Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi
Pamukkale University Journal of Engineering Sciences

Attention in the frequency domain: a complex phase-shifting spatial attention mechanism for 3D generative models

Frekans düzleminde dikkat: 3B üretken modeller için karmaşık bir faz kaydırmalı uzamsal dikkat mekanizması

Zafer Serin[1], Cihan Karakuzu[2], Uğur Yüzgeç[2]*

[1]Department of Web Design and Coding, Bilecik Seyh Edebali University Pazaryeri Vocational School, Bilecik, Türkiye.
zafer.serin@bilecik.edu.tr
[2]Department of Computer Engineering, Bilecik Seyh Edebali University Faculty of Engineering, Bilecik, Türkiye.
cihan.karakuzu@bilecik.edu.tr, ugur.yuzgec@bilecik.edu.tr

Received/Geliş Tarihi: 01.09.2025 Revision/Düzeltme Tarihi: 03.03.2026 doi: 10.65206/pajes.78703
Accepted/Kabul Tarihi: 03.04.2026 Research Article/Araştırma Makalesi

Abstract

Extracting 3D object geometry from a single 2D image remains a fundamental challenge in computer vision due to inherent information scarcity and ambiguity. To encourage the model to learn fundamental geometric structure rather than over-relying on texture and color cues, we employ a novel data augmentation strategy termed Silhouette Augmentation (SA). Our approach builds upon a voxel-based 3D-VAE-GAN architecture and introduces Complex Phase-Shifting Attention (CPSA), a frequency-domain attention mechanism that performs learned modulation of both phase and amplitude components of feature maps. The CPSA modules are integrated into the encoder after the second, third, and fourth convolutional layers to enhance mid- and high-level feature representations while preserving low-level inductive biases. Experiments are conducted on the widely adopted 13-category ShapeNet subset using an object-instance-level 80/20 train-test split, where all rendered views of each 3D object are assigned exclusively to a single split to prevent data leakage. Quantitative evaluation using the Intersection over Union (IoU) metric demonstrates consistent improvements over the baseline and spatial-domain attention mechanisms (SE and CBAM). When CPSA and SA are jointly applied, the proposed model achieves an average IoU of 0.577 across 13 categories. Multi-seed experiments further confirm the robustness of the improvement, reporting mean ± standard deviation values across independent runs (e.g., 0.6308 ± 0.0040 for Airplane, 0.4326 ± 0.0130 for Display, and 0.5270 ± 0.0023 for Table). Qualitative results show improved global structural coherence, particularly for categories with dominant global geometry. A computational analysis indicates that these gains are obtained with a moderate training-time increase, demonstrating the practical feasibility of frequency-domain attention in voxel-based 3D generative models.

Keywords: Complex Phase-Shifting Attention (CPSA), Single-View 3D Reconstruction, 3D-VAE-GAN, Frequency Domain Attention, Generative Adversarial Networks (GANs)

Öz

Tek bir 2D görüntüden 3D nesne geometrisini çıkarmak, içsel bilgi kıtlığı ve belirsizlik nedeniyle bilgisayar görüsünde temel bir zorluk olmaya devam etmektedir. Modelin doku ve renk ipuçlarına aşırı güvenmek yerine temel geometrik yapıyı öğrenmesini teşvik etmek için, Siluet Artırma (SA) adı verilen yeni bir veri artırma stratejisi kullanıyoruz. Yaklaşımımız, voksel tabanlı bir 3D-VAE-GAN mimarisi üzerine kuruludur ve özellik haritalarının hem faz hem de genlik bileşenlerinin öğrenilmiş modülasyonunu gerçekleştiren bir frekans alanı dikkat mekanizması olan Karmaşık Faz Kaydırma Dikkatini (CPSA) sunar. CPSA modülleri düşük seviyeli tümevarımsal eğilimleri korurken, orta ve yüksek seviyeli öznitelik temsillerini güçlendirmek amacıyla kodlayıcının ikinci, üçüncü ve dördüncü evrişimli katmanlarından sonra yapıya entegre edilmiştir. Deneyler, yaygın olarak kullanılan 13 kategorili ShapeNet alt kümesinde, nesne örneği düzeyinde 80/20 eğitim-test bölünmesi kullanılarak gerçekleştirilmiştir. Bu bölünmede, veri sızıntısını önlemek için her 3D nesnenin tüm işlenmiş görünümleri tek bir bölüme atanmıştır. Kesişim Üzerine Birleşim (IoU) metriği kullanılarak yapılan nicel değerlendirme, temel ve uzamsal alan dikkat mekanizmalarına (SE ve CBAM) göre tutarlı iyileştirmeler olduğunu göstermektedir. CPSA ve SA birlikte uygulandığında, önerilen model 13 kategoride ortalama 0,577 IoU değerine ulaşmaktadır. Çoklu tohum deneyleri, bağımsız çalışmalarda ortalama ± standart sapma değerlerini bildirerek iyileştirmenin sağlamlığını daha da teyit etmektedir (örneğin, Uçak için 0,6308 ± 0,0040, Ekran için 0,4326 ± 0,0130 ve Masa için 0,5270 ± 0,0023). Nitel sonuçlar, özellikle baskın küresel geometriye sahip kategorilerde, küresel yapısal tutarlılığın iyileştiğini göstermektedir. Hesaplamalı bir analiz, bu kazanımların eğitim süresinin makul bir artışıyla elde edildiğini göstererek, voksel tabanlı 3D üretken modellerde frekans alanı dikkatinin pratik uygulanabilirliğini ortaya koymaktadır.

Anahtar kelimeler: Karmaşık Faz Kaydırmalı Dikkat, Tek Görüntüden 3B Yeniden Yapılandırma, 3B-VAE-GAN, Frekans Düzlemi Dikkati, Çekişmeli Üretici Ağlar

1 Introduction

In recent years, rapid development has been observed in fields such as Artificial Intelligence (AI) and Deep Learning (DL) [1]. As a result, both discriminative and generative architectures benefit from these developments, leading to the emergence of new and more successful neural network architectures [2-3]. AI architectures have also begun to be used in subfields of the entertainment industry, such as video games, virtual reality (VR), and augmented reality (AR), which serve as auxiliary tools for those working in design and modeling [4]. These developments, also a part of the computer vision and computer graphics fields, have caused a gradual shift in emphasis from discriminative architectures to generative architectures [2].

Professionals in modeling and design typically create three-dimensional (3D) objects in digital environments. This process

*Corresponding author/Yazışılan Yazar

often begins with either a hand-drawn sketch or, more recently, a single photograph [5]. In either case, a two-dimensional (2D) draft is required before transitioning to a 3D environment. This approach not only provides a preliminary study of the object in a 2D space but also helps to form a foundational concept before reaching the final result [6]. This practice is particularly evident in industries like video games, where nearly every high-budget project is based on 2D concept designs, for which dedicated artbooks are often published.

The representation of a 3D object in a digital environment is a challenging and complex process. Designers and researchers in this field primarily utilize three distinct methods for 3D object representation: voxel, point cloud, and mesh [7]. Voxel representation defines a 3D object using a grid of volumetric pixels or voxels. This method, which is based on whether a point in the 3D grid is occupied, inherently contains volumetric information. Although this is an indispensable advantage for problems requiring volumetric data, it also incurs significant memory costs. Due to its high compatibility with standard neural network (NN) architectures like [8-9] Convolutional Neural Networks (CNNs), voxel representation is the method adopted in this study. In contrast, point cloud representation creates an object by storing only the coordinates of its surface points. Although this is memory-efficient, it lacks volumetric information, and its unordered nature makes it incompatible with standard NN architectures [10-11]. The mesh representation defines the surface of the object via vertices and faces (typically triangles) that connect them. Although arguably the most popular method among designers, mesh representation also offers limited volumetric information and requires specialized approaches for use with neural networks [12]. Figure 1 illustrates a representation of the same cat object using these three methods.

a b c

Figure 1. Display of the same cat object in a 3D environment using three different object representation methods. (a) the voxel, (b) point cloud, (c) mesh representations.

Generative architectures, such as the Variational Autoencoder (VAE) and Generative Adversarial Network (GAN), which are part of unsupervised learning, have been significantly developed in recent years [2-3]. Whereas the VAE architecture is based on learning a probabilistic representation of the input and generating new examples from this representation, the GAN architecture is based on generating new data that resemble real data. Recent advancements include powerful diffusion models that generate high-quality data by iteratively denoising a random signal [13]. Architectures such as 3D-VAE-GAN aim to combine VAE and GAN architectures and combine the advantages of both [4]. The 3D-VAE-GAN architecture, which enables the production of 3D objects from 2D inputs, can be described as highly advanced. This architecture, consisting of three neural networks: Encoder, Generator, and Discriminator, maps 2D input to the latent space using the encoder. This mapped data is fed into the generator as input, and production takes place. The discriminator attempts to distinguish between fake objects produced by the generator and real data already present in the dataset [4].

One limitation of the 3D-VAE-GAN architecture, as well as many standard models, is the lack of explicit modeling of long-range pixel relationships within the 2D input. While advanced architectures such as CNN can successfully learn local relationships, they may fall short in learning global relationships [14]. To address this, a mechanism known as the attention mechanism has been proposed in the literature, which identifies the relationships between pixels and other pixels [15]. Attention mechanisms such as SE and CBAM enable learning this relationship in the spatial domain [16-17]. An image can be expressed in several ways, such as in the spatial, frequency, and temporal domains. The spatial domain, where an image is represented by pixels and is the primary domain that comes to mind when thinking of an image today, is referred to as the spatial domain. While the spatial domain is sufficient in many aspects, it may be inadequate for problems such as compression, noise reduction, edge detection, and filtering. Transitioning to the frequency domain is known to be more efficient in such problems [18]. By applying the Fourier transform to an image, it can be decomposed into its frequency components, thereby enabling transition to the frequency domain. Another shortcoming of standard architectures is the data. Data, which is one of the most fundamental requirements for AI architectures to be successful, is one of the main reasons for entering a period referred to in the literature as the "AI Winter" for AI [19]. While data production continues to increase, the more data there is, the better the performance. Therefore, data augmentation techniques are used in current architectures, and their effects on performance have been demonstrated [20]. In addition to traditional methods, recent advancements include techniques such as Mixup, CutMix, and RandAugment, which further enhance the robustness [21-23]. Data augmentation is also of vital importance in problems such as 3D object production, because data are quite limited in 3D environments. While datasets such as ShapeNet, ModelNet, and Pascal3D contain 3D objects, they are known to be quite limited in number and large in size compared to 2D data [8-24-25]. Therefore, when producing 3D objects, it is important to extract as much information as possible from the available 2D data to improve the generalization performance.

In recent years, implicit, neural field, and diffusion-based approaches have significantly advanced the state of single-view 3D reconstruction. However, these paradigms often involve fundamentally different representation assumptions and training pipelines. In this study, rather than introducing a new representation paradigm, we deliberately adopt a controlled voxel-based generative framework to isolate and evaluate the effect of frequency-domain attention on encoder representations. The 3D-VAE-GAN backbone is therefore selected as a stable and widely adopted benchmark architecture that allows systematic comparison of attention modules under consistent experimental conditions.

1.1 Related works

This section presents studies related to the CPSA mechanism and 3D object generation that we recommend. In general, these studies are important for laying the foundation for the attention mechanism and consisting of generative-based architectures. Additionally, while relatively older studies on 3D object generation are based on taking specific parts of objects in the dataset and combining them, the studies presented in this section differ from the standard methods in that they are NN-based, learn probabilistic distributions, and compare them with real objects. The studies used for comparison and those

highlighting the importance of data augmentation are also included in this section.

The work proposed by Funkhouser et al. in 2004 [26] is considered one of the foundational works in the field of 3D object generation. At that time, deep neural networks (DNNs) and modern generative architectures had not yet been developed. Therefore, the proposed approach creates a database by breaking the 3D objects into meaningful parts. When a user wants to design a new object, the system selects appropriate parts from the database and assembles them. This study, which laid the foundation for part-based modeling, is important as a generative method, even if it does not produce entirely new objects.

In 2014, Goodfellow et al. [2] proposed a GAN architecture. A generative architecture, GAN, is one of the most important tasks in the field of unsupervised learning. Based on the principle of zero-sum games, as expressed in game theory, GANs rely on two neural networks that compete with one another. At the end of the competition, two highly advanced neural networks were obtained, and both generation and classification were achieved through competition.

VAE architecture was proposed by Kingma and colleagues in 2014 [3]. The VAE is an important milestone for generative architectures and unsupervised learning methods, enabling the probabilistic modeling of data distributions and the creation of a regular latent space. In this respect, it differs from a standard Autoencoder (AE) architecture. While the standard AE architecture proceeds through a single latent vector, the VAE architecture enables the modeling of probability distributions using the mean and log-variance values of the input in the latent space. This prevents gaps or meaningless regions from occurring in latent space.

