Enhancing Shelf-Supervised Mesh Prediction in the Wild

Florin-Vlad Sabău
TUM
florin.sabau@tum.de

Janis Köhler
LMU
janis.koehler@campus.lmu.de

Răzvan-Andrei Lazăr
TUM
razvan.lazar@tum.de

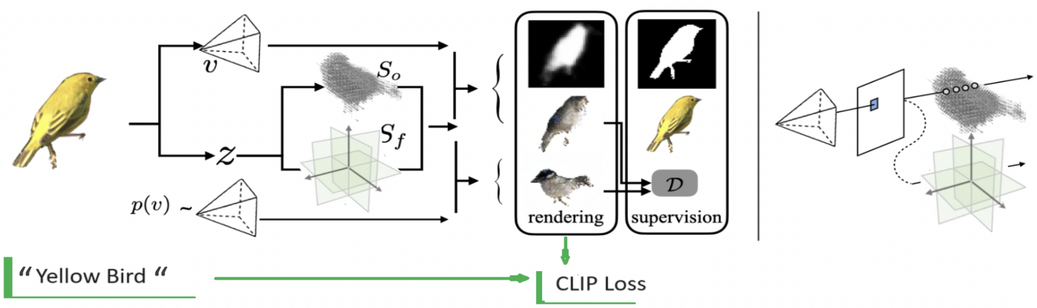


Figure 1. An overview of the architecture. The modifications to the original are highlighted in green.

Abstract

This study explores enhancements to shelf-supervised mesh prediction for 3D reconstruction from 2D images, addressing the limitations of requiring explicit 3D supervision. We introduce three main improvements: natural language supervision via pre-trained CLIP encoders, SE(3)-equivariant convolutions for geometric invariance, and a more efficient tri-plane feature representation. Experiments conducted on the 3D-FUTURE dataset show varied impacts of these modifications on model performance, evaluated using Chamfer distance. Despite implementation challenges and dataset compatibility issues, preliminary results highlight the potential of integrating language understanding and geometric considerations in mesh prediction models. Further research is needed to fully assess the effectiveness of these approaches.

1. Introduction

The field of 3D mesh reconstruction from 2D images has witnessed substantial advancements in recent years, fueled by the surge in computational capabilities and the availability of vast datasets. However, most works rely on explicit 3D [9] or multi-view supervision [15], which are expensive to obtain. A method that doesn't rely on these is presented by Ye et al. [16] While achieving remarkable results considering the difficulty of the task, we present some ideas for improvement.

First, since getting text descriptions for the images is usually not challenging, we propose to also add natural language supervision using pre-trained CLIP encoders [13]. Secondly, since the complicated loss landscape poses a challenge for optimization, we propose to add more inductive biases by changing the normal 3D convolutions to ones that are also equivariant to rotation and reflection (SE(3)-equivariant) [3]. And thirdly, we change the 3D voxel grid feature representation to a more efficient tri-plane one [5].

2. Method

In this section we present the theoretical and practical details of our work.

2.1. Original

The original paper proposes a learning-based approach to infer the 3D shape and pose of objects from single images. It introduces a method called 'shelf-supervised' learning, which can train from unstructured image collections using segmentation outputs from off-the-shelf recognition systems.

The approach first infers a volumetric representation and camera pose in a canonical frame, ensuring geometric consistency with appearance and masks, as well as synthesizing novel views that are indistinguishable from image collections.

Then, the coarse volumetric prediction is converted into a mesh-based representation, refined in the predicted camera frame. This approach enables both shape-pose factorization from image collections and per-instance reconstruction in finer details.

Architecture

The model is composed of the following components:

Encoder: Image → View & Latent Representation
Given an image $I \in [0,1]^{3xH_I xW_I}$, the encoder predicts a view v and a low-dimensional latent variable z that contains no information about the pose. It is implemented as a ResNet [8].

$$(v, z) = \phi_E(I) \tag{1}$$

Decoder: Latent → 3D Features & Occupancy Grid
The latent vector is then transformed to a canonical-frame voxel grid of features $S_f \in \mathbb{R}^{F \times D_f \times H_f \times W_f}$, from which the explicit occupancy grid is also decoded $S_o \in [0,1]^{D_o \times H_o \times W_o}$. It uses the StyleGAN2 [10, 11] decoder architecture, with z as the 'style'.

$$(S_o, S_f) = \phi_D(z) \tag{2}$$

Neural Renderer: View & 3D Features → Image
Using this 3D representation, we can then render an image and a mask from an arbitrary view. For each pixel we send a ray and sample points from the occupancy grid. These points are used to sample features from the voxel grid, which are then aggregated and decoded into a color.

$$(\hat{M}_v, \hat{I}_v) = \pi(S_o, S_f, v) \tag{3}$$

Discriminator: Image → Real or Fake
In the style of GANs [7, 14], the discriminator is a classifier that tries to classify images as being real or generated.

