DOI: 10.1111/cgf.14664
Pacific Graphics 2022
N. Umetani, E. Vouga, and C. Wojtan
(Guest Editors)

Volume 41 (2022), Number 7

USTNet: Unsupervised Shape-to-Shape Translation via Disentangled Representations

Haoran Wang [1], Jiaxin Li [1], Alexandru Telea [2], Jiří Kosinka [3] and Zizhao Wu [1†]

[1]Hangzhou Dianzi University, Hangzhou, China
[2] Utrecht University, Utrecht, the Netherlands
[3] University of Groningen, Groningen, the Netherlands

Abstract

We propose USTNet, a novel deep learning approach designed for learning shape-to-shape translation from unpaired domains in an unsupervised manner. The core of our approach lies in disentangled representation learning that factors out the discriminative features of 3D shapes into content and style codes. Given input shapes from multiple domains, USTNet disentangles their representation into style codes that contain distinctive traits across domains and content codes that contain domain-invariant traits. By fusing the style and content codes of the target and source shapes, our method enables us to synthesize new shapes that resemble the target style and retain the content features of source shapes. Based on the shared style space, our method facilitates shape interpolation by manipulating the style attributes from different domains. Furthermore, by extending the basic building blocks of our network from two-class to multi-class classification, we adapt USTNet to tackle multi-domain shape-to-shape translation. Experimental results show that our approach can generate realistic and natural translated shapes and that our method leads to improved quantitative evaluation metric results compared to 3DSNet. Codes are available at https://Haoran226.github.io/USTNet.

CCS Concepts

• ***Computing methodologies*** → ***Point-based models;*** *Artificial intelligence;*

1. Introduction

3D modeling [CRW*20] is a central problem in the computer graphics community. The demand for large varieties of 3D models is growing due to the popularity of gaming, AR/VR, 3D films, and the metaverse. While traditional methods to address this task relies on modeling tools and domain experts, recent techniques work purely data-driven by using deep learning technologies [XKHK17], enabling them to create diverse sets of 3D models without manual intervention. A well-known example is content reuse [XKHK17, GLL*22], which reuses shapes and shape parts from model collections by mining their semantic structure information. While such topics have been extensively explored in the past years, it is challenging to identify and transfer the *style* element for the modeling task, which plays a key role in conveying high-level and abstract notions [HLvK*17]. Following this, and also related developments in stylized image processing [JYF*20], the style has gained increasing interest in the 3D modeling world in recent years [YGS*21], including the task of shape-to-shape translation [YCH*19, SGST20].

Following the idea of image-to-image translation [ZPIE17], shape-to-shape translation aims to translate a shape from a source domain to a target domain; here, *domain* refers to a collection of shapes that share a common trait that is distinctive to other domains, *e.g.*, for the armchair and straight chair domains, the trait is the presence of arm rests. We define the common trait as *style* and the rest of the contained information as *content*. The challenge in this task is to distinguish style elements and transfer them across domains, particularly in an unsupervised and unpaired setting. Figure 1 illustrates this process: If the source domain consists of armchairs (chairs having arm rests), and the target domain consists of straight chairs (with no arm rests), one would like to generate new shapes that have the particular traits of the target, combined with the domain-invariant information of the source domain (pose and structure of armchairs, including arm rests).

Several works have aimed to address the above goal. Yin *et al.* proposed LOGAN [YCH*19], a general shape-to-shape translation framework that trains a translator to map a source latent code into a target latent code and achieves high-quality cross-domain shape transforms. Yet, the proposed translator can only generate a single translated result with one input shape. To address this issue, 3DSNet [SGST20] proposed a multi-modal shape-to-shape translation framework that aims to disentangle, *i.e.*, separate, content and style information. In this approach, the content space is domain-

† Corresponding author: wuzizhao@hdu.edu.cn

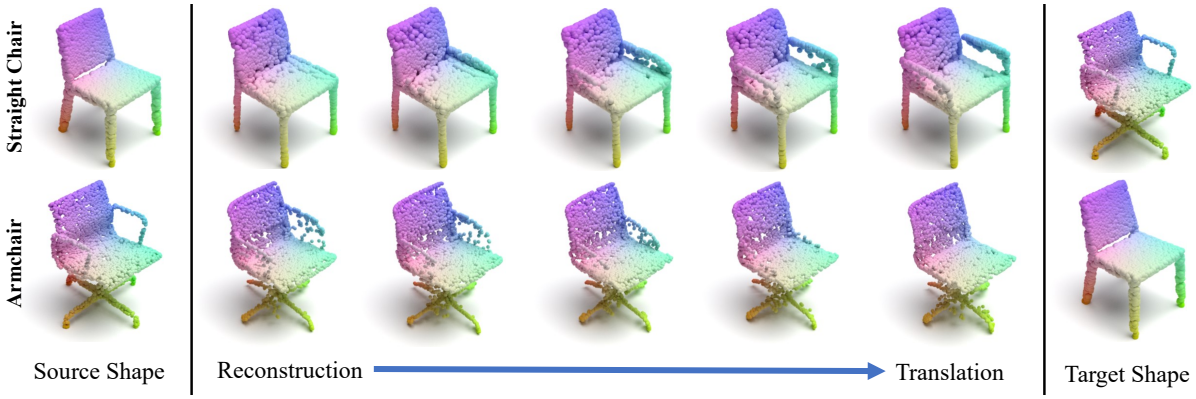

Figure 1: *Shape interpolation results produced by our method between an armchair and a straight chair model. The results are generated by uniformly interpolating in the style space between the source and target domains while keeping the content codes constant.*

invariant, and the style spaces of different domains are independent. While 3DSNet showed impressive results, the assumption of independent style spaces across domains makes its application difficult for *e.g.*, style interpolation, and multi-domain translation. Also, considering that style information is separated by a domain discriminator, we argue that this strategy cannot guarantee the faithful disentanglement of style from content due to the discriminator, which tries to find out all different traits between two domains (see Fig. 3). As an alternative, we investigate a translation approach that only changes the critical traits, *e.g.*, arm rests for armchairs.

In this paper, we propose a novel 3D shape-to-shape translation method, called USTNet, that uses a disentangled representation to generate variations of shapes. In contrast to 3DSNet's usage of independent style spaces, our method learns a shared style space and a shared content space across domains, yielding a full disentangled representation for the input shapes. While learning disentangled representations is difficult, especially in an unsupervised scenario, once these are obtained, they enable one to perform complex and highly useful operations on the data [BCV13]. In our context, learning to disentangle the style and content of shapes from unpaired domains enables us to perform flexible manipulations of styles, *e.g.*, style transfer, and style interpolation.

Our method uses a deep learning network which contains a *reconstruction* stage that generates the disentangled representations, and a *translation* stage which generates the translated results; see Fig. 2. Given a pair of shapes from unpaired domains, our network first encodes input shapes x into content codes z_c^x and style codes z_s^x based on a content encoder E_c and a style encoder E_s. These codes are next merged into z^x by a fusion block F, whose output is used by a generator block G to create the translated shapes. We use different loss functions to control disentanglement. To constrain content features to capture domain-invariant information, we use a content adversarial loss [LTH*18]. To encourage the representations of the translated shape to be consistent with the style of the target shape and also with the content of the source shape, we design a latent consistency loss.

We also extend the above model to address multi-domain shape-to-shape translation. For this, we propose a refined network called USTNet-M, which adapts the building blocks of USTNet from dual-class to multi-class classification.

We show the effectiveness of our proposed method by comparing its results with 3DSNet on various shapes from the ShapeNet and SMAL databases.

Summarizing, the contributions of our method are as follows:

- we present USTNet, a novel unpaired shape-to-shape translation network with disentangled representations;
- we introduce a content adversarial loss and a latent regression loss into the shape translation realm for latent space disentanglement;
- we propose USTNet-M, the first shape-to-shape translation network for multi-domain translation, which extends USTNet without any additional blocks;
- we show how our network achieves promising performance and generalizes to style manipulation.

The rest of the paper is structured as follows. Section 2 reviews related work. Section 3 details our model. Section 4 presents and evaluates our results. Finally, Section 5 concludes the paper and outlines directions for future work.

2. Related Work

We now describe relevant work related to shape-to-shape translation (Sec. 2.1), disentangled representation (Sec. 2.2), and image-to-image translation (Sec. 2.3).

2.1. 3D Shape Translation

Recently, deep learning based 3D shape generation has gained considerable attention, given its ability to generate diverse and realistic 3D shapes in a purely data-driven way. To date, flow-based and GAN-based methods are critical approaches for this task. We discuss both of these approaches next, followed by additional related work on 3D shape translation.

