electronics

MDPI

Article

3DRecNet: A 3D Reconstruction Network with Dual Attention and Human-Inspired Memory

Muhammad Awais Shoukat [1], Allah Bux Sargano [1], Lihua You [2] and Zulfiqar Habib [1,*]

[1] Department of Computer Science, COMSATS University Islamabad, Lahore Campus, Lahore 54000, Pakistan; sp19-pcs-001@cuilahore.edu.pk (M.A.S.); allahbux@cuilahore.edu.pk (A.B.S.)
[2] National Centre for Computer Animation, Bournemouth University, Poole BH12 5BB, Dorset, UK; lyou@bournemouth.ac.uk
* Correspondence: drzhabib@cuilahore.edu.pk

Abstract: Humans inherently perceive 3D scenes using prior knowledge and visual perception, but 3D reconstruction in computer graphics is challenging due to complex object geometries, noisy backgrounds, and occlusions, leading to high time and space complexity. To addresses these challenges, this study introduces 3DRecNet, a compact 3D reconstruction architecture optimized for both efficiency and accuracy through five key modules. The first module, the Human-Inspired Memory Network (HIMNet), is designed for initial point cloud estimation, assisting in identifying and localizing objects in occluded and complex regions while preserving critical spatial information. Next, separate image and 3D encoders perform feature extraction from input images and initial point clouds. These features are combined using a dual attention-based feature fusion module, which emphasizes features from the image branch over those from the 3D encoding branch. This approach ensures independence from proposals at inference time and filters out irrelevant information, leading to more accurate and detailed reconstructions. Finally, a Decoder Branch transforms the fused features into a 3D representation. The integration of attention-based fusion with the memory network in 3DRecNet significantly enhances the overall reconstruction process. Experimental results on the benchmark datasets, such as ShapeNet, ObjectNet3D, and Pix3D, demonstrate that 3DRecNet outperforms existing methods.

Keywords: 3DRecNet; human-inspired memory network; 3D estimation; dual attention mechanism; fusion-based 3D reconstruction; point clouds

Citation: Shoukat, M.A.; Sargano, A.B.; You, L.; Habib, Z. 3DRecNet: A 3D Reconstruction Network with Dual Attention and Human-Inspired Memory. *Electronics* **2024**, *13*, 3391. https://doi.org/10.3390/electronics13173391

Academic Editor: Beiwen Li

Received: 24 June 2024
Revised: 16 August 2024
Accepted: 21 August 2024
Published: 26 August 2024

1. Introduction

Three-dimensional creation has various industrial applications including robot vision and navigation, autonomous driving, 3D printing and modeling, image-assisted surgery, clinical assessment and guidance, augmented reality, 3D buildings and maps visualization [1–5]. Despite their significance, most scene-capturing devices are limited to recording 2D pixels (x, y) and projecting them onto a flat image plane. In contrast, 3D models differ from the image plane as they include depth information for each pixel, represented mainly through voxel grids and point clouds. Capturing or designing 3D models manually is resource-intensive, costly, and economically challenging [6]. To address these challenges, researchers have proposed methods to estimate 3D models from 2D images [1,4,5]. The 2D image along with its corresponding 3D models are shown in Figure 1.

Three-dimensional reconstruction techniques can be categorized into conventional and learning-based methods. Conventional methods rely on principles of multi-perspective geometry, such as 3D reconstruction from multiple views. For instance, stereo-based techniques compare features between images to measure perspective and viewpoint differences, creating a disparity map to reconstruct the 3D coordinates of the image pixels [1]. Other notable studies in this category include structure from motion [7], and simultaneous localization

and mapping [8], which derive 3D coordinates from multiple 2D views. Despite their ability to produce high-quality reconstructions, these techniques are often time-consuming and require significant expertise for data acquisition [1].

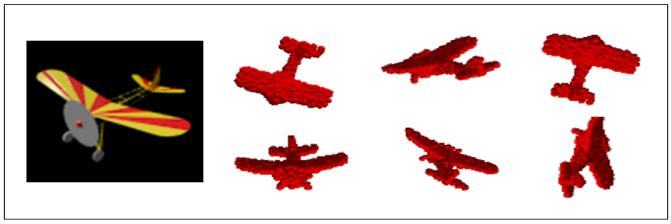


Figure 1. 2D Image along with corresponding different views of 3D Model.

In contrast, learning-based methods have gained considerable importance due to significant advancements in machine learning and computational power. These methods facilitate the reconstruction of 3D geometry directly from single 2D images, effectively addressing challenges related to missing depth cues and complex geometrical data [9–14]. Many of these methods often treat this challenge as a classification task, utilizing diverse datasets to predict the 3D shapes and structures. For example, some methods retrieve and deform shape components to create composite shapes [9,15], but directly extracting shape information from images is complex due to objects' diverse shapes and geometries. Other techniques decompose the task into multiple sub-tasks, such as learning depth, segmentation, and normal maps as intermediate steps, followed by geometric transformations or 3D back-projection to restore the 3D geometry [1]. Early methods trained these components separately; whereas, recent advancements have introduced end-to-end architectures [16,17], typically using an auto-encoder and decoder to estimate intermediate steps and produce a complete 3D volumetric grid, with some methods incorporating shape priors to enhance reconstruction accuracy [18,19].

Recently, researchers have increasingly focused on end-to-end learning methods that directly learn 3D representations (e.g., voxels and point clouds) from single images. Within voxel-based approaches, Tahir et al. [20] proposed a variational autoencoder to create smoother and higher-resolution 3D models, where the encoder learns a latent representation from the image, and the decoder generates the corresponding 3D voxels. Building on these advancements, researchers have explored multi-scale context-aware fusion [21], RNN-based discriminative neural networks [22], generative adversarial networks [23], and the integration of transformers into encoders [24] to enhance these representations. A memory-based framework has also been developed to handle occluded 3D models [5]. Despite these advancements, voxel reconstruction still faces challenges due to sparse spatial information and high computational costs, resulting in inefficient sampling and low-detail models [6].

Alternatively, point cloud-based methods represent geometric information as a set of data points defined by (x, y, z) coordinates. This structure is simpler, requires less memory, and conveys rich 3D information. The point set generation network (PSGN) in [12] was the first to apply a deep learning architecture (i.e., encoder-decoder) to generate 3D point clouds directly from single images. Since then, various single-stage and two-stage networks have enhanced its evaluation results in [6,25,26]. However, reconstructing complex shapes and backgrounds, especially in the presence of occlusions, remains challenging, prompting ongoing research to develop more efficient and accurate methodologies.

Considering these factors, we introduced 3DRecNet, a novel 3D reconstruction network composed of five key modules. The first module, the Human-Inspired Memory Network (HIMNet), is designed for initial point cloud estimation by localizing objects in occluded and complex regions and retrieving similar structured proposals within the same object category. These proposals help 3DRecNet learn and generalize complex shapes and geometric parameters. Next, separate image and 3D encoders perform feature extraction from input images and initial point clouds. These features are combined using a Dual Attention-based Feature Fusion module, which emphasizes features from the image branch

over those from the 3D encoding branch. This approach ensures independence from the proposal at inference time and filters out irrelevant information, leading to more accurate and detailed reconstructions. Finally, a Decoder Branch transforms the fused features into a 3D representation. The integration of attention-based fusion with the memory network in 3DRecNet significantly enhances the overall reconstruction process.

The complete steps of the proposed architecture are discussed in the research methodology section. To evaluate the effectiveness of the method, experiments were conducted on ObjectNet3D [15], ShapeNet [27] and Pix3D datasets [16]. These datasets have different data distributions: ObjectNet3D contains real images with complex backgrounds and occluded objects; ShapeNet consists of rendered images of 3D Models with clean backgrounds; and Pix3D dataset includes real images with cameras focused on objects. Three samples from each dataset are shown in Figure 2.

The major contributions of the paper are as follows:

- The Human-Inspired Memory Network (HIMNet) is proposed, drawing inspiration from the human brain's ability to remember and utilize past experiences. HIMNet retains crucial spatial information regarding an object's category, shape, and geometry, even in complex and occluded regions. This capability significantly enhances the accuracy and completeness of 3D reconstruction within the end-to-end learning framework of 3DRecNet.
- A deep attention-based fusion mechanism is presented for the intelligent fusion of image-encoded features with the initial point cloud features from HIMNet. This approach helps to learn complex object shapes and geometries, enhancing learning accuracy and reducing feature loss.
- Extensive experiments are conducted and the results analyzed while tuning different hyperparameters to make the architecture more adaptable and accurate.

The rest of the paper is organized as follows: Section 2 reviews the literature on 3D model reconstruction. Section 3 describes the proposed approach in detail. Section 4 provides the experimental results and analysis of the findings. Lastly, the paper is concluded in Section 5.

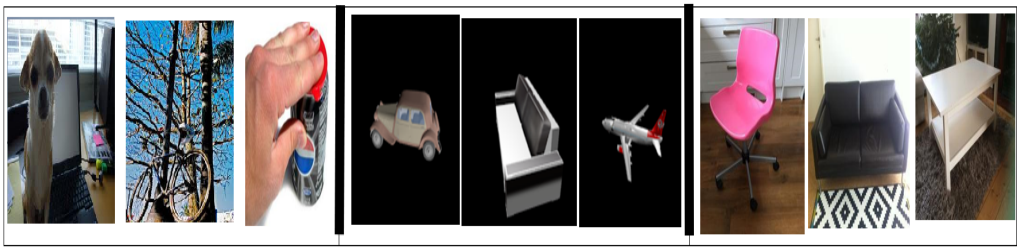


Figure 2. Different distributions of datasets are illustrated. From Left to Right: Images 1–3 sourced from ObjectNet3D, depict occluded objects with complex backgrounds. Images 4–6, from ShapeNet, showed CAD models with plain background. Images 7–9 originate from Pix3D, feature real objects.

2. Related Work

Recent research has focused on direct learning 3D shapes by representing geometric information as a set of (x, y, z) coordinates, providing a simpler, more memory-efficient, and richer representation of 3D information. The initial approach [12], utilized an encoder-decoder to directly generate 3D point clouds from single images, employing chamfer and earth mover's distances as loss functions for object geometry learning. Subsequent studies enhanced this by introducing diverse network architectures, such as a point cloud autoencoder in [28], which mapped images to learned embeddings and incorporated a diversity loss for uncertain reconstruction. Another novel method involved a conditional flow-based generative model [28], distinct from Variational Autoencoders (VAEs) or Generative Adversarial Networks (GANs). The single-encoder multiple-decoder deep network (SE-MD network) in [29] utilized an autoencoder for feature extraction and multiple decoding

networks for point cloud generation, with the final result obtained by fusing all outputs. In contrast, [30] applied a residual network and multi-layer perceptron (MLP) for feature extraction and point set prediction, refining self-occluded parts with a learned Gaussian probability distribution.

