Attention-Guided Multi-scale Neural Dual Contouring

Fuli Wu[1], Chaoran Hu[1], Wenxuan Li[1] and Pengyi Hao[†1]

[1]School of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China

Abstract

Reconstructing high-quality meshes from binary voxel data is a fundamental task in computer graphics. However, existing methods struggle with low information density and strong discreteness, making it difficult to capture complex geometry and long-range boundary features, often leading to jagged surfaces and loss of sharp details. We propose an Attention-Guided Multi-scale Neural Dual Contouring (AGNDC) method to address this challenge. AGNDC refines surface reconstruction through a multi-scale framework, using a hybrid feature extractor that combines global attention and dynamic snake convolution to enhance perception of long-range and high-curvature features. A dynamic feature fusion module aligns multi-scale predictions to improve local detail continuity, while a geometric postprocessing module further refines mesh boundaries and suppresses artifacts. Experiments on the ABC dataset demonstrate the superior performance of AGNDC in both visual and quantitative metrics. It achieves a Chamfer Distance ($CD\times10^5$) of 9.013 and an F-score of 0.440, significantly reducing jaggedness and improving surface smoothness.

CCS Concepts

• ***Computing methodologies*** → ***Mesh generation; Neural networks; Volumetric models;***

1. Introduction

Binary voxel data is a widely used form of 3D representation, favored in many scenarios for its compact storage and computational efficiency [CZ19]. For example, in medical image segmentation, binary voxel data is often generated to represent regions or structures of interest [SFD*23]; in virtual reality and game development, it is also commonly used to construct 3D environments and objects [BWT24]. This data format not only intuitively indicates the presence of target regions but also provides a concise and structured input for subsequent processes such as shape analysis and model generation [SGEB00]. Converting binary voxel data into high-quality surface meshes containing only boundary information enables clearer geometric presentation and provides effective visualization support in fields such as medical analysis, virtual reality, and augmented reality [CHY*24].However, since binary voxel data is encoded as sparse 3D arrays using 0 and 1 to denote the presence of matter, the sparsity of information limits the feature extraction capacity of deep learning models—especially in the presence of complex topology or sharp features [KYZB19]. Additionally, the inherent discreteness of voxel data often results in aliasing artifacts, making it difficult to reconstruct smooth curves or surfaces. Therefore, designing effective feature extraction and surface mesh reconstruction methods for binary voxel data remains a key challenge in this domain.

Current mainstream solutions are mostly model-driven and typically rely on assumptions about shape properties (e.g., watertightness), surface interpolation (e.g., trilinear), sampling conditions, or prior information such as surface normals [MKFW12]. While these methods can achieve relatively good reconstruction results under specific assumptions, their strong dependence on priors significantly limits their performance in handling complex geometries and topologies [WBZ*20]. In recent years, a growing number of deep learning-based mesh reconstruction approaches have been developed for binary voxel data. These methods adopt end-to-end learning frameworks and are capable of recovering certain geometric details and topological structures.

However, due to the discreteness and low information density of voxel inputs, these methods still face major challenges in high-precision surface reconstruction. For example, Deep Marching Cubes (DMC) [LDG18] and Neural Marching Cubes (NMC) [CZ21] extend the classic Marching Cubes (MC) [LC98] algorithm by incorporating learned local shape priors and adaptive template selection, enabling more flexible reconstruction. Nevertheless, like the original MC algorithm and its popular variant MC33, these template-based methods often generate a large number of triangles to accurately describe 3D shapes, leading to increased computational cost and poor retention of sharp features [NY06]. In contrast, Neural Dual Contouring (NDC) [CTFZ22] eliminates template dependence by directly predicting vertex positions, achieving better detail preservation. However, its feature extraction relies on fixed-receptive-field 3D con-

† Corresponding author: haopy@zjut.edu.cn

DOI: 10.2312/pg.20251291

volutions [LLY*21], which lack the capacity to capture long-range geometric features. Additionally, many existing methods struggle to extract both high-quality local and long-range information from binary voxel data, resulting in jagged edges and unsmooth surfaces in complex reconstructions.

To further improve reconstruction quality, it is crucial to develop a feature extraction mechanism capable of perceiving both global structures and fine-grained geometry. Traditional convolution operations are limited by their fixed receptive fields and cannot effectively handle high-curvature areas or long-range dependencies, resulting in insufficient responses around sharp boundaries. To address this, we propose an Attention-Guided Multi-scale Neural Dual Contouring (AGNDC) method, which integrates global structural perception with local detail modeling to improve adaptability to complex surface geometries. In the feature extraction stage, AGNDC introduces a hybrid perception mechanism that combines self-attention with dynamic snake convolution [QHQ*23]. The attention module captures long-range dependencies, while the dynamic convolution adaptively unfolds based on geometric gradients, enabling effective perception of sharp edges and high-curvature regions, thereby enhancing the geometric expressiveness of the network.

In addition, AGNDC incorporates a dynamic feature fusion module that aligns voxel-level predictions and geometric features across different scales throughout the multi-scale reconstruction process. This module leverages attention mechanisms to guide the direction and strength of feature fusion, achieving efficient information integration and enhancing cross-scale continuity. Furthermore, to improve the final surface reconstruction quality, AGNDC introduces a geometric postprocessing module in the output stage, which employs a lightweight convolutional structure to refine boundary position predictions and suppress artifacts and aliasing effects. This results in reconstructions with structural continuity and sharp boundaries.

Experimental results show that AGNDC achieves outstanding 3D reconstruction performance across multiple public datasets, demonstrating significant advantages in surface smoothness, geometric accuracy, and detail preservation. The code is publicly available at https://github.com/chaoran4532/AGNDC.

2. Related Work

2.1. Traditional Surface Reconstruction Methods

Surface reconstruction techniques aim to convert discrete voxel data into explicit triangular meshes, thereby enabling efficient representation of object geometry and avoiding redundant data interference. For binary voxel data, traditional algorithms must reconstruct surface geometry through interpolation or fitting under conditions of sparsity and discreteness.

Marching Cubes (MC) [LC98] is a classical isosurface extraction algorithm and one of the most widely used methods for surface reconstruction from binary voxel data. MC divides the 3D array into regularly arranged cells and determines whether a surface intersects a cell based on the binary status (0 or 1) of its eight corner vertices. By matching one of 15 predefined surface templates, MC linearly interpolates surface positions to generate explicit triangular mesh representations. However, since MC relies solely on binary vertex states, it performs poorly in representing sharp features and capturing complex topological structures, limiting its effectiveness in high-precision applications.

Dual Contouring (DC) [JLSW02] improves upon MC by introducing local normal information to enhance sharp feature representation. Like MC, DC also partitions the 3D space into grid cells, but instead of interpolating vertex values, it fits a vertex within each cell using normal vectors at voxel boundaries and trilinear interpolation. These vertices are then optimized by minimizing a Quadratic Error Function (QEF) [GH97], resulting in smoother surfaces with clearer geometric details. Nevertheless, DC's performance is highly dependent on the accurate estimation of high-quality normals, which is challenging to obtain from sparse binary voxel data, limiting its broad applicability.

Occupancy-Based Dual Contouring (ODC) [HS24] improves DC's computational process for binary voxel data by introducing 2D auxiliary points to enhance the precision of QEF-based vertex fitting. However, ODC does not utilize deep learning to optimize vertex prediction and still relies on limited receptive fields confined to adjacent cells, which affects the quality and generality of reconstructed surfaces.

Traditional surface reconstruction methods play an important role in handling binary voxel data and offer practical approaches for low-complexity mesh generation and high-fidelity geometric fitting. However, their performance is often constrained by data sparsity, voxel discreteness, and limited receptive fields. These limitations make it difficult to capture complex topologies or detail-rich surface geometries effectively in high-precision scenarios. To overcome these issues, learning-based surface reconstruction methods have become a growing research focus, demonstrating potential for enhancing geometric detail and overall reconstruction performance.

2.2. Learning-Based Surface Reconstruction Methods

With the advancement of deep learning, neural-network-based surface reconstruction methods have gradually become mainstream. DMC [LDG18] enhances the classic MC [LC98] algorithm by directly extracting finer geometric features from voxel data. Its end-to-end architecture improves the expressiveness of MC, but since DMC relies on global feature encoding, its ability to capture fine details is limited, especially when dealing with complex geometries. Furthermore, its performance is highly sensitive to the information density of voxel inputs, making it difficult to handle detail-intensive recon-struction tasks, particularly for sharp edges and intricate topologies.

