Lightweight Transformer Occupancy Networks for 3D Virtual Object Reconstruction

Claudia Melis Tonti[a] and Irene Amerini[b]
Sapienza, University of Rome, Italy
{*melistonti, amerini*}*@diag.uniroma1.it*

Keywords: Mesh, Generation, Point Cloud, Computer Vision, 3D.

Abstract: The increasing demand for edge devices highlights the necessity for modern technologies to be adaptable to general-purpose hardware. Specifically, in fields like augmented reality, virtual reality, and computer graphics, reconstructing 3D objects from sparse point clouds is highly computationally intensive, presenting challenges for execution on embedded devices. In previous works, the speed of 3D mesh generation has been prioritized with respect to preserving a high level of detail. Our focus in this work is to enhance the speed of the inference in order to get closer to real-time mesh generation.

1 INTRODUCTION

In recent years, the growing importance of computer vision in technologies like autonomous navigation (Urmson et al., 2008), robotic mapping (Zhu et al., 2017), augmented reality (AR) (Alhaija et al., 2017), and gaming has increased the demand for efficient processing of multidimensional data such as images and videos.

Traditional methods for 3D reconstruction, such as Signed Distance Functions (SDFs), require heavy computations and slow processes for mesh generation. Newer approaches like ConvONet (Peng et al., 2020) achieve faster and more efficient 3D reconstruction compared to earlier methods like DeepSDF (Park et al., 2019). However, real-time reconstruction from point clouds remains underexplored, despite its importance for applications on low-performance hardware, such as AR or immersive reality.

Point clouds, created using LiDAR sensors or estimated via Monocular Depth Estimation (MDE) (Papa et al., 2023), act as lightweight 3D mesh compressions. Their sparsity can be adjusted to balance detail retention and computational efficiency, allowing on-demand reconstruction of 3D objects within specific areas, such as those observed in AR viewers.

Our work focuses on efficient 3D object reconstruction for AR applications, with use cases including interior design, game development, and virtual artifact reconstruction (Patil et al., 2023; Virtanen et al., 2020). We leverage lightweight implicit representations based on Convolutional Occupancy Networks (ConvONet) (Tonti et al., 2024), extracting meshes from point clouds to prioritize speed over accuracy. This ensures smooth interaction in AR environments, avoiding delays that could disrupt the user experience.

To enhance ConvONet, we integrate an efficient transformer, capitalizing on its ability to process complex spatial relationships effectively. Our method is validated on the ShapeNet benchmark dataset.

The paper is structured as follows: Section 2 reviews related work, Section 3 introduces the proposed ConvONet with a Dynamic Vision Transformer, Section 4 provides implementation details, Section 5 presents experimental results, and Section 6 concludes with future directions.

2 RELATED WORKS

This section reviews key methods for representing and reconstructing 3D shapes, focusing on various representation techniques. Early approaches, such as Signed Distance Functions (SDF) and Deep SDF (Mescheder et al., 2019; Park et al., 2019), describe surfaces as a continuous function that assigns a signed distance value to every point in 3D space. While effective for defining geometry, SDFs require significant memory and cannot infer surfaces in unobserved regions, leading to gaps in the reconstructed 3D rep-

[a] https://orcid.org/0009-0002-0574-612X
[b] https://orcid.org/0000-0002-6461-1391

