Self-Supervised Generative-Contrastive Learning of Multi-Modal Euclidean Input for 3D Shape Latent Representations: A Dynamic Switching Approach

Chengzhi Wu[1], Julius Pfrommer[2], Mingyuan Zhou[1], and Jürgen Beyerer[1,2]

[1]Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology, Germany
[2]Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB, Germany

chengzhi.wu@kit.edu julius.pfrommer@iosb.fraunhofer.de
mingyuan.zhou@student.kit.edu juergen.beyerer@iosb.fraunhofer.de

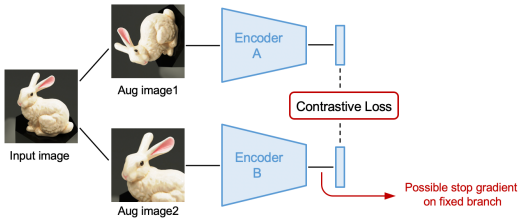
***Abstract*—We propose a combined generative and contrastive neural architecture for learning latent representations of 3D volumetric shapes. The architecture uses two encoder branches for voxel grids and multi-view images from the same underlying shape. The main idea is to combine a contrastive loss between the resulting latent representations with an additional reconstruction loss. That helps to avoid collapsing the latent representations as a trivial solution for minimizing the contrastive loss. A novel dynamic switching approach is used to cross-train two encoders with a shared decoder. The switching approach also enables the stop gradient operation on a random branch. Further classification experiments show that the latent representations learned with our self-supervised method integrate more useful information from the additional input data implicitly, thus leading to better reconstruction and classification performance.**

***Index Terms*—Self-supervised learning, contrastive learning, multi-modal input, 3D shapes, dynamic switching**

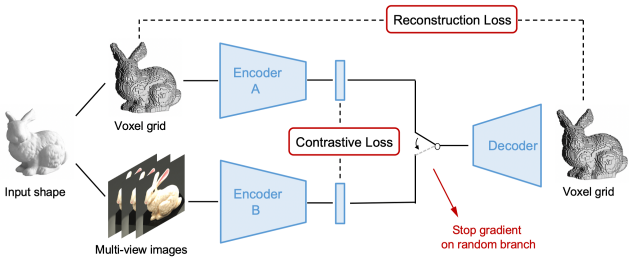
I. Introduction

3D shapes can be represented in a range of different formats. On the Euclidean side, they may be represented as RGB-D images, multi-view images or volumetric data. On the Non-Euclidean side, they may be represented as point clouds or meshes. For computer vision tasks like classification, segmentation, or even generative tasks like shape reconstruction, the target 3D shape is usually converted into a latent representation first. Before the rise of deep learning [17], popular latent representations (or, 3D shape descriptors) were Laplacian spectral eigenvectors [42], or heat kernel signatures [44]. With neural networks, the latent representation is usually the result of an encoder that reduces the 3D shape to a vector representation with fixed dimensionality.

When multi-modal input data is available, the question arises of how to use them jointly. For 3D learning tasks, take 3D Euclidean data as an example, most state-of-the-art methods in computer vision that deal with both image and voxel grid input data either concatenate individual latent representations for supervised tasks [23], or use only one of them on the input and loss side separately [12], [45], [47], [53], or use them jointly but with pre-training and finetuning [16]. We are interested in seeking a better self-supervised way for learning better latent representations for 3D volumetric shapes, with additional input from other modalities.



(a) General pipeline for contrastive learning from augmented data.



(b) Pipeline for the proposed generative-contrastive learning from multi-modal input.

Fig. 1: An illustration of (a) a general pipeline of contrastive learning methods and (b) our proposed generative contrastive learning pipeline for 3D shapes.

Apart from pretext tasks-based methods [15], [18], the other two main self-supervised learning ways are generative-based methods [2], [12] and contrastive-based methods [6], [19]. For 3D volumetric shapes, it is easy to implement a generative model. But it is still an open question of how to do it in a contrastive way, let alone the combination of these two. In a recent review paper of self-supervised learning [31], the authors argue that the only way of doing generative-contrastive learning is to train an encoder-decoder to generate synthetic samples and a discriminator to distinguish them from real samples. We disagree with this argument. In their definition, the discriminator is the contrastive part thus the model only focuses on negative pairs. We think it is also possible to use or

only use positive pairs, e.g. in our case, using multi-modalities from the same input shape for two branches.