Subsequent studies, such as 3D-CDRNet [26], used an autoencoder to design a two-stage point cloud reconstruction network and fuse the feature of the image and proposal retrieval branch. Pixel2point [25] proposed a single-stage network using an initial sphere point cloud rather than the proposal retrieval network to make the algorithm efficient. This method has been further enhanced using Detnet architecture with Exponential Linear Unit (ELU) as an activation function to directly learn the point clouds from a single image and merge the Earth Mover's Distance (EMD) and Chamfer Distance (CD) loss into a unified loss function [6]. Although these methods aim to convert image features directly into 3D representations, they have limitations in learning complex object shapes and backgrounds due to high feature loss.

Inspired by recent approaches [6,25,26], we propose 3DRecNet, an improved 3D reconstruction architecture. Our network integrates a human-inspired memory module (HIMNet Section 3.1) and an attention-based fusion module Section 3.4, retrieving the benefits of two-stage networks into a single-stage design. Unlike traditional two-stage networks, which require extensive proposal searching and may not be as efficient and practical for real-time applications, our single-stage approach overcomes these limitations. It also addresses the challenge of learning complex shapes and geometries directly from images, a common issue in single-stage networks. The dual attention mechanism in 3DRecNet enables the network to focus on relevant features effectively, while the HIMNet module assists in identifying and localizing objects in occluded and complex regions, ensuring the preservation of critical spatial information. Additionally, HIMNet searches for similar-structure proposals within the same category and generates initial point cloud estimates, similar to how humans draw on past experiences and memories to visualize an object's shape, thereby enhancing the reconstruction process.

Moreover, the attention mechanism enhances efficiency by making the HIMNet module optional during inference. This leads to faster real-time 3D estimations, as the HIMNet is exclusively utilized during the network's training phase. Finally, a decoder (MLP branch) transforms these features into accurate 3D representations. Further, the details of the proposed methods are discussed in the next section.

3. Proposed Method

The proposed 3DRecNet combines a Human-Inspired Memory Network (HIMNet) with a deep attention-based fusion mechanism to achieve efficient and accurate 3D reconstructions directly from single images. The network learns a function f to estimate the 3D model M, represented in point clouds, by fusing input image features I with initial geometry from the HIMNet, aiming to closely match the unknown Model Y. This process can be formulated as

$$\begin{aligned} M &= \#(HIMNet, I), \\ f &= loss(M, Y), \end{aligned} \tag{1}$$

where the hash (#) symbol represents the attention-based fusion mechanism discussed in Section 3.3. The loss function involves Earth Mover's Distance (EMD) and Chamfer Distance (CD) to compare the generated and ground-truth 3D models. This strategy is implemented in a five-module deep network:

I. Human-Inspired Memory Network (HIMNet): Preserves critical spatial information by searching for similar-structure proposals within the same category, generating an initial guess of the point clouds.
II. Image Encoder: Extracts high-level features from the input image.
III. 3D Encoder: Processes HIMNet's proposals to extract high-level 3D features.
IV. Dual Attention-based Feature Fusion: Fuses features from the Image and 3D Encoders using a dual attention mechanism.

V. Decoder Branch: Transforms the fused features into a 3D representation.

These modules are illustrated in Figure 3 and discussed in detail in the following subsections.

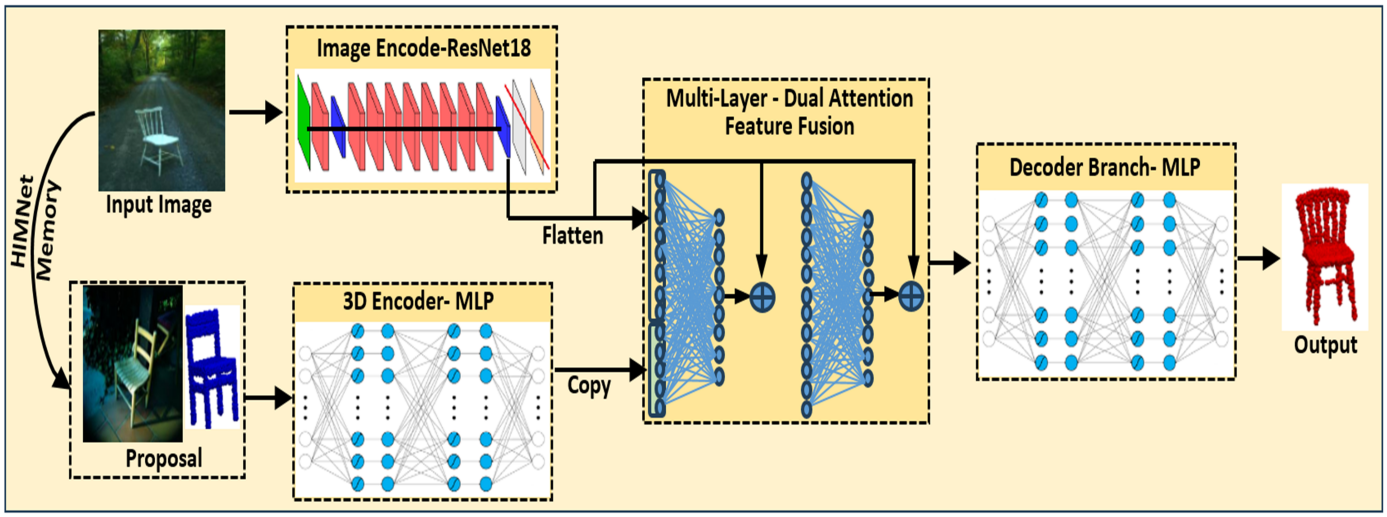


Figure 3. The proposed method, 3DRecNet, reconstructs the 3D shape of an object from a single image. It learns the geometry of the object present in the input image by embedding the attention-based fusion technique on the image encoder and 3D encoder features. Different modules of 3DRecNet are highlighted with dark colors and dotted boundaries.

3.1. Human-Inspired Memory Network-HIMNet

The Human-Inspired Memory Network (HIMNet) addresses the information gap between 2D images and 3D representations by searching for proposals with similar structures within the same category. It generates an initial estimate of point clouds to maintain critical spatial information within the 3DRecNet architecture, thereby enhancing the network's capability to predict 3D shapes from occluded objects and complex backgrounds. The concept is analogous to how humans draw on past experiences and memories to visualize an object's shape. HIMNet represents an evolution of the proposal retrieval network, specifically designed to filter outliers by focusing on proposals with similar structures within the same category. The steps involved in this module are discussed below, and presented in Figure 4.

3.1.1. Image Analysis: Preprocessing, Feature Extraction, and Representation

Initially, preprocessing is applied uniformly across the dataset to normalize and make them compatible with the pre-trained deep learning model, i.e., EfficientNet. Each image I is split into four equal patches to preserve the objects' location, then resized and normalized to meet the EfficientNet's input requirements. The pre-trained model then extracts features from each pre-processed image tiles I_i using the feature extraction function $f_{\text{extract}}(I_i)$ to map the input image to a high-dimensional feature vector $H_{EfficientNet}$ as

$$H_{\text{EfficientNet}}(I_i) = \text{Conv}_L(\text{ReLU}(\text{BN}(\ldots \text{Conv}_2(\text{ReLU}(\text{BN}(\text{Conv}_1(I_i)))) \ldots))). \quad (2)$$

After feature extraction, each image I_i is represented by its feature vector $H_{I,i}$ which captures the image's structure and content, and all the four patches of each image are merged to represent it into a single representation.

$$F_I = [H_{I,i=1}\ H_{I,i=2}\ H_{I,i=3}\ \ldots\ H_{I,i=n}], \quad (3)$$

where, $H_{I,i=1}$ represents the feature vector for the ith tile of image I. After that, the normalization of the feature vector is performed to remove the irregularities/high deviation

in features, by calculating the mean M, standard deviation SD, and then adjusting each feature value relatively. The following equations are used to normalize the feature vectors

$$\begin{aligned} M &= \frac{\sum_i F_i}{N}, \\ SD &= \sqrt{\frac{(F_i - M)^2}{N-1}}, \\ F &= \frac{(F_i - M)}{SD}. \end{aligned} \tag{4}$$

3.1.2. Memory Design Using Structural Similarity Proposals and Metrics

To measure the similarity between two images I_i and I_j, cosine similarity $\text{sim}(F_i, F_j)$ measure is used between their feature vectors as

$$sim(F_i, F_j) = \frac{\mathbf{F}_i \cdot \mathbf{F}_j}{\|\mathbf{F}_i\| \|\mathbf{F}_j\|}. \tag{5}$$

This process efficiently finds similar structured images based on the calculated similarities. For example, the k-nearest neighbors (KNN) search is used to find the proposals further filtered by the classifier trained on dataset classes. The classifier removes the outlier (if any) selected through calculated similarities. After then, proposals are again filtered based on their 3D structures. This step is illustrated in Figure 4. A case study related to HIMNet for finding the similar structure proposals from the datasets is presented in Figure 5. Additionally, it has been validated using real images having complex backgrounds and occlusions in Figure 6.

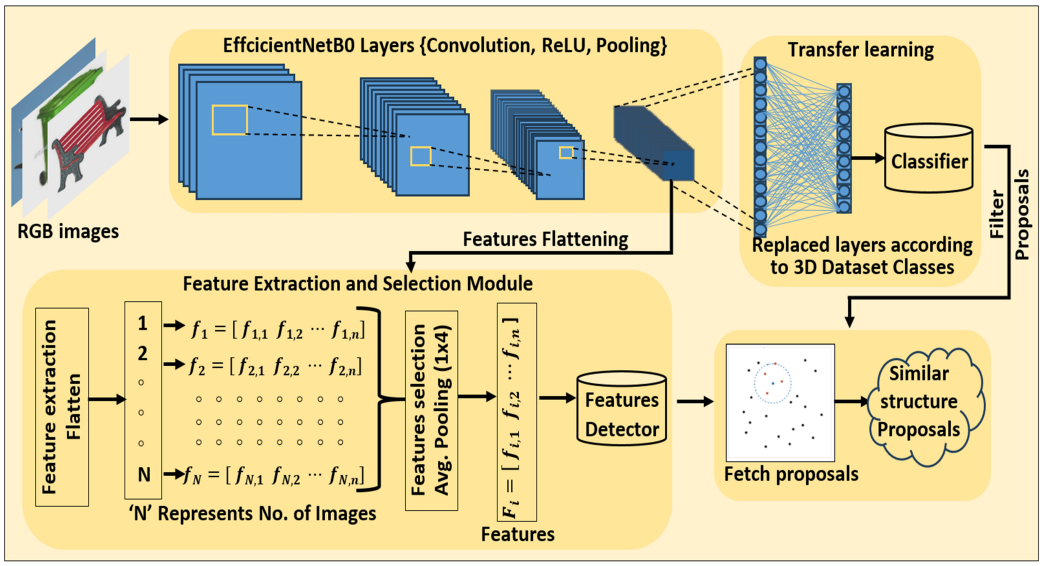


Figure 4. Human-inspired memory network (HIMNet) utilizes high-level feature extraction to search for structurally similar proposals, refining results through dual filtration based on object category and structure similarity. The geometries of the proposals are used as an initial guess in the learning-based network.