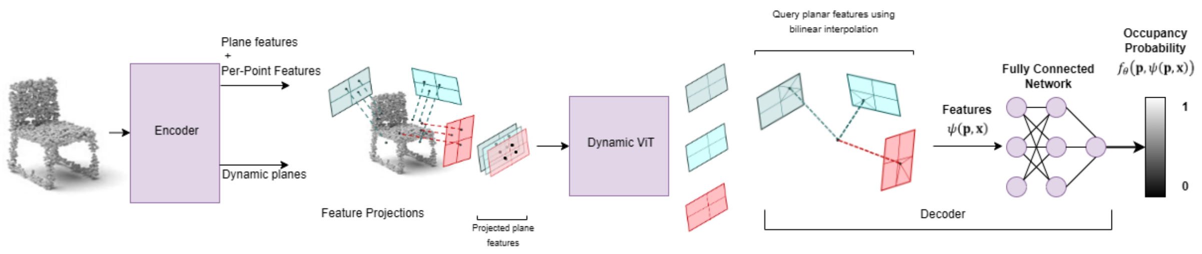

Figure 1: Graphical representation of the proposed ConvONet pipeline. The ConvONet pipeline begins with an input point cloud containing N points. In the Encoder, the ResNet PointNet (Qi et al., 2017) individually encodes these points, while a plane predictor network globally encodes the entire point cloud to learn L dynamic planes and their specific features. These plane features are combined with the per-point features and projected onto the learned dynamic planes. The dynamic planes are summed and overlapped between each other, then processed in the Dynamic Vision Transformer. In the decoder, for a given query point $\mathbf{p} \in \mathbb{R}^3$, the feature vector $\psi(\mathbf{x},\mathbf{p})$, where $\mathbf{x}$ is the input, is derived from bilinear interpolation of the processed feature planes. Finally, a fully-connected network predicts the occupancy probability $f_\theta(\mathbf{p},\psi(\mathbf{p},\mathbf{x}))$ at point $\mathbf{p}$.

resentation (Curless and Levoy, 1996; Choy et al., 2016).

Deep Local Shapes (Chabra et al., 2020) address these limitations by using neural networks to approximate SDFs in small regions, improving inference efficiency by breaking down scenes into independent local features. Implicit function-based methods (Chibane et al., 2020) further advance this idea by representing 3D objects with multi-scale 3D vectors that encode both local and global structures, enabling better handling of complex geometries.

Genova et al. (Genova et al., 2020) introduced Local Deep Implicit Functions (LDIF), a hybrid representation that combines local implicit functions with structured templates to decompose space efficiently. LDIF uses a latent vector representation for finer detail and leverages PointNet (Qi et al., 2017) to encode both local and global features for 3D reconstruction tasks. These representations focus on breaking down large-scale geometries into manageable components while retaining critical surface details.

ConvONet (Peng et al., 2020) proposed a continuous implicit grid representation that combines convolutional layers and occupancy networks to estimate the probability of a point belonging to the surface of the object. This hierarchical encoding of 3D features enables accurate reconstructions and inspired methods like DPConvONet (Lionar et al., 2021a) and Neuralblox (Lionar et al., 2021b). DPConvONet projects point cloud features onto dynamic planes, capturing surface normals for unoriented inputs, while Neuralblox supports nearly real-time 3D scene generation using occupancy grids from point clouds.

Despite these advancements, achieving real-time reconstruction remains challenging. Neuralblox offers faster mesh generation but sacrifices detail, and Lightweight ConvONet (Tonti et al., 2024) attempts to optimize efficiency through architectural compression, yet its inference time is still not real-time. To address these challenges, we propose integrating Dynamic Vision Transformers (Dynamic ViT) (Rao et al., 2023) into Lightweight ConvONet. This integration replaces the bottleneck U-Nets, enabling more efficient processing of point cloud projections while leveraging the transformer's ability to handle complex spatial relationships. This approach aims to improve both speed and accuracy, moving closer to real-time performance.

3 PROPOSED METHOD

Our research aims to increase inference speed to generate fast and high-quality 3D meshes. In this work, we analyze the effects of integrating an efficient transformer architecture into the ConvONet pipeline in order to decrease the generation time.

The rest of this section is organized as follows: Section 3.1 introduces the structure of the network, and Section 3.1.3 presents the architecture of the Dynamic Vision Transformer.

3.1 The Proposed Network

The architecture of the proposed network, shown in Figure 1, has an encoder-decoder structure.