Figure 1 shows the main idea of our proposed generative-contrastive learning pipeline. Compared to the existing contrastive learning methods, our method shares some similarities with them while some significant differences also exist. Similarities are: (i) we both use a two-branches scheme to encode two inputs that originate from the same "raw data"; (ii) after getting encoded latent representations, we both compute a contrastive loss in the latent space; (iii) they use positive pairs for training (optionally with additional negative pairs), we also use positive pairs. Differences are: (i) they use different augmented data from the "raw data", while we use different modalities from the "raw data"; (ii) thus our network architectures of encoder A and B are not identical, while theirs are identical mostly; (iii) we add a decoder part and a reconstruction loss; (iv) they possibly have stop gradient on one fixed branch, while we do stop gradient on random branch with a switching approach.

The main contributions of this paper are as follows:

- We propose a novel generative-contrastive learning pipeline for 3D volumetric shapes, which makes the joint training of encoders for multi-modal input data possible.
- With the switching approach doing the work of stopping gradient on random branch, model collapse is avoided. End-to-end training is also possible without the requirements of special pre-training.
- Using the voxel encoder as a self-supervised pre-trained feature extractor, we outperform 3D-GAN on the ModelNet40 classification task with much shorter latent vector representations (128, compared to ca. 2.5 million dimensions in 3D-GAN).
- The voxel encoder pre-trained on one single category still performs surprisingly well as a feature extractor on the full dataset with other categories during the testing.

II. Related Work

Contrastive learning: The work of contrastive learning was pioneered by Yann LeCun's group for face verification [11]. This topic has been getting more and more popular recently since people find self-supervised learning is important for feature extraction and we now have really mature deep learning techniques. SimCLR [6] proposes to use two identical encoders for two branches, both positive pairs and negative pairs are used. MoCo [8] stops the gradient for the second branch, while using a momentum-based method to update the parameters of its encoder. SwAV [3] proposes to use a memory bank to get negative pairs out of the batch, the contrastive loss in their case is computed after clustering. For methods that only use positive samples, BYOL [19] keeps the idea of momentum updating from MoCo, but adds an additional block in the first branch and only uses positive pairs. SimSiam [9] reports an observation of competitive results may still be achieved when modifying BYOL by making two encoders identical. A review of the most relevant methods and their comparisons are given in [13]. For 3D data, contrastive learning-based frameworks have been proposed mainly for the point cloud data representation, e.g. PointContrast [52] and Contrastive Scene Contexts [26]. More recently, several contrastive learning frameworks have been proposed for multi-modality input. Most of them focus on text-to-image learning [30], [54], [55], [57]. For 3D shapes, the closest work to ours is CrossPoint [1], which uses images and point clouds as input for better point cloud latent representation learning. However, it only uses the contrastive loss, no decoder or reconstruction loss is used. A more detailed comparison is given in subsection IV-D.

Learning on 3D shapes with Euclidean data: For supervised tasks, VoxNet [34] is the pioneer in using 3D convolutional network to learn features from volumetric data for recognition. Its subsequent work of multi-level 3D CNN [14] learns multi-scale spatial features by considering multiple resolutions of the voxel input. Qi et al. [37] propose to use multi-resolution filtering in 3D for multi-view CNNs, as well as using subvolume supervision for auxiliary training. FusionNet [23] fuses three networks together: two VoxNets [34] and one MVCNN [43]. The three networks fuse at the score layers where a linear combination of scores is taken before the classification prediction. A more recent work of Simple3D-Former [46] uses all image, voxel, and point cloud data as the input and co-trains the framework with multi-tasks in a supervised manner.

For self-supervised tasks, ShapeNet [48] uses a reverse VoxNet to reconstruct 3D volumetric shapes from latent representations that are learned from depth maps. The T-L network [16] combines a 3D autoencoder with an image regressor to encode a unified vector representation given a 2D image. Autoencoders have also been widely use for 3D shape retrieval in other papers [51], [56]. Its variant, VAE, has been used in a similar way for 3D shape learning [2]. View information from images has also been widely investigated for 3D shape reconstruction. Choy et al. [12] proposed a framework named 3D-R2N2 to reconstruct 3D shapes from multi-view images by leveraging the power of recurrent neural networks [25]. [39] also uses a recurrent-based approach, but taking depth images as input. Some other methods use view information as auxiliary constraints [20], [45], [53]. Method uses GAN for 3D volumetric shape generation has been proposed in [47]. Some other latest works [29], [35] have also used multi-modal input data for joint end-to-end training, but they did not use a switching approach for dynamic training.

