Learning Dual Hierarchical Representation for 3D Surface Reconstruction

Jiyoon Shin[1], Youngwook Kim[2*], Sangwoo Hong[1], and Jungwoo Lee[1]

[1] Seoul National University
[2] Kookmin University
jiyoonshin@cml.snu.ac.kr, youngwook@kookmin.ac.kr,
{tkddn0606, junglee}@snu.ac.kr

Abstract. Neural implicit functions have proved successful in representing 3D shapes or surfaces at arbitrary resolutions and with high fidelity. Unfortunately, between the various forms of reconstruction tasks neural implicit representation methods target, reconstructing from discrete voxels remains limited because of the computational complexity involved. We address this problem by introducing Dual Hierarchical Representation (DHR), which allows for faithful reconstructions under constrained computation by hierarchical encoding, decoding, and training procedures. A hierarchical latent feature code set is produced by first encoding the sparse voxelized shape into multi-scale feature grids and then grid-sampling each feature with a query point. The proposed transformer decoder then incorporates individual latent codes in hierarchical order, directing feature-to-3D projection and modeling the interaction of latent features with occupancies via cross-attention. At the training phase, representations derived from all feature hierarchies are integrated with varying contributions for another global-to-local learning technique. Experiments verify that DHR gains representation power by outperforming various baselines by voxel reconstruction tasks. It also shows robustness against different shape categories and gains the potential for being useful in the wild thanks to the generalization ability the transformer carries. Our code is available at `https://github.com/JYeShin/DualHierarchicalRepresentation`.

1 Introduction

Representing and reconstructing shapes are two inseparable tasks that have long been fundamental problems in 3D computer vision. To reconstruct high-fidelity shapes or surfaces from discrete or inefficient 3D data (such as point clouds, voxel grids, and single or multiple images), ways to represent the shape compactly is a vital task. The immense field of research in shape representation thus proves their fidelity by reconstructing sparse shapes into dense ones.

In recent years, neural fields for implicitly representing shapes have gained huge popularity by achieving faithful reconstruction performances. Representative methods include occupancy fields [24], signed distance function [26], neural

* Corresponding author.

radiance fields (NeRF) [25, 30], and Gaussian splatting [17]. The former two focus on modeling the correspondences between a particular point in space and the underlying surface; thus are suitable for reconstructing synthetic shapes and objects. On the other hand, the latter two focus on recovering a 3D scene from multiple images taken from several viewpoints; thus are applicable to reconstructions in real-world circumstances.

Unfortunately, between the reconstruction tasks the former two attempts to solve, reconstructing from sparse voxels remains inactive and limited to a few papers [4, 5, 24, 28, 38]. While voxels are the early building blocks of shapes and can be easily processed by adapting learning-based image processing techniques, they suffer from the computation cost and efficiency that grows cubically with the resolution. It thus restricts reconstruction quality given limited computation.

In this paper, we propose **Dual Hierarchical Representation (DHR)** for 3D surface reconstruction from sparse voxels given a restricted input resolution and computation. DHR leverages a hierarchical encoding, decoding, and training scheme to capture global-to-local features of the particular shape, integrate the relationships, and provide updated occupancy information. The key insights are two-fold.

1) Transformer decoder: Inspired by the recent great success of transformers in the field of computer vision [12, 34, 42] by their capacity, scalability, and ability to model the data distribution, we integrate a decoder leveraging transformer blocks. It is discovered that the attention mechanism has remarkable capability in learning intricate semantic abstractions from input tokens. Thus, the transformer is naturally suitable for exploring the feature-to-feature relationships of a 3D shape's different semantic parts. Multi-level correspondences and associations between the 3D volume features and occupancies within the transformer decoder structure can be jointly explored.

2) Global-to-local learning: Additionally, the transformer decoder block and training objectives implement a global-to-local learning paradigm. A hierarchical latent feature code set, including coarse to fine-level features, encoded from the encoder is incorporated in the proposed feature-to-occupancy decoder in order of global to local. Representations can start at a coarse geometry and gradually add finer-featured details that are relatively hard to capture. In the training phase, output predictions from all stages of latent codes are exposed to learn the neural field representing the continuous surface. A mild training regime is applied to representations derived from global stages, whereas strict training is employed to those derived from local stages. As part of the global-to-local learning strategy, this learning process also allows the representations to start from an outlined rough geometry and progressively add refinements.

The strengths of DHR are demonstrated by various experiments, compared with previous voxel reconstruction methods. Faithful reconstruction results are shown in terms of both visualization and measures and the potential to perform in the wild is verified by out of category generalization tasks. Diverse ablation studies verify the efficacy of the proposed architecture choice and global-to-local learning strategy. Additionally, auxiliary experiments on sparse point cloud com-

pletion tasks validate that our method not only improves voxel reconstructions but shows representation power by another form of input.

The main contributions are summarized as follows.

- We propose Dual Hierarchical Representation (DHR), a novel framework leveraging hierarchical encoding, decoding, and training to reconstruct high-fidelity surfaces from sparse voxels given a limited computation.
- A decoder based on the transformer architecture is designed to project latent features onto occupancy tokens via cross-attention.
- A novel global-to-local learning paradigm is devised, integrating hierarchical inputs to the decoder and hierarchical losses in the training phase.
- Powerful representations by various reconstruction tasks and sparse input forms are demonstrated, and the generalization ability is also evaluated.

2 Related Work

2.1 Explicit Representations

Three methods are popular for explicit shape representations: voxel grids, point clouds, or meshes. Voxels [6, 21–23, 31] are easily used in learning as they are an intuitive extension of pixels in 2D images. Unfortunately, memory usage grows cubically with respect to the resolution, which limits the reconstruction of high-fidelity shapes. Point clouds [10, 29, 35] are popular representations as they are direct outputs of various sensors and computer vision algorithms. However, point clouds can not topologically represent true watertight shape. Meshes [16, 18, 32] are flexible for deformation and alignment between shapes, but they face challenges in representing detailed geometry. They have difficulty with thin or intricate structures like hair, fur, or grass, often requiring excessively dense meshes, which can be inefficient and result in artifacts.