The pipeline, as shown in the figure, begins with a sparse point cloud input. The encoder, detailed in Section 3.1.1, processes this input to extract per-point and global features, which are then used to estimate the dynamic planes. These planes simplify the environment by transforming it from 3D to 2D, reducing computational complexity during encoding.

Once the points are projected onto these planes, the overlapping projections are processed by the Dynamic ViT, as described in Section 3.1.3. In the decoder, outlined in Section 3.1.2, the planar features are utilized to perform bilinear interpolation for each point in 3D space, calculating its occupancy probability. This approach balances efficiency and precision in 3D reconstruction.

3.1.1 Encoder

The encoder-decoder network is based on the groundwork of Lionar et al. The network processes a point cloud input with a ResNet-based PointNet. The encoder extracts per-point features using a ResNet PointNet and learns a set of dynamic planes with a plane predictor network. The ResNet PointNet consists of five ResNet blocks, a PointNet, and a final ResNet block, encoding the point cloud pointwise. The plane predictor network uses a PointNet for global max pooling to predict the set of planes. These features are converted to plane parameters, defining the plane by its normal vector (a, b, c).

The per-point and planar features are summed and projected onto the predicted dynamic planes using a basis change and orthographic projection, followed by normalization as proposed by Lionar et al. Every plane with projection is summed up with the others and then processed with Dynamic Vit (DyVit), presented in Section 3.1.3. The outputs of the transformer are the planar features that will be processed in the Decoder.

3.1.2 Decoder

Given the estimated planar features, the decoder predicts the probability of each point in 3D space being within the occupancy grid. This probability is derived from the sum of the bilinear interpolations of the estimated planar features. The network is based on a fully connected network of five ResNet blocks, with input features summed to the feature vector in each block. After estimating the occupancy probability of each point, the point cloud is turned into a mesh.

3.1.3 Dynamic Vision Transformer

Compared to Convolutional Neural Networks (CNNs), Vision Transformers (ViTs) (Touvron et al., 2022) are better at capturing long-range relationships between pixels, enabling them to process images with more comprehensive detail. The main components of a ViT are:

- *Patch Embedding:* The input image is divided into smaller patches, which are flattened and projected into a fixed-dimensional space. Positional embeddings are added to retain spatial information for each patch.
- *Multi-head Self-Attention (MSA):* This mechanism allows the model to understand global relationships by enabling interactions between different patches (tokens). MSA calculates query (Q), key (K), and value (V) matrices, using softmax to create an attention matrix that captures dependencies across the input. Parallel attention layers enhance the model's ability to capture complex features. It computes query (Q), key (K), and value (V) matrices, transforming scores into probabilities with softmax, thereby forming an attention matrix that aggregates global information across the sequence.
- *Feed-Forward Neural Network (FNN):* After the self-attention step, the output is normalized and processed by a feed-forward network to refine the features further.

Rao et al. (Rao et al., 2023) proposed DyVit which dynamically removes less important tokens from transformer blocks based on their contribution to the output. During training, the threshold for selecting important tokens is adjusted automatically, and only the most relevant tokens are retained during inference. This reduces computational costs while maintaining accuracy. To handle performance drops caused by token pruning, DyVit uses knowledge distillation (KD). A teacher model with all tokens intact guides the pruned student model, helping it learn rich representations and retain performance.

The structure of DyViT adapted for our task is presented in Figure 2. The first part of the student and teacher networks is almost identical: the input, which consists of the 3 overlapped projection planes, is divided into patches and processed by the patch embedder, then, the processed patches are transmitted to the self-attention mechanism. The student model, differently from the teacher model, processes the patches with self-attention. The attention output is processed with the prediction module which is in between the transformer blocks to delete less informative tokens selected by the features given by the previous layer. In this way, fewer tokens are processed in the following layers. This technique is called sparsification and the sparse tokens are processed in the self-attention which returns the output of the model, which is the resulting planar features.