DefTet [GCX*20] employs a tetrahedral mesh-based reconstruction framework suited for generating low-resolution initial models and supporting shape deformation and to-pology optimization. However, its capacity to preserve complex geometric details and sharp features in high-precision reconstruction remains insufficient, especially at object boundaries.

NMC [CZ21] further improves the MC template system to better

represent sharp edges. However, its reliance on fixed resolution and increased computational complexity restricts its adaptability to local geometric variation. FlexiCubes [SMH*23] introduces vertex-level flexibility to reduce boundary artifacts, but relies on continuous input fields and richer features to support the increased degrees of freedom. SurfR [RPW*25] adopts a hierarchical implicit framework with multi-scale attention, yet lacks an official implementation. Both methods assume continuous inputs and dense supervision, limiting their applicability to binary voxel data with low information density and quantization noise.

Neural Dual Contouring (NDC) [CTFZ22] , based on the DC [JLSW02] algorithm, predicts vertex positions and occupancy information via neural networks, thereby avoiding template limitations and gaining advantages in sharp feature reconstruction [SW02]. However, due to the sparse nature of binary voxel data, NDC still struggles to recover complex geometric details, particularly in densely detailed regions. HRE-NDC [LXL*23] incorporates sparse convolutional networks to enhance high-resolution reconstruction but depends heavily on low-resolution initial features in its multi-resolution fusion process, failing to fully leverage high-resolution corrections. This results in instability in reconstructed detail quality.

In summary, current learning-based surface reconstruction methods still face significant limitations in high-precision reconstruction and detailed feature representation when processing binary voxel data. In particular, the reconstruction of sharp boundaries and the capture of complex geometry remain challenging, necessitating more effective feature extraction mechanisms to further enhance reconstruction performance. It is important to clarify that this reconstructive task is fundamentally distinct from the goals of generative models, such as diffusion-based [GSW*22] approaches for shape synthesis, or novel view synthesis methods like Neural Radiance Fields [MST*21] or 3D Gaussian Splatting [KKLD23] , which focus on creating novel content rather than recovering pre-existing geometry.

3. Method

This paper proposes an Attention-Guided Multi-scale Neural Dual Contouring (AGNDC) method for surface reconstruction. As illustrated in Fig. 1, the proposed approach adopts a multi-scale reconstruction strategy aimed at achieving high-precision surface reconstruction through multi-stage long-range feature extraction and fusion.

3.1. Network Architecture

The goal of surface reconstruction is to convert a binary voxel volume $I \in \mathrm{R}^{M\times N\times K}$ into a predicted triangular mesh $Q = (V,A)$, where V denotes the set of mesh vertices and A denotes the set of triangular faces formed by these vertices.We map the ground-truth mesh $\widetilde{Q}$ onto a 3D grid of size $M \times N \times K$. For each cell intersected by the mesh surface, a vertex coordinate is sampled as a ground-truth vertex $T_v \in \mathrm{R}^{M\times N\times K\times 3}$ at this resolution, while cells not intersected by the mesh are marked as negative. Similarly, the edge occupancy status is recorded to indicate whether a triangle passes through a cell edge, forming a binary edge label $T_\varepsilon \in \mathrm{R}^{M\times N\times K\times 3}$.

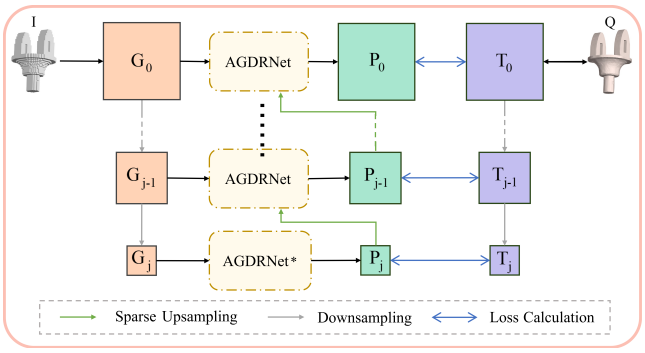

Figure 1: *Multi-scale framework of AGNDC.*

The complete ground-truth tensor at resolution $M \times N \times K$ is thus denoted as $T_0 = (T_v, T_\varepsilon)$, $T_0 \in \mathrm{R}^{M\times N\times K\times 6}$.Our objective is to train a network to predict the vertex positions $P_v \in \mathrm{R}^{M\times N\times K\times 3}$ and edge occupancies $P_\varepsilon \in \mathbb{R}^{M\times N\times K\times 3}$ from the binary voxel input $G_0 = I$, $G_0 \in \mathbb{R}^{M\times N\times K}$, thereby obtaining the reconstructed mesh $P_0 = (P_v, P_\varepsilon)$, $P_0 \in \mathbb{R}^{M\times N\times K\times 6}$.The prediction process can be formulated as:

$$P_0 = AGNDC(G_0) \tag{1}$$

AGNDC adopts a multi-scale architecture. We assume that each downsampling operation reduces the spatial size of the voxel array by half, while each upsampling operation doubles it. Let $s \in [0, j]$ denote the set of network scales, where j is the maximum number of downsampling steps. Starting from the original binary voxel input I, we perform downsampling j times to obtain a set of binary voxel volumes at each scale, denoted as $G = \{G_0, \cdots, G_{j-1}, G_j\}$. Similarly, the corresponding ground-truth values T_0 are downsampled j times to produce a set of multi-scale ground-truth labels $T = \{T_0, \cdots, T_{j-1}, T_j\}$. The predicted mesh results at each corresponding scale are denoted as $P = \{P_0, \cdots, P_{j-1}, P_j\}$.

At each scale, we independently train an Attention-Guided Dynamic Reconstruction Network (AGDRNet). Since the lowest scale has no input from any previous scale, we denote the special AGDRNet at the lowest scale as AGDRNet*. Starting from the lowest scale with the maximum downsampling level $s = j$, we input the corresponding binary voxel data $G_j \in \mathbb{R}^{\frac{M}{2^j}\times\frac{N}{2^j}\times\frac{K}{2^j}}$ into AGDRNet*, and obtain the predicted mesh result $P_j \in \mathbb{R}^{\frac{M}{2^j}\times\frac{N}{2^j}\times\frac{K}{2^j}\times 6}$, which can be expressed as:

$$P_j = AGDRNet^*(G_j) \tag{2}$$

For each higher scale $s = 0, \cdots, j-1$, we input the binary voxel data $G_s \in \mathbb{R}^{\frac{M}{2^s}\times\frac{N}{2^s}\times\frac{K}{2^s}}$ together with the predicted mesh result $P_{s+1} \in \mathbb{R}^{\frac{M}{2^{s+1}}\times\frac{N}{2^{s+1}}\times\frac{K}{2^{s+1}}\times 6}$ from the adjacent lower scale into AGDRNet to obtain the predicted mesh at that scale, denoted as:

$$P_s = AGDRNet(G_s, P_{s+1}), s \in \{0, \cdots, j-1\} \tag{3}$$

Therefore, the final predicted mesh result at the original resolu-

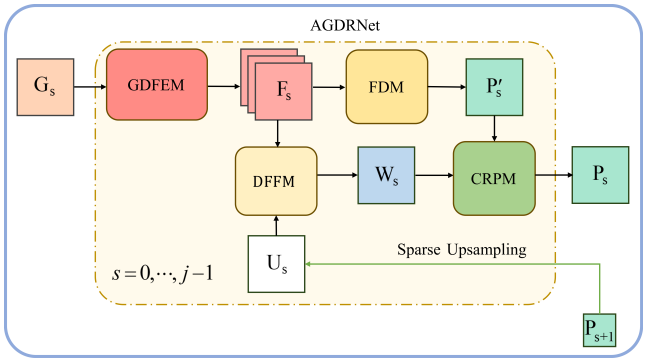

Figure 2: *Framework of the Attention-Guided Dynamic Reconstruction Network (AGDRNet).*

tion is:

$$\begin{aligned} P_0 &= \text{AGDRNet}(G_0, P_1) \\ &= \text{AGDRNet}\left(G_0, \text{AGDRNet}(G_1, P_2)\right) \\ &= \text{AGDRNet}\left(G_0, \text{AGDRNet}\left(G_1, \cdots \text{AGDRNet}^*(G_j)\cdots\right)\right) \end{aligned} \tag{4}$$

The final prediction from the network consists of a vertex position tensor, P_v, which contains a predicted vertex for each surface-relevant voxel cell, and an edge occupancy tensor, P_ε. The final mesh, $Q = (V, A)$, is generated by applying the geometric construction principles from our baseline, Unsigned Neural Dual Contouring (UNDC) [CTFZ22]. Specifically, the topology is determined by P_ε; for each occupied edge, the algorithm retrieves the four corresponding vertices from P_v in the adjacent cells to form a quadrilateral face, which is then triangulated to create the final faces A. It should be noted that, characteristic of the UNDC framework, this method prioritizes high-fidelity reconstruction of complex geometric details and sharp features over a strict guarantee of producing watertight or manifold meshes.