Flow-based models. These methods typically model 3D shapes by considering the distribution of point samplings. PointFlow [YHH*19] introduced continuous normalizing flows to model

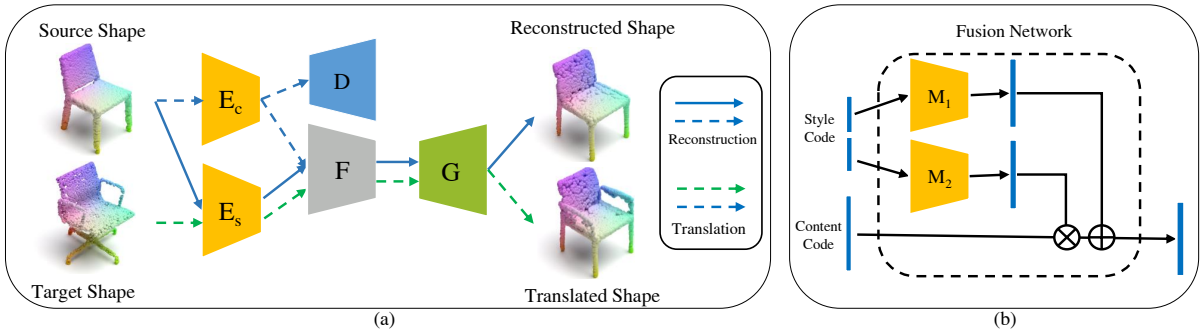

Figure 2: *Overview of our method. (a) Our network contains a content encoder E_c, a style encoder E_s, a fusion network F, and a generator G. Blue arrows show the reconstruction stage. Green dashed arrows show the translation stage. During reconstruction, the network encodes the same shape using content and style encoders. During translation, the* source *shape (straight chair) is encoded by E_c; the* target *shape (armchair) is embedded by E_s, and G generates a translated armchair which has the straight chair structure. (b) The fusion network is an affine transformation process for a content code. A style code is equally divided into two subcodes and mapped to a bias vector and a scale vector, respectively.*

a transformation of a prior distribution, leading to expressive modeling of shapes. PointGrow [SWL*20] uses an auto-regressive model to estimate the conditional distribution of the point samples. More recently, Luo *et al.* [LH21] leveraged reverse diffusion probabilistic models to estimate the distribution of points, enabling the transformation of a noise distribution to the distribution of the desired shape. Zhou *et al.* [ZDW21] proposed Point-Voxel Diffusion (PVD) for unconditional shape generation and conditional shape completion, combining the merits of denoising diffusion models with the point-voxel representation of 3D shapes.

GAN-based models. These approaches explore adversarial learning for shape generation with the help of a discriminator. Li *et al.* [LZZ*19] proposed PC-GAN, a GAN variant that learns to generate point clouds by using ideas from hierarchical Bayesian modeling and implicit generative models. Achlioptas *et al.* [ADMG18] introduced two generators for 3D shape creation: an r-GAN that operates in the raw space and an l-GAN that operates in the latent space of a pretrained autoencoder. Related methods that have been used for shape generation include spectral-domain GANs [RKBG20], tree-GAN [SPK19], progressive deconvolution networks [HXX*20], conditional generative adversarial networks [AB20], and SP-GAN [LLHF21].

Shape-to-Shape Translation. Inspired by the progress of image-to-image translation methods, researchers have exploited 3D shape-to-shape translation for shape generation. Yin *et al.* [YHCZ18] pioneered the idea by developing a bidirectional point displacement network that learns geometric transformations between point sets from two domains. Yin *et al.* proposed LOGAN [YCH*19], a general-purpose deep neural network that learns shape-to-shape translation from unpaired inputs. LOGAN features an overcompleted autoencoder to explicitly assign features at different shape scales to different portions of the latent codes and a translator network to distinguish and translate the latent vector of the source shape to the target domain. UNIST [CMS*21] introduced a new autoencoder structure based on neural implicit representations, which can generate higher-quality and more natural shapes than LOGAN while reusing the latter method's translator. Such methods, however, learn a *deterministic* mapping between domains, which means they can only generate a single translated result from a given input. DecorGAN [CKF*21] proposes a voxel-based method to disentangle shapes into overall structure and detail code. Limited by the explicit definition of content information, DecorGAN is not only difficult to edit content information but also unable to generate unseen patterns.

3DSNet [SGST20] proposes a multi-modal shape-to-shape translation model which can generate diverse translated results by disentangling content and style representations. The disentangled representations are embedded by a shared content encoder and two domain-specific style encoders. However, several problems exist here. 1) Sharing the same content encoder does not guarantee that the same content representations capture the same information for both domains [LTH*18]. 2) The results of 3DSNet show that the network cannot correctly distinguish content and style information (see the examples in Fig. 3). 3) The architectures of such methods limit the possibility of exploring the relationships between different style domains. For instance, extending these methods to *multi-domain* translation requires designing additional blocks because the translator in LOGAN and UNIST is specific for a given ordered pair, *e.g.*, *armchair* $\rightarrow$ *straight chair*, and the style encoders and decoder in 3DSNet are domain-specific.

Our USTNet offers several improvements to address the above limitations. First, we introduce content adversarial loss [LTH*18] to help our disentanglement structure learn useful representations. The shapes generated by our model show better disentanglement than in previous methods. Secondly, the common style encoder in our USTNet is used for each domain, so our network can explore relationships among multiple domains. Finally, our architecture with a common encoder and generator can be easily extended to a multi-domain translation network without requiring additional blocks.

2.2. Disentangled Representation

Disentangled representation learning aims to model the factors of data variations, thereby solving subsequent challenging real-world

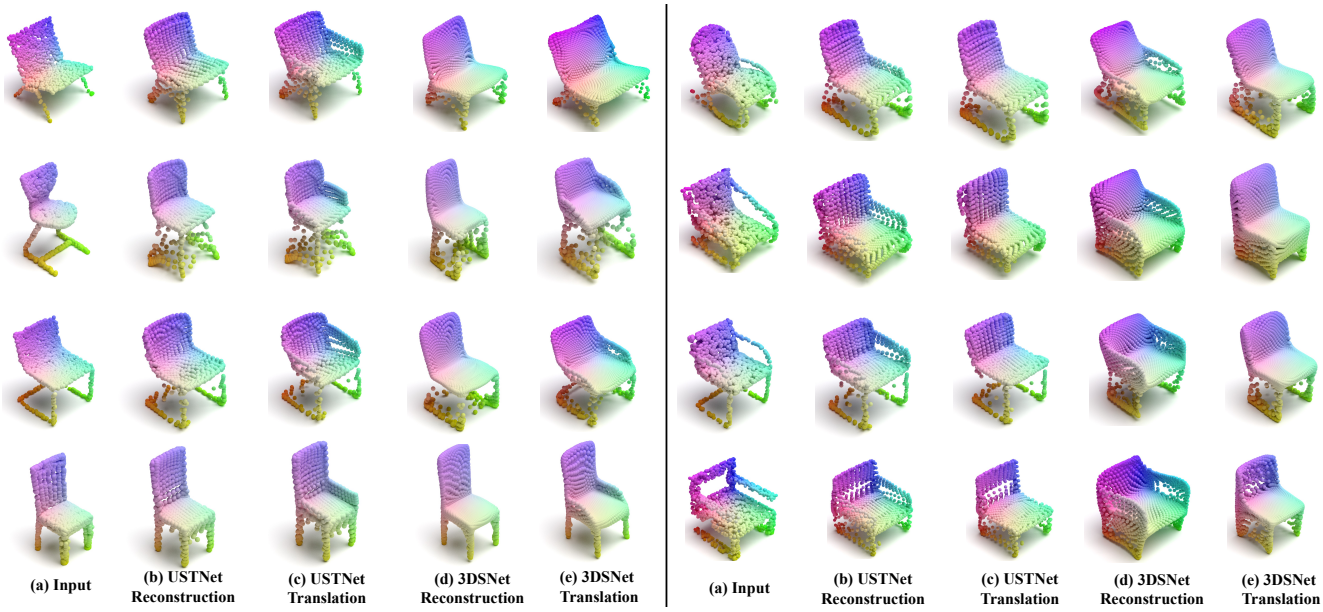

Figure 3: *Comparison of translation results by USTNet (ours) and 3DSNet on an* armchair ↔ straight chair *shape translation example. Left:* straight chair → armchair. *Right:* armchair → straight chair.

tasks, such as classification and style editing [BCV13, LBL*19]. Many approaches have been proposed to force the emergence of disentanglement into learned representations with labeled data. Gatys *et al.* [GEB16] pioneered learning an image representation to separate content and style. Liu *et al.* [LWS*18] trained an autoencoder model supervised by ground truth labels to learn a content-invariant representation.

Learning various disentangled representations based on GANs [CDH*16, LTFO20] and Variational Autoencoders (VAEs) [HMP*17, KM18, KSB18] from unlabeled data has gained increasing popularity. To create disentanglement representations, the InfoGAN [CDH*16] method maximizes the mutual information between latent variables and data variation. The β-VAE [HMP*17] model enforces greater disentanglement by reducing the influence of the reconstruction loss.

However, newer studies [LBL*19] showed that most such unsupervised disentangled methods only work for simple datasets and can have trouble disentangling more complex information. [LBL*19] also suggests that either supervision or inductive biases should be added to a disentanglement method to achieve meaningful representations. Inspired by this, we propose an inductive bias that represents the distributions of content representation in domains that share semantic information (*i.e.*, are similar) to ensure that codes are useful for style and content representations.

2.3. Unpaired Image-to-Image Translation

In recent years, many notable supervised/paired and unsupervised/unpaired cross-domain image translation works have emerged, which inspired developments in shape-to-shape translation. One of the most representative works of supervised image-to-image translation is Pix2Pix [IZZE17], which uses a conditional GAN with a reconstruction loss. Additionally, many unsupervised methods for image-to-image translation have been proposed [ZPIE17, YZTG17, LBK17].

A significant limitation of such image translation methods is that the learned mapping between two domains is deterministic, thereby lacking diversity in the translated outputs. To tackle this, some disentanglement methods [LTH*18, HLBK18] proposed to divide the latent space into content and style spaces so that the framework can generate diverse outputs by sampling the style distribution.

However, the above methods can learn relations between only two different domains at a time. To handle multiple domains, these methods have to append additional domain-specific blocks and separately train for each pair of domains. To alleviate this, several frameworks have been proposed to learn relations among multiple domains using a single model. StarGAN [CCK*18] uses a single generator and a discriminator to train from images of multiple domains, but it learns a deterministic mapping between each pair of domains. DRIT++ [LTM*20] introduces one-hot domain codes to each block of its network to perform a multi-domain translation with diverse translated image outputs. Although all blocks in DRIT++ are shared among all domains, the method can capture the relation among domains up to inherent limitations given by its over-reliance on the domain code. In comparison, our multi-domain translation structure embeds the style feature into a shared space which can be next explored across *all* domains and only uses the domain labels in the training phase.

3. USTNet Method Description

We now detail our USTNet method (depicted in Fig. 2) in terms of its architecture (Sec. 3.1), the two core loss functions it uses

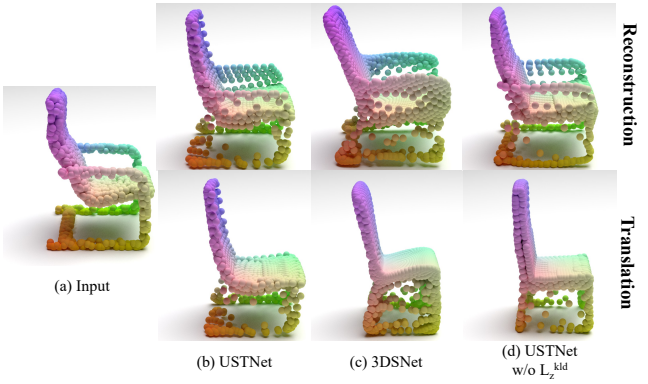

Figure 4: *Comparison of reconstructed and translated chairs by 3DSNet, USTNet, and USTNet without L_z^{KL}, conditioned on the same input armchair shape. 3DSNet generates the worst reconstructed output since the translation process it uses lacks content features such as the height and shape of the chair surface. In contrast, USTNet and USTNet without L_z^{KL} generate similar reconstructed shapes, but USTNet without L_z^{KL} cannot capture structure information,* e.g., *the inclination angle of the chair's back.*

(Secs. 3.2 and 3.3), its fusion network component (Sec. 3.4), and also discuss the reconstruction loss and our full objective functions (Sec. 3.5). Finally, we present USTNet-M that extends USTNet to multi-domain translation (Sec. 3.6).

3.1. Method Overview

The core goal of our method is to learn a mapping between two shape domains X_1 and X_2 containing the same semantic information without using paired training data. Our network contains several components (see Fig. 2): a *style* encoder E_s, a *content* encoder E_c, an adaptive fusion network F, a generator G, and a content discriminator D. Given two input shapes $x_i \in X_i$, the style and content encoders E_s and E_c map the shape into the style and content spaces as $z_s^{x_i}$ and $z_c^{x_i}$, respectively. The fusion network F synthesizes a latent code z^{x_i} capturing both the style and content representations. The generator G maps this vector into point cloud space, *i.e.*, creates a 3D point cloud model of the generated shape. Finally, the content discriminator D aims to distinguish the domain membership of content representations.

Our network contains two training stages (see again Fig. 2): The reconstruction stage (blue arrows in the figure) yields disentangled representations; the translation stage (green arrows) achieves shape translation. In the reconstruction stage, each input shape is encoded into a style code and a content code. The fusion network F and generator G create a reconstructed shape from these codes. In the translation stage, the generated shape is conditioned on a style code from the target domain and a content code from the source domain. These two stages are used alternatively during the training phase. In the inference phase, only the encoders E_s and E_c, the fusion network F, and the generator G are used to produce translated shapes $x_{1\to 2}$ and $x_{2\to 1}$ for the input shapes x_1 and x_2. Our network enables bi-directional shape-to-shape translation of two input shapes with a single forward translation.

3.2. Content Adversarial Loss

To obtain the disentangled style and content representations by the two encoders E_c and E_s (which have the same structure up to the output layers), we use a content adversarial loss to provide an additional constrain for the content representation, similar to [LTH*18], but now applied to the 3D shape domain rather than the 2D image domain. Specifically, we impose a content discriminator D, which aims to distinguish the domain membership of content representations encoded by E_c. The content encoder E_c learns to produce indistinguishable content representations on the domain membership for the content discriminator D. The content encoder E_c encodes the common information between different domains, while the style encoder E_s captures the domain-specific traits of shapes. If we train the given architecture without the content adversarial loss, the content encoder has no guidance for distinguishing content and style information, so the two encoders will produce similar latent codes which cannot be used in the shape translation process. Specifically, we formulate the content adversarial loss for D as:

$$L_c^D = \mathbb{E}_{x_1\in X_1}\left(\log\left(1-D(E_c(x_1))\right)\right)+\mathbb{E}_{x_2\in X_2}\left(\log\left(D(E_c(x_2))\right)\right),$$

and the content adversarial loss for the auto-encoder module as:

$$\begin{aligned} L_c^{AE} = & \\ & \mathbb{E}_{x_1\in X_1}\left(\frac{1}{2}\log(1-D(E_c(x_1)))+\frac{1}{2}\log(D(E_c(x_1)))\right) \\ +&\mathbb{E}_{x_2\in X_2}\left(\frac{1}{2}\log(1-D(E_c(x_2)))+\frac{1}{2}\log(D(E_c(x_2)))\right) \end{aligned}$$

3.3. Latent Consistency Loss

In shape-to-shape translation, ensuring that the translated shape exhibits content features of the source shape and style features of the target shape is crucial. Prior cross-domain translation methods [LTH*18, HLBK18, ZPIE17, SGST20] impose a cross-cycle consistency loss for images or shapes to constrain the translated results. Specifically, such models use translated results as inputs to perform the second translation to generate cross-cycle reconstruction results of the original inputs, and use the cross-cycle consistency loss to enforce this constraint. These methods include domain-specific encoders and/or decoders, making the embedding of shapes ambiguous for different domains, thus requiring indirect constraints to control the translation results.

In contrast to the above, our network shares each block among domains. Hence, we can constrain the style and content codes *directly* between input and translated shapes to help our network to generate shapes containing the target style. Note that, in contrast to this, indirect constraints at image or point cloud levels are ambiguous since they try to constrain the codes of inputs and corresponding cycle-reconstructed outputs. Rather, we state that it is desirable that an input shape x_i and the translated shape $x_{i\to j}$ have the same content code, while the target shape x_j and the translated shape $x_{i\to j}$ share the same style representation. Using these explicit constraints, our method helps to achieve the desired disentangled representations. Moreover, the constraints also reduce the computation cost by removing a second translation phase. We model these

constraints by the latent consistency loss

$$L_l = \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} L_l^{x_i \to j},$$

which aggregates the individual latent consistency losses between x_i and $x_{i\to j}$ given by

$$\begin{aligned} L_l^{x_i\to j} =& \|E_c\big(G(F(E_c(x_i), E_s(x_j)))\big) - E_c(x_i)\|_2 \\ &+ \|E_s\big(G(F(E_c(x_i), E_s(x_j)))\big) - E_s(x_j)\|_2, \end{aligned}$$

where n is the number of shape domains and $\|\cdot\|_2$ denotes the L_2 norm. The latent consistency loss L_l encourages our network to learn an invertible mapping between disentangled latent spaces and the 3D point cloud space. This constraint is also helpful for learning disentangled representations since successful disentanglement means the content codes of shapes should be constant when the style codes are changed, and vice versa.