2.2 Neural Implicit Representations

In recent years, implicit representation methods have shown advances over explicit methods by the ability to generate shapes with arbitrary topology and infinite resolution. They represent continuous iso-surfaces through neural fields, which learn a mapping between a query point and a context vector to a specific value that indicates the relationship between the point and surface.

Global Deep Implicit Function Methods. Given a coarse shape encoded as a single latent vector and the coordinates of a query point, early global methods learn a function that maps it to either a binary occupancy value (OccNet) [3, 24] or a signed distance function (deepSDF) [8, 26]. In another branch, template-based methods [7, 36, 41] address issues in algorithms that do not provide any correspondences between shapes of the same class. Providing a guideline for every category, they provide flexibility and scalability to deform the template to

another. Unfortunately, all global methods reveal limitations in preserving details of local surfaces due to the fixed dimensionality of a single global code [20].

Local Deep Implicit Function Methods. Local methods use localized latent vectors either defined on a regular [2,4,5,15,28,33,38] or irregular [1,9,11,20,39] grid to overcome the problems shown in global representations [40].

In the case of regular grids, for example, ConvONet [28] learns regular latents on a 2D or 3D grid as a follow-up work of OccNet [39]. IFNet [4], NDF [5], and GIFS [38] utilize multi-scale latent grids of different resolutions to provide shape details from various scales. Shapes or scenes are divided into volumes and local latent codes are learned afterward in LIG [15] and DeepLS [2]. Different from regular grids, irregular grids allow each latent to have an arbitrary position in space. POCO [1] proposes to use point cloud convolutions and computes latent vectors at each input point. For dynamic optimization, local codes with learnable position vectors are explicitly associated in DCC-DIF [20]. It also introduces a novel code position loss to guide more local codes to be distributed around shape details. To be compatible with the transformer architecture, shapes are encoded as a fixed length sequence of position and latent pair tuples in 3DILG [39]. LDIF [11] provides 3D shape representations that decompose space into a structured set of learned implicit functions

Similar to local DIF methods, we seek to capture detailed shape features, and by incorporating a novel dual-hierarchical process and a transformer architecture, it achieves superior performance compared to other methods.

3 Method

In the subsequent sections, the proposed Dual Hierarchical Representation, which leverages hierarchical encoding, decoding, and optimization schemes, is presented in detail. Given an input shape and continuous query point, they are encoded into a set of hierarchical latent feature codes in Section 3.1. It is then followed by a feature-to-occupancy decoder that projects the latent features onto occupancy tokens via cross-attention in hierarchical order (Section 3.2). The final decoder output is passed to a multi-layer perception to predict the occupancy for surface reconstruction (Section 3.3). The training objectives including hierarchical losses for global-to-local learning and the surface extraction process at the desired resolution are described in Section 3.4 and Section 3.5, respectively.

3.1 Hierarchical Latent Feature Code Set Encoder

The voxelized input shape $X \in \mathcal{X}$, where $\mathcal{X} = \mathbb{R}^{N\times N\times N}$, is first encoded into a set of multi-scale feature grids as

$$\{F_k\}_{k\in[1,\dots,R]} = \text{multi-feature}(X), \quad F_k \in \mathcal{F}_k^{K\times K\times K}. \tag{1}$$

$\mathcal{F}_k \in \mathbb{R}^{C_k}$ is a deep feature with channels C_k, $K = \frac{N}{2^{k-1}}$ is the grid size varying with scale, and R is the number of feature grids. The multi-feature layers are

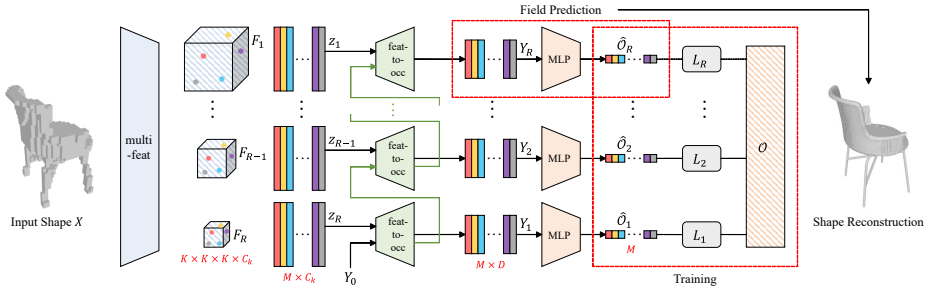

Fig. 1: Overview of DHR. Given an input shape as a coarse voxel grid, it is first encoded into multi-scale feature grids of varying scales. A set of hierarchical latent feature codes is acquired by grid-sampling a continuous query point on each feature grid. The feature-to-occupancy decoder takes learnable spatial positional embeddings as initial input and projects the latent codes onto occupancy tokens of respective 3D representations via cross-attention. To facilitate global-to-local learning, the latent codes are hierarchically incorporated in the transformer block. Surface reconstruction uses representations obtained from the final decoder output, whereas the optimization phase uses representations generated from all stages of latent coding to provide global-to-local learning.

composed of 3D CNNs with growing receptive fields and channels but shrinking resolution. Feature grids of early stages with small receptive fields include local details of the shape while grids of late stages with large receptive fields capture global structures [4, 38].

Given a continuous query point $q \in \mathbb{R}^3$ from a query set $Q \in \mathbb{R}^{M\times 3}$, a hierarchical latent feature code set is acquired by grid-sampling [13] the particular location on each feature grid as

$$\forall k \in [1, \ldots, R], \quad \mathrm{z}_k^q = \text{grid-sample}(F_k, q) \tag{2}$$

where $\mathrm{z}_k^q \in \mathbb{R}^{C_k}$ and thus $\mathrm{z}_k \in \mathbb{R}^{M\times C_k}$. Trilinear interpolation is used to align continuous 3D points on the discrete feature grids. Different query points and corresponding latent codes are marked with respective colors in Figure 1.