3.2. Attention-Guided Dynamic Reconstruction Network

In the multi-scale architecture, the Attention-Guided Dynamic Reconstruction Network (AGDRNet) at each scale $s = 0, \cdots, j-1$ shares the same structure, as shown in Fig. 2. First, the binary voxel input G_s at scale s is fed into the Global-Dynamic Feature Extraction Module (GDFEM) to obtain the dynamic feature representation F_s, which can be expressed as:

$$F_s = GDFEM(G_s), s \in \{0, \cdots, j-1\} \tag{5}$$

where $F_s \in \mathbb{R}^{\frac{M}{2^{s+1}} \times \frac{N}{2^{s+1}} \times \frac{K}{2^{s+1}} \times \omega}$, and ω denotes the number of feature channels.

For the predicted mesh result P_{s+1} from the adjacent lower scale, we apply Sparse Upsampling u [LXL*23]. Each voxel cell is evenly divided into 8 sub-cells with half the original edge length. The vertex and edge occupancy information contained in the original cells is redistributed into the new sub-cells, while empty cells are initialized with null values to prevent introducing artifacts. The resulting sparse prediction data is denoted as $U_s \in \mathbb{R}^{\frac{M}{2^s} \times \frac{N}{2^s} \times \frac{K}{2^s} \times 6}$, which can

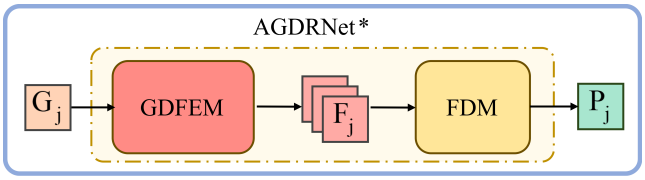

Figure 3: *Framework of the Dynamic Reconstruction Network AGDRNet* at the Lowest Scale.*

be expressed as:

$$U_s = u(P_{s+1}), s \in \{0, \cdots, j-1\} \tag{6}$$

The dynamic feature F_s and the sparse prediction data U_s are fed into the Dynamic Feature Fusion Module (DFFM) to obtain the fused sparse prediction result $W_s \in \mathbb{R}^{\frac{M}{2^s} \times \frac{N}{2^s} \times \frac{K}{2^s} \times 6}$, which can be expressed as:

$$W_s = DFFM(F_s, U_s), s \in \{0, \cdots, j-1\} \tag{7}$$

The dynamic feature F_s is also passed through the Feature Decomposition Module (FDM) to decode an isolated predicted mesh result $P'_s \in \mathbb{R}^{\frac{M}{2^s} \times \frac{N}{2^s} \times \frac{K}{2^s} \times 6}$, which can be expressed as:

$$P'_s = FDM(F_s), s \in \{0, \cdots, j-1\} \tag{8}$$

To ensure effective integration between the fused sparse prediction result W_s and the isolated predicted mesh result P'_s, we combine them and feed the result into the Contour Refinement Postprocess Module (CRPM) for further processing. The final predicted mesh result P_s is obtained as:

$$P_s = CRPM\left(W_s, P'_s\right), s \in \{0, \cdots, j-1\} \tag{9}$$

As shown in Fig. 3, since the lowest scale has no input from any other scale, we denote the special attention-guided dynamic reconstruction network at this scale as AGDRNet*. The binary voxel data $G_j \in \mathbb{R}^{\frac{M}{2^j} \times \frac{N}{2^j} \times \frac{K}{2^j}}$ is first passed into the Dynamic Feature Extraction Module (DFEM) to obtain the dynamic feature representation F_j, expressed as:

$$F_j = GDFEM(G_j) \tag{10}$$

where $F_j \in \mathbb{R}^{\frac{M}{2^j} \times \frac{N}{2^j} \times \frac{K}{2^j} \times \omega}$, ω denotes the number of feature channels.

The dynamic feature F_j is then decoded by the Feature Decomposition Module (FDM) to produce the isolated predicted mesh result $P'_j \in \mathbb{R}^{\frac{M}{2^j} \times \frac{N}{2^j} \times \frac{K}{2^j} \times 6}$. The final predicted mesh result P_j is given by:

$$P_j = P'_j = FDM(F_j) \tag{11}$$

3.2.1. Global Dynamic Feature Extraction Module

Due to the low information density of binary voxel data, we adopt a strategy of enlarging the receptive field to enhance the surface information captured by individual voxel cells during the feature extraction process. Since the input is raw binary voxel data without

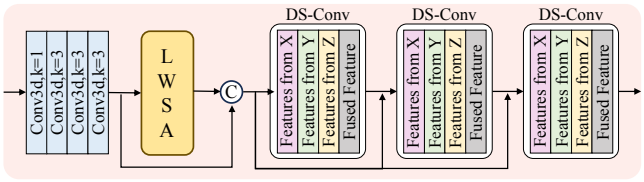

Figure 4: *Structure of the Global Dynamic Feature Extraction Module.*

explicit structural guidance, conventional convolution kernels often fail to distinguish the importance of different regions. This results in a tendency toward "uniform exploration" or "blind extension" under sparse voxel distributions. To address this, we introduce a local structural attention mechanism to improve the network's sensitivity to high-curvature boundaries and complex geometric regions.

As shown in Fig. 4, the Global-Dynamic Feature Extraction Module (GDFEM) aggregates features from binary voxel inputs using three 3D convolutional layers with kernel size $3 \times 3 \times 3$.

To enhance the effectiveness of dynamic snake convolution over binary voxel inputs, we apply a Local Window Self-Attention (LWSA) module to process the aggregated features:

$$A_s = LWSA\left(Conv3d^3\left(G_s\right)\right) \tag{12}$$

Where $A_s \in \mathbb{R}^{\frac{M}{2^s} \times \frac{N}{2^s} \times \frac{K}{2^s}}$ denotes the structural attention map. LWSA is designed based on the Swin3D-Tiny [YGL24] architecture, and calculates self-attention within local windows to strengthen the representation of structural information such as edges and corners. We fuse the original binary voxel input with the attention map to form a structure-enhanced feature $F_s^A \in \mathbb{R}^{\frac{M}{2^s} \times \frac{N}{2^s} \times \frac{K}{2^s} \times (\omega+1)}$, defined as:

$$F_s^A = Concat\left(A_s, Conv3d^3\left(G_s\right)\right) \tag{13}$$

Where $s \in \{0, \cdots, j\}$, *Concat* denotes channel-wise concatenation.

To further enlarge the receptive field and enable voxel cells to capture more distant and complex geometric features, we are inspired by the continuity and curvature characteristics of surfaces in reconstruction tasks. Thus, we adopt the Dynamic Snake Convolution (DS-Conv) [YQZ*24] module in the feature extraction process. In DS-Conv, the learned deformable offset Δ [DQX*17] is used to iteratively extend the convolution kernel in the learned direction for each coordinate, selecting the next sampling position at each step. This ensures continuity of attention and enhances edge perception in 3D shapes under topological and geometric constraints. To address the gradient vanishing problem in DS-Conv, we stack three DS-Conv layers with residual skip connections, allowing dynamic long-range feature extraction of complex geometry. The final dynamic feature output is given by:

$$F_s = DS\_Conv(DS\_Conv(DS\_Conv(F_s^A) + F_s^A) + F_s^A) \tag{14}$$

Where $s \in \{0, \cdots, j\}$, F_s denotes the output dynamic feature, and *DS_Conv* represents the dynamic snake convolution module.

The flexible and continuous deformation of the convolution kernel allows the network to better extract long-range geometric features, particularly in regions where neighboring voxel values change minimally. By expanding the receptive field, each cell can more accurately capture distant structural information. The surface-aligned extension of DS-Conv kernels strengthens surface perception and helps aggregate complex long-range geometric features, ultimately leading to more accurate vertex position estimation within each voxel cell.

