Unpaired 3D Shape-to-Shape Translation via Gradient-Guided Triplane Diffusion

Wenxiao Zhang, Hossein Rahmani, Jun Liu*

Abstract—Unpaired shape-to-shape translation refers to the task of transforming the geometry and semantics of an input shape into a new shape domain without paired training data. Previous methods utilize GAN-based architectures to perform shape translation, employing adversarial training to transform the source shape encoding into the target domain in the low-dimensional latent feature space. However, these methods encounter difficulties in generating diverse and high-quality results, as they often suffer from issues such as "mode collapse". This leads to limited generation diversity and makes it challenging to find an accurate latent code that adequately represents the input shape. In this paper, we achieve unpaired shape-to-shape translation via a triplane diffusion model, in which we factorize 3D objects into triplane representations and conduct a diffusion process on these representations to accomplish shape domain transformation. We observe that by adding an appropriate amount of noise to an input object during the forward diffusion process, domain-specific shape structures are smoothed out while the overall structure is still preserved. Subsequently, we progressively remove the noise via an unconditional diffusion model trained on the target shape domain in the reverse diffusion process. This allows us to obtain a denoised output that retains the structural similarities of the source input while aligning with the distribution of the target shape domain. During this process, we propose two gradient-based guidance mechanisms to guide the translation process to guarantee more faithful results during the denoising process. We conduct extensive experiments on different shape domains, and the experimental results demonstrate that our method achieves superior shape fidelity with high quality compared to current state-of-the-art baselines.

Index Terms—Shape Translation, Shape Modeling, Diffusion Model.

1 Introduction

SHAPE-to-shape translation is a frequently encountered fundamental task in computer graphics and geometric modeling. It involves transforming one shape into another, preserving its structure or topology variation, particularly in the context of 3D modeling and animation. However, acquiring paired data for 3D modeling tasks poses a significant challenge. As geometric deep learning gains attention in the graphics community, it becomes pertinent to explore whether learning-based approaches can achieve shape transformations without relying on direct correspondences between shapes in the source and target domains.

The problem of unpaired domain translation in 2D images has garnered considerable attention from researchers in computer vision and computer graphics. Early successful models like [1] have employed cycle-consistency loss to enable bidirectional translation between two domains. Building on this work, subsequent models such as STARGAN [2], SEAN [3], U-GAT-IT [4], and CUT [5] have been proposed to enhance the quality and diversity of generated images. However, these techniques primarily focus on transferring stylistic image features and have not proven effective for 3D shape transformation.

P2P-NET [6] was the first proposed method to learn general-purpose shape transformations on point clouds through point displacements, but it requires paired shapes from two domains for training. More recently, advancements have been made in learning-based methods for unpaired shape translation. Yin et al. introduced LOGAN [7], a shape translation network that employs a generative adversarial network operating in the latent space. LOGAN enforces cross-domain translation through an adversarial loss and ensures the preservation of shape features for natural shape transformations using a feature preservation loss. However,

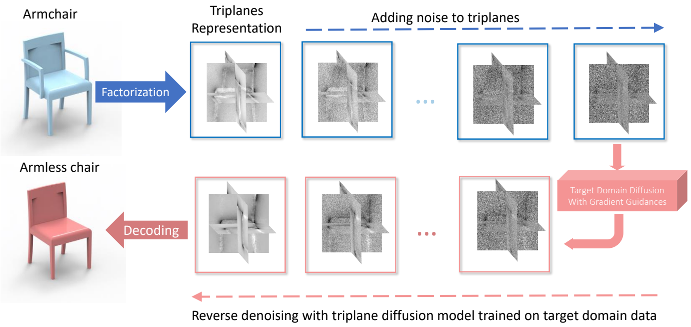
Armchair
Triplanes Representation
Adding noise to triplanes
Factorization
Armless chair
Target Domain Diffusion With Gradient Guidances
Decoding
Reverse denoising with triplane diffusion model trained on target domain data

Fig. 1: We propose an unpaired 3D shape-to-shape translation method through a gradient-guided triplane diffusion model. The input source object is first factorized into triplane representation. We add noise to source triplanes with the forward diffusion process, where domain-specific shape structures will be progressively smooth out while the overall shape structure will be preserved. We then denoise these noisy triplanes via a diffusion model trained on the target domain with our designed gradient-based guidances. Consequently, the denoised output exhibits similarity to the source object in terms of overall structure, while conforming to the distribution of objects in the target domain.

- *Wenxiao Zhang is with the School of Information Science and Technology, University of Science and Technology of China, Hefei, 230026, China. E-mail: wenxxiao.zhang@gmail.com*
- *Jun Liu and Hossein Rahmani are with School of Computing and Communications, Lancaster University, LA1 4YW Lancaster. E-mail: j.liu81@lancaster.ac.uk, h.rahmani@lancaster.ac.uk.*

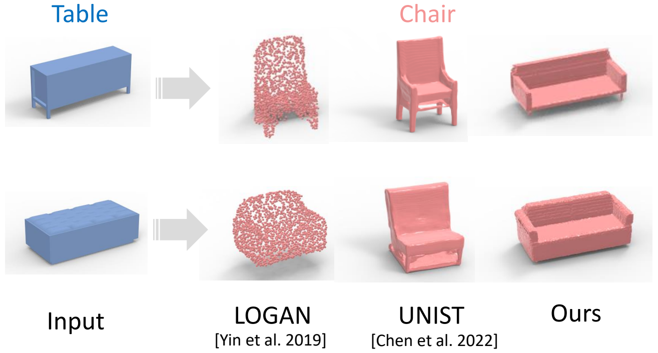

Fig. 2: Illustration of the "mode collapse" issue of previous translation methods with Table → Chair examples. LOGAN [7] and UNIST [8] fail to generate an ideal chair that accurately represents the input chair. In contrast, our method successfully achieves faithful results that closely resemble the input shape.

LOGAN has limitations in handling high-resolution point clouds, often resulting in low-quality 3D translations. To overcome this challenge, UNIST [8] was developed, incorporating position-aware latent grids and implicit representations to generate higher-quality results. It is based on autoencoding implicit fields and trained using a similar adversarial loss function as LOGAN.

While LOGAN and UNIST have made progress in unpaired shape translation, there are still challenges that hinder translation quality. First, both LOGAN and UNIST rely on GAN-like architectures with adversarial training. Adversarial training is used to model the high-dimensional shape space and map it to a low-dimensional latent space, but it is prone to issues such as "mode collapse". This occurs when the discriminator becomes too good at distinguishing between real and fake samples, causing the generator to produce limited or low-quality samples, failing to explore the entire data distribution. Consequently, the generator fails to find a latent code that faithfully represents the input shape. We illustrate the "mode collapse" issue in Figure 2 where we show a comparison between our method and the previous method in Table → Chair translation. Second, LOGAN and UNIST convert the input shape into low-dimensional latent representations, such as latent vectors or latent grids, which struggle to capture rich shape patterns and often result in blurry outputs. For example, LOGAN transfers the input object into a global latent vector followed by a point cloud generator. UNIST utilizes a latent grid representation to better capture spatial features during translation, but the resolution of the latent grid is still limited in capturing detailed shape information.

In this paper, we propose a novel unpaired shape-to-shape translation method via a gradient-guided triplane diffusion model. We factorize the input source object into an effective triplane representation [9], and conduct a shape translation via a diffusion process. The main idea behind our method is illustrated in Figure 1. Our observation is that by adding appropriate noise during the forward diffusion process, domain-specific shape structures are gradually smoothed out while the overall structure of the input shape is still preserved. Subsequently, we progressively remove the noise via an unconditional diffusion model trained on target shape domain data with gradient-based guidances. This allows us to obtain a denoised output that retains the structural similarities of the source input while aligning with the distribution of the target shape domain.

In comparison to previous shape translation methods such as LOGAN and UNIST, our method offers several advantages. Firstly, instead of relying on a discriminator network for feedback, our method is based on a diffusion model that operates progressively, enabling the exploration of the entire data distribution and the generation of diverse samples. Secondly, instead of converting the input shape into latent vector or latent grid representations, our approach utilizes triplane representations, which can effectively and efficiently capture 3D-aware shape structures. Furthermore, since triplane representations share a similar data format with 2D images, thus we could incorporate the pre-existing 2D diffusion techniques when conducting the diffusion process for triplane representations.

To facilitate the translation process during reverse triplane diffusion, we introduce two gradient-based guidance mechanisms. Firstly, we introduce a structure preservation guidance that promotes the retention of the overall structure of the source input during denoising. Secondly, we propose a classifier guidance that encourages the denoised object to closely conform to the target domain object distribution.

Though our method could draw inspiration from 2D diffusion techniques [10], [11] for translation, directly treating the triplane representations as 2D images and directly applying these 2D techniques is non-trivial and presents significant challenges. The main difficulty arises from the fact that triplane images are fundamentally different from natural 2D images, as they are represented as separate planes from different axes. Simply concatenating the triplane feature maps from different axes can result in suboptimal performance, as it neglects the inherent spatial relationships among these feature maps. To overcome this challenge, we design a cross-plane convolutional layer that we have incorporated into the existing 2D diffusion model. This proposed cross-plane convolution layer is designed to enhance the transmission of information between different planes, ensuring an effective diffusion process.

Thorough evaluations of our approach demonstrate that our method is capable of producing more faithful shapes compared to other methods, while preserving the fundamental geometric structures of the original input shape.

The main contributions of our work can be summarized as:

- In contrast to previous GAN-based methods for 3D shape-to-shape translation, our approach introduces a novel paradigm that utilizes a triplane diffusion model on triplane representations to achieve progressive 3D shape translation, generating high-quality transformed shapes with high fidelity and better original structure preservation.
- We introduce two gradient-based guidance mechanisms that capture structural features and domain-specific signals, respectively. These mechanisms aim to encourage the translated shape to retain the original geometric structure and enhance its alignment with the distribution of data in the target domain.
- Through extensive experiments, we demonstrate that our method achieves state-of-the-art performance in shape-to-shape translation across various shape domains. Our approach is capable of transforming shapes with high quality and fidelity, showcasing its effectiveness in shape translation tasks.

