory and time disadvantages as the computational complexity increases cubically with resolution. Addressing this issue, Roth and Richter [30] treated 3D shapes as a sequence of 2D images, transforming the reconstruction task into predicting a multi-channel 2D image. Their proposed method achieved higher resolution. However, this resolution is still insufficient for practical applications, encouraging researchers to explore different representations.

2.3. Mesh Representation

Mesh representation holds significance in computer graphics, yet modeling it with deep learning poses challenges due to its discrete nature. Pixel2Mesh [31] utilizes graph convolutional networks (GCNs) [32] to describe the nodes, edges, and faces of a triangular mesh. The model initiates deformation from an ellipsoidal mesh, gradually increasing the resolution towards the final mesh output by augmenting the number of nodes. Although this iterative approach implies variable resolution by adding more deformation blocks, the original model restricts itself to three deformation blocks, considering the associated parameter cost. Wen et al. [33] extended Pixel2Mesh by proposing the multi-view deformation network Pixel2Mesh++. It extends the ability to handle multi-view reconstruction.

Gkioxari et al. [34] addressed the impracticality of working on isolated object images with the Mesh R-CNN model. They introduced an object detection module and used voxels as an intermediate form, deviating from prior studies. While their primary contribution is evident in images containing multiple objects, it also serves as a solution for images with complex backgrounds.

AtlasNet [35] learns to map 2D input points to 3D surface points. It samples 2D points from deformed patches, which are then used to create the 3D surface. Unlike point cloud generation which produces points both inside and on the surface of an object, AtlasNet focuses only on the surface. The network achieves this through parametrization in its MLP decoder, employing implicit surfaces in the process. Additionally, AtlasNet is capable of directly reconstructing meshes by effectively establishing triangular connectivity among the sampled points from the same patch. Thus it eliminates the need for post-processing mesh generation.

Sketch2Mesh [36] focused on generating shapes from 2D sketches. The model outlined objects by projecting generated 3D shapes onto a 2D plane, allowing comparison using bidirectional 2D Chamfer Distance. The study also presented an alternative method where the model obtained 2D images from 3D shapes using a differentiable renderer, comparing it with the projection method. Despite the need for camera parameters, the projection-based method is preferable due to its directness, superior quantitative performance, and production of more robust shapes. However, a drawback lies in the requirement for camera parameters.

In a recent development, UNICORN [37] learns mesh deformation through an autoencoder that jointly predicts 3D shape and pose. The method effectively learns from large collections of unlabeled images depicting the same object category. Two notable contributions include progressive conditioning, a training strategy gradually customizing the model from category to instance in a curriculum learning fashion, and neighbor reconstruction, a loss term enforcing consistency between instances with similar shapes or textures. The drawback is that for each category, UNICORN requires training in a separate model.

2.4. Point Cloud Representation

Point cloud representation has gained widespread use due to its flexibility, particularly in the development of unsupervised methods through rendering or projection. In-

safutdinov and Dosovitskiy [38] introduced unsupervised learning using a differentiable renderer. Their neural network leveraged predicted object and camera parameters to obtain 2D views with the renderer, training the model with the mean square error between the 2D view and the input. Han et al. [39] claimed that rendering is costly by proposing a well-designed loss function for projection as a more efficient alternative. Their loss function comprised two parts: the first part aligned points onto the input silhouette image, while the second part increased the distance between projected points to ensure uniform silhouette coverage. This approach also required camera parameters. Chen et al. [40] chose to sample 2D points from input images instead of comparing pixels. Their method expected the sampled 2D projections from the generated 3D object to match the samples, and the Chamfer Distance proved effective for this purpose. To enhance the model's efficiency, they introduced the Structure Adaptive Sampling method, which strikes a balance between dense and sparse sampling.

The sparse and irregular nature of point cloud representation poses challenges for convolutional methods. Liu et al. [41] proposed a point-voxel hybrid representation to address this inefficiency. Points are voxelized for convolution, and after convolution, the final point cloud form is obtained. Similarly, Zhou et al. [42] utilized a point-voxel CNN model that introduces a diffusion model to reproduce shapes from the noise. While this model cannot be utilized for single-view reconstruction from 2D images, it has been proven significant to correct the disadvantages of point cloud representation by showcasing the method's effectiveness in partial point cloud completion tasks.

While not directly related, we note that the 2-stage approach of Sketch2Sampler [43] is promising. In this study, final point sampling occurs through an intermediate 2D density map. This is an interesting alternative to the common approach of directly predicting 2D to 3D in this line of work.

More similar to our work, PGNet [44] is a point cloud generator that uniquely operates at the part level. It sequentially generates parts based on corresponding renderings, rather than creating the entire shape at once. These parts are then assembled

to form the complete shape. Each part decoder functions independently but also utilizes the encoded latent from the preceding part and the embedding of the original image. This approach requires 2D renderings at part level as well. Recently Yu et al. proposed Part-Wise AtlasNet [45]. It enhances the original AtlasNet [35] for 3D object reconstruction. It differs by assigning individual networks to reconstruct specific parts of a 3D object, unlike AtlasNet's more general patch-wise approach. This structured reconstruction allows for more precise local structure and detail in the generated 3D point clouds. However, both PGNet and Part-Wise AtlasNet lack in providing detailed part-level categorical results, are not extensive in the number of shape categories and do not transfer part labels from holistic reconstructions for joint evaluations, unlike our method. Additionally, their evaluations focus not only on surface points but also on points inside a shape.

2.5. Implicit Functions

Implicit functions offer promising representations for 3D shapes, primarily due to their adaptability to variable resolutions. In contrast to voxel representations, implicit functions are efficient in terms of memory cost and excel in capturing surface boundaries with smoother precision. Implicit functions encompass various surface-defining functions, including binary mapping functions, signed distance functions (SDF), truncated signed distance functions (T-SDF), and neural radiance fields (NERF). These functions collectively contribute to a versatile framework for representing 3D surfaces.

Mescheder et al. [4] introduced the Occupancy Networks model, designed to determine object surfaces. This model processes points in 3D space through binary classification, distinguishing between points inside and outside the object. The decision boundaries derived from this classification effectively describe the object surface. Peng et al. [46] extended this concept with Convolutional Occupancy Networks, employing three new encoders to encode a given 3D input in a single 2D plane, multiple 2D planes, or a 3D plane. This approach enhances local feature encoding, resulting in a richer representation and enabling a more lightweight binary classification network. The de-

veloped model demonstrates improved reconstruction quality and increased training speed for 3D inputs. Venkatesh et al. [47] extended the modeling scope to encompass not only closed-form shapes but also open-form shapes with CSPNet. Utilizing the 3D feature vector from Convolutional Occupancy Networks' encoder, CSPNet mapped query points to their nearest surface points. While excelling in modeling open-form shapes, it requires a 3D point cloud as input and is not suitable for 2D-to-3D reconstruction.

Mildenhall et al. [48] proposed training a neural network with multiple images, camera positions, and viewpoints for each scene to create implicit and continuous radiance fields. The resulting neural radiance fields can be utilized for advanced scene representation. Trevithick and Yang [49] introduced the general radiance field method, enhancing NeRF's generalization capacity by replacing 3D point locations with 3D feature vectors. Oechsle et al. [50] combined implicit functions and radiance fields from Occupancy Networks to present UNISURF, which effectively smoothes the gappy surfaces generated by NeRF.

Yang et al. [51] emphasized the inadequacy of classical training methods in processing prior knowledge of networks. The authors focused on optimizing the learned prior shape information. Ibing et al. [52] proposed using multiple local functions to describe objects, encoded via an autoencoder as implicit function grids. Generative Adversarial Networks (GANs) were employed to learn shape generation from the autoencoder's latent space. Recently, Liu et al. [53] leveraged 2D GANs and neural renderers to learn shapes. They propose to learn shapes by matching with multi-view pseudo-images generated by pretrained StyleGAN [54] given an input image. The rendered views of the generated shape allow the network to learn camera parameters to facilitate this matching while using pseudo-images as a method for supervision. Similarly, ShapeClipper [55] learns from only 2D images, with a neural renderer and SDF. It introduces a CLIP-based [56] shape consistency to bring shapes with similar input image CLIP encodings closer instead of supervising with GAN generations. It benefits from the semantic understanding and robustness to viewpoint changes of CLIP.

Implicit functions have been used to address interpretability as well. Zheng et al. [57] tackled it by proposing implicit templates by learning implicit functions that represent categories in their most general forms. A warping function was introduced to map input points to their canonical positions for shape reconstruction. Similarly, Deng et al. [58] dealt with implicit deformed functions. They propose a method to learn template representation for categories and subsequently learn deformation and correction fields for object instances. Both studies are valuable in addressing an important gap with SDF.

2.6. 3D Shape Datasets

Collecting datasets for 3D reconstruction poses unique challenges, primarily due to the high cost of manually designing 3D shapes and the necessity for precise 2D to 3D matching. Accurately predicting how a 3D shape would appear from specific viewpoints is essential. While synthetic generation offers tools for easy rendering, achieving exact 2D-to-3D matching in real-world scenarios is significantly more costly. Additionally, significant variation exists in shape representation methods. Ideally, a collection of Computer-Aided Design (CAD) models should be available in various forms, such as meshes, point clouds, and voxels, to suit different reconstruction needs. Despite these challenges, numerous efforts aim to collect such datasets for both direct reconstruction tasks and related areas of 3D computer vision. This section evaluates a wide range of datasets to identify high-quality resources that could enhance current test benchmarks in the field.

Given the limited availability of large-scale datasets, models often rely on the same datasets for benchmarking and comparison. ShapeNet [1], particularly its subset ShapeNetCore, is the most widely used dataset. It includes 55 categories, but only 13 of these, comprising about 44,000 3D shapes, are commonly used in models. The pioneering model using ShapeNet for 3D reconstruction, 3D-R2N2, selected these 13 categories. Subsequent studies often employ the rendered views of objects provided by 3D-R2N2, offering 360-degree perspectives and with exactly 24 views per shape.

However, the perspectives across different shapes within the same category may not be consistent. PartNet [59] provides fine-grained part-level annotations for 24 categories within ShapeNetCore. It aims to facilitate part recognition and segmentation tasks in 3D. This detailed information proves valuable for assessing part-level performance in 3D reconstructions trained on ShapeNetCore. However, PartNet is not entirely novel, as its annotations are so detailed that they do not adequately highlight the more challenging patterns. Moreover, these annotations are generated through an automated process, which can affect their reliability.

ModelNet [60], sharing similarities with ShapeNet, contains a vast collection of 3D shapes. The entire ModelNet project includes over 600 categories, but the most commonly used subsets are ModelNet10 and ModelNet40. Despite some studies using ModelNet to assess models' generalization capacity, the similarities between ShapeNet and ModelNet limit ModelNet's effectiveness as a distinct benchmark. Both datasets comprise synthetic images, and although there are attempts to enrich 2D views with complex backgrounds, these are not widely used or entirely realistic.

Pix3D [20] stands out for its real-world approach in 3D reconstruction. It features 392 distinct objects like chairs, desks, and sofas, paired with both 3D models and 2D images. This dataset's alignment of 3D models with real-world images offers a more realistic benchmark for 3D reconstruction tasks, especially when compared to synthetic datasets like ShapeNet and ModelNet. Pix3D's resemblance to real-world scenarios makes it a valuable tool for evaluating model performance in practical applications. Models trained on synthetic data often face challenges in accurately predicting from natural, in-the-wild images. Using masked 2D images for benchmarking is not ideal, as it mirrors the limitations of ShapeNet, offers fewer shapes, and does not significantly differ in the evaluation aspect. Similar datasets like the Ikea Dataset [61] and Pascal3D+ [62] also present natural 2D images, but with their limitations: the Ikea Dataset has limited examples with precise 2D-3D matching, and Pascal3D+ only offers approximate matchings. Thus, Pix3D provides a more balanced approach for natural image contexts. ObjectNet3D [63], with its vast collection of 100 categories

and 44,000 3D shapes, also presents natural 2D images with exact matching. Its drawback lies in the overlap of shapes in the training and test sets, which is not ideal for training purposes. As a benchmark, ObjectNet3D is suitable, but we look for a more comprehensive evaluation beyond mere shape reconstruction performance in zero-shot contexts with more enriched datasets.

The Things3D dataset [3], [25], is a large-scale dataset specifically designed for multi-view 3D object reconstruction from naturalistic images. It comprises 1.68 million images of 280,000 objects, collected from over 39,000 indoor scenarios from the 3D scene dataset SUNCG [64]. These images are rendered from diverse virtual scenes. This dataset stands out for its realistic representation of everyday objects in various settings. However, currently, SUNCG is not publicly available and it can not be redistributed as Things3D either.

The Blobby dataset [65] takes a distinctive approach in the field of 3D reconstruction, particularly regarding its data and intended usage. It is composed primarily of abstract, blob-like shapes. These amorphous forms are specifically curated to challenge and enhance the capabilities of 3D reconstruction algorithms in processing non-standard, smooth, or irregular shapes. This aspect is crucial for advancing algorithms that often encounter difficulties with surfaces that are continuously curving or lack distinct edges, which poses a challenge in accurate reconstruction from 2D images. A key aspect of the Blobby dataset is its utility in assessing the fundamental understanding of models in the 2D-to-3D reconstruction process, particularly in bridging the gap between simple shape recognition and complex object structure interpretation. However, due to its focus on abstract shapes, this dataset is significantly different from the object structures typically used in model training. As a result, the Blobby dataset has been proposed as a method for pretraining in 3D reconstruction tasks [65]. Exposing models to a variety of shape complexities early in the training process aims to enhance their adaptability and depth of understanding when later confronted with more structured and realistic datasets.

The Common Objects in 3D (CO3D) dataset [66] introduces a substantial collection of multi-view images for real-world 3D object reconstruction. It contains 1.5 million images from nearly 19,000 videos, capturing objects from 50 MS-COCO categories. Each object is annotated with accurate cameras and 3D point clouds, making it significantly larger and more diverse than previous datasets. CO3D's real-world focus and extensive annotations offer a significant advancement for in-the-wild evaluations of 3D reconstruction methods, surpassing alternatives in category and object diversity. The CO3D dataset, despite being a vast collection, intersects with the most commonly used subset of ShapeNet in only a few categories: chair, bench, TV/display, cellphone, and couch/sofa. While its textured 2D images are valuable, our goal is to establish a benchmark with higher-quality data. Therefore, we are seeking a dataset that, even if it has fewer shapes, offers a broader range of categories intersecting with those in ShapeNet, to better meet our benchmarking needs for textured images.

Shrestha et al. [67] propose a comprehensive collection for multi-view 3D reconstruction. It includes 998 high-resolution textured 3D models and 847K real-world RGB-D images, each accurately annotated with camera and object poses. This dataset tries to solve the problems in previously mentioned Pascal3D+, Pix3D, and ObjectNet3D by collecting a large-scale multi-view dataset, but it is done for tabletop objects hence offering only a few intersecting categories with ShapeNet. The dataset is suitable for training and testing as demonstrated through 3D reconstruction experiments. However, we look for a benchmarking to evaluate the line of work that uses ShapeNetCore, instead of a completely new dataset for training.

3DCoMPaT++ [5, 6] is an enhanced, large-scale dataset for 3D vision, focusing on compositional recognition. It includes 160 million rendered views of over 10 million stylized 3D shapes, annotated at the part-instance level. The dataset encompasses 41 shape categories, 275 fine-grained part categories, and 293 material classes. A novel aspect of 3DCoMPaT++ is its focus on Grounded CoMPaT Recognition (GCR), a task to recognize and ground compositions of materials on 3D object parts. The dataset combines textured point clouds, textured meshes, depth maps, and segmentation masks,

offering a rich resource for 3D vision research. It also includes textured 2D renderings at part level from 8 comprehensive view angles. We show that this dataset is suitable to be used as a novel enriched test set for models trained on ShapeNet. Table 2.6 presents a comparison of the datasets discussed, in terms of the number of shapes, the number of categories, the nature of the data, and annotations.

Table 2.1. Summary of reviewed models. U: unsupervised, S: supervised, PC: point cloud, IF: implicit function.

Model	**Year**	**S**	**Dataset**	**# of Input**	**Output Type**	**Architecture**
AttSets [24]	2020	S	ShapeNet, ModelNet40 [60], Blobby [65]	Multiple	Voxel	Attentional Aggregation
Pix2Vox [3]	2020	S	ShapeNet, Pix3D [20], Things3D [3], [25]	Single / Multiple	Voxel	Encoder - Decoder - Fusion Network- Refiner Network
Mem3D [26]	2021	S	ShapeNet, Pix3D [20]	Single	Voxel	Encoder - Memory Network - LSTM - Decoder
Wang et al. [28]	2021	S	ShapeNet	Single / Multiple	Voxel	Transformer
LegoFormer [29]	2021	S	ShapeNet	Single / Multiple	Voxel	Transformer
3D-C2FT [2]	2022	S	ShapeNet	Single / Multiple	Voxel	Transformer
MPCN [27]	2022	S	ShapeNet, Pascal3D+ [62]	Single	Voxel	Encoder - Memory Network - Attention Module - Decoder
AtlasNet [35]	2018	S	ShapeNet	Single	Mesh	Encoder - MLP
Pixel2Mesh++ [33]	2019	S	ShapeNet	Multiple	Mesh	Encoder - GCN - Deformation Network

Table 2.2. Summary of reviewed models. U: unsupervised, S: supervised, PC: point cloud, IF: implicit function.

Model	**Year**	**S**	**Dataset**	**# of Input**	**Output Type**	**Architecture**
Sketch2Mesh [36]	2021	U	Chairs [68], Cars [36], ShapeNet	Single	Mesh	Encoder - Decoder
UNICORN [37]	2022	U	Pascal3D+ [62], CUB [69], LSUN [70], CompCars [71], ShapeNet	Single	Mesh	Autoencoder
PGNet [44]	2020	U	Part Annotated ShapeNet [72]	Single	PC	Encoder - LSTM - Decoder
Chen et al. [40]	2021	U	ShapeNet	Single	PC	Encoder - MLP
Part-Wise AtlasNet [45]	2022	S	Part Annotated ShapeNet [73]	Single	PC	Generator - Discriminator
OccNet [4]	2019	S	ShapeNet	Single	IF	Encoder - FCN
Ibing et al. [52]	2021	S	ShapeNet, PartNet [59]	Single	IF	Autoencoder - GAN - Decoder
Liu and Liu [53]	2022	U	LSUN [70], Pascal3D+ [62]	Single	IF	Pretrained StyleGAN [54] - Encoder - MLP
ShapeClipper [55]	2023	U	Pix3D [20], Pascal3D+ [62], OpenImages [74]	Single	IF	CLIP [56] - Encoder - MLP

Table 2.3. Comparison of reviewed datasets.