3.1.3. Time Complexity of HIMNet

The time complexity of the Human-Inspired Memory Network (HIMNet) is determined by its key components:

- Convolutional Layers: Each convolutional layer executes convolutions across the input image or feature maps. The time complexity for a single convolutional layer is $O(K^2 \times C_{\text{in}} \times C_{\text{out}} \times H \times W)$, where K is the kernel size (e.g., 3×3), C_{in} is the number

of input channels, C_{out} is the number of output channels, and H and W are the height and width of the input feature map, respectively.

- Batch Normalization: This layer normalizes the output of the convolutional layers to stabilize and accelerate training. Its time complexity is $O(C_{\text{out}} \times H \times W)$.
- ReLU Activation: The ReLU function introduces an element-wise non-linearity. The time complexity for ReLU is $O(C_{\text{out}} \times H \times W)$.
- Depthwise Separable Convolutions: Depthwise Separable Convolutions significantly reduce the computational cost compared to standard convolutions by decomposing the operation into a depthwise convolution, which filters each input channel separately, followed by a pointwise convolution that combines these outputs across channels. The time complexity for depthwise convolution is $O(K^2 \times C_{\text{in}} \times H \times W)$, and for pointwise convolution, it is $O(C_{\text{in}} \times C_{\text{out}} \times H \times W)$.
- Feature Comparison: After feature extraction, comparing features with those of other images using cosine similarity has a time complexity of $O(N \times d)$, where N is the number of images compared, and d is the dimensionality of the feature vectors.

The overall time complexity of HIMNet is approximated by summing the complexities across all layers and including the feature comparison, given by $O\left(\sum_{l=1}^{L}\left(K^2 \times C_{\text{in}}^l \times C_{\text{out}}^l \times H^l \times W^l\right) + (N \times d)\right)$, here, L represents the total number of layers, and the summation includes all convolutional, batch normalization, and activation layers. This expression encapsulates the computational cost of the feature extraction process as defined by the network architecture and the subsequent similarity comparisons.

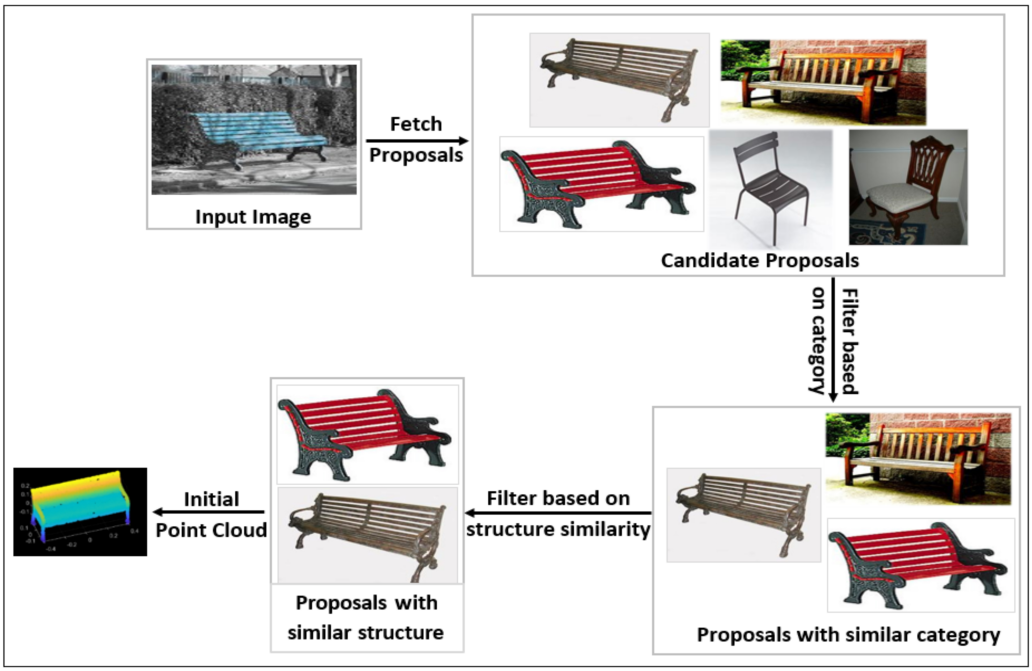


Figure 5. The figure shows the steps involved in HIMNet for the filtration of proposals and generating initial guess for the training of end-end architecture 3DRecNet.

3.2. Image Encoder

In this module, image high-level features are extracted using a pre-trained deep learning model, i.e., ResNet-18, with its classification layer frozen. The designed network consists of a convolutional layer at the beginning, followed by four residual blocks, each having two convolutional layers. These residual blocks facilitate the network in acquiring and preserving significant features. The module concludes with global average pooling and the flattening of the acquired features, which are then passed into the deep attention network, as explained in Section 3.4.

Time Complexity Analysis: Image Encoder Branch

Image Encoder Branch consists of the following layers, with each contributing to the overall time complexity:

- Convolutional Layers: Image Encoder branch has 17 convolutional layers (when including layers within residual blocks). The time complexity for these layers is $O\left(\sum_{i=1}^{17}\left(C_{\text{in}_i} \times K_i^2 \times W_i \times H_i \times C_{\text{out}_i}\right)\right)$, where C_{in_i} and C_{out_i} are the input and output channels for layer i, K_i is the kernel size, and $W_i \times H_i$ are the dimensions of the feature maps at layer i.
- Batch Normalization Layers: Each convolutional layer is typically followed by a batch normalization layer, contributing to the time complexity $O\left(\sum_{i=1}^{17}(C_{\text{out}_i} \times W_i \times H_i)\right)$.

Figure 6. This figure presents real-world images alongside their corresponding estimated 3D point clouds, illustrating the model's robustness in handling diverse challenges, including occlusions, environmental complexities, visual noise, and intricate backgrounds.

- ReLU Activation Layers: ReLU activation follows each batch normalization layer, with the same complexity: $O\left(\sum_{i=1}^{17}(C_{\text{out}_i} \times W_i \times H_i)\right)$.
- Max Pooling Layers: Image Encoder Branch has one max pooling layer after the initial convolutional layer, reducing the spatial dimensions: $O(C_2 \times W_2' \times H_2')$, where C_2 is the number of channels after the first pooling layer, and $W_2' \times H_2'$ are the reduced dimensions.

The total time complexity for the image branch can be expressed as: $O\left(\sum_{i=1}^{17}\left(C_{\text{in}_i} \times K_i^2 \times W_i \times H_i \times C_{\text{out}_i} + C_{\text{out}_i} \times W_i \times H_i\right) + C_2 \times W_2' \times H_2'\right)$.

3.3. 3D Encoding Branch

A Multi-Layer Perceptron (MLP) is utilized in this module to learn the geometry of the initial point cloud generated by the HIMNet network (based on similar structure proposals to the input image) with multiple layers of interconnected neurons. This network consists of a sequence of fully connected layers. It starts by flattening the input (initial point cloud sphere), and then passes through three hidden layers with shapes 4096, 1024, and 512 neurons, respectively. Each of these layers is followed by a Leaky ReLU activation function with a slope of 0.2, which introduces non-linearity to the network. The network layers are represented in the following Equation

$$X_P = \text{LeakyReLU}(\text{LeakyReLU}(\text{Flatten}(P) \cdot W_{E1} + b_1) \cdot W_{E2} + b_2) \cdot W_{E3} + b_3, \quad (6)$$

where W_{E1}, W_{E2}, W_{E3} are the weight matrices with dimensions (5000 $\times$ 3) $\times$ 4096, (4096 $\times$ 1024), and (1024 $\times$ 512), with b_1, b_2, b_3 are the biases on each layer, respectively. The time complexity of the 3D Encoding Branch is $O(W_{E1} + W_{E2} + W_{E3})$.

3.4. Dual Attention-Based Feature Fusion

The fusion layers in this module combine information from two branches: image encoding and 3D encoding branches. The fusion process starts with a linear layer merging the outputs of the two branches, reducing the dimensionality from 512 in each branch to 256. This fused representation is then further combined with the output of the image encoding branch through the second fusion layer, resulting in a representation of 128 dimensions. Finally, the third fusion layer combines this representation resulting in a fused representation of 512 dimensions. These fusion layers play a crucial role in capturing and combining relevant information from both the image and point cloud branches, creating a comprehensive feature representation for the model. Within each fusion layer, attention mechanisms are incorporated to assign high-importance scores to features from the image encoding branch. This generalizes the model towards the input image.The mathematical representation of the fusion layers discussed above is as follows

$$X_{fused} = [X_I, [X_I, [X_I, X_P] \cdot W_{f1} + b_{f1}] \cdot W_{f2} + b_{f2}] \cdot W_{f3} + b_{f3}, \quad (7)$$

where X_I represents the features of image-branch computed in Section 3.1, and X_P represents the features of the 3D-encoding branch computed in Section 3.2. W_{f1}, W_{f2}, W_{f3} are the weight matrices with dimension (512 + 512) $\times$ 256, (512 + 256) $\times$ 128, (512 + 128) $\times$ 512, with b_{f1}, b_{f2}, b_{f3} are the biases on each layer, respectively. The time complexity of the fusion layers are $O(W_{f1} + W_{f2} + W_{f3})$. The integration of fusion layers contributes to learning complex shapes and geometry, thereby alleviating the requirement for proposals within test images. This accelerates the generation of 3D models in real-time, as demonstrated by the visual representation in Figures 7 and 8 (optimized).

In the next subsection, the decoder branch is discussed, which transforms the fused and learned features into a 3D shape (Point Clouds).