3.4. Fusion Network

The fusion network F (Fig. 2) is designed to synthesize a uniform latent space from the style space and the content space, which provides conditions for the generator to learn the mapping from the latent space to the 3D point cloud space. Inspired by AdaIN [HB17], we propose an affine transformation method to fuse content and style codes. Specifically, our fusion network F uses two multi-layer perceptrons (MLPs) M_1 and M_2 with the identical structure to produce the bias (M_1) and scale (M_2) parameters of the affine transformation from the first and second halves of the style code, respectively, and then computes the fused latent code based on the content code. When the content code and the style code are encoded from the same shape x_i, the fusion network F will synthesize them to the code z^{x_i} for reconstruction by our generator G as

$$z^{x_i} = M_1(z_{s1}^{x_i}) + M_2(z_{s2}^{x_i}) * z_c^{x_i},$$

where $z_c^{x_i} = E_c(x_i)$ and the codes $z_{s1}^{x_i}$ and $z_{s2}^{x_i}$ are obtained by simply splitting the code $z_s^{x_i} = E_s(x_i)$ in two halves. If the two codes are from shapes from different domains, such as the content code of x_i and the style code of x_j, our fusion network F produces the code $z^{x_i\to j}$ for generating the translated shape $x_{i\to j}$ as

$$z^{x_i\to j} = M_1(z_{s1}^{x_j}) + M_2(z_{s2}^{x_j}) * z_c^{x_i}.$$

In contrast to AdaIN, our fusion network F does not use Instance Normalization (IN) on the content representation. Rather, we use a Kullback-Leibler (KL) loss L_c^{KL} to align the content space with a prior standard Gaussian distribution $\mathcal{N}(0,I)$, defined by

$$L_c^{KL} = \mathrm{KL}\big(p(z_c^x|x)\big\|\mathcal{N}(0,I)\big).$$

This achieves an affine transformation of content codes without the need for IN. The irreversibility of IN makes it inappropriate for our architecture, which, in contrast, tries to learn an invertible mapping between the latent and point cloud spaces. Since it is hard to measure the similarity of style information directly, we use the distance between the style codes of shapes as a proxy to quantify it. Specifically, the mapping between style information in point cloud space and the style code in style latent space needs to be invertible to ensure that style similarity is well captured by Euclidean distance.

In addition, to assist our two MLPs, M_1 and M_2, to learn an *adaptive* distance between two style spaces (coming from different domains), we introduce a new loss L_z^{KL} which aims to align the sum of fused representations from different domains with a prior Gaussian distribution on each dimension. We define this new loss as

$$L_z^{KL} = \mathrm{KL}\left(\sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} p(z^{x_i\to j}|x_i, x_j)\big\|\mathcal{N}(0,I)\right). \tag{1}$$

Under this loss, M_1 and M_2 can automatically select dimensions of z^{x_i} and $z^{x_i\to j}$ which represent the discrepancy information between different domains.

3.5. Reconstruction Loss

Besides the above L_c^{KL} and L_z^{KL} losses, we need a reconstruction loss to train our autoencoder. The reconstruction loss compels the style and content encoders E_s and E_c to produce meaningful representations and the generator G to synthesize output shapes as similar as possible to the inputs. To produce high-quality codes with low computational cost, we use PointNet [QSMG17] as the backbone for the style and content encoders. Our generator is a custom version of AtlasNet [GFK*18], which can generate point clouds based on patches, or SP-GAN [LLHF21], a state-of-the-art point cloud GAN for learning a bias for each point.

To define a reconstruction loss, we need a method to measure the similarity of the input and reconstructed shape. For this, we use the bidirectional Chamfer Distance (CD) between two shapes. The Chamfer distance computes the distance between two point clouds x_1 and x_2 by summing up the squared distances between each point in x_1 to its closest point in x_2. This encourages our autoencoder to capture features as completely as possible. In contrast to CD, the Hausdorff distance, another typical instrument to compare 3D shapes, is sensitive to outlier points in the clouds, which hinders it from being an efficient loss for point cloud learning. The Earth Mover's Distance (EMD) [RTG00], yet another common metric for comparing images or shapes, was also used for 3D shape reconstruction [FSG17], and showed to deliver results more focused on local features of the input point cloud than when using CD, which is not desired in our context. Given all the above, we set our reconstruction loss as

$$L_r = \frac{1}{n}\sum_{i=1}^{n} \mathrm{CD}\Big(G\big(F(E_c(x_i), E_s(x_i))\big), x_i\Big).$$

Putting it all together, our complete loss function is given by

$$\begin{aligned} L_{E,F,G} &= -\lambda_c L_c^{AE} + \lambda_r L_r + \lambda_l L_l + \lambda_c^{KL} L_c^{KL} + \lambda_z^{KL} L_z^{KL}, \\ L_D &= L_c^D, \end{aligned} \tag{2}$$

where $L_{E,F,G}$ is used to update parameters of the encoder network, the fusion network, and the generator network, and L_D is used to update parameters of the discriminator network. The λ values in

$L_{E,F,G}$ are hyperparameters giving the weights of the various terms (set as discussed next in Sec. 4.3).

3.6. Multi-Domain Shape-to-Shape Translation

Many styles exist in real application scenarios. This makes dual-domain translation methods inefficient for multi-domain translation tasks [CCK*18]. Using existing methods, learning all mappings among domain pairs means that we must train $k(k-1)$ translators (for LOGAN and UNIST) or k autoencoders (for 3DSNet). Meanwhile, each autoencoder cannot use the entire training dataset but only the input shapes belonging to the related two domains or even just one domain.

For multi-domain translation, given k domains $\{X_n\}_{n=1...k}$, the inputs of the network are two shapes (x_i, x_j) and their one-hot domain codes (z_i^d, z_j^d), which are randomly sampled ($x_i \in X_i, x_j \in X_j$, $z^d \in \mathbb{R}^k$). Compared to prior shape translation networks, our encoders and generator are general for all domains, so our USTNet-M for multi-domain translation only needs to be adapted for F and D to leverage the domain codes of the input shapes. For this, we extend D to a multi-classification discriminator. The content encoder E_c still tries to confuse the discriminator to encode the domain-invariant information. L_z^{KL} in F aligns the sum of distributions among all domains with a prior Gaussian distribution on each dimension. Note that the domain labels are only used in the training phase. For the inference phase, the network can recognize the style features of the target shape, so it can translate shapes from multi-domains without using the domain label.

4. Results

We validate our proposed framework through several experiments, and qualitative and quantitative analyses. Section 4.1 outlines the datasets used in our experiments. Section 4.2 introduces the evaluation metrics we use. Section 4.3 provides implementation details including hyperparameter settings and network architecture details. Sections 4.4 and 4.5 show our results for dual-domain and multi-domain shape translation, respectively. Section 4.6 illustrates the results and analyses of an ablation study.

4.1. Datasets

As we lack datasets for specific shape-to-shape translation validation, we use pairs of subcategories of established benchmarks for 3D reconstruction, such as ShapeNet [CFG*15] and SMAL [ZKJB17] referring to LOGAN and 3DSNet.

ShapeNet is a widely used dataset in shape analysis and point cloud learning, containing 51300 unique 3D models covering 55 common object categories. To generate translated results and show the preservation of content information, we choose specific category pairs which have common semantic information but different characteristics, *e.g.*, *straight chairs* and *armchairs*. In our experiments, we use the point cloud version of ShapeNet by random sampling from models provided by [GFK*18].

SMAL is a 3D non-rigid animal shape dataset similar to the SMPL [LMR*15] body models. SMAL is suitable for shape translation as it contains different animal categories which have a common content distribution expressed by diverse poses of animals. To obtain the point clouds from SMAL, we use the Point Cloud Library [RC11] to sample surface points randomly with a leaf size of 0.001.