Dataset	**Num. of Categories**	**Num. of Shapes**	**Real / Synthetic**	**Part Annotation**	**Textured**
ShapeNetCore	55	∼ 51,000	Synthetic	No	No
PartNet	24	∼ 27,000	Synthetic	Yes Fine-Grained	No
ModelNet	662	∼ 128,000	Synthetic	No	No
Pix3D	9	392	Real	No	No
Pascal3D+	12	79	Real	No	No
Ikea Dataset	7	219	Real	No	No
ObjectNet3D	100	∼ 44,000	Real	No	Yes
CO3D	50	5,625	Real	No	Yes
Real World MVR Dataset	15	998	Real	No	Yes
3DCoMPaT++	41	10,000	Synthetic	Yes Fine-Grained & Coarse-Grained	Yes

3. BASELINE 3D RECONSTRUCTION METHODS

3D monocular reconstruction methods have shown success in their undertaken tasks. However, careful evaluation and comparison are needed to highlight failures in realistic scenarios and suggest solutions. Current evaluation methods lack the necessary depth to effectively guide further research in the field. Information about where models fail is limited, and comparative analysis across different models is needed. In response, our study designs multiple experiments to enhance current evaluation benchmarks. Specifically, we propose evaluating models using different input image scales and background colors to assess robustness over the known dataset (ShapeNet). Additionally, we advocate for evaluation on a higher-quality completely unseen dataset, 3DCoMPaT, to measure zero-shot performance in the same categories with changing instance structures. For this zero-shot case, we select interesting viewpoints to compare performance across different views and test models against varying textures of the same shapes. Finally, we provide insights into the reconstructed shapes through part-based evaluation instead of holistic measurements.

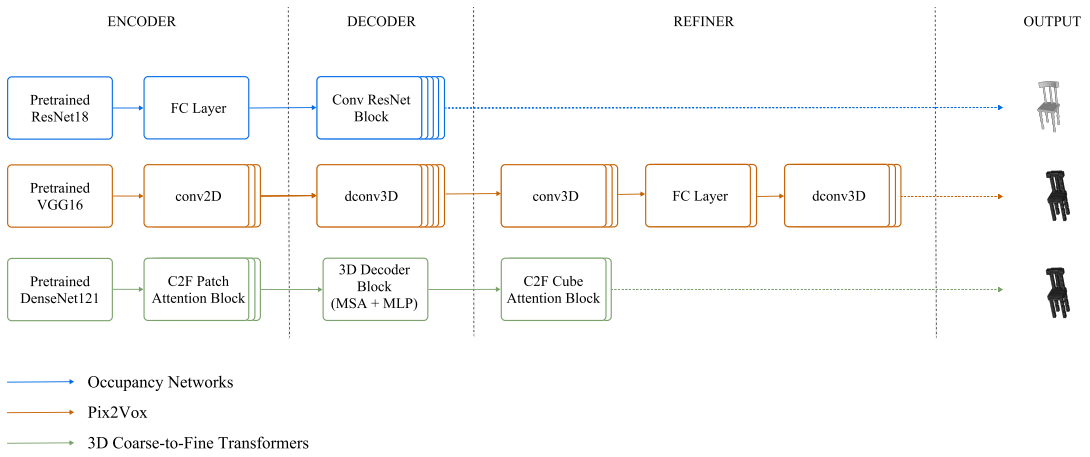


Figure 3.1. A visual summary of selected models' architectures.

We demonstrate the effectiveness of our experiments using three carefully selected methods, chosen based on popular representation types and diverse architectural choices. For a fair comparison, we limited the choice of models to those trained on the

same dataset: We chose models trained on the same subset of ShapeNetCore. The first model in our selection is Occupancy Networks, which utilizes the occupancy function as an implicit surface representation. We also choose two voxel-based methods: Pix2Vox and 3D-C2FT. Pix2Vox is a CNN model, while 3D-C2FT utilizes a Transformer architecture. In this section, the selected models are explained in detail. Figure 3.1 summarizes the model architectures.

3.1. Background on Occupancy Networks

Occupancy Networks [4] is one of the pioneering approaches in 3D reconstruction through the use of implicit functions. The method proposes an occupancy function to classify 3D points as either occupied or unoccupied allowing reconstruction in arbitrary resolution. This is done by learning occupancy probabilities between 0 and 1 for every possible point in 3D space. Then a point in 3D is categorized as inside or outside the shape based on a probability threshold. Instead of directly using point probabilities, the surface is represented by the decision boundary. We find this approach similar to the truncated signed distance function (T-SDF) as it describes the surface based on a decision boundary. However, learning only occupancy probabilities ranging from 0 to 1 is more simplistic than learning truncated distances. The true potential of this representation and the network is demonstrated across multiple 3D tasks such as point cloud completion, voxel super-resolution, and single-view reconstruction. Additionally, the model's representation capacity is proven through an experiment. The authors trained an unconditional generator with KL-divergence loss. This additional model demonstrates the diversity of learned shapes. While our focus centers on single-view 3D reconstruction, these experiments increase our confidence in the efficacy of various potential applications of Occupancy Network decoders.

The Occupancy Network architecture conventionally adopts an encoder-decoder structure, employing ResNet18 [75] as the encoder for single-view reconstruction. The encoder processes the 2D image to extract feature representations. The decoder, a Fully Convolutional Network (FCN) with 5 ResNet blocks, plays a crucial role in learning

the intricate mapping between visual information and underlying 3D geometry. For each point in the sampled points from the output space, the model learns a probability between 0 and 1, with 1 indicating the probability that a shape occupies the space and 0 indicating empty space. The network is trained end-to-end through cross-entropy loss between ground truth point occupancy values and predicted point occupancy probabilities. Even though the model is capable of reconstructing at arbitrary resolutions, during training occupancy probabilities are computed over the finite set of N sampled points. Hence the cross-entropy loss is formulated as

$$\mathcal{L}_{\text{bce}}(O^G, O^P) = \frac{1}{N} \sum_{i=1}^{N} \left[O_i^G \log(O_i^P) + (1 - O_i^G) \log(1 - O_i^P) \right], \tag{3.1}$$

where O^G and O^P are GT and predicted point occupancy values respectively.

During inference, the volumetric space is initially discretized into a grid with a resolution of 32^3. At this resolution, the occupancies are computed, where probabilities are converted to binary values based on a predefined threshold. Subsequently, to achieve higher resolution detail, the occupancy grid is processed using the Multiresolution IsoSurface Extraction (MISE) algorithm. MISE iteratively refines the grid by subdividing cells and identifying the transitions between occupied and unoccupied spaces, particularly boundaries where occupied cells meet unoccupied ones. These boundary cells are subdivided into 8 cells. This adaptive process continues until the grid reaches the target resolution. In practice, it is repeated twice. The final geometric shape is extracted by applying the marching cubes algorithm [76] to the refined occupancy grid, which constructs the isosurface by interpolating the occupied points.

As explained in Section 2, there are many approaches of implicit functions, particularly in recent years. Although Occupancy Networks were first introduced in 2019, we are impressed by its achievements by only using a straightforward implicit function form. Its remaining competitive performance to this date leads us to select Occupancy Networks as one of the baselines in our study.

3.2. Background on Pix2Vox

Pix2Vox [3] is an object reconstruction model leveraging Convolutional Neural Networks (CNNs). Because of the high memory requirements of voxel models, two versions are proposed: Pix2Vox/F and Pix2Vox/A. We explain and utilize the latter one considering its superior performance.

The whole architecture of Pix2Vox is a large CNN framework. It employs a VGG16 [77] network as the encoder, followed by a series of convolutional layers to capture image features. The decoder uses the encoded feature to reconstruct a coarse 3D voxel representation of the object, applying deconvolutional layers for upsampling. In the context of multi-view reconstruction, these coarse volumes are integrated using a context-aware fusion module. It is a 3D convolutional network that processes outputs from the final two layers of the decoder to produce score maps for each reconstruction. These score maps allow the smart selection of the reconstructed parts from the coarse volumes. However, in our study, which focuses solely on single-view reconstruction, this fusion module is not needed.

Additionally, Pix2Vox includes a refinement module that enhances the global structure of the voxel reconstructions without altering the local details. This module processes the 3D volume to correct it, effectively reducing voxelization artifacts which is a common issue in voxel-based models. The refinement module incorporates U-Net-like connections to preserve local details.

The whole framework is trained end-to-end using binary cross-entropy loss

$$\mathcal{L}_{\text{bce}}(G, P) = \frac{1}{M} \sum_{i=1}^{M} [G_i \log(P_i) + (1 - G_i) \log(1 - P_i)] \tag{3.2}$$

between ground truth G voxel and predicted P voxel probabilities. For each voxel at 32^3 dimension, Pix2Vox is optimized to learn 1 for occupied voxels and 0 for unoccupied voxels with this loss.

The advanced version of the model, Pix2Vox++ [25], offers two notable enhancements. First, it has the capability to reconstruct voxel models at higher resolutions, specifically at 128^3, effectively addressing the poor visual results of low resolution associated with voxel representations. Furthermore, it utilizes features extracted at various levels from the encoder, allowing the replacement of the context-aware fusion module with a multi-scale context-aware fusion module. These changes are the most effective with the multi-view reconstruction hence it remains beyond the scope of our current research focus. In addition to these advancements, Pix2Vox++ transitions from the VGG16 to the ResNet50 [75] architecture for its encoder. This change has implications for single-view reconstruction as well; however, our observations indicate that the improvement is marginal (approximately 0.09). Given the low improvement and our interest in exploring a diversity of encoders, we opt to focus our analysis on the original Pix2Vox model equipped with the VGG16 encoder.

3.3. Background on 3D-C2FT

The attention mechanism is used to assign varying degrees of importance to different elements within input data. This functionality allows models to prioritize relevant information during predictions, instead of treating all elements uniformly. The attention mechanism is proven to be effective in processing sequential data. Recently, Vaswani et al. [19] introduced the Transformer architecture, characterized by a network of multi-head self-attention blocks in both the encoder and decoder. Initially designed for natural language processing tasks, the Transformer architecture has found successful adaptation across various domains.

The 3D Coarse-to-Fine Transformer (3D-C2FT) is a state-of-the-art transformer model for processing 2D images to reconstruct 3D shapes. The encoding process involves utilizing DenseNet121 [78] for image encoding, followed by the application of the Coarse-to-Fine (C2F) patch attention block to encode patch embeddings. This attention block comprises a 4-layer multi-head self-attention mechanism, followed by a multilayer perceptron (MLP). The encoder module consists of three of these blocks

in cascading resolutions to work in a coarse-to-fine-grained manner. Starting from the full resolution d=768, subsequent blocks reduce the dimension by half such as d/2, and d/4. The final patch embedding is obtained by concatenating.

The decoder consists of a single block of Multi-head self-attention (MSA) and an MLP. Because the decoder is very simple, the output volume lacks smooth surfaces. Similar to Pix2Vox, they propose a refiner that processes the 3D volume output of the decoder to improve the structure. The 3D refiner within the model consists of two sequential C2F attention blocks that refine the output from the decoder in a hierarchical fashion. The first block processes the decoded volume at a certain patch resolution determined by the output size, which corresponds to $8 \times 8 \times 8$ patches. The second block then takes the output from the first and further refines it at a higher resolution, at $4 \times 4 \times 4$ patches. Each of these cubic C2F attention blocks comprises 6 layers of MSA and MLP operations. These layers work in tandem within each block to iteratively enhance the detail and structure of the 3D volume. The MSA layers capture the global dependencies across the volume and the MLP layers process these features to refine the reconstruction further. This two-staged, fine-grained approach allows the refiner to significantly improve the quality and accuracy of the 3D shape generated from the 2D input images. The transformer is trained end-to-end with two losses: mean square error and structural similarity. These losses are defined as

$$\mathcal{L}_{total}(G,P) = \mathcal{L}_{\text{mse}}(G,P) + \mathcal{L}_{ssim}(G,P)\,, \tag{3.3}$$

$$\mathcal{L}_{\text{mse}}(G,P) = \frac{1}{M}\sum_{i=0}^{M}(G_i - P_i)^2\,, \tag{3.4}$$

$$\mathcal{L}_{ssim}(G,P) = 1 - \frac{(2\mu_G\mu_P + c_1)(2\sigma_{GP} + c_2)}{(\mu_G^2 + \mu_P^2 + c_1)(\sigma_G^2 + \sigma_P^2 + c_2)}\,, \tag{3.5}$$

where G is the ground truth voxel values, P is the predicted voxel probabilities, σ_G^2 and σ_P^2 are the variance of G and the P, μ_G and μ_P are the mean of G and P, σ_{GP} is the covariance between G and P. While the success of the model is attributed to its adept fusion of multi-view inputs through multi-head attention, the proposed C2F patch attention block exhibits superior or, at the very least, comparable performance scores, even in the context of single-view reconstruction.

4. EXPERIMENTS

In this chapter, we evaluate and compare three well-known 3D reconstruction methods, as outlined in the Chapter 3. Among these, Pix2Vox and Occupancy Networks are recognized as established techniques, while 3D-C2FT represents a newer, transformer-based approach. While our focus is on these three methods, we aim to establish a new, competitive benchmark for evaluating 3D reconstruction techniques.

Our testing approach involves using images with varied backgrounds, in various sizes, in various textures, and poses. We conduct cross-dataset tests utilizing a novel dataset, 3DCoMPaT++. Moreover, we detail our process of transferring tests from this new dataset, providing a clear guide for integrating new datasets into 3D reconstruction research. This step is crucial for advancing the field and ensuring that new techniques are adaptable to diverse and evolving data sources. Finally, we explain our method to evaluate reconstructions at part level using 3DCoMPaT++.

4.1. Datasets

4.1.1. ShapeNet

ShapeNetCore is the most widely used subset of the ShapeNet [1] collection. It includes 55 categories. However, following the 3D-R2N2 model [21], only 13 selected categories, of approximately 44,000 unique shapes, have become the standard for most models. These categories are detailed in Table 4.1.

Although the dataset does not provide 2D images, many studies utilize the same rendered views from 3D-R2N2. Each shape is captured from 24 distinct views to achieve a 360-degree perspective, starting from a random camera location. Hence the camera position with the same render index can vary across the shapes within a category. Such variation poses a difficulty in measuring reconstruction performance to view changes.

Voxelization, a process that is sensitive to setup, requires consistent application across different models for reliable results. Conveniently, the 3D-R2N2 model releases voxelized shapes in binvox format. This not only achieves standardization but also sets a benchmark for voxel shapes within the ShapeNet collection. Figure 4.2 shows an example of renderings, mesh, and voxel shapes from the used ShapeNetCore subset.

Table 4.1. Categories in ShapeNet and 3DCoMPaT++.

ShapeNet	**3DCoMPaT++**
airplane	airplane
bench	bench
cabinet	cabinet, dresser, shelf*
car	car
chair	chair, ottoman*, stool*, sun loungers*
lamp	lamp
sofa	sofa, love seat
table	table*, sports table*
watercraft	boat
display	-
speaker	-
rifle	-
telephone	-
-	bags, basket, bbq grill, bed, bicycle, birdhouse, candle holder, clock, coat racks, curtains, dishwasher, fans, faucets, garbage bin, gazebos, jug, ladder, parasol, planter, rug, shower, sinks, skateboard, toilet, tray, trolley, vase

4.1.2. 3DCoMPaT++

3DCoMPaT++ [5,6] is a large-scale dataset for grounded compositional recognition in 3D, which involves identifying parts and materials of 3D shapes. For example, it might classify a sofa as having a metal base with leather seats and cushions. The dataset provides both fine-grained and coarse-grained part annotations for each shape. While the fine-grained annotations are too sensitive for current 3D reconstruction models, we find the coarse-grained part annotations suitable. The dataset includes 43 unique part labels for 10,000 shapes across 41 categories. The top row of Figure 4.2 shows an example of a car mesh and its parts in the point cloud. It should be noted that even though the shapes within this dataset are normalized to the same orientation, we need to establish an alignment configuration for each category. This is necessary to ensure that the shapes are oriented in the same direction as those in ShapeNet.

We selected 16 categories with similar shape structures to those in ShapeNet. Table 4.1 shows these categories and their corresponding category names in ShapeNet. This selection covers 16 categories, matching 9 categories in ShapeNet. These 16 categories reflect our initial selection. However, we later decided to remove the table and sports table categories because their rendering viewpoints are from above, providing very little information about the table legs, which are a primary focus in our study due to the challenging nature of thin parts. Additionally, while ottoman, stool, and sun lounger categories are structurally similar to chairs, we find that chairs already encompass a wide range of structures and do not sufficiently overlap with the ShapeNet category. We also decided to exclude the shelf category. As it consists of many discrete parts, it is overly challenging for the scope of our experiments and beyond the capabilities of the selected models. These eliminated categories are denoted with an asterisk in Table 4.1.

Although 3DCoMPaT++ is ideal for training deep-learning models for compositional recognition, the number of unique shapes is crucial in our study. We use the entire dataset as a test set. At the time of this study, only 2D-level test sets were

available, and meshes were not public. Therefore, we only used meshes from the training and validation sets in our benchmarks from the 10 selected categories, resulting in 3088 unique shapes.

The dataset includes 8 2D renderings for each shape. The first 4 viewpoints are consistent across shapes, unlike ShapeNet. We leverage this consistency to test robustness against viewpoint changes. Additionally, we use varying materials in 2D to introduce a robustness benchmark. These renderings are stylized in both coarse and fine grains with realistic material annotations. We do not use materials at part level but only treat them as different textures. Hence for our purposes, coarse-grained annotations are sufficiently good to obtain varying textures. An ideal model should be able to reconstruct shapes accurately, even with different textures. Figure 4.2 shows an example of renderings from 8 views with 3 styles.

Previous studies have conducted qualitative analyses to identify where models fail in reconstruction. Our part evaluation experiment uses part annotations from this dataset to quantitatively demonstrate these failure points at the part level.

Although rich in quality CAD models, 3DCoMPaT++ lacks voxel shapes and watertight meshes, which are crucial for perfectly evaluating our baselines. We address this gap, as detailed in Section 4.2.

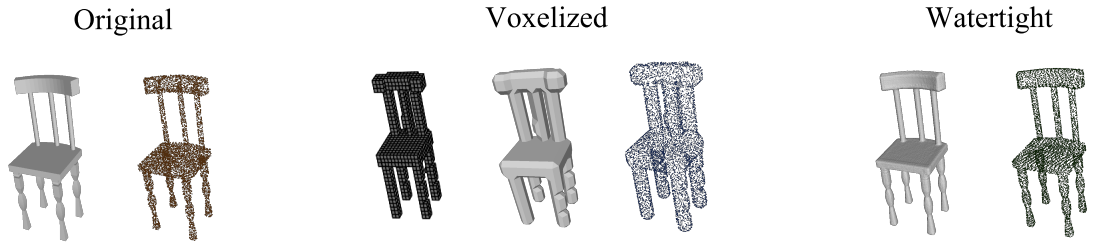


Figure 4.1. An example of GT mesh in various formats. The difference between 1 and 4 shows an example of voxelization distortion on a mesh. The difference between 1 and 6 shows an example of the effect of simplification on a mesh.