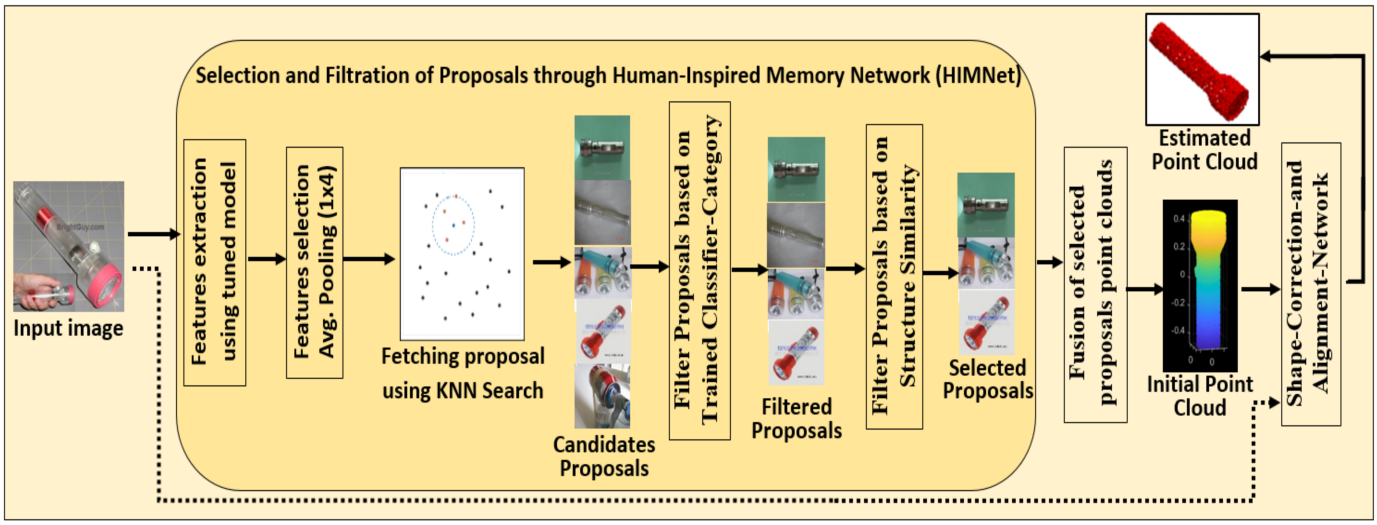


Figure 7. Sequential Steps for Estimating Point Cloud from Input Images: Feature Extraction, Selection, KNN-based Proposal Retrieval, Category Filtering, Structural Similarity Filtration, Generating Initial Point Cloud, which is further Refined by Shape Correction and Alignment Network (i.e., 3DRecNet) for Final 3D Point Cloud Estimation.

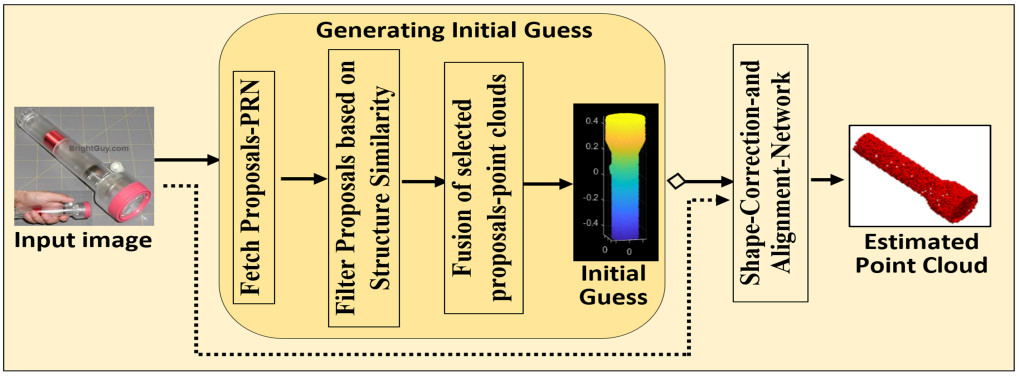


Figure 8. Optimized sequence for faster point cloud estimation from input images. Includes an optional initial guess and subsequent refinement via shape correction and alignment network for final 3D point cloud estimation.

3.5. Decoder Branch

The decoder branch employs a series of linear layers and Leaky ReLU activation in a step-by-step fashion, after the fusion process. This gradually increases the dimensions from 512 to 1024, 4096, and 8192, resulting in a final 3D shape with dimensions of 5000 × 3. The mathematical representation of the layers is represented in the equations

$$M = \text{LeakyReLU}\Big(\text{LeakyReLU}\Big(\text{LeakyReLU}\Big(X_{fused} \cdot W_{D1} + b_{D1}\Big) \cdot W_{D2} + b_{D2}\Big) \cdot W_{D3} + b_{D3}) \cdot W_{D4} + b_{D4}, \tag{8}$$

where M is the models output, W_{D1}, W_{D2}, W_{D3}, W_{D4} are the weight matrices with dimensions (512 × 1024), (1024 × 4096), (4096 × 8192), and (8192 × 5000) × 3 with b_{D1}, b_{D2}, b_{D3}, and b_{D4} are the biases on each layer, respectively. The time complexity can be represented as $O(W_{D1} + W_{D2} + W_{D3} + W_{D4})$.

3.6. Summary and Aggregated Time Complexity

The overall time complexity of the proposed architecture, 3DRecNet, can be expressed as follows:

$$O\Bigg(\sum_{l=1}^{L}\Big(K^2 \times C_{\text{in}}^{l} \times C_{\text{out}}^{l} \times H^{l} \times W^{l}\Big) + (N \times d) + \sum_{i=1}^{17}\Big(C_{\text{in}_i} \times K_i^2 \times W_i \times H_i \times C_{\text{out}_i}\Big) + \sum_{i=1}^{17}(C_{\text{out}_i} \times W_i \times H_i) + (C_2 \times W_2' \times H_2') + (W_{E1} + W_{E2} + W_{E3}) + \Big(W_{f1} + W_{f2} + W_{f3}\Big) + (W_{D1} + W_{D2} + W_{D3} + W_{D4})\Bigg),$$

where (W_{E1}, W_{E2}, W_{E3}) are the encoder branch weights matrices; (W_{f1}, W_{f2}, W_{f3}) are the fusion branch weights matrices; and (W_{D1}, W_{D2}, W_{D3}, W_{D4}) are the decoder branch weights matrices; K is the kernel size; C_{in}^{l} and C_{out}^{l} are the input and output channels at layer l; H^{l} and W^{l} are the height and width at layer l; N is the number of points; d is the dimensionality of the features; and C_{in_i}, C_{out_i}, K_i, W_i, H_i, C_2, W_2', and H_2' correspond to the respective input/output channels, kernel sizes, and spatial dimensions in other parts of the network.

The architectural modules discussed in the previous sections define the forward pass equations of the 3DRecNet neural network, enabling it to learn the shapes and geometries of complex and occluded objects. The model is inspired by recent approaches from both two-stage [26] and single-stage [6,25] networks, offering an improved 3D reconstruction architecture. As two-stage networks require extensive proposal searching and may not be efficient for real-time applications, single-stage networks are efficient but experience difficulties in learning complex shapes and geometries directly from images. 3DRecNet combines the strength of two-stage networks into a single-stage design.

The primary contribution of this paper lies in the design and integration of a human-inspired memory module (HIMNet) and an attention-based fusion module, which converts a two-stage network into a single stage. Several experiments has been conducted on real-world images as presented in Figure 6, designed to evaluate the algorithm's robustness in handling various types of occlusion, environmental challenges, visual noise, and complex backgrounds. The experiments cover diverse scenarios such as a kitten partially obscuring a laptop, a bicycle hidden behind tree branches, and children fully obscured on a bench. Additional test cases involve a soda can splash creating visual noise, a folding chair blending into a low-contrast background, and various environmental occlusions–such as a park bench surrounded by clutter, a silhouette of an airplane, overlapping children at a piano, a tennis player in motion, and a wheelchair in a sports field. Each scenario presents unique challenges for accurately capturing details. The results indicate that the proposed model effectively handles these complexities, producing precise 3D reconstructions.

Further details on backward propagation, gradient descent, ablation studies, and optimization parameters, including learning rate, batch size, and regularization, are discussed in the Section 4.

4. Experimentation and Results

This section provides insight into the dataset details utilized in the experiments, including discussions on experimentation settings. It further explores parameter optimization and the use of gradient loss functions during network training.

4.1. 3D Datasets

To assess the effectiveness of the method, experiments were conducted on the ObjectNet3D [15], ShapeNet [27], and Pix3D datasets [16]. All these datasets exhibit distinct data distributions. The ObjectNet3D dataset, as described in [15], comprises 90,127 images of objects in real-world scenes, with 44,500 3D CAD models organized into 100 categories.

This dataset proves valuable for tasks involving understanding object shapes and orientations in diverse environments. It offers a diverse set of object categories and a wide range of scenes, thereby enhancing the robustness and generalization capabilities of computer vision models. We utilized this dataset to validate the proposed architecture on complex and occluded images.

The second dataset used for the experiments is ShapeNet. It is a widely used dataset in computer vision and graphics research for understanding 3D shapes. It covers over 55 object categories, including furniture, vehicles, and animals, with more than 51,000 unique models. In our research experiments, we incorporate this dataset because of its large-scale and well-organized structure. Researchers focused on a subset of the dataset with 13 categories (Airplane, Bench, Cabinet, Car, Chair, Lamp, Monitor, Rifle, Sofa, Speaker, Table, Telephone, Vessel). They generated 2D scans from 3D CAD models, and this dataset is relatively easy to train due to its plain backgrounds.

Another dataset used for the experiments is Pix3D, a widely recognized benchmark for progressing research in understanding and reconstructing 3D objects in real-world settings. It includes a variety of scenes and objects typically placed indoors, with more than 1500 annotated images across 10 object categories such as furniture, electronics, and tools. In this dataset, the camera focused on objects within the images. We used this dataset in our experiments to confirm the algorithm's strength on real images.

4.2. Experimentation Settings

The model architecture employed for experimentation consists of two main branches: an image branch utilizing a pre-trained ResNet-18 network for feature extraction and a point cloud branch consisting of fully connected layers for processing 3D point cloud data. The image branch, initialized with pre-trained weights, employs adaptive average pooling for spatial information, followed by flattening and normalization, while the entire model utilizes Xavier uniform initialization for linear layers. The point cloud branch comprises three fully connected layers with leaky ReLU activation functions. After extracting features from both branches, fusion layers (linear transformations) combine features of both branches. The experimentation involves three fusion layers to progressively combine image and point cloud branch features.

During training, we utilized the L2 normalization for both image and point cloud features. The final merging branch (decoder) consists of a series of linear layers that create the 3D point cloud shape, converting fused features into the 3D model. The entire model is trained using backpropagation and optimized with default parameters using stochastic gradient descent (SGD). Including the initial point features helps the model learn the complexities in the geometry of objects' shapes. Without this integration, the model struggles to understand occluded objects and complex shapes, as confirmed through multiple images in Figure 9 of the dataset and the training loss functions in Figure 10. These figures show that when initial point cloud features (as memory of the network) are included, the model can handle complex object geometries and shapes.

As the proposed approach involves the initial point cloud (as an algorithmic memory), the final output relies on selecting good candidate proposals. To address this reliance, we undertook a comprehensive series of experiments, resulting in the integration of deeply fused attention layers (as discussed in detail in the research methodology Section 3.4). These layers reduce the dependency on proposals at test time and improve the algorithm's time complexity. Also, the model becomes sufficiently trained to retrieve the 3D shape directly from a single 2D image. To validate this claim, we provided incorrect estimations to the model alongside input images, demonstrating its robustness across inter-class and intra-class variations in Figure 11. Furthermore, extensive experiments were conducted to minimize the loss function across datasets and fine-tune all hyperparameters with various settings as discussed above. Details of the loss function used are available in the next section. The reduction of training loss on each dataset is presented in Figures 12 and 13.

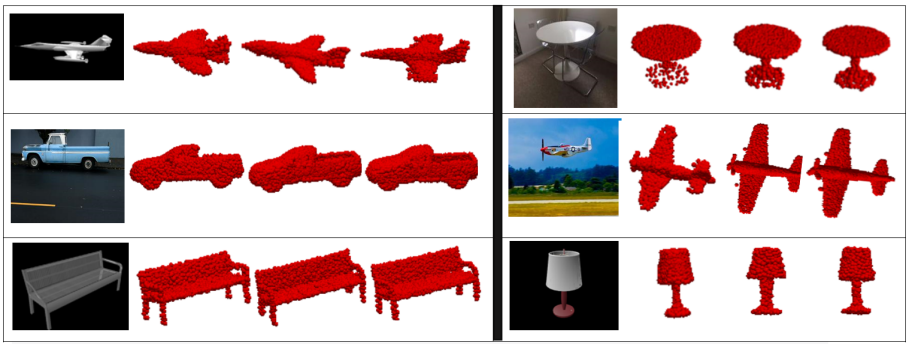


Figure 9. Experimental results presented in the image demonstrate the sequence from left to right: the input image, the model output without an initial guess, with an initial guess, and finally with enhanced test time performance and induction of dual attention mechanism.

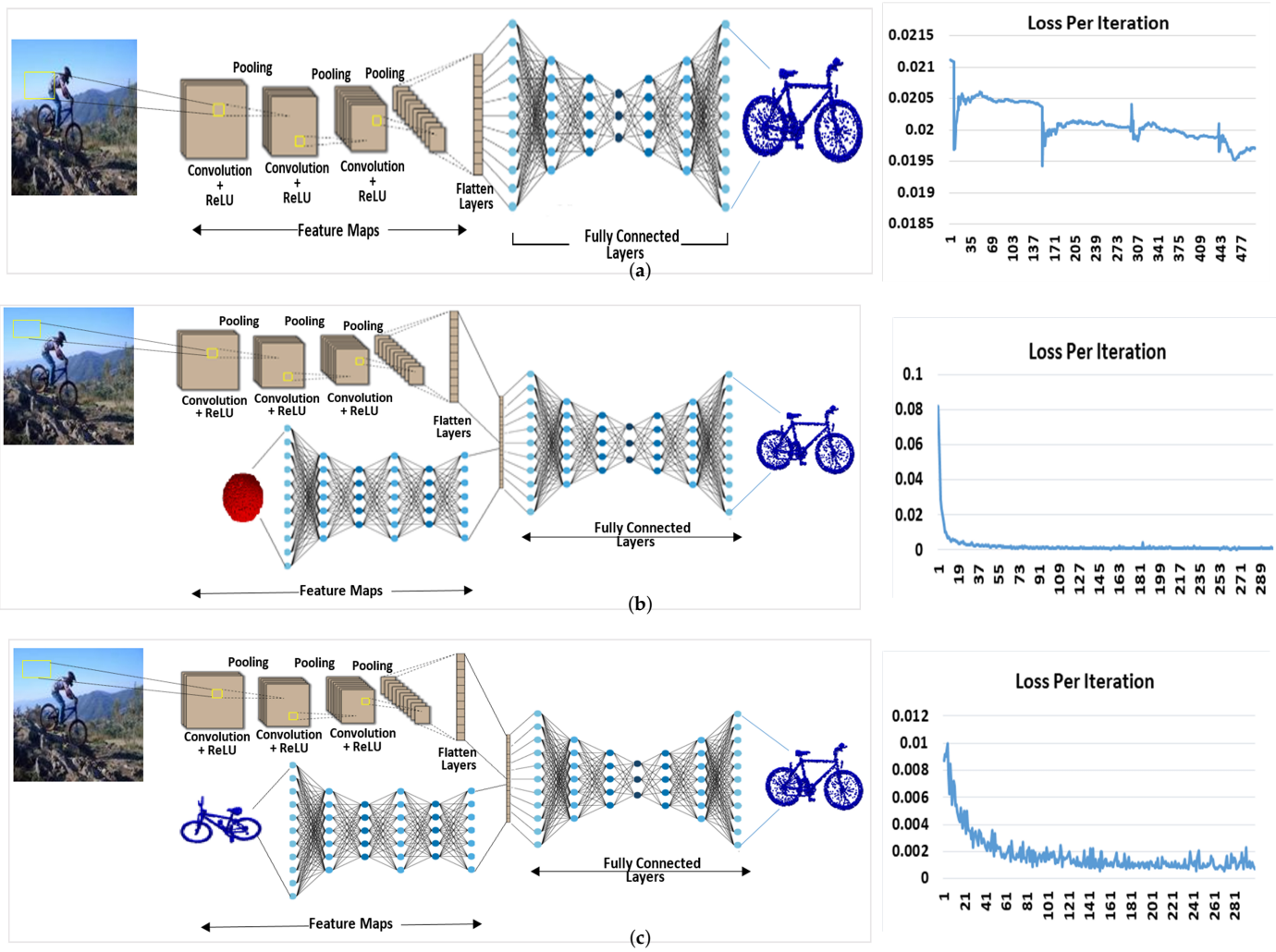


Figure 10. Sample design architectures: (**a**,**b**) without memory integration and (**c**) with memory integration along with their loss function behavior during training. Without memory, the loss function converges very slowly, requiring a large number of iterations, while with memory it reduces continuously. In case (**b**), the loss initially reduces exponentially due to the constant 3D sphere used as an initial guess. However, it fails to decrease in later epochs, resulting in vanishing gradients, making the model suitable only for synthetic datasets. (**a**) Single-stage architecture without initial guess (memory). (**b**) Two-stage architecture without initial guess (memory). (**c**) Two-stage architecture with memory integration (initial guess from human-inspired memory net HIMNet).

4.3. Ablation Study

This section analyzes the impact of the dual attention mechanism and human-inspired memory integration within the proposed 3DRecNet model.

4.3.1. Impact of Memory Integration on 3DRecNet Performance

The network's memory system resembles human memory: just as humans rely on past experiences to visualize the shape of objects, the model utilizes its memory of related objects to predict and design 3D shapes. Experimental results show that excluding memory integration results in slower convergence and less accurate reconstructions (Figures 9 and 10). Conversely, incorporating the memory net (HIMNet) significantly enhanced the model's performance, enabling the model to understand occluded and complex shapes better. Furthermore, the comparative analysis in Figure 10 shows that the model with memory integration begins to effectively learn shapes early in the training process, while the model without memory requires many more iterations to reach comparable performance, which results in higher computing resource consumption.

4.3.2. Handling Inter-Class and Intra-Class Variations

To validate the model's robustness, we evaluated its performance on both inter-class and intra-class variations. The results demonstrate the model's ability to correct wrong estimations and handle diverse object geometries effectively. Figure 11 illustrates the model's performance on inter-class variations, and across different classes, showing the model's ability to correct errors within a specific class and across varied classes. The visualizations highlight how the model consistently refines its predictions, regardless of the complexity of the objects involved.

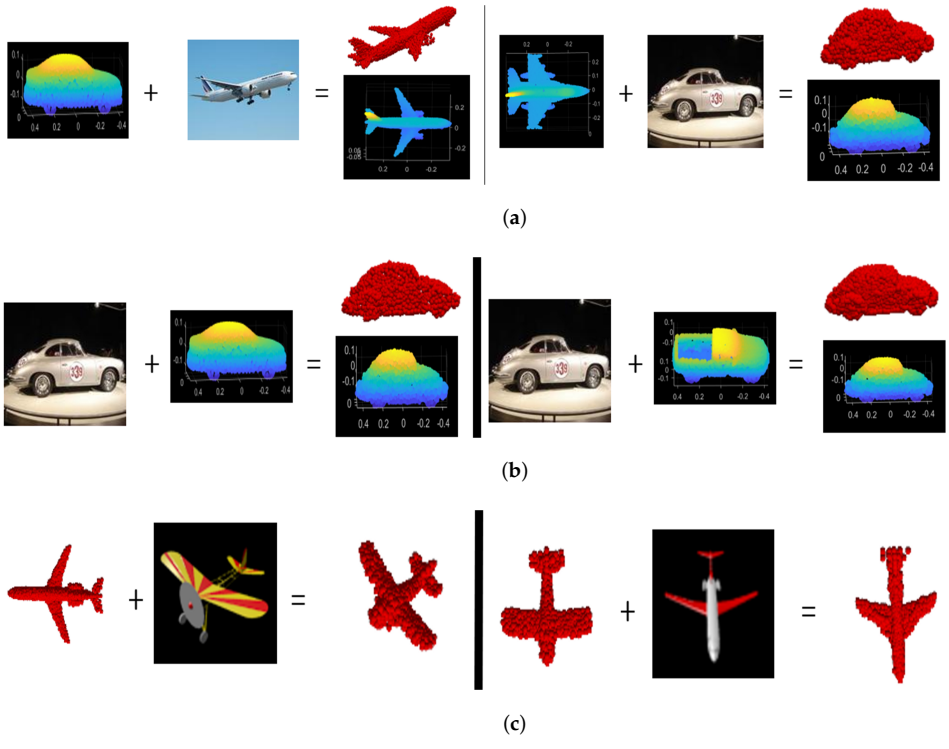
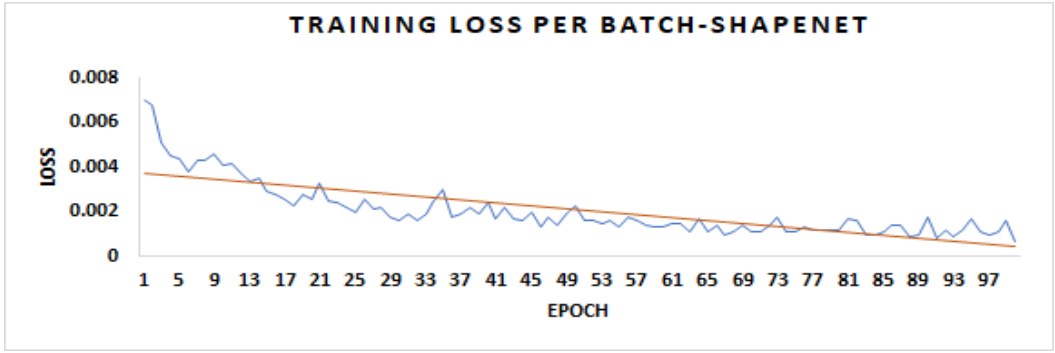
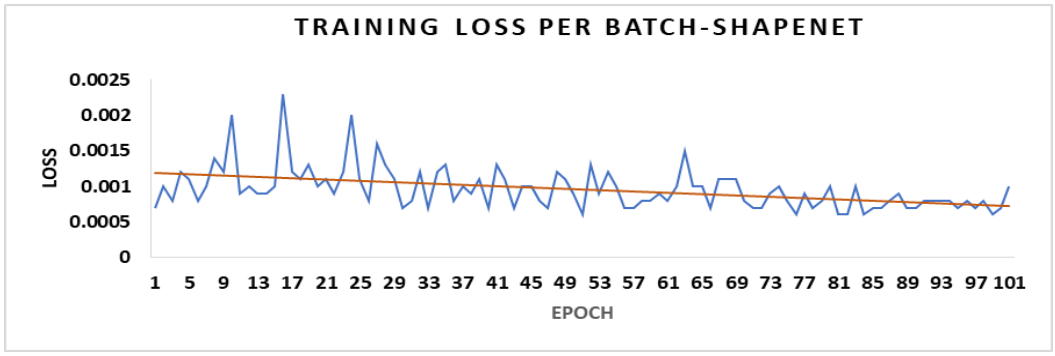


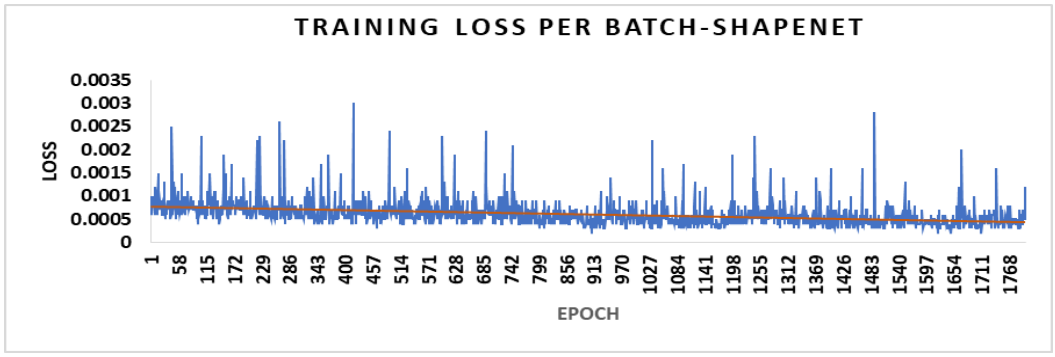
Figure 11. Model performance on inter and intra-class variations with incorrect initial guesses across two datasets: (**a**) ObjectNet3D with intra-class variations, (**b**) ObjectNet3D with inter-class variations and (**c**) ShapeNet with inter-class variations. The figures demonstrate the model's ability to correct wrong estimations in all the cases. (**a**) ObjectNet3D dataset with incorrect initial guesses, showing intra-class performance. (**b**) ObjectNet3D dataset with incorrect initial guesses, showing inter-class performance. (**c**) ShapeNet dataset with incorrect initial guesses, showing inter-class performance.



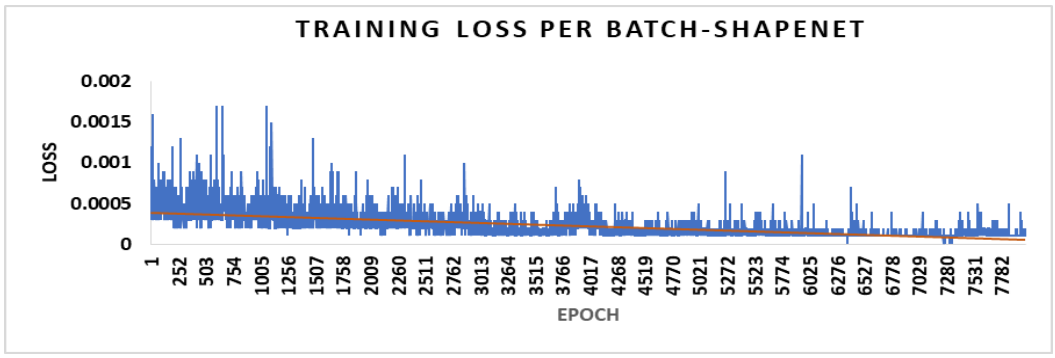
(**a**)



(**b**)

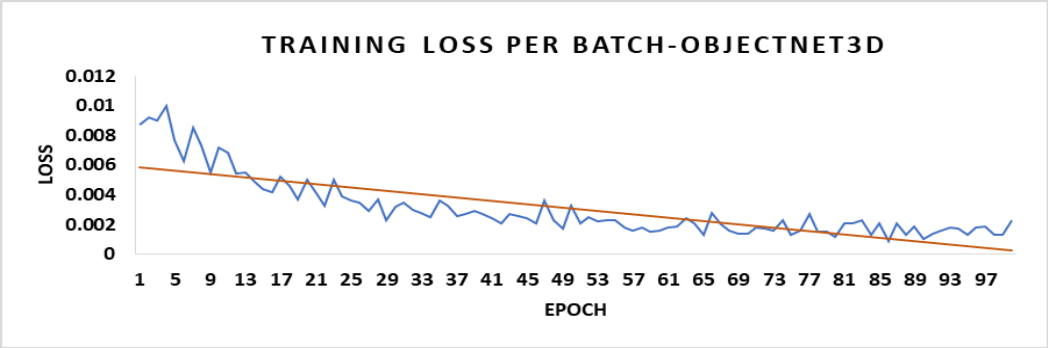


(**c**)

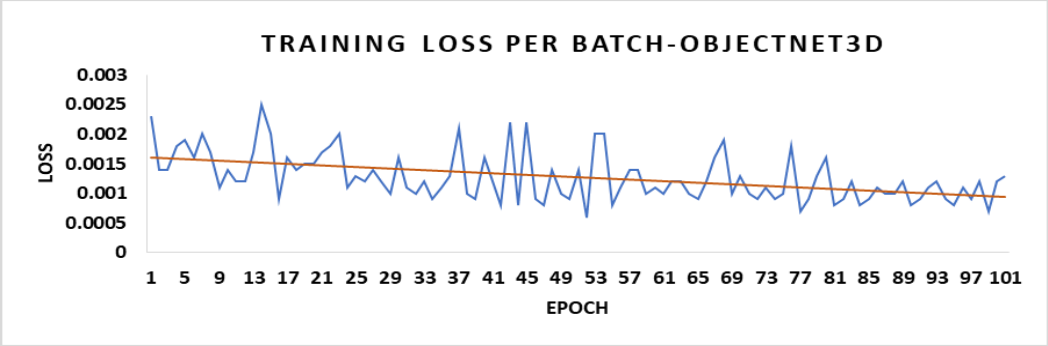


(**d**)

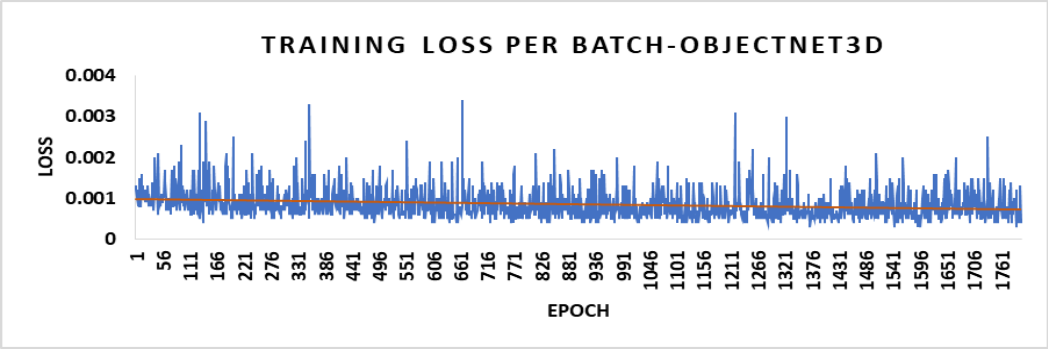
Figure 12. Training loss on the ShapeNet dataset over different epochs demonstrates model convergence, as indicated by the trend line. The loss is presented in four figures due to high deviation at the initial epochs and minimal reduction in the later epochs. (**a**) Training loss over the first 1–100 epochs. (**b**) Training loss for the 100–200 epochs. (**c**) Training loss for the 200–2000 epochs. (**d**) Training loss over the first 2000–10,000 epochs.



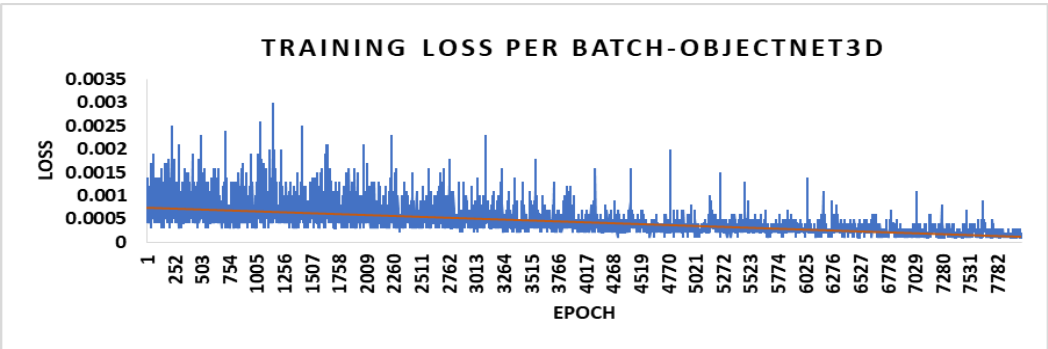
(**a**)



(**b**)



(**c**)



(**d**)

Figure 13. Training loss on the ObjectNet3D dataset over different epochs demonstrates model convergence, as indicated by the trend line. The loss is presented in four figures due to high deviation at the initial epochs and minimal reduction in the later epochs. (**a**) Training loss over the first 1–100 epochs. (**b**) Training loss for the 100–200 epochs. (**c**) Training loss for the 200–2000 epochs. (**d**) Training loss over the first 2000–10,000 epochs.

4.3.3. Impact of Attention-Based Fusion

Attention-based fusion is introduced in 3DRecNet to integrate image-encoded data with the initial point cloud from the Human-Inspired Memory (HIMNet); This integration allows the model to learn complex object shapes and geometries, enhancing accuracy and reducing feature loss. We introduced an attention-based mechanism to focus on the most relevant parts of the data while preserving long-range information, thereby improving overall performance. Within each fusion layer, attention mechanisms are used to assign high-importance scores to features from the image encoding branch. This adaptation improves efficiency by making the HIMNet module optional during inference, resulting in faster real-time 3D estimations since the HIMNet is used exclusively during training. Experimental results in Figure 9 showed that the attention mechanism performed best compared to models without an initial guess, without an attention mechanism, or both. Figure 14 illustrates the effects of the attention mechanism on reconstruction results. The first row shows outputs without the attention mechanism, while the second row displays outputs with it. The intermediate columns contain results derived from incorrect initial guess. Involvement of attention mechanism allows the network to generate accurate shapes even with incorrect initial guesses (point clouds).