In 2015, Wu et al. [8] proposed a 3D ShapeNet study. As one of the first studies to use voxel-based representation and probabilistic distribution to represent 3D objects, this study has been recognized in the literature as one of the pioneering studies in the fields of 3D object recognition and completion. This study, which fundamentally uses Deep Belief Networks, has also contributed one of the most fundamental and important datasets for 3D object problems, known as ShapeNet, to the literature, which we also used in our study.

Wu and colleagues proposed the 3D-VAE-GAN architecture in 2016 [4]. This work stands out as one of the fundamental studies that combines the advantages of the VAE and GAN architectures and presents them under a single framework. Based on the principle of generating 3D objects from 2D inputs, this study has become one of the pioneering works in this field and has been one of the approaches that inspired our work. The same study also proposed a 3D-GAN architecture that focused entirely on 3D object generation. Although the generated shapes may lack fine detail and structural consistency, this study remains one of the foundational works in the field.

In 2016, Choy et al. [5] proposed a 3D Recurrent Reconstruction Neural Network (3D-R2N2) architecture. This architecture enables voxel-based three-dimensional (3D) object generation from single or multiple images. The 3D-R2N2 architecture, which consists of Long-Short-Term Memory (LSTM) cells distributed in 3D space and can store the 3D representation of the relevant object in memory thanks to these cells, distinguishes itself from other studies in this regard. In this study, evaluations were performed on 13 object categories in the ShapeNet dataset. 3D-R2N2 is important in terms of contributing to this subset of the ShapeNet dataset, which was used in our study and many other studies. Additionally, a 3D-R2N2 study is important in terms of adopting IoU, one of the basic performance evaluation metrics in 3D object generation, as a standard.

Hu and his colleagues proposed the SE Networks in 2017 [16]. This is based on the fact that while classic CNN architectures can successfully learn spatial relationships, they generally ignore the relationships between channels. The proposed approach enables compression by expressing channels as a single value, and then excitation by extracting an importance score from these values using fully connected layers. The original feature map is multiplied by these importance scores, enabling the architecture to learn the relationships between channels and determine the channel to prioritize. This study is significant because it uses channel-based attention.

In 2017, DeVries et al. [27] proposed the "CutOut" data augmentation technique. This method is based on the principle of randomly selecting a rectangular area in an image and removing it from the image. This forced the model to generalize and learn. Tests and evaluations have shown that this method is successful.

Wang et al. proposed a Pixel2Mesh study [29] in 2018. This study is important in that it can generate a 3D object from a single 2D image using a mesh representation. It is based on a Graph Convolutional Network (GCN) [28] that starts with an ellipsoidal structure and is gradually optimized. It extracts the features of the 2D input using a CNN-like structure, and maps these features to each point on the mesh. Thus, the 2D features were mapped to the 3D part. Learning is guided by information from the CNN component [29-30]. In our study, voxel representation was preferred over mesh representation because of the absence of volumetric information loss and its full compatibility with traditional convolution operations.

In 2018, Zhou et al. [9] proposed the VoxelNet method. This work focuses on the sparse and irregular structure of 3D data obtained from LiDAR systems, which makes it difficult to directly process these data using deep neural networks. The proposed method divides point cloud data into 3D voxels and processes the points within each voxel to create a feature map. Convolution operations were applied to this feature map to estimate object locations and classes. VoxelNet is important because it allows LiDAR data to learn directly from raw data without requiring manually designed features.

In 2018, Woo et al. [17] proposed the CBAM, which combines channel-based and spatial attention. CBAM utilizes channel attention to identify important channels (*what*) in an image, and spatial attention to determine the location of important pixels (*where*). The mechanism first applies an SE-like channel attention module. Subsequently, spatial attention was computed from this refined feature map, and by applying both average and max pooling across the channel dimension, two 2D maps were generated. These maps were concatenated and processed using a single convolutional layer to create a final spatial attention mask. This mask was then multiplied by the feature map to highlight the salient regions. This two-stage approach is important because it learns both what and where to focus.

Zhong and his colleagues proposed the RE data augmentation technique in 2020 [31]. This technique covers or modifies a random area of the input image into a rectangular shape. When performing learning, it is necessary to make an accurate prediction and generalization from the remaining parts, despite not seeing this part. This prevents the architecture from

memorizing and ensures that it achieves successful results in the test part. The difference between this method and the CutOut method is that the size and ratio of the hidden area can also be random. In addition, in this method, the hidden area can be filled with random pixels or average values.

When reviewing studies in the literature, an interest in the 3D field is clearly evident. The importance of attention mechanisms, potential of generative architectures, effects of data augmentation and object representation in 3D environments on the results, and innovations they bring are clear. However, the fact that related studies have focused on topics such as learning in the spatial domain and attention in the spatial domain prevents them from benefiting from the vast pool of information in the frequency domain, making it difficult to learn this information. This situation can create limitations, particularly when capturing the overall structure and patterns of an object. The proposed CPSA offers a precise solution at this point, enabling learning by obtaining global information from the frequency domain.

In recent years, single-view 3D reconstruction has evolved beyond voxel-based generative models toward implicit and neural field representations [30,32,33]. Function-based approaches such as Occupancy Networks and related continuous representation methods model 3D geometry as a learned implicit function in continuous space, enabling higher fidelity reconstructions compared to fixed-resolution voxel grids [30,32]. Similarly, Neural Radiance Field (NeRF) inspired methods have introduced volumetric rendering and coordinate-based representations for reconstructing geometry and appearance from images [33,34]. These approaches emphasize continuous spatial modeling and high-resolution detail recovery.

More recently, diffusion-based generative models have also been adapted for 3D shape synthesis and single-image 3D reconstruction [35,36]. By iteratively refining noisy representations, diffusion frameworks have demonstrated strong generative capability and improved realism in 3D outputs. However, such models often involve substantial computational overhead and complex training pipelines.

Parallel to these representation-level advancements, global-context modeling strategies have been explored within convolutional architectures. Non-local neural networks and transformer-based encoders explicitly model long-range spatial dependencies using self-attention mechanisms [14,15]. Additionally, Fourier feature encoding techniques introduce frequency components into spatial representations to enhance positional awareness and structural modeling [37].

In contrast to transformer-based global modeling, which computes pairwise spatial self-attention and often replaces or heavily modifies convolutional backbones, the proposed CPSA introduces global interaction implicitly through learned complex-valued modulation of phase and amplitude components in the frequency domain. Rather than performing token-to-token attention in the spatial domain or merely injecting static Fourier encodings, CPSA adaptively modulates spectral representations and reconstructs a spatial attention mask via inverse Fourier transformation. This phase-aware spectral attention mechanism enables global structural reasoning while preserving convolutional inductive bias, maintaining architectural simplicity, and avoiding quadratic self-attention complexity.

Therefore, the objective of this study is not to compete directly with implicit, diffusion-based, or transformer-replacement paradigms, but to investigate whether frequency-domain attention can systematically strengthen encoder representations within a controlled voxel-based generative framework.

Table 1. List of Abbreviations.

Abbreviation	Full Form	Abbreviation	Full Form
AI	Artificial Intelligence	IoU	Intersection over Union
DL	Deep Learning	KL	Kullback–Leibler
CNN	Convolutional Neural Network	GT	Ground Truth
VAE	Variational Autoencoder	FFT	Fast Fourier Transform
GAN	Generative Adversarial Network	IFFT	Inverse Fast Fourier Transform
CPSA	Complex Phase-Shifting Attention	RGBA	Red-Green-Blue-Alpha
CA	Channel Attention	ReLU	Rectified Linear Unit
CFAM	Complex Frequency Attention Mask	BatchNorm	Batch Normalization
SA	Silhouette Augmentation	NeRF	Neural Radiance Field
SDF	Signed Distance Function	VRAM	Video Random Access Memory

1.2 Contributions

The main contributions of this study can be summarized as follows:

The primary contribution of this work is the proposed Complex Phase-Shifting Attention (CPSA) mechanism. Silhouette Augmentation (SA) is introduced as a complementary training strategy to improve geometric generalization; however, the core methodological novelty lies in the frequency-domain attention design.

1. We propose Complex Phase-Shifting Attention (CPSA), a frequency-domain attention mechanism that differs fundamentally from conventional FFT-based feature augmentation. Unlike approaches that simply concatenate Fourier features or apply frequency transforms as auxiliary inputs, CPSA performs learned complex-valued modulation of both magnitude and phase components to generate a spatial attention mask. By explicitly manipulating phase information, CPSA enables global structural interaction across the entire feature map, which

cannot be achieved through local convolution or static Fourier feature encoding.

2. We demonstrate that the integration of CPSA into the encoder, under the SA training protocol, enhances global structure capture in single-view 3D reconstruction, as evidenced by consistent quantitative improvements in IoU (average 0.577) and significant relative gains in structurally global categories such as Display (+11.9%) and Airplane (+6.19%), supported by qualitative visual reconstructions showing improved structural coherence.
3. The introduction and evaluation of Silhouette Augmentation (SA), a specialized data augmentation method designed to improve the robustness and generalization capabilities of single-view-to-3D object generation models by forcing reliance on geometric silhouettes and reducing texture/color dependencies.

The remainder of this paper is organized as follows. Section 2, Materials and Methods, provides a detailed description of the dataset, baseline 3D-VAE-GAN architecture, and our proposed CPSA mechanism, including its architectural integration and experimental setup. Section 3 presents a comprehensive evaluation of the proposed method through quantitative, qualitative, and time-based analyses. Finally, Section 4 summarizes the findings and discusses future research directions.

2 Materials and methods

This section details the methodology and experimental setup used in this study. First, we introduced the ShapeNet dataset used in our experiments and the specific subset prepared for this study. Next, we present the baseline architecture, 3D-VAE-GAN, upon which the proposed approach was built. Building on this foundation, we elaborate on our main contribution, the CPSA mechanism, and its integration into architecture. Finally, we outline the experimental framework used to evaluate our model, including implementation details, benchmarking methods (SE and CBAM), analysis of the data augmentation (SA) we proposed, and the primary evaluation metric, IoU.

2.1 Dataset

In this study, a subset of the ShapeNet dataset [24] was used to obtain 2D and 3D data. The ShapeNet dataset comprises of 55 distinct object categories. This dataset provides 2D images of objects from various viewpoints as well as 3D representations, which can be converted into voxels if needed. The proposed method employs a subset of the ShapeNet dataset containing 13 different objects. The selection of these 13 objects is based on their frequent use in many state-of-the-art (SOTA) studies in the literature [5].

The distributions of the training and test samples for the objects used in this study are shown in Figure 2. The horizontal stacked bars illustrate the split between the training (light blue) and test (dark blue) sets for each of the 13 objects. The numbers inside the light blue segments indicate the number of samples in the training set. For clarity, the numbers corresponding to the test set are placed to the right of each bar and color-coded (dark blue) to match their respective segments.

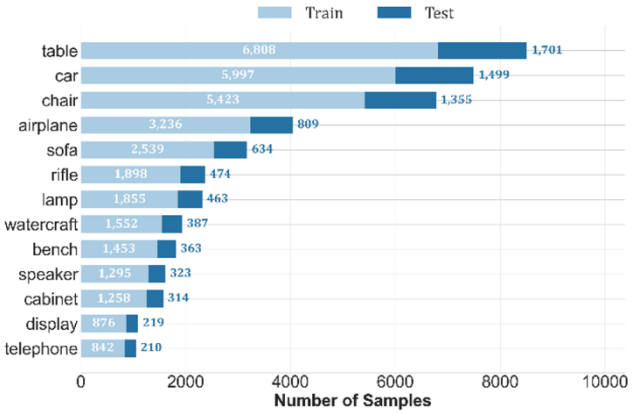


Figure 2. Distribution of samples per object in the ShapeNet dataset.

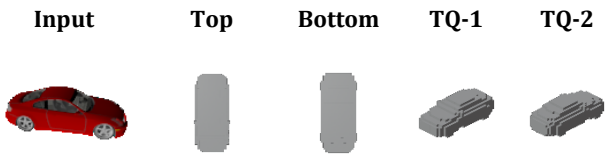
We utilized a large-scale subset of the ShapeNetCore v1 dataset, following the evaluation protocol established by Choy et al. (3D-R2N2) [5]. This subset consists of 13 major object categories (Synset IDs: Airplane 02691156, Bench 02828884, Cabinet 02933112, Car 02958343, Chair 03001627, Display 03211117, Lamp 03636649, Loudspeaker 03691459, Rifle 04090263, Sofa 04256512, Table 04379242, Telephone 04401088, and Watercraft 04530566).

For 2D views, we used pre-rendered images (137x137 pixels) provided with the ShapeNet subset, which include 4 channels (RGBA) to capture silhouette information. The 3D ground truth models were voxelized into a 32x32x32 grid using the binvox tool with default settings (bounding box normalization and binary occupancy).