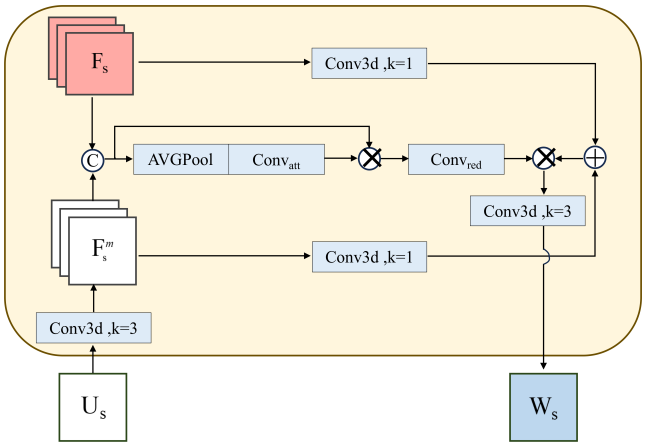

Figure 5: *Structure of the Dynamic Feature Fusion Module.*

3.2.2. Feature Decomposition Module and Dynamic Feature Fusion Module

For the dynamic feature F_s output by the Dynamic Feature Extraction Module (DFEM), we use the Feature Decomposition Module (FDM) to decode it and obtain an isolated predicted mesh result P'_s. Specifically, FDM processes the local features using three 3D convolutional layers with kernel size $3 \times 3 \times 3$, followed by two fully connected layers as the prediction head:

$$P'_s = MLP\left(Conv3d^3\left(F_s\right)\right) \tag{15}$$

Where $s \in \{0, \cdots, j\}$, $Conv3d^3$ denotes the 3-layer 3D convolutional block, and *MLP* represents the 2-layer fully connected network.

For cross-scale information and dynamic feature fusion, we use the Dynamic Feature Fusion Module (DFFM) to integrate the sparse prediction data U_s with the current scale's dynamic features F_s. As shown in Fig. 5, DFFM first transforms the sparse prediction data through a feature conversion layer to obtain sparse features $F_s^m \in \mathbb{R}^{\frac{M}{2^s} \times \frac{N}{2^s} \times \frac{K}{2^s} \times \omega}$, represented as:

$$F_s^m = Conv3d(U_s) \tag{16}$$

Next, DFFM concatenates the dynamic feature F_s and the sparse feature F_s^m, applies pooling, and feeds the result into a channel attention mapping function to generate attention weights:

$$A_s^{ch} = Conv_{\text{att}}\left(AVGPool\left(Concat\left(F_s, F_s^m\right)\right)\right) \tag{17}$$

Where $Conv_{\text{att}}$ is the channel attention mapping function, and *AVGPool* represents the pooling operation.

The channel attention weights A_s^{ch} are applied to the concatenated feature map to produce a weighted feature, which is then passed through a convolutional layer for dimensionality reduction to obtain channel-refined features:

$$F_s^{red} = Conv_{\text{red}}\left(Concat(F_s, F_s^m)\cdot A_s^{ch}\right) \tag{18}$$

Where $Conv_{\text{red}}$ denotes the reduction convolution layer.

In the spatial attention stage, we perform separate convolutions on the dynamic feature F_s and the sparse feature F_s^m to obtain intermediate spatial maps:

$$A_s^{\text{sp1}} = Conv_1(F_s), A_s^{\text{sp2}} = Conv_2\left(F_s^m\right) \tag{19}$$

These maps are added and passed through a Sigmoid activation function to generate the final spatial attention map:

$$A_s^{\text{sp}} = \sigma\left(A_s^{\text{sp1}} + A_s^{\text{sp2}}\right) \tag{20}$$

The spatial attention map A_s^{sp} and channel-refined feature F_s^{red} are fused, then reduced in dimension to produce the sparse prediction result W_s:

$$W_s = Conv3d\left(F_s^{red}\cdot A_{\text{sp}}^{(s)}\right) \tag{21}$$

By incorporating both channel and spatial attention in the DFFM module [DQX*17],the model achieves more coherent and natural cross-scale feature integration. This effec-tively mitigates local geometric distortions caused by feature misalignment during scale transitions, significantly improving the continuity, detail completeness, and visual fidelity of the reconstructed surfaces.

3.2.3. Contour Refinement Postprocess Module

To ensure effective integration between the fused sparse prediction result W_s and the isolated predicted mesh result P_s', we apply the Contour Refinement Postprocess Module (CRPM) to obtain the final predicted mesh result P_s. As shown in Fig. 6, our CRPM first concatenates the fused sparse prediction result W_s and the isolated predicted mesh result P_s' along the channel dimension, and then applies one 3D convolutional layer (with kernel size $3\times3\times3$) to aggregate the concatenated features. The resulting aggregated feature can be expressed as:

$$F_s^1 = Conv3d\left(Concat(W_s, P'_s)\right) \tag{22}$$

Where $s \in \{0,\cdots,j-1\}$, and $F_s^1 \in \mathbb{R}^{\frac{M}{2^s}\times\frac{N}{2^s}\times\frac{K}{2^s}\times\omega}$ is the aggregated feature with ω channels.

The aggregated feature F_s^1 is then downsampled, passed through a 3-layer 3D convolutional block, and upsampled to produce the upsampled residual feature $F_s^2 \in \mathbb{R}^{\frac{M}{2^s}\times\frac{N}{2^s}\times\frac{K}{2^s}\times\omega}$, which can be written as:

$$F_s^2 = Up\left(Conv3d^3\left(Down\left(F_s^1\right)\right)\right) \tag{23}$$

Where $s \in \{0,\cdots,j-1\}$, $Down$ and Up denote downsampling and upsampling operations respectively, and $Conv3d^3$ is a 3-layer 3D convolutional block.

Next, we concatenate the residual feature F_s^2 and the original aggregated feature F_s^1 along the channel dimension and feed the result into a 2-layer 3D convolutional block, followed by two fully connected layers as the prediction head. The final mesh prediction result P_s is given by:

$$P_s = MLP\left(Conv3d^2\left(Concat\left(F_s^1, F_s^2\right)\right)\right) \tag{24}$$

Where $s \in \{0,\cdots,j-1\}$, $Conv3d^2$ denotes the 2-layer 3D convolutional block, and MLP represents the 2-layer fully connected network.

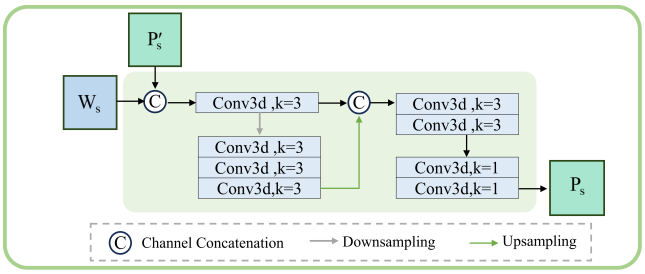

Figure 6: *Structure of the Contour Refinement Postprocess Module.*

3.3. Training Strategy and Loss Function

For each scale $s \in \{0,\cdots,j\}$, we independently train a dynamic reconstruction network to predict the corresponding mesh result $P_s \in \mathbb{R}^{\frac{M}{2^s}\times\frac{N}{2^s}\times\frac{K}{2^s}\times6}$ from the binary voxel input. The predicted output P_s can be decomposed into a predicted vertex set $P_{Vs} \in \mathbb{R}^{\frac{M}{2^s}\times\frac{N}{2^s}\times\frac{K}{2^s}\times3}$ and predicted edge occupancy $P_{\varepsilon s} \in \mathbb{R}^{\frac{M}{2^s}\times\frac{N}{2^s}\times\frac{K}{2^s}\times3}$. Similarly, the ground-truth label $T_s \in \mathbb{R}^{\frac{M}{2^s}\times\frac{N}{2^s}\times\frac{K}{2^s}\times6}$ can be decomposed into the ground-truth vertex set $T_{Vs} \in \mathbb{R}^{\frac{M}{2^s}\times\frac{N}{2^s}\times\frac{K}{2^s}\times3}$ and ground-truth edge occupancy $T_{\varepsilon s} \in \mathbb{R}^{\frac{M}{2^s}\times\frac{N}{2^s}\times\frac{K}{2^s}\times3}$. Specifically, for vertex prediction, we apply Mean Squared Error (MSE) loss to supervise the predicted vertex set P_{Vs}. For edge occupancy prediction, we use Binary Cross-Entropy (BCE) loss to supervise $P_{\varepsilon s}$. These two losses are combined into a unified total loss function $\mathcal{L}(\theta)$, weighted by a learnable parameter λ, and used to train and update the network parameters:

$$\mathcal{L}(\theta) = \mathcal{L}_{\mathcal{V}}(\theta) + \lambda\mathcal{L}_{\varepsilon}(\theta) \tag{25}$$

$$\mathcal{L}_{\mathcal{V}}(\theta) = \mathbb{E}^{\sim}\sum_{\frac{M}{2^s}\times\frac{N}{2^s}\times\frac{K}{2^s}}\left[\|P_{Vs} - T_{Vs}\|_2^2\right] \tag{26}$$

$$\mathcal{L}_{\varepsilon}(\theta) = \mathbb{E}^{\sim}\sum_{\frac{M}{2^s}\times\frac{N}{2^s}\times\frac{K}{2^s}}\left[\|P_{\varepsilon s} - T_{\varepsilon s}\|_2^2\right] \tag{27}$$

Where $s \in \{0,\cdots,j\}$, θ denotes the set of learnable model parameters, and λ is a learnable dynamic weighting parameter that balances the contributions of the two loss components.