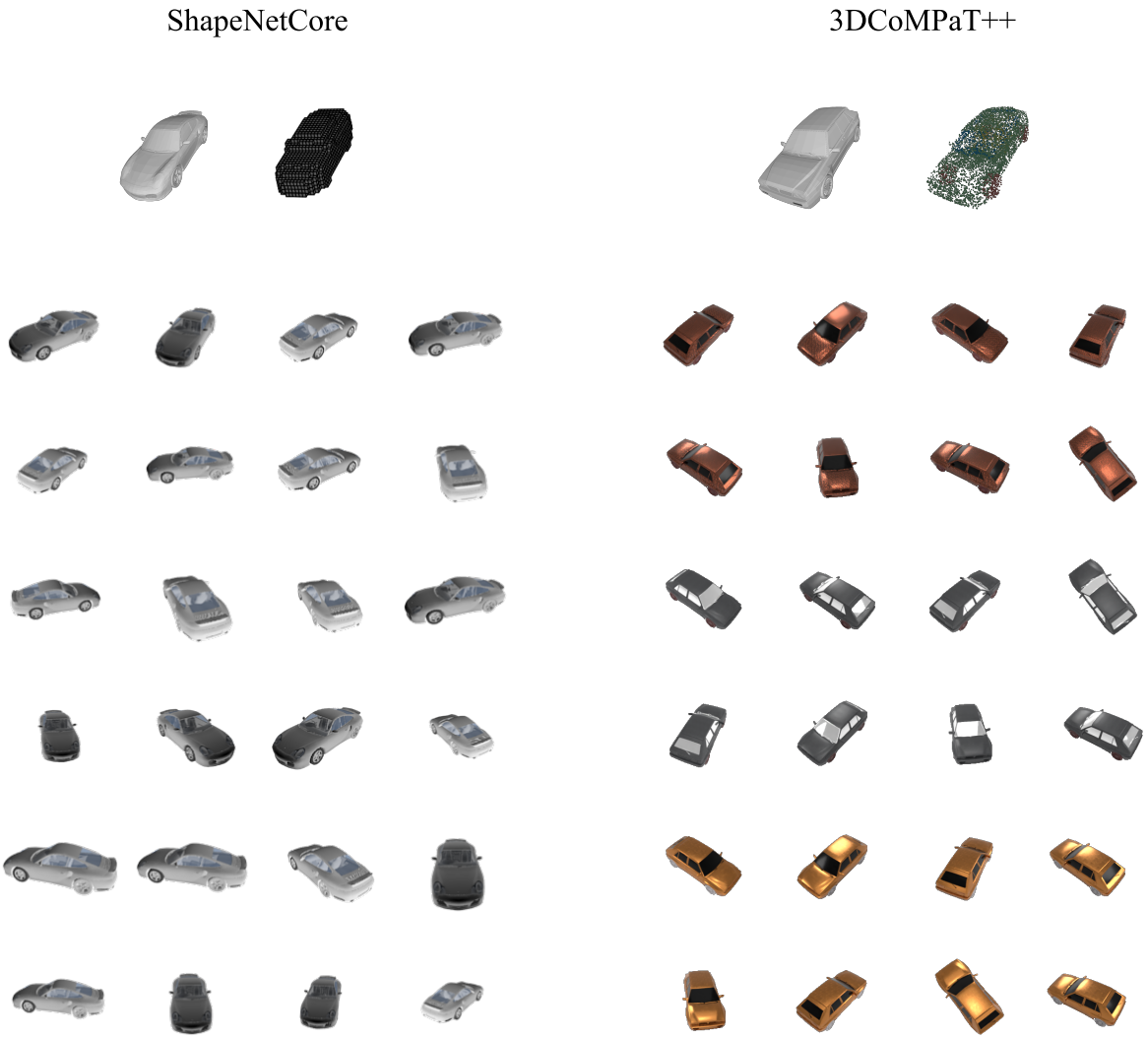


Figure 4.2. Example renderings from ShapeNet and 3DCoMPaT++. The top row shows available GT shapes.

4.2. Experimental Setup

In this section, we explain 2D images and 3D shape formats and how to obtain them in detail. We need variations of 2D images for some experiments instead of the original ones. We require point clouds to measure the F-Score and Chamfer Distance for 3D shapes and voxels for assessing volumetric IoU. Typically, sampling from original meshes should suffice for generating ground truth (GT) point clouds. However, direct comparisons between reconstructions and GT meshes are not always meaningful. For instance, watertight meshes tend to show relatively low error when compared with the

GT mesh. However, a significant discrepancy arises when comparing surfaces fitted to GT voxels with those fitted to GT meshes; here, the F-Score often falls below 0.10. This is attributed to the distortion inherent in voxelization, given its lower dimensionality and larger voxel size. Therefore, we choose to evaluate reconstructions against their respective GT shape representations for a more informative evaluation by eliminating inherent disadvantages of representation. For point cloud metrics, we use 8,192 points sampled from each shape's surface following Xie et al. [3], [25]. In the case of voxels, we fit surfaces to both reconstructed and GT voxels before sampling. Likewise, for models learned from watertight meshes like Occupancy Networks, we sample from both reconstructed and GT watertight meshes following the original work [4]. This approach is used in the literature to avoid the inherent problems of the representations.

Figure 4.1 illustrates examples of original meshes, voxelized shapes, marching cubes applied to voxels, watertight meshes, and point cloud sampling on these surfaces. Figure 4.3 shows the preprocessing pipeline for one GT shape in ShapeNet to obtain the required representations for evaluation. Figure 4.4 shows the preprocessing pipeline for one GT shape in 3DCoMPaT++ to obtain the required representations for evaluation. This pipeline only differs from ShapeNet's pipeline in the first step since it requires an alignment to ShapeNet rotations first. Additionally, we discuss the efficient transfer of part segmentations across different mesh variations in this section.

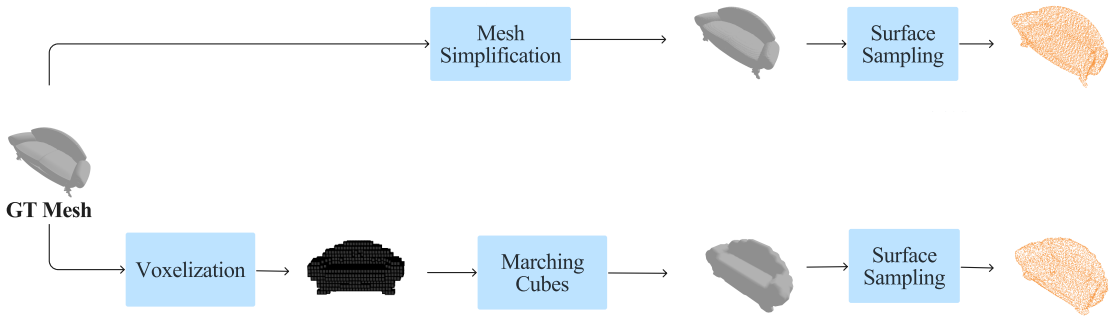


Figure 4.3. Preprocessing pipeline of GT shapes in ShapeNet.

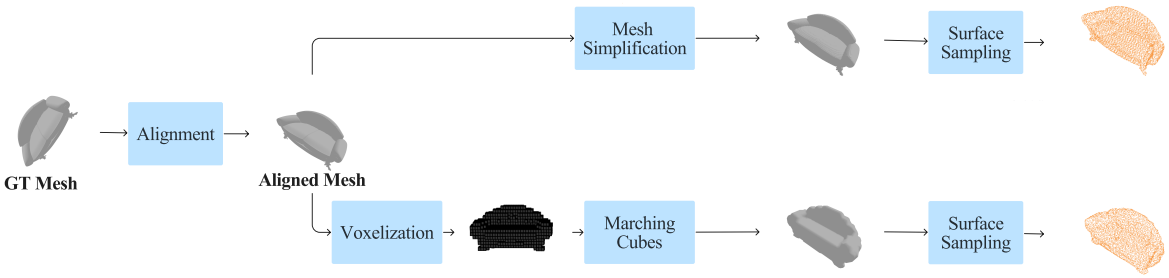


Figure 4.4. Preprocessing pipeline of GT shapes in 3DCoMPaT++.

4.2.1. Input

In our experiments with ShapeNet, we use renderings from 3D-R2N2 and modify them to increase variety. All input types consist of the original rendering, a variant with a plain black background, and versions where the object's scale is altered to 0.90, 0.75, and 0.50. We consistently use the first viewpoint for these renderings. Figure 4.5 shows an example of all input types we use in the following experiments.

In the case of our 3DCoMPaT++ experiments, we resize the renderings to 137 × 137 pixels to align with the dimensions of the ShapeNet renderings. For these experiments, we vary the input by using three different viewpoints and selecting three distinct shape textures. Figure 4.6 shows an example of all input types we use in the following experiments.

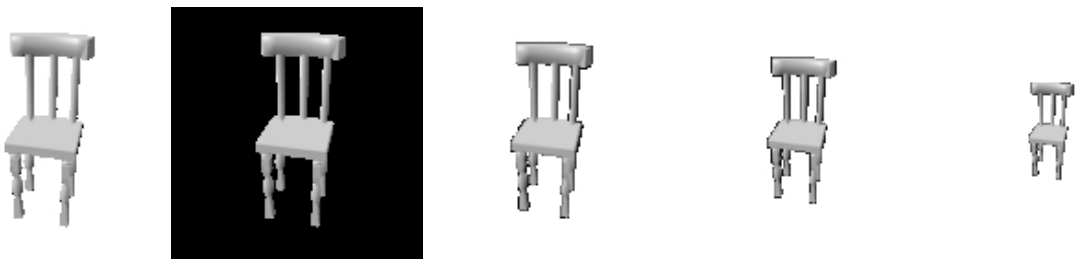


Figure 4.5. Input types for ShapeNet experiments.

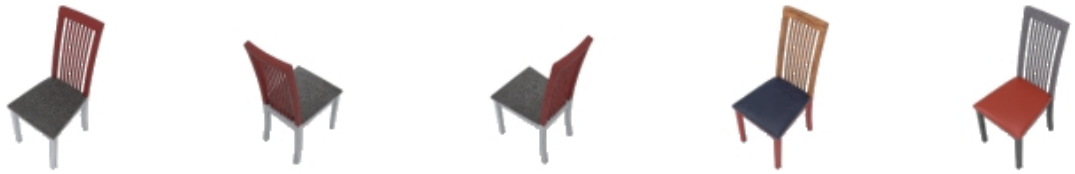


Figure 4.6. Input types for 3DCoMPaT++ experiments.

4.2.2. Voxels

For ShapeNet experiments, for Pix2Vox and 3D-C2FT, we use voxels provided readily. Since OccNet generates mesh, we generate voxel reconstructions to evaluate by voxel metrics. The binvox tool [79, 80], commonly used for voxelization, is sensitive to hardware and software setups. Therefore, for OccNet, we revoxelize both the GT meshes and reconstructions in the same setup and measure IoU from these new voxels for a more accurate evaluation.

We evaluate 3DCoMPaT++ only using point cloud metrics, hence we do not need to voxelize OccNet reconstructions. However, we still need voxel GT shapes to accurately assess voxel-based methods Pix2Vox and 3D-C2FT. Since we cannot access the voxelization setup used for ShapeNet, we perform voxelization at our setup. Although retraining Pix2Vox and 3D-C2FT with voxels from the same setup would be more accurate, the difference is not significant enough to justify the cost of retraining.

4.2.3. Watertight Shapes

Although the OccNet authors released points sampled from watertight meshes for ShapeNet, we also require watertight meshes for visualization. Therefore, we apply the mesh simplification pipeline proposed by Stutz and Geiger [81] following OccNet authors. We apply the same pipeline for both ShapeNet and 3DCoMPat++. Figure 4.7 illustrates three examples of mesh simplification, showing that watertight meshes have simpler, smoother surfaces and removed holes.

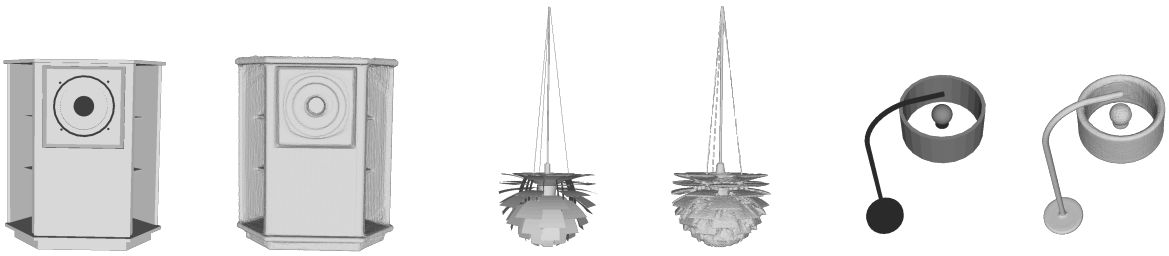


Figure 4.7. Mesh simplification example for categories: speaker and lamp.

4.2.4. Transferring Part Annotation

In our 3DCoMPaT++ experiments, we have part annotations for the GT meshes. However, two main issues arise. First, voxel and watertight reconstructions cannot be directly compared to the GT mesh without the processing steps outlined in Sections 4.2.3 and 4.2.2. This is because when a mesh form is transformed into voxel surfaces or watertight meshes, part information tends to be lost in these new formats. Second, when considering parts separately and sampling from part surfaces, some points may end up inside the overall mesh surface. To address these issues, we annotate points on the target mesh (whether voxel surface or watertight) based on the closest point's part label in the original mesh. This method effectively transfers part labels and eliminates points lying inside the mesh. The reason is that the search for annotation points is directed from the target mesh to the original mesh, and the target mesh only contains surface points.

To perform this annotation process efficiently, we utilize a k-d tree, a space-partitioning data structure for organizing points in a k-dimensional space. This tree structure significantly accelerates the search for the nearest neighbors in our point cloud. We employ the Fast Library for Approximate Nearest Neighbors (FLANN) [82] for this search. FLANN optimizes the search process within the k-d tree, providing a balance between search accuracy and computational efficiency. This combination of a k-d tree and FLANN search allows for a fast transfer from the original to the target mesh. We implement the pipeline using Open3D library [83].

Figure 4.8 illustrates the updates in the preprocessing pipeline in Figure 4.4 for part annotation transfer. Our qualitative evaluations confirm the effectiveness of our approach in maintaining accurate part annotations while eliminating internal points. Figure 4.9 illustrates one example of this qualitative assessment for each selected category.

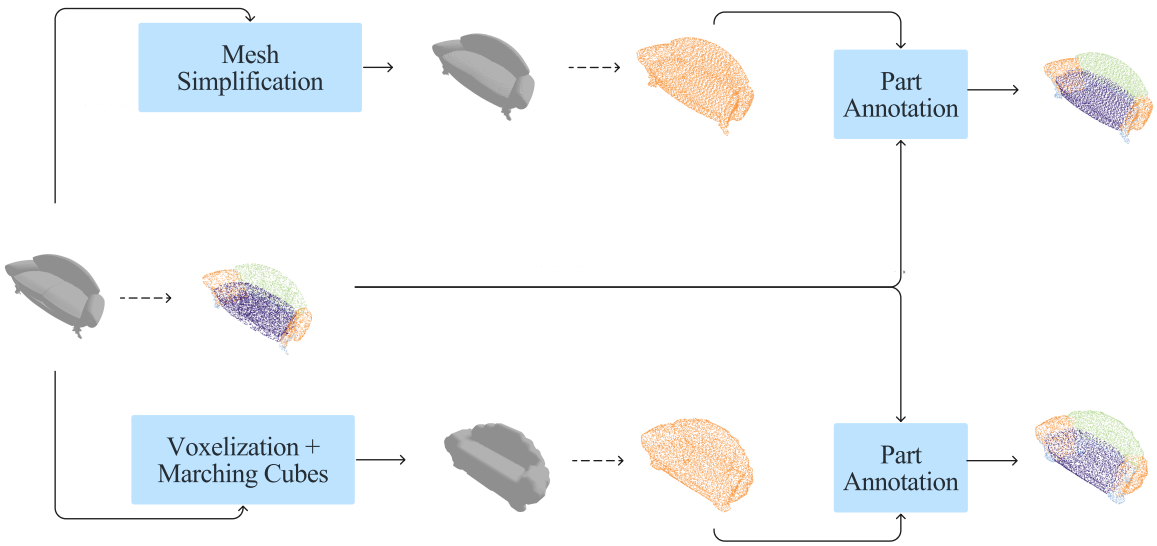


Figure 4.8. Updates in the preprocessing pipeline for part annotation transfer.

4.2.5. Baselines

We utilize the released pretrained versions of Pix2Vox and 3D-C2FT for our experiments. Although we attempted to retrain OccNet by training it over 500K iterations, we encountered some issues. Even though the loss approached that of the pretrained version and ceased to improve further, the performance on ShapeNet did not match the reported results. Consequently, we chose to use the pretrained version of OccNet. It is important to note that the data split for OccNet is different from that of the other two models. Therefore, we compare each model using their respective test sets in ShapeNet experiments. For 3DCoMPaT++ experiments, all shapes are unseen for all three models, hence we use the same set.