4.2. Evaluation Metrics

No universally accepted quantitative evaluation metrics exist for unsupervised 3D shape-to-shape translation. To evaluate style transfer results, 3DSNet [SGST20] proposed using the Style Transfer Score for shape translation to measure the distance of input and generated shapes in latent space. However, we have found that this generates high scores even when the translation process discards the content information entirely.

UNIST [CMS*21] claims that it is difficult to design a general evaluation metric because *correct* translation for shapes can be highly varied and dependant on the selected domains. For example, for the translation between armchairs and straight chairs, UNIST used the *one-sided Chamfer Distance* to evaluate the reserved content information. Specifically, the one-sided Chamfer Distance from the straight chair to the corresponding armchair is computed regardless of the direction of the translation. We argue that this metric cannot evaluate the change of style information. For example, if the translated and reconstructed results are the same (which means that the translation failed), the one-sided Chamfer Distances will still be low. Hence, to evaluate the style information, we design an extra classifier to predict the domains of reconstructed and translated results and measure whether the translation worked correctly. For this, we use a PointNet-based network pre-trained on the armchair/straight chair dataset for 50 epochs, with a training accuracy of 99.69% and a testing accuracy of 96.27%. We next use the one-sided Chamfer Distance and the classification accuracy of the generated shapes to evaluate our shape-to-shape translation network quantitatively. A method that can generate translated chairs with high classification accuracy and low one-sided Chamfer Distance is, thus, a method that can effectively disentangle style and content information.

4.3. Experimental Setting

We implemented USTNet with PyTorch [PGC*17]. For the content and style encoders, we use a PointNet architecture consisting of three 1D convolution layers followed by two fully-connected layers. We set the size of the content vector z_c to 1024 and the length of the style vector z_s to 512. For the fusion block F, we use architecture with two MLPs, each with two fully-connected layers and a ReLU layer. We use two alternative architectures for the generator G: AtlasNet [GFK*18] with 25-patches and SP-GAN [LLHF21]. We modify the global prior of SP-GAN from uniform constant points to points randomly sampled on a sphere, which enables our method to change the point count for the outputs in the inference phase. This characteristic also exists in vanilla AtlasNet, since primary points are sampled on each patch. We set the sampling point count from each input shape to 2500. For the content discriminator used to distinguish latents, we use an MLP architecture containing three full-connected layers and two ReLU layers.

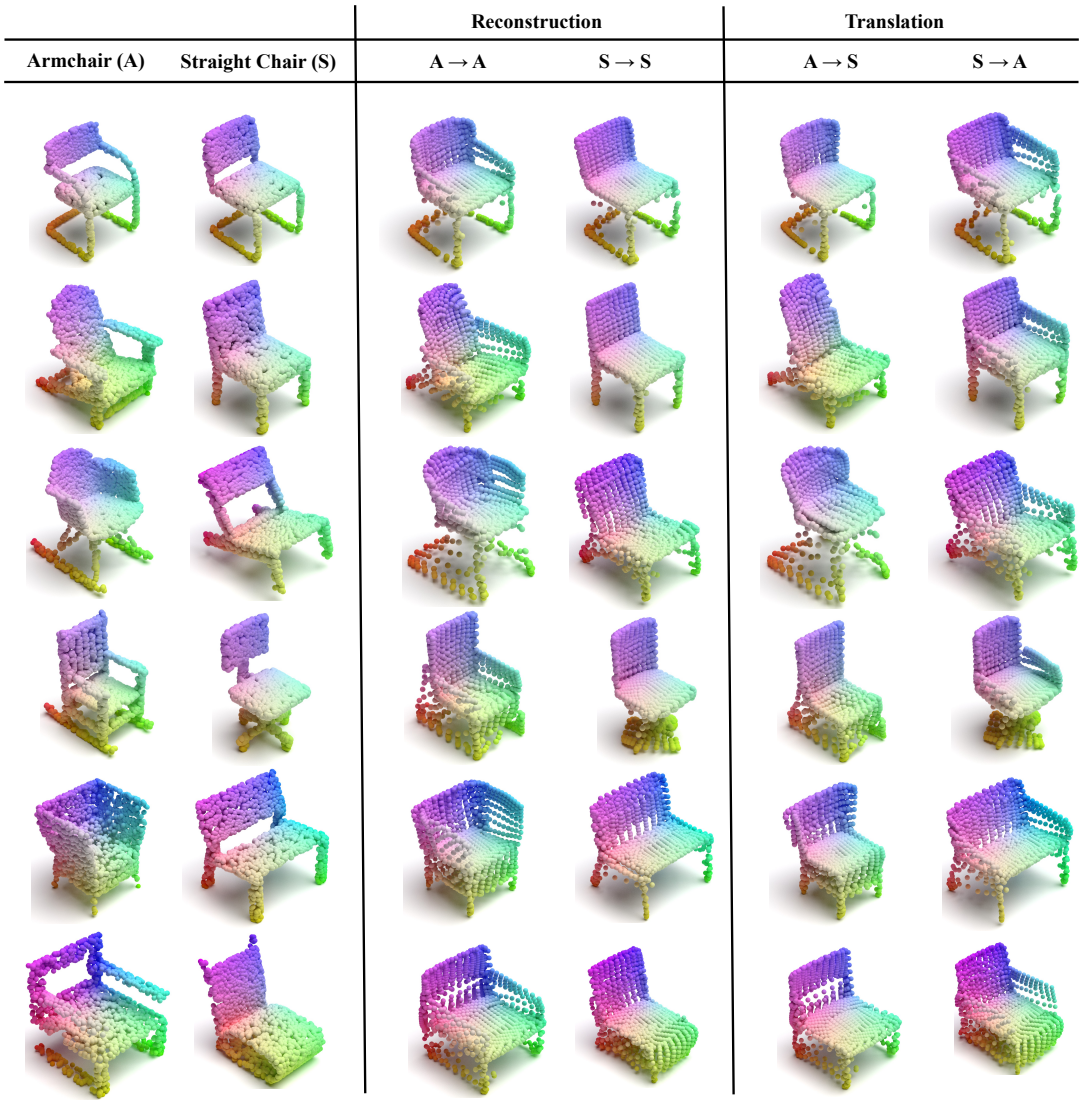

Figure 5: *Results obtained by USTNet on 6 pairs from the* armchair ↔ straight chair *dataset. The first two columns show the inputs of our network; the second two columns show the reconstruction results for each of the two inputs; the last two columns depict translation results, including from* armchair *to* straight chair *(A → S) and* straight chair *to* armchair *(S → A). The decoder is an AtlasNet-based structure with 25-patches.*

We train our framework for 180 epochs using the Adam optimizer [KB15] with a batch size of 8, initial learning rate of 0.001 for AtlasNet and 0.0001 for SP-GAN, a decay factor of 0.1 at epochs 120, 140 and 145, and exponential decay rates $(\beta_1, \beta_2) = (0.5, 0.999)$. The loss function weights (Eqn. 2) are set to $\lambda_c = 1$, $\lambda_r = 5$, $\lambda_c^{KL} = 1$, $\lambda_l = 0.1$, and $\lambda_z^{KL} = 0.01$.

During inference, we generate reconstruction and translation results with 2500 points for each shape and render them using Mitsuba [NVZJ19] for a fair comparison with 3DSNet (see Fig. 3). We ran our model on a PC with an NVIDIA GeForce GTX 3090 GPU for all experiments. For the *chairs* dataset, training time is 34 minutes for AtlasNet and 166 minutes for SP-GAN, and inference time is 0.12s per shape. For the SMAL dataset, training times are 8 hours and 10 minutes for AtlasNet; inference times are 0.12s per shape.

4.4. Evaluation of Shape-to-Shape Translation

We first validate our approach on three different pairs of subcategories of a family from ShapeNet, *i.e.*, *armchairs* and *straight chairs*, *fighter* and *jet*, and *table* and *chair*. *Armchairs* and *straight chairs* having 1995 and 1974 shapes, respectively, are widely used in shape research testing due to their similar structures but also marked discrepancies. From each of the paired sets, we randomly choose 30% shapes as the test set, with the rest used for training our network. The following results (all generated on the test set)

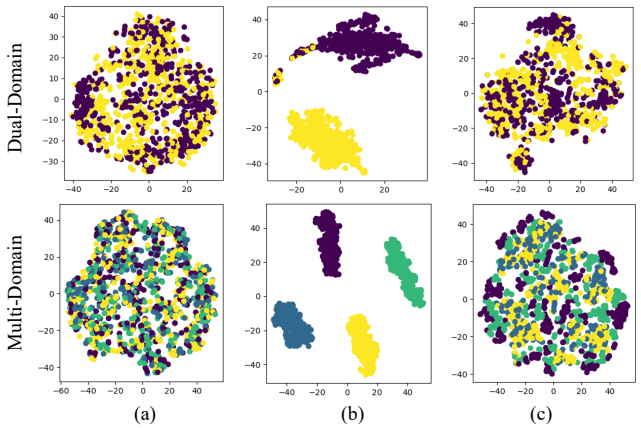

Figure 6: *Visualizing disentangled representations of* armchair *and* straight chair *latent codes generated by USTNet in the (a) content space, (b) style space, and (c) fusion space. Top row: Dual-domain translation, purple = armchairs, yellow = straight chairs. Bottom row: Multi-domain translation, purple = horses, yellow = hippos, blue = dogs, green = cows.*

show that USTNet can recognize such discrepancies and generate reasonable translation outputs.