To train DyVit effectively, the total loss is a combination of three components:

- *Cross-Entropy Loss* Rao et al. use the standard cross-entropy to represent the distance between the ground truth and the result of DyViT. In our case, we don't have a ground truth for the projection planes so, we represent instead the distance between the DyVit result and the plane projections given as input. We took this decision because we hypothesize that the nearest points to the input

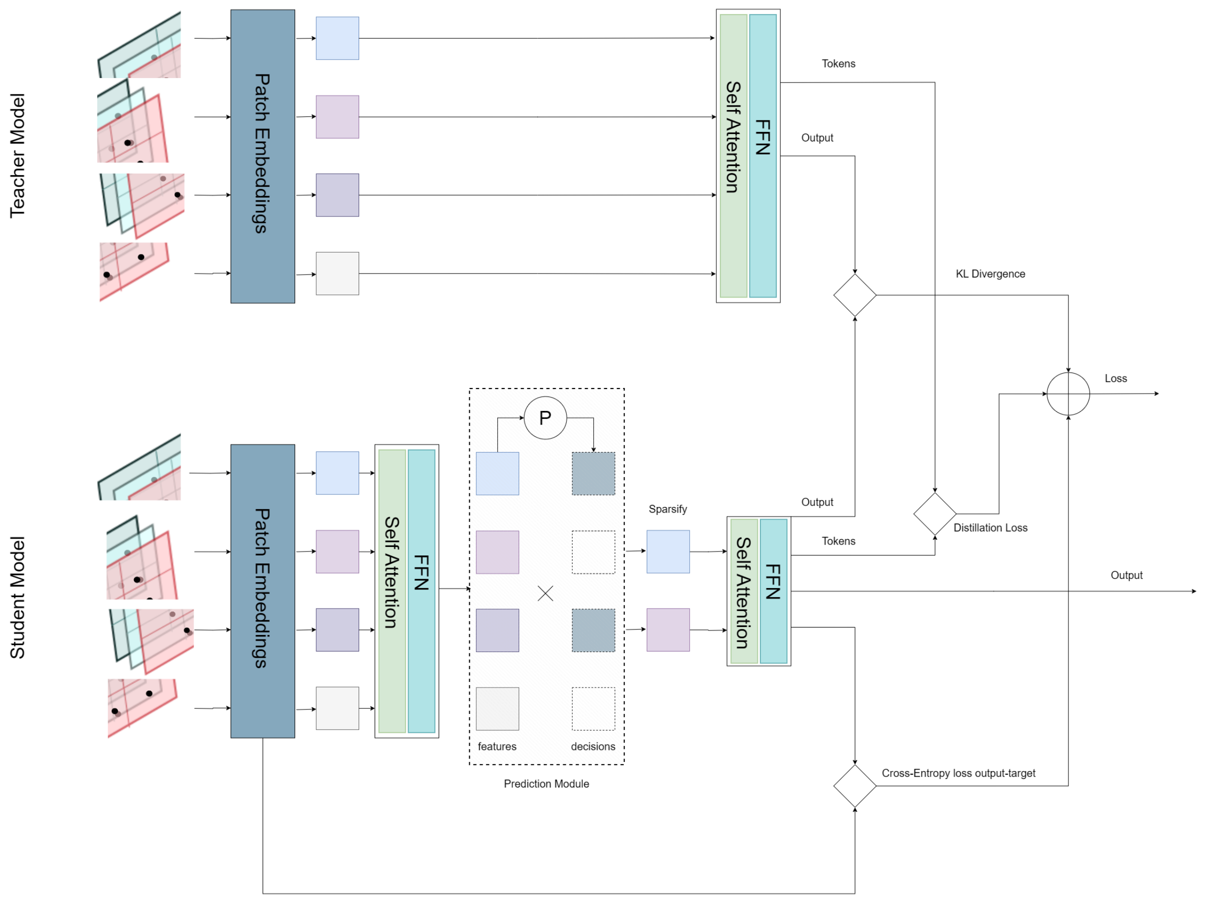

Figure 2: Graphical representation of the student-teacher architecture of Dynamic Vision Transformer with an input of 3 overlapped projection planes.

point cloud points have the highest probability of lying on the surface of the mesh so, this distance should be minimized. The loss is defined in this shape:

$$\mathcal{L}_{ce} = CrossEntropy(\mathbf{y}, \overline{\mathbf{y}}) \tag{1}$$

where $\overline{\mathbf{y}}$ is the input patch of the overlapped projection layers and $\mathbf{y}$ is the prediction of DyViT.

- *Self-Distillation loss* This loss is used to minimize the influence on performance caused by token sparsification. To do so, the final remaining tokens of DyViT should be as close as possible to the tokens of the teacher model. This minimization is represented by the following equation:

$$\mathcal{L}_{distill} = \frac{1}{\sum_{b=1}^{B}\sum_{i=1}^{N}\hat{\mathbf{D}}_i^{b,S}} \sum_{b=1} B \sum_{i=1} N \hat{\mathbf{D}}_i^{b,S}(\mathbf{t}_i - \mathbf{t'}_i) \tag{2}$$

where $\mathbf{t}_i$ and $\mathbf{t'}_i$ are the resulting token of DyVit and the teacher model respectively, $\hat{\mathbf{D}}_i^{b,S}$ is the decision mask for the b-th step and the s-th sparsification stage.

- *KL Divergence* with this loss it is minimized the difference between the results of DyViT and the results of the teacher using KL divergence:

$$\mathcal{L}_{KL} = KL(\mathbf{y}||\mathbf{y}') \tag{3}$$

where $\mathbf{y}'$ is the prediction of the teacher model

The total loss is computed as:

$$\mathcal{L} = \mathcal{L}_{ce} + \lambda_{KL}\mathcal{L}_{KL} + \lambda_{distill}\mathcal{L}_{distill} \tag{4}$$

with λ_{KL} and $\lambda_{distill}$ are set to 0.5. This approach enables DyVit to achieve efficiency by dynamically pruning tokens while retaining the performance of a full-token transformer.

4 IMPLEMENTATION DETAILS

The evaluation of the performances has been executed through object-level reconstruction, using a subset of the ShapeNet dataset (Chang et al., 2015) similar to the Choi et al. (Choy et al., 2016) work. The subset consists of 50,000 meshes along with their corresponding point clouds, covering 13 different object classes. For each class 1/5 of the samples are used for testing, 2/5 for validation, and the rest for training. We used an NVIDIA GeForce GTX 1080 Ti to train the model.

The hidden feature dimensions are 32 for the encoder and the decoder, the resolution of the x, y, and z planes is set to 64, the batch size is 64 and the validation is done every 200 iterations. The learning rate of the Adam optimizer is set at 10^{-4}, and the betas values are set at $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Moreover, we set the depth of the DyVit to 2. We set the number of heads of the transformer attention to 8 and the patch size to 4. In this way, the model reaches convergence after around 12k iterations.

The model employs a Sinusoidal Representation Network (SIREN) as its activation function, introduced by (Sitzmann et al., 2020). SIREN uses sine functions as its basis, which naturally encode high-frequency details, making it particularly effective for processing inputs like images or 3D implicit representations. This periodic activation function ensures efficient training by carefully controlling the activation values and spatial frequencies. The authors of SIREN demonstrated its effectiveness across domains such as images, videos, sound, and 3D data. Their method outperforms traditional architectures, like those using ReLU-based networks, in terms of accuracy, speed, and the ability to capture fine details.

During inference, we apply *Multiresolution Iso-Surface Extraction* (MISE) to extract meshes given the occupancy values as inputs (Mescheder et al., 2019). Because the objective of this work is to decrease the time of mesh generation, we are interested in the inference time, which is expressed in seconds per mesh.