3.2 Feature-to-Occupancy Hierarchical Decoder

Drawing inspiration from recent breakthroughs in 3D reconstruction using transformers, a decoder based on transformer blocks is implemented to project shape features onto learnable spatial positional embeddings and translate them into occupancy representations. The adoption of a transformer architecture ensures scalability and supports category-agnostic training, thereby enhancing the model's generalizability to real-world objects.

To obtain the occupancy representation $\hat{\mathcal{O}}$, we first define the initial learnable spatial positional embeddings Y_0 which guide feature-to-3D projection and are used to query the features. Specifically, $Y_0 = [y_1; y_2; \cdots; y_M] \in \mathbb{R}^{M\times D}$ is

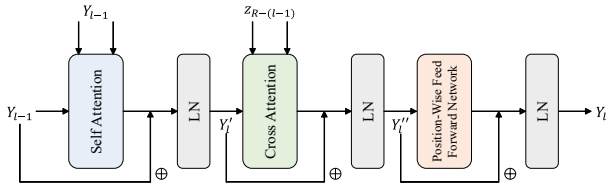

Fig. 2: Feature-to-occupancy decoder block. At the l-th step of the decoder, the $(R-(l-1))$-th latent feature code is fed as the key and value of the cross-attention layer. Every step of the process accumulates features from various levels to the decoder output in a hierarchical order.

used as the sequential 3D-volume queries at the input end of the decoder, where $y_m \in \mathbb{R}^{1\times D}$ corresponds to the m-th volume and D is the hidden dimension of the transformer decoder. Then to model the interaction of latent features with occupancies, feature codes from the 3D input shape are integrated into the transformer blocks via cross-attention. The decoder thus explores the connections between the feature and spatial domains and learns correlations of different spatial locations in attention layers.

However, in contrast to general transformers where the blocks are repeatably stacked with the inclusion of a single input, we allow the feature input to be updated at each stage; which we refer to as ***hierarchical inputs***. By providing the most recent iteration of the latent feature code set as an input for the cross-attention layer, the decoder begins to learn an approximated and rough shape. Subsequently, previous stages of the latent code set (achieved from feature grids with smaller receptive fields) are processed step by step in hierarchical order to progressively incorporate more delicate and tiny information. The final stack includes the first latent code as input and afterwards, decodes the final output including both the most local information and the global structures acquired from earlier stacks. Accordingly, the feature-to-occupancy hierarchical decoder can be formulated as

$$\forall l \in [1,\ldots,R], \quad Y_l = \text{feat-to-occ}(Y_{l-1}, \mathrm{z}_{R-(l-1)}) \tag{3}$$

and is visualized by the green blocks of Figure 1.

Transformer Block. Similar to other transformer architecture designs [14], each block of the transformer comprises a self-attention layer, a cross-attention layer, and a feed-forward layer. At a particular l-th step, suppose sequential volume Y_{l-1} is the input of a transformer block. Given that they correspond to the final occupancy features $\hat{\mathcal{O}}$, Y_{l-1} may be considered as the occupancy hidden features. As shown in Figure 2, the hidden features are passed to a self-attention layer that models the intra-modal relationships across the spatially structured occupancy entries. The cross-attention module then attends from the updated occupancy hidden features to the latent feature code $\mathrm{z}_{R-(l-1)}$, which can help link volume information to the occupancies. Then, a feed-forward layer (position-wise FFN) follows as in the original transformer design. The output occupancy features Y_l will become the input to the next transformer block.

Overall, the l-th block of the transformer can be demonstrated in steps as

$$Y_l' = \text{LN}(\text{self-attn}(Y_{l-1}) + Y_{l-1}), \tag{4}$$
$$Y_l'' = \text{LN}(\text{cross-attn}(Y_l', \text{z}_{R-(l-1)}) + Y_l'), \tag{5}$$
$$Y_l = \text{LN}(\text{pos-FFN}(Y_l'') + Y_l''). \tag{6}$$

3.3 Occupancy Field Prediction

The final decoder output Y_R is passed through an MLP containing multiple linear layers with ReLU activation to predict occupancy values:

$$\hat{\mathcal{O}_R} = \text{MLP}(Y_R). \tag{7}$$

The singular output $\hat{\mathcal{O}_R}$ is used as occupancy predictions of continuous point locations in shape reconstruction procedures. The field prediction process is visualized in the left dotted box of Figure 1.

3.4 Training

In the training phase, additional procedures are taken to expose all latent codes, including global and local features. This requires each of the occupancy hidden feature Ys to be passed through the MLP of Section 3.3 and compared with ground truth occupancy $\mathcal{O}$s by binary cross-entropy. For a particular l-th step, the loss function is formulated as below:

$$\forall l \in [1, \ldots, R], \quad \hat{\mathcal{O}}_l = \text{MLP}(Y_l), \tag{8}$$
$$\mathcal{L}_l(q) = \text{BCE}(\hat{\mathcal{O}}_l(q), \mathcal{O}(q)). \tag{9}$$

Since the later occupancy hidden features have accumulated more structural information than the early ones, thorough training is needed to ensure that they match the ground truth as closely as possible. Conversely, it is possible to train earlier hidden features more lightly.

Two types of ***hierarchical losses*** are implemented; the first simply regulates each loss's inclusion rate:

$$\mathcal{L}_l^{reg}(q) = \lambda_l \mathcal{L}_l(q). \tag{10}$$

The second type adopts an error tolerance rate ε that indicates the upper bound of permitted errors in predicted values:

$$\mathcal{L}_l^{tol}(q) = \max\{\mathcal{L}_l(q) - \varepsilon_l, 0\}. \tag{11}$$

To provide a global-to-local learning approach with varying loss contributions, hyperparameters λ and ε must meet the needs of $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_R$ and $\varepsilon_1 \geq \varepsilon_2 \geq \cdots \geq \varepsilon_R$.