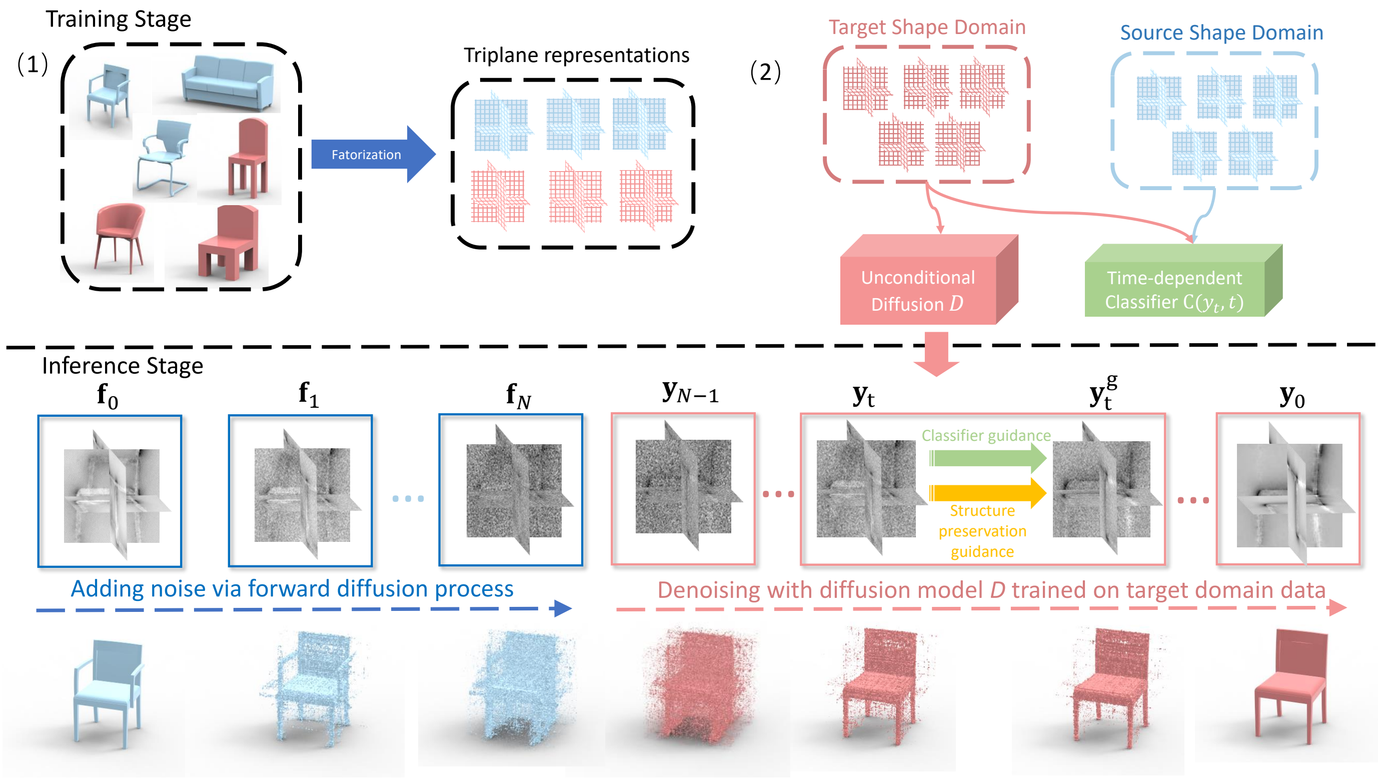

Fig. 3: Method overview. Given the source and target objects datasets, we factorize them into triplane representations. Before we perform shape translation, we use the triplane representations of target objects to train a diffusion model D, and train a time-independent binary classifier C on the entire triplane representations which classifies whether an object is from the source domain or target domain. Denoting the triplane image of the input object as $\mathbf{f}_0$, we map the input triplane image $\mathbf{f}_0$ to the target domain by running the forward process followed by the reverse process with the diffusion model D. We propose two guidance mechanisms that are applied in each step of the denoising process. One is a structure preservation guidance to guarantee structure preservation. For another guidance, we leverage the pre-trained classifier C as a classifier guidance to ensure the object could be completely translated into the target domain.

2 Related Work

The problem of shape-to-shape translation is a significant concern in the realm of visual data processing, and there exists a vast body of literature dedicated to this topic. In traditional shape translation, the primary objective is to deform the source shapes into target shapes while maintaining the original structure or topology variation with corresponding parts. In this work, we are specifically focused on developing learning-based approaches for 3D cross-domain shape transforms. We provide a comprehensive review of relevant methods from the fields of graphics and vision that are most pertinent to our research.

Image Style Translation. The style translation of images to other images can be accomplished in a paired or unpaired supervision manner. Pix2pix [12] is a typical paired approach that involves using a conditional GAN with a reconstruction loss. In contrast, unpaired translation is more challenging and more applicable in real-world scenarios. [13] introduce the first unsupervised I2I translation method, and [14] introduces the multimodal unsupervised I2I translation. Most unpaired methods are based on GAN and rely on cycle consistency, including CycleGAN [1], DualGAN [15], U-GAT-IT [4], and the recent UVCGAN [16]. This class of algorithms requires two generator networks that translate images in opposite directions. ACLGAN [17] attempts to relax the cycle-consistency constraint and replace it with a weaker adversarial one. CUT [5] takes an alternative route and uses a contrastive loss to maximize the information between the source and the translated images. More recently, multiple works have attempted to employ diffusion models for unpaired image-to-image translation. For instance, ILVR [18] achieves an unpaired image translation by modifying the standard Gaussian denoising process. SDEdit [10] uses a source image perturbed by Gaussian noise as a seed image and runs the standard diffusion process on top of it. EGSDE [10] introduces a special energy function to guide the denoising process. However, all these GAN-based or diffusion-based methods focus on transferring stylistic image features and have not been successful in transforming the shapes of contents.

Shape Translation. Learning-based shape-to-shape translation has been first studied by P2P-Net [6], which operates on point clouds via point displacements with paired supervision. Following P2P-Net, LOGAN [7] is the first deep model proposed for general-purpose, unpaired shape-to-shape translation. LOGAN is based on CycleGAN, which encodes shapes from both input domains into a common latent space and performs shape translations in that space. [19] propose a similar cycle-gan-based point cloud transformation method with a novel autoencoder and loss function for preserving shape characteristics. Inspired by LOGAN, UNIST [8] is another unpaired shape-to-shape translation work. Differing from LOGAN, UNIST employs implicit representations that can generate topology-varying shape translations instead of point clouds, and use position-aware latent grids rather than holistic latent codes that only encode global features. UNIST also uses a similar set of GAN-based losses as LOGAN, such as adversarial loss, and cycle-consistency loss. The majority of GAN-based shape-to-shape translation is that they suffer from the mode collapse issue, which means that they may fail to find a latent representation that faithfully represents the input shape due to the limited sampling diversity.

Diffusion Models on 3D generation. In recent years, diffusion models have emerged as an effective method for learning a data distribution that can be easily sampled from. [20] introduced these models for generating images, and since then, several works [21], [22] have simplified and accelerated the approach.

Diffusion models have also been applied to 3D fields. Early works of 3D diffusion models deal with point cloud data [23], [24]. Due to the high freedom degree of regressed coordinates, it is always difficult to obtain clean manifold surfaces via post-processing. Alternatively, researchers turn to neural field representations which are generally more suitable than point clouds for 3D shape generation. NeuralWavelet [25] employs the wavelet transform to encode shapes into the frequency domain. Subsequently, diffusion models are trained on the frequency coefficients to generate shapes. Additionally, recent concurrent works [26], [27], [28], [29], [30], [31], [32] have explored latent diffusion models for SDF (Signed Distance Field) and occupancy generation. These approaches involve training an SDF autoencoder to establish a latent space similar to latent-GAN. Subsequently, a diffusion model is trained to generate the latent code, which can then be transformed into an SDF by using a pre-trained decoder. Some other works focus on leveraging 2D diffusion for 3D generation [33], [34], [35], [36], [37], such as using multi-view 2D images by considering the view consistency. NFD [38] extracts triplane feature representations for 3D objects and treats the triplanes as 2D images, which enables the direct application of existing 2D diffusion architecture to 3D generation.

Triplane Representation. The triplane representation is a hybrid explicit–implicit network architecture for neural fields that are particularly efficient to evaluate, which is first proposed in [39]. This representation uses three 2D feature planes to represent features from different dimensions, and a multilayer perceptron-based decoder for interpreting features sampled from the planes. A 3D coordinate is queried by projecting it onto each of the axis-aligned planes, querying and aggregating the respective features, and decoding the required resulting feature for downstream tasks. Triplane representation is proven for its effectiveness in several downstream tasks, such as 3D-aware image synthesis [9], 3D generation [38], [40], [41], [42], [43], [44], and nerf-related works [45], [46], [47]. Instead of directly training the diffusion model on popular 3D data formats, such as point clouds or SDF voxels, our method leverages a diffusion model trained on triplane representations to generate high-quality 3D presentations.

3 Preliminary of diffusion model

In this section, we present a concise introduction to the theoretical foundations of Denoising Diffusion Probabilistic Models (DDPM), as our method relies on the diffusion process. DDPM is built upon the concept of modeling the temporal diffusion of noise and employing a denoising function to eliminate the noise from observed data. For instance, in the context of image generation, DDPM learns to generate high-quality images by iteratively denoising a sequence of noisy images. First, it defines a diffusion process for generating noisy images $\mathbf{x}_t$ given a clean image $\mathbf{x}_0$:

$$q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right) := \mathcal{N}\left(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}\right), \tag{1}$$

where β_t is a scheduler that determines the level of noise added at each iteration. The forward process gradually introduces Gaussian noise to the initial image $\mathbf{x}_0$ through a series of T time steps, yielding a sequence of noisy images $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T$. For 2D image diffusion, noise is added to individual pixels within images.

The reverse process of the diffusion model is parameterized by a neural network, typically a convolutional network, which estimates the mean $\mu_\theta(\mathbf{x}_t, t)$ and variance $\mathbf{\Sigma}_\theta(\mathbf{x}_t, t)$ with learnable parameter θ. These parameters are optimized to progressively eliminate the noise from the initial image $\mathbf{x}_T$ through iterative denoising. The denoising process can be defined as follows:

$$p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right) := \mathcal{N}\left(\mathbf{x}_{t-1}; \mu_\theta\left(\mathbf{x}_t, t\right), \sigma_t^2\left(\mathbf{x}_t, t\right) \mathbf{I}\right). \tag{2}$$

The DDPM model is optimized by calculating the variational bound of the negative log-likelihood, specifically $\mathbb{E}\left[-\log p_\theta\left(\mathbf{x}_0\right)\right]$. The noise prediction network $\boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t, t\right)$ can be optimized by function:

$$L(\theta) = \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}, t}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t, t\right)\right\|^2\right]. \tag{3}$$

After the optimization, we are able to sample from the learned Gaussian transitions $p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)$ by:

$$\mathbf{x}_{t-1} = \boldsymbol{\mu}_\theta\left(\mathbf{x}_t, t\right) + \mathbf{\Sigma}_\theta^{1/2}\left(\mathbf{x}_t, t\right)\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{4}$$

By iteratively conducting forward and backward passes, DDPM can effectively learn to generate high-quality images that exhibit both similarity to the training data and consistency with the diffusion process.

4 Method

We present an overview of our method in Figure 3. Our approach begins by decomposing each individual object into triplane representations. Subsequently, a 2D DDPM diffusion model D is trained on target shape domain data using the corresponding triplane representations. Finally, we perform a guided shape translation using the pre-trained triplane diffusion model D.

This section is organized as follows to present our method: In Section 4.1, we elaborate on the procedure of obtaining triplane feature representations for 3D objects and how we train a 2D diffusion model on the obtained triplane representations. Following that, Section 4.2 provides a comprehensive explanation of our shape translation method, which leverages the triplane diffusion process. Finally, in Section 4.3, we describe our proposed guidance mechanisms and how they facilitate the translation process.

4.1 Triplane Diffusion

Learning triplane representations on occupancy fields. In our approach, the original 3D objects are represented implicitly using occupancy fields [48], which have been introduced as continuous and expressive 3D scene representations. For any 3D location $p \in \mathbb{R}^3$ on an object, an occupancy function $\mathbf{OF}(\cdot) : \mathbb{R}^3 \to \{0, 1\}$ will output a binary value indicating whether a 3D location p is inside or outside an object. We utilize a trainable Multilayer Perceptron (MLP) to parameterize the occupancy function $\mathbf{OF}(\cdot)$.