The dataset was randomly split into approximately 80% for training and 20% for testing, which is the standard approach in the literature. In total, 27972 training samples and 6991 test samples were used. The split was performed at the object-instance level, where each unique 3D model in ShapeNet (identified by its model ID within each category) was assigned exclusively to either the training or the test set. Consequently, all rendered 2D views corresponding to a given 3D object instance were placed entirely within a single split to prevent data leakage across training and testing.

The 2D data consisted of 32-bit Red Green Blue Alpha (RGBA) images in a 137×137×4 format. The 3D data comprised voxelized objects with dimensions of 32×32×32. As observed in Figure 2, the table object had the highest number of samples in the data distribution, whereas the telephone had the fewest.

To provide a qualitative understanding of our dataset, Figure 3 illustrates several examples from the car, table, and telephone objects.



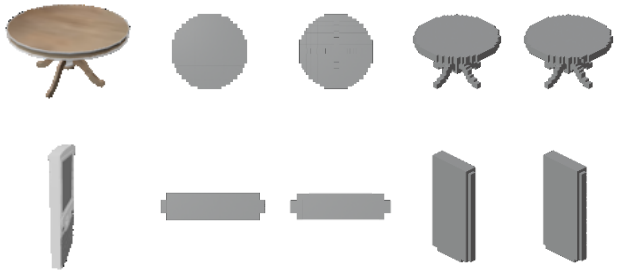


Figure 3. Representative dataset samples for car, table, and telephone objects.

Each row shows a 2D input image and its corresponding 3D ground-truth voxel model from multiple viewpoints. Each example consisted of a 2D input image and its corresponding 3D ground-truth (GT) voxel model shown from multiple viewpoints. The term GT refers to the 3D object found in the dataset, which corresponds to the 2D input. The term three-quarters refers to the representation of the relevant object with intermediate angles, such as 45° and 60°. Two different angles are shown in the figure: TQ-1 and TQ-2.

2.2 Baseline architecture: 3D-VAE-GAN model

The 3D-VAE-GAN architecture [4] is a generative architecture proposed by Wu et al. in 2016. It is based on generating 3D data from 2D data and forms the basis of our proposed work. It presents a combination of the VAE and GAN architectures. The architecture consists of three neural networks: Encoder, Generator (which also functions as the decoder neural network of the VAE), and Discriminator. 2D inputs of size 137×137×4 was passed through five convolutional blocks of the encoder neural network to obtain the mean (μ) and log-variance ($log\ (\sigma^2)$) values. These values enable the input to be represented probabilistically in the latent space. Under normal circumstances, these values cannot be differentiated. Therefore, end-to-end training cannot be achieved. The reparameterization technique solves this problem by enabling end-to-end training and back propagation. The result of the encoder neural network is a latent-space vector called *z*. This vector was passed through the five deconvolutional blocks of the generator neural network to generate a 3D object corresponding to the 2D input. The real 3D objects in the dataset and the fake 3D objects generated by the generator were passed through the five convolutional blocks of the discriminator neural network to distinguish between real and fake objects.

The VAE component of the architecture is responsible for reconstructing a 3D object from a 2D input image and regularizing the learned latent space. The reconstruction loss (L_{recon}) given in Eq. 1 measures the performance of the object generated by the generator by using the *z-value* passed through the encoder. This loss is minimized by both the encoder and generator. The Kullback-Leibler (KL) loss (L_{KL}) given in Eq. 2 measures the similarity of the latent-space distribution learned by the encoder to the standard normal distribution and is minimized only by the encoder.

$$L_{recon} = \left|\left|x - G\big(E(I)\big)\right|\right|^2 \quad (1)$$

Here, *I* represents the input image, and *x* represents the actual 3D voxel object.

$$L_{KL} = -0.5 \times \sum_{i=1}^{z_{size}} (1 + log\ (\sigma^2)\ - \mu^2 - \sigma^2) \quad (2)$$

Here, μ and σ^2 represent the mean and variance of the latent-space distribution estimated by the encoder. z_{size} denotes the dimension of the latent space vector.

The GAN component of the architecture is based on the competition between the Generator and Discriminator. The Generator wants the discriminator to label the fake examples it generates as real, whereas the discriminator attempts to successfully classify the real examples in the dataset and the fake examples generated by the generator. This competitive game is mathematically expressed in Eq. 3, which is minimized by the Discriminator, and Eq. 4, which was minimized by the generator.

$$L_D = -E[log\ D(x)\ ] - E\left[log\ \Big(1 - D\big(G(z)\big)\Big)\right] \quad (3)$$

$$L_G = -E[log\ (D(G(z))\ )\ ] \quad (4)$$

The training process follows an alternating optimization procedure. In each iteration, the discriminator is updated first, followed by the encoder and the generator. The total loss for each network is formulated as follows:

$$L_{D_{total}} = \lambda_{adv} L_D \quad (5)$$

$$L_{E_{total}} = \lambda_{recon} L_{recon} + \lambda_{KL} L_{KL} \quad (6)$$

$$L_{G_{total}} = \lambda_{adv} L_G + \lambda_{recon} L_{recon} \quad (7)$$

In our experiments, all weighting coefficients are set to unity ($\lambda_{recon} = \lambda_{KL} = \lambda_{adv} = 1.0$), prioritizing an equal contribution from each loss term.

The expression E in the equations represents the expected value. A detailed block diagram of our complete architecture, which highlights the integration of our proposed attention mechanism, is presented in the next section.

The 3D-VAE-GAN backbone consists of three primary networks: an encoder, a generator, and a discriminator. While the general architecture follows Wu et al. [4], we provide the exact layer specifications, including channel counts, kernel sizes, and strides, in Table 2. The networks are composed of five convolutional (or deconvolutional) blocks, each integrating batch normalization and activation functions to ensure stable training.

2.3 Proposed Complex Phase-Shifting Attention (CPSA) mechanism

The problem of generating 3D objects from 2D images is a challenging task due to the inherent lack of information. Classical attention mechanisms in the literature may be insufficient for capturing global relationships because they focus on the spatial domain and local relationships. To address such shortcomings, we propose CPSA to identify relationships that are not present in the spatial domain or that cannot be detected by classical convolution layers by applying attention in the frequency domain. Applying attention to the frequency domain offers significant advantages in terms of preserving

Table 2. Architectural specifications of the Baseline 3D-VAE-GAN. Each block consists of a Convolution/Deconvolution layer, Batch Normalization (BN), and Activation.

Network	Layer/	Type	Channels	Kernel/	Normalization	Activation	Output

	Block		(In/Out)	Stride/ Padding			Size
Encoder	Block 1	Conv2d	4/64	5/2/2	BN	ReLU	69x69x64
	Block 2	Conv2d	64/128	5/2/2	BN	ReLU	35x35x128
	Block 3	Conv2d	128/256	5/2/2	BN	ReLU	18x18x256
	Block 4	Conv2d	256/512	5/2/2	BN	ReLU	9x9x512
	Block 5	Conv2d	512/400	5/2/2	BN	ReLU	5x5x400
	FC 1, 2	Linear	10000/200	-	-	-	200
Generator	Block 1	Deconv3d	200/256	4/2/1	BN	ReLU	2x2x2x256
	Block 2	Deconv3d	256/128	4/2/1	BN	ReLU	4x4x4x128
	Block 3	Deconv3d	128/64	4/2/1	BN	ReLU	8x8x8x64
	Block 4	Deconv3d	64/32	4/2/1	BN	ReLU	16x16x16x32
	Block 5	Deconv3d	32/1	4/2/1	-	Sigmoid	32x32x32x1
Discriminator	Block 1	Conv3d	1/32	4/2/1	BN	LReLU (0.2)	16x16x16x32
	Block 2	Conv3d	32/64	4/2/1	BN	LReLU (0.2)	8x8x8x64
	Block 3	Conv3d	64/128	4/2/1	BN	LReLU (0.2)	4x4x4x128
	Block 4	Conv3d	128/256	4/2/1	BN	LReLU (0.2)	2x2x2x256
	Block 5	Conv3d	256/1	4/2/1	-	Sigmoid	1x1x1x1

structural integrity, capturing fine details, and ensuring geometric compatibility.

2.3.1 CPSA architecture

CPSA consists of two components that enhance the information processing capabilities of attention mechanisms within extracted feature maps. These components are the classic channel attention (*CA*) and the spatial-frequency attention (*SFA*) mechanism that operates in the frequency domain, which is our main innovation. The general operation and processes of the CPSA are described in equations 8-10 below.

$$Y = CA(X) \cdot X \quad (8)$$

$$M_{freq} = SFA(Y) \quad (9)$$

$$Output = Y \times M_{freq} \quad (10)$$

Here, X represents the input feature map, Y represents the feature map with the channel attention applied, M_{freq} represents the spatial attention mask derived from the frequency domain, and the *output* represents the final output of the CPSA module.

- Channel Attention (CA): This component identifies the relationships between the channels of the extracted feature maps and determines which channel is more important. First, the input feature maps are compressed using global average pooling and max pooling operations. This results in a feature vector for each channel that contains global context information. This vector was then used with a multilayer perceptron and sigmoid activation to obtain the channel weights. These weights are multiplied by the original feature map, enabling the architecture to determine which channels to focus on and which are more important.
- Complex Frequency Attention Mask (CFAM): This component represents CPSA's primary innovation and contribution to the literature. This component performs operations directly in the frequency domain, thereby enabling spatial information to be handled from a different perspective.

Figure 4 shows how an image is represented in the spatial domain and frequency domain using five example images. In the spatial domain, images are represented by pixels, which is what most people think of when they hear a word image.

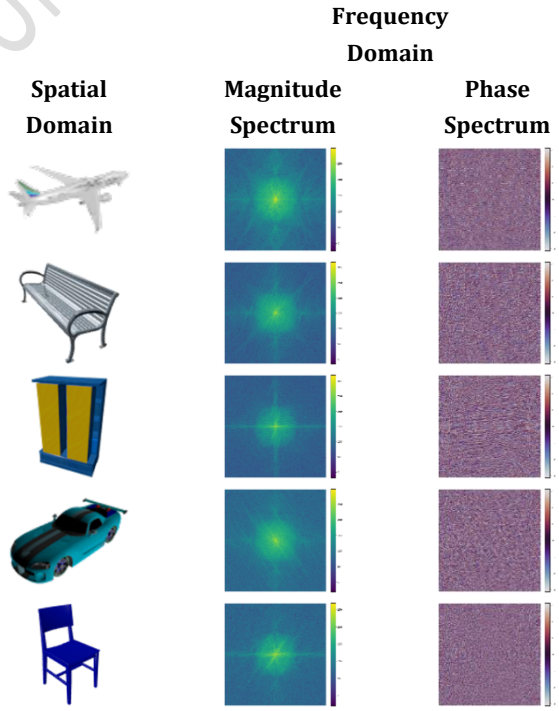


Figure 4. The display of five different image data in the spatial and frequency domains.

The frequency domain, on the other hand, is a slightly more abstract representation and is created by combining the magnitude spectrum and the phase spectrum. The magnitude spectrum allows us to determine the frequencies that are more dominant and intense in the image. A high-magnitude value indicates sharp transitions, edges, and details. On the other hand, a low-magnitude value represents a flatter and more homogeneous area. The overall structure and contrast of the

image were revealed by the magnitude spectrum. The phase spectrum provides spatial position information regarding the frequency components. Spatial information about the edges and details in the magnitude spectrum is provided by the phase spectrum. Only the phase information can preserve the main structure of the image, but only the magnitude information often makes the image unrecognizable. A Fourier transform was used for the transition to the frequency domain. The images in Figure 4 are purely illustrative and are provided for the sake of clarity. The proposed method is based on expressing the features obtained from the images in the frequency domain rather than the images themselves.

When an input feature map (Y) is given, the CFAM component follows the following steps:

1. The input feature map Y is subjected to a 2D Fast Fourier Transform (FFT) using the FFT_{2D} operator, thereby converting it from the spatial domain to the frequency domain. This allows for a more detailed analysis of the architecture's global patterns, periodic structures, and relationships between different frequency components by expressing them in terms of the magnitude and phase spectrum. As a result of this process, the F_Y value given in Eq. 11 is obtained.

$$F_Y = FFT_{2D}(Y) \quad (11)$$

2. F_Y is a complex number. The real ($Re(F_Y)$) and imaginary ($Im(F_Y)$) parts of this number are combined and fed into our specially designed modulator network as the input. The main task of this network is to estimate a complex modulation mask (M_{mod}) that adaptively adjusts the magnitude and phase of the F_Y value in the frequency domain. This process is shown in Eq. 12. The estimated mask is multiplied by the F_Y value to perform the process we call Complex Phase-Shifting. As a result of this process, the value we call $F_{Y,modulated}$, given in Eq. 13, is obtained. This step enables the architecture to focus on specific frequency components or to give them less importance. This presents a frequency-based approach to the attention mechanism applied in the spatial domain. The Modulator neural network has an architecture consisting of two convolution layers (Conv2d) and ReLU activations with BatchNorm2d (2D batch normalization layer) between them.

$$M_{mod} = Modulator\big(Re(F_Y), Im(F_Y)\big) \quad (12)$$

$$F_{Y,modulated} = F_y \times M_{mod} \quad (13)$$

3. To determine how the modulation process applied in the frequency domain affects the spatial domain, a 2D Inverse Fast Fourier Transform ($IFFT_{2D}$) was applied to the $F_{Ymodulated}$ value. As a result, a complex feature map showing the relevant effect, which we call $Y_{spatial}$, was obtained. This process is shown in Eq. 14.

$$Y_{spatial} = IFFT_{2D}\big(F_{Y,modulated}\big) \quad (14)$$

4. Similar to F_Y, the $Y_{spatial}$ is also a complex number. The $Y_{spatial}$ was processed by a refiner network by combining its real ($Re(Y_{spatial})$) and imaginary ($Im(Y_{spatial})$) parts. This refiner network produces a single-channel spatial mask (M_{freq}) to render the complex feature map more meaningful. This mask was passed through a sigmoid activation function and normalized to the 0-1 range, resulting in the main attention map given in Eq. 15. In Eq. 15, the refined neural network is abbreviated as in Ref. As a result of this process, modulation in the frequency domain gains meaning in the spatial domain and is successfully transformed. The refiner network also has an architecture similar to that of the modulator network, consisting of two convolution layers (Conv2d) and ReLU activations between them, along with BatchNorm2d. The final attention map obtained was multiplied by the feature map passing through the CA to obtain the final output of the CPSA.

$$M_{freq} = Sigmoid\Big(Ref\Big(Re\big(Y_{spatial}\big), Im\big(Y_{spatial}\big)\Big)\Big) \quad (15)$$

The steps described here are shown in detail in the block diagram shown in Figure 5. The input feature map X (B, C, H, W) corresponded to X (batch size, number of channels, height, and width). The convolution operations are expressed as Conv2d (input channels, output channels, kernel size, stride, and padding). The $\oplus$ symbol in the block diagram indicates that the inputs are added, while $\otimes$ indicates that the inputs are multiplied. The feature maps used as input and obtained in the intermediate layers belong to chair images. The chair object was chosen solely as an example and had no special significance.

To provide a clear and structured overview of the proposed mechanism, the complete step-by-step procedure of the CPSA module is summarized in Algorithm 1. This algorithmic sequence details the transition between spatial and frequency domains, the complex spectral modulation process, and the final feature fusion, corresponding to the architectural block diagram shown in Figure 5.

Algorithm 1: Complex Phase-Shifting Attention (CPSA)

Input: Feature Map $X \in \mathbb{R}^{B\times C\times H\times W}$

Output: Refined Feature Map $Y \in \mathbb{R}^{B\times C\times H\times W}$

1. Initialize Channel Attention **CA**, $Modulator\phi, Refiner\psi$
2. Channel Attention: $w = \mathbf{CA}(X); X_{ca} = X \odot w$
3. Spectral Transformation: $F = \text{FFT2D}(X_{ca}) \in \mathbb{C}^{B\times C\times H\times W}$
4. Complex Modulation:
   - Concatenate: $F_{in} = [\text{Re}(F); \text{Im}(F)]$
   - Predict Mask: $M = \phi(F_{in}) \in \mathbb{C}^{B\times C\times H\times W}$
   - Apply: $F_{mod} = F \cdot M$
5. Spatial Reconstruct: $X_{ifft} = \text{IFFT2D}(F_{mod}) \in \mathbb{C}^{B\times C\times H\times W}$
6. Mask Refinement:
   - Concatenate $X_{in} = [\text{Re}(X_{ifft}); \text{Im}(X_{ifft})]$
   - Generate Spatial Mask: $S = \text{Sigmoid}(\psi(X_{in}))$
7. Output Fusion: Return $Y = X_{ca} \odot S$

2.3.2 Architectural integration

CPSA modules were strategically integrated into the 3D-VAE-GAN architecture. In the 3D-VAE-GAN architecture, the task of the encoder neural network is to map 2D inputs to the latent space and fit them to a normal distribution. Therefore, successful representation of these 2D inputs is very important. Based on this, a CPSA module was integrated into the architecture after the 2nd, 3rd, and 4th layers of the Encoder neural network. Since very low-level features (edges, corners, color transitions, etc.) were extracted in the first layer, it was

considered that the attention mechanism would be less meaningful at this point and would incur additional computational costs. However, after the second layer, while low-level features were still extracted, increasingly abstract and high-level features (objects, faces, meaning, etc.) were also extracted. Therefore, this choice was made to take advantage of all low, medium, and high-level features while also considering computational cost. The fifth layer produces the final feature vector before mapping it to the latent space. Figure 6 shows a block diagram of the entire 3D-VAE-GAN architecture and the integration process of the proposed CPSA modules.

2.4 Experimental setup

This section outlines the experimental framework designed to evaluate the performance of the proposed CPSA-enhanced architecture. We first present implementation details and the SA technique applied during training. Subsequently, we describe the baseline models (SE and CBAM) used for comparison. Finally, we define the primary quantitative metric, IoU, as the basis of our objective evaluation.

2.4.1 Implementation details

The experiments were conducted on a computer equipped with an i7-4790 CPU, 16 GB RAM, and a 6 GB NVIDIA GTX 980 Ti FE GPU. To ensure the reproducibility of our results, the key hyperparameters used for the training are listed in Table 3. The complete source code for our model and experiments is publicly available on GitHub at https://github.com/zafer-sn/CPSA-3D . The implementation was developed using Python 3.12.9, PyTorch 2.6.0, and CUDA 12.6.

Table 3. Key hyperparameters used for training.

Hyperparameter	Value
Input Image Size	137x137
Voxel Grid Size	32x32x32
Latent Vector Dimension	200
Optimizer	Adam
Encoder Learning Rate	0.0001
Generator Learning Rate	0.0025
Discriminator Learning Rate	0.001
Adam Betas	(0.5, 0.5)
Batch Size	32
Epochs	100
LeakyReLU Slope	0.2
Voxel Occupancy Threshold	0.5
Reconstruction Loss Weight	1.0
KL Divergence Weight	1.0
Adversarial Loss Weight	1.0
LR Scheduler/Gradient Clipping	None
Early Stopping Strategy	None
Tuning Strategy	Global

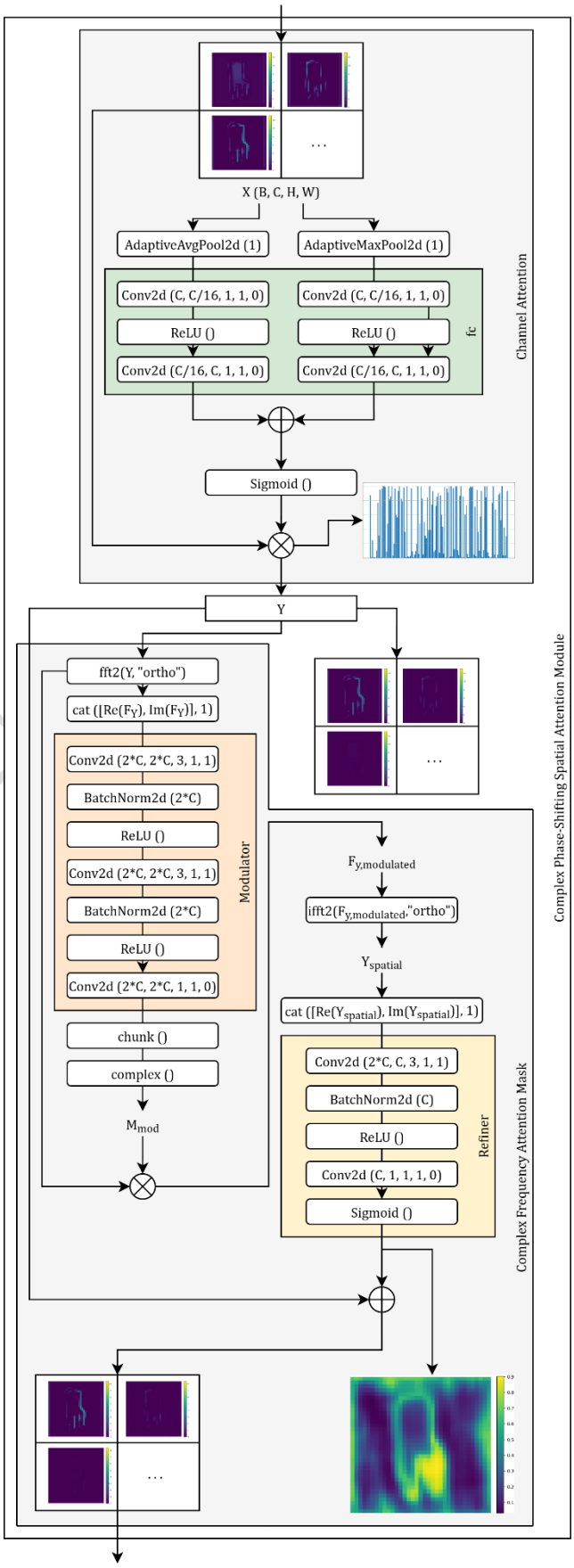


Figure 5. CPSA module block diagram.

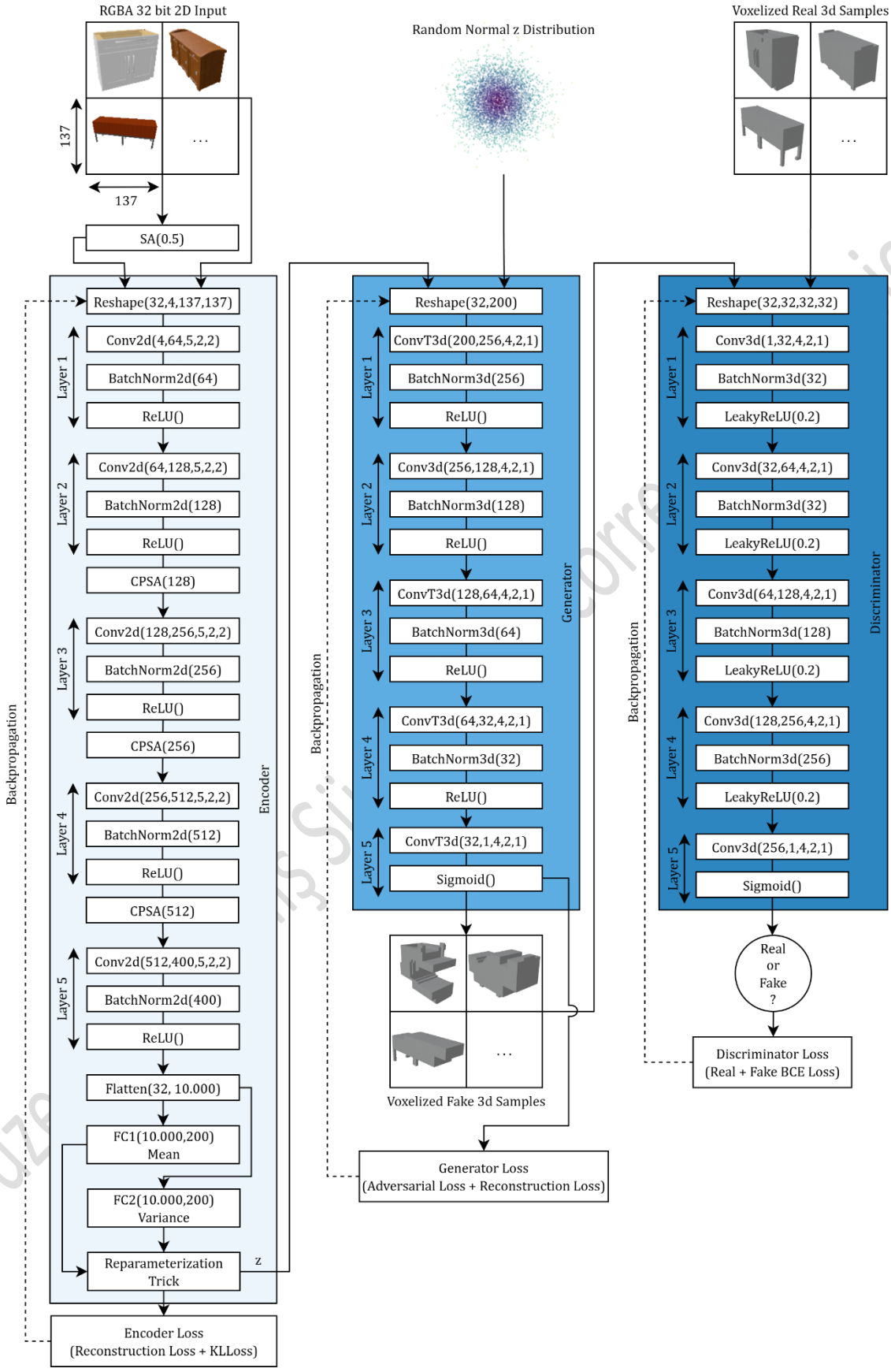


Figure 6. Block diagram of the 3D-VAE-GAN architecture enhanced with CPSA and SA modules.