Other metrics used for evaluation are essential to determine the precision of the results and to understand the consistency of the model. These metrics are presented in the following:

Accuracy: calculated using the mean absolute distance (MAD) between the dense points of the predicted mesh and the nearest surface of the ground truth mesh. A lower MAD indicates better accuracy.

$$MAD_{accuracy} = \frac{\sum_i^n \|p_i - g_i\|}{n}$$

Where p_i is the point of index i of the predicted mesh, and g_i is the point of index i of the ground truth.

Completeness: mean absolute distance between the dense points of the ground truth and the surface of the predicted mesh. The lower, the better.

$$MAD_{completeness} = \frac{\sum_i^n \|g_i - p_i\|}{n}$$

Volumetric Intersection over Union: the ratio of the volume of the intersection to the volume of the union of the ground truth and predicted mesh. The higher, the better.

$$IoU = \frac{Volume\,of\,Overlap}{Volume\,of\,Union}$$

Chamfer-$L1$: the average of the accuracy and completeness metrics. The lower, the better.

$$Chamfer\text{-}L1 = \frac{MAD_{accuracy} + MAD_{completeness}}{2}$$

F-score: the average of precision and recall between the predicted mesh and the ground truth. The higher, the better.

$$F\text{-}score = 2\frac{precision \cdot recall}{precision + recall}$$

5 EXPERIMENTS

In this section, we evaluate the performance of our model compared to state-of-the-art approaches in Section 5.1 and perform an ablation study in Section 5.2 to examine how changes in the network architecture impact its results.

5.1 Comparison with Baseline Models

We compared our model with existing methods, referred to as baselines, on the ShapeNet dataset. As shown in Table 1, our model generates a 3D mesh in an average of 0.40 seconds per mesh, making it the fastest among the models tested. As explained in Section 4, for the scope of this work, whose purpose is generating a roughly estimated mesh in less time as possible (plausible mesh) to be refined in a subsequent stage, these metrics are less critical compared to inference time. In Figure 3 we demonstrate the consistency of our generated meshes by doing a visual comparison with the result of the DPConvONet model.

Table 1: Performance comparison between our model and state-of-the-art model. We can easily see that despite our model giving non-optimal results, it is the fastest model with respect to previous works.

Model	Accuracy ↓	Completeness ↓	Chamfer-L_1 ↓	F-score ↑	IoU ↑	Inference Time ↓
(Lionar et al., 2021a)	**0.0047**	**0.0039**	**0.0043**	**0.9433**	**0.8879**	0.86 s/mesh
(Lionar et al., 2021b)	0.0986	0.0999	0.0992	0.1851	0.7901	0.65 s/mesh
(Tang et al., 2021)	0.0097	0.0082	0.0090	0.7045	0.7429	0.45 s/mesh
(Tonti et al., 2024)	0.0058	0.0048	0.0053	0.9037	0.8499	0.48 s/mesh
Our	0.0342	0.0263	0.0302	0.3297	0.4924	**0.40** s/mesh

Table 1 presents the evaluation metric values obtained by the compared models. These values were computed using the pre-trained weights provided in the GitHub repositories of DPConvONet, Neuralblox, SA-ConvONet, and LConvONet.