We are aware that there are lots of other works applying deep learning-based methods on other 3D Non-Euclidean data formats, e.g. point clouds [1], [26], [32], [36], [38], [52]. However, it should be acknowledged that these methods have been primarily designed and optimized for point cloud data, and as such, their encoders are mostly not plug-and-play modules. Several critical intricacies, such as the sampling of point cloud patches, and the processing of the perspective-variant point cloud input, cannot be directly transferred to volumetric data. (An exception is CrossPoint [1], whose encoder is a plug-and-play module and we can easily replace it with a voxel encoder for volumetric data training.) Overall, they are out of the scope of this work, but we plan to include other 3D non-Euclidean data representations, e.g. point cloud, in our future work so that we could perform a fairer comparison with them directly.

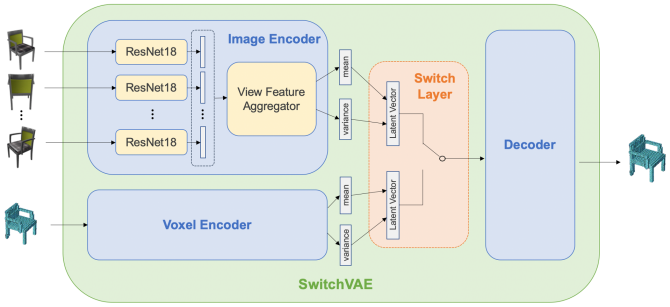


Fig. 2: The SwitchVAE architecture based on our proposed generative-contrastive learning pipeline.

III. Methodology

A. Generative-Contrastive Learning

Figure 1(b) shows the main idea of our proposed generative-contrastive learning pipeline. Similar to most existing contrastive learning methods, we use an architecture with two encoder branches to compute a contrastive loss between the latent representations from each branch. The inputs to the two branches are the voxel grid and the multi-view images of an identical 3D shape. A generative decoder part is added to compute the reconstruction loss. The decoder is shared by two encoders and the two encoder branches are co-trained with the help of a switching approach.

For contrastive learning methods, mode collapse is a big issue. Possible ways of dealing with it are adding additional blocks for encoder A, or stopping gradient for encoder B and updating its parameters in a momentum way slowly along with the updated parameters in encoder A. In our case, encoders A and B are already different network architectures thus the momentum method can not be applied, but we still managed to avoid model collapse successfully during the experiments. We attribute this success to two things: the reconstruction loss, and the switching approach. The reconstruction loss has a strong supervision over the representational capacity of latent representations, while the switching approach does the work of stopping the gradient on random branch.

To further improve the latent representations, Variational Autoencoders (VAE) [28] are used instead of vanilla autoencoders. In a VAE, each input is mapped to a multivariate normal distribution around a point in the latent space, which makes a continuous latent space. A continuous latent space makes the smooth transition of 3D shapes possible with latent representations. The learned features are usually more smooth and meaningful.

B. Switch Encoding

When dealing with multi-modal inputs, most state-of-the-art methods just encode them separately into latent representations and then perform concatenation. Unlike them, we propose to use a switching approach in the latent space to jointly train both encoders with a shared decoder. During the training, the switch is actuated for every training epoch with a preset probability to randomly select the encoded output from one encoder as the latent representation. This operation of switching between encoders continues during the whole end-to-end training.

The decoder is tasked to reconstruct the voxel representation of the 3D shape. Since the switched encoders are trained concurrently for the same decoder, they are forced to produce "mutually compatible" latent representations. The different input modalities result in different features that naturally emerge for the respective latent representation. For example, the voxel encoder tends to generate the latent feature of the full shape, while the image encoder can only generate the latent feature based on specific views, which may lose information due to occlusions. By cross-training with switched encoding, useful features for the latent representation can be translated from one encoder to the other via the shared decoder. This results in improved latent representations also for the individual encoder when just one input modality is used after the training.

C. Loss Functions

The SwitchVAE loss function consists of three parts: a reconstruction loss L_{recon}, a KL divergence L_{KL} between latent representations and the normal prior distribution, and a contrastive loss L_{contras} between the latent representations from the different input formats. The overall network is parameterized by $\boldsymbol{\theta} = (\boldsymbol{\theta}_{\text{vox}}, \boldsymbol{\theta}_{\text{img}}, \boldsymbol{\theta}_d)^\top$ for the voxel and image encoder and the voxel decoder respectively. The training samples are denoted $\boldsymbol{x}^\alpha$ for the input $\alpha \in \{\text{img}, \text{vox}\}$. The switch value for α is randomly selected prior to every training epoch. The latent representations resulting from the VAE encoders are $(\boldsymbol{\mu}^\alpha, \boldsymbol{\sigma}^\alpha) = e^\alpha(\boldsymbol{x}^\alpha)$. The latent representation is sampled for the current training epoch as $\boldsymbol{z}^\alpha \sim \mathcal{N}(\boldsymbol{\mu}^\alpha, \boldsymbol{\sigma}^\alpha)$. The decoder part is shared by both input modalities to reconstruct the voxel representation $\hat{\boldsymbol{x}}^\alpha = d(\boldsymbol{z}^\alpha)$. Formally, the overall loss function decomposes into three terms