Due to the performance of the encoder and decoder, the translated results may not be similar in detail to the source inputs. To verify the effectiveness of translation and disentanglement, the structural similarity between the reconstructed and translated shapes is more critical than the similarity of such shapes to the input ones. For example, if the armchairs and straight chairs generated by a network in its translation and reconstruction processes have the same structures (disregarding arm rests), the network can be considered as a successful disentanglement framework for the *armchairs* ↔ *straight chairs* dataset. Figure 5 shows several results of the reconstruction and translation for six pairs of straight chairs and armchairs randomly selected from the two subcategories. We see that our method generates realistic and natural translated outputs. In the translation process, USTNet successfully removes arm rests of armchairs and, conversely, adds arm rests to straight chairs while other parts of the shapes stay fixed.

To provide more insight into the translation process, Fig. 1 shows the generated shapes by uniformly interpolating between the armchair and straight chair styles while keeping the content feature constant. We see that domain-invariant traits such as the legs and backrests of chairs are stable, while the interpolated shapes still look natural as the arm rests change. These results show that our method can adaptively embed shapes from different domains into a common style space and clearly disentangle style and content information. In contrast, 3DSNet, which is also a disentanglement-based translation method, cannot interpolate in style space since it assumes that style spaces are independent.

To evaluate the quality of our shape translation on the *armchairs* and *straight chairs* datasets, we compute the one-sided Chamfer Distance and domain classification accuracy metric, and compare these values with those of 3DSNet. Table 1 and Table 2 show the one-sided Chamfer Distance comparison. In Table 1, we see that USTNet significantly outperforms 3DSNet in both translation directions, which means that USTNet can recognize and preserve more shared structure information in the translation process. Table 2 shows that compared with LOGAN and UNIST, our method also preserves more content information. Note that the numerical differences between the two tables are due to different normalization methods for the input point clouds. Table 3 shows the comparison of the classification accuracy. Higher accuracy means more generated shapes are classified into the target domain. The results show that both methods can faithfully generate translated shapes. Figures 3 and 4 qualitatively compare the translated results for 3DSNet and our USTNet. Compared to 3DSNet, our method preserves more domain-invariant detail features such as the height of the seat and the outline of the backrest. Figure 7 shows more results of our method for the *fighter* ↔ *jet*, and *table* ↔ *chair* datasets, with similar outcomes as those discussed for Figures 3 and 4.

Figure 6 (top row) visualizes the three latent spaces of USTNet: the *style* space, *content* space, and the *fusion* space, by projecting the *armchair* and *straight chair* latent codes to 2D using t-SNE. The representations in the fusion space (c) are neither completely confused nor distinguishable because the domain-invariant and domain-specific features are entangled. The disentangled latent distributions of the two domains in the content (a) and style (b) spaces are extremely confused, respectively very well separated. The few mixed-up yellow points on the left of the top purple cluster in the image (b) are due to some wrongly labeled shapes in the ShapeNet database. Overall, the above support our claims of the effectiveness of disentanglement achieved by USTNet.

Table 1: *One-sided Chamfer Distance for 3DSNet, USTNet, and USTNet without* L_z^{KL} *on the* armchair ↔ straight chair *dataset pair. Results are multiplied by* 10^2 *for ease of reading. The best values are highlighted in bold.*

Method	Decoder	*armchair* → *straight chair*	*straight chair* → *armchair*
3DSNet	MeshFlow	3.49	3.02
	AtlasNet	3.17	3.00
USTNet	AtlasNet	**1.66**	**1.47**
	SP-GAN	2.53	2.14
USTNet w/o L_z^{KL}	AtlasNet	4.55	4.21
	SP-GAN	5.18	4.58

Table 2: *One-sided Chamfer Distance for UNIST, LOGAN, and USTNet on the* armchair ↔ straight chair *dataset pair. The normalization method is the same as that of UNIST. Results are multiplied by* 10^2 *for ease of reading. The best values are highlighted in bold.*

Method	Decoder	*armchair* → *straight chair*	*straight chair* → *armchair*
UNIST	-	2.34	2.35
LOGAN	-	2.49	2.73
USTNet	AtlasNet	**2.33**	**2.11**

Table 3: *Classification accuracy for 3DSNet, USTNet, and USTNet without L_z^{KL},* armchair ↔ straight chair *dataset pair. Higher values mean more generated shapes are classified to the target domain. The best values are highlighted in bold.*

Method	Decoder	*armchair → straight chair*	*straight chair → armchair*	*armchair → armchair*	*straight chair → straight chair*
3DSNet	MeshFlow	87.82%	92.44%	**99.79%**	98.74%
	AtlasNet	56.63%	68.37%	98.47%	**98.98%**
USTNet	AtlasNet	93.13%	94.79%	95.21%	96.67%
	SP-GAN	**93.57%**	**97.72%**	97.72%	95.02%
USTNet w/o L_z^{KL}	AtlasNet	82.92%	96.14%	97.31%	95.63%
	SP-GAN	84.44%	93.36%	95.23%	94.19%

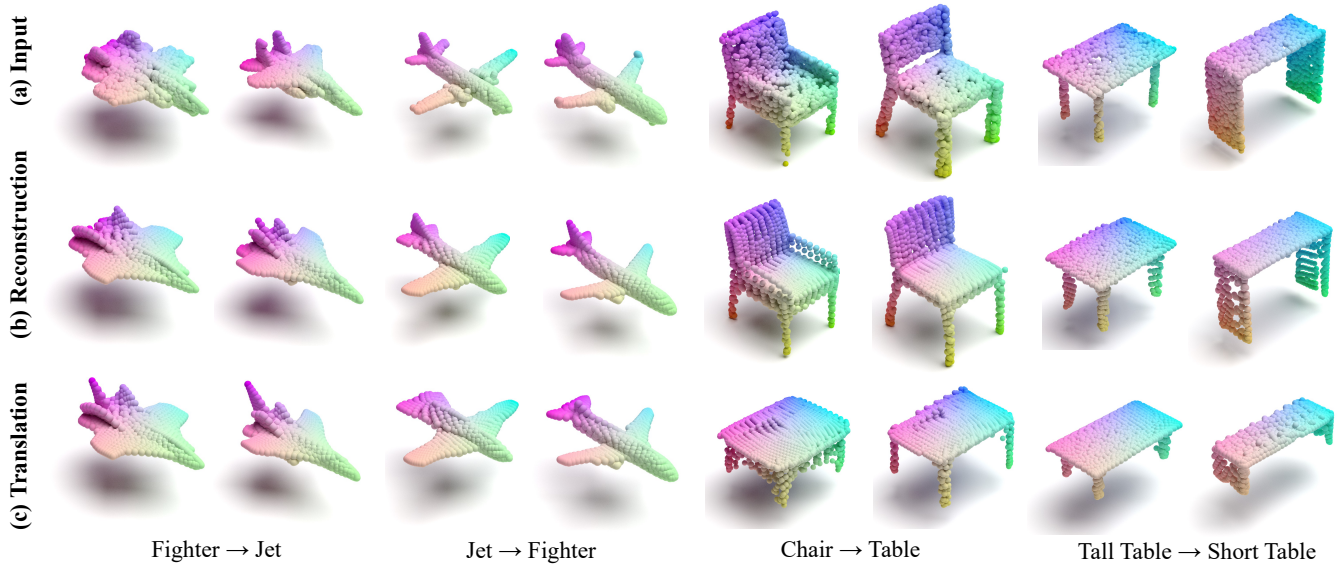

Figure 7: *Translation results generated by USTNet on the* fighter ↔ jet, chair → table *and* tall table → short table *datasets. Note how USTNet considers the type of the aircraft tail as a domain trait, the details of chair legs as shared content information, and the height of tables as the style feature, respectively. More results are shown in the supplementary material.*

4.5. Multi-domain Shape-to-Shape Translation

To evaluate the effectiveness of our multi-modal shape translation approach, we apply our model to the SMAL database on shapes from different domains which contain common semantics (the poses of animals). For a pair of domains in SMAL, the common information is the pose of the animal, including the orientation of its limbs, head, and tail. Successful translation from a hippo shape to the domain of horses, for example, means generating a horse shape with the same pose as the input hippo shape. We generate 1000 shapes for the training set and 500 shapes as the validation set for each category by sampling the model parameters from a Gaussian distribution with a variance of 0.2.