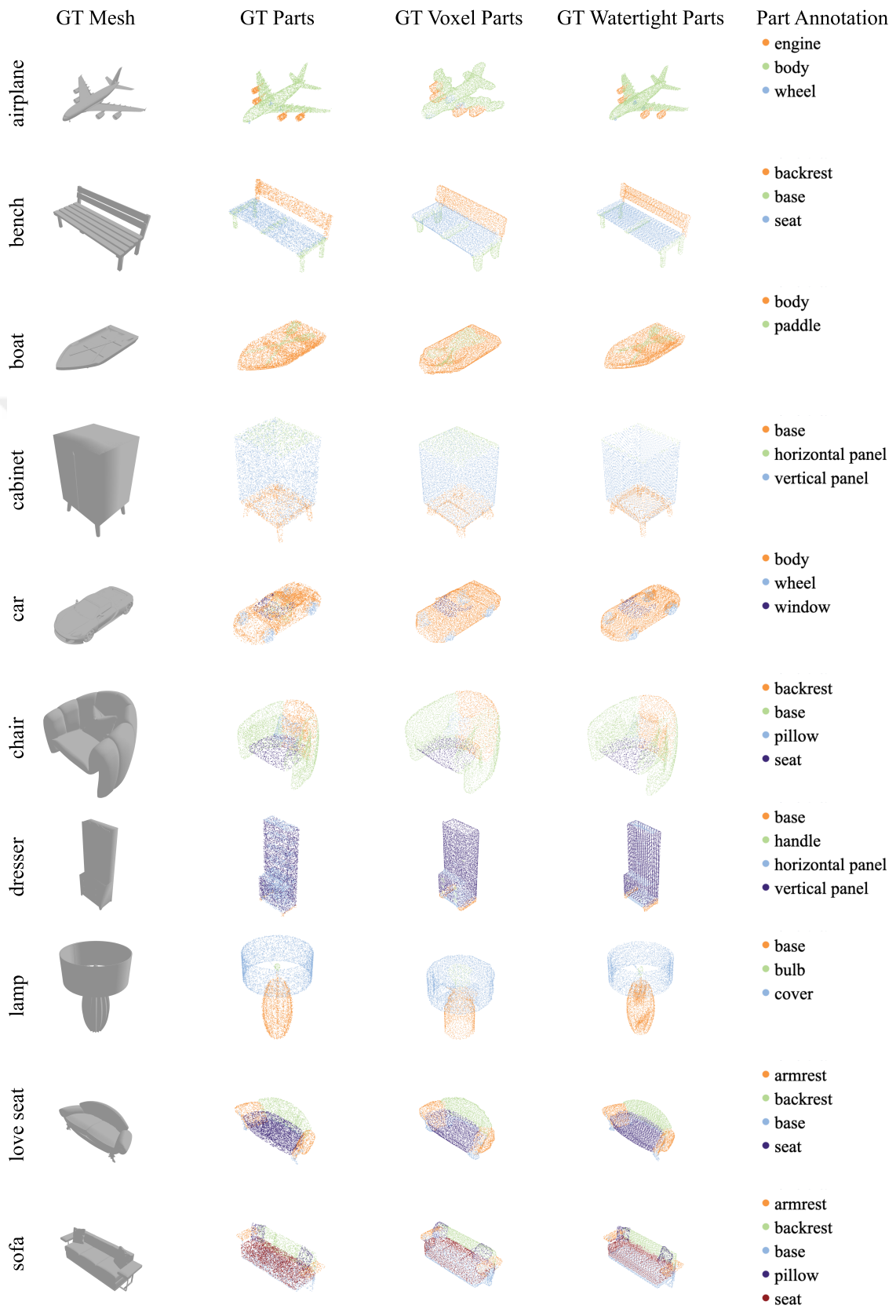


Figure 4.9. Part annotation transfer example for voxel surface and watertight mesh.

4.3. Evaluation Metrics

We evaluate by using voxel intersection over union (IoU) and point cloud metrics: F-Score and Chamfer Distance. In this section, we explain them in detail and also show the adaptation of F-Score for part-based evaluation.

4.3.1. Voxel Metrics

Volumetric Intersection over Union (IoU) is a metric utilized for evaluating the accuracy of 3D reconstructions in the context of voxel-based models. This metric measures the ratio of the intersection of correctly occupied voxels (those identified accurately as part of the object) to the union of all occupied voxels (both correctly and incorrectly identified). It is particularly useful in assessing the match between a reconstructed shape and the original model in voxel-based systems.

This measurement is straightforward, as it directly compares the occupied spaces in the voxel grid of the reconstructed shape with those in the original shape. In the context of voxel reconstruction, instead of directly assigning binary values, an occupancy probability is usually assigned to each voxel. These probabilities are then converted to binary values using a predetermined threshold. IoU is computed as

$$\text{IoU} = \frac{V_{\text{intersection}}}{V_{\text{union}}}, \tag{4.1}$$

$$V_{\text{intersection}} = \sum_{i=1}^{N} [G_i > th] \cdot [R_i > \text{th}], \tag{4.2}$$

$$V_{\text{union}} = \sum_{i=1}^{N} \max\{[G_i > th], [R_i > \text{th}]\}, \tag{4.3}$$

where th is the voxel occupancy threshold, G is the GT voxel volume and P is the reconstructed voxel volume. For voxel reconstruction models, Pix2Vox and 3D-C2FT, we set the threshold at 0.3 for ShapeNet experiments and 0.2 for the 3DCoMPaT++ experiments.

4.3.2. Point Cloud Metrics

In this section, we explain two point cloud evaluation metrics: F-Score and Chamfer Distance. We also adapt F-Score for part evaluation and name this metric Part F-Score.

4.3.2.1. F-Score and Part F-Score. The F-score metric compares reconstructed points with GT points, focusing on those within a specified distance threshold d. It is a bidirectional metric: precision measures the number of reconstructed points found within the GT points' search radius, while recall measures the number of GT points found within the reconstructed points' search radius. The F-Score is a combination of both precision and recall, using the L2 distance as the distance function. F-Score is formulated as

$$F\text{-}Score(d) = \frac{2 \times P(d) \times R(d)}{P(d) + R(d)}, \tag{4.4}$$

$$P(d) = \frac{1}{n_R} \sum_{r \in R} \left[\min_{g \in G} \|g - r\| < d \right], \tag{4.5}$$

$$R(d) = \frac{1}{n_G} \sum_{g \in G} \left[\min_{r \in R} \|g - r\| < d \right], \tag{4.6}$$

where d is the distance threshold, G is the GT point cloud and P is the reconstructed point cloud. For simplicity in the experimental results tables, we refer to this score as FS for short.

In our adaptation of the formula for part-based evaluation, we consider the entire reconstructed point cloud while selecting GT point clouds at the part level. Thus, the complete reconstructed point cloud serves as a candidate for the part point clouds. In practice, due to the carefully selected distance threshold, only points nearby can contribute. In this adapted formula, we cap precision at 1. This adjustment addresses scenarios where poor reconstruction quality results in unbalanced point densities. In such cases, the number of points aligning with the part surface may surpass the total number of sampled points in the part point cloud. By limiting the maximum precision to 1, we aim to prevent skewed evaluations that could arise from these density imbal-

ances. For both F-Score metrics, we set the distance threshold to 0.01 following Xie et. al. [3], [25]. We define part-level scoring as

$$F\text{-}Score_{part}(d) = \frac{2 \times P_{part}(d) \times R_{part}(d)}{P_{part}(d) + R_{part}(d)}, \tag{4.7}$$

$$P_{part}(d) = min(\frac{1}{n_{Gpart}} \sum_{r \in R} \left[\min_{g \in G_{part}} \|g - r\| < d \right], 1), \tag{4.8}$$

$$R_{part}(d) = \frac{1}{n_{Gpart}} \sum_{g \in G_{part}} \left[\min_{r \in R} \|g - r\| < d \right], \tag{4.9}$$

where d is the distance threshold, G_{part} is the part point cloud and P is the reconstructed point cloud.

4.3.2.2. Chamfer Distance. Chamfer Distance, or Chamfer-L2-distance, measures the two-way distance between the ground truth and reconstructed points. It calculates the average squared distance from each ground truth point to its nearest point in the reconstruction, and vice versa, providing a comprehensive assessment of the reconstruction's accuracy. It is defined as

$$D_{\text{Chamfer}}(G, R) = \frac{1}{|G|} \sum_{g \in G} \min_{r \in R} \|g - r\|^2 + \frac{1}{|R|} \sum_{r \in R} \min_{g \in G} \|g - r\|^2, \tag{4.10}$$

where G and P are the GT and reconstructed point clouds respectively.

4.4. ShapeNet Experiments

4.4.1. Original ShapeNet Test

We begin by examining the optimal scenario, which involves using data directly from ShapeNet and the models' unseen test set. The results, as presented in Table 4.2, indicate that Pix2Vox generally outperforms other models in most categories, with OccNet being a close second. However, it's important to note that the numerical differences between the models are quite narrow, as evidenced by the variation in metrics used to determine the best category for each model. It's also worth mentioning that the base papers report differing results based on their retraining approaches, making it challenging to select the best-performing model even from those studies. This incon-

sistency underscores the need for further investigation into the models' generalization capabilities and robustness within enriched benchmarks. We also note that although the best results change across different metrics for some categories, all metrics showcase a similar pattern with close assessments. Figure 4.10 and 4.11 provides an example reconstruction for each category.

Table 4.2. ShapeNet test results of original input images.

Category	OccNet			Pix2Vox			3D-C2FT		
	CD	FS	IoU	CD	FS	IoU	CD	FS	IoU
airplane	0.164	**0.599**	0.555	**0.104**	0.564	**0.684**	0.154	0.510	0.640
bench	0.211	**0.457**	0.551	**0.179**	0.450	**0.616**	0.406	0.381	0.548
cabinet	**0.162**	0.359	**0.797**	0.208	**0.390**	0.792	0.302	0.358	0.761
car	0.130	0.429	0.828	**0.080**	**0.536**	**0.855**	0.098	0.496	0.832
chair	0.292	**0.281**	**0.577**	**0.268**	0.270	0.567	0.360	0.237	0.525
display	**0.439**	0.260	**0.580**	0.458	**0.272**	0.537	0.686	0.234	0.489
lamp	1.498	**0.321**	0.447	**0.788**	0.302	0.443	0.793	0.312	**0.468**
rifle	**0.138**	0.541	0.591	0.163	**0.542**	**0.615**	0.282	0.513	0.583
sofa	**0.192**	0.335	**0.745**	0.281	**0.355**	0.709	0.407	0.313	0.668
speaker	0.454	0.232	0.708	**0.430**	**0.267**	**0.714**	0.520	0.256	0.697
table	**0.289**	**0.401**	0.582	**0.289**	0.356	**0.601**	0.367	0.330	0.571
telephone	**0.133**	0.536	0.758	0.182	**0.555**	**0.776**	0.334	0.498	0.718
watercraft	**0.228**	**0.377**	0.540	0.271	0.370	**0.595**	0.395	0.342	0.557
Total	0.295	0.397	0.639	**0.251**	**0.405**	**0.661**	0.334	0.371	0.629

4.4.2. Adding Background Color

We examine a simple scenario, where we change the input images' background from the default white to a plain black color. The scores are presented in Table 4.3. We observe a significant performance drop for Pix2Vox and OccNet. Although in rare cases the object is still recognizable, across most categories, their overall performance is undoubtedly reduced. Interestingly, they are still capable of reconstructing shapes

that are mostly rectangular prism-like, such as cabinets, speakers, sofas, and telephones. This suggests a specific resilience in handling certain geometries or simply these models predict black regions, or any colored regions, as occupied. In contrast, 3D-C2FT demonstrates a robust baseline for inputs with varying backgrounds. It outperforms in all categories across almost all metrics using two metrics. It is a very close second in two categories (cabinet and speaker) when measured with IoU. Despite being the least effective in the vanilla test case, this scenario highlights 3D-C2FT's superior robustness compared to the other models. This unexpected result is an indication of the need for better assessments in the current reconstruction benchmarks. Figure 4.10 and 4.11 illustrate the variations in reconstruction performance for all categories when compared to the vanilla case.

Table 4.3. ShapeNet test results of images with black background.

Category	OccNet			Pix2Vox			3D-C2FT		
	CD	FS	IoU	CD	FS	IoU	CD	FS	IoU
airplane	5.331	0.029	0.060	5.423	0.046	0.044	**0.829**	**0.262**	**0.374**
bench	2.250	0.061	0.156	2.796	0.067	0.131	**1.707**	**0.177**	**0.282**
cabinet	1.224	0.061	0.488	1.179	0.100	**0.496**	**1.105**	**0.161**	0.488
car	3.934	0.045	0.214	3.466	0.045	0.199	**0.325**	**0.282**	**0.671**
chair	1.971	0.051	0.237	1.949	0.068	0.208	**0.702**	**0.152**	**0.382**
display	**1.935**	0.052	0.272	2.541	0.058	0.231	2.054	**0.138**	**0.311**
lamp	4.741	0.040	0.136	3.669	0.063	0.162	**1.200**	**0.239**	**0.382**
rifle	9.501	0.017	0.025	9.523	0.021	0.016	**1.540**	**0.325**	**0.380**
sofa	1.496	0.074	0.390	2.142	0.087	0.342	**1.160**	**0.173**	**0.461**
speaker	1.399	0.065	**0.499**	1.472	0.094	0.495	**1.259**	**0.139**	0.488
table	2.339	0.062	0.191	2.095	0.078	0.222	**1.233**	**0.160**	**0.321**
telephone	1.710	0.058	0.279	1.986	0.066	0.270	**1.110**	**0.237**	**0.449**
watercraft	6.005	0.034	0.094	5.849	0.040	0.088	**0.865**	**0.216**	**0.406**
Total	3.346	0.050	0.213	3.252	0.063	0.205	**0.985**	**0.207**	**0.429**

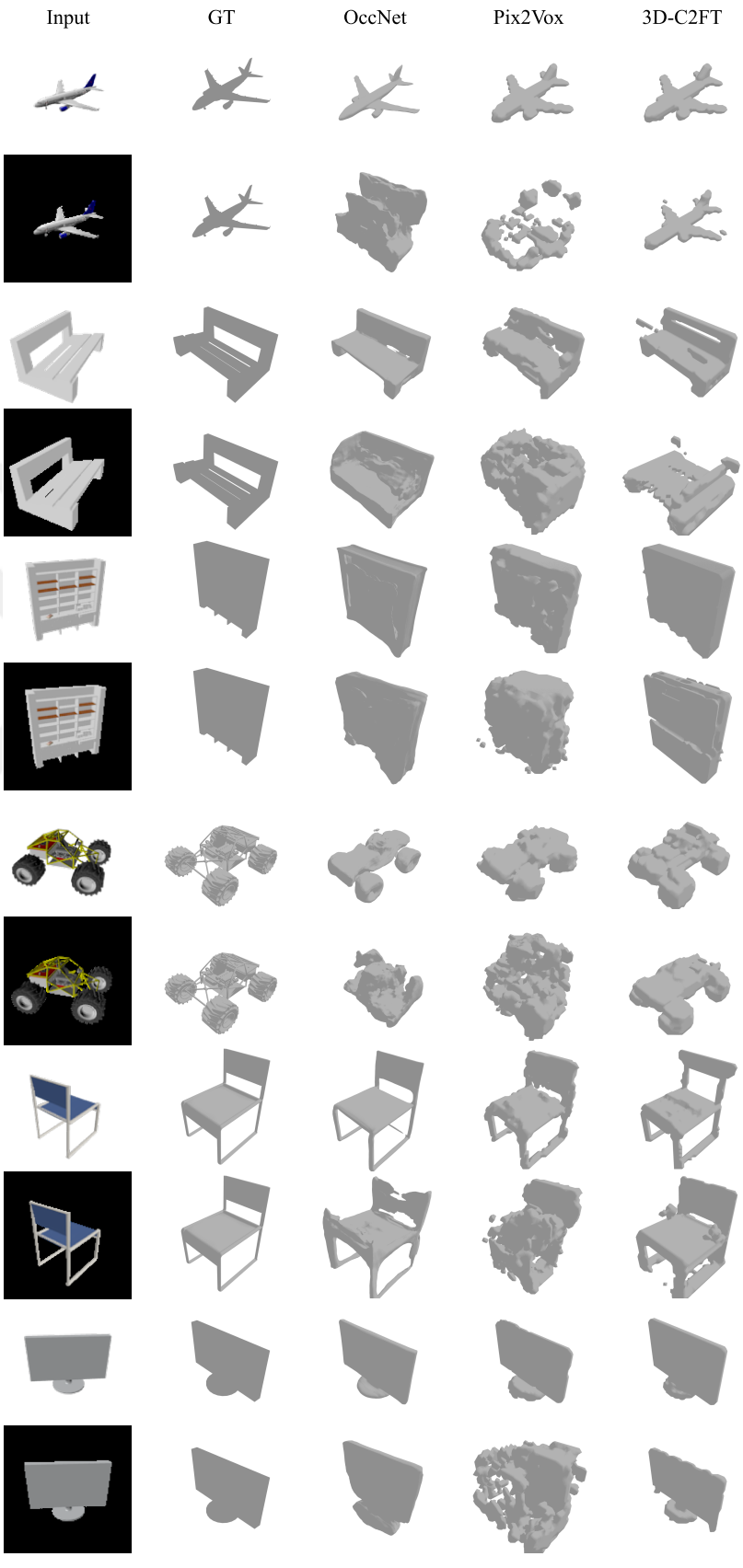


Figure 4.10. Reconstruction examples of images with the black background.



Figure 4.11. Reconstruction examples of images with the black background.

4.4.3. Changing Object Size in Rendering

ShapeNet provides shape renderings with objects of approximately the same size. In this experiment, we aim to explore the effect of object size changes on reconstruction by scaling the objects in 2D renderings by 0.90, 0.75, and 0.50. The impact of this scaling process can be observed in Figure 4.5. Pix2Vox and 3D-C2FT, which reconstruct voxels in the same 32^3 space, do not require resizing the reconstructed shapes. In contrast, OccNet operates in a free reconstruction space. We observe that it allows for the reconstruction of smaller-sized objects at their respective scales. To mitigate this size variability, we scale both the GT and reconstructed shapes to fit within a unit cube and align their center to the origin.

Table 4.4. ShapeNet test results of input images scaled by 90%.

Category	OccNet			Pix2Vox			3D-C2FT		
	CD	FS	IoU	CD	FS	IoU	CD	FS	IoU
airplane	0.135	**0.544**	0.542	**0.118**	0.490	**0.633**	0.152	0.458	0.603
bench	**0.215**	0.401	0.535	0.228	**0.408**	**0.576**	0.426	0.365	0.527
cabinet	**0.167**	0.318	**0.793**	0.254	**0.364**	0.765	0.373	0.339	0.742
car	0.132	0.363	0.835	**0.084**	**0.519**	**0.847**	0.094	0.484	0.826
chair	**0.252**	0.231	**0.569**	0.282	**0.251**	0.550	0.369	0.225	0.514
display	**0.406**	**0.290**	**0.578**	0.506	0.249	0.519	0.663	0.238	0.486
lamp	1.181	0.232	**0.462**	0.767	0.292	0.437	**0.748**	**0.302**	0.461
rifle	**0.124**	**0.589**	0.587	0.170	0.511	**0.592**	0.239	0.487	0.564
sofa	**0.203**	0.302	**0.743**	0.305	**0.338**	0.695	0.411	0.308	0.663
speaker	0.479	0.190	0.711	**0.456**	**0.264**	**0.712**	0.516	0.242	0.690
table	**0.289**	0.286	0.556	0.300	**0.316**	**0.558**	0.371	0.300	0.535
telephone	**0.133**	**0.546**	0.748	0.195	0.535	**0.762**	0.316	0.496	0.719
watercraft	**0.223**	**0.376**	0.548	0.256	0.366	**0.590**	0.344	0.340	0.560
Total	0.270	0.342	0.633	**0.263**	**0.377**	**0.638**	0.331	0.352	0.612

Table 4.5. ShapeNet test results of input images scaled by 75%.