To sum up, the parameters are optimized by minimizing the loss below:

$$\mathcal{L} = \mathbb{E}_{q \in Q}\left[\mathbb{E}_{l \in [1,\ldots,R]}\left[\mathcal{L}_l^{reg}(q) + \mathcal{L}_l^{tol}(q)\right]\right]. \tag{12}$$

The training process is also illustrated in the right dotted box of Figure 1.

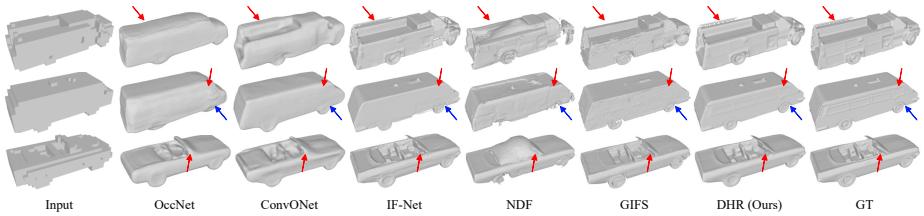

Fig. 3: Visual comparison of methods under ShapeNet. Shape reconstruction details can be discovered with closer zoom-in. Red and blue arrows point out local details that are improved by our method.

3.5 Surface Extraction

To extract the continuous surface of the required shape at the inference phase, $\hat{\mathcal{O}}_R(q)$ is used as the occupancy prediction of a continuous point location $q \in \mathbb{R}^3$. The hierarchical isosurface extraction algorithm [24] is additionally used to speed the extraction process. Starting from a coarse resolution, the evaluation of whether a point is occupied or unoccupied is repeated with subdivisions of voxels until the desired resolution is reached. The occupancy grid of high resolution is finally transformed to a smooth mesh by the traditional marching cubes [22] algorithm.

4 Experiments

Tasks. Shape reconstruction results from sparse voxels are qualitatively and quantitatively analyzed in Section 4.1. The computational costs are also presented alongside the baselines in this section. Section 4.2 validates the generalization ability by out of category reconstructions. We ablate the architecture choices and global-to-local learning components of the proposed framework in Section 4.3. As an auxiliary experiment to validate the representation power, point cloud completion results are compared with various methods in Section 4.4. To highlight subtleties and local areas that the proposed method has improved, the majority of the visualized comparison results are presented as shape categories with complicated structures.

Metrics. For quantitative evaluation, we use three metrics: intersection over union (IoU) to assess volume matching, F-score to evaluate prediction accuracy, and Chamfer distance to measure the average error between points. Higher IoU and F-score values indicate better performance, while lower Chamfer distance values reflect reduced error. Chamfer distances in the results are scaled by 10^2, with detailed metric formulations provided in the supplementary material.

Baselines. Experiment results of Section 4.1 - 4.3 are compared with several voxel reconstruction baselines - OccNet [24], ConvONet [28], IF-Net [4], NDF [5], and GIFS [38]. Two versions are implemented for ConvONet and IF-Net - discrete input voxel grid resolutions of 32 (indicated with *) and 128. Point cloud completion results of Section 4.4 are evaluated against OccNet [24],

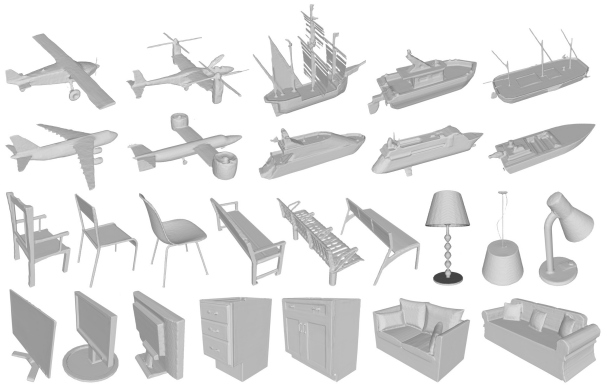

Fig. 4: Reconstruction of shapes from several categories. The capability of reconstructing various shapes with high quality by our method can be visually inspected.

ConvONet [28], IF-Net [4], SAP [27], POCO [1], DCC-DIF [20], NVF [37], and GridFormer [19]. All baselines were reimplemented with their original architectures and settings unchanged.

4.1 Shape Reconstruction from Sparse Voxels

Qualitative Analysis. A visualized comparison of shape reconstructions by our method against baselines is shown in Figure 3. OccNet and ConvONet reconstruct shapes in a very rough way, including flawed, crashed, or bumpy parts. For example, the reconstructions miss out structures on the ceiling and the ladder of the fire engine in the first row. IF-Net shows reconstructions in a fairly accurate way and captures subtle parts as well. However, compared to our method, it fails to contain patterns or thin parts, for example, bumper patterns of the fire engine or window frames of the bus in the second row. NDF fails to reconstruct several parts - for example, frontal structures of the fire engine, tires of the bus, or car seats of the open car in the third row. GIFS visualizes meshes with rough or bumpy parts, and some include holes. Local details such as ladder holes or bumper patterns of the fire engine are also lost. On the other hand, our method represents solid structures with the inclusion of complex and subtle parts. Notice that it captures the holes of the ladder, square patterns on the side, and bumper patterns of the fire engine. Visualizing thin window frames of the bus and car seat details of the open car is also achieved. Moreover, the main body shapes are very close to the ground truth shapes, without any flaws or missing parts.

Reconstructions from various categories of ShapeNet by our method are visualized in Figure 4. Regardless of category, it successfully builds every shape, with solid structures and the inclusion of local and subtle details - where we visually inspect the power and robustness.

Quantitative Evaluation. An assessment of reconstruction quality for our method against baselines by 13 shape categories is provided in Table 1a, 1b, and

Table 1: Reconstruction accuracy by three measures.