$$\begin{aligned} L_{\boldsymbol{\theta}}(\alpha, \boldsymbol{x}^{\text{img}}, \boldsymbol{x}^{\text{vox}}) = \\ L_{\text{recon}}(\boldsymbol{x}^{\text{vox}}, \hat{\boldsymbol{x}}^\alpha) + \lambda_{\text{KL}} L_{\text{KL}}(\boldsymbol{\mu}^\alpha, \boldsymbol{\sigma}^\alpha) + \\ \lambda_{\text{contras}} L_{\text{contras}}(\boldsymbol{z}^{\text{img}}, \boldsymbol{z}^{\text{vox}}) \end{aligned} \tag{1}$$

with weights λ_{KL} and λ_{contras}.

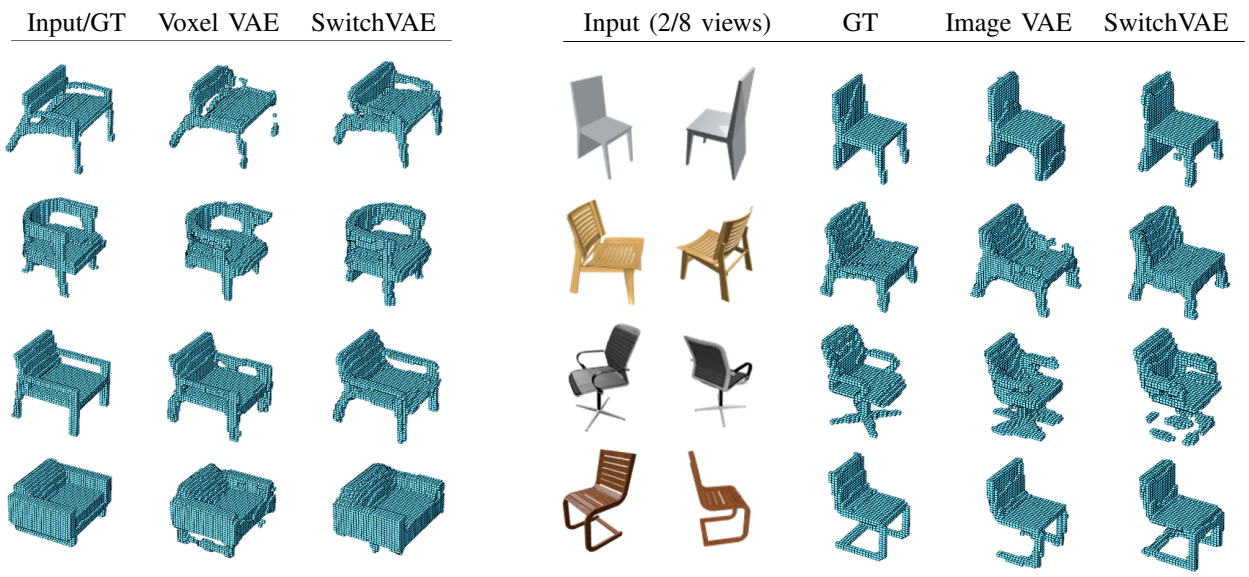
A modified Binary Cross Entropy (BCE) against the voxel ground truth is used for the reconstruction loss. To improve the training, modification has been made by the introduction of a hyper-parameter γ that weights the relative importance of false positives against false negatives. The reconstructed voxels are indexed by k with value $\hat{x}_k^\alpha \in [0, 1]$.

$$\begin{aligned} L_{\text{recon}}(\boldsymbol{x}^{\text{vox}}, \hat{\boldsymbol{x}}^\alpha) = \sum_k \Big[-\gamma \cdot x_k^{\text{vox}} \cdot \log(\hat{x}_k^\alpha) - \\ (1-\gamma)(1 - x_k^{\text{vox}}) \log(1 - \hat{x}_k^\alpha) \Big] \end{aligned} \tag{2}$$

We set the hyperparameter $\gamma = 0.8$ during training for all of the experiments conducted in Section IV.

In the training of VAE, the Kullback-Leibler (KL) divergence is used between the actual distribution of latent vectors and the $\mathcal{N}(\mathbf{0}, I)$ Gaussian distribution. Note that the latent representation has n dimensions.

$$L_{\text{KL}}(\boldsymbