Project: Convolutional Occupancy Networks

Kirill Gelvan 03781708
Anastasiia Chaikova 03786365
Omar Al-Dabooni 03726693
Yi Wang 03781653

Abstract

This paper reviews and extends a deep learning framework for implicit 3D shape representation, particularly focusing on the Convolutional Occupancy Networks approach [4]. We investigate how noisy 3D point clouds can be transformed into reliable and detailed occupancy fields by leveraging convolution-based encoders and decoders, as well as specialized techniques such as sinusoidal activation functions and positional encodings. Furthermore, we explore incorporating auxiliary conditioning (e.g. class labels, text descriptions) via learned embeddings to improve robustness, especially on thin or complex structures. We present comprehensive experimental results on variations of our method, using different positional encoding configurations and activation functions, highlighting the conditions under which each variant excels. Our findings suggest that positional encodings applied before the decoder, combined with sinusoidal activations and effective handling of global conditioning, yield superior performance on challenging tasks while preserving desirable translational equivariance properties. The code for the reproduction of the experiments can be found on GitHub.

1. Introduction

Learning-based implicit representations have recently transformed the landscape of 3D shape reconstruction. In more traditional approaches, practitioners often discretize the 3D space into voxels or try to deform a template mesh, which can lead to large memory requirements, restricted topologies, or coarse geometry approximations. By contrast, implicit methods like Occupancy Networks [3] and DeepSDF [5] represent geometry as a continuous function that maps 3D coordinates (conditioned on input observations) to occupancy or signed distance values. One key advantage of this paradigm is that it forgoes explicit 3D grids or meshes, thereby offering flexibility and more efficient use of model capacity.

Despite these benefits, early implicit methods relied on fully-connected decoders, which capture global features but are not well-suited for encoding spatial detail. As introduced in [4], Convolutional Occupancy Networks address this limitation by embedding the input (e.g., a noisy point cloud) into planar or volumetric feature maps processed by convolutional encoders and decoders. Convolutional layers, already well-established for image-based tasks such as classification and semantic segmentation, bring translation equivariance and hierarchical receptive fields to the 3D reconstruction setting. This design better handles large scenes and fine-grained local details alike.

Our work further develops this direction by investigating new techniques to handle noisy point clouds while preserving the fidelity of complex shapes, including thin and intricate structures. First, we adopt *positional encodings* [7] and *sinusoidal activations* [6] to inject high-frequency signals into the occupancy function. This enables the network to resolve subtle local details without losing overall shape coherence. Second, we incorporate *multimodal information*, such as class labels or text-based embeddings, which provide additional semantic cues to enhance robustness under noise and ambiguity. We compare different fusion strategies for these embeddings, showing that the right approach can yield significant performance gains.

In the following sections, we give details on how each of these components (positional encoding, sinusoidal activations, and multimodal conditioning) can be integrated into the Convolutional Occupancy Networks framework. Through experiments on both synthetic and real-world data, we demonstrate consistent improvements in reconstruction quality, offering a reliable and scalable approach for 3D tasks that demand accuracy, robustness, and efficiency.

2. Related Works

Implicit 3D Representations There has been a surge of interest in neural implicit representations for 3D shapes. Occupancy Networks [3] learn a continuous function that outputs an occupancy probability for any 3D coordinate, while DeepSDF [5] models signed distance fields. Although these methods perform well for single objects, they typically rely on fully connected decoders, which can struggle with fine local details in larger scenes. Recent work has also investigated hierarchical and locally adaptive representations to overcome these limitations, further expanding the

scope of implicit models in representing complex geometries.

Convolutional Occupancy Networks Peng *et al.* [4] introduce Convolutional Occupancy Networks, which encode 3D inputs into either 2D or 3D feature maps processed by convolutional encoders and decoders, effectively capturing both local and global information. This framework outperforms standard Occupancy Networks on complex or larger-scale tasks by leveraging the inherent translation equivariance of convolutions. Subsequent studies have explored various modifications, such as different projection schemes and fusion strategies, to further enhance the reconstruction of fine details and reduce sensitivity to noise.

Positional Encoding and Sinusoidal Activations Positional encoding, popularized in Transformers [7], has been shown to be beneficial for coordinate-based representations such as those used in neural radiance fields. Similarly, sinusoidal activations (SIREN) [6] enable a richer representation of local variations in implicit neural fields, improving the capture of high-frequency details. These techniques allow networks to approximate subtle geometric variations more effectively. Their integration into 3D reconstruction frameworks has led to significant improvements in capturing thin or intricate structures.

More Enhancements to Implicit Representations Reconstruction Recent advances have extended the implicit representation framework with localized modeling and transformer-based architectures. For example, Genova *et al.* [8] propose *Local Deep Implicit Functions for 3D Shape*, where the overall shape is decomposed into overlapping local regions, each modeled by its own implicit function. This localized approach not only captures fine geometric details but also enhances the integration of local cues into a coherent global reconstruction. Similarly, Jiang *et al.* [9] introduce *Local Implicit Grid Representations for 3D Scenes* by partitioning a scene into small volumetric grids, each with its dedicated implicit representation, which allows the network to better capture spatial variations and complex local structures. Extending the scope further, Niemeyer *et al.* [10] present *Occupancy Flow: 4D Reconstruction by Learning Particle Dynamics*, where the model not only reconstructs static geometry but also learns dynamic changes over time by tracking particle flows, thereby enabling robust reconstruction of dynamic scenes. In addition, transformer-based architectures such as the Point Transformer [11] have been introduced to process point clouds by leveraging self-attention mechanisms, which capture long-range dependencies and complex spatial relationships; this ability to model global context holds promise for enhancing implicit representations in 3D reconstruction tasks.

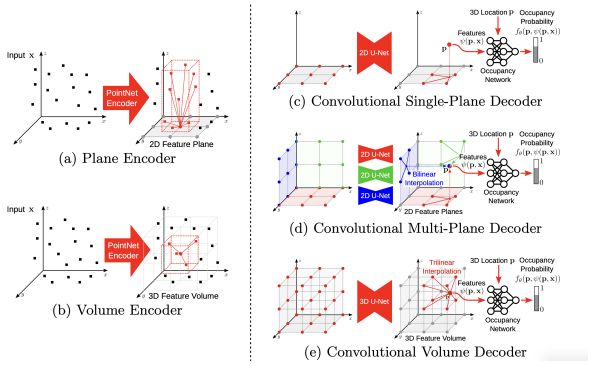

Figure 1. Overall architecture. A noisy point cloud is transformed into a 2D or 3D feature representation via a convolutional encoder. These features are then interpolated at query points and passed to an MLP that outputs occupancy values.

3. Method

Our approach builds on the Convolutional Occupancy Networks paradigm while introducing key modifications aimed at handling noisy point clouds and incorporating additional semantic information. Figure 1 provides an overview of the main pipeline. The input is a noisy point cloud $\mathcal{P}$ of a shape or scene, and optionally additional modality embeddings. The network produces a continuous occupancy function $f(\mathbf{p}) : \mathbb{R}^3 \rightarrow [0, 1]$. In broad terms, we map an input point cloud into a structured feature representation (2D or 3D) via a convolutional encoder, then decode this representation into continuous occupancy predictions. We apply positional encodings and sinusoidal activations to better capture high-frequency details, and we also leverage optional semantic embeddings that can guide the reconstruction process when noisy or ambiguous data are encountered. Below, we elaborate on each main component of this pipeline.

3.1. Convolutional Occupancy Networks

Encoder and Feature Aggregation We begin by normalizing the input point cloud coordinates, ensuring that the data lies within a defined range. The encoder then extracts local features by projecting each point onto one or more canonical planes or into a 3D volume. These planar or volumetric features are aggregated through either average or max pooling, effectively converting the sparse point-based input into a dense grid of features. To learn richer representations, we stack convolutional layers (often in a UNet-like fashion) that capture progressively larger receptive fields. This process yields a feature map that encodes both local geometry (e.g., fine edges and corners) and broader contextual cues (e.g., overall shape structure). By incorporating