2.4.2 Benchmark methods

The performance of the proposed architecture was rigorously evaluated using a carefully selected set of baseline and state-of-the-art models. The experimental design was structured to measure performance along two main axes: attention mechanism selection and the impact of the SA technique. This approach made it possible to clearly distinguish the contributions from the proposed CPSA module from the gains obtained solely from data augmentation. Five different model configurations were compared.

1. Baseline 3D-VAE-GAN: Original 3D-VAE-GAN architecture without any attention mechanism or data augmentation technique. This model served as the absolute baseline for all comparisons.
2. Baseline + SA: Baseline model trained with the SA technique applied to the input images. This configuration was critical for measuring the performance impact of data augmentation in isolation.
3. SE + SA: A model trained with SA with SE attention blocks added to the baseline architecture. This provides a benchmark for comparison with the classic and effective channel-based attention mechanism.
4. CBAM + SA: A model trained with SA, with CBAM added to the baseline architecture. This serves as a comparison with a more advanced mechanism that combines both channel and spatial attention.
5. CPSA + SA (Ours): The proposed final architecture integrates our CPSA modules into the encoder and is trained with SA.

To ensure a fair comparison, all attention modules (SE, CBAM, and CPSA) were integrated into the encoder network at the same strategic positions, specifically after the second, third, and fourth convolutional layers, respectively.

2.4.3 Data augmentation

In this study, an SA data augmentation technique was proposed and used to increase the generalization ability of the architecture and prevent overfitting. In this method, all channels (RGB) except the alpha channel of the image are set to zero (blacked out), leaving only the silhouette of the object. This process is applied with a 50% probability to prevent the architecture from becoming overly dependent on texture, color, and other superficial features, thereby ensuring that it can generate successful results even when these features are not present. Because many architectures accept a 3-channel RGB input, the successful application of this technique is directly dependent on the use of a 4-channel (RGBA) input. The 50% probability here means that this process will be applied to half of the inputs and not to the other half. Applying this process to all inputs means the architecture never sees standard inputs and could cause more harm than good. Therefore, it is important to apply it at the right rate and to the appropriate problem. Figure 7 shows example inputs and their states after SA application.

2.4.4 Evaluation Metrics

The IoU metric was used to quantitatively evaluate the proposed method. It is widely adopted in 3D object generation tasks and measures the ratio between the intersection and the union of two volumes. In our setting, IoU is computed between the generated voxel grid and the corresponding ground truth object.

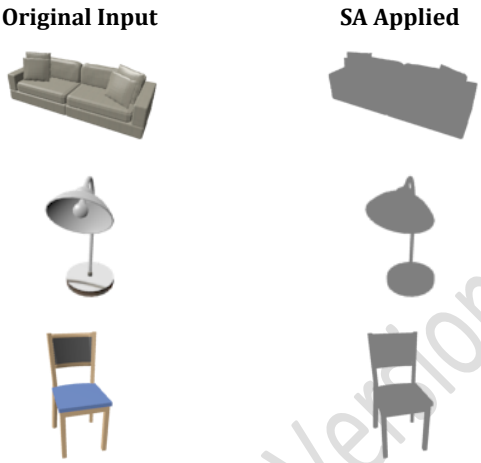


Figure 7. Visual examples of SA applications. (The original 4-channel (RGBA) input images are shown in the left column, while the right column shows the images after SA is applied.)

Before computing IoU, the generator outputs voxel occupancy probabilities in the range [0,1]. These probabilities are converted into binary voxel grids using a fixed threshold $\tau = 0.5$. No additional post-processing techniques (e.g., smoothing, filtering, or morphological operations) are applied to the generated voxel outputs. IoU values range between 0 and 1, where 1 indicates a perfect match and 0 indicates no overlap. Eq. (16) presents the IoU formulation for two volumes denoted as A and B.

$$IoU = \frac{|A \cap B|}{|A \cup B|} \quad (16)$$

Figure 8 illustrates the calculation of the IoU metric. A and B represent the two volumes compared. The intersection of A and B is represented by the volume shown with white dashed lines, while their union is represented by the volume shown with red dashed lines. The IoU was calculated as the ratio of the intersection volume to the union volume.

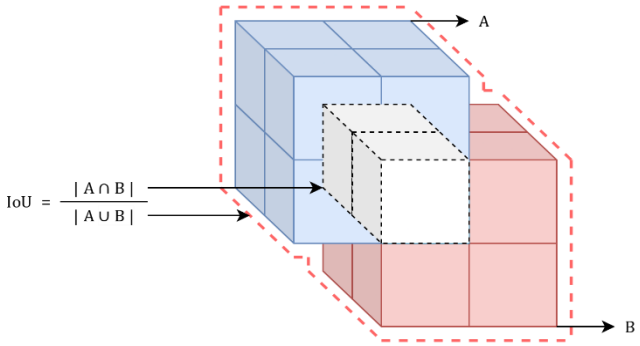


Figure 8. Visual calculation principle of the IoU metric.

3 Results

In this section, we discuss the methods used to evaluate the proposed method. These are described as quantitative evaluations based on the IoU metric, observation-based qualitative evaluations, and time-based analyses. All evaluations were conducted using the Baseline, Baseline+SA, SE+SA, CBAM+SA, and CPSA+SA architectures, as detailed in Section 2.4.2 for 13 different objects.

3.1 Quantitative evaluations

In the literature, quantitative evaluations are described as methods based on a mathematical measurement metric and a

performance measurement based on it. To evaluate the proposed method in this regard, the IoU metric, the details of which are provided in Section 2.4.4, was used.

Table 4 lists the test results for the proposed and other methods. Upon examining the table, the dominance of the CPSA + SA architecture was evident. CPSA+SA achieved the most successful results for the other 11 objects, except for lamp and rifle objects. Particularly, successful results were obtained for airplanes, displays, and watercraft objects. For lamps and rifles, the difference between the other architectures and CPSA is quite small, and it is clear that CPSA is promising.

The average IoU reaches 57.6%, providing relative improvements of 4.1% over Baseline, 3.1% over Baseline+SA, 2.9% over SE+SA, and 2.2% over CBAM+SA. On an object-wise basis, CPSA+SA yields relative gains ranging from 0.4% (Car) to 11.9% (Display) compared with Baseline, from 0.2% (Chair) to 10.1% (Display) compared with Baseline+SA, from –0.8% (Lamp) to 9.2% (Display) compared with SE+SA, and from –0.6% (Lamp) to 8.1% (Display) compared with CBAM+SA. It should be noted that the reported percentages correspond to relative improvements rather than absolute differences. In addition, the integration of the SA alone provides a small yet consistent improvement over the Baseline (average IoU increases from 55.4% to 55.9%). Another notable point in the table is that all architectures were successful for the car object and achieved very high IoU values. In contrast, the results for the lamp object were low for all architectures. The reason for this is thought to be that the lamp object is overly complex, and it is quite difficult to make predictions for the invisible parts of the lamp object with 2D inputs. The results clearly show that the use of attention in the encoder architecture has a positive effect on the results, and the success of the proposed CPSA is evident. The SA method had a slightly positive effect on the results.

Table 4. Table showing test IoU values for 13 different objects in 5 different architectural structures

Object	Baseline	Baseline + SA	SE+SA	CBAM+SA	CPSA+SA(Ours)
Airplane	0.575980	0.582330	0.589495	0.589028	**0.611608**
Bench	0.416848	0.423741	0.422960	0.417870	**0.428522**
Cabinet	0.641112	0.637036	0.630690	0.665157	**0.672622**
Car	0.817119	0.819607	0.819141	0.801314	**0.820698**
Chair	0.458249	0.465507	0.446671	0.466675	**0.468890**
Display	0.418954	0.425813	0.429476	0.433793	**0.468800**
Lamp	0.381543	0.400790	0.413315	**0.418204**	0.415884
Speaker	0.605094	0.616506	0.613985	0.621789	**0.626232**
Rifle	0.551763	0.547947	**0.555059**	0.536750	0.550325
Sofa	0.632300	0.641757	0.632332	0.630248	**0.644645**
Table	0.522163	0.518084	0.513315	0.518846	**0.535415**
Telephone	0.650187	0.653251	0.675090	0.683888	**0.688977**
Watercraft	0.531614	0.534351	0.540153	0.546879	**0.561956**
Average	0.554071	0.558978	0.560129	0.563880	**0.576506**

To validate the performance and statistical reliability of the proposed CPSA+SA model, a comprehensive evaluation was conducted using three randomly selected initial seeds (0, 42, 123). When examining the results in Table 5, it can be seen that the CPSA+SA method provides a consistent increase in the average IoU score compared to the Baseline model in all tested categories (Airplane, Display, Table). Specifically, the average IoU score increased from 60.72% to 63.08% in the Airplane category and from 50.64% to 52.70% in the Table category. This demonstrates that the proposed CFAM and SA components systematically improve 3D reconstruction accuracy independently of the stochastic nature of deep learning models.

When examining the standard deviation (std) values in terms of the model's robustness, it was observed that CPSA+SA exhibited lower variance compared to the Baseline model. The decrease in variance from 0.84% to 0.40% in the Airplane category and from 0.46% to 0.23% in the Table category indicates that the model is less sensitive to initial weights and data shuffling operations. This statistical narrowing demonstrates that the proposed architecture stabilizes the training process and can produce highly accurate results in a reproducible manner, even under different initial conditions. Despite a slight increase in variance in the Display category, the general trend is that the model follows a more reliable learning curve when learning geometric details.

When evaluating training times in terms of computational cost, it was determined that CPSA+SA requires approximately 1.7 to 1.8 times the additional time compared to the Baseline model. For example, in the Table category, the average training time per epoch increased from 60.92 seconds to 107.22 seconds. This increase is directly related to the FFT/IFFT operations performed in the frequency domain and the architectural complexity introduced by the complex modulation network. However, considering the achieved IoU increase and the high stability demonstrated by the model, this computational cost is considered an acceptable trade-off. Furthermore, the extremely low standard deviation in training times (±0.01 to ±0.04 seconds) confirms that the computational load of the model is predictable and stable.

The computational complexity analyses presented in Table 6 reveal the structural depth and resource usage of the proposed CPSA+SA model in comparison with competing methods. When examining the number of parameters, it is seen that the Baseline, SE+SA, and CBAM+SA models are at a similar scale with approximately 22.2 million parameters, but this value increases to 54.63 million in the CPSA+SA model. This approximately 145% increase in parameters stems from the complex frequency modulation network (modulator) within the CPSA module and the convolutional layers that extract spectral features. While standard attention mechanisms (SE and CBAM) have a negligible effect on the

Table 5. Statistical performance evaluation and training efficiency comparison of the proposed model and baseline across multiple random seeds.

Category	Model	Average IoU (mean ± std)	Training Time (s/epoch)
Airplane	Baseline	0.6072 ± 0.0084	**27.62 ± 0.03**
	CPSA+SA (Ours)	**0.6308 ± 0.0040**	49.40 ± 0.01
Display	Baseline	0.4242 ± 0.0104	**8.25 ± 0.01**
	CPSA+SA (Ours)	**0.4326 ± 0.0130**	14.25 ± 0.02
Table	Baseline	0.5064 ± 0.0046	**60.92 ± 0.05**
	CPSA+SA (Ours)	**0.5270 ± 0.0023**	107.22 ± 0.04

Bold values indicate the superior performance in each metric: highest values for Average IoU and lowest values for Training Time.

number of parameters, CPSA's denser architecture is considered a structural counterpart to its information processing capacity in the frequency domain.

In terms of the model's operational efficiency, inference time and memory (VRAM) usage data clearly summarize the additional computational load introduced by the CPSA mechanism. Inference time increased by approximately 58%, rising from an average of 11.4 ms per sample to 18.0 ms. Similarly, maximum VRAM usage has increased from 1021 MB to 2405 MB, representing an increase of approximately 2.3 times. This confirms that FFT and IFFT operations consume GPU memory resources more intensively. However, the 18 ms inference speed shows that the model can still perform well in near real-time applications and that the provided IoU increase justifies this additional hardware cost.

When evaluating the training processes, it was determined that the CPSA+SA model's training time per epoch is approximately 1.8 times longer than the Baseline model. While attention mechanisms such as SE+SA and CBAM+SA have a very limited effect on training time (approximately 1-3% increase), the increase in time introduced by CPSA+SA is a result of the model's architectural complexity. However, it should be emphasized that the performance increase offered by CPSA+SA is based not only on the increase in the number of parameters but also on the qualitative improvement provided by feature modulation in the frequency domain. In light of these findings, CPSA+SA offers a strong "accuracy-efficiency trade-off" for 3D reconstruction tasks that aim for high accuracy.

Table 6. Comprehensive evaluation of computational complexity, execution efficiency, and hardware resource utilization across different object categories.

Category	Model	Parameters (M)	Avg. Epoch Time (s)	Inf. Time (ms/sample)	Max VRAM (MB)
Airplane	Baseline	**22.24**	**27.66**	**11.38**	**1021.3**
	SE+SA	22.28	27.85	11.53	1058.6
	CBAM+SA	22.28	28.83	12.23	1092.7
	CPSA+SA (Ours)	54.63	49.39	17.91	2404.6
Display	Baseline	**22.24**	**8.25**	**11.42**	**1021.4**
	SE+SA	22.28	8.33	11.63	1056.7
	CBAM+SA	22.28	8.61	12.17	1092.5
	CPSA+SA (Ours)	54.63	14.23	17.94	2404.1
Table	Baseline	**22.24**	**60.97**	**11.42**	**1021.8**
	SE+SA	22.28	61.65	11.58	1057.1
	CBAM+SA	22.28	63.96	12.36	1092.7
	CPSA+SA (Ours)	54.63	107.25	18.05	2405.2

Bold values represent the most efficient results in each metric. Parameters and VRAM usage are determined by the model architecture and remain consistent across categories. Average Epoch Time varies based on the sample size of each category's dataset

A comprehensive ablation study was conducted to isolate the unique contribution of each module of the proposed architecture to the success of 3D reconstruction. Examining the results in Table 7, it is observed that each component added to the base model provides a gradual improvement in the IoU score. For example, in the Airplane category, the base model's IoU score of 0.6188 increased to 0.6232 with the addition of SA alone, and to 0.6256 with the addition of the CPSA module alone [CPSA (w/o SA)]. This confirms that both the data augmentation strategy (SA) and the attention mechanism in the frequency domain (CPSA) can enhance the model's ability to understand object geometry even when used independently.

One of the most striking findings of the study is the interaction between channel-based attention (CA) and frequency-based spatial attention (CFAM) mechanisms. In the Airplane and Table categories, the CFAM+SA configuration with frequency masking capability was observed to perform better than channel attention alone (CA+SA). Particularly in the Airplane category, the score of 0.6318 achieved by the CFAM module demonstrates that feature modulation in the frequency domain is more effective than standard channel attention in capturing complex geometric details (wing structures, etc.). The full CPSA+SA configuration, which combines these two mechanisms, demonstrates synergy between the components

by delivering balanced and high performance across all categories.

Phase and Amplitude experiments conducted for an in-depth analysis of the masking strategy in the frequency domain provide a critical technical insight for 3D reconstruction tasks. In the Airplane category, phase information [CPSA (Phase-only) + SA: 0.6330] provides higher accuracy than amplitude information [CPSA (Amp.-only) + SA: 0.6279], which is consistent with the theory in the literature that "phase information takes precedence in representing object structure and contours." The fact that phase-focused masking yields more consistent results than amplitude-focused approaches in categories such as Display and Table demonstrates that the proposed CFAM module achieves success in preserving object silhouettes and structural integrity by optimizing phase information.

The ablation study results demonstrate that the full CPSA+SA model combines the best features of all subcomponents. The approximately 7% improvement (0.4394) over the Baseline model (0.4111) in the Display category and the increases in the Table category confirm the complementary relationship between SA and CFAM. While the SA module focuses the model on external contours, the CPSA module refines the structural details within these contours in the frequency domain, maximizing the accuracy of the final 3D model. These data confirm at an academic level that the proposed integrated architecture offers a high-performance and reliable solution for 3D object generation tasks.

Table 7. Ablation study of the proposed components on the test set.

Configuration	SA	CA	CFAM	Airplane	Display	Table
Baseline				0.6188	0.4111	0.5126
Baseline + SA	✓			0.6232	0.4349	0.5097
CPSA (w/o SA)		✓	✓	0.6256	0.4219	0.5156
CA + SA	✓	✓		0.6291	0.4417	0.5249
CFAM + SA	✓		✓	0.6381	**0.4538**	0.5248
CPSA (Phase-only) + SA	✓	✓	Phase	**0.6330**	0.4421	0.5164
CPSA (Amp. -only) + SA	✓	✓	Amplitude	0.6279	0.4416	0.5196
CPSA + SA (Ours)	✓	✓	Full	0.6326	0.4394	**0.5255**

Bold values indicate the best performance in each category.

Table 8 presents a quantitative comparison between the proposed method and representative single-view 3D reconstruction approaches reported in the literature, including voxel-based models such as 3DVAEGAN [4], 3D-R2N2 [5] and DRC [38], implicit representation-based methods such as ONet [30], and mesh-based approaches such as AtlasNet [39] and Pixel2Mesh [29]. As observed, the proposed CPSA-based model achieves the highest average IoU (0.577) across the 13 ShapeNet categories. In particular, the

proposed method outperforms competing approaches in categories such as Airplane, Car, Lamp, Table, and Watercraft.

It is worth noting that implicit representation-based methods such as ONet demonstrate strong performance in structurally complex categories (e.g., Cabinet and Sofa), while mesh-based approaches (e.g., AtlasNet and Pixel2Mesh) exhibit competitive results in selected object classes. However, the proposed CPSA framework consistently improves performance within a voxel-based generative architecture, achieving competitive results even against methods employing fundamentally different reconstruction paradigms.

We emphasize that results for competing methods are reported from their respective original publications and may differ in voxel resolution, training protocol, or evaluation settings. Therefore, the primary contribution of this study lies not in replacing implicit or mesh-based reconstruction approaches, but in demonstrating that frequency-domain attention can significantly enhance encoder representations within voxel-based generative models.

3.2 Qualitative evaluations

In the literature, qualitative evaluations are often referred to as observation-based. This type of evaluation, which can also be described as a subjective evaluation, is particularly important in the field of 3D generation. It is crucial to visually observe the generated objects and determine their suitability rather than relying solely on numerical evaluations.

Figure 9 shows the 3D object generated for the same 2D input by each architecture, as detailed in the previous sections. A visual inspection of the figure clearly demonstrates the success of the CPSA+SA architecture. For the airplane object, both engines were detected using the proposed architecture, and relevant voxels were successfully generated. Other architectures failed to recognize this detail. Sharp lines on the handles of the chair object are successfully generated using the proposed architecture. The difference in the display objects is clearly evident. For a display object shot from behind, other architectures can be said to have failed. Some architectures have even struggled to create display stands.

Table 8. Quantitative comparison of representative single-view 3D reconstruction methods on the ShapeNet 13-category subset using IoU. Results for competing methods are reported from their respective original publications.

Category	3D-R2N2 [5]	3DVAEGAN [4]	DRC [38]	ONet [30]	AtlasNet [39]	Pixel2Mesh [29]	Ours
Airplane	0.513	0.420	0.571	0.571	0.493	0.420	**0.612**
Bench	0.421	0.340	0.453	**0.485**	0.431	0.323	0.429
Cabinet	0.716	0.600	0.635	**0.733**	0.257	0.664	0.673
Car	0.798	0.760	0.755	0.737	0.282	0.552	**0.821**

Category	3D-R2N2 [5]	3DVAEGAN [4]	DRC [38]	ONet [30]	AtlasNet [39]	Pixel2Mesh [29]	Ours
Chair	0.466	0.360	0.469	**0.501**	0.328	0.396	0.469
Display	0.468	0.400	0.419	0.471	0.457	**0.490**	0.469
Lamp	0.381	0.320	0.415	0.371	0.261	0.323	**0.416**
Speaker	**0.662**	0.590	0.609	0.647	0.296	0.599	0.626
Rifle	0.544	0.540	**0.608**	0.474	0.573	0.402	0.550
Sofa	0.628	0.570	0.606	**0.680**	0.354	0.613	0.645
Table	0.513	0.330	0.424	0.506	0.301	0.395	**0.535**
Telephone	0.661	0.680	0.413	**0.720**	0.543	0.661	0.689
Watercraft	0.513	0.480	0.556	0.530	0.355	0.397	**0.562**
Average	0.560	0.491	0.533	0.571	0.352	0.479	**0.577**

Bold values indicate the best IoU per category, and underlined values indicate the second-best results.

The CPSA+SA architecture, on the other hand, created the relevant object and stand very smoothly. It is clear that the created object is similar to a GT. The lamp is another object for which the difference is clearly visible. It is observed that other architectures, except the proposed architecture, failed to establish a connection between the beginning and end of the lamp object. The proposed architecture successfully established a connection. It also successfully created socket sections. The difference is also very clear for the rifle object; in particular, the front handle section was detected and created only by the CPSA+SA architecture. Although there was some surface roughness for the table object, the gap between them was created only by the proposed architecture. A general performance can be said for cabinets, cars, sofas, and telephone objects. For Speaker objects, all architectures were relatively unsuccessful. In particular, the interior and exterior details of the Speaker were not detected by their respective architectures. Table 3 shows the relatively good IoU values for Speaker objects.

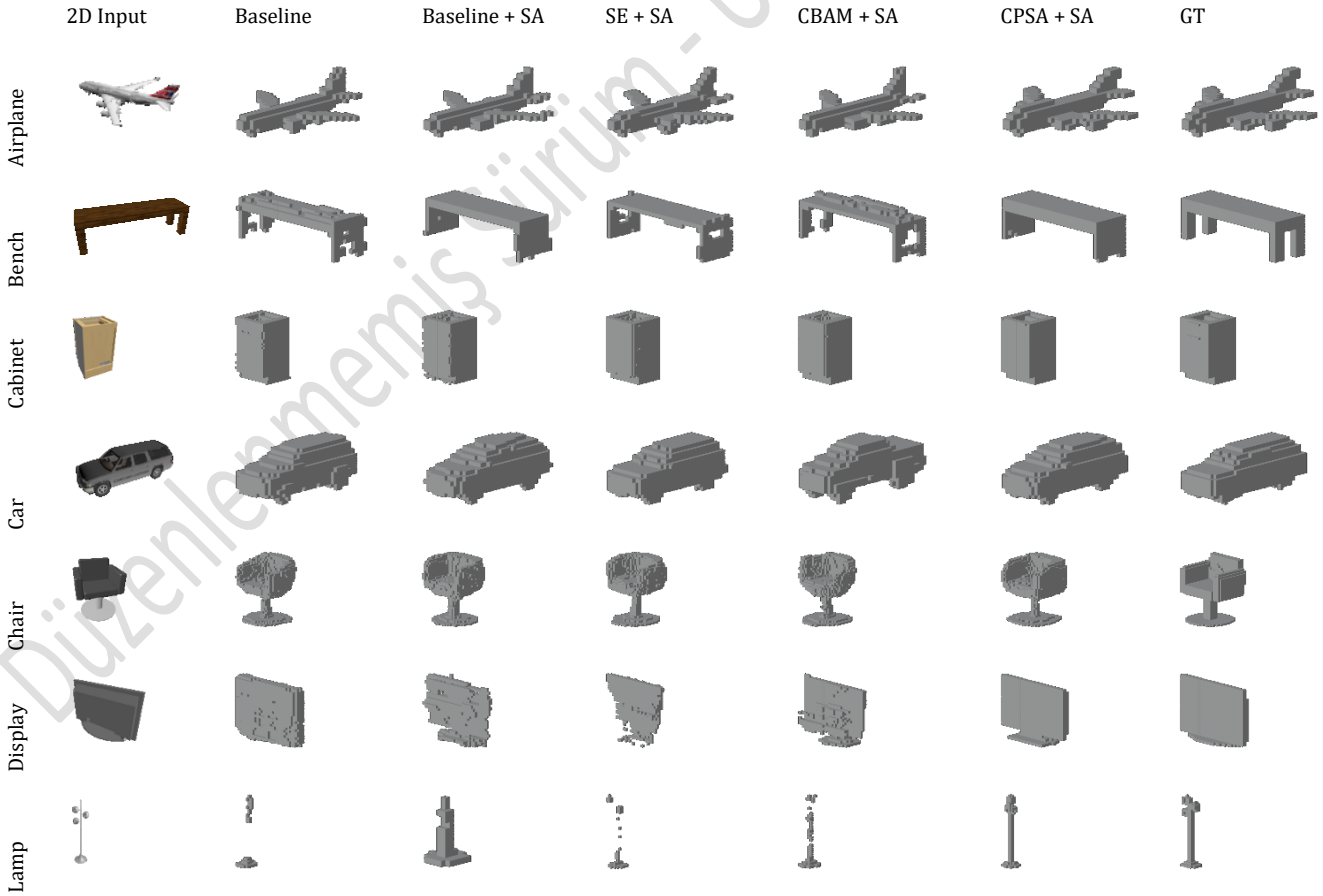


Figure 9. Qualitative comparison of 3D reconstruction results for samples selected from the test set. For each row, the same input view is provided to all architectures under identical training settings. The generated voxel outputs are binarized using a threshold of 0.5 and compared against the corresponding ground truth (GT) model.

However, this result is thought to be due to the angle of the 2D image, and in this case, the problem of edge detection. The architecture appears to struggle with the surface of the bench object. The proposed architecture successfully creates a smooth surface. However, it also appears to bridge the gap between the bench legs, and this is where it appears to fail. Outlines of the watercraft objects were successfully created. However, the architectures seem to follow a normal distribution without relying heavily on 2D input, as the simulation is clearly based on general watercraft objects in the dataset.

Figure 10 shows examples of the generation for five different object categories (Table, Lamp, Telephone, Airplane, and Bench) using the proposed CPSA+SA architecture. The red voxels (Generated Object) represent the 3D voxelized object generated by the CPSA+SA architecture against the 2D input image, whereas the blue voxels (Ground Truth) represent the desired 3D voxelized object corresponding to the 2D input image. The Overlaying section allows for a comparison by centering the Generated Object and Ground Truth. The green voxels represent the voxels where the Generated Object and Ground Truth are matched, that is, correctly estimated. As can be seen in the figure, the proposed architecture struggles to generate voxels, particularly around the edges of the table object and legs. There were deficiencies in the top and bottom sections of the lamp object. Telephone and airplane objects were successfully generated. The bench object exhibits significant shortcomings, particularly in the production of the seat and edges of the bench. Figures 9 and 10 show that objects belonging to the same category contain different types. For example, the lamp and bench objects appear very different in Figures 9 and 10. These examples represent completely random selections from the dataset and clearly demonstrate its diversity. Even though they share the same category, many different objects are present in the dataset.

2D Input Generated Object Ground Truth Overlaying

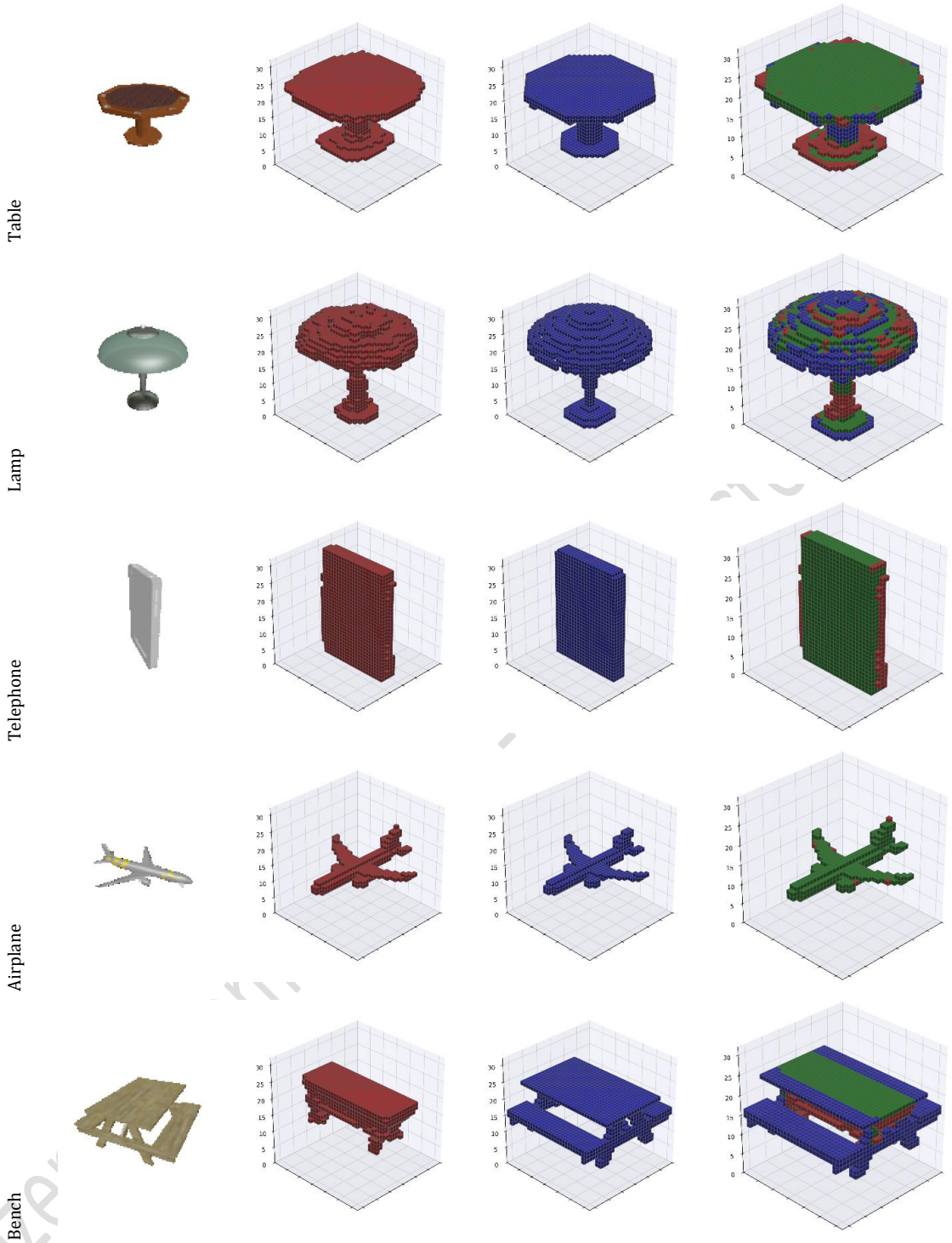


Figure 10. Detailed qualitative comparison of CPSA+SA reconstructions on test samples. The generated voxel outputs (red) are binarized using a threshold of 0.5 and directly overlaid with the ground truth (blue). Green voxels indicate correctly predicted occupancy. No post-processing is applied to the generated volumes.

3.3 Time Evaluation

Training time is also important in deep-learning architectures. If a highly successful architecture takes significantly longer to train than standard architectures, it is considered a disadvantage. Figure 11 graphically illustrates the training times for five different architectures for the telephone object with the fewest examples in the dataset, lamp object with the average examples, and table object with the most examples. The average training time for 100 epochs is presented in the graph.

For all three objects, the Baseline and Baseline + SA architectures completed the training in approximately the same amount of time. For telephone and lamp objects, the SE + SA and CBAM + SA architectures completed training at similar times. However, for the table object, these two architectures took slightly longer than the Baseline and Baseline + SA architectures, respectively. It is clear that the CPSA + SA architecture took longer than the other architectures for all the three objects. The CPSA + SA architecture took approximately

4.99 seconds longer for the telephone object, 11.05 seconds longer for the lamp object, and 13.05 seconds longer for the table object. It is clear that the proposed architecture has a disadvantage in terms of time; in this case, it is recommended to make a problem-specific choice. Although these increments are generally not very long, even a single second can be significant for problems such as cybersecurity attack detection, autonomous vehicles, and high-frequency algorithmic trading. It is clear that CPSA + SA is successful in the 3D generation field; however, this disadvantage should be considered when applying it to other problems. In the first epoch, the training time was visibly longer because the data were loaded into the GPU and the RAM. Furthermore, fluctuations were observed during training of the table objects. This is thought to be due to the large amount of data in the table object and the delay between loads depending on the batch size value.

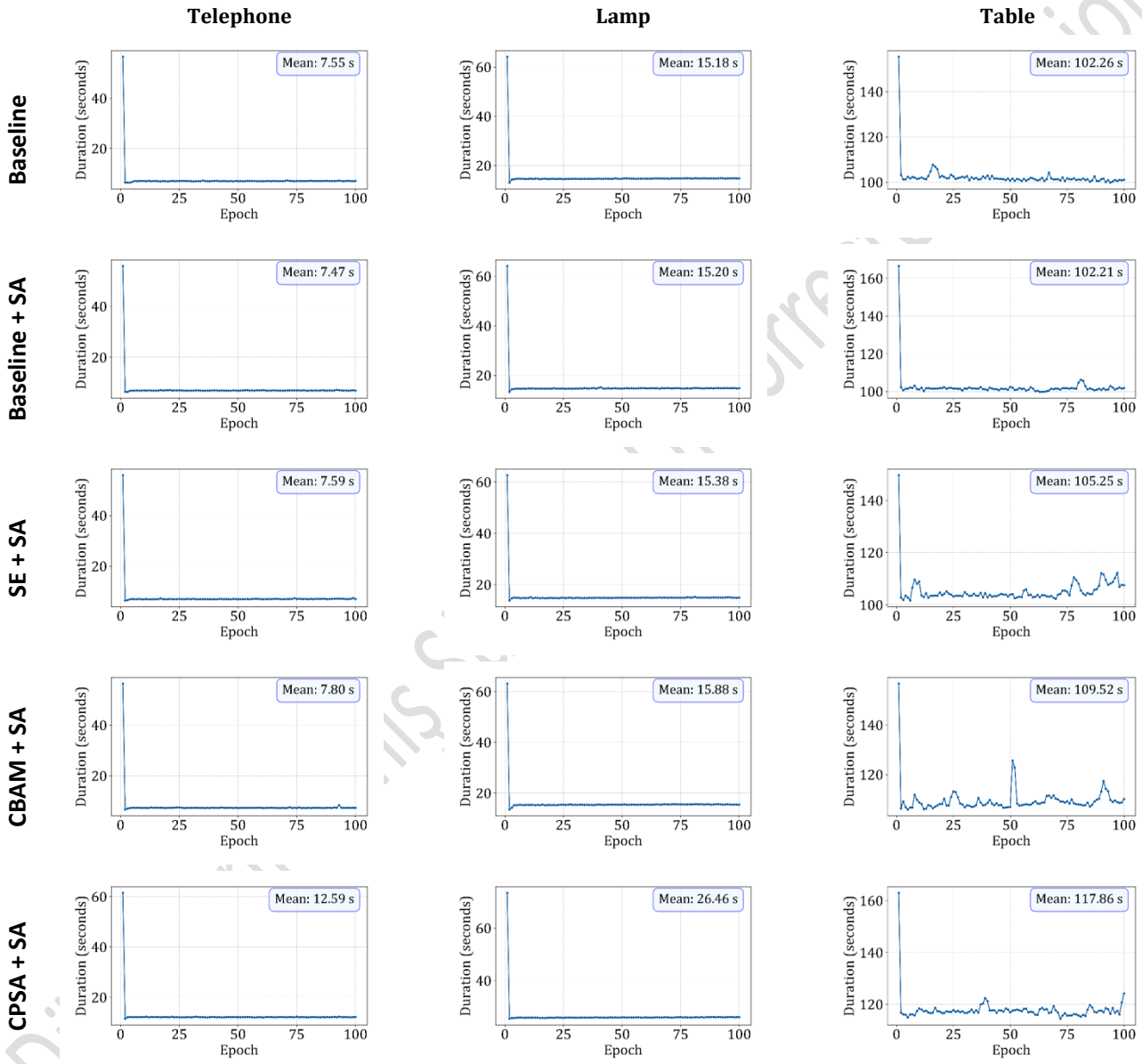


Figure 11. Training epoch durations for the five tested architectures across the three object datasets.

3.4 Discussion

The quantitative results in Table 4 indicate that the impact of CPSA varies across object categories. The largest relative improvement over the Baseline model is observed in the Display category (+11.9%), followed by Lamp (+9.0%) and Airplane (+6.19%). In contrast, improvements are marginal for categories such as Car (+0.44%), and slightly negative for Rifle (−0.26%).

The substantial gain in the Display category suggests that CPSA is particularly effective in modeling objects characterized by dominant global structures. Display objects typically consist of a large planar surface (screen) supported by a structured base. Such strong geometric regularities generate distinct frequency components, and the explicit phase modulation performed by CPSA appears to enhance the encoder's ability to capture these large-scale spatial configurations.

Interestingly, the Lamp category also demonstrates a considerable relative improvement (+9.0%). Although lamps may contain thin or curved components, they often exhibit a coherent global structure (e.g., base–stem–head configuration). The results suggest that CPSA improves the reconstruction of this overall structural arrangement, even if very fine sub-voxel details remain limited by the 32^3 voxel resolution.

Airplane objects, which contain clear bilateral symmetry and consistent structural elements such as wings and fuselage, also benefit from frequency-domain modeling. The ability of CPSA to modulate phase information likely contributes to improved global structural alignment in such symmetric categories.