Category	OccNet			Pix2Vox			3D-C2FT		
	CD	FS	IoU	CD	FS	IoU	CD	FS	IoU
airplane	0.210	**0.513**	0.519	0.189	0.421	0.559	**0.186**	0.431	**0.575**
bench	**0.390**	**0.359**	**0.463**	0.502	0.302	0.448	0.623	0.312	0.459
cabinet	**0.202**	**0.256**	**0.760**	0.521	0.250	0.652	0.538	0.244	0.647
car	0.136	0.330	**0.818**	**0.124**	**0.421**	0.789	0.126	0.391	0.770
chair	**0.342**	0.209	**0.533**	0.468	**0.210**	0.474	0.494	0.193	0.462
display	**0.566**	**0.272**	**0.547**	0.805	0.213	0.447	1.146	0.208	0.429
lamp	1.971	0.187	0.410	1.084	0.253	0.381	**0.966**	**0.271**	**0.428**
rifle	**0.144**	**0.601**	**0.576**	0.184	0.490	0.564	0.266	0.460	0.531
sofa	**0.278**	**0.271**	**0.707**	0.489	0.265	0.612	0.487	0.261	0.611
speaker	**0.558**	0.160	**0.681**	0.709	0.186	0.628	0.723	**0.196**	0.633
table	**0.405**	**0.268**	**0.476**	0.478	0.223	0.423	0.486	0.221	0.419
telephone	**0.205**	**0.468**	**0.706**	0.363	0.412	0.656	0.643	0.391	0.612
watercraft	**0.235**	**0.370**	0.526	0.245	0.347	**0.567**	0.282	0.329	0.549
Total	**0.380**	**0.315**	**0.593**	0.412	0.305	0.555	0.439	0.296	0.549

We test three different scales: 0.90, 0.75, and 0.50. Quantitative results are provided in Table 4.4, Table 4.5, and Table 4.6 respectively. For the 0.90 and 0.75 scales, a minor decrease in the performance of all three models is observed. This decrease is considered insignificant due to the distortion and loss of fine details in the input images. However, the 0.50 scale results in a substantial loss of information, which is also expected as this scale significantly reduces structural details in the images. In this scale, OccNet outperforms the other two models. Example reconstructions with the 0.75 scale, alongside the original reconstructions, are shown in Figure 4.13 and 4.14. A chair reconstruction example at the three selected scales is given in Figure 4.12.

Table 4.6. ShapeNet test results of input images scaled by 50%.

Category	OccNet			Pix2Vox			3D-C2FT		
	CD	FS	IoU	CD	FS	IoU	CD	FS	IoU
airplane	1.784	**0.264**	**0.311**	2.023	0.117	0.130	**0.982**	0.187	0.262
bench	4.119	**0.133**	**0.147**	**3.409**	0.101	0.126	4.040	0.088	0.126
cabinet	**2.659**	**0.077**	**0.393**	3.984	0.043	0.172	4.269	0.041	0.203
car	**0.676**	**0.158**	**0.564**	1.135	0.081	0.331	0.853	0.081	0.373
chair	**2.945**	**0.082**	**0.215**	4.362	0.054	0.089	3.153	0.064	0.145
display	**4.650**	**0.094**	**0.229**	5.480	0.052	0.105	6.229	0.046	0.104
lamp	7.069	0.059	0.123	5.007	0.068	0.075	**3.431**	**0.107**	**0.171**
rifle	**0.378**	**0.486**	**0.448**	0.525	0.272	0.304	0.749	0.284	0.322
sofa	**2.155**	**0.107**	**0.379**	3.003	0.063	0.183	2.924	0.071	0.210
speaker	**3.204**	**0.065**	**0.404**	4.647	0.035	0.187	3.922	0.043	0.248
table	3.752	**0.101**	**0.133**	3.519	0.060	0.057	**3.200**	0.064	0.078
telephone	3.671	**0.108**	**0.229**	**3.040**	0.077	0.180	5.458	0.050	0.127
watercraft	0.886	**0.255**	**0.399**	1.122	0.165	0.296	**0.606**	0.201	0.369
Total	2.672	**0.148**	**0.307**	2.968	0.085	0.166	**2.561**	0.098	0.211

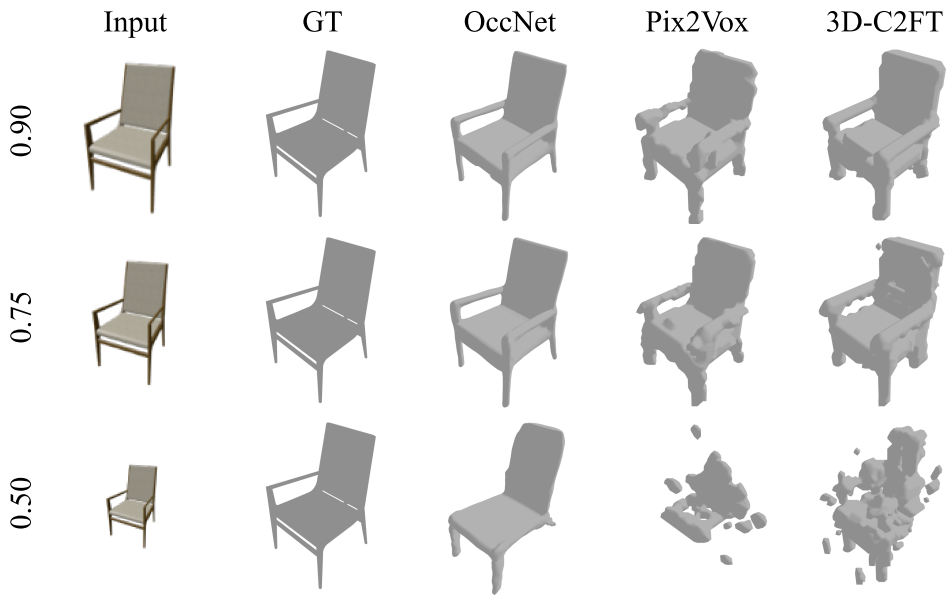


Figure 4.12. Reconstruction examples from the chairs category at the selected scales.

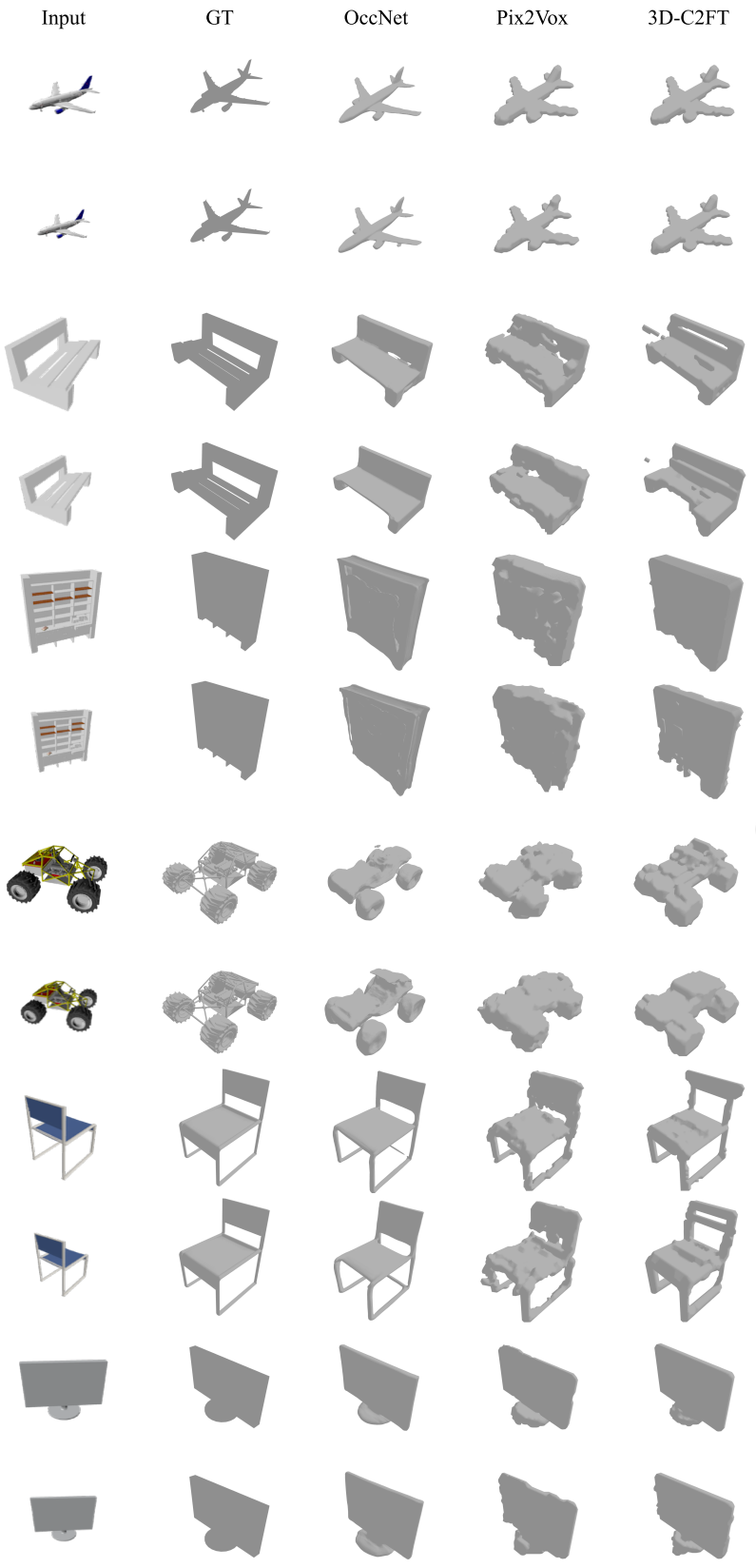


Figure 4.13. Reconstruction examples of original input and scaled by 0.75.

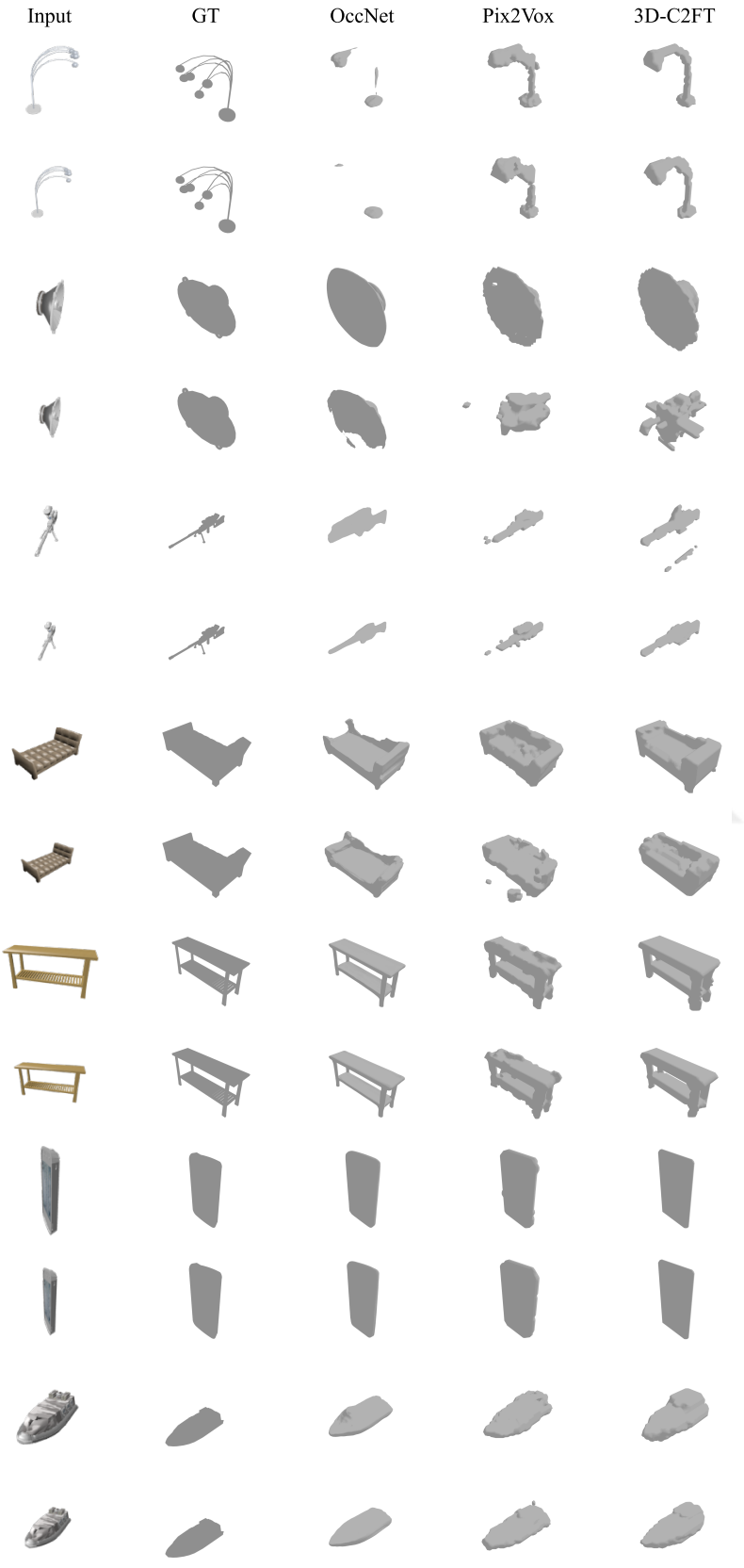


Figure 4.14. Reconstruction examples of original input and scaled by 0.75.

4.5. 3DCoMPaT++ Experiments

4.5.1. Changing Input Viewpoint

This experiment serves as both an ablation study and a zero-shot benchmark, considering that the dataset, viewpoints, and textures are unseen. We first assess the robustness of our selected baseline models against renderings from different viewpoints. For Viewpoints 0 and 1, we select render indices 0 and 1 from 3DCoMPaT++. They are mirrored views of each other for most objects and can be considered ideal since they closely resemble the renderings in ShapeNet. For viewpoint 2, we select one of the random viewpoints that has a render index of 4 in the 3DCoMPaT++ dataset. It looks at the objects from above compared to the other two and is significantly more challenging considering renderings in ShapeNet. Figure 4.15 shows rendering examples of the selected viewpoints in two categories to give an insight into the selected camera positions.



Figure 4.15. Examples of the selected viewpoints from two categories.

The results indicate that the baseline models perform better with the first two viewpoints. From this result, it is important to note that simply mirroring a rendered input does not significantly affect performance, as evidenced by the data in Table 4.7 and illustrated in Figure 4.16, 4.17, and 4.18. Our findings suggest that informative viewpoints influence performance reasonably well. When comparing the models with each other, we first observe that Pix2Vox performs worse than the other two in this benchmark. We find this interesting considering that the vanilla test case in Section 4.4.1 indicates its superior performance. The overall performances of OccNet and 3D-C2FT are close to each other. However, all models reconstruct significantly worse than the original test, hence indicating a poor adaptation to novel tests. The success of viewpoint 2 in the lamp category is unexpected. We argue its implications since the performance across viewpoints in lamps is close to each other and lower than others.

Table 4.7. 3DCoMPaT++ F-Score@0.01 for 3 Different Viewpoints. Bold values show the best result for the model. Underlined values show the best result across all models.

Category	OccNet			Pix2Vox			3D-C2FT		
	V0	V1	V2	V0	V1	V2	V0	V1	V2
airplane	**<u>0.499</u>**	0.464	0.356	0.151	**0.206**	0.146	0.302	**0.319**	0.261
bench	**<u>0.209</u>**	0.199	0.150	**0.147**	0.140	0.104	**0.192**	0.169	0.132
boat	**0.230**	0.223	0.166	0.177	**0.195**	0.154	0.206	**<u>0.234</u>**	0.178
cabinet	**<u>0.183</u>**	0.154	0.131	0.114	**0.115**	0.099	**0.177**	0.168	0.140
car	**0.261**	0.232	0.128	0.199	**0.210**	0.140	**<u>0.275</u>**	0.201	0.167
chair	**<u>0.128</u>**	0.079	0.102	**0.086**	0.072	0.083	**0.113**	0.075	0.093
dresser	0.147	**<u>0.159</u>**	0.145	0.088	**0.098**	0.091	0.130	**0.133**	0.120
lamp	0.102	0.104	**0.112**	0.079	0.080	**0.085**	0.150	**<u>0.155</u>**	0.146
love seat	0.150	**0.157**	0.129	**0.138**	0.131	0.109	**<u>0.187</u>**	0.174	0.141
sofa	**0.209**	0.202	0.162	**0.174**	0.147	0.122	**<u>0.217</u>**	**<u>0.217</u>**	0.162
Total	**<u>0.159</u>**	0.139	0.128	**0.112**	0.105	0.097	**0.158**	0.141	0.127

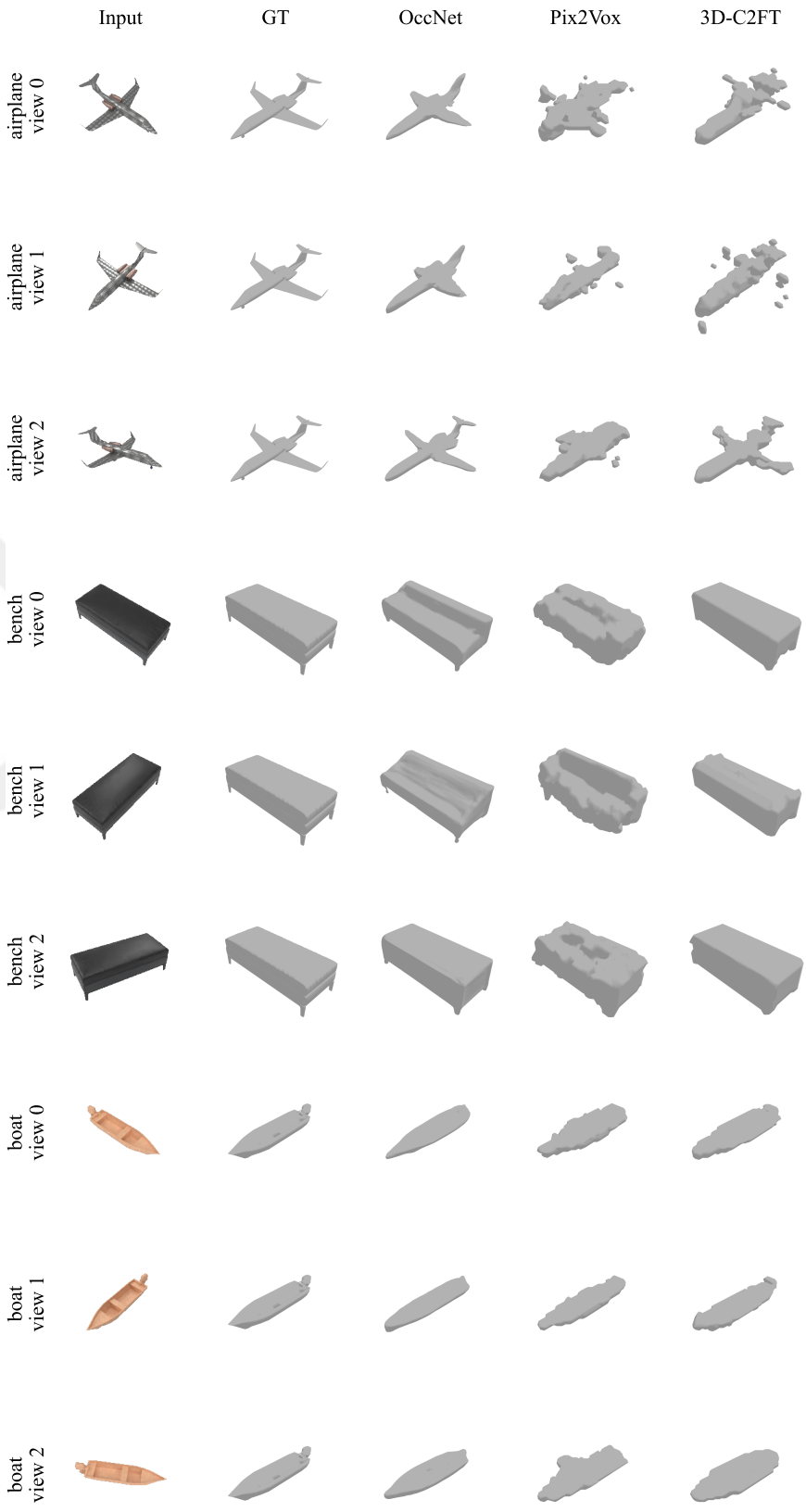


Figure 4.16. Reconstruction examples of images from 3 viewpoints.