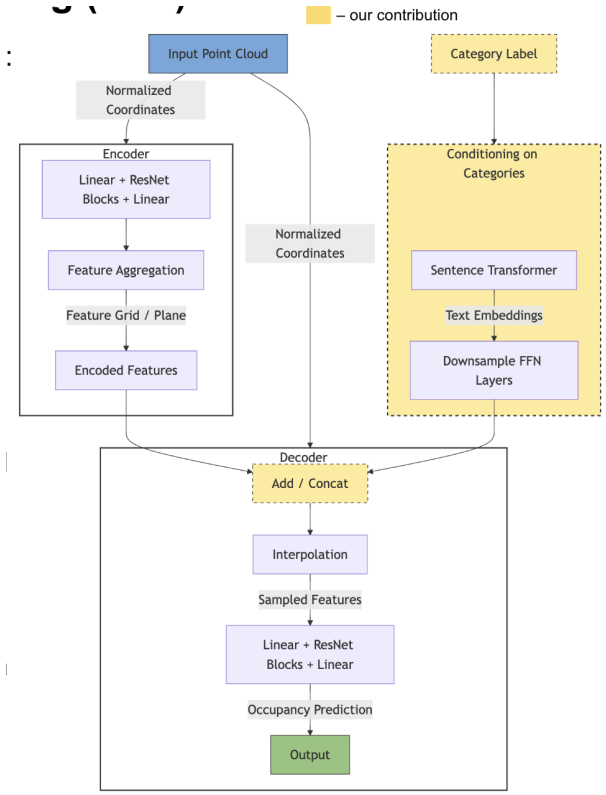

Figure 2. Our modification to incorporate text descriptions or category labels. The embeddings (e.g., from a SentenceTransformer model) are downsampled and then fused (via Add or Concat) with the interpolated convolutional features before occupancy prediction.

enough depth and skip connections, the encoder can adapt to significant levels of noise while retaining important shape details. Finally, we store the resulting features in a plane or volume of dimension $H \times W$ (for 2D) or $H \times W \times D$ (for 3D), enabling flexible resolution choices depending on memory constraints.

Decoder and Interpolation Once we have planar or volumetric features, the model must infer the occupancy probability at arbitrary 3D query points. To this end, we perform bilinear or trilinear interpolation on the feature map based on the normalized coordinate of the query point. The interpolated feature vector, denoted by $\psi(\mathbf{p}, x)$, is then fed into an MLP that learns to map from $(\mathbf{p}, \psi(\mathbf{p}, x))$ to a scalar occupancy value. We retain the structure of a small ResNet-based decoder [3], where $\psi(\mathbf{p}, x)$ can be concatenated with or added to intermediate activations in the MLP layers. By preserving local details through convolutional encoding and then reintroducing them at inference time via interpolation, the network gains both local awareness and global context. This setup also inherently supports querying at any spatial resolution, since the occupancy function is continuous by design.

3.2. Positional Encoding and Sinusoidal MLP

Hoping to capture high-frequency details, we apply *positional encoding* to the 3D query $\mathbf{p}$ (especially before the decoder MLP). We use:

$$\gamma(\mathbf{p}) = \big[\mathbf{p},\ \sin(2^0\pi\mathbf{p}),\ \cos(2^0\pi\mathbf{p}),\ \ldots,\ \sin(2^L\pi\mathbf{p}),\ \cos(2^L\pi\mathbf{p})\big],$$

where L is typically small (e.g. $L = 6$).

We also explore *sinusoidal activations* (SIREN) in the final occupancy MLP f_θ, for instance:

$$f_\theta(\mathbf{p}, \psi(\mathbf{p}, x)) \ = \ \mathrm{SineLayer}\big(\ldots(\gamma(\mathbf{p}) + \psi(\mathbf{p}, x))\big).$$

These trigonometric expansions enable modeling of very thin or complex shapes that standard ReLU MLPs may struggle with.

3.3. Multimodal Embedding Fusion

A critical extension of our model is the ability to incorporate additional semantic signals, such as category labels or text embeddings, to disambiguate noisy input data. Figure 2 illustrates our modification, where we feed the external input (e.g., a SentenceTransformer-based embedding) through a small feed-forward downsampling network to match the dimension of the convolutional features. When text or class labels are available, we embed this information via:

$$\mathbf{e} = \mathrm{Embed}(\mathrm{class}) \quad \text{or} \quad \mathbf{e} = \mathrm{SentenceTransformer}(\mathrm{class}).$$

We then fuse these embeddings with the interpolated features at the decoder stage by either summing them (*Add*) or concatenating them (*Concat*). In practice, the choice between Add or Concat can be decided by validation performance, with Concat providing a more expressive representation but slightly increasing computational overhead. These semantic cues often prove crucial when the point cloud is sparse or contains heavy noise, as the textual or label-based context can guide the network toward more plausible local geometry.

3.4. Occupancy Prediction and Training

Finally, the occupancy probability for query point $\mathbf{p}$ is:

$$o(\mathbf{p}) \ = \ \sigma\big(\underbrace{f_\theta\big(\gamma(\mathbf{p}),\ \psi(\mathbf{p}, x)\big)}_{\text{MLP}}\big), \tag{1}$$

where $\psi(\mathbf{p}, x)$ is the interpolated convolutional feature at location $\mathbf{p}$, and σ is the logistic sigmoid.

We train by minimizing binary cross-entropy over randomly sampled points:

$$\mathcal{L} \ = \ -\sum_{(\mathbf{p},\, o^*)} \Big[o^* \log o(\mathbf{p}) \ + \ (1 - o^*) \log(1 - o(\mathbf{p}))\Big],$$

where o^* is the ground-truth occupancy (1 inside the surface, 0 outside).

During inference, we use standard iso-surface extraction (*e.g.* Marching Cubes) or MISE [3] to extract meshes from $o(\mathbf{p})$.

As a result, we have reproduced the results of the authors an all classes closely (due to limited amount of compute resources). Details can be seen on the Figure 3:

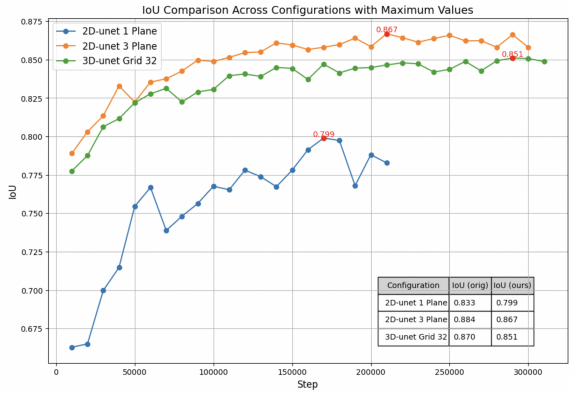

Figure 3. Comparison of the baseline trainings

3.5. New Data Classes

Our work wants to explore the novel classes in the ShapeNet dataset. Specifically, we employed the "bus" and "bathtub" classes from ShapeNet, subjecting the original meshdata to rendering and voxelization processes. The general workflow is illustrated in Figure 4:

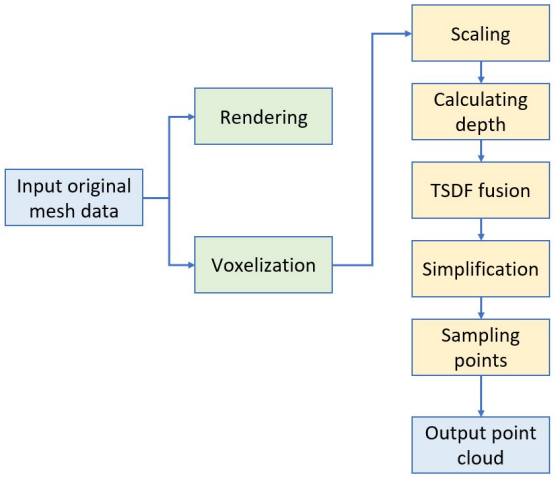

Figure 4. Our processing flow on the new classes of ShapeNet data: bus and bathtub.

However, the efficacy of the generated data in subsequent experiments was suboptimal, primarily due to our limitation of sampling points solely on the object surfaces in the final step. Furthermore, the selection of buses and bathtubs as new classes introduced a predominance of cubic shapes within the dataset. Consequently, sampling points within the object interiors provided minimal enhancement to shape characterization. As a result, we have decided to conduct all of our experiments on 3 of the most popular classes in the dataset - "table", "car" and "airplane".

4. Results

We validate the proposed framework on **ShapeNet** for single-object reconstruction inspired by [4]. Our primary metric is IoU (intersection over union). IoU is the ratio of the volume intersection over the volume union between the predicted mesh and ground-truth shape. We also use Chamfer distance and F-Score as additional measures. We train a total of 20 models to explore how different configurations of positional encoding, sinusoidal activation, and textual embeddings affect performance under varying noise conditions.