(a) Reconstruction accuracy under ShapeNet in terms of IoU (↑).

Category	IoU ↑							
	OccNet	ConvO.*	ConvO.	IF-Net*	IF-Net	NDF	GIFS	DHR (Ours)
Airplane	0.605	0.739	0.778	0.658	0.923	0.903	0.908	**0.951**
Bench	0.740	0.642	0.683	0.739	0.895	0.887	0.882	**0.917**
Cabinet	0.843	0.839	0.871	0.699	0.902	0.918	0.871	**0.920**
Car	0.702	0.813	0.851	0.764	0.913	0.874	0.923	**0.933**
Chair	0.763	0.737	0.787	0.779	0.884	0.881	0.852	**0.891**
Display	0.660	0.768	0.811	0.722	**0.947**	0.905	0.899	0.909
Lamp	0.547	0.647	0.705	0.602	0.851	0.873	0.931	**0.932**
Loudspeaker	0.802	0.832	0.879	0.671	0.857	0.869	0.862	**0.884**
Rifle	0.698	0.661	0.707	0.750	0.902	0.860	0.903	**0.928**
Sofa	0.682	0.832	0.875	0.723	0.865	0.872	0.915	**0.917**
Table	0.812	0.674	0.725	0.874	**0.918**	0.850	0.841	0.890
Telephone	0.747	0.821	0.861	0.802	0.868	0.921	0.870	**0.960**
Vessel	0.590	0.734	0.777	0.629	0.871	0.843	0.904	**0.924**
Mean	0.707	0.750	0.794	0.724	0.892	0.881	0.889	**0.919**
Std	0.086	0.072	0.068	0.071	0.027	0.024	0.027	**0.021**

(b) Reconstruction accuracy under ShapeNet in terms of F-Score (↑).

Category	F-Score ↑							
	OccNet	ConvO.*	ConvO.	IF-Net*	IF-Net	NDF	GIFS	DHR (Ours)
Airplane	0.667	0.849	0.926	0.837	0.946	0.864	0.953	**0.990**
Bench	0.713	0.751	0.799	0.896	0.940	0.881	0.931	**0.974**
Cabinet	0.754	0.664	0.688	0.908	**0.974**	0.890	0.928	0.961
Car	0.608	0.708	0.766	0.853	0.908	0.857	0.972	**0.991**
Chair	0.801	0.710	0.790	0.942	0.897	0.908	0.920	**0.965**
Display	0.794	0.667	0.727	0.880	**0.981**	0.926	0.890	0.966
Lamp	0.597	0.703	0.802	0.856	0.854	0.849	**0.987**	0.977
Loudspeaker	0.649	0.611	0.690	0.791	0.945	0.900	0.893	**0.959**
Rifle	0.713	0.798	0.876	0.898	0.887	0.819	0.916	**0.947**
Sofa	0.699	0.689	0.759	0.925	0.978	0.939	0.949	**0.985**
Table	0.610	0.690	0.745	0.877	0.948	0.822	0.874	**0.957**
Telephone	0.782	0.765	0.830	0.819	0.916	0.894	0.939	**0.970**
Vessel	0.570	0.700	0.770	0.789	0.950	0.869	0.957	**0.994**
Mean	0.689	0.716	0.783	0.867	0.932	0.878	0.931	**0.972**
Std	0.076	0.060	0.065	0.046	0.036	0.035	0.031	**0.014**

(c) Reconstruction accuracy under ShapeNet in terms of Chamfer distance (↓). Chamfer distance results $\times 10^{-2}$.

Category	Chamfer distance ↓							
	OccNet	ConvO.*	ConvO.	IF-Net*	IF-Net	NDF	GIFS	DHR (Ours)
Airplane	1.498	0.567	0.466	0.213	0.091	0.158	0.195	**0.087**
Bench	1.097	0.734	0.594	0.199	0.117	0.116	0.162	**0.107**
Cabinet	1.989	1.002	0.885	0.200	0.132	0.180	0.117	**0.099**
Car	1.671	1.111	0.833	0.197	0.103	0.215	0.094	**0.073**
Chair	0.974	0.848	0.637	0.249	**0.068**	0.097	0.126	0.096
Display	1.389	0.905	0.714	0.204	0.109	0.136	0.138	**0.078**
Lamp	1.530	0.979	0.734	0.264	0.134	0.159	**0.085**	0.093
Loudspeaker	1.973	1.199	0.903	0.354	0.126	0.129	0.170	**0.116**
Rifle	0.950	0.650	0.480	0.266	0.090	0.185	0.103	**0.085**
Sofa	1.858	0.881	0.685	0.271	0.123	0.114	0.098	**0.091**
Table	1.683	0.877	0.689	0.301	0.096	0.162	0.162	**0.089**
Telephone	1.079	0.720	0.524	0.184	0.120	0.144	0.129	**0.103**
Vessel	1.296	0.867	0.671	0.219	0.099	0.098	0.105	**0.094**
Mean	1.461	0.873	0.679	0.240	0.108	0.146	0.129	**0.093**
Std	0.352	0.171	0.136	0.047	0.018	0.034	0.032	**0.011**

Table 2: Computational complexity. Our approach delivers inference times that are on par with, or even faster than, the baseline methods.

Method	OccNet	ConvONet	IF-Net	NDF	GIFS	DHR (Ours)
Inference Time (s)	8.732	17.907	85.831	142.529	288.282	10.932

1c (in terms of IoU, F-Score, and Chamfer distance, respectively). The mean and std. computed over all categories are indicated below the category measures. By the best results in bold, it is inspected that the proposed method outperforms baselines for almost every measure. Furthermore, while other methods reveal large variances between categories, our method shows rigidity by similar metric values and small variances.

Computational Complexity. We compare the inference time for generating 10 meshes between our method and several baseline approaches, as shown in Table 2. Our approach not only outperforms the baseline methods in terms of performance metrics but also matches or exceeds their inference speeds. This demonstrates both the accuracy and efficiency of our method, highlighting its effectiveness in shape reconstruction from sparse voxels.