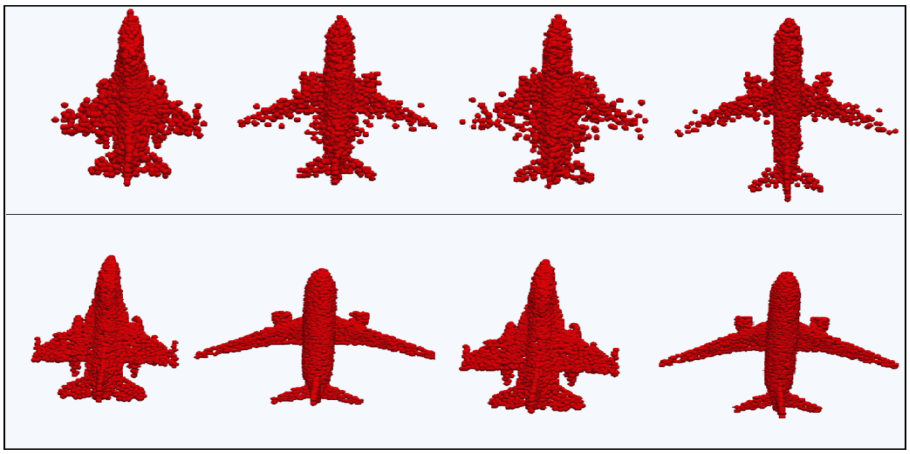


Figure 14. Images in the leftmost and rightmost columns represent correct samples, while the intermediate column displays incorrect samples. The top row shows results without the attention-based mechanism, and the bottom row demonstrates how the attention-based mechanism enables the network to produce accurate shapes, even in the presence of incorrect samples.

The ablation study confirms that the dual attention mechanism and human-inspired memory significantly enhance the 3DRecNet's performance, leading to faster convergence, reduced loss, and more accurate 3D reconstructions.

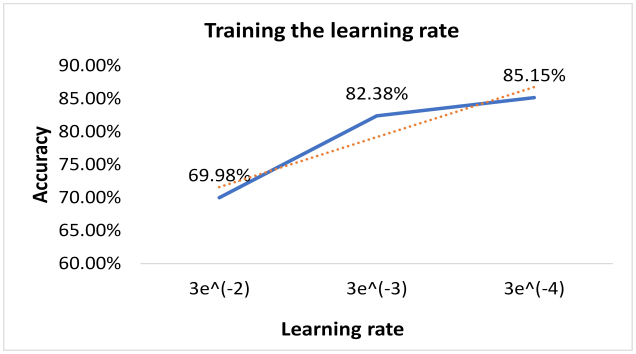
4.4. Parameter Optimization and Gradient Loss

Parameter optimization is a crucial step in updating a model's parameters during training. Chamfer distance (CD) is employed to measure the dissimilarity between estimated point clouds M and ground truth point clouds Y. It calculates the average minimum squared euclidean distance between each point in one cloud and its nearest neighbor in the other. For instance, for each point in set M, it finds the nearest point in the set Y and adds up the squared distances. It then repeats this process for each point in the set Y, finding the nearest point in set M and adding up those squared distances. Finally, the sum of both

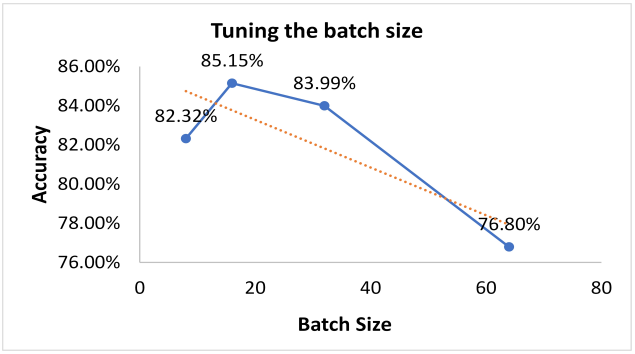
results gives the Chamfer Distance/Loss (CD). This loss function is commonly used in point cloud reconstruction tasks to guide model optimization for accurate reconstructions.

$$loss = \sum_{m \in M} \min_{y \in Y} \|m - y\|^2 + \sum_{y \in Y} \min_{m \in M} \|m - y\|^2,$$
$$\theta_{new} = \theta_{old} - \alpha \cdot \frac{\partial_{loss}}{\partial_{\theta}}, \tag{9}$$

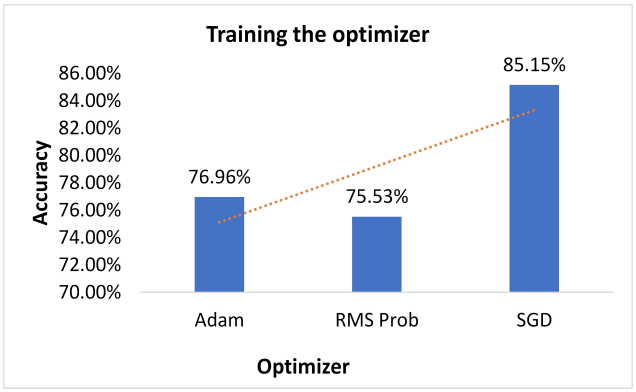
where θ_{new} is the updated parameter, θ_{old} is the current parameter and α is the learning rate (set to 0.0001 in this case). The parameter update equation employs the SGD optimizer a widely used optimization algorithm, that adapts learning rates based on historical gradients and efficiently converges with less hyperparameter tuning. All these hyperparameters are tuned through extensive experiments, as illustrated in Figure 15. Following this tuning process, the model is trained using these optimized hyperparameters, with the resulting training loss (on datasets: ObjectNet3D, ShapeNet, and Pix3D) depicted in Figures 12 and 13. Tables 1–5 compare our method with state-of-the-art approaches like RealPoint3D [31], PSGN [12], 3D-LMNet [32], 3D-VENet [33], 3D-FEGNet [34], 3D-CDRNet [26]. Results for other models were taken from their publications or cited from other works. The proposed method shows better results than most state-of-the-art algorithms, due to the involvement of a deep fusion attention mechanism over both the image and point cloud branches. The visual results of the generated 3D point clouds using the proposed architecture are presented in Figure 16, and the comparison of the proposed architecture with other methods is shown in Figure 17.



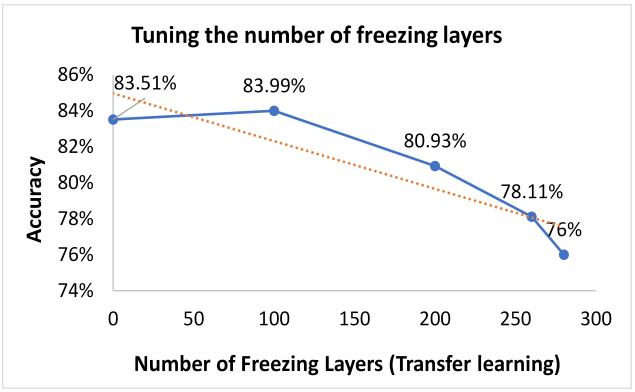
(**a**)



(**b**)



(**c**)



(**d**)

Figure 15. The different hyperparameters are tuned to improve the algorithm performance, including (**a**) learning rate of 0.0001, (**b**) batch size of 16, (**c**) optimizer Stochastic Gradient Descent, and (**d**) freezing 100 layers for transfer learning. (**a**) Tuning of the learning rate. (**b**) Tuning of the batch size. (**c**) Tuning of the optimizer for backpropagation and weight updates to minimize the loss function. (**d**) Tuning for the number of layers to freeze in the model during transfer learning.

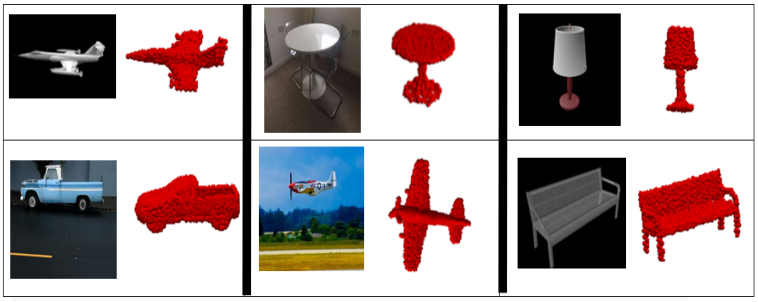


Figure 16. Evaluating model performance on diverse datasets with varying backgrounds.

Figure 17. Visualization results on the ShapeNet dataset. From left to right: input 2D images, 3D-LMNet [32], 3D-CDRNet [26], Proposed (Ours), and Ground Truth.

4.5. Time Complexity

The experiments were conducted on a workstation equipped with a GTX-1080 Ti GPU and configuration of PyTorch deep learning framework. To train 3DRecNet, the network was configured to run for 500 epochs, with multiple iterations per epoch and a batch size of 16, taking approximately nine days. Most benchmark studies did not report time complexity, and due to the unavailability of code, direct comparison was not possible. Among the relevant studies, RealPoint3D [31] reported a time complexity of approximately 10 s per 100 outputs using a system with multiple P100 GPUs, comparably similar time complexity to PSGN [12], while OGN [35] required about 160 s per 100 outputs.

In comparison, the proposed 3DRecNet achieved significantly better performance, taking only 9.24 s per 100 outputs, despite utilizing a system with lower specifications. Regardless of runtime efficiency, the time complexity, expressed using Big O notation and independent of machine specifications, is also discussed in the explanation of each module of research methodology section.

Table 1. Comparison of the 3D models generated by our proposed method on Pix3D dataset with state-of-the-art approaches using Chamfer Distance (CD). A lower value indicates superior results.

CD Comparison		**Methods**					
		PSGN [12]	**3D-LMNet [32]**	**3D-ARNet [36]**	**3D-ReconstNet [30]**	**3D-FEGNet [34]**	**Proposed**
Category	Chair	8.05	7.35	7.22	5.59	5.66	5.16
	Sofa	8.45	8.18	8.13	6.14	6.23	5.91
	Table	10.85	11.2	10.31	7.04	7.58	6.22

Table 2. Comparison of the 3D models generated by our proposed method on ObjectNet3D dataset with state-of-the-art approaches using Chamfer Distance (CD). A lower value indicates superior results.

CD Comparison		**Methods**			
		PSGN [12]	**Real-Point3D [31]**	**EGN [37]**	**Proposed**
Category	Sofa	2.0	1.95	1.82	0.98
	Aeroplane	1.0	0.79	0.77	0.41
	Bench	2.51	2.11	2.18	1.29
	Car	1.28	1.26	1.25	0.72
	Chair	2.38	2.13	1.97	1.22

Table 3. Comparison of the 3D models generated by our proposed method on Pix3D dataset with state-of-the-art approaches using Earth Mover's Distance (EMD). A lower value indicates superior results.

EMD Comparison		**Methods**					
		PSGN [12]	**3D-LMNet [32]**	**3D-ARNet [36]**	**3D-ReconstNet [30]**	**3D-FEGNet [34]**	**Proposed**
Category	Chair	12.55	9.14	7.94	5.99	8.24	6.15
	Sofa	9.16	7.22	6.69	5.02	6.77	4.79
	Table	15.16	12.73	10.42	7.60	11.40	7.43

Table 4. Comparison of the 3D models generated by our proposed method on the ShapeNet dataset with state-of-the-art approaches using Earth Mover's Distance (EMD). A lower value indicates superior results.