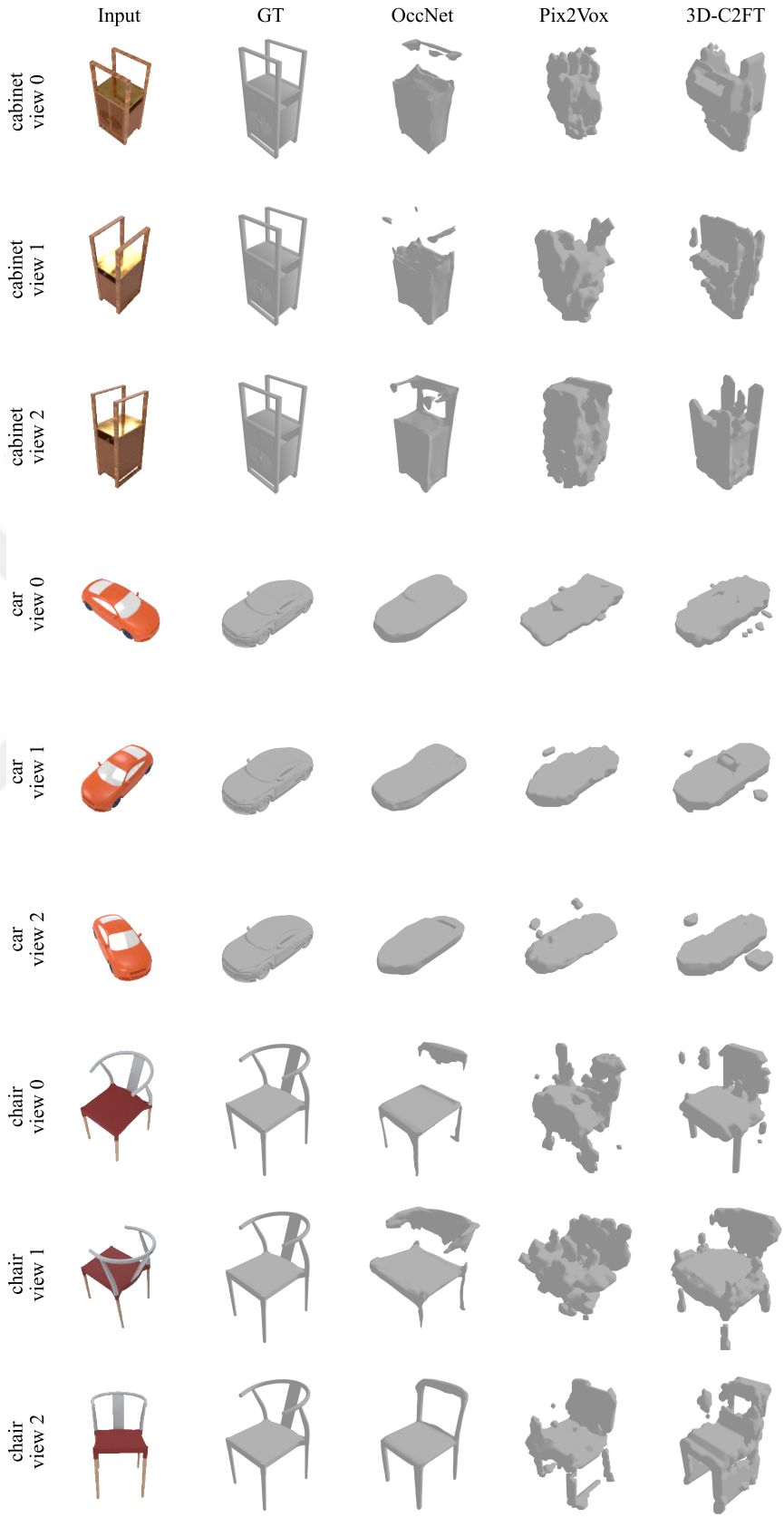


Figure 4.17. Reconstruction examples of images from 3 viewpoints.

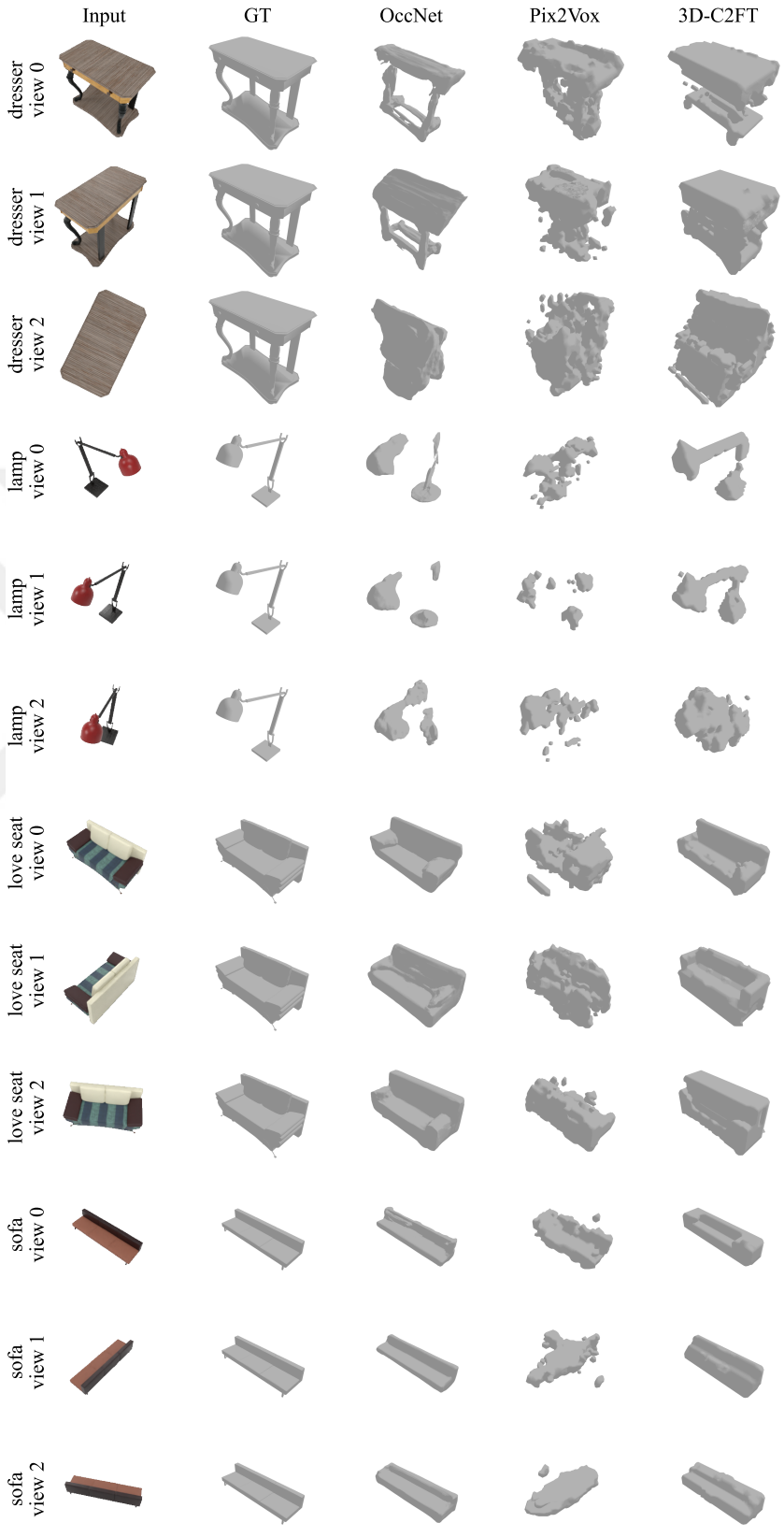


Figure 4.18. Reconstruction examples of images from 3 viewpoints.

4.5.2. Changing Input Texture

Thanks to the rich renderings available in the 3DCoMPaT++ dataset, we choose renderings with three realistic, different textures from the 100 available options. F-Score for these selections from the best performing first viewpoint is reported in Table 4.8. The performance across different textures did not show significant variation. Therefore, we believe it is reasonable to claim that the models are capable of recognizing the object structure across various textures. Still, similar to the observations in Section 4.5.1, we note that the models perform very poorly compared to the original case, especially Pix2Vox which is the best in the vanilla test and the worst in novel benchmarks. Figure 4.19, 4.20, and 4.21 show one example for each category in the selected three textures.

Table 4.8. 3DCoMPaT++ F-Score@0.01 results for 3 different object textures. Bold values show the best result for the model. Underlined values show the best result across all models.

Category	OccNet			Pix2Vox			3D-C2FT		
	S0	S1	S2	S0	S1	S2	S0	S1	S2
airplane	**0.499**	0.494	0.485	**0.151**	0.141	0.127	0.302	**0.316**	0.293
bench	0.209	0.210	**0.215**	0.147	**0.150**	0.145	0.192	0.185	**0.193**
boat	**0.230**	0.220	0.224	**0.177**	0.165	0.176	0.206	0.222	**0.231**
cabinet	0.183	0.182	**0.192**	**0.114**	0.110	0.113	0.177	0.173	**0.182**
car	0.261	0.247	**0.264**	0.199	0.197	**0.200**	0.275	0.275	**0.276**
chair	0.128	**0.129**	**0.129**	**0.086**	**0.086**	**0.086**	0.113	0.114	**0.115**
dresser	**0.147**	0.141	**0.147**	0.088	**0.095**	0.089	**0.130**	0.124	0.124
lamp	**0.102**	0.101	0.101	0.079	0.081	**0.082**	0.150	**0.152**	0.151
love seat	0.150	**0.162**	**0.162**	0.138	**0.141**	0.133	0.187	**0.191**	0.187
sofa	**0.209**	0.207	**0.209**	**0.174**	0.169	0.170	**0.217**	0.211	0.215
Total	0.159	0.159	**0.162**	**0.112**	**0.112**	0.110	0.158	0.156	**0.159**

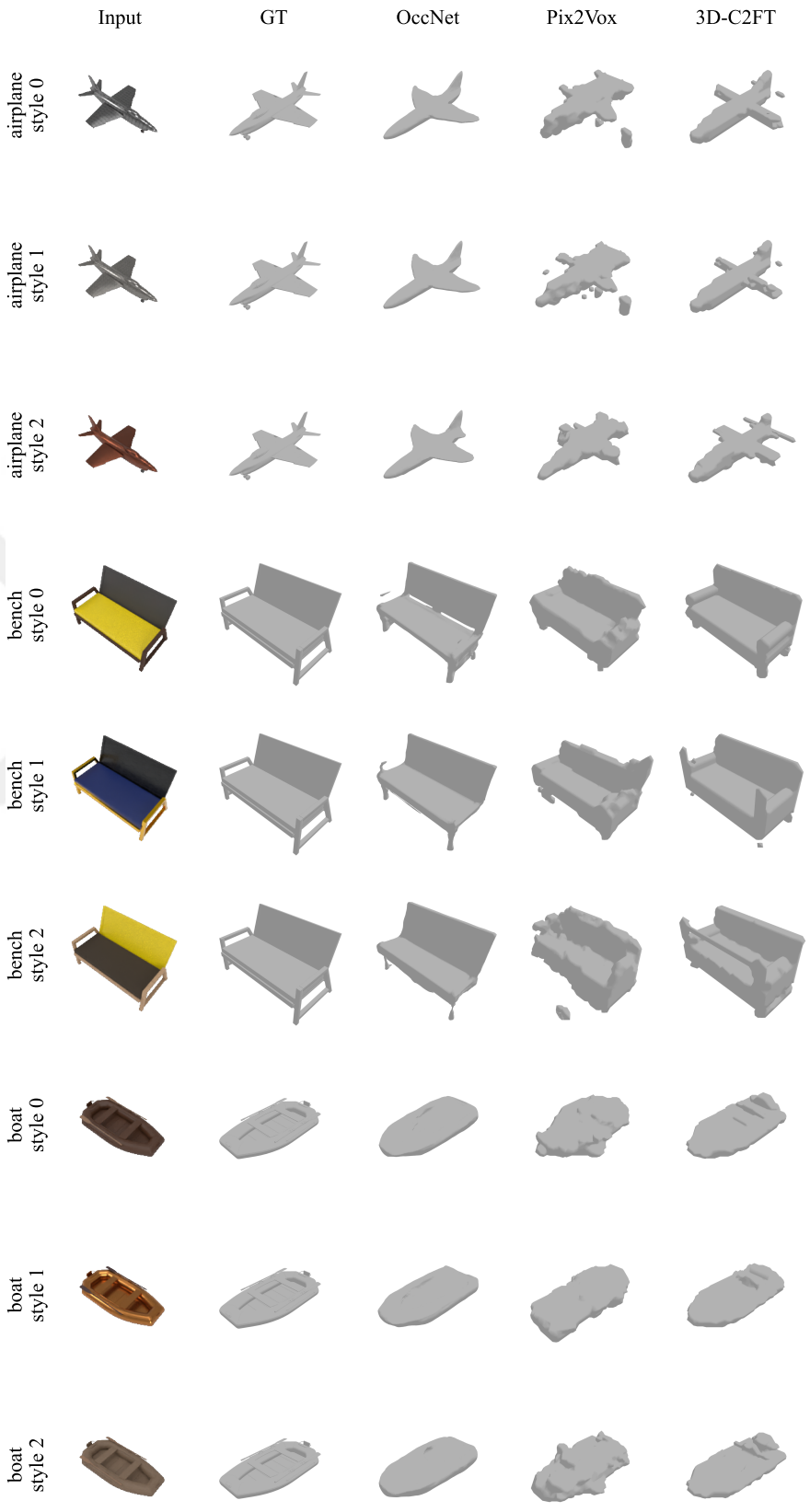


Figure 4.19. Reconstruction examples of images with 3 different object textures.

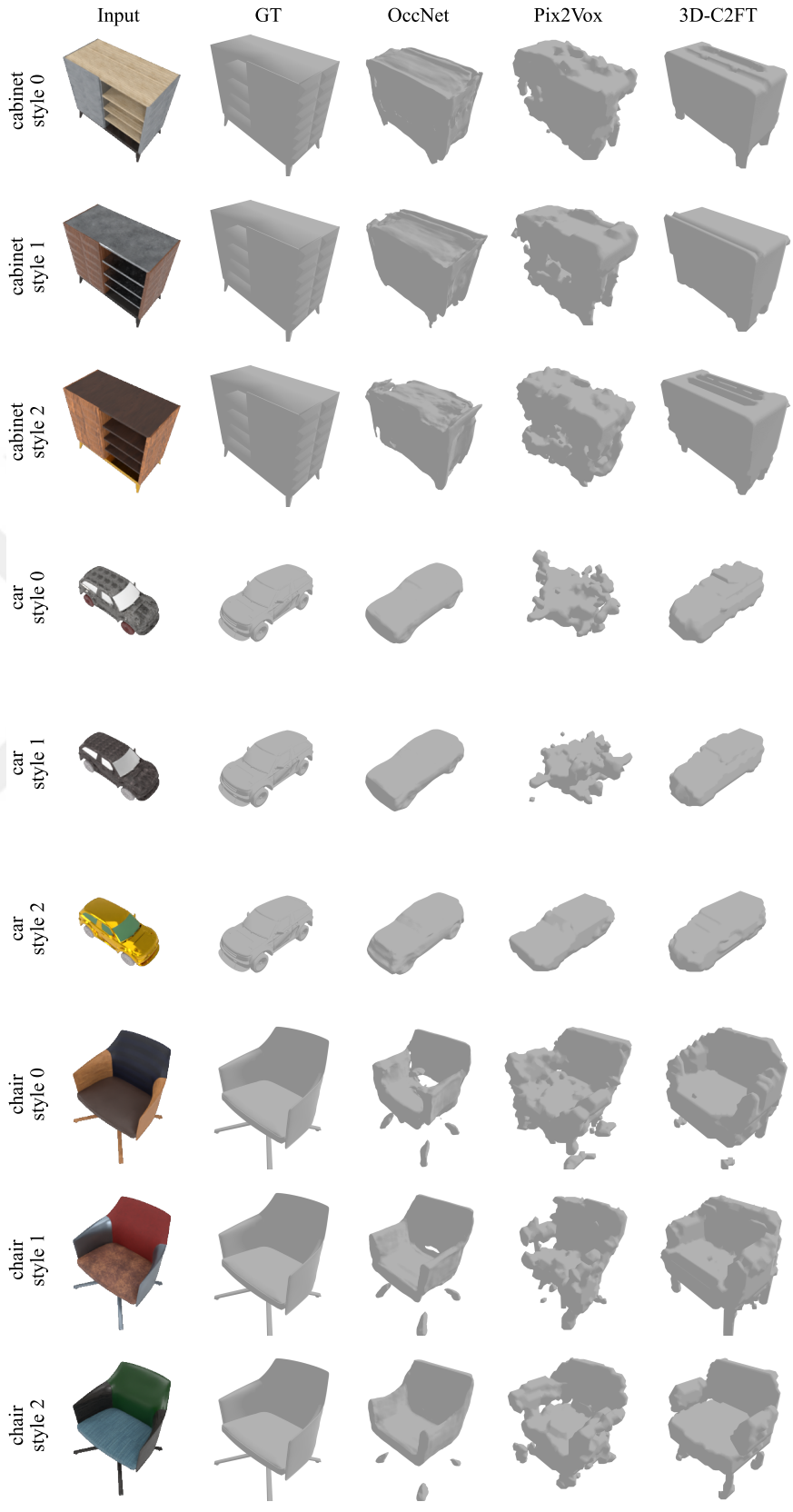


Figure 4.20. Reconstruction examples of images with 3 different object textures.