4. Experiments

4.1. Datasets

In all experiments conducted in this work, we largely follow the experimental protocol defined in UNDC [CTFZ22], and train our

model on the ABC dataset [KMJ*19]. The ABC dataset consists of over 30,000 CAD mesh models with clear boundaries and diverse curved surfaces, making it well-suited for high-quality reconstruction tasks. For the training process, we use 80% (4,200 shapes) of the first module in the ABC dataset for model training. The remaining 20% (1,050 shapes) from the same module, excluding the training set, is used as the test set. To further evaluate the generalization and robustness of our method, we additionally include 1,100 models from the Thingi10K dataset [ZJ16] for testing. Thingi10K is a 3D printing model dataset that contains high-quality surface models and is widely used in 3D reconstruction and validation tasks, serving as an ideal benchmark for evaluating the applicability and robustness of reconstruction methods. During experiments, our multi-scale network is configured with three scales $s \in \{0,1,2\}$, i.e., the maximum number of downsampling steps is set to $j = 2$. The binary voxel volume resolution $M \times N \times K$ is set to 64^3 to facilitate training and evaluation.

4.2. Evaluation Metrics

We quantitatively evaluate surface reconstruction by uniformly sampling 100,000 points from both the ground-truth mesh and the predicted mesh, and compute the following metrics, which are used to assess reconstruction accuracy and the preservation of sharp features.

For reconstruction accuracy, we use common metrics such as Chamfer Distance (CD), F-Score (F1), and Normal Consistency (NC) to evaluate the overall mesh reconstruction accuracy. These metrics are typically used to measure surface fitting and are good at capturing significant errors, but they may not provide information for evaluating visual quality, such as sharp features at boundaries that significantly affect visual appearance. Therefore, following the approach of UNDC, we add the following two metrics to evaluate the reconstruction of sharp features and boundary details.

For sharp feature preservation, in order to assess the quality of boundary reconstruction, we use Edge Chamfer Distance (ECD) and Edge F-Score (EF1) to evaluate the preservation of sharp edges. For a given shape, points are sampled near sharp edges and corners to form a set of edge samples, and we replace uniform sampling with dense sampling near the boundaries to compute the above two metrics. The ECD and EF1 between two shapes are simply the CD and F1 computed between their edge samples.

4.3. Comparative Experiments

In surface reconstruction tasks, accurately capturing geometric details and sharp edge features is an important criterion for evaluating the effectiveness of a method, and it is especially challenging when dealing with binary voxel data. This section presents the comparative experimental results of AGNDC and several existing mainstream methods on the ABC dataset through Table. 1 and Fig. 7, in order to verify the significant improvements of AGNDC in sharp edge capture and detail preservation. In addition, Table. 2 shows the comparative experimental results of AGNDC on the Thingi10K dataset, further demonstrating its generalization and applicability across different datasets.

4.3.1. Comparative Experiments on the ABC Dataset

We compare AGNDC with the classical MC33 [Che95], NMC [CZ21], the unsigned variant of NDC named UNDC [CTFZ22], and HRE-NDC [LXL*23], analyzing their differences in reconstruction accuracy, boundary clarity, and detail preservation. Through these comparisons, the advantages of AGNDC in handling binary voxel data can be intuitively evaluated.

Table 1: *Comparison of experimental results on the ABC dataset.*

Model	CD↓ ($\times 10^5$)	F1↑	NC↑	ECD↓ ($\times 10^2$)	EF1↑
MC33 [Che95]	26.144	0.103	0.920	10.234	0.021
NMC [CZ21]	9.828	0.420	0.927	0.604	0.356
UNDC [CTFZ22]	9.815	0.401	0.930	0.555	0.373
HRE-NDC [LXL*23]	9.612	0.408	0.934	0.541	0.381
AGNDC	**9.013**	**0.440**	**0.942**	**0.507**	**0.398**

From the experimental results in Table 1, it can be seen that AGNDC demonstrates significant advantages across all key metrics. In terms of reconstruction accuracy, the Chamfer Distance (CD) of AGNDC is 9.013, which is approximately 65.5% lower than that of the traditional method MC33 (26.144). Compared to NMC (9.828), UNDC (9.815), and HRE-NDC (9.612), AGNDC achieves reductions of about 8.3%, 8.2%, and 6.2%, respectively. This indicates that AGNDC significantly improves overall mesh fitting accuracy and can more precisely reconstruct surfaces, especially in recovering complex geometric features, avoiding large errors commonly seen in traditional methods.

In terms of boundary detail preservation, AGNDC also performs remarkably well. Its Edge Chamfer Distance (ECD) is 0.507, which is 16.0% lower than that of NMC (0.604), 8.6% lower than UNDC (0.555), and 6.3% lower than HRE-NDC (0.541). At the same time, AGNDC achieves an Edge F1 score (EF1) of 0.398, which is 11.8% higher than NMC (0.356), 6.7% higher than UNDC (0.373), and 4.5% higher than HRE-NDC (0.381). These results indicate that AGNDC can better preserve sharp edges and geometric details when processing binary voxel data.

AGNDC also excels in Normal Consistency (NC), with a value of 0.942, which is 1.6% higher than NMC (0.927), 1.3% higher than UNDC (0.930), and 0.9% higher than HRE-NDC (0.934). This further demonstrates that AGNDC not only has advantages in detail reconstruction but can also effectively maintain surface smoothness and geometric accuracy.

The overall F1 score of AGNDC is 0.440, showing improvements of 4.8% over NMC (0.420), 9.7% over UNDC (0.401), and 7.8% over HRE-NDC (0.408). This result further validates AGNDC's advantage in preserving detail and improving accuracy, demonstrating its comprehensive superiority in binary voxel surface reconstruction tasks.

As shown in the comparative results in Fig. 7, the visual evidence further supports the quantitative analysis in Table 1. Although MC33, as a traditional method, provides lower computational complexity in certain cases, the generated surfaces often exhibit obvious jagged artifacts and blurriness. This is particularly