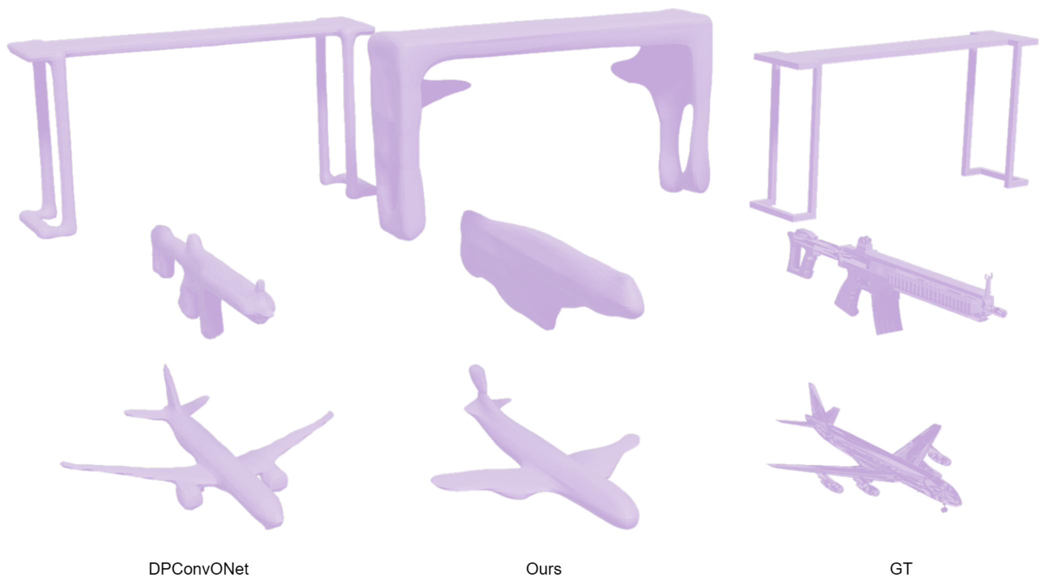

Figure 3: Visual comparison of DPConvONet, our model, and the ground truth (shown from left to right). This comparison demonstrates that our model produces consistent results, comparable to the meshes generated by the state-of-the-art method, DPConvONet.

The obtained results present a boost of +53% (-0.46s) with respect to DPConvONet, which is the model with the best-resulting evaluation metrics, a boost of +11% (-0.05s) for SA-ConvONet, which has the lowest inference time, and a boost of +16% (-0.08s) to LConvONet, which generates high-quality details maintaining the inference time low. Furthermore, when compared to Neuralblox, our model demonstrates a significant improvement with a +38% boost (-0.25s) and achieves higher evaluation metrics, indicating more precise mesh generation.

5.2 Ablation Study

In Table 2 we propose different network configurations. We replace the Siren activation function with ReLU to analyze the performance difference between the two architectures. We may notice that in this case, the evaluation metrics values present a general worsening, and the inference time increased. This confirms the results in (Sitzmann et al., 2020), which demonstrate SIREN's superior efficiency and accuracy compared to standard activation functions like ReLU, Tanh, or GELU.

Table 2: Performance comparison between different configurations of the network.

Model	Accuracy ↓	Completeness ↓	Chamfer-L_1 ↓	F-score ↑	IoU ↑	Inference Time ↓
Our	0.0342	0.0263	0.0302	0.3297	0.4924	**0.40** s/mesh
Network without Siren	0.0539	0.0312	0.0426	0.2293	0.3904	0.4521 s/mesh
Network with 3 DyViT	0.0361	0.026	0.031	0.3436	0.5124	0.4585 s/mesh

In the third row of the table, we present the results given by the model when it processes separately the 3 planes with 3 different DyViTs. From such results, we can notice that not only does the inference time increase but processing the planes separately doesn't really affect the accuracy of mesh generation. The motivation lies in the ability of the transformer to extract both local and global features, which allows DyViT to easily interpret the projections onto the 3 planes also when such planes are overlapped.

These findings highlight that the use of SIREN and overlapping projection planes contributes significantly to the efficiency and quality of our model.

6 CONCLUSIONS

This study explores the integration of an efficient vision transformer, such as DyViT, into a Convolutional Occupancy Network, replacing three U-Nets used for processing point cloud projections. The transformer's ability to handle both local and global features allows for overlapping projection planes, significantly alleviating bottlenecks between the encoder and decoder.

We also highlight the importance of the SIREN activation function in preserving a high level of detail within the model's output.

Looking ahead, we aim to further reduce inference time by refining the mesh representation process. In computer graphics, 3D artists often create a coarse object shape and enhance it with bump, normal, and parallax mapping to maintain detail. Inspired by this, our future objective is to represent the point cloud using a single projection plane and use our model to densify it. This approach would enable us to generate parallax mapping for a coarse 3D mesh, maintaining visual detail while optimizing computational efficiency.