In contrast, the Rifle category shows a slight performance decrease relative to the Baseline. Rifle objects typically contain elongated thin parts and highly intricate geometries that are sensitive to voxel discretization. Since CPSA operates at the representation level rather than increasing spatial resolution, improvements in extremely thin structures may remain constrained by voxel quantization effects.

Overall, the results suggest that CPSA primarily enhances global structural coherence and dominant shape consistency, rather than fine-scale geometric precision. Categories characterized by strong global layouts and structural regularity benefit the most from frequency-domain attention, while improvements remain bounded in cases dominated by sub-voxel thin geometry.

To further clarify the nature of the observed improvements, it is important to distinguish between global structural alignment and thin-part geometric precision. The IoU metric primarily captures volumetric overlap at the object scale and therefore reflects improvements in dominant structural components. Since CPSA operates by modulating frequency-domain phase information, which encodes spatial configuration, it is particularly effective in strengthening large-scale geometric coherence.

However, extremely thin structures remain constrained by the fixed 32^3 voxel resolution. Even minor discretization shifts may significantly affect volumetric occupancy in narrow components. Therefore, while CPSA improves overall IoU through enhanced global alignment, it does not fundamentally alter the resolution-dependent limitations of voxel-based representation. Incorporating surface-distance-based metrics or higher-resolution representations may provide further insight into thin-part reconstruction fidelity and represents an important direction for future work.

4 Conclusions

Single-view 3D reconstruction remains a fundamentally ill-posed problem due to missing depth information, structural ambiguity, and self-occlusion. In this study, we proposed Complex Phase-Shifting Attention (CPSA) as a frequency-domain attention mechanism integrated into the encoder of a 3D-VAE-GAN architecture. Unlike conventional spatial attention modules, CPSA explicitly modulates both the phase and amplitude components of feature representations in the frequency domain to generate a spatial attention mask. By leveraging phase information, which preserves spatial configuration, CPSA strengthens global structural modeling within voxel-based generative reconstruction. In addition, a Silhouette Augmentation (SA) strategy was introduced to reduce reliance on texture and color cues and to encourage geometry-focused learning.

Quantitative and qualitative evaluations on the ShapeNet 13-category subset demonstrate the effectiveness of the proposed CPSA + SA configuration. Based on the IoU metric, the model achieved an average IoU of 0.577 and obtained the highest performance in 11 out of 13 object categories. The largest relative improvement over the Baseline model was observed in the Display category, followed by Lamp and Airplane. The strong improvement in Display can be attributed to its dominant global geometric layout, while the notable gain in Lamp suggests that CPSA enhances reconstruction of coherent structural arrangements even in objects containing curved or moderately thin components. Improvements in Airplane further indicate that frequency-domain phase modulation benefits categories characterized by structural symmetry and consistent global shape organization. Qualitative analysis confirms that CPSA enhances global shape consistency while preserving structural integrity. Time-based evaluations reveal a moderate increase in training time due to additional frequency-domain operations, reflecting a measurable but controlled computational overhead.

It is important to clarify the scope of this contribution. The objective of this study is not to replace implicit, diffusion-based, or transformer-based reconstruction paradigms, but rather to investigate whether frequency-domain attention can systematically enhance encoder representations within a controlled voxel-based 3D-VAE-GAN framework. Within this defined scope, the results demonstrate that phase-aware frequency modulation provides a consistent improvement in global structural coherence for voxel-based single-view 3D reconstruction.

The present study has several limitations that should be acknowledged. First, experiments were conducted on the commonly used 13-category subset of the ShapeNet dataset. Although this subset ensures comparability with prior voxel-based reconstruction works, it does not cover the full diversity of the ShapeNet taxonomy, which contains 55 categories. Therefore, category selection bias may exist, and future studies should evaluate CPSA on a broader range of object classes.

Second, the voxel resolution was fixed at 32×32×32, following the standard 3D-VAE-GAN protocol to ensure fair comparison with prior voxel-based studies. This resolution inherently limits geometric fidelity, particularly for thin structures and high-frequency details. Although CPSA enhances encoder representations, it does not overcome discretization constraints imposed by voxel quantization. Evaluating CPSA under higher voxel resolutions (e.g., 64^3) or within hybrid implicit representations remains an important direction for future work to further assess scalability and high-fidelity reconstruction behavior.

Finally, the proposed Silhouette Augmentation (SA) assumes the availability of RGBA inputs with an alpha channel. While this design enables controlled geometric emphasis, it reduces direct applicability to standard RGB-only real-world datasets. Extending SA to RGB images via segmentation-based silhouette estimation represents an important direction for future research.

Given the observed trade-off between reconstruction performance and training time, computational efficiency should be systematically considered alongside accuracy improvements when proposing enhanced architectural components. Although the proposed CPSA-based architecture achieves superior reconstruction performance, it introduces a moderate increase in training time. Therefore, the selection of

such architectural enhancements should be aligned with application-specific requirements and computational constraints.

Voxel-based representations are inherently resolution-dependent. Increasing voxel resolution can improve geometric fidelity and enable finer structural detail in reconstructed objects. However, higher spatial resolution significantly increases memory consumption, including both RAM and GPU VRAM usage, as well as computational cost. Consequently, there exists a practical balance between geometric detail and resource requirements.

Beyond voxel representations, alternative 3D representations such as point clouds and mesh-based models may also be considered. Each representation paradigm offers distinct advantages and challenges, particularly in terms of compatibility with deep neural network architectures and computational efficiency. Ensuring that these representations are appropriately processed and integrated into learning frameworks remains a critical consideration for future research.

5 References

[1] LeCun Y, Bengio Y, Hinton G. "Deep Learning". *Nature*, 521(7553), 436-444, 2015.

[2] Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. "Generative Adversarial Nets". *Advances in Neural Information Processing Systems*, Montreal, Canada, 8-13 December, 2014.

[3] Kingma DP, Welling M. "Auto-Encoding Variational Bayes". arXiv, arXiv:1312.6114, 2013.

[4] Wu J, Zhang C, Xue T, Freeman B, Tenenbaum J. "Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling". *Advances in Neural Information Processing Systems*, 29, 2016.

[5] Choy CB, Xu D, Gwak J, Chen K, Savarese S. "3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction". *European Conference on Computer Vision*, Amsterdam, The Netherlands, 8-16 October, 628-644, 2016.

[6] Lun Z, Gadelha M, Kalogerakis E, Maji S, Wang R. "3D Shape Reconstruction from Sketches via Multi-view Convolutional Networks". *2017 International Conference on 3D Vision (3DV)*, Qingdao, China, 10-12 October, 254-263, 2017.

[7] Guo Y, Wang H, Hu Q, Liu H, Liu L, Bennamoun M. "Deep Learning for 3D Point Clouds: A Survey". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12), 4338-4364, 2020.

[8] Wu Z, Song S, Khosla A, Yu F, Zhang L, Tang X, Xiao J. "3D ShapeNets: A Deep Representation for Volumetric Shapes". *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA, 7-12 June, 19-27, 2015.

[9] Zhou Y, Tuzel O. "Voxelnet: End-to-end learning for point cloud based 3d object detection". *Proceedings of the IEEE conference on computer vision and pattern recognition*, Salt Lake City, UT, USA, 18-23 June, 4490-4499, 2018.

[10] Qi CR, Su H, Mo K, Guibas LJ. "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation". *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 21-26 July, 77-85, 2017.

[11] Qi CR, Yi L, Su H, Guibas LJ. "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space". *Advances in Neural Information Processing Systems*, Long Beach, CA, USA, 4-9 December, 5099-5108, 2017.

[12] Tan Q, Gao L, Lai Y-K, Yang J, Xia S. "Mesh-Based Autoencoders for Localized Deformation Component Analysis". *Proceedings of the AAAI Conference on Artificial Intelligence*, New Orleans, LA, USA, 2-7 February, 4011-4018, 2018.

[13] Ho J, Jain A, Abbeel P. "Denoising Diffusion Probabilistic Models". *Advances in Neural Information Processing Systems*, 6840-6851, 2020.

[14] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. "Attention is All you Need". *Advances in Neural Information Processing Systems*, Long Beach, CA, USA, 4-9 December, 5998-6008, 2017.

[15] Wang X, Girshick R, Gupta A, He K. "Non-Local Neural Networks". *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 18-23 June, 7794-7803, 2018.

[16] Hu J, Shen L, Sun G. "Squeeze-and-Excitation Networks". *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 18-23 June, 7132-7141, 2018.

[17] Woo S, Park J, Lee J-Y, Kweon IS. "CBAM: Convolutional Block Attention Module". *Proceedings of the European Conference on Computer Vision*, Munich, Germany, 8-14 September, 3-19, 2018.

[18] Gonzalez RC, Woods RE. *Digital Image Processing*. 3rd ed. Upper Saddle River, NJ, USA, Pearson Education India, 2009.

[19] Hendler J. "Avoiding Another AI Winter". *IEEE Intelligent Systems*, 23(2), 2-4, 2008.

[20] Shorten C, Khoshgoftaar TM. "A survey on Image Data Augmentation for Deep Learning". *Journal of Big Data*, 6(1), 1-48, 2019.

[21] Zhang H, Cisse M, Dauphin YN, Lopez-Paz D. "mixup: Beyond empirical risk minimization". arXiv, arXiv:1710.09412, 2017.

[22] Yun S, Han D, Oh SJ, Chun S, Choe J, Yoo Y. "CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features". *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Seoul, South Korea, 27 October-2 November, 6000-6009, 2019.

[23] Cubuk ED, Zoph B, Shlens J, Le QV. "Randaugment: Practical Automated Data Augmentation With a Reduced Search Space". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, Seattle, WA, USA, 14-19 June, 762-771, 2020.

[24] Chang AX, Funkhouser T, Guibas L, Hanrahan P, Huang Q, Li Z, Savarese S, Savva M, Song S, Su H. "ShapeNet: An Information-Rich 3D Model Repository". arXiv, arXiv:1512.03012, 2015.

[25] Xiang Y, Mottaghi R, Savarese S. "Beyond PASCAL: A benchmark for 3D object detection in the wild". *IEEE Winter Conference on Applications of Computer Vision*, Steamboat Springs, CO, USA, 24-26 March, 711-718, 2014.

[26] Funkhouser T, Kazhdan M, Shilane P, Min P, Kiefer W, Tal A, Rusinkiewicz S, Dobkin D. "Modeling by example". *ACM Transactions on Graphics (TOG)*, 23(3), 652-663, 2004.

[27] DeVries T, Taylor GW. "Improved Regularization of Convolutional Neural Networks with Cutout". arXiv, arXiv:1708.04552, 2017.

[28] Kipf TN, Welling M. "Semi-Supervised Classification with Graph Convolutional Networks". arXiv, arXiv:1609.02907, 2016.
[29] Wang N, Zhang Y, Li Z, Fu Y, Liu W, Jiang Y-G. "Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images". *Proceedings of the European Conference on Computer Vision*, Munich, Germany, 8-14 September, 54-71, 2018.
[30] Mescheder L, Oechsle M, Niemeyer M, Nowozin S, Geiger A. "Occupancy Networks: Learning 3D Reconstruction in Function Space". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 15-20 June, 4765-4775, 2019.
[31] Zhong Z, Zheng L, Kang G, Li S, Yang Y. "Random Erasing Data Augmentation". *Proceedings of the AAAI Conference on Artificial Intelligence*, New York, NY, USA, 7-12 February, 13001-13008, 2020.
[32] Park J, Florence P, Straub J, Newcombe R, Lovegrove S. "DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 15–20 June, 165-174, 2019.
[33] Mildenhall B, Srinivasan PP, Tancik M, Barron JT, Ramamoorthi R, Ng R. "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis". *Communications of the ACM*, 65(1), 99-106, 2021.
[34] Xie Y, Takikawa T, Saito S, Litany O, Yan S, Khan N, Sridhar S. "Neural Fields in Visual Computing and Beyond". *Computer Graphics Forum*, 41(2), 641-676, 2022.
[35] Nichol AQ, Dhariwal P. "Improved Denoising Diffusion Probabilistic Models". *International Conference on Machine Learning, Virtual Event*, 18–24 July, 8162-8171, 2021.
[36] Poole B, Jain A, Barron JT, Mildenhall B. "DreamFusion: Text-to-3D using 2D Diffusion". arXiv, arXiv:2209.14988, 2022.
[37] Tancik M, Srinivasan P, Mildenhall B, Fridovich-Keil S, Raghavan N, Singhal U, Ng R. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains". *Advances in Neural Information Processing Systems*, 33, 7537-7547, 2020.
[38] Ben-Hamu H, Maron H, Kezurer I, Avineri G, Lipman Y. "Multi-Chart Generative Surface Modeling". *ACM Transactions on Graphics*, 37(6), 1-15, 2018.
[39] Groueix T, Fisher M, Kim VG, Russell BC, Aubry M. "A Papier-Mâché Approach to Learning 3D Surface Generation". *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 18–23 June, 216-224, 2018.