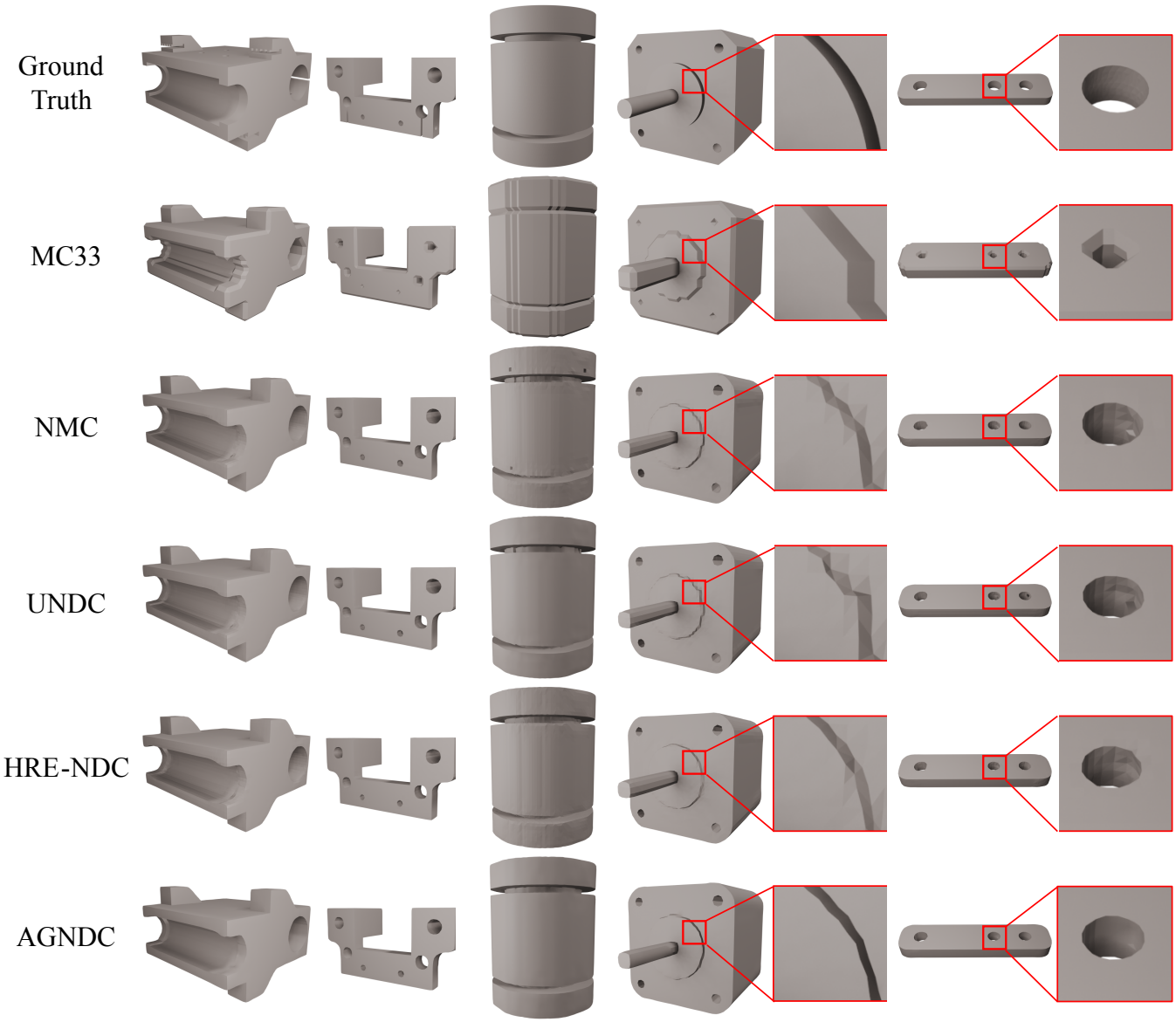

Figure 7: *Comparative visualization of experimental results on the ABC.*

severe in complex structures and sharp corners, where geometric features of the target objects cannot be accurately recovered.

In contrast, while NMC improves surface smoothness and reduces jaggedness to some extent through deep learning optimization, it still suffers from blurring when handling complex boundaries and sharp corners due to its reliance on templates. Especially in sharp-edge regions, the reconstruction results of NMC appear overly smoothed, resulting in the loss or over-simplification of sharp features. UNDC abandons traditional template construction and adopts neural feature extraction, showing better performance in detail reconstruction, especially in the recovery of simple edges. However, due to the low information density of binary voxel data, UNDC still faces challenges in extracting long-range edge information and preserving high-curvature detail regions. In some areas, jagged artifacts and overly smoothed sharp edges still occur. HRE-NDC uses sparse convolution for multi-scale reconstruction, but lacks effective cross-scale feature fusion for binary voxel data. As a result, some regions still exhibit jaggedness and unsmooth surface artifacts.

Compared to the above methods, AGNDC demonstrates significantly superior performance. In every example, the generated mesh accurately preserves sharp edges and complex geometric details, almost eliminating jaggedness. The results in Fig. 7 clearly show that AGNDC can accurately reconstruct details in complex shapes, especially in sharp-edge regions, where AGNDC effectively avoids the distortion and detail loss commonly seen in other methods.

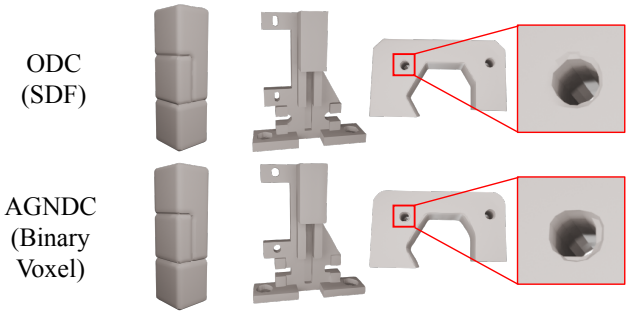

Figure 8: *Qualitative comparison at 128³ resolution on ABC dataset. AGNDC uses binary input; ODC uses SDF input. AGNDC achieves similar geometric detail without requiring continuous fields.*

To further evaluate the performance of our method, we qualitatively compare AGNDC (with binary voxel input) against ODC [HS24] (with continuous signed distance field input) on ABC shapes at a resolution of 128³. As illustrated in Fig. 8, although AGNDC exhibits slightly less surface smoothness—primarily due to the limited granularity and discontinuous nature of binary input—it achieves comparable results in terms of overall geometry preservation and edge fidelity. This comparison highlights the robustness of AGNDC under significantly weaker input conditions.It is worth emphasizing that signed distance fields (SDFs) provide continuous scalar values that encode sub-voxel geometric information, resulting in much higher information density than binary voxel grids. Despite this disadvantage, AGNDC maintains competitive reconstruction quality without relying on such priors, demonstrating its effectiveness in practical binary-to-mesh applications.

Overall, experiments on the ABC dataset confirm that AGNDC effectively reconstructs sharp edges and complex structures, outperforming baselines in preserving geometric fidelity. Its strong generalization and robustness make it a reliable solution for binary voxel surface reconstruction.

4.3.2. Comparative Experiments on the Thingi10K Dataset

To verify the generalization and applicability of our method, we conducted comparative experiments on the Thingi10K dataset to evaluate its adaptability to diverse data and to further complement the comprehensiveness of our experimental analysis. The corresponding results are presented in Table. 2.

From the data in Table. 2, it can be observed that AGNDC demonstrates significant advantages across multiple key metrics. In terms of reconstruction accuracy, the Chamfer Distance (CD) of AGNDC is 6.020, which is approximately 76.4% lower than that of the traditional method MC33 (25.523). Compared to NMC (6.256), UNDC (6.187), and HRE-NDC (6.083), it is reduced by approximately 3.8%, 2.7%, and 1.0%, respectively. This result indicates that AGNDC performs excellently in overall reconstruction accuracy and can more precisely fit the surface geometry.

In terms of boundary detail preservation, AGNDC also shows outstanding performance. Its Edge Chamfer Distance (ECD) is 0.621, which is 19.6% lower than NMC (0.772), 8.8% lower than UNDC (0.681), and 5.0% lower than HRE-NDC (0.654). Meanwhile, AGNDC achieves an Edge F1 score (EF1) of 0.339, which is 10.8% higher than NMC (0.306), 5.3% higher than UNDC (0.322), and 3.0% higher than HRE-NDC (0.329). These results indicate that AGNDC has a significant advantage in preserving sharp edge details.

Table 2: *Comparison of experimental results on the Thingi10K dataset.*

Model	CD↓ ($\times 10^5$)	F1↑	NC↑	ECD↓ ($\times 10^2$)	EF1↑
MC33 [Che95]	25.523	0.069	0.907	7.542	0.017
NMC [CZ21]	6.256	0.471	0.916	0.772	0.306
UNDC [CTFZ22]	6.187	0.477	0.921	0.681	0.322
HRE-NDC [LXL*23]	6.083	0.481	0.927	0.654	0.329
AGNDC	**6.020**	**0.487**	**0.935**	**0.621**	**0.339**

In terms of Normal Consistency (NC), AGNDC achieves a score of 0.935, which is 1.7% higher than NMC (0.916), 1.2% higher than UNDC (0.921), and 0.9% higher than HRE-NDC (0.927). This further demonstrates AGNDC's ability to generate smooth and highly consistent meshes.

For the overall F1 score (F1), AGNDC reaches 0.487, representing an improvement of 3.4% over NMC (0.471), 2.1% over UNDC (0.477), and 1.2% over HRE-NDC (0.481). This metric confirms the superior comprehensive performance of AGNDC, especially in balancing global reconstruction accuracy and detailed feature representation.

In summary, in the comparative experiments on the Thingi10K dataset, AGNDC exhibits comprehensive advantages in reconstruction accuracy, boundary detail preservation, and overall consistency, further validating its powerful performance and broad applicability in handling complex binary voxel data.

4.4. Implementation Details and Performance Analysis

Compared to the baselines, the AGNDC method exhibits higher computational requirements, a deliberate trade-off resulting from its complex architecture necessary to achieve significant improvements in reconstruction quality. We argue this trade-off is justified by the substantial gains in geometric fidelity. Crucially, we employ a sequential training strategy where each multi-scale model is trained independently. Consequently, as detailed in Table. 3, the reported GPU memory of approximately 10GB reflects the requirement for a single model, while the total training time of approximately 42 hours is the sum of all stages. This design ensures that the entire framework can be trained and used for inference on a single consumer-grade GPU (e.g., NVIDIA GeForce RTX 2080 Ti), affirming the method's practical viability.

4.5. Module Effectiveness Evaluation

To verify the actual contribution of each module in AGNDC, we conducted ablation experiments on the ABC test set by sequen-

Table 3: *Performance benchmark on an NVIDIA GeForce RTX 2080 Ti.*

Method	Training Time	Inference Time	GPU Memory
UNDC [CTFZ22]	≈12 h	≈0.039s	≈2GB
HRE-NDC [LXL*23]	≈18 h	≈0.061s	≈3GB
AGNDC	≈42 h	≈0.122s	≈10GB

tially introducing the Global-Dynamic Feature Extraction Module (GDFEM), Dynamic Feature Fusion Module (DFFM), and Contour Refinement Postprocess Module (CRPM), in order to evaluate their impact on reconstruction accuracy, detail preservation, and edge performance. The experimental results are shown in Table. 4.

Table 4: *Ablation comparison of module effectiveness.*

GDFEM	DFFM	CRPM	CD↓ ($\times 10^5$)	F1↑	NC↑	ECD↓ ($\times 10^2$)	EF1↑
			9.815	0.401	0.930	0.555	0.373
✓			9.297	0.426	0.933	0.522	0.382
✓	✓		9.027	0.439	0.940	0.511	0.397
✓	✓	✓	**9.013**	**0.440**	**0.942**	**0.507**	**0.398**

Table. 4 shows the impact of introducing each module on the reconstruction performance. First, after adding GDFEM, the Chamfer Distance (CD) decreased from 9.815 to 9.297, a reduction of 5.3%; the Edge Chamfer Distance (ECD) dropped from 0.555 to 0.522, a decrease of 6.0%; and the Edge F1 score (EF1) increased to 0.382, an improvement of approximately 2.4%. This indicates that the attention-guided dynamic feature extraction module can effectively enhance the perception of long-range geometric structures and alleviate information loss caused by voxel sparsity.

With the addition of DFFM on top of GDFEM, the model performance further improved: the CD dropped to 9.027, a 2.9% reduction from the previous stage; EF1 increased to 0.397, an improvement of 3.9%, indicating that DFFM plays a positive role in multi-scale information alignment and fusion, enhancing overall reconstruction continuity and detail recovery capability.

Finally, with CRPM introduced on top of GDFEM and DFFM, the model achieved optimal performance across all metrics. CD further decreased to 9.013, ECD dropped to 0.507, and EF1 reached 0.398, representing improvements of 0.2%, 0.8%, and 0.3% respectively compared to the previous configuration, validating the effectiveness of the post-processing module in fine boundary reconstruction and error suppression.

In summary, the ablation experiment results show that the proposed GDFEM, DFFM, and CRPM play key roles in improving the reconstruction quality, and together form the optimal module combination within the AGNDC framework. These enhancements demonstrate strong advantages in restoring sharp edges and accurately reconstructing geometric details, providing an effective solution for high-quality surface reconstruction from binary voxel data.

4.6. Hyperparameter Experiments

To further optimize the performance of the AGNDC network, we conducted experiments on two key hyperparameters: the number of multi-scale layers and the feature channel width. In the experiments, we explored how different settings of multi-scale layers and feature channels affect reconstruction accuracy, edge detail preservation, and overall reconstruction quality. The experimental results are shown in Table. 5 and Table. 6.

4.6.1. Multi-scale Layer Comparison Experiment

We conducted experiments on the number of multi-scale layers in the AGNDC network, configuring the multi-scale network as $s \in \{0, \cdots, j\}$, and studied the effect of different maximum downsampling times j on performance. The experimental results are shown in Table. 5.

Table 5: *Comparison results of different multi-scale layer configurations.*

j	$s \in \{\cdot\}$	CD↓ ($\times 10^5$)	F1↑	NC↑	ECD↓ ($\times 10^2$)	EF1↑
0	0	9.297	0.426	0.933	0.522	0.382
1	0,1	9.027	0.438	0.939	0.511	0.394
2	0,1,2	**9.013**	**0.440**	**0.942**	**0.507**	**0.398**
3	0,1,2,3	9.163	0.428	0.937	0.512	0.394

In the experiment, when $j = 0$, the model relied only on GDFEM and FDM for reconstruction, showing performance consistent with the results in the module ablation experiment, and serving as a baseline configuration for comparison. After introducing one layer of the multi-scale structure and DFFM, the model performance improved. Specifically, when $j = 1$, CD decreased by 3.5%, F1 increased by 3.3%, ECD decreased by 1.7%, and EF1 increased by 1.8%, indicating that an appropriate multi-scale structure can enhance the model's local detail modeling and overall reconstruction accuracy.

With a further increase in downsampling steps to $j = 2$, the network scale $s \in \{0,1,2\}$ achieved the best results across all metrics. Compared to the single-scale configuration, CD decreased by 3.7%, F1 increased by 3.8%, ECD decreased by 2.5%, and EF1 increased by 2.8%, suggesting that the model can better fuse multi-layer features and balance geometric restoration with edge detail representation.

However, when the downsampling steps were further increased to $j = 3$, and the network scale became $s \in \{0,1,2,3\}$, the model performance degraded. The binary voxel data at the lowest scale $s = 3$ had undergone three downsampling steps, resulting in a resolution of $8 \times 8 \times 8$, which led to significant geometric information loss and a performance drop: CD = 9.163, F1 = 0.428, NC = 0.937, ECD = 0.512, EF1 = 0.394, all worse than those at $j = 2$.

In conclusion, an appropriate multi-scale configuration significantly improves the reconstruction quality. When $j = 2$, the network scale $s \in \{0,1,2\}$ achieves optimal performance, effectively balancing computational complexity and reconstruction accuracy.

4.6.2. Comparison of Feature Channel Numbers

We examined the impact of the feature channel number ω on the performance of the AGNDC network. In the experiment, we adjusted the feature channel number in the Attention-Guided Dynamic Reconstruction Network (AGDRNet) and evaluated the network's performance under different channel settings.

Table 6: *Comparison of different feature channel numbers.*

ω	CD↓ ($\times 10^5$)	F1↑	NC↑	ECD↓ ($\times 10^2$)	EF1↑
16	9.137	0.425	0.936	0.521	0.392
32	9.046	0.432	0.939	0.518	0.391
64	**9.013**	**0.440**	**0.942**	**0.507**	**0.398**
128	9.279	0.421	0.933	0.527	0.388

From the results in Table. 5, it can be seen that the model performs best when the channel number is $\omega = 64$. Compared to $\omega = 16$, CD is reduced by approximately 1.4%, F1 increases by around 3.5%, ECD decreases by 2.7%, and EF1 improves by 1.5%. This improvement indicates that for sparse binary voxel data, appropriately increasing the number of feature channels can enhance the model's feature representation capability, thereby improving the overall reconstruction performance.

When the channel number is $\omega = 32$, compared to $\omega = 64$, CD increases by approximately 0.4%, F1 decreases by about 1.8%, ECD increases by 2.2%, and EF1 decreases by 1.8%. When the channel number is $\omega = 128$, the model performance drops significantly. Compared to $\omega = 64$, CD increases by about 2.9%, F1 decreases by 4.3%, EF1 decreases by 2.5%, and ECD increases by 4.0%. This indicates that an excessively high number of channels increases the network's computational complexity without improving reconstruction quality, and even leads to performance degradation. Therefore, the experimental results show that setting $\omega = 64$ achieves the optimal balance between maintaining high reconstruction accuracy and detail preservation.

5. Conclusion

High-quality surface reconstruction from binary voxel data is a key task in computer vision and graphics, supporting accurate 3D visualization in areas like medical image segmentation. To overcome the limitations of existing data-driven methods in producing high-quality meshes, we propose AGNDC, a method designed to enhance voxel-to-mesh reconstruction accuracy. AGNDC employs an attention-guided feature extraction module that combines global attention with dynamic convolution to better capture complex structures. A dynamic feature fusion module aligns multi-resolution features to refine geometry progressively, while a contour refinement module improves boundary quality and suppresses artifacts. Experiments show that this collaborative design significantly enhances structural continuity and detail restoration [VSP*17], reducing jaggedness and preserving true surface shapes in 3D reconstruction. We believe the synergy of attention and dynamic features offers promising directions for high-precision 3D modeling [GXL*22].