Figure 8 shows the results of USTNet-M for shape-to-shape translation on multi-domain inputs. We perform translation among four domains of SMAL (hippos, horses, cows, and dogs). In Fig. 8, the first column shows the source shapes, and the first row shows the target shapes. The other shapes in the figure are generated by our method's reconstruction and translation results. Each row has the same content (pose) feature, and each column has the same style feature. For example, in the second row of Fig. 8, using the source shape (far left) from the horse domain, our network can generate multiple translated results according to the target shapes shown in the first row. The results show the effectiveness of our multi-domain translation.

Figure 6 (bottom row) illustrates the latent distributions of domains in the disentangled content, style, and fusion spaces, projected with t-SNE (as discussed earlier). Significantly, we see that features in the style space are perfectly well separated because of USTNet's successful disentanglement.

4.6. Ablation Study

We next qualitatively and quantitatively evaluate the influence of the loss L_z^{KL} (Eqn. 1) in the fusion network F for content preservation and disentangled representation learning. The bottom two rows of Tables 1 and 3 show the results of USTNet with and without L_z^{KL} for the one-sided Chamfer Distance and classification accuracy. These results indicate that USTNet can still change the style of input shapes without L_z^{KL}. However, the one-sided Chamfer Distance value is significantly higher than when using L_z^{KL}, which means

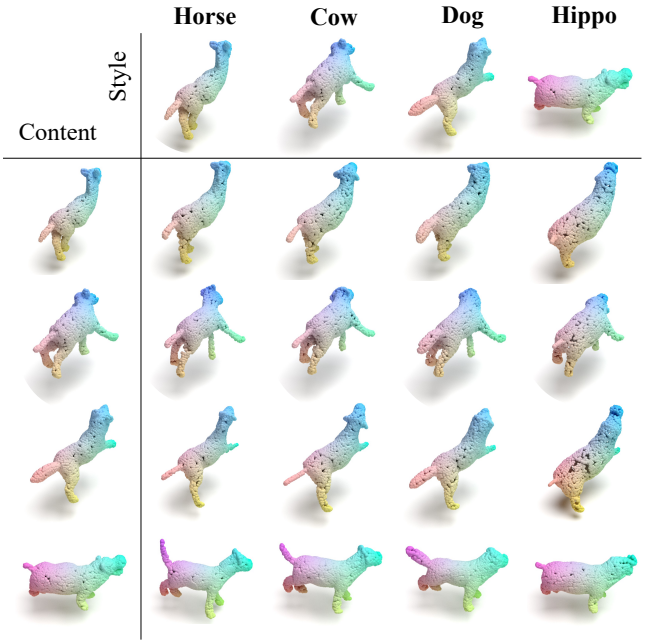

Figure 8: *Results obtained by USTNet-M on four categories of SMAL. Each row has the same content (pose) feature, and each column has the same style feature. Shapes on the diagonal are reconstruction results. Off-diagonal shapes are translated shapes.*

that the translation results preserve less content information. This confirms the added value of the L_z^{KL} term.

Figure 4 (b, c) compares the results of USTNet with, respectively, without the L_z^{KL} term. When not using this term, USTNet can still generate good reconstruction results (top row). However, the translated results have obvious structural changes, *e.g.*, the wrong-looking inclination of the chair back and the sharp aspect of the chair seat in Fig. 4c (bottom row), because the network cannot correctly disentangle style and content information of shapes. Hence, we claim that the loss term L_z^{KL} is critical for realizing a good disentanglement and content-stable shape translation.

5. Conclusion

We have presented USTNet, a novel disentangled representation framework for shape-to-shape translation with unpaired inputs. USTNet disentangles the input shape into a content space that encodes common information between domains and a style space that models domain-specific information. For this, we use a fusion network with two losses to adaptively synthesize content and style codes, and to embed style codes from different domains into a common space. We use a content adversarial loss to encourage the content encoder to learn domain-invariant features, and a latent consistency loss to ensure translated shapes consist of content information of the source shape and style information of the target shape.

We extend our network to USTNet-M to tackle the multi-domain translation problem without introducing additional blocks. Qualitative and quantitative results show that our method produces realistic and natural reconstructed and translated shapes and generates equal or better results than 3DSNet.

Still, our method has several limitations. The generated shapes by USTNet cannot completely preserve all details of the input shapes, such as the fine structure of chair legs and backs, mainly due to the max-pooling operation in encoders (see some of the results in Figs. 3 and 5). In future work, we aim to further research shape-to-shape translation to preserve the detail of source shapes over the whole translation process.

Acknowledgments

This work was partially supported by the Zhejiang Provincial Natural Science Foundation (LGF21F20012) and Zhejiang Provincial Science and Technology Program i n China (2021C03137).

References

[AB20] ARSHAD M. S., BEKSI W. J.: A progressive conditional generative adversarial network for generating dense and colored 3D point clouds. In *Proc. Intl Conf 3D Vision* (2020), Struc V., Fernández F. G., (Eds.), IEEE, pp. 712–722. 3

[ADMG18] ACHLIOPTAS P., DIAMANTI O., MITLIAGKAS I., GUIBAS L. J.: Learning representations and generative models for 3D point clouds. In *Proc. ICML* (2018), Dy J. G., Krause A., (Eds.), vol. 80 of *PMLR*, pp. 40–49. 3

[BCV13] BENGIO Y., COURVILLE A. C., VINCENT P.: Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell. 35*, 8 (2013), 1798–1828. 2, 4

[CCK*18] CHOI Y., CHOI M., KIM M., HA J., KIM S., CHOO J.: StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proc. IEEE CVPR* (2018), pp. 8789–8797. 4, 7

[CDH*16] CHEN X., DUAN Y., HOUTHOOFT R., SCHULMAN J., SUTSKEVER I., ABBEEL P.: InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *Proc. NIPS* (2016), pp. 2172–2180. 4

[CFG*15] CHANG A. X., FUNKHOUSER T. A., GUIBAS L. J., HANRAHAN P., HUANG Q., LI Z., SAVARESE S., SAVVA M., SONG S., SU H., XIAO J., YI L., YU F.: ShapeNet: An information-rich 3D model repository. *CoRR abs/1512.03012* (2015). 7

[CKF*21] CHEN Z., KIM V. G., FISHER M., AIGERMAN N., ZHANG H., CHAUDHURI S.: DECOR-GAN: 3D shape detailization by conditional refinement. In *CVPR* (2021), Computer Vision Foundation / IEEE, pp. 15740–15749. 3

[CMS*21] CHEN Q., MERZ J., SANGHI A., SHAYANI H., MAHDAVI-AMIRI A., ZHANG H.: UNIST: unpaired neural implicit shape translation network. *CoRR abs/2112.05381* (2021). 3, 7

[CRW*20] CHAUDHURI S., RITCHIE D., WU J., XU K., ZHANG H. R.: Learning generative models of 3D structures. *Comput. Graph. Forum 39*, 2 (2020), 643–666. 1

[FSG17] FAN H., SU H., GUIBAS L. J.: A point set generation network for 3D object reconstruction from a single image. In *Proc. IEEE CVPR* (2017), pp. 2463–2471. 6

[GEB16] GATYS L. A., ECKER A. S., BETHGE M.: Image style transfer using convolutional neural networks. In *Proc. IEEE CVPR* (2016), pp. 2414–2423. 4

[GFK*18] GROUEIX T., FISHER M., KIM V. G., RUSSELL B., AUBRY M.: AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *Proc. IEEE CVPR* (2018). 6, 7

[GLL*22] GUAN Y., LIU H., LIU K., YIN K., HU R., VAN KAICK O., ZHANG Y., YUMER E., CARR N., MECH R., ZHANG H. R.: FAME: 3D shape generation via functionality-aware model evolution. *IEEE Trans. Vis. Comput. Graph. 28*, 4 (2022), 1758–1772. 1

[HB17] HUANG X., BELONGIE S. J.: Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV* (2017), IEEE Computer Society, pp. 1510–1519. 6