4.2 Out of Category Generalization

We evaluate the generalization capability of our method by training on 6 categories (loudspeaker, rifle, sofa, table, telephone, and vessel) from ShapeNet and testing on 7 unseen categories (airplane, bench, cabinet, car, chair, display, and lamp). The results, shown in Table 3, demonstrate that our method outperforms all baseline approaches, with only a minimal performance drop when compared to models trained on all 13 categories. This small decline, highlighted by red numbers in brackets, showcases the method's robustness in handling new, unseen data. Visual inspection of the reconstructions, as shown in Figure 5, further supports our approach's superior performance. These findings emphasize the method's strong generalization and applicability in real-world scenarios with diverse categories.

4.3 Ablation Study

Number of Latent Feature Codes. The performances of the proposed method including different numbers of latent feature codes are evaluated in Table 4 by measures. Using a number of 7 codes achieves the best measures compared to 5 and 6 codes. Visualized reconstructions by each number of latent codes are provided in Figure 6, where local details are improved in the higher number of codes.

Decoder Architecture. Our method with different architecture choices of the decoder is visualized in Figure 7 and reported in Table 5. The proposed hierarchical transformer decoder is replaced with either a CNN or a general transformer decoder to justify the efficacy of the decoder architecture. Figure 7 shows that

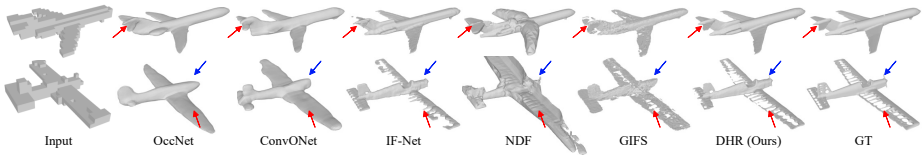

Fig. 5: Generalization capacity in unseen categories. Reconstruction of shapes from categories unseen during training are visualized. The robustness of our method is visually inspected, which is a distinct difference against baselines.

Table 3: Out of category generalization. Red numbers in the brackets denote the drop in performance compared to the models trained on all categories. Our method shows a minimal drop by all measures, thus validating generalization capacity.

Method	IoU ↑	F-Score ↑	Chamfer ↓
OccNet [24]	0.564 (-14.3%)	0.598 (-9.1%)	1.951 (0.490)
ConvONet [28]	0.713 (-8.1%)	0.710 (-7.3%)	0.768 (0.089)
IF-Net [4]	0.860 (-3.2%)	0.884 (-4.8%)	0.135 (0.027)
NDF [5]	0.824 (-5.7%)	0.816 (-6.2%)	0.191 (0.045)
GIFS [38]	0.831 (-5.8%)	0.872 (-5.9%)	0.170 (0.041)
DHR (Ours)	**0.911 (-0.8%)**	**0.961 (-1.1%)**	**0.101 (0.008)**

the hierarchical transformer enables to capture of local details, and all measures in Table 5 present that reconstruction performances are better achieved without replacement.

Hierarchical Inputs and Losses. We ablate the components that contribute to the global-to-local learning strategy in three ways: (1) hierarchical inputs, which are latent feature code inputs included in order to the feature-to-occupancy decoder, (2) hierarchical losses, which are losses from all decoder outputs, and (3) the different degrees of contribution of the hierarchical losses. For models without hierarchical inputs, the lowest-level latent code is fed into the transformer decoder and stacked with the same number of steps as the full model. For models without hierarchical losses, the singular computed output (also used in the field prediction, Section 3.3) is compared with the ground truth value to optimize the model. The last model is trained with equal contribution of all hierarchical losses, thus $\lambda_1 = \lambda_2 = \cdots = \lambda_R$ and $\varepsilon_1 = \varepsilon_2 = \cdots = \varepsilon_R$.

As reported in Table 6, it is observed that all components play an important role in boosting the performance of reconstruction, thus acquiring better representations.

4.4 Point Cloud Completion

Although our method's primary strength lies in its ability to learn implicit neural fields and produce promising reconstruction outcomes from discrete voxel grids, in this subsection we test the representation power with further experiments on sparse point cloud completion. Unlike voxel grids, point clouds cannot be directly fed into 3D CNN layers. Instead, during the encoding stage, they are processed

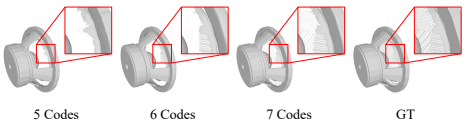

Fig. 6: Different choices of the number of latent codes. Fine details are better captured as the number of latent codes is progressed from 5 to 7.

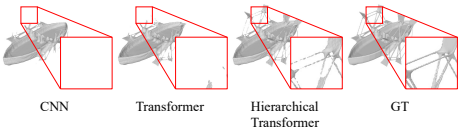

Fig. 7: Different decoder architecture choices. Using the proposed hierarchical transformer as a decoder block captures shape details better than using either a CNN or a transformer.

Table 4: Number of latent codes. Using a number of 7 latent codes shows the best performance by all measures, compared to 5 or 6 codes.

Codes (#)	IoU ↑	F-Score ↑	Chamfer ↓
5	0.893	0.906	0.149
6	0.901	0.928	0.110
7	**0.919**	**0.972**	**0.093**

Table 5: Decoder architecture. Using the proposed hierarchical transformer as a decoder block shows the best reconstruction performance by three measures.

Decoder Architecture	IoU ↑	F-Score ↑	Chamfer ↓
CNN	0.873	0.908	0.154
Transformer	0.881	0.939	0.137
Hierarchical Transformer	**0.919**	**0.972**	**0.093**

Table 6: Components of global-to-local learning strategy. Including both hierarchical inputs and hierarchical losses boosts the reconstruction performance. Different degrees of contribution to the losses also encourage higher quality.