Since we cannot directly apply a diffusion process to the implicit occupancy fields, which are not in a discrete data format, we involve a factorization step to convert them into triplane representations. Triplane is an explicit-implicit representation first proposed in EG3D [9], striking a desirable balance between efficiency and quality in multi-view-consistent image generation. Triplane can be viewed as a collection of 2D feature maps $\mathbf{f_{xz}}, \mathbf{f_{xy}}, \mathbf{f_{yz}} \in \mathbb{R}^{H \times W \times C}$ that represent a 3D object. Each 2D plane has a resolution of $H \times W$ and C channels. To obtain a 3D coordinate triplane feature **Tri(p)** for a given 3D location $\mathbf{p} \in \mathbb{R}^3$, we project it onto the axis-aligned planes ($x-y$, $x-z$, and $y-z$ planes) and sum the respective features:

$$\mathbf{Tri(p)} = \mathbf{f}_{xy}(\mathbf{p_{x,y}}) + \mathbf{f}_{yz}(\mathbf{p_{y,z}}) + \mathbf{f}_{xz}(\mathbf{p_{x,z}}). \tag{5}$$

The occupancy function $\mathbf{OF}(\cdot)$ is represented by a lightweight multilayer perceptron (MLP_ϕ) with trainable parameters ϕ. This MLP decodes the resulting triplane feature to determine the final occupancy value at the given 3D location $\mathbf{p}$:

$$\mathbf{OF}(\mathbf{p}) = \mathrm{MLP}_\phi\left(\mathbf{Tri(p)}\right). \tag{6}$$

To obtain the triplane representation for each object, we follow a similar process as in NFD [38], where we consider the triplane feature maps $\mathbf{f_{xz}}, \mathbf{f_{xy}}, \mathbf{f_{yz}}$ as learnable parameters. We optimize both the triplane feature maps $\mathbf{f_{xz}}, \mathbf{f_{xy}}, \mathbf{f_{yz}}$ and the MLP_ϕ simultaneously. The training objective is a straightforward L_2 reconstruction loss between the predicted occupancy values $\mathbf{OF}\left(\mathbf{p}_j^{(i)}\right)$ and the ground-truth occupancy values $\mathrm{O}_j^{(i)}$ for each 3D location. Here, $\mathbf{p}_j^{(i)}$ represents the j-th point of the i-th object:

$$\mathcal{L}_{\text{recon}} = \sum_i^I \sum_j^J \left\| \mathbf{OF}\left(\mathbf{p}_j^{(i)}\right) - \mathrm{O}_j^{(i)} \right\|_2. \tag{7}$$

Thus we can jointly optimize ϕ along with the triplane feature maps for each object in the training dataset:

$$\left\{\phi, \mathbf{f}_{xy}^{(i)}, \mathbf{f}_{xz}^{(i)}, \mathbf{f}_{yz}^{(i)}\right\} = \underset{\left\{\phi, \mathbf{f}_{xy}^{(i)}, \mathbf{f}_{xz}^{(i)}, \mathbf{f}_{yz}^{(i)}\right\}}{\arg\min} \mathcal{L}_{\text{recon}}. \tag{8}$$

The training process for obtaining triplane representations is depicted in Figure 4. Initially, the triplane feature maps and the occupancy function $\mathbf{OF}(\cdot)$ are jointly optimized to capture the occupancy field of a subset of the training dataset. Subsequently, we fix the pre-trained decoder MLP_ϕ and extract the triplane representations for the remaining objects in the training dataset.

Training Triplane-aware diffusion model. Once we have obtained the respective triplane representations for each object, we proceed to train a diffusion model on triplane representations using available existing 2D diffusion architectures.

A direct way to conduct diffusion on triplane is to concatenate $\mathbf{f_{xz}}, \mathbf{f_{xy}}, \mathbf{f_{yz}} \in \mathbb{R}^{H \times W \times C}$ into $\mathbf{f} = [\mathbf{f_{xz}}, \mathbf{f_{xy}}, \mathbf{f_{yz}}] \in \mathbb{R}^{H \times W \times 3C}$, resulting in a concatenated triplane feature map $\mathbf{f}$ that can be treated as a 2D image with $3C$ channels (Figure 5 (a)), which is similar to the way in NFD [38]. Then we could directly apply the existing 2D diffusion model to it.

However, simply concatenating the feature maps from different axes can lead to poor performance [44], [49], as it ignores the intrinsic spatial relationships among these feature maps. To address this issue, we have developed a cross-plane convolutional layer and inserted it into the current 2D diffusion model.

The designed cross-plane convolution layer is depicted in Figure 5 (b). Here, we denoise each triplane feature map separately but concatenate axis-related information from the other planes through an axis-aligned aggregation process. We show the axis-aligned aggregation operation in Figure 5 (c). Specifically, we exact and concatenate the axis-related features of other planes, by performing average pooling across the feature maps along the shared axis. Though each of the three planes is denoised respectively, the 2D convolution used in the cross-plane convolution layer shares the same weights.

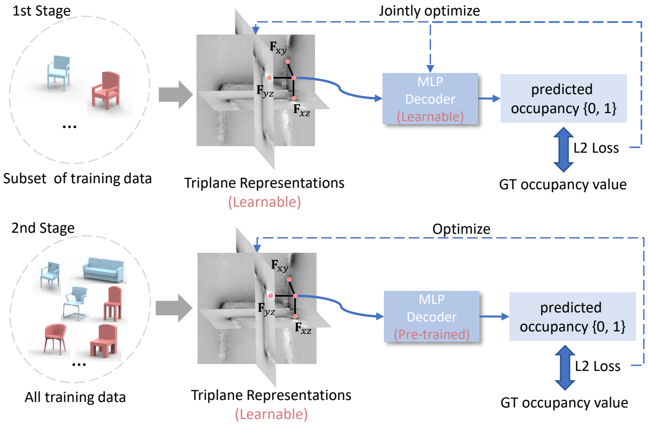

Fig. 4: Triplane obtaining. In the first stage, the triplane representations and the MLP decoder are jointly optimized with a subset of the training data to guarantee that the output occupancy values from the triplane representations are correctly predicted. In the second stage, we fixed the pre-trained MLP decoder, and extract the triplane representations of all the objects in the training dataset.

We choose the diffusion architecture from [50] as our baseline, but replace all the 2D convolutional layers in the UNet autoencoder, which are part of the diffusion model, with our specially designed cross-plane convolutional layers.

Building upon the preliminary of the diffusion model discussed in Section 3, we can train an unconditional DDPM model using the normalized triplane feature maps. We denote a triplane image as $\mathbf{f}$, analogous to a natural 2D image $\mathbf{x}$. The diffusion model is trained on triplane maps $\mathbf{f}_{0,\ldots,T} \in \mathbb{R}^{N \times N \times 3C}$, where $\mathbf{f}_0$ represents a clean triplane image from the training dataset, and $\mathbf{f}_T$ corresponds to a completely noisy triplane image sampled from a Gaussian distribution. The training objective aims to minimize the mean-squared error loss, as described in Equation 3, between the predicted noise $\epsilon_\theta\left(\mathbf{f}_t, t\right)$ and the actual noise ϵ present in $\mathbf{f}_t$.

Before we feed the triplanes into the diffusion model, we normalize each channel using the mean and variance of the entire triplane dataset. This normalization ensures that each channel has a zero mean and a standard deviation of 0.5. In NFD [38], each channel value of the triplane image is clipped to ensure it remains within a certain standard deviation of the mean. However, in our experiments, we observed that there are extreme pixel channel values in the triplane feature image. Instead of directly clipping these extreme values, which may disrupt the object shape structure, we employ a median filter to smooth these extreme values by considering neighboring pixel values within the corresponding channel.

4.2 Progressive Triplane Diffusion Translation

In this section, we describe our approach for conducting shape-to-shape translation using a diffusion process on triplane feature maps.

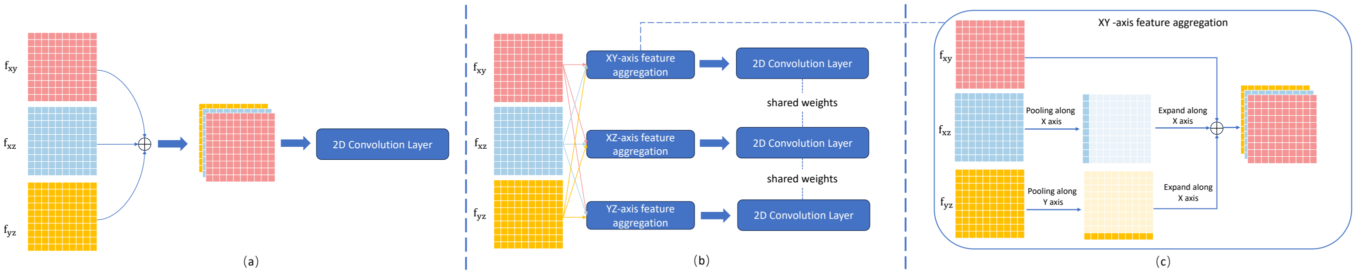

Fig. 5: (a) NFD [38] directly concatenating the triplane feature maps and feeding it to the 2D diffusion model. (b) Our designed cross-plane convolution layer. (c) Illustration of the axis-aligned feature aggregation operation.

We depict the method pipeline in Figure 3, where we begin with factorizing both the source and target object dataset into triplane representations following the aforementioned rules. During the training stage, we exclusively use the triplane maps of target objects to train a diffusion model D. Additionally, we train a time-independent binary classifier C on the entire dataset, which serves as classifier guidance for the diffusion process described in Section 4.3.

In the inference stage, given a source input object, we perform a progressive shape translation via a forward and backward diffusion process. In the forward diffusion process, we gradually add noise to the input triplane feature image $\mathbf{f}_0$. We perform N forward steps, where the obtained triplane sequences are denoted as $\mathbf{f}_0, \mathbf{f}_1, \cdots, \mathbf{f}_N$. N is a hyper-parameter that controls the level of noise added to the input triplane feature image. In the reverse process, we progressively remove noise for N steps to get the denoised triplane sequence $\mathbf{y}_{N-1}, \mathbf{y}_{N-2}, \cdots, \mathbf{y}_0$ using the diffusion model D which is trained on the target object triplanes. Our motivation stems from the expectation that the generated triplane image $\mathbf{y}_0$ would exhibit a bias towards the distribution of the target object, as the diffusion model D is trained on the target shape domain data. Also, $\mathbf{y}_0$ could preserve the source input structure as it is denoised from $\mathbf{f}_N$ which contains the fundamental structure information of $\mathbf{f}_0$.

Although we find this process can successfully transfer the source shape to the target distribution, the selection of the optimal diffusion step N is critical and involves a trade-off. A small value of N may lead to an incomplete translation from the source shape to the target shape distribution due to insufficient diffusion steps. Conversely, a large value of N may cause the loss of the original input structure, resulting in an output that significantly deviates from the source input.

To better transfer the shape of the source object to the target object while maintaining its overall geometric structure, we incorporate two gradient-based guidance mechanisms to enhance the shape-to-shape translation during the denoising translation process. These mechanisms are designed to provide additional support and improve the effectiveness of the translation process.

4.3 Gradient-based Translation Guidance Mechanisms

Structure preservation guidance. To get better fidelity of the shape translation results, a key consideration is to preserve the original structure or topology of the source object during the translation.

We introduce a structure preservation guidance that regularizes the diffusion process to better preserve the original structure of the source input. In particular, we introduce a structural feature filter $Filter(\cdot) : \mathbb{R}^{H\times W\times 3C} \rightarrow \mathbb{R}^{H\times W\times 3C}$, which is a low-pass filter inspired by [10], [18]. The low-pass filter performs a traditional low-pass filtering operation, which is a fixed, non-learnable process. Intuitively, this low-pass filter will retain the overall geometric structures of a triplane image, which is illustrated in Figure 6. Building upon it, we compute the squared L_2 distance between the filtered triplane feature maps from the denoised sample $\mathbf{y}_t$ and the noisy sample $\mathbf{f}_t$ as follows:

$$\mathcal{L}_{str} = \|Filter(\mathbf{y}_t) - Filter\,(\mathbf{f}_t)\|_2^2\,. \tag{9}$$

We regard the current triplane $\mathbf{y}_t$ as learnable parameters, and optimize it by backpropagating the gradients according to $\mathcal{L}_{str}$. We update $\mathbf{y}_t$ according to the gradients:

$$\mathbf{y}_t \leftarrow \mathbf{y}_t - \eta_{str} \nabla_{\mathbf{y}_t} \mathcal{L}_{str}(\mathbf{y}_t, t), \tag{10}$$

where η_{str} is the updating rate.

Intuitively, minimizing the loss defined in Equation 9 encourages the transferred triplane image $\mathbf{y}_t$ to retain its structural features, thereby enhancing its faithfulness to the source object.