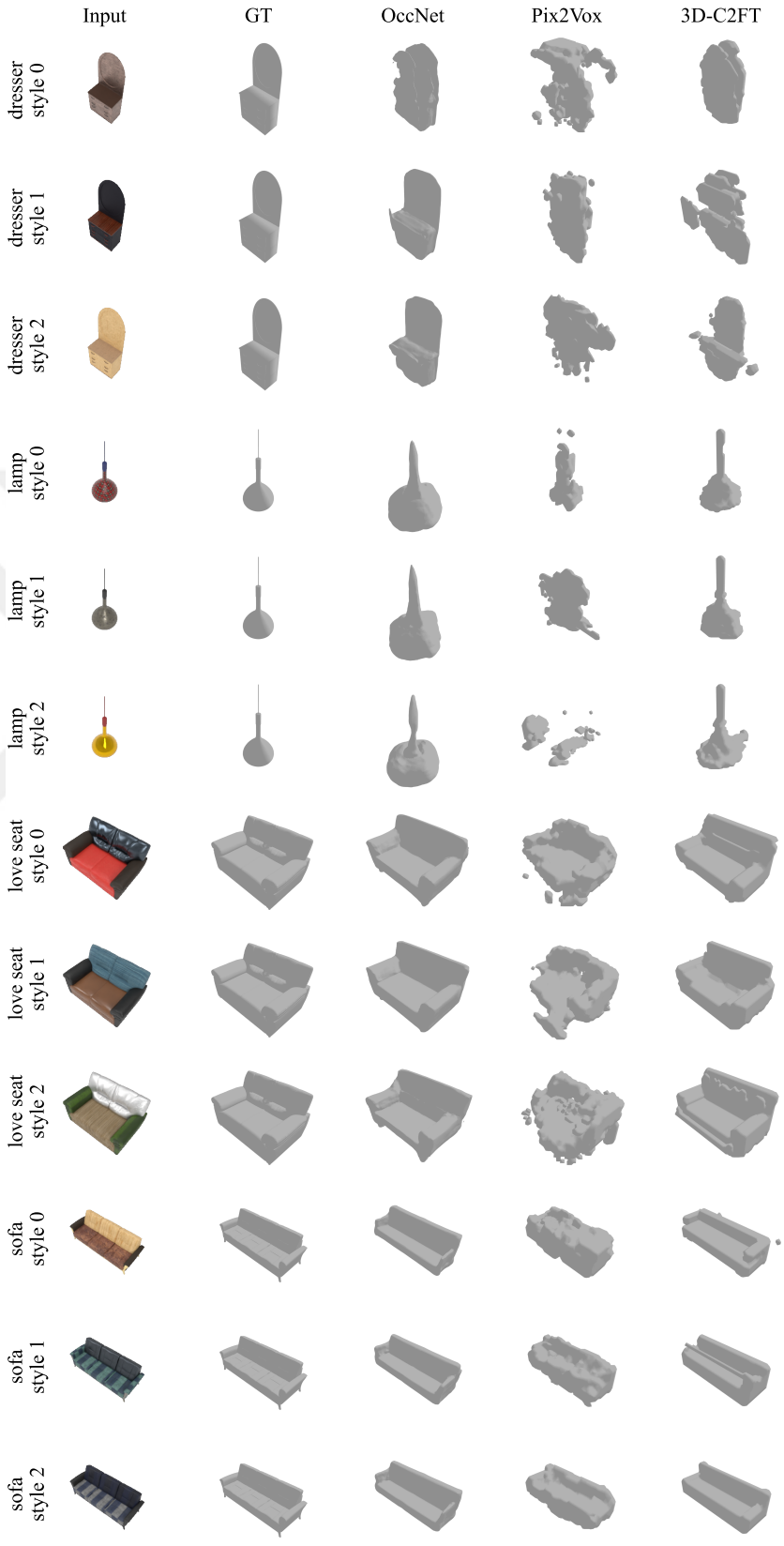


Figure 4.21. Reconstruction examples of images with 3 different object textures.

4.5.3. Part-Based Evaluation

In this section, we evaluate the reconstruction performance at the part level. The categorical performance evaluations suggest that the baseline models struggle most with thin parts. This experiment quantitatively substantiates that observation. Table 4.9 presents the F-Score at the part level for each category. These results can be interpreted in two ways: firstly, to identify where models fail, serving as an ablation study, and secondly, to determine which model is better at reconstructing difficult parts. For the latter, we do not observe any significant improvement between models. All models struggle to reconstruct finer details, and the scores are quite low. However, it's notable that even these low scores are primarily attributable to the larger parts of the objects. Figure 4.22 shows examples of the observations for each category. In the figure, we annotate the points with the labels of the closest points in the target mesh, similar to part annotation transfer explained in Section 4.2.4. It should be noted that this does not equate to the $\text{F-Score}_{\text{part}}$ in Equation 4.7 but rather shows the potential candidate points for part-based precision P_{part} in Equation 4.8.

Our qualitative and quantitative observations suggest that Pix2Vox is not ideally suited for a confident interpretation within this dataset. However, we have conducted part-level observations for OccNet and 3D-C2FT. We note that larger parts, such as armrests in sofa and loveseat categories, perform significantly better than the base part. The base part refers to skeleton-like structures in these categories, typically denoting leg or metallic, wooden structures that form the object's skeleton. Similar patterns are observed in boat, car, and airplane categories, where finer details are reconstructed with significantly less accuracy. Notably, OccNet's reconstruction of the airplane body is much better than its finer details, even though these details are not classified as thin parts. Categories rich in structure, such as benches, chairs, and lamps, present inherent challenges due to their numerous thin parts, resulting in too low scores to discern a clearer pattern. In the case of cabinets and dressers, we observe that the most prominent contributing parts are panels, which are the larger components in these categories.

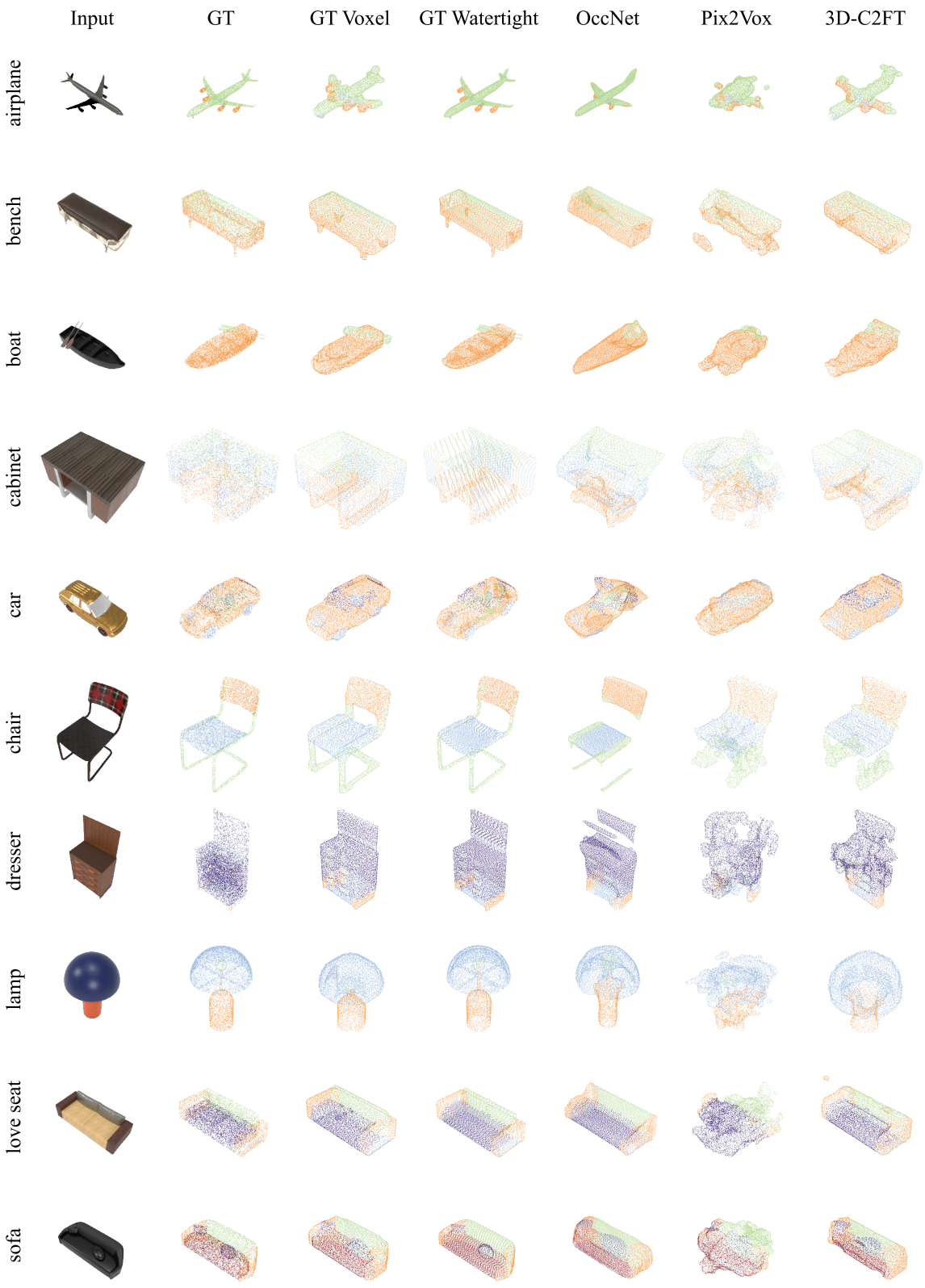


Figure 4.22. Example reconstructions at part level. The closest points between shapes in each row are painted in the same color.

Table 4.9. Part F-Score@0.01 results. Bold values show the best part-level result for the model. Underlined values show the best part-level result across all models.

Category	Part	OccNet	Pix2Vox	3D-C2FT
airplane	body	**0.527**	**0.128**	**0.313**
	engine	0.265	0.147	0.209
	wheel	0.196	0.106	0.255
bench	backrest	0.197	0.136	0.159
	base	0.203	0.126	0.181
	seat	**0.222**	**0.163**	**0.202**
	pillow	0.107	0.091	0.139
boat	body	**0.224**	**0.179**	**0.234**
	paddle	0.209	0.139	0.196
	sail	0.126	0.107	0.130
	window	0.161	0.190	0.156
cabinet	base	0.123	0.090	0.105
	horizontal panel	0.201	0.074	0.106
	vertical panel	**0.216**	**0.141**	**0.252**
car	body	0.275	**0.224**	0.297
	seat	0.091	0.050	**0.335**
	wheel	0.174	0.069	0.150
	window	**0.379**	0.180	0.277
chair	base	0.103	0.066	0.093
	backrest	**0.141**	**0.100**	**0.132**
	pillow	0.050	0.055	0.046
	seat	0.136	0.097	0.127
lamp	base	**0.118**	0.091	**0.186**
	bulb	0.105	**0.093**	0.142
	cover	0.103	0.078	0.135

Table 4.9. Part F-Score@0.01 results. Bold values show the best part-level result for the model. Underlined values show the best part-level result across all models. (cont.)

Category	Part	OccNet	Pix2Vox	3D-C2FT
dresser	base	0.096	0.053	0.062
	handle	0.129	0.082	0.103
	horizontal panel	0.131	**0.105**	0.134
	vertical panel	**0.162**	0.095	**0.135**
love seat	armrest	**0.196**	**0.164**	**0.307**
	base	0.105	0.072	0.098
	backrest	0.159	0.138	0.161
	pillow	0.117	0.103	0.113
	seat	0.169	0.150	0.185
sofa	armrest	**0.251**	**0.209**	**0.368**
	backrest	0.195	0.165	0.176
	base	0.125	0.113	0.145
	pillow	0.131	0.121	0.158
	seat	0.234	0.183	0.208

4.6. Benchmark Results

In this section, we present an overall assessment of the selected models based on our benchmark results. Unexpectedly, Pix2Vox, which demonstrated superior performance in the ShapeNet test, appears to be the least robust when tested with novel cases and is sensitive to minor changes in the known dataset. Additionally, we observe that both OccNet and 3D-C2FT perform closely and reasonably well when compared to Pix2Vox. However, both models have difficulties in adapting to novel shape reconstruction. We also observe a clear comparison result in two experiments between these models. OccNet outperforms 3D-C2FT in reconstructing shapes from lower-resolution input images, while 3D-C2FT demonstrates significant robustness in handling background changes. These findings are summarized in Table 4.10.

Table 4.10. Summary of benchmark results. ⋆: Outperforms the other models, ✓: Relatively good performance, -: Fails.

Dataset	Test	OccNet	Pix2Vox	3D-C2FT
ShapeNet	Original Test	✓	⋆	✓
ShapeNet	Black Background	-	-	⋆
ShapeNet	Scaling by 0.90	✓	✓	✓
ShapeNet	Scaling by 0.75	✓	✓	✓
ShapeNet	Scaling by 0.50	⋆	-	-
3DCoMPaT++	Varying Viewpoint	✓	-	✓
3DCoMPaT++	Varying Texture	✓	-	✓
3DCoMPaT++	Finer Details	-	-	-

5. CONCLUSION AND FUTURE WORK

In this thesis, we delved into the less-explored challenges of 3D reconstruction by employing realistic scenarios and innovative experiments. Our findings indicated a notable decrease in the performance of the selected models with variations on the known dataset and a novel dataset. Contrary to expectations set by common test scenarios, Pix2Vox did not excel under our novel benchmarks. We observed that some models struggled with changes in background color, while a more robust one remained significantly less affected by such changes despite that it performs the worst under standard benchmarks. Our study also highlighted the effect of input object size in 2D renderings on reconstruction performance. We observed that the selected models performed sufficiently robust to this variation, considering the information loss caused by scaling.

Using 3DCoMPaT++, we revealed a significant drop in the models' performance when tested with an unknown dataset, suggesting a high sensitivity of the models to the characteristics of their training data. We expanded our experiments to include tests against texture and viewpoint changes. While the overall low scores make it challenging to assess model performance conclusively, we noted an apparent invariance to these factors.

Additionally, we provided quantitative evidence for a commonly discussed issue, previously highlighted only through qualitative observations: the difficulty of reconstructing fine details and thin parts. We adapted part-level information of 3DCoMPaT++ for 3D reconstruction. Our approach successfully transferred annotations in part point clouds to surface-level points. The correlation between our qualitative findings and the newly proposed Part F-Score indicates that our methodology enables accurate part-level measurements. This improved benchmark serves not only as an advanced test environment but also as a potential dataset for fine-tuning models to better capture fine details and enhance their robustness.

Adopting the 3DCoMPaT++ dataset for evaluation enables new directions for future improvements. This dataset's detailed nature and close resemblance to ShapeNet, particularly after implementing our suggested processes, makes it ideal for testing models at part level. These insights are crucial for refining reconstruction evaluation methods. However, a clear understanding of part-level achievements requires models with better performance. Currently, low overall scores make it challenging to confidently assess such achievements. To address this, we propose several strategies:

- The similarity of 3DCoMPaT++ to ShapeNet can be increased by rendering from textureless meshes using the same view angles. This adjustment will make the dataset ideal for part-level assessment by eliminating slight viewpoint and texture changes.
- Although PartNet directly annotates ShapeNet, given the current limitations in model performance, evaluating fine details with PartNet is not the most effective approach. An alternative is to group fine-grained labels into broader categories to create pseudo-coarse-grained labels.
- Although the number of unique shapes may be insufficient for comprehensive training, our adapted version of 3DCoMPaT++ is well-suited for fine-tuning existing models. Our annotations closely resemble ShapeNet's, and with an improved loss scheme, effective part-based learning is promising. While some studies focus on learning parts individually, we advocate for an approach directly applicable to holistic reconstructions with our version of 3DCoMPaT++.

A common issue we observe in reconstructions is the presence of unconnected parts. For instance, rather than reconstructing a chair's leg as a whole stick, it often appears as discrete, multi-part segments. This type of discrete structure is inherently unlikely and necessitates strong structural clues. We suggest that this issue requires special attention, considering its frequent occurrence in reconstructions and rarity in actual GT shapes. We claim that developing a new method with special attention to this structural integrity problem could significantly boost performance, similar to the use of refiners in some research to correct voxel artifacts.

Lastly, the topic of texture reconstruction is closely related. Given that 3DCoM-PaT++ is rich in textures and includes part annotations with material information, our methodology offers a promising start for evaluating 3D texture reconstruction in greater detail.

Our research offers insights into future directions based on our observations. It guides the creation of robust solutions in 3D reconstruction by quantitatively unveiling limitations. We also suggested ways to advance this benchmark even further.

REFERENCES

1. Chang, A. X., T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi and F. Yu, "ShapeNet: An Information-Rich 3D Model Repository", ArXiv 1512.03012 [cs.GR], 2015.

2. Tiong, L. C. O., D. Sigmund and A. B. J. Teoh, "3D-C2FT: Coarse-to-fine Transformer for Multi-view 3D Reconstruction", *The Proceedings of the Asian Conference on Computer Vision*, Macao, China, pp. 1438–1454, 2022.

3. Xie, H., H. Yao, X. Sun, S. Zhou and S. Zhang, "Pix2Vox: Context-Aware 3D Reconstruction From Single and Multi-View Images", *The Proceedings of the International Conference on Computer Vision*, Seoul, Korea (South), pp. 2690–2698, 2019.

4. Mescheder, L., M. Oechsle, M. Niemeyer, S. Nowozin and A. Geiger, "Occupancy Networks: Learning 3D Reconstruction in Function Space", *The Proceedings of the Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, pp. 4455–4465, 2019.

5. Yuchen Li, H. S. A. A. A. P. S. P. P. W. M. E., Ujjwal Upadhyay, "3DCoMPaT: Composition of Materials on Parts of 3D Things", *The Proceedings of the European Conference on Computer Vision*, Tel Aviv, Israel, pp. 110–127, 2022.

6. Slim, H., X. Li, M. A. Yuchen Li, M. Ayman, U. U. A. Abdelreheem, S. P. Arpit Prajapati, P. Wonka and M. Elhoseiny, "3DCoMPaT++: An improved Large-scale 3D Vision Dataset for Compositional Recognition", ArXiv 2310.18511 [cs.CV], 2023.

7. Han, X.-F., H. Laga and Bennamoun, "Image-Based 3D Object Reconstruction: State-of-the-Art and Trends in the Deep Learning Era", *Transactions on Pattern*

Analysis and Machine Intelligence, Vol. 43, No. 5, pp. 1578–1604, 2019.

8. Fahim, G., K. Amin and S. Zarif, "Single-View 3D reconstruction: A Survey of deep learning methods", *Computers & Graphics*, Vol. 94, No. 1, pp. 164–190, 2021.

9. Samavati, T. and M. Soryani, "Deep Learning-Based 3D Reconstruction: A Survey", *Artificial Intelligence Review*, Vol. 56, No. 9, pp. 9175–9219, 2023.

10. Sarmah, M., A. Neelima and H. R. Singh, "Survey of Methods and Principles in Three-Dimensional Reconstruction from Two-Dimensional Medical Images", *Visual Computing for Industry, Biomedicine, and Art*, Vol. 6, No. 1, p. 15, 2023.

11. Yurtsever, E., J. Lambert, A. Carballo and K. Takeda, "A Survey of Autonomous Driving: Common Practices and Emerging Technologies", *IEEE Access*, Vol. 8, No. 1, pp. 58443–58469, 2020.

12. Pang, H. E. and F. Biljecki, "3D Building Reconstruction from Single Street View Images Using Deep Learning", *International Journal of Applied Earth Observation and Geoinformation*, Vol. 112, No. 1, p. 102859, 2022.

13. Guo, C., T. Jiang, X. Chen, J. Song and O. Hilliges, "Vid2Avatar: 3D Avatar Reconstruction from Videos in the Wild via Self-supervised Scene Decomposition", *The Proceedings of the Conference on Computer Vision and Pattern Recognition*, Vancouver, BC, Canada, pp. 12858–12868, 2023.

14. Hartley, R. and A. Zisserman, *Multiple View Geometry in Computer Vision*, Second Edition, Cambridge University Press, Cambridge, 2004.

15. Furukawa, Y., B. Curless, S. M. Seitz and R. Szeliski, "Manhattan-World Stereo", *The Proceedings of the Conference on Computer Vision and Pattern Recognition*, Miami, FL, USA, pp. 1422–1429, 2009.

16. Sinha, S. N., D. Steedly and R. Szeliski, "Piecewise Planar Stereo for Image-based

Rendering", *The Proceedings of the International Conference on Computer Vision*, Kyoto, Japan, pp. 1881–1888, 2009.

17. Tatarchenko, M., S. R. Richter, R. Ranftl, Z. Li, V. Koltun and T. Brox, "What Do Single-View 3D Reconstruction Networks Learn?", *The Proceedings of the Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, pp. 3400–3409, 2019.

18. Engelmann, F., K. Rematas, B. Leibe and V. Ferrari, "From Points to Multi-Object 3D Reconstruction", *The Proceedings of the Conference on Computer Vision and Pattern Recognition*, Nashville, TN, USA, pp. 4586–4595, 2021.

19. Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser and I. Polosukhin, "Attention is All you Need", *The Proceedings of the International Conference on Neural Information Processing Systems*, Long Beach, CA, USA, pp. 6000–6010, 2017.

20. Sun, X., J. Wu, X. Zhang, Z. Zhang, C. Zhang, T. Xue, J. B. Tenenbaum and W. T. Freeman, "Pix3D: Dataset and Methods for Single-Image 3D Shape Modeling", *The Proceedings of the Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 2974–2983, 2018.

21. Choy, C. B., D. Xu, J. Gwak, K. Chen and S. Savarese, "3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction", *The Proceedings of the European Conference on Computer Vision*, Amsterdam, The Netherlands, pp. 628–644, 2016.

22. Sundermeyer, M., R. Schlüter and H. Ney, "LSTM Neural Networks for Language Modeling", *The Proceedings of Interspeech*, Portland, OR, USA, pp. 194–197, 2012.

23. Cho, K., B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio, "Learning Phrase Representations using RNN Encoder–Decoder for

Statistical Machine Translation", *The Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar, pp. 1724–1734, 2014.

24. Yang, B., S. Wang, A. Markham and N. Trigoni, "Robust Attentional Aggregation of Deep Feature Sets for Multi-view 3D Reconstruction", *International Journal of Computer Vision*, Vol. 128, No. 1, pp. 53–73, 2018.

25. Xie, H., H. Yao, S. Zhang, S. Zhou and W. Sun, "Pix2Vox++: Multi-scale Context-aware 3D Object Reconstruction from Single and Multiple Images", *International Journal of Computer Vision*, Vol. 128, No. 12, pp. 2919–2935, 2020.

26. Yang, S., M. Xu, H. Xie, S. Perry and J. Xia, "Single-View 3D Object Reconstruction from Shape Priors in Memory", *The Proceedings of the Conference on Computer Vision and Pattern Recognition*, Nashville, TN, USA, pp. 3151–3160, 2021.

27. Xing, Z., Y. Chen, Z. Ling, X. Zhou and Y. Xiang, "Few-Shot Single-View 3D Reconstruction with Memory Prior Contrastive Network", *The Proceedings of the European Conference on Computer Vision*, Tel Aviv, Israel, pp. 55–70, 2022.

28. Wang, D., X. Cui, X. Chen, Z. Zou, T. Shi, S. E. Salcudean, Z. J. Wang and R. K. Ward, "Multi-View 3D Reconstruction with Transformers", *The Proceedings of the International Conference on Computer Vision*, Montreal, QC, Canada, pp. 5702–5711, 2021.

29. Yagubbayli, F., A. Tonioni and F. Tombari, "LegoFormer: Transformers for Block-by-Block Multi-view 3D Reconstruction", ArXiv 2106.12102 [cs.CV], 2022.

30. Richter, S. R. and S. Roth, "Matryoshka Networks: Predicting 3D Geometry via Nested Shape Layers", *The Proceedings of the Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 1936–1944, 2018.

31. Wang, N., Y. Zhang, Z. Li, Y. Fu, W. Liu and Y.-G. Jiang, "Pixel2Mesh: Generat-

ing 3D Mesh Models from Single RGB Images", *The Proceedings of the European Conference on Computer Vision*, Munich, Germany, pp. 52–67, 2018.

32. Kipf, T. N. and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks", *The Proceedings of the International Conference on Learning Representations*, Toulon, France, pp. 1–14, 2017.

33. Wen, C., Y. Zhang, Z. Li and Y. Fu, "Pixel2Mesh++: Multi-View 3D Mesh Generation via Deformation", *The Proceedings of the International Conference on Computer Vision*, Seoul, Korea (South), pp. 1042–1051, 2019.

34. Gkioxari, G., J. Malik and J. Johnson, "Mesh R-CNN", *The Proceedings of the International Conference on Computer Vision*, Seoul, Korea (South), pp. 9784–9794, 2019.

35. Groueix, T., M. Fisher, V. G. Kim, B. C. Russell and M. Aubry, "A Papier-Mache Approach to Learning 3D Surface Generation", *The Proceedings of the Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 216–224, 2018.

36. Guillard, B., E. Remelli, P. Yvernay and P. Fua, "Sketch2Mesh: Reconstructing and Editing 3D Shapes from Sketches", *The Proceedings of the International Conference on Computer Vision*, Montreal, QC, Canada, pp. 13003–13012, 2021.

37. Monnier, T., M. Fisher, A. A. Efros and M. Aubry, "Share with Thy Neighbors: Single-View Reconstruction by Cross-Instance Consistency", *The Proceedings of the European Conference on Computer Vision*, Tel Aviv, Israel, pp. 285–303, 2022.

38. Insafutdinov, E. and A. Dosovitskiy, "Unsupervised Learning of Shape and Pose with Differentiable Point Clouds", *The Proceedings of the International Conference on Neural Information Processing Systems*, Montreal, QC, Canada, pp. 2807–2817, 2018.

39. Han, Z., C. Chen, Y.-S. Liu and M. Zwicker, "DRWR: A Differentiable Renderer without Rendering for Unsupervised 3D Structure Learning from Silhouette Images", ArXiv 2007.06127 [cs.CV], 2020.

40. Chen, C., Z. Han, Y. Liu and M. Zwicker, "Unsupervised Learning of Fine Structure Generation for 3D Point Clouds by 2D Projection Matching", *The Proceedings of the International Conference on Computer Vision*, Montreal, QC, Canada, pp. 12446–12457, 2021.

41. Liu, Z., H. Tang, Y. Lin and S. Han, "Point-Voxel CNN for Efficient 3D Deep Learning", *The Proceedings of the International Conference on Neural Information Processing Systems*, Vancouver, BC, Canada, pp. 965–975, 2019.

42. Zhou, L., Y. Du and J. Wu, "3D Shape Generation and Completion through Point-Voxel Diffusion", *The Proceedings of the International Conference on Computer Vision*, Montreal, QC, Canada, pp. 5806–5815, 2021.

43. Gao, C., Q. Yu, L. Sheng, Y.-Z. Song and D. Xu, "SketchSampler: Sketch-Based 3D Reconstruction via View-Dependent Depth Sampling", *The Proceedings of the European Conference on Computer Vision*, Tel Aviv, Israel, pp. 464–479, 2022.

44. Zhang, Y., K. Huo, Z. Liu, Y. Zang, Y. Liu, X. Li, Q. Zhang and C. Wang, "PGNet: A Part-Based Generative Network for 3D Object Reconstruction", *Knowledge-Based Systems*, Vol. 194, No. 1, p. 105574, 2020.

45. Yu, Q., C. Yang and H. Wei, "Part-Wise AtlasNet for 3D Point Cloud Reconstruction from a Single Image", *Knowledge-Based Systems*, Vol. 242, No. 1, p. 108395, 2022.

46. Peng, S., M. Niemeyer, L. Mescheder, M. Pollefeys and A. Geiger, "Convolutional Occupancy Networks", *The Proceedings of the European Conference on Computer Vision*, Glasgow, UK, pp. 523–540, 2020.

47. Venkatesh, R., T. Karmali, S. Sharma, A. Ghosh, R. Babu, L. A. Jeni and M. Singh, "Deep Implicit Surface Point Prediction Networks", *The Proceedings of the International Conference on Computer Vision*, Montreal, QC, Canada, pp. 12633–12642, 2021.

48. Mildenhall, B., P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi and R. Ng, "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis", *Communications of the ACM*, Vol. 65, No. 1, pp. 99–106, 2021.

49. Trevithick, A. and B. Yang, "GRF: Learning a General Radiance Field for 3D Representation and Rendering", *The Proceedings of the International Conference on Computer Vision*, Montreal, QC, Canada, pp. 15162–15172, 2020.

50. Oechsle, M., S. Peng and A. Geiger, "UNISURF: Unifying Neural Implicit Surfaces and Radiance Fields for Multi-View Reconstruction", *The Proceedings of the International Conference on Computer Vision*, Montreal, QC, Canada, pp. 5569–5579, 2021.

51. Yang, M., Y. Wen, W. Chen, Y. Chen and K. Jia, "Deep Optimized Priors for 3D Shape Modeling and Reconstruction", *The Proceedings of the Conference on Computer Vision and Pattern Recognition*, Nashville, TN, USA, pp. 3268–3277, 2021.

52. Ibing, M., I. Lim and L. P. Kobbelt, "3D Shape Generation with Grid-based Implicit Functions", *The Proceedings of the Conference on Computer Vision and Pattern Recognition*, Nashville, TN, USA, pp. 13554–13563, 2021.

53. Liu, F. and X. Liu, "2D GANs Meet Unsupervised Single-view 3D Reconstruction", *The Proceedings of the European Conference on Computer Vision*, Tel Aviv, Israel, pp. 497–514, 2022.

54. Karras, T., S. Laine and T. Aila, "A Style-Based Generator Architecture for Gen-

erative Adversarial Networks", *Transactions on Pattern Analysis & Machine Intelligence*, Vol. 43, No. 12, pp. 4217–4228, 2021.

55. Huang, Z., V. Jampani, A. Thai, Y. Li, S. Stojanov and J. M. Rehg, "ShapeClipper: Scalable 3D Shape Learning from Single-View Images via Geometric and CLIP-based Consistency", *The Proceedings of the Conference on Computer Vision and Pattern Recognition*, Vancouver, BC, Canada, pp. 12912–12922, 2023.

56. Radford, A., J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger and I. Sutskever, "Learning Transferable Visual Models From Natural Language Supervision", ArXiv 2103.00020 [cs.CV], 2021.

57. Zheng, Z., T. Yu, Q. Dai and Y. Liu, "Deep Implicit Templates for 3D Shape Representation", *The Proceedings of the Conference on Computer Vision and Pattern Recognition*, Nashville, TN, USA, pp. 1429–1439, 2021.

58. Deng, Y., J. Yang and X. Tong, "Deformed Implicit Field: Modeling 3D Shapes with Learned Dense Correspondence", *The Proceedings of the Conference on Computer Vision and Pattern Recognition*, Nashville, TN, USA, pp. 10281–10291, 2021.

59. Mo, K., S. Zhu, A. X. Chang, L. Yi, S. Tripathi, L. J. Guibas and H. Su, "PartNet: A Large-Scale Benchmark for Fine-Grained and Hierarchical Part-Level 3D Object Understanding", *The Proceedings of the Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, pp. 909–918, 2019.

60. Wu, Z., S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang and J. Xiao, "3D ShapeNets: A Deep Representation For Volumetric Shapes", *The Proceedings of the Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA, pp. 1912–1920, 2015.

61. Lim, J. J., H. Pirsiavash and A. Torralba, "Parsing IKEA Objects: Fine Pose

Estimation", *The Proceedings of the International Conference on Computer Vision*, Sydney, NSW, Australia, pp. 2992–2999, 2013.

62. Xiang, Y., R. Mottaghi and S. Savarese, "Beyond PASCAL: A Benchmark for 3D Object Detection in the Wild", *The Proceedings of the Winter Conference on Applications of Computer Vision*, Steamboat Springs, CO, USA, pp. 75–82, 2014.

63. Xiang, Y., W. Kim, W. Chen, J. Ji, C. Choy, H. Su, R. Mottaghi, L. Guibas and S. Savarese, "ObjectNet3D: A Large Scale Database for 3D Object Recognition", *The Proceedings of the European Conference Computer Vision*, Amsterdam, The Netherlands, pp. 160–176, 2016.

64. Song, S., F. Yu, A. Zeng, A. X. Chang, M. Savva and T. Funkhouser, "Semantic Scene Completion from a Single Depth Image", *The Proceedings of the Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, pp. 190–198, 2017.

65. Wiles, O. and A. Zisserman, "SilNet : Single- and Multi-View Reconstruction by Learning from Silhouettes", ArXiv 1711.07888 [cs.CV], 2017.

66. Reizenstein, J., R. Shapovalov, P. Henzler, L. Sbordone, P. Labatut and D. Novotny, "Common Objects in 3D: Large-Scale Learning and Evaluation of Real-life 3D Category Reconstruction", *The Proceedings of the International Conference on Computer Vision*, Montreal, QC, Canada, pp. 10881–10891, 2021.

67. Shrestha, R., S. Hu, M. Gou, Z. Liu and P. Tan, "A Real World Dataset for Multi-View 3D Reconstruction", *The Proceedings of the European Conference on Computer Vision*, Tel Aviv, Israel, pp. 56–73, 2022.

68. Zhong, Y., Y. Qi, Y. Gryaditskaya, H. Zhang and Y.-Z. Song, "Towards Practical Sketch-Based 3D Shape Generation: The Role of Professional Sketches", *Transactions on Circuits and Systems for Video Technology*, Vol. 31, No. 9, pp. 3518–3528,

2021.

69. Welinder, P., S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie and P. Perona, *Caltech-UCSD Birds 200*, Tech. rep., California Institute of Technology, 2010.

70. Yu, F., A. Seff, Y. Zhang, S. Song, T. Funkhouser and J. Xiao, "LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop", ArXiv 1506.03365 [cs.CV], 2016.

71. Yang, L., P. Luo, C. C. Loy and X. Tang, "A Large-Scale Car Dataset for Fine-Grained Categorization and Verification", *The Proceedings of the Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA, pp. 3973–3981, 2015.

72. Kalogerakis, E., M. Averkiou, S. Maji and S. Chaudhuri, "3D Shape Segmentation with Projective Convolutional Networks", *The Proceedings of the Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, pp. 6630–6639, 2017.

73. Yi, L., V. G. Kim, D. Ceylan, I.-C. Shen, M. Yan, H. Su, C. Lu, Q.-X. Huang, A. Sheffer and L. J. Guibas, "A Scalable Active Framework for Region Annotation in 3D Shape Collections", *ACM Transactions on Graphics*, Vol. 35, No. 6, pp. 1–12, 2016.

74. Kuznetsova, A., H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, T. Duerig and V. Ferrari, "The Open Images Dataset V4: Unified Image Classification, Object Detection, and Visual Relationship Detection at Scale", *International Journal of Computer Vision*, Vol. 128, No. 7, pp. 1956–1981, 2020.

75. He, K., X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition", *The Proceedings of the Conference on Computer Vision and Pattern Recog-*

nition, Las Vegas, NV, USA, pp. 770–778, 2016.

76. Lorensen, W. E. and H. E. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm", *The ACM Special Interest Group on Computer Graphics*, Vol. 21, No. 4, pp. 163–169, 1987.

77. Simonyan, K. and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", *The Proceedings of the International Conference on Learning Representations*, San Diego, CA, USA, pp. 1–14, 2015.

78. Huang, G., Z. Liu and K. Q. Weinberger, "Densely Connected Convolutional Networks", *The Proceedings of the Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, pp. 2261–2269, 2017.

79. Nooruddin, F. S. and G. Turk, "Simplification and Repair of Polygonal Models Using Volumetric Techniques", *Transactions on Visualization and Computer Graphics*, Vol. 9, No. 2, pp. 191–205, 2003.

80. Min, P., "binvox", 2004, `http://www.patrickmin.com/binvox`, accessed on September 30, 2023.

81. Stutz, D. and A. Geiger, "Learning 3D Shape Completion from Laser Scan Data with Weak Supervision", *The Proceedings of the Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 1955–1964, 2018.

82. Muja, M. and D. G. Lowe, "Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration", *The Proceedings of the International Conference on Computer Vision Theory and Applications*, Lisboa, Portugal, pp. 331–340, 2009.

83. Zhou, Q.-Y., J. Park and V. Koltun, "Open3D: A Modern Library for 3D Data Processing", ArXiv 1801.09847 [cs.CV], 2018.