REFERENCES

Alhaija, H. A., Mustikovela, S. K., Mescheder, L., Geiger, A., and Rother, C. (2017). Augmented reality meets deep learning for car instance segmentation in urban scenes. In *British machine vision conference*, volume 1.

Chabra, R., Lenssen, J. E., Ilg, E., Schmidt, T., Straub, J., Lovegrove, S., and Newcombe, R. (2020). Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16*, pages 608–625. Springer.

Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al. (2015). Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*.

Chibane, J., Alldieck, T., and Pons-Moll, G. (2020). Implicit functions in feature space for 3d shape reconstruction and completion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6970–6981.

Choy, C. B., Xu, D., Gwak, J., Chen, K., and Savarese, S. (2016). 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pages 628–644. Springer.

Curless, B. and Levoy, M. (1996). A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312.

Genova, K., Cole, F., Sud, A., Sarna, A., and Funkhouser, T. (2020). Local deep implicit functions for 3d shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4857–4866.

Lionar, S., Emtsev, D., Svilarkovic, D., and Peng, S. (2021a). Dynamic plane convolutional occupancy networks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1829–1838.

Lionar, S., Schmid, L., Cadena, C., Siegwart, R., and Cramariuc, A. (2021b). Neuralblox: Real-time neural representation fusion for robust volumetric mapping. In *2021 International Conference on 3D Vision (3DV)*, pages 1279–1289. IEEE.

Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., and Geiger, A. (2019). Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470.

Papa, L., Russo, P., and Amerini, I. (2023). Meter: a mobile vision transformer architecture for monocular depth estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–1.

Park, J. J., Florence, P., Straub, J., Newcombe, R., and Lovegrove, S. (2019). Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174.

Patil, A. G., Qian, Y., Yang, S., Jackson, B., Bennett, E., and Zhang, H. (2023). Rosi: Recovering 3d shape interiors from few articulation images. *arXiv preprint arXiv:2304.06342*.

Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., and Geiger, A. (2020). Convolutional occupancy networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 523–540. Springer.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660.

Rao, Y., Liu, Z., Zhao, W., Zhou, J., and Lu, J. (2023). Dynamic spatial sparsification for efficient vision transformers and convolutional neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9):10883–10897.

Sitzmann, V., Martel, J. N. P., Bergman, A. W., Lindell, D. B., and Wetzstein, G. (2020). Implicit neural representations with periodic activation functions.

Tang, J., Lei, J., Xu, D., Ma, F., Jia, K., and Zhang, L. (2021). Sa-convonet: Sign-agnostic optimization of convolutional occupancy networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6504–6513.

Tonti, C. M., Papa, L., and Amerini, I. (2024). Lightweight 3-D Convolutional Occupancy Networks for Virtual Object Reconstruction . *IEEE Computer Graphics and Applications*, 44(02):23–36.

Touvron, H., Cord, M., El-Nouby, A., Verbeek, J., and Jégou, H. (2022). Three things everyone should know about vision transformers. In *European Conference on Computer Vision*, pages 497–515. Springer.

Urmson, C., Anhalt, J., Bagnell, D., Baker, C., Bittner, R., Clark, M., Dolan, J., Duggins, D., Galatali, T., Geyer, C., et al. (2008). Autonomous driving in urban environments: Boss and the urban challenge. *Journal of field Robotics*, 25(8):425–466.

Virtanen, J.-P., Daniel, S., Turppa, T., Zhu, L., Julin, A., Hyyppä, H., and Hyyppä, J. (2020). Interactive dense point clouds in a game engine. *ISPRS Journal of Photogrammetry and Remote Sensing*, 163:375–389.

Zhu, Y., Mottaghi, R., Kolve, E., Lim, J. J., Gupta, A., Fei-Fei, L., and Farhadi, A. (2017). Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3357–3364. IEEE.