In [10], similar guidance mechanisms are employed in each reverse step $[N, .., 0]$ of the denoising process to preserve the object outline in image style translation tasks. In contrast, we only apply the structure preservation guidance in the reverse steps from $[N, .., S]$. This is motivated by the fact that the $Filter(\cdot)$ will incorporate more domain-specific structures as the triplane image becomes clearer during the denoising process. For instance, when translating from an armchair to an armless chair, the armrest structure of the armchair captured by $Filter(\mathbf{f}_t)$ becomes increasingly clear during denoising. However, this domain-specific structure is not desirable to preserve in shape-to-shape translation tasks, as demonstrated in the ablation study in Section 6.5.

To this end, we soften the structural guidance by limiting its application to denoising steps $[N, .., S]$, where S is a hyper-parameter that controls the strength of the structural guidance.

Classifier Guidance. Another form of guidance that we introduce is classifier guidance. This type of guidance is designed to encourage the denoised triplane feature map to become closer to the distribution of the target domain during the reverse process.

We involve a time-dependent binary classifier, represented as $\mathcal{C}(\mathbf{f}, t) : \mathbb{R}^{H\times W\times 3C} \times \mathbb{R} \rightarrow \mathbb{R} \in \{0, 1\}$, which determines if a noisy triplane image $\mathbf{f}$ belongs to the source or target domain. This classifier is trained on both the source and target domain objects using noisy object triplane sequences $(\mathbf{f}_0, \cdots, \mathbf{f}_t, \mathbf{y}_0, \cdots, \mathbf{y}_t)$ generated during the forward diffusion process, where the classifier loss is denoted as $\mathcal{L}_{cls}(\mathbf{f}, t)$. After we have finished the classifier training, we leverage this pre-trained classifier to optimize the current triplane $\mathbf{y}_t$ by treating it as learnable parameters and

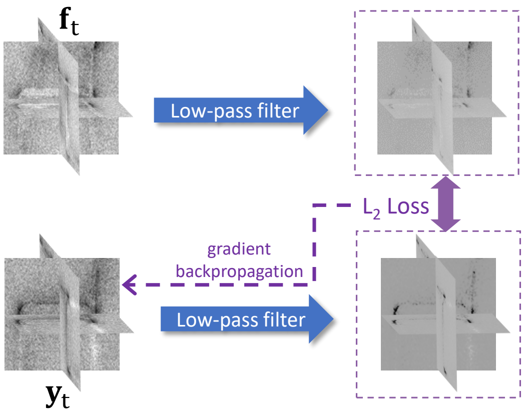

Fig. 6: Illustration of the proposed gradient-based structure preservation guidance. To preserve the overall structure of $\mathbf{f}_t$, we employ a low-pass filter to extract the structural information from both the triplane image $\mathbf{f}_t$ and the translated triplane image $\mathbf{y}_t$. Subsequently, we update $\mathbf{y}_t$ by minimizing the discrepancy between the filtered triplane maps to align the structural characteristics of the translated shape with those of the input shape.

backpropagating the gradients based on $\mathcal{L}_{cls}$, where $\mathbf{y}_t$ is assigned the label of 1, indicating it belongs to the target domain. We could update $\mathbf{y}_t$ according to the gradients:

$$\mathbf{y}_t \leftarrow \mathbf{y}_t - \eta_{cls} \nabla_{\mathbf{y}_t} \mathcal{L}_{cls}(\mathbf{y}_t, t), \tag{11}$$

where η_{cls} is the updating rate. In our implementation, we utilize the same classifier architecture proposed in [50].

In summary, for every reverse time step t, we refer to the $\mathbf{y}_t$ with two guidances applied as $\mathbf{y}_t^g$, which is represented as:

$$\mathbf{y}_t^g = \mathbf{y}_t - \eta_{cls} \nabla_{\mathbf{y}_t} \mathcal{L}_{cls}(\mathbf{y}_t, t) - \eta_{str} \nabla_{\mathbf{y}_t} \mathcal{L}_{str}(\mathbf{y}_t, t). \tag{12}$$

With our proposed guidance mechanisms, we can provide additional support to enhance the fidelity and quality of the translation results.

Equation 12 is applied only once in each time step and we adjust the hyperparameter η_{cls} and η_{str} to control the effect of these two guidances. This is the same as applying it for more than one iteration steps to control the effects of the two guidances.

5 Implementation and Training Details

3D object pre-processing. To prepare the 3D objects in the dataset, we employ a series of pre-processing steps. Initially, we utilize ManifoldPlus [51] to convert the objects into watertight meshes. Subsequently, we leverage the approach described in [48] to calculate occupancy values for arbitrary 3D coordinates. We involve a sampling strategy to sample 500K query points, where 250K query points are uniformly and randomly distributed throughout the volume, and the other 250K query points are sampled near the surface of the watertight mesh.

Triplane representation obtaining. We follow the pipeline in [38], where we used a triplane feature map of dimension 128×128×32×3 for each object. The triplane feature maps are initialized with Gaussian noise having a mean of zero and a standard deviation of 0.1. The occupancy function $\mathbf{OF}(\cdot)$ is implemented using an MLP layer, consisting of a Fourier feature mapping layer with a scale factor of 1, followed by three fully connected layers of dimension 128, each employing ReLU activation functions.

The training methodology for triplanes and MLP comprises two stages. In the first stage, we jointly train them on a subset of the data, while in the second stage, we focus on learning the triplane feature maps in the dataset while keeping the MLP frozen. For the initial stage, we select 200 shapes and train with a batch size of 1 object per iteration, incorporating all 500K occupancy value points per object. This stage is trained for 200 epochs using a learning rate of 1e-3 on a single A5000 GPU. The training process for a single diffusion model takes approximately 3 hours to complete. In the subsequent stage, the shared MLP is frozen, and we individually train the triplane feature maps for each object in the dataset. During this stage, we train the triplane representations for 30 epochs with a learning rate of 1e-3. The learned triplane feature maps serve as pseudo-ground truth images for the subsequent triplane diffusion model training. Notably, triplane diffusion models are trained separately on each shape domain.

Triplane diffusion training. For the training of the diffusion models, we adopt a similar setup as described in [38], utilizing the implementation of the 2D DDPM diffusion model presented in [50]. Unless explicitly mentioned, we use the same set of hyperparameters as the class-specific LSUN model outlined in [50]. In the cross-plane convolution layer, we use the same 2D convolution layer in the original 2D diffusion model, so there are no extra learnable parameters.

The training of all diffusion models consists of 200K steps with a learning rate of 1e-4. Before we train the diffusion model, we perform a normalization step on the triplane maps. Specifically, we center the feature channels around zero means and clip the standard deviation of each channel within 16. Subsequently, we rescale each channel to fit within the range of [-1, 1]. Finally, we apply a Gaussian filter to smooth out any extreme channel values using neighboring pixel channel values. We use 4 A5000 GPUs with batch size 32 for training the diffusion model. The training process takes approximately 4 days when utilizing a dataset consisting of 2,500 objects.

Triplane Classifier training. For training the Triplane Classifier, we adopt the architecture and training rules presented in [50]. The distinction is that we use an input image channel size of 96 instead of 3.

Hyper-parameters. In our default experimental setting, we set the diffusion step $N = 500$ and $S = 250$. The gradient weight η_{str} and η_{cls} is set 1 and 0.2 respectively. The total time step of the diffusion model is 10^3.

6 Experiments

6.1 Datasets

We conduct 3D shape-to-shape translation experiments leveraging the objects in ShapeNet [52] dataset. Specifically, we perform Armchair ↔ Armless chair, Chair ↔ Table, and Tall table ↔ Short table translation, and compare our method with existing unpaired shape-to-shape translation network LOGAN and UNIST. We use the same training setting with LOGAN and UNIST, containing 1,710/2,857 training objects for armchair/armless chair, 4,786/5,993 training objects for chair/table, 2,500/2,500 for tall table/short table, and about 500 testing objects for each category. We use Marching Cubes [53] to obtain a mesh from the occupancy fields, sampled at 256^3 resolution which is the same as that used

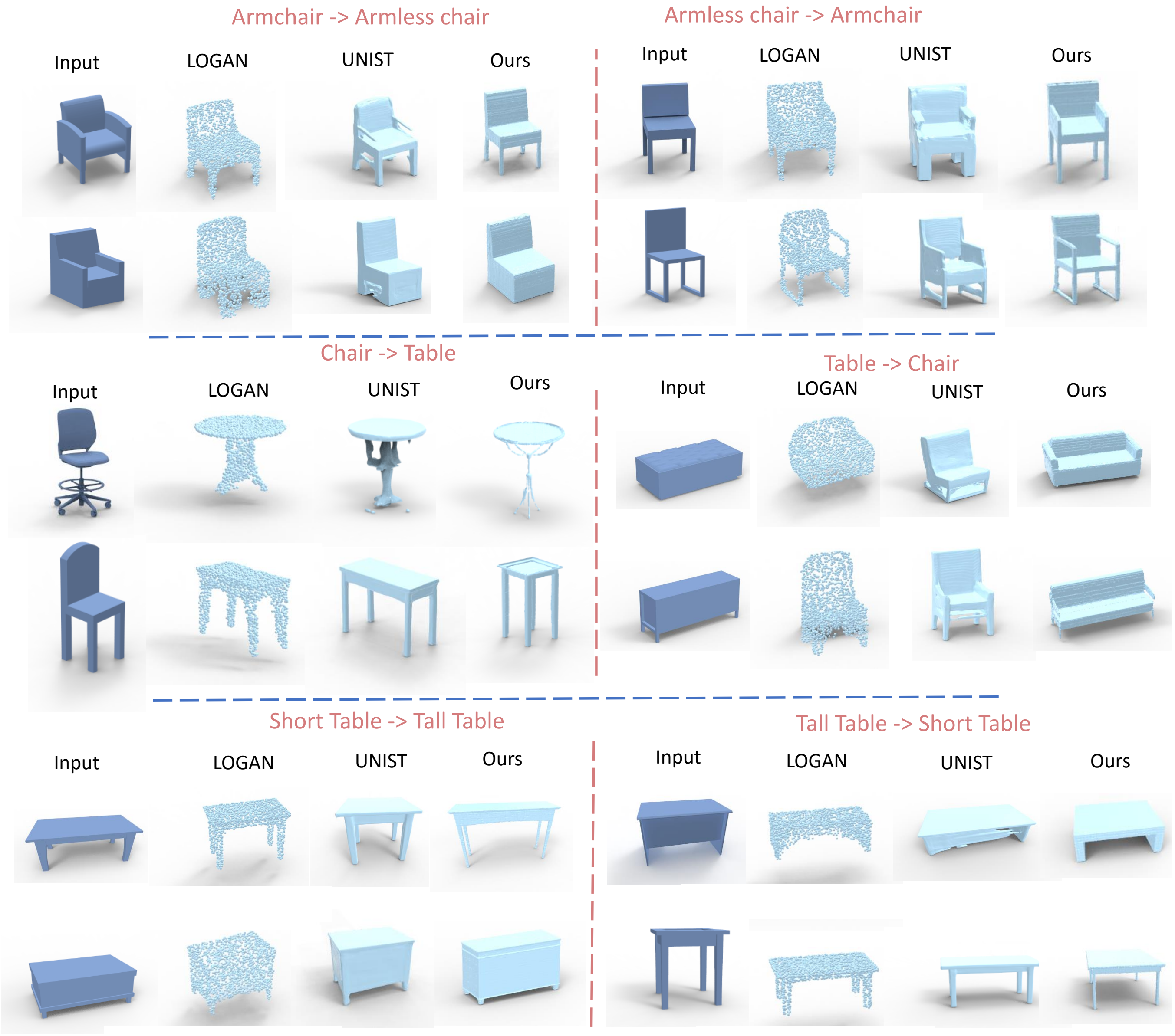

Fig. 7: Qualitative shape-to-shape translation comparison with LOGAN [7] and UNIST [8]. We conduct the comparison on Armchair ↔ Armless chair, Chair ↔ Table, and Tall table ↔ Short table translation tasks.

in UNIST to ensure a fair comparison. For LOGAN, we use their official implementation in which results contain 2,048 points.