Mesh Refiner: Coarse Mesh → Refined Mesh
From the occupancy grid we can create a mesh. However this occupancy grid is quite coarse. To refine it, as a final step, the mesh is optimized using a differentiable rasterizer so that it better matches the input image from the predicted view.

Training

To train these components they proposed the following supervision signals:

Pixel Consistency Loss: When rendered from the predicted view, the image and mask should match the ground-truth.

$$\mathcal{L}_{rgb} = \|\hat{I}_v - I\|_1 \tag{4}$$

$$\mathcal{L}_{mask} = 1 - \frac{\|\hat{M}_v \otimes M\|_1}{\|\hat{M}_v \oplus M - \hat{M}_v \otimes M\|_1} \tag{5}$$

$$\mathcal{L}_{perc} = \|h(\hat{I}_v) - h(I)\|_2^2 \tag{6}$$

where h represents the feature extracted by a pre-trained AlexNet [12].

View Adversarial Loss: If we use only the previous losses, a degenerate solution could arise in which the shape only makes sense from the predicted view. To avoid this, we sample a random view from a prior distribution $v' \sim p(v)$ in addition to the predicted view.

$$\mathcal{L}_{adv} = \log D(I) + \log(1 - D(\hat{I}_v)) + \log(1 - D(\hat{I}_{v'})) \tag{7}$$

Content Consistency Loss: To regularize the network, we want the encoder and decoder to be self-consistent. If we give the encoder a synthesized image from the decoder we should recover the same v and z.

$$\mathcal{L}_{content} = \|\phi_E(\hat{I}_v) - (v, z)\|_2^2 + \|\phi_E(\hat{I}_{v'}) - (v', z)\|_2^2 \tag{8}$$

Total Loss: Finally we aggregate all losses together.

$$\mathcal{L} = \mathcal{L}_{rgb} + \mathcal{L}_{mask} + \mathcal{L}_{perc} + \mathcal{L}_{adv} + \mathcal{L}_{content} \tag{9}$$

And we optimize the network jointly.

2.2. 3D-FUTURE Dataset

The 3D-FUTURE dataset [6] is a large-scale benchmark focused on household scenarios, featuring extensive 3D and 2D annotations. It comprises 20,240 realistic synthetic images and 9,992 high-quality 3D CAD furniture shapes. Notable features of the dataset include meticulously designed interior layouts by experienced designers, photo-realistic renderings, accurate 2D-3D alignments, and importantly, industrial 3D furniture shapes with informative textures.

While the number of images is impressive, we only used the 3D models, which we rendered from random views. Additionally, each model comes with a collection of tags, such as **Category**, **Style**, **Material** and **Theme**.

2.3. Tri-Plane Representation

While the voxel grid feature representation of the shape confers many advantages, it is limited by the poor memory scaling with resolution, in the order of $O(N^3)$. An alternative representation called the tri-plane recently emerged which keeps most of the advantages but remedies the drawback.

It works by factorizing the 3D grid into three 2D planes (XY, XZ and YZ). To recover the features at a certain point, we take features at the projection of the point on all three planes and aggregate them.

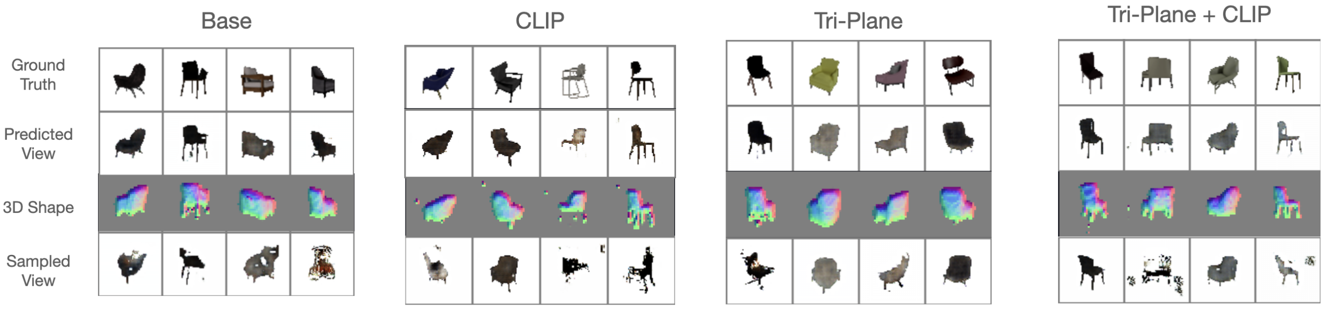


Figure 2. Visualization of the results.

So now instead of a grid $S_f \in \mathbb{R}^{F\times D\times H\times W}$ we have a tri-plane $T_f = (T_{XY}, T_{XZ}, T_{YZ})$ with $T_{XY} \in \mathbb{R}^{F\times H\times W}$, $T_{XZ} \in \mathbb{R}^{F\times D\times W}$ and $T_{YZ} \in \mathbb{R}^{F\times D\times H}$, thereby reducing the memory complexity to $O(N^2)$.

Due to the better scaling, this representation has been used with great success in many recent works, such as EG3D [1], LRM [9], and k-planes [5].

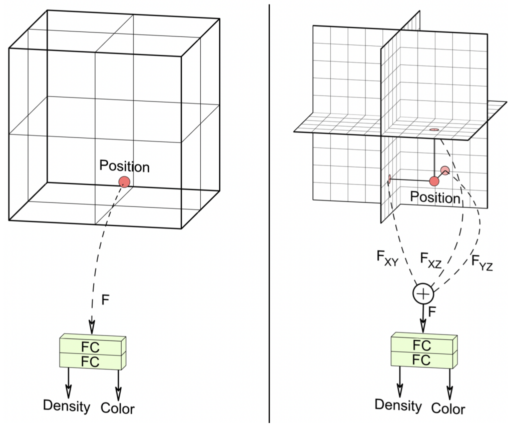


Figure 3. **Left**: The original voxel 3D grid feature representation. **Right**: The new tri-plane feature representation.

2.4. SE(3)-Equivariant Convolutions

As a specific instance of G-CNN layers we integrated SE(3)-Equivariance into the network, replacing every 3DConv layer with an SE(3)-equivariant layer [3]. G-CNNs or Group Convolutional Neural Networks are modules that are equivariant under all isometries of the space $\mathbb{R}^3$. This means, that for a group transformation ρ and a layer represented by a function f:

$$f(\rho(x)) = \rho(f(x)) \tag{10}$$

By embedding SE(3)-equivariance into the convolutional layers, the network is designed to better capture the inherent geometric properties of the 3D world, even when presented with a limited two-dimensional view. We expected to facilitate the extraction of features from the input image, enhancing the model's ability to reconstruct the underlying 3D structure.

Issues

We first thought of the equivariant convolutional layers as an interesting addition to the model. However after diving into the topic more deeply we discovered several issues with its implementation into the framework. For one, in the data we used there are no obvious symmetries. Most chairs for example are photographed upright. In addition the latent representation is learned to be position independent.

Without redesigning large parts of the network structure it is impossible to mathematically render all the outputs of the network equivariant to their input features.

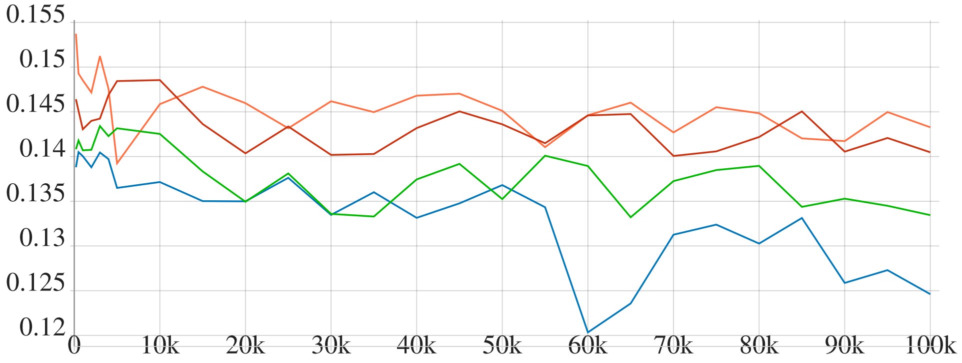
Our first attempt was in line with this line of thought. To the original structure of the network we did not find significant changes, not in loss, nor in output quality. Because of these findings and the amount of changes it would have needed to modify the network accordingly we disregarded further work with this model.

2.5. Natural Language Supervision using CLIP

Recently, large image-text models, such as CLIP (*Contrastive Language–Image Pre-training*) [13], have proven great zero-shot accuracy by training on an immense array of text-annotated images scraped from the internet. These models present great capabilities of capturing visual concepts and relating them to text features and offer good generalisation due to the diversity of training data.