4.1. Positional Encoding Variants

Table 1 shows how placing positional encodings only before the decoder yields higher IoU (0.824) than also placing them in the encoder or in both locations. We hypothesize that preserving translational equivariance in the encoder is beneficial. Adding normalization further boosted results.

Table 1. Positional Encodings variants

Configuration	IoU
Baseline (3plane_3class)	0.798
+PosEnc(Decoder)	0.820
+PosEnc(Decoder)+Norm	**0.824**
+PosEnc(2x)	0.608
+PosEnc(2x)+Norm	0.527

(2x) indicates applying PosEnc before both encoder & decoder.

4.2. Activation Function Variants

Table 2 compares ReLU (baseline) and sinusoidal ("Sin") activations in the final MLP. The sinusoidal approach yields a significant improvement (0.837 vs. 0.798 IoU) over the baseline. However, combining sinusoidal with repeated positional encodings can lead to overfitting or training instabilities if not carefully managed.

4.3. Embedding Fusion: Add vs. Concat

We additionally test the effect of how we incorporate an external embedding from a pretrained embedding model or single Embedding layer for short text descriptions or class labels accordingly. Table 3 shows that, particularly for an

Table 2. Activation Function variants

Configuration	IoU
Baseline (3plane_3class)	0.798
+Sin	**0.837**
+Sin+PosEnc(Decoder)	0.816
+Sin+PosEnc(2x)	0.510

unfrozen pretrained model, concatenation outperforms addition (0.809 vs. 0.787 IoU). A standalone embedding layer also benefits more from concatenation.

Table 3. Comparison of training on 3 popular classes (IoU).

Configuration	Add	Concat
Baseline (3plane_3class)	0.798	
SentenceTransformer* (frozen)	**0.798**	0.794
SentenceTransformer* (unfrozen)	0.787	**0.809**
Embedding Layer	0.793	**0.810**

*MiniLMv2 with down-projection.

4.4. Combining All

Finally, we tested the combination of *Sin + positional encodings (decoder) + SentenceTransformer embedding* in a single model. We achieve an IoU of $\mathbf{0.842}$, surpassing all other tested configurations. This combination particularly excels at reconstructing thin or intricate geometry, validating our design hypothesis.

Table 4. All Combined: Sin + PosEnc(Decoder)+Norm + SentenceTransformer concat. The values are averaged by all classes.

Configuration	IoU	Chamfer-L1	F-Score
Baseline	0.798	0.0076	0.82
All Combined	**0.842**	**0.0071**	**0.85**

5. Conclusion

We presented an in-depth review and set of extensions to Convolutional Occupancy Networks for recovering implicit surfaces from noisy 3D point clouds. We demonstrated that:

- **Positional encodings** should be applied primarily before the decoder, preserving the translational equivariance learned by convolutional encoders.
- **Sinusoidal activations** help recover high-frequency details, boosting IoU over standard ReLU networks.
- **Class Embeddings** are most effective when *concatenated* with convolutional features, rather than added, especially when the encoder is pre-trained on a text corpus.

Future work can investigate alternative normalization layers for large-scale 3D scenes, more sophisticated fusion with language or image embeddings, or robust uncertainty estimation. Additional directions include conducting more experiments to quantify the variance in IoU and further refining the data generation pipeline, especially to integrate new ShapeNet classes and evaluate broader object categories. Finally, we plan to port these successful ideas to other 3D tasks, such as scene completion or dynamic reconstruction, to explore their full potential in real-world applications.

References

[1] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, 2019.

[2] B. Mildenhall, et al. NeRF: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[3] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, 2019.

[4] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger. Convolutional occupancy networks. In *ECCV*, 2020.

[5] J. J. Park, et al. DeepSDF: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019.

[6] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein. Implicit neural representations with periodic activation functions. In *NeurIPS*, 2020.

[7] A. Vaswani, et al. Attention is all you need. In *NeurIPS*, 2017.

[8] K. Genova, F. Cole, A. Sud, A. Sarna, and T. Funkhouser. Local Deep Implicit Functions for 3D Shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[9] C. Jiang, A. Sud, A. Makadia, and T. Funkhouser. Local Implicit Grid Representations for 3D Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[10] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger. Occupancy Flow: 4D Reconstruction by Learning Particle Dynamics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2020.

[11] H. Zhao, L. Jiang, J. Fu, and H. Yu. Point Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.