6.2 Evaluation Metrics.

For quantitative evaluations, the shape-to-shape translation is an ill-posed problem, where a correct translation can be highly varied. In the original unpaired shape translation setting in LOGAN [7], there are no one-to-one mapping relationships between the source and target domain objects for both the training dataset and testing dataset. Therefore, we adopt the following measures for quantitative evaluation: 1) One-side CD, which we follow LOGAN and UNISIT to compute the one-sided Chamfer Distance (CD) from the input object to the output object, where we uniformly sample 2,048 points from the meshes from both our and UNIST results. This measures how well the original structure in the input is preserved; 2) Minimal Matching Distance (MMD), which is the Chamfer Distance (CD) between the output and a target object from the target domain training dataset which is closest to the output object in terms of CD. This measures how much the output resembles a typical target domain object. We also uniformly sample 2,048 points from the output meshes and the target domain objects for this evaluation.

6.3 Qualitative Comparison

Figure 7 illustrates the qualitative results obtained from our proposed methods compared to other baselines. It is evident that our method produces meshes of higher quality, demonstrating greater fidelity to the input shape. While UNIST is capable of generating transformed shapes with smooth surfaces, some results are not completely transformed into the target domain. For example, the armrest partly remains in the Armchair → Armless chair translation. Additionally, both UNIST and LOGAN exhibit a more pronounced issue of mode collapse. As exemplified by the translated shapes in Short table → Tall table, UNIST and LOGAN fail to accurately represent the original input shapes. In contrast, our results are more faithful to the original input objects.

6.4 Quantitative Comparison

We present the quantitative results using the one-side Chamfer Distance (CD) measurement in Table 1. Similar to LOGAN and UNIST, we only perform this measurement for the Armchair ↔ Armless chair translation. In cases such as Table ↔ Chair and Tall table ↔ Short table translations, it is required to not only modify the geometry of the input shape, but also change its semantics. Therefore, the one-side CD metric cannot accurately assess the quality of structure preservation.

It could be observed that our method achieves the lowest one-side CD, indicating that our method could well preserve the original input structure. This observation is also supported by the qualitative results in Figure 7, where our results exhibit a higher similarity in terms of overall structure and semantics to the original input. We also evaluate the performance of our method without the structure preservation guidance (w/o spg), and we observe a significant improvement in one-side CD when the guidance is applied. We have also evaluated the impact of the classification guidance (denoted as w/o cls) on the one-sided CD. However, we found that classification guidance exerts a minor influence on this metric. This is primarily because it chiefly governs category translation, which has minimal effect on structure preservation.

	Armchair→Armless Chair	Armless Chair→Armchair
LOGAN	0.0249	0.0273
UNIST	0.0234	0.0235
Ours w/o spg	0.0253	0.0231
Ours w/o cls	0.0183	0.0161
Ours w/o cls&spg	0.0261	0.0233
Ours	**0.0181**	**0.0158**

TABLE 1: Quantitative comparison results on one-side CD.

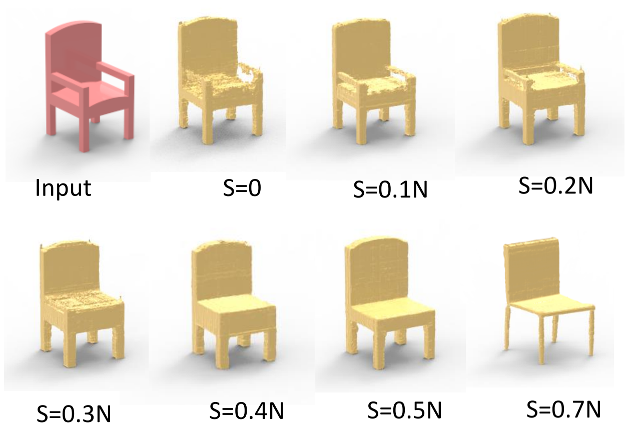

Fig. 8: Evaluation of structure preservation guidance. We show the visual results with different ending step S. The N is set to 500 for all the results.

The results with MMD measurement are shown in Table 2. Our method achieves the lowest MMD in most cases, demonstrating our results are well translated to the target domain. Furthermore, we evaluate the results of our method without the classifier guidance (w/o cls), and the comparison demonstrates the advantages of the proposed classifier guidance in facilitating the translation process towards the target distribution, as indicated by lower MMD achieved when using the classifier guidance. Additionally, we have also tested the impact of structure preservation guidance (denoted as w/o spg) on the MMD metric. Similarly, the structure preservation guidance has a modest impact on this metric. This can be attributed to the fact that even if the structure is not preserved, the translated shapes still fall within the target domain.

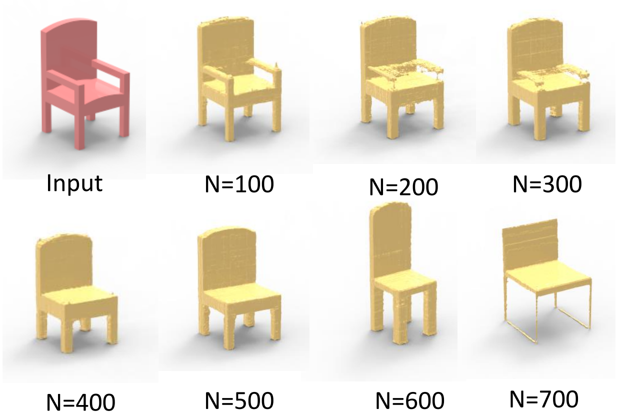

Fig. 9: Visual results with different diffusion steps N. The S in structure preservation guidance is set to $0.5N$ for all the results.

6.5 Abliation Study

Evaluation of the diffusion step N. The evaluation of the diffusion step parameter N is crucial as it significantly impacts the quality of the translation results. As mentioned in Section 4.2, choosing an appropriate value for N is essential to ensure a complete and accurate translation from the source shape to the target shape distribution. If N is set to a small value, the translation may be incomplete, and the resulting shape may not fully align with the target domain. However, if a large N is chosen, excessive noise will be added during the diffusion process, leading to the loss of the original input structure.

In Figure 9, we present visual results of our method with different values of N for the Armchair → Armless chair translation task. It can be observed that when N is set to a small value (e.g., 200), the generated result still retains some characteristics of an armchair, such as the presence of an armrest. However, as N increases to 500, the translated result becomes more plausible, preserving the original structure while successfully transforming the armchair into a chair without an armrest. The results gradually lose their connection to the input object when N is big than 500, indicating the loss of original information due to excessive noise addition.

Structure preservation guidance evaluation. We conducted an evaluation of our proposed structure preservation guidance using different hyperparameter values for S. The structure preservation guidance is applied during the reverse denoising steps from $[N, ..., S]$, where S controls the extent of applying the guidance. When S is equal to N, it indicates that no structure preservation guidance is applied. Conversely, when S is set to 0, the structure preservation guidance is applied in every denoising step.

In Figure 8, we present visual results for different values of S while keeping N fixed at 500. It can be observed that the domain-specific structures, such as the armrest, are preserved when S is

	Armchair→Armless Chair	Armless Chair→Armchair	Chair→Table	Table→Chair	Tall Table→Short Table	Short Table→Tall Table
LOGAN	0.0125	0.0132	0.0081	0.0151	0.0161	0.0133
UNIST	0.0138	0.0135	0.0069	0.0168	0.0132	**0.0108**
Ours w/o cls	0.0112	0.0120	0.0070	0.0145	0.0140	0.0123
Ours w/o spg	0.0102	0.0115	0.0063	0.0142	0.0135	0.0127
Ours w/o cls&spg	0.0113	0.0122	0.0071	0.0148	0.0141	0.0123
Ours	**0.0100**	**0.0113**	**0.0062**	**0.0142**	**0.0129**	0.0126

TABLE 2: Quantitative comparison results measured by Minimal Matching Distance (MMD).

smaller than $0.4 \times N$. However, if S is set to a value larger than $0.5 \times N$, the original structure cannot be well preserved. The best visual results are obtained when S is set to 0.5 times N, indicating that this value strikes a good balance between preserving the original structure and achieving successful shape translation.

Additionally, we also include the quantitative results about the diffusion step, the S in structure preservation in terms of MMD and CD in Figure 10. We report the average one-side CD on Armchair $\leftrightarrow$ Armless chair translation, and average MMD on all translation categories.

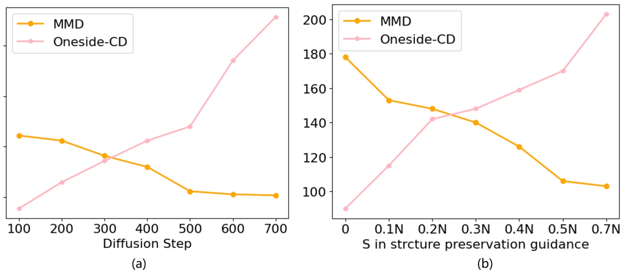

Fig. 10: Quantitative evaluation of the hyperparameter N and S. The results are multiplied by 10^4.

Classifier guidance evaluation. Figure 11 showcases the evaluation results of our proposed classifier guidance. The comparison reveals the advantages conferred by the classifier guidance in facilitating the translation process. In the absence of the classifier guidance (w/o cls guidance), we can observe that the armrest is still partially present in the translated results.

Cross-plane Convolution Evaluation Table 3 shows an evaluation of the design cross-plane convolution, where we compare the results between using the cross-plane convolution and using the original convolution with concatenated triplanes. We could observe that the performance drops much when without using the cross-plane convolution layer.

	One-side CD	MMD
w/o cross-plane conv	0.0226	0.0122
w/ cross-plane conv	**0.0170**	**0.0112**

TABLE 3: Quantitative comparison of using and without using cross-plane convolution layer. The results display the average one-side CD for the Armchair $\leftrightarrow$ Armless chair translation task, as well as the average MMD across all translation tasks.

Diffusion on Point Cloud vs on Triplanes. In our implementation, we employ a triplane factorization approach to represent 3D objects, enabling us to perform forward and backward diffusion using a 2D diffusion model for shape-to-shape translation. However, it is also possible to directly apply the diffusion process on the point cloud representation using existing point diffusion models [23], [24]. Specifically, instead of factorizing objects into triplane maps, we could utilize the point cloud representation and apply the same principles as our method with a point cloud diffusion model. We use the diffusion model implementation in PVD [24] as the backbone for point cloud diffusion. During the denoising process, we also involve a similar classifier guidance, where the classifier is a PointNet directly operating on the point cloud instead of triplanes for binary classification. Similar to LOGAN, we represent each 3D shape using 2,048 points in our experiments. We compare the results generated by conducting diffusion processes on triplane and on point cloud representations, respectively.

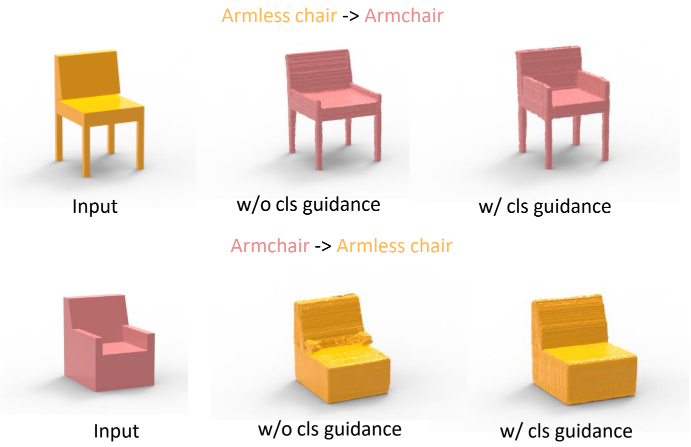

Fig. 11: Results comparison when denoising with proposed classifier guidance (w/ cls guidance) and without classifier guidance (w/o cls guidance).

Figure 12 visualize the denoising process on triplane maps versus point clouds. It reveals that while the results obtained through point cloud diffusion can undergo successful transformation into the target domain, they tend to exhibit higher noise levels and cannot well preserve the structure of the input. In contrast, the shapes generated through triplane diffusion exhibit smoother surfaces and better fidelity to the input object. We also give a quantitative comparison in Table 4, where we show the average one-side CD on Armchair $\leftrightarrow$ Armless chair translation, and average MMD on all translation categories. We could observe that by employing the diffusion process on triplane representations, we achieve lower one-side Chamfer Distance (CD) and Minimal Matching Distance (MMD) in comparison to point clouds.

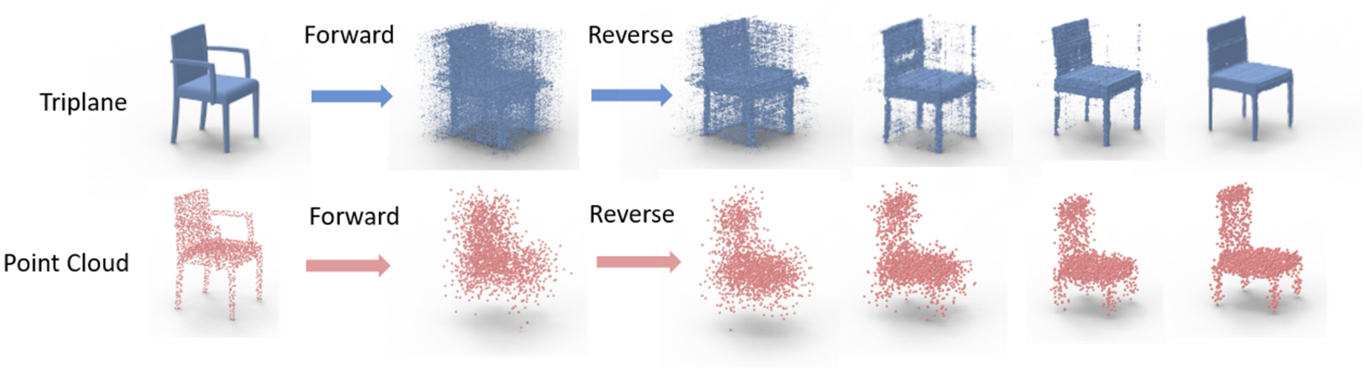

Fig. 12: The translation process comparison of our method with the diffusion process implemented on triplane and point cloud.

Diffusion on	One-side CD	MMD
Point cloud	0.0251	0.0126
Triplane	**0.0170**	**0.0112**

TABLE 4: Quantitative comparison of conducting our diffusion translation on triplane maps versus point clouds.

6.6 User Study

We also conducted a user study to evaluate the results generated by LOGAN, UNIST, and our method. The study utilized a questionnaire that incorporated a visual results comparison, akin to Figure 7 in our paper. The questionnaire was designed to address two key areas: 1) For each set of translation samples, participants were asked to determine which translation result was most reasonable and faithful to the original shape (Translation Quality). 2) At the end of the questionnaire, participants were asked to identify which method achieved the greatest diversity in translated shapes (Diversity). We included 30 shape translation comparisons in the questionnaire with 50 participants. The statistics from the questionnaire are presented in Table 5. We observed that the majority of users perceive our results as being of higher quality and exhibiting greater diversity.

Methods	Translation Quality	Diversity
LOGAN	5.7%	2%
UNIST	16.2%	12%
Ours	**78.1%**	**86%**

TABLE 5: User study results on the translation quality and generation diversity.

6.7 Application on unpaired deformation transfer

In this experiment, we aim to further assess the generality of our shape transformation network on the task of unpaired deformation transfer [54]. To accomplish this, we utilize training datasets provided in [54] consisting of animals and humans 3D meshes. Specifically, we employ the horse ↔ camel and cat ↔ lion datasets, each containing approximately 380 objects in their respective training sets. Additionally, we explore the Thin man ↔ Fat man translation, which involves 495 objects. Compared to previous experiments on ShapeNet, these datasets are significantly smaller, thus the trained diffusion model is not as effective as those in previous experiments. To this end, we first retrieve a triplane image in the training set that has the lowest mean squared error (MSE) to the triplane image of the input object. Subsequently, we perform a linear interpolation between these two triplane maps and use the resulting interpolated triplane image as the input for the diffusion model. We show the translation results in Figure 13. It shows that our method is able to keep the original skeleton with pose preserving, but also achieves a proper shape style transformation.

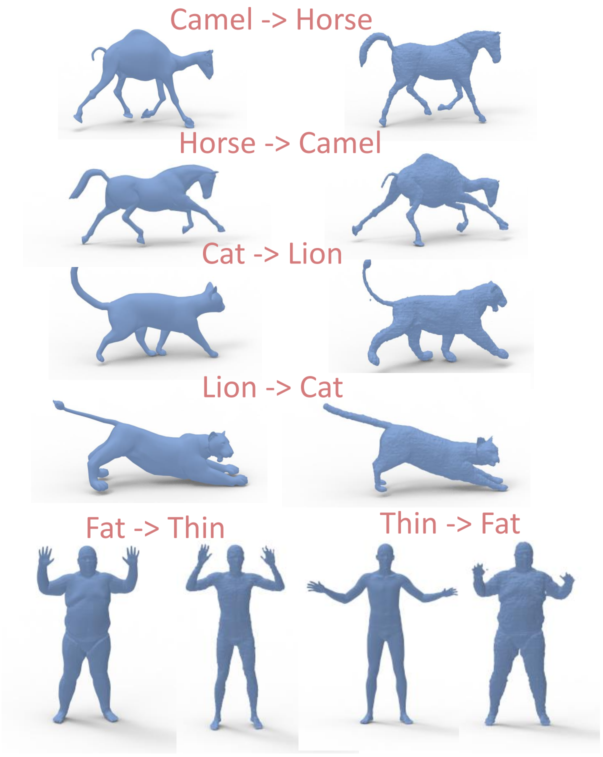

Fig. 13: Application on unpaired deformation transfer.

7 Limitations

As discussed in the ablation study section, we determined that the optimal value for the diffusion step parameter N is 500, based on the average performance across various input categories. However, we acknowledge that the ideal value of N may vary depending on the specific characteristics of the input shapes. For instance, when the input point could contains more complex armrest structures typically require larger diffusion steps N compared to those with narrower or simpler armrests. This is because it needs additional diffusion steps to blur the complex domain-specific structures.

8 Conclusion and future work

In this paper, we present a novel gradient-guided triplane diffusion architecture for unpaired 3D shape-to-shape translation. Unlike previous methods that rely on latent features and adversarial training to transform source shape encoding into the target shape, our approach utilizes a diffusion process with proposed gradient-based guidances on triplane representations. This progressive shape transformation approach could effectively alleviate the mode collapse issue. Experimental results demonstrate that our method outperforms state-of-the-art methods across various shape-to-shape translation tasks.

In terms of future research, we envision two potential directions. First, as we mentioned in the limitation section, we aim to explore the possibility of designing a mechanism to determine the best diffuson step N automatically for different input objects. This would alleviate the burden of manual parameter tuning and enhance the adaptability of our approach. Second, we employ a simple low-pass filter to extract the overall input shape structure in the proposed structure preservation guidance. We are interested in investigating more sophisticated structure extraction techniques. For instance, incorporating disentangled representation learning methods [55], [56], [57] may enable us to extract more precise domain-independent structures, thereby further improving the fidelity of our shape-to-shape translation.

References

[1] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[2] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8789–8797.

[3] P. Zhu, R. Abdal, Y. Qin, and P. Wonka, "Sean: Image synthesis with semantic region-adaptive normalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5104–5113.

[4] J. Kim, M. Kim, H. Kang, and K. H. Lee, "U-gat-it: Unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation," in *International Conference on Learning Representations*.

[5] T. Park, A. A. Efros, R. Zhang, and J.-Y. Zhu, "Contrastive learning for unpaired image-to-image translation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*. Springer, 2020, pp. 319–345.

[6] K. Yin, H. Huang, D. Cohen-Or, and H. Zhang, "P2p-net: Bidirectional point displacement net for shape transform," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–13, 2018.

[7] K. Yin, Z. Chen, H. Huang, D. Cohen-Or, and H. Zhang, "Logan: Unpaired shape transform in latent overcomplete space," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 6, pp. 1–13, 2019.

[8] Q. Chen, J. Merz, A. Sanghi, H. Shayani, A. Mahdavi-Amiri, and H. Zhang, "Unist: unpaired neural implicit shape translation network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 614–18 622.

[9] E. R. Chan, C. Z. Lin, M. A. Chan, K. Nagano, B. Pan, S. De Mello, O. Gallo, L. J. Guibas, J. Tremblay, S. Khamis *et al.*, "Efficient geometry-aware 3d generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 123–16 133.

[10] M. Zhao, F. Bao, C. Li, and J. Zhu, "Egsde: Unpaired image-to-image translation via energy-guided stochastic differential equations," *Advances in Neural Information Processing Systems*, vol. 35, pp. 3609–3623, 2022.

[11] C. Meng, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon, "Sdedit: Image synthesis and editing with stochastic differential equations," *arXiv preprint arXiv:2108.01073*, 2021.

[12] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[13] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," *Advances in neural information processing systems*, vol. 30, 2017.

[14] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 172–189.

[15] Z. Yi, H. Zhang, P. Tan, and M. Gong, "Dualgan: Unsupervised dual learning for image-to-image translation," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2849–2857.

[16] D. Torbunov, Y. Huang, H. Yu, J. Huang, S. Yoo, M. Lin, B. Viren, and Y. Ren, "Uvcgan: Unet vision transformer cycle-consistent gan for unpaired image-to-image translation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 702–712.

[17] Y. Zhao, R. Wu, and H. Dong, "Unpaired image-to-image translation using adversarial consistency loss," in *ECCV*, 2020.

[18] J. Choi, S. Kim, Y. Jeong, Y. Gwon, and S. Yoon, "Ilvr: Conditioning method for denoising diffusion probabilistic models," *arXiv preprint arXiv:2108.02938*, 2021.

[19] J.-W. Zheng, J.-Y. Hsu, C.-C. Li, and I.-C. Lin, "Characteristic-preserving latent space for unpaired cross-domain translation of 3d point clouds," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–15, 2023.

[20] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International Conference on Machine Learning*. PMLR, 2015, pp. 2256–2265.

[21] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[22] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *arXiv preprint arXiv:2010.02502*, 2020.

[23] S. Luo and W. Hu, "Diffusion probabilistic models for 3d point cloud generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2837–2845.

[24] L. Zhou, Y. Du, and J. Wu, "3d shape generation and completion through point-voxel diffusion," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5826–5835.

[25] K.-H. Hui, R. Li, J. Hu, and C.-W. Fu, "Neural wavelet-domain diffusion for 3d shape generation," in *SIGGRAPH Asia 2022 Conference Papers*, 2022, pp. 1–9.

[26] Y.-C. Cheng, H.-Y. Lee, S. Tulyakov, A. Schwing, and L. Gui, "Sdfusion: Multimodal 3d shape completion, reconstruction, and generation," *arXiv preprint arXiv:2212.04493*, 2022.

[27] G. Nam, M. Khlifi, A. Rodriguez, A. Tono, L. Zhou, and P. Guerrero, "3d-ldm: Neural implicit 3d shape generation with latent diffusion models," *arXiv preprint arXiv:2212.00842*, 2022.

[28] G. Chou, Y. Bahat, and F. Heide, "Diffusionsdf: Conditional generative modeling of signed distance functions," *arXiv preprint arXiv:2211.13757*, 2022.

[29] M. Li, Y. Duan, J. Zhou, and J. Lu, "Diffusion-sdf: Text-to-shape via voxelized diffusion," *arXiv preprint arXiv:2212.03293*, 2022.

[30] A. Karnewar, N. J. Mitra, A. Vedaldi, and D. Novotny, "Holofusion: Towards photo-realistic 3d generative modeling," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 22 976–22 985.

[31] J. Koo, S. Yoo, M. H. Nguyen, and M. Sung, "Salad: Part-level latent diffusion for 3d shape generation and manipulation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 14 441–14 451.

[32] B. Zhang, J. Tang, M. Niessner, and P. Wonka, "3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models," *arXiv preprint arXiv:2301.11445*, 2023.

[33] B. Kim, P. Kwon, K. Lee, M. Lee, S. Han, D. Kim, and H. Joo, "Chupa: Carving 3d clothed humans from skinned shape priors using 2d diffusion probabilistic models," *arXiv preprint arXiv:2305.11870*, 2023.

[34] J. Xiang, J. Yang, B. Huang, and X. Tong, “3d-aware image generation using 2d diffusion models,” *arXiv preprint arXiv:2303.17905*, 2023.

[35] Y. Shi, P. Wang, J. Ye, M. Long, K. Li, and X. Yang, “Mvdream: Multi-view diffusion for 3d generation,” *arXiv preprint arXiv:2308.16512*, 2023.

[36] Y. Xu, H. Tan, F. Luan, S. Bi, P. Wang, J. Li, Z. Shi, K. Sunkavalli, G. Wetzstein, Z. Xu *et al.*, “Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model,” *arXiv preprint arXiv:2311.09217*, 2023.

[37] P. Wang and Y. Shi, “Imagedream: Image-prompt multi-view diffusion for 3d generation,” *arXiv preprint arXiv:2312.02201*, 2023.

[38] J. R. Shue, E. R. Chan, R. Po, Z. Ankner, J. Wu, and G. Wetzstein, “3d neural field generation using triplane diffusion,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 20 875–20 886.

[39] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger, “Convolutional occupancy networks,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*. Springer, 2020, pp. 523–540.

[40] K. Han, S. Sun, T.-T. Le, X. Yan, H. Ma, C. You, and X. Xie, “Hybrid neural diffeomorphic flow for shape representation and generation via triplane,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 7707–7717.

[41] A. Gupta, W. Xiong, Y. Nie, I. Jones, and B. Oğuz, “3dgen: Triplane latent diffusion for textured mesh generation,” *arXiv preprint arXiv:2303.05371*, 2023.

[42] S. Hu, F. Hong, T. Hu, L. Pan, H. Mei, W. Xiao, L. Yang, and Z. Liu, “Humanliff: Layer-wise 3d human generation with diffusion model,” *arXiv preprint arXiv:2308.09712*, 2023.

[43] E. Ntavelis, A. Siarohin, K. Olszewski, C. Wang, L. Van Gool, and S. Tulyakov, “Autodecoding latent 3d diffusion models,” *arXiv preprint arXiv:2307.05445*, 2023.

[44] T. Wang, B. Zhang, T. Zhang, S. Gu, J. Bao, T. Baltrusaitis, J. Shen, D. Chen, F. Wen, Q. Chen *et al.*, “Rodin: A generative model for sculpting 3d digital avatars using diffusion,” *arXiv preprint arXiv:2212.06135*, 2022.

[45] J. Gu, A. Trevithick, K.-E. Lin, J. Susskind, C. Theobalt, L. Liu, and R. Ramamoorthi, “Nerfdiff: Single-image view synthesis with nerf-guided distillation from 3d-aware diffusion,” *arXiv preprint arXiv:2302.10109*, 2023.

[46] Y. Wang, I. Skorokhodov, and P. Wonka, “Pet-neus: Positional encoding triplanes for neural surfaces,” 2023.

[47] W. Hu, Y. Wang, L. Ma, B. Yang, L. Gao, X. Liu, and Y. Ma, “Tri-miprf: Tri-mip representation for efficient anti-aliasing neural radiance fields,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 19 774–19 783.

[48] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, “Occupancy networks: Learning 3d reconstruction in function space,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4460–4470.

[49] R. Wu, R. Liu, C. Vondrick, and C. Zheng, “Sin3dm: Learning a diffusion model from a single 3d textured shape,” *arXiv preprint arXiv:2305.15399*, 2023.

[50] P. Dhariwal and A. Nichol, “Diffusion models beat gans on image synthesis,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 8780–8794, 2021.

[51] J. Huang, Y. Zhou, and L. Guibas, “Manifoldplus: A robust and scalable watertight manifold surface generation method for triangle soups,” *arXiv preprint arXiv:2005.11621*, 2020.

[52] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, “ShapeNet: An Information-Rich 3D Model Repository,” Stanford University — Princeton University — Toyota Technological Institute at Chicago, Tech. Rep., 2015.

[53] W. E. Lorensen and H. E. Cline, “Marching cubes: A high resolution 3d surface construction algorithm,” *ACM siggraph computer graphics*, vol. 21, no. 4, pp. 163–169, 1987.

[54] L. Gao, J. Yang, Y.-L. Qiao, Y.-K. Lai, P. L. Rosin, W. Xu, and S. Xia, “Automatic unpaired shape deformation transfer,” *ACM Transactions on Graphics (TOG)*, vol. 37, no. 6, pp. 1–15, 2018.

[55] H. Kim and A. Mnih, “Disentangling by factorising,” in *International Conference on Machine Learning*. PMLR, 2018, pp. 2649–2658.

[56] A. H. Liu, Y.-C. Liu, Y.-Y. Yeh, and Y.-C. F. Wang, “A unified feature disentangler for multi-domain image translation and manipulation,” *Advances in neural information processing systems*, vol. 31, 2018.

[57] E. H. Sanchez, M. Serrurier, and M. Ortner, “Learning disentangled representations via mutual information estimation,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*. Springer, 2020, pp. 205–221.

SUPPLEMENTARY MATERIAL

1 MORE VISUAL RESULTS

We add more diverse results in the testing dataset in Figure 3. Instead of only the typical square chair or table with four legs, we show results of diverse types of chairs and tables. We could observe that our method achieves better quality than UNIST and LOGAN. Also, due to the claimed mode collapse issue, the overall diversity of LOGAN and UNIST is limited.

We have also included additional results from other ShapeNet categories to further illustrate the effectiveness of our proposed method. To establish a shape translation between two categories, these categories should share similar structures or topologies. For instance, a $plane \leftrightarrow bed$ translation is not feasible due to their significantly different geometric structures. Based on this principle, we have included additional translations between Bed ↔ Table, Bed ↔ Chair, and Sofa ↔ Table to further demonstrate the effectiveness of our proposed method. These results are displayed in Figure 4.

Addtionally, we add the comparison between our method with LOGAN on dyna and animal datasets in Figure 5. As LOGAN does not release its pretrained model on these datasets, we use the visual examples shown in their paper on arxiv for comparison.

2 FRÉCHET INCEPTION DISTANCE EVALUATION

We also add the Fréchet Inception Distance (FID) comparison with your kind suggestion. In the case of 2D data generation problems, FID usually adopts pre-trained inception V3 models [1] to utilize their feature spaces for evaluation. In our point cloud case, we follow the setting in [2] to calculate FID. As a reference model for FID, we used the classification module of PointNet [3]. we first trained a classification module for 40 epochs to attain an accuracy of 97% for classification tasks. We then extracted a 1024-dimensional feature vector from the output of the layer before max-pooling layer to calculate the mean and covariance. Specifically, we calculate the 2-Wasserstein distance between real and fake Gaussian measures in the feature spaces extracted by the pretrained PointNet as follows:

$$\mathrm{FID}(\mathbb{P},\mathbb{Q}) = \mathbf{m}_{\mathbb{P}} - \mathbf{m}_{\mathbb{Q}_2^2}^{2} + \mathrm{Tr}\left(\Sigma_{\mathbb{P}} + \Sigma_{\mathbb{Q}} - 2\left(\Sigma_{\mathbb{P}}\Sigma_{\mathbb{Q}}\right)^{\frac{1}{2}}\right), \quad (1)$$

where $\mathbf{m}_{\mathbb{P}}$ and $\Sigma_{\mathbb{P}}$ are the mean vector and covariance matrix of the points calculated from real point clouds of the trianing data $\{x\}$, respectively, and $\mathbf{m}_{\mathbb{Q}}, \Sigma_{\mathbb{Q}}$ are the mean vector and covariance matrix calculated from generated point clouds $\{x'\}$, respectively, where $x \sim \mathbb{P}$ and $x' = G(z) \sim \mathbb{Q}$. Shapes from all baselines are sampled to 2048 points before passed into the pre-trained PointNet.

We report the FID of LOGAN, UNIST, and our method in Table 1. We could observe that our method achieves lower FID compared to the other baselines in all categories translation, indicating that our method could generate more diverse shapes which alleviates the mode collapse problem in GAN-based methods like LOGAN and UNIST.

	Methods	FID ↓
Chair ↔ Table	**LOGAN**	2.35
	UNIST	2.26
	Ours	**2.13**
Armchair ↔ Armless Chair	**LOGAN**	2.06
	UNIST	2.21
	Ours	**2.03**
Tall Chair ↔ Short Chair	**LOGAN**	1.96
	UNIST	1.93
	Ours	**1.86**

TABLE 1: Quantitative comparison in terms of the proposed FID.

3 EFFECT OF TRIPLANE RESOLUTION

During the revision, a reviewer point out that the qualitative visual results do not appear to be smooth when zoomed in, and there are numerous small bumps on the surface that should ideally be smooth. We also notice these cases where small bumps are happened. We look into this and find the bumps are related to the triplane resolution. In our cases with limited GPU resources (a single A5000), we train triplane of size 128×128. We show the comparison in Figure 1.

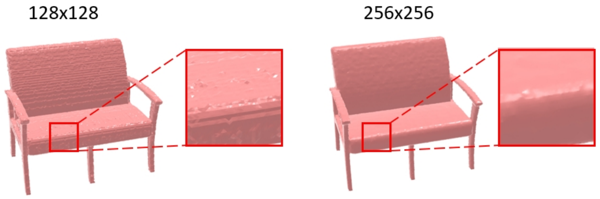

Fig. 1: Reconstructed shapes from triplane representations of different resolutions.

4 COMPARISON WITH RETRIEVED SHAPE

To investigate whether our method merely translates the source shape to shapes already present in the target domain training dataset or whether it generates novel shapes, we visualized the shapes in the training set that were most similar to the transferred shapes, as determined by the Minimum Mean Discrepancy (MMD). As can be seen below, these shapes do exhibit structural similarities to some extent. However, they are not identical, indicating that our method is capable of generating novel shapes rather than simply replicating existing ones in the target domain.

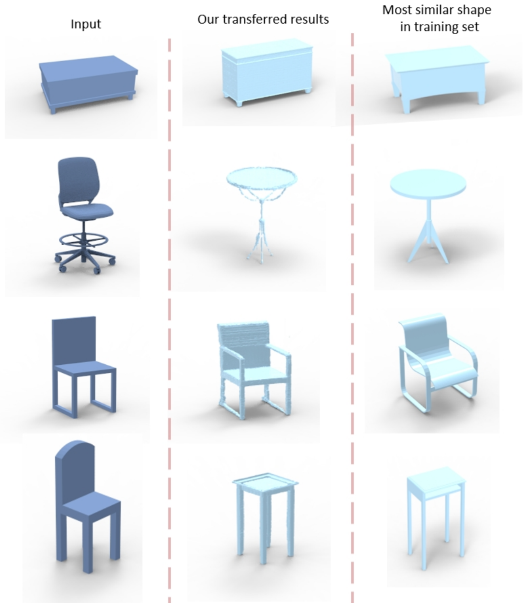

Fig. 2: Illustration of the most similar shape in the training set to the transferred shape.