EMD Comparison		Methods					
		PSGN [12]	**3D-LMNet [32]**	**3D-VENet [33]**	**3D-FEGNet [34]**	**3D-CDRNet [26]**	**Proposed**
Category	Airplane	6.38	4.77	3.56	2.67	2.98	2.54
	Bench	5.88	4.99	4.09	3.75	3.65	2.92
	Cabinet	6.04	6.35	4.69	4.75	4.47	3.38
	Car	4.87	4.1	3.57	3.4	3.21	2.91
	Chair	9.63	8.02	6.11	4.52	4.65	4.17
	Lamp	16.17	15.8	9.97	6.11	7.26	5.41
	Monitor	7.59	7.13	5.63	4.88	4.65	4.11
	Rifle	8.48	6.08	4.06	2.91	3.31	2.17
	Sofa	7.42	5.65	4.8	4.56	4.04	3.71
	Speaker	8.7	9.15	6.78	6.24	5.73	4.22
	Table	8.4	7.82	6.1	4.62	4.32	3.66
	Telephone	5.07	5.43	3.61	3.39	3.08	2.67
	Vessel	6.18	5.68	4.59	4.09	3.59	2.88

Table 5. Comparison of the 3D models generated by our proposed method on the ShapeNet dataset with state-of-the-art approaches using Chamfer Distance (CD). A lower value indicates superior results.

CD Comparison		Methods					
		PSGN [12]	**3D-LMNet** [32]	**3D-VENet** [33]	**3D-FEGNet** [34]	**3D-CDRNet** [26]	**Proposed**
Category	Airplane	3.74	3.34	3.09	2.36	2.22	1.94
	Bench	4.63	4.55	4.26	3.6	3.38	3.01
	Cabinet	6.98	6.09	5.49	4.84	4.08	3.57
	Car	5.2	4.55	4.3	3.57	3.05	3.46
	Chair	6.39	6.41	5.76	4.35	4.17	4.09
	Lamp	6.33	7.1	6.07	5.13	5.07	4.36
	Monitor	6.15	6.4	5.76	4.67	4.08	4.02
	Rifle	2.91	2.75	2.67	2.45	1.91	1.35
	Sofa	6.98	5.85	5.34	4.56	4	3.89
	Speaker	8.75	8.1	7.28	6	5.05	4.45
	Table	6	6.05	5.46	4.42	3.8	3.14
	Telephone	4.56	4.63	4.2	3.5	2.53	2.19
	Vessel	4.38	4.37	4.22	3.75	2.95	2.42

5. Conclusions

This study presents 3DRecNet, an efficient fusion-based network for single-view 3D reconstruction. Inspired by recent advancements, 3DRecNet uses the strengths of both two-stage and single-stage networks to address the challenges of extensive proposal searching in two-stage networks and the direct learning of complex shapes from images in single-stage networks. By integrating a deep attention-based fusion network with a human-inspired memory network (HIMNet), 3DRecNet learns geometric parameters and shape details of objects accurately, resulting in faster and more precise 3D estimations. Extensive evaluation of benchmark datasets clearly shows that 3DRecNet outperforms existing methods, suggesting its potential for practical use. In addition, future research will focus on integrating part-level fusion to enhance the model's ability to handle unseen object deformations, making it more robust and adaptable for diverse applications. Additionally, exploring different attention mechanisms could further improve performance.

Author Contributions: M.A.S., A.B.S., L.Y. and Z.H. collaboratively conceived and designed the research project. M.A.S. proposed and implemented the methodology, conducted experiments, and drafted the research paper. A.B.S., Z.H. and L.Y. analyzed the results and contributed to improve the proposed architecture. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The datasets generated during and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Han, X.F.; Laga, H.; Bennamoun, M. Image-based 3D object reconstruction: State-of-the-art and trends in the deep learning era. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 1578–1604. [CrossRef] [PubMed]
2. Sra, M.; Garrido-Jurado, S.; Schmandt, C.; Maes, P. Procedurally generated virtual reality from 3D reconstructed physical space. In Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology, Munich, Germany, 2–4 November 2016; pp. 191–200.
3. Montefusco, L.B.; Lazzaro, D.; Papi, S.; Guerrini, C. A fast compressed sensing approach to 3D MR image reconstruction. *IEEE Trans. Med. Imaging* **2010**, *30*, 1064–1075. [CrossRef]
4. Pang, H.E.; Biljecki, F. 3D building reconstruction from single street view images using deep learning. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *112*, 102859. [CrossRef]
5. Yang, S.; Xu, M.; Xie, H.; Perry, S.; Xia, J. Single-view 3D object reconstruction from shape priors in memory. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 3152–3161.
6. Li, B.; Zhu, S.; Lu, Y. A single stage and single view 3D point cloud reconstruction network based on DetNet. *Sensors* **2022**, *22*, 8235. [CrossRef] [PubMed]
7. Schonberger, J.L.; Frahm, J.M. Structure-from-motion revisited. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4104–4113.
8. Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, I.; Leonard, J.J. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Trans. Robot.* **2016**, *32*, 1309–1332. [CrossRef]
9. Huang, Q.; Wang, H.; Koltun, V. Single-view reconstruction via joint analysis of image and shape collections. *ACM Trans. Graph. TOG* **2015**, *34*, 87–91. [CrossRef]
10. 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 Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 628–644.
11. Girdhar, R.; Fouhey, D.F.; Rodriguez, M.; Gupta, A. Learning a predictable and generative vector representation for objects. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 484–499.
12. Fan, H.; Su, H.; Guibas, L.J. A point set generation network for 3D object reconstruction from a single image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 605–613.
13. Feng, Y.; Wu, F.; Shao, X.; Wang, Y.; Zhou, X. Joint 3D face reconstruction and dense alignment with position map regression network. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 534–551.
14. Sinha, A.; Unmesh, A.; Huang, Q.; Ramani, K. Surfnet: Generating 3D shape surfaces using deep residual networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6040–6049.

15. Xiang, Y.; Kim, W.; Chen, W.; Ji, J.; Choy, C.; Su, H.; Mottaghi, R.; Guibas, L.; Savarese, S. Objectnet3D: A large scale database for 3D object recognition. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 160–176.
16. 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 Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2974–2983.
17. Richter, S.R.; Roth, S. Matryoshka networks: Predicting 3D geometry via nested shape layers. In Proceedings of the IEEE Conference on Computer vision And Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1936–1944.
18. Wu, J.; Wang, Y.; Xue, T.; Sun, X.; Freeman, B.; Tenenbaum, J. Marrnet: 3D shape reconstruction via 2.5 D sketches. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
19. Wu, J.; Zhang, C.; Zhang, X.; Zhang, Z.; Freeman, W.T.; Tenenbaum, J.B. Learning shape priors for single-view 3D completion and reconstruction. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 646–662.
20. Tahir, R.; Sargano, A.B.; Habib, Z. Voxel-based 3D object reconstruction from single 2D image using variational autoencoders. *Mathematics* **2021**, *9*, 2288. [CrossRef]
21. Xie, H.; Yao, H.; Sun, X.; Zhou, S.; Zhang, S. Pix2vox: Context-aware 3D reconstruction from single and multi-view images. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–3 November 2019; pp. 2690–2698.
22. Han, Z.; Qiao, G.; Liu, Y.S.; Zwicker, M. SeqXY2SeqZ: Structure learning for 3D shapes by sequentially predicting 1D occupancy segments from 2D coordinates. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; pp. 607–625.
23. Kniaz, V.V.; Knyaz, V.A.; Remondino, F.; Bordodymov, A.; Moshkantsev, P. Image-to-voxel model translation for 3D scene reconstruction and segmentation. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 105–124.
24. Peng, K.; Islam, R.; Quarles, J.; Desai, K. Tmvnet: Using transformers for multi-view voxel-based 3D reconstruction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 222–230.
25. Afifi, A.J.; Magnusson, J.; Soomro, T.A.; Hellwich, O. Pixel2Point: 3D object reconstruction from a single image using CNN and initial sphere. *IEEE Access* **2020**, *9*, 110–121. [CrossRef]
26. Tong, Y.; Chen, H.; Yang, N.; Menhas, M.I.; Ahmad, B. 3D-CDRNet: Retrieval-based dense point cloud reconstruction from a single image under complex background. *Displays* **2023**, *78*, 102438. [CrossRef]
27. Chang, A.X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al. Shapenet: An information-rich 3D model repository. *arXiv* **2015**, arXiv:1512.03012.
28. Pumarola, A.; Popov, S.; Moreno-Noguer, F.; Ferrari, V. C-flow: Conditional generative flow models for images and 3D point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 7949–7958.
29. Mueed Hafiz, A.; Alam Bhat, R.U.; Parah, S.A.; Hassaballah, M. SE-MD: A Single-encoder multiple-decoder deep network for point cloud generation from 2D images. *arXiv* **2021**, arXiv:2106.15325.
30. Li, B.; Zhang, Y.; Zhao, B.; Shao, H. 3D-ReConstnet: A single-view 3D-object point cloud reconstruction network. *IEEE Access* **2020**, *8*, 83782–83790. [CrossRef]
31. Xia, Y.; Wang, C.; Xu, Y.; Zang, Y.; Liu, W.; Li, J.; Stilla, U. RealPoint3D: Generating 3D point clouds from a single image of complex scenarios. *Remote Sens.* **2019**, *11*, 2644. [CrossRef]
32. Mandikal, P.; Navaneet, K.; Agarwal, M.; Babu, R.V. 3D-LMNet: Latent embedding matching for accurate and diverse 3D point cloud reconstruction from a single image. *arXiv* **2018**, arXiv:1807.07796.
33. Ping, G.; Esfahani, M.A.; Chen, J.; Wang, H. Visual enhancement of single-view 3D point cloud reconstruction. *Comput. Graph.* **2022**, *102*, 112–119. [CrossRef]
34. Wang, E.; Sun, H.; Wang, B.; Cao, Z.; Liu, Z. 3D-FEGNet: A feature enhanced point cloud generation network from a single image. *IET Comput. Vis.* **2023**, *17*, 98–110. [CrossRef]
35. Tatarchenko, M.; Dosovitskiy, A.; Brox, T. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2088–2096.
36. Chen, H.; Zuo, Y. 3D-ARNet: An accurate 3D point cloud reconstruction network from a single-image. *Multimed. Tools Appl.* **2022**, *81*, 12127–12140. [CrossRef]
37. Zhang, Y.; Liu, Z.; Liu, T.; Peng, B.; Li, X. RealPoint3D: An efficient generation network for 3D object reconstruction from a single image. *IEEE Access* **2019**, *7*, 57539–57549. [CrossRef]