Future work will proceed along three primary directions: enhancing robustness, broadening generalization, and ensuring topological integrity. We plan to investigate the model's performance on noisy data and extend our evaluation to non-CAD organic shapes. To complement our method's design trade-off, which prioritizes high geometric fidelity over topological integrity, we will also explore integrating post-processing techniques such as hole filling and non-manifold topology repair. Furthermore, a crucial direction for future work is to investigate the model's scalability to higher-resolution inputs, analyzing the trade-offs between computational cost and the gains in geometric fidelity required for precision-critical applications. Successfully addressing these challenges will further broaden AGNDC's potential in domains like medical imaging, digital dentistry, and virtual reality.

References

[BWT24] BAI Y., WONG L., TWAN T.: Survey on fundamental deep learning 3d reconstruction techniques. *arXiv preprint arXiv:2407.08137* (2024). 1

[Che95] CHERNYAEV E.: *Marching cubes 33: Construction of topologically correct isosurfaces*. Tech. rep., 1995. 7, 9

[CHY*24] CHEN M.-X., HU H., YAO R., QIU L., LI D.: A survey on the design of virtual reality interaction interfaces. *Sensors 24*, 19 (2024), 6204. doi:10.3390/s24196204. 1

[CTFZ22] CHEN Z., TAGLIASACCHI A., FUNKHOUSER T., ZHANG H.: Neural dual contouring. *ACM Transactions on Graphics (TOG) 41*, 4 (2022), 1–13. doi:10.1145/3528223.3530108. 1, 3, 4, 6, 7, 9, 10

[CZ19] CHEN Z., ZHANG H.: Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2019), pp. 5939–5948. 1

[CZ21] CHEN Z., ZHANG H.: Neural marching cubes. *ACM Transactions on Graphics (TOG) 40*, 6 (2021), 1–15. doi:10.1145/3478513.3480518. 1, 2, 7, 9

[DQX*17] DAI J., QI H., XIONG Y., LI Y., ZHANG G., HU H., WEI Y.: Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision* (2017), pp. 764–773. 5, 6

[GCX*20] GAO J., CHEN W., XIANG T., JACOBSON A., MCGUIRE M., FIDLER S.: Learning deformable tetrahedral meshes for 3d reconstruction. *Advances in neural information processing systems 33* (2020), 9936–9947. 2

[GH97] GARLAND M., HECKBERT P. S.: Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (1997), pp. 209–216. 2

[GSW*22] GAO J., SHEN T., WANG Z., CHEN W., YIN K., LI D., LITANY O., GOJCIC Z., FIDLER S.: Get3d: A generative model of high quality 3d textured shapes learned from images. *Advances in neural information processing systems 35* (2022), 31841–31854. 3

[GXL*22] GUO M.-H., XU T.-X., LIU J.-J., LIU Z.-N., JIANG P.-T., MU T.-J., ZHANG S.-H., MARTIN R. R., CHENG M.-M., HU S.-M.: Attention mechanisms in computer vision: A survey. *Computational visual media 8*, 3 (2022), 331–368. doi:10.1007/s41095-022-0271-y. 11

[HS24] HWANG J., SUNG M.: Occupancy-based dual contouring. In *SIGGRAPH Asia 2024 Conference Papers* (2024), pp. 1–11. 2, 9

[JLSW02] JU T., LOSASSO F., SCHAEFER S., WARREN J.: Dual contouring of hermite data. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques* (2002), pp. 339–346. 2, 3

[KKLD23] KERBL B., KOPANAS G., LEIMKÜHLER T., DRETTAKIS G.: 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph. 42*, 4 (2023), 139–1. 3

[KMJ*19] KOCH S., MATVEEV A., JIANG Z., WILLIAMS F., ARTEMOV A., BURNAEV E., ALEXA M., ZORIN D., PANOZZO D.: Abc: A big cad model dataset for geometric deep learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2019), pp. 9601–9611. 7

[KYZB19] KAISER A., YBANEZ ZEPEDA J. A., BOUBEKEUR T.: A survey of simple geometric primitives detection methods for captured 3d data. In *Computer Graphics Forum* (2019), vol. 38, Wiley Online Library, pp. 167–196. doi:10.1111/cgf.13451. 1

[LC98] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*. 1998, pp. 347–353. 1, 2

[LDG18] LIAO Y., DONNE S., GEIGER A.: Deep marching cubes: Learning explicit surface representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 2916–2925. 1, 2

[LLY*21] LI Z., LIU F., YANG W., PENG S., ZHOU J.: A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems 33*, 12 (2021), 6999–7019. doi:10.1109/TNNLS.2021.3084827. 2

[LXL*23] LIU Q., XIAO J., LIU L., WANG Y., WANG Y.: High-resolution and efficient neural dual contouring for surface reconstruction from point clouds. *Remote Sensing 15*, 9 (2023), 2267. doi:10.3390/rs15092267. 3, 4, 7, 9, 10

[MKFW12] MAJKA P., KUBLIK E., FURGA G., WÓJCIK D. K.: Common atlas format and 3d brain atlas reconstructor: infrastructure for constructing 3d brain atlases. *Neuroinformatics 10* (2012), 181–197. doi:10.1007/s12021-011-9138-6. 1

[MST*21] MILDENHALL B., SRINIVASAN P. P., TANCIK M., BARRON J. T., RAMAMOORTHI R., NG R.: Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM 65*, 1 (2021), 99–106. 3

[NY06] NEWMAN T. S., YI H.: A survey of the marching cubes algorithm. *Computers & Graphics 30*, 5 (2006), 854–879. doi:10.1016/j.cag.2006.07.021. 1

[QHQ*23] QI Y., HE Y., QI X., ZHANG Y., YANG G.: Dynamic snake convolution based on topological geometric constraints for tubular structure segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision* (2023), pp. 6070–6079. 2

[RPW*25] RANADE S., PAIS G. D., WHITAKER R. T., MIRALDO P., NASCIMENTO J., RAMALINGAM S.: Surfr: Surface reconstruction with multi-scale attention. In *International Conference on 3D Vision 2025* (2025). 3

[SFD*23] SCHMIED A., FISCHER T., DANELLJAN M., POLLEFEYS M., YU F.: R3d3: Dense 3d reconstruction of dynamic scenes from multiple cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), pp. 3216–3226. 1

[SGEB00] STEINBACH E., GIROD B., EISERT P., BETZ A.: 3-d object reconstruction using spatially extended voxels and multi-hypothesis voxel coloring. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000* (2000), vol. 1, IEEE, pp. 774–777. doi:10.1109/ICPR.2000.905504. 1

[SMH*23] SHEN T., MUNKBERG J., HASSELGREN J., YIN K., WANG Z., CHEN W., GOJCIC Z., FIDLER S., SHARP N., GAO J.: Flexible isosurface extraction for gradient-based mesh optimization. *ACM Transactions on Graphics (TOG) 42*, 4 (2023), 1–16. doi:10.1145/3592430. 3

[SW02] SCHAEFER S., WARREN J.: Dual contouring: The secret sauce. *Technical Report 2*, 408 (2002). 3

[VSP*17] VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER Ł., POLOSUKHIN I.: Attention is all you need. *Advances in neural information processing systems 30* (2017). 11

[WBZ*20] WU Y., BOOMINATHAN V., ZHAO X., ROBINSON J. T., KAWASAKI H., SANKARANARAYANAN A., VEERARAGHAVAN A.: Freecam3d: Snapshot structured light 3d with freely-moving cameras. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16* (2020), Springer, pp. 309–325. doi:10.1007/978-3-030-58583-9_19. 1

[YGL24] YANG Y.-Q., GUO Y.-X., LIU Y.: Swin3d++: Effective multi-source pretraining for 3d indoor scene understanding. *arXiv preprint arXiv:2402.14215* (2024). 5

[YQZ*24] YANG J., QIU P., ZHANG Y., MARCUS D. S., SOTIRAS A.: D-net: Dynamic large kernel with dynamic feature fusion for volumetric medical image segmentation. *arXiv preprint arXiv:2403.10674* (2024). 5

[ZJ16] ZHOU Q., JACOBSON A.: Thingi10k: A dataset of 10,000 3d-printing models. *arXiv preprint arXiv:1605.04797* (2016). 7