We decided to use CLIP to add another layer of supervision using the model tags provided in the **3D-FUTURE Dataset**, by calculating the similarity between the rendered images of the generated 3D model and the text description.

$$\mathcal{L}_{CLIP} = \cos\theta = \frac{I * T}{||I||\ ||T||} \tag{11}$$



Color	Model	Chamfer
Green	Base	0.133
Orange	Tri-Plane	0.144
Red	CLIP	0.140
Blue	Tri-Plane + CLIP	0.123

Figure 4. Experiment results after 100k iterations. The y-axis is the Chamfer Distance and the x-axis is the iterations of the training loop.

We then used this cosine similarity as an additional loss in the training of our model. Thus, we aimed to better quantify the accuracy of the recreation, especially when observed from a novel viewpoint, where we do not have another reliable method but the adversarial loss.

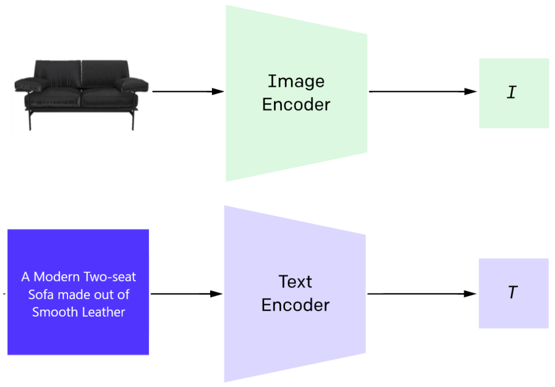


Figure 5. We obtain the image and text encoding from the respective pre-trained CLIP encoders

In our implementation, we used the pre-trained encoders for image and text corresponding to model **CLIP-ViT-B/16** [2], and, after experimenting with a variety of text compositions, we decided to use the following pattern from the 3D-FUTURE tages: ”A {**Style**} {**Type**} made out of {**Material**}”.

3. Results

Since we have the ground truth 3D model, we use the Chamfer distance as a metric for our experiments. We trained the models on all chairs from the 3D-FUTURE dataset with the default hyperparameters from the original.

In figure 4 we present mean validation Chamfer distance throughout the training of each model, along with their test performance. The tri-plane modification makes the model perform worse than the baseline, which was to be expected since we greatly decreased the number of parameters.

However, CLIP makes the tri-plane model perform much better but, weirdly enough, makes the base model perform worse. Looking at the final visualization in figure 2, we can see the models using CLIP perform worse at reconstructing the sampled views, while also adding some artifacts (floating points) to the reconstructed 3D models. We hypothesise these might be due to the fact that CLIP was trained on images that have background, and the new loss in turn pushes the model to also ’paint’ the background.

4. Conclusion

Taking into consideration the conflicting results and the low amount of experiments we ran, we cannot draw a definite conclusion of the effectiveness of our approaches. Using CLIP resulted in an overall improvement of the presented method but lead to side effects worth avoiding. Using tri-planes enhanced computation but deteriorated our result. Overall more experiments would need to be conducted, ideally using a cleaner dataset.

5. Challenges

Employing the network from scratch proved difficult. This was because there were no provided requirements to run the code and because the code was written using an unnecessarily old version of Pytorch, not compatible with the CUDA running on our GPU. The code was not documented clearly, making it necessary to reread and reinterpret every line of the implementation.

We also faced challenges with the datasets we were trying, as in our first attempt we failed leveraging the web-scraped dataset Objaverse [4] due to a lack of instances of single categories. With 3D Future we circumvented this issue but still achieved significantly worse results than with the originally used segmented pictures of chairs, likely due to the cleanliness of the dataset labels.

References

[1] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3d generative adversarial networks, 2022. 3

[2] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning, 2022. 4

[3] Taco Cohen and Max Welling. Group equivariant convolutional networks. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2990–2999, New York, New York, USA, 20–22 Jun 2016. PMLR. 1, 3

[4] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects, 2022. 4

[5] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12479–12488, 2023. 1, 3

[6] Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 3d-future: 3d furniture shape with texture, 2020. 2

[7] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. 2

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 2

[9] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d, 2023. 1, 3

[10] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2019. 2

[11] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan, 2020. 2

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. 2

[13] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 1, 3

[14] Juergen Schmidhuber. Generative adversarial networks are special cases of artificial curiosity (1990) and also closely related to predictability minimization (1991), 2020. 2

[15] Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision, 2017. 1

[16] Yufei Ye, Shubham Tulsiani, and Abhinav Gupta. Shelf-supervised mesh prediction in the wild, 2021. 1