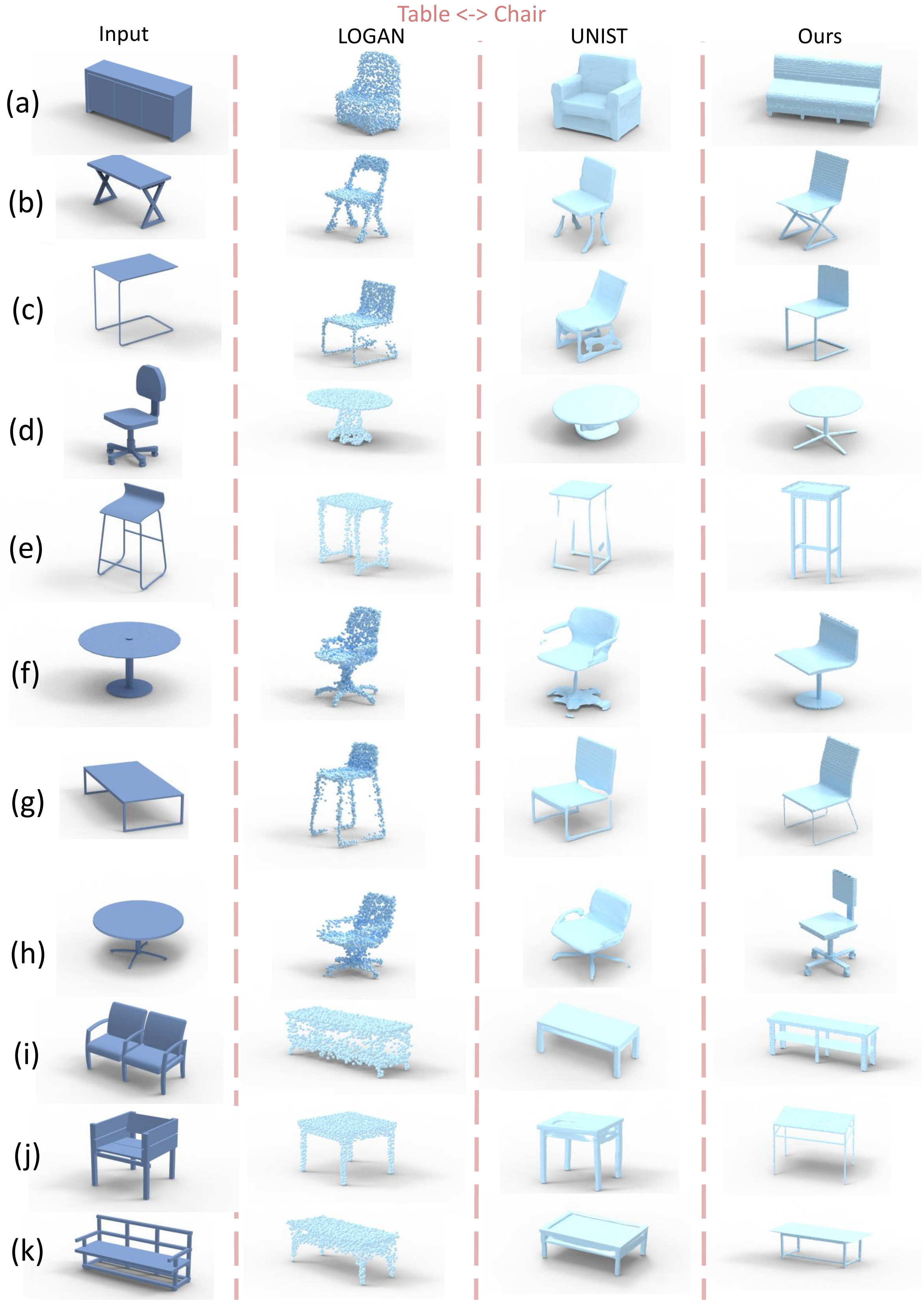

Fig. 3

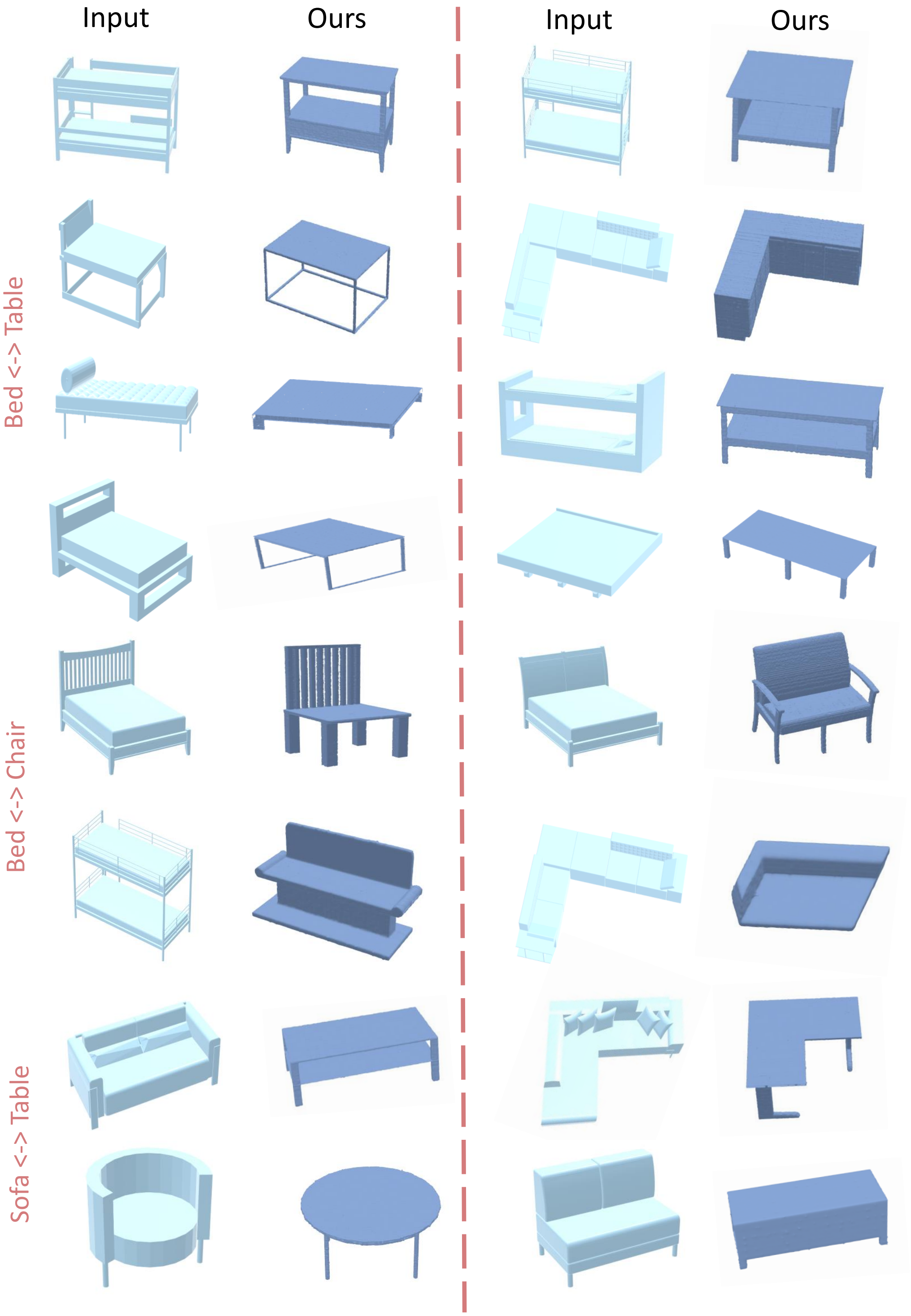

Fig. 4

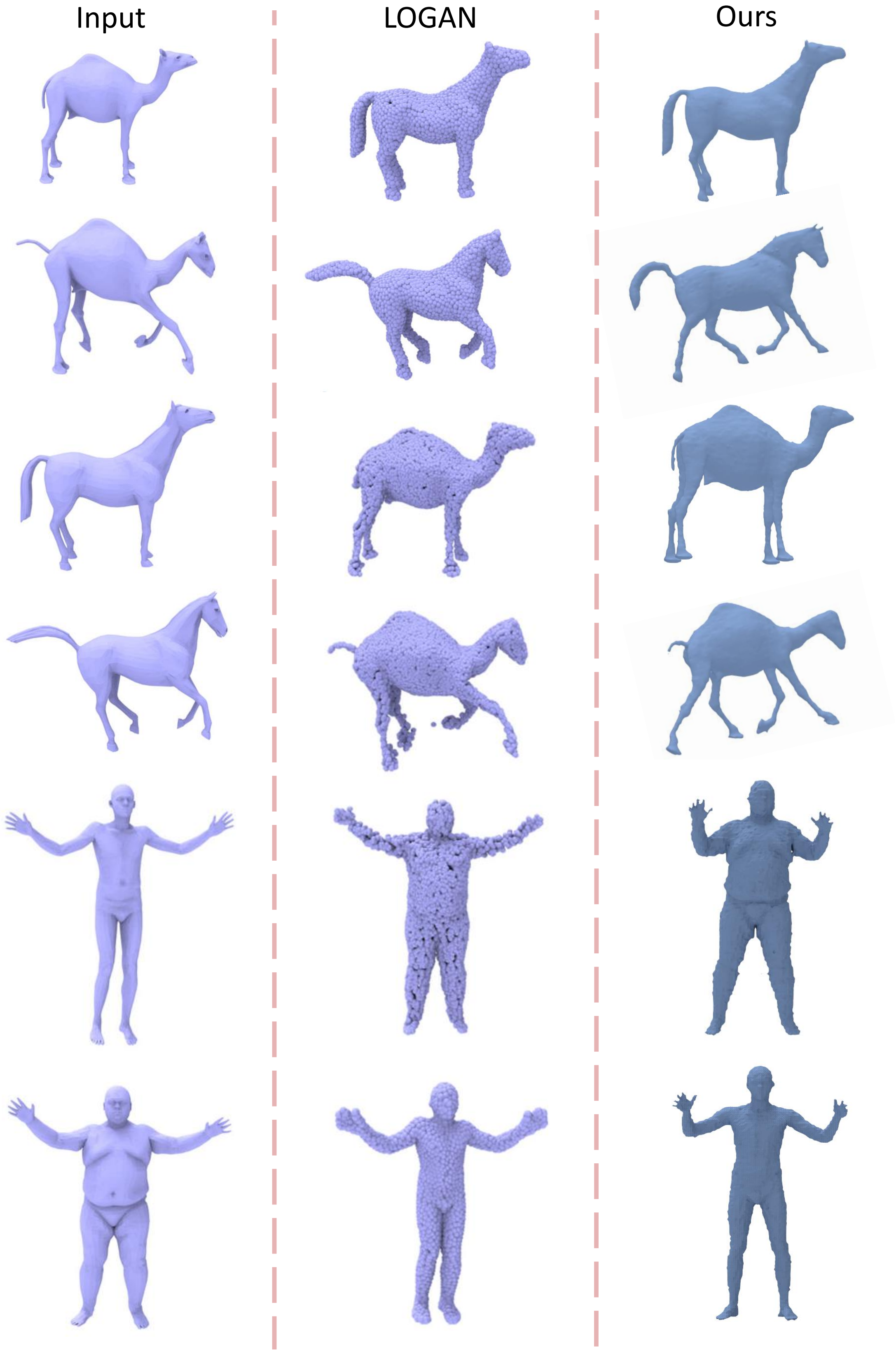

Fig. 5

References

[1] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[2] D. W. Shu, S. W. Park, and J. Kwon, "3d point cloud generative adversarial network based on tree structured graph convolutions," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 3859–3868.

[3] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," *CVPR*, vol. 1, no. 2, p. 4, 2017.