Methods	IoU ↑	F-Score ↑	Chamfer ↓
w/o Hierarchical inputs	0.859	0.901	0.128
w/o Hierarchical losses	0.898	0.952	0.103
Equal contribution	0.907	0.964	0.101
Full model	**0.919**	**0.972**	**0.093**

using mini-PointNet-style modules [29, 39] to generate multi-scale feature grids, replacing the use of 3D CNNs with varying receptive fields. Further details and formulations on acquiring multi-scale feature grids and embedding query points are in the supplementary material.

Qualitative Analysis. Completion from sparse point clouds is visually compared against baseline methods in Figure 8. Similar to voxel reconstructions, OccNet and ConvONet reconstruct shapes in a rough way and fail to reconstruct structural details. For example, boundaries between substructures are missing and small parts are merged into large chunks. IF-Net visualizes meshes that are bumpy and some parts are removed, for example, the frontal part of the open car in the second row. SAP and POCO are better at preserving details - however, specific structures such as tires or side mirrors are merged. DCC-DIF reconstructs shapes quite accurately and includes small structures. However when compared to our method, thin or subtle parts are still missing or visualized as wrong shapes, for example, side mirrors or patterns of the side of the sports car in the third row. Our method, on the other hand, shows faithful reconstruction

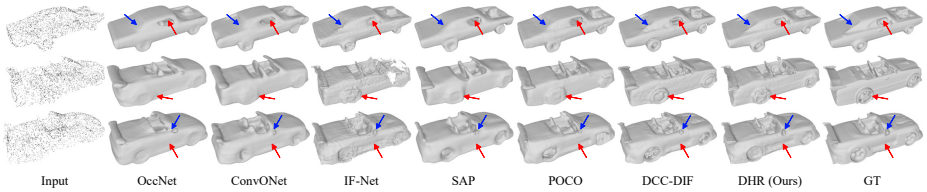

Fig. 8: Visual comparison of methods by sparse point cloud completion. Shape reconstruction details can be discovered with closer zoom-in. Red and blue arrows point out local details that are improved by our method.

Table 7: Assessment of methods by point cloud completion. Measure means are computed over 13 ShapeNet categories. Chamfer distance results $\times 10^{-2}$.

Method	IoU ↑	F-Score ↑	Chamfer ↓
OccNet [24]	0.777	0.819	0.794
ConvONet [28]	0.870	0.933	0.479
IF-Net [4]	0.874	0.927	0.211
SAP [27]	0.887	0.961	0.415
POCO [1]	0.896	0.960	0.126
DCC-DIF [20]	0.917	0.970	0.105
NVF [37]	0.918	0.957	0.081
GridFormer [19]	0.922	0.968	0.090
DHR (Ours)	**0.931**	**0.979**	**0.070**

results by including small structures, local details, and patterns, especially those that are pointed out by red and blue arrows on the figure.

Quantitative Evaluation. An assessment of point cloud completion for our method against baselines is tabulated in Table 7. IoU, F-Score, and Chamfer distance means are computed over 13 ShapeNet categories and the best results are in bold. Our method outperforms all baselines by three measures, where we inspect the representation power. Specific measure values for all 13 shape categories are provided in the supplementary material.

5 Conclusion

This work presents Dual Hierarchical Representation (DHR) for 3D surface reconstruction from sparse voxel grids. DHR uses a transformer-based decoder to model interactions between latent features and occupancies. It introduces hierarchical inputs and losses to enable global-to-local learning, progressively refining geometry. Experiments show DHR reconstructs high-quality surfaces and generalizes well across categories and tasks.

Acknowledgements. This work is in part supported by the National Research Foundation of Korea (NRF, RS-2024-00451435(20%), RS-2024-00413957(20%)), Institute of Information & communications Technology Planning & Evaluation (IITP, 2021-0-01059(20%), 2021-0-00106(20%), 2021-0-00180(20%)) grant funded by the Ministry of Science and ICT (MSIT), Institute of New Media and Communications(INMAC), BK21 FOUR program of the Education and Research Program for Future ICT Pioneers, Seoul National University in 2024, and Ascending SNU Future Leader Fellowship through Seoul National University.

References

1. Boulch, A., Marlet, R.: POCO: point convolution for surface reconstruction. In: Proc. Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 6292–6304 (2022)
2. Chabra, R., Lenssen, J.E., Ilg, E., Schmidt, T., Straub, J., Lovegrove, S., Newcombe, R.: Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In: Proc. European Conf. on Computer Vision (ECCV). pp. 608–625 (2020)
3. Chen, Z., Zhang, H.: Learning implicit fields for generative shape modeling. In: Proc. Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 5939–5948 (2019)
4. Chibane, J., Alldieck, T., Pons-Moll, G.: Implicit functions in feature space for 3d shape reconstruction and completion. In: Proc. Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 6968–6979 (2020)
5. Chibane, J., Mir, A., Pons-Moll, G.: Neural unsigned distance fields for implicit function learning. In: Proc. Advances in Neural Information Processing Systems (NeurIPS). pp. 21638–21652 (2020)
6. Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In: Proc. European Conf. on Computer Vision (ECCV). pp. 628–644 (2016)
7. Deng, Y., Yang, J., Tong, X.: Deformed implicit field: Modeling 3d shapes with learned dense correspondence. In: Proc. Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 10286–10296 (2021)
8. Duan, Y., Zhu, H., Wang, H., Yi, L., Nevatia, R., Guibas, L.J.: Curriculum deepsdf. In: Proc. European Conf. on Computer Vision (ECCV). pp. 51–67 (2020)
9. Erler, P., Guerrero, P., Ohrhallinger, S., Mitra, N.J., Wimmer, M.: Points2surf learning implicit surfaces from point clouds. In: Proc. European Conf. on Computer Vision (ECCV). pp. 108–124 (2020)
10. Fan, H., Yu, X., Ding, Y., Yang, Y., Kankanhalli, M.: Pstnet: Point spatio-temporal convolution on point cloud sequences. In: Proc. Int. Conf. on Learning Representations (ICLR) (2021)
11. Genova, K., Cole, F., Sud, A., Sarna, A., Funkhouser, T.: Local deep implicit functions for 3d shape. In: Proc. Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 4857–4866 (2020)
12. Hong, Y., Zhang, K., Gu, J., Bi, S., Zhou, Y., Liu, D., Liu, F., Sunkavalli, K., Bui, T., Tan, H.: Lrm: Large reconstruction model for single image to 3d. arXiv preprint arXiv:2311.04400 (2023)
13. Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial transformer networks. In: Proc. Advances in Neural Information Processing Systems (NeurIPS). pp. 2017–2025 (2015)
14. Jaegle, A., Gimeno, F., Brock, A., Vinyals, O., Zisserman, A., Carreira, J.: Perceiver: General perception with iterative attention. In: Proc. Int. Conf. on Machine Learning (ICML). pp. 4651–4664 (2021)
15. Jiang, C., Sud, A., Makadia, A., Huang, J., Nießner, M., Funkhouser, T., et al.: Local implicit grid representations for 3d scenes. In: Proc. Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 6001–6010 (2020)
16. Kanazawa, A., Tulsiani, S., Efros, A.A., Malik, J.: Learning category-specific mesh reconstruction from image collections. In: Proc. European Conf. on Computer Vision (ECCV). pp. 371–386 (2018)

17. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. on Graphics (TOG) (2023)
18. Kolotouros, N., Pavlakos, G., Daniilidis, K.: Convolutional mesh regression for single-image human shape reconstruction. In: Proc. Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 4501–4510 (2019)
19. Li, S., Gao, G., Liu, Y., Liu, Y., Gu, M.: Gridformer: Point-grid transformer for surface reconstruction. In: Proc. AAAI Conf. on Artificial Intelligence (AAAI). pp. 3163–3171 (2024)
20. Li, T., Wen, X., Liu, Y., Su, H., Han, Z.: Learning deep implicit functions for 3d shapes with dynamic code clouds. In: Proc. Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 12840–12850 (2022)
21. Liao, Y., Donne, S., Geiger, A.: Deep marching cubes: Learning explicit surface representations. In: Proc. Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 2916–2925 (2018)
22. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. In: Proc. Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH). pp. 163–169 (1987)
23. Maturana, D., Scherer, S.: Voxnet: A 3d convolutional neural network for real-time object recognition. In: Int. Conf. on Intelligent Robots and Systems (IROS). pp. 922–928 (2015)
24. Mescheder, L.M., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: Proc. Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 4460–4470 (2019)
25. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: Proc. European Conf. on Computer Vision (ECCV). pp. 405–421 (2020)
26. Park, J., Florence, P., Straub, J., Newcombe, R.A., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: Proc. Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 165–174 (2019)
27. Peng, S., Jiang, C., Liao, Y., Niemeyer, M., Pollefeys, M., Geiger, A.: Shape as points: A differentiable poisson solver. In: Proc. Advances in Neural Information Processing Systems (NeurIPS). pp. 13032–13044 (2021)
28. Peng, S., Niemeyer, M., Mescheder, L.M., Pollefeys, M., Geiger, A.: Convolutional occupancy networks. In: Proc. European Conf. on Computer Vision (ECCV). pp. 523–540 (2020)
29. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proc. Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 652–660 (2017)
30. Shin, J., Hong, S., Lee, J.: Nerflex: Flexible neural radiance fields with diffeomorphic deformation. IEEE Access (2024)
31. Sun, X., Wu, J., Zhang, X., Zhang, Z., Zhang, C., Xue, T., Tenenbaum, J.B., Freeman, W.T.: Pix3d: Dataset and methods for single-image 3d shape modeling. In: Proc. Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 2974–2983 (2018)
32. Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., Jiang, Y.: Pixel2mesh: Generating 3d mesh models from single rgb images. In: Proc. European Conf. on Computer Vision (ECCV). pp. 52–67 (2018)
33. Williams, F., Gojcic, Z., Khamis, S., Zorin, D., Bruna, J., Fidler, S., Litany, O.: Neural fields as learnable kernels for 3d reconstruction. In: Proc. Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 18500–18510 (2022)

34. Wu, C., Johnson, J., Malik, J., Feichtenhofer, C., Gkioxari, G.: Multiview compressive coding for 3d reconstruction. In: Proc. Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 9065–9075 (2023)
35. Wu, W., Qi, Z., Fuxin, L.: Pointconv: Deep convolutional networks on 3d point clouds. In: Proc. Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 9621–9630 (2019)
36. Yang, J., Wickramasinghe, U., Ni, B., Fua, P.: Implicitatlas: Learning deformable shape templates in medical imaging. In: Proc. Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 15861–15871 (2022)
37. Yang, X., Lin, G., Chen, Z., Zhou, L.: Neural vector fields: Implicit representation by explicit learning. In: Proc. Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 16727–16738 (2023)
38. Ye, J., Chen, Y., Wang, N., Wang, X.: Gifs: Neural implicit function for general shape representation. In: Proc. Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 12829–12839 (2022)
39. Zhang, B., Nießner, M., Wonka, P.: 3dilg: Irregular latent grids for 3d generative modeling. Proc. Advances in Neural Information Processing Systems (NeurIPS) pp. 21871–21885 (2022)
40. Zhang, B., Tang, J., Niessner, M., Wonka, P.: 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. ACM Trans. on Graphics (TOG) pp. 1–16 (2023)
41. Zheng, Z., Yu, T., Dai, Q., Liu, Y.: Deep implicit templates for 3d shape representation. In: Proc. Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 1429–1439 (2021)
42. Zou, Z., Yu, Z., Guo, Y., Li, Y., Liang, D., Cao, Y., Zhang, S.: Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers. Proc. Conf. on Computer Vision and Pattern Recognition (CVPR) pp. 10324–10335 (2024)