[HLBK18] HUANG X., LIU M., BELONGIE S. J., KAUTZ J.: Multimodal unsupervised image-to-image translation. In *Proc. ECCV* (2018), vol. 11207 of *Lecture Notes in Computer Science*, Springer, pp. 179–196. 4, 5

[HLvK*17] HU R., LI W., VAN KAICK O., HUANG H., AVERKIOU M., COHEN-OR D., ZHANG H. R.: Co-locating style-defining elements on 3D shapes. *ACM Trans. Graph. 36*, 3 (2017), 33:1–33:15. 1

[HMP*17] HIGGINS I., MATTHEY L., PAL A., BURGESS C., GLOROT X., BOTVINICK M., MOHAMED S., LERCHNER A.: beta-VAE: Learning basic visual concepts with a constrained variational framework. In *Proc. ICLR (Poster)* (2017), OpenReview.net. 4

[HXX*20] HUI L., XU R., XIE J., QIAN J., YANG J.: Progressive point cloud deconvolution generation network. In *Computer Vision - ECCV* (2020), Vedaldi A., Bischof H., Brox T., Frahm J., (Eds.), vol. 12360 of *Lecture Notes in Computer Science*, Springer, pp. 397–413. 3

[IZZE17] ISOLA P., ZHU J., ZHOU T., EFROS A. A.: Image-to-image translation with conditional adversarial networks. In *Proc. IEEE CVPR* (2017), pp. 5967–5976. 4

[JYF*20] JING Y., YANG Y., FENG Z., YE J., YU Y., SONG M.: Neural style transfer: A review. *IEEE Trans. Vis. Comput. Graph. 26*, 11 (2020), 3365–3385. 1

[KB15] KINGMA D. P., BA J.: Adam: A method for stochastic optimization. In *Proc. ICLR (Poster)* (2015). 8

[KM18] KIM H., MNIH A.: Disentangling by factorising. In *Proc. ICML* (2018), vol. 80 of *Machine Learning Research*, PMLR, pp. 2654–2663. 4

[KSB18] KUMAR A., SATTIGERI P., BALAKRISHNAN A.: Variational inference of disentangled latent concepts from unlabeled observations. In *Proc. ICLR (Poster)* (2018), OpenReview.net. 4

[LBK17] LIU M., BREUEL T. M., KAUTZ J.: Unsupervised image-to-image translation networks. In *Proc. NIPS* (2017), pp. 700–708. 4

[LBL*19] LOCATELLO F., BAUER S., LUCIC M., RÄTSCH G., GELLY S., SCHÖLKOPF B., BACHEM O.: Challenging common assumptions in the unsupervised learning of disentangled representations. In *ICML* (2019), vol. 97 of *Proceedings of Machine Learning Research*, PMLR, pp. 4114–4124. 4

[LH21] LUO S., HU W.: Diffusion probabilistic models for 3D point cloud generation. In *CVPR* (2021), Computer Vision Foundation / IEEE, pp. 2837–2845. 3

[LLHF21] LI R., LI X., HUI K., FU C.: SP-GAN: sphere-guided 3D shape generation and manipulation. *ACM Trans. Graph. 40*, 4 (2021), 151:1–151:12. 3, 6, 7

[LMR*15] LOPER M., MAHMOOD N., ROMERO J., PONS-MOLL G., BLACK M. J.: SMPL: a skinned multi-person linear model. *ACM Trans. Graph. 34*, 6 (2015), 248:1–248:16. 7

[LTFO20] LIN Z., THEKUMPARAMPIL K. K., FANTI G. C., OH S.: InfoGAN-CR and modelcentrality: Self-supervised model training and selection for disentangling GANs. In *Proc. ICML* (2020), vol. 119 of *Machine Learning Research*, PMLR, pp. 6127–6139. 4

[LTH*18] LEE H., TSENG H., HUANG J., SINGH M., YANG M.: Diverse image-to-image translation via disentangled representations. In *ECCV (1)* (2018), vol. 11205 of *Lecture Notes in Computer Science*, Springer, pp. 36–52. 2, 3, 4, 5

[LTM*20] LEE H., TSENG H., MAO Q., HUANG J., LU Y., SINGH M., YANG M.: DRIT++: diverse image-to-image translation via disentangled representations. *Int. J. Comput. Vis. 128*, 10 (2020), 2402–2417. 4

[LWS*18] LIU Y., WEI F., SHAO J., SHENG L., YAN J., WANG X.: Exploring disentangled feature representation beyond face identification. In *Proc. IEEE CVPR* (2018), pp. 2080–2089. 4

[LZZ*19] LI C., ZAHEER M., ZHANG Y., PÓCZOS B., SALAKHUTDINOV R.: Point cloud GAN. In *Proc. ICLR* (2019), OpenReview.net. 3

[NVZJ19] NIMIER-DAVID M., VICINI D., ZELTNER T., JAKOB W.: Mitsuba 2: a retargetable forward and inverse renderer. *ACM Trans. Graph. 38*, 6 (2019), 203:1–203:17. 8

[PGC*17] PASZKE A., GROSS S., CHINTALA S., CHANAN G., YANG E., DEVITO Z., LIN Z., DESMAISON A., ANTIGA L., LERER A.: Automatic differentiation in PyTorch. 7

[QSMG17] QI C. R., SU H., MO K., GUIBAS L. J.: PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proc. IEEE CVPR* (2017), pp. 77–85. 6

[RC11] RUSU R. B., COUSINS S.: 3D is here: Point Cloud Library (PCL). In *Proc. IEEE ICRA* (2011). 7

[RKBG20] RAMASINGHE S., KHAN S. H., BARNES N., GOULD S.: Spectral-GANs for high-resolution 3D point-cloud generation. In *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)* (2020), pp. 8169–8176. 3

[RTG00] RUBNER Y., TOMASI C., GUIBAS L. J.: The Earth mover's distance as a metric for image retrieval. *Int. J. Comput. Vis. 40*, 2 (2000), 99–121. 6

[SGST20] SEGÙ M., GRINVALD M., SIEGWART R., TOMBARI F.: 3DSNet: Unsupervised shape-to-shape 3D style transfer. *CoRR abs/2011.13388* (2020). 1, 3, 5, 7

[SPK19] SHU D. W., PARK S. W., KWON J.: 3D point cloud generative adversarial network based on tree structured graph convolutions. In *Proc. IEEE ICCV* (2019), IEEE, pp. 3858–3867. 3

[SWL*20] SUN Y., WANG Y., LIU Z., SIEGEL J. E., SARMA S. E.: PointGrow: Autoregressively learned point cloud generation with self-attention. In *Proc. IEEE Winter Conf. on Applications of Computer Vision (WACV)* (2020), pp. 61–70. 3

[XKHK17] XU K., KIM V. G., HUANG Q., KALOGERAKIS E.: Data-driven shape analysis and processing. *Comput. Graph. Forum 36*, 1 (2017), 101–132. 1

[YCH*19] YIN K., CHEN Z., HUANG H., COHEN-OR D., ZHANG H.: LOGAN: unpaired shape transform in latent overcomplete space. *ACM Trans. Graph. 38*, 6 (2019), 198:1–198:13. 1, 3

[YGS*21] YIN K., GAO J., SHUGRINA M., KHAMIS S., FIDLER S.: 3DStyleNet: Creating 3D shapes with geometric and texture style variations. In *Proc. IEEE ICCV* (2021), pp. 12436–12445. 1

[YHCZ18] YIN K., HUANG H., COHEN-OR D., ZHANG H. R.: P2P-NET: bidirectional point displacement net for shape transform. *ACM Trans. Graph. 37*, 4 (2018), 152:1–152:13. 3

[YHH*19] YANG G., HUANG X., HAO Z., LIU M., BELONGIE S. J., HARIHARAN B.: PointFlow: 3D point cloud generation with continuous normalizing flows. In *Proc. IEEE ICCV* (2019), pp. 4540–4549. 2

[YZTG17] YI Z., ZHANG H., TAN P., GONG M.: DualGAN: Unsupervised dual learning for image-to-image translation. *CoRR abs/1704.02510* (2017). 4

[ZDW21] ZHOU L., DU Y., WU J.: 3D shape generation and completion through point-voxel diffusion. In *Proc. IEEE ICCV* (2021), pp. 5806–5815. 3

[ZKJB17] ZUFFI S., KANAZAWA A., JACOBS D. W., BLACK M. J.: 3D menagerie: Modeling the 3D shape and pose of animals. In *Proc. IEEE CVPR* (2017), pp. 5524–5532. 7

[ZPIE17] ZHU J., PARK T., ISOLA P., EFROS A. A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR abs/1703.10593* (2017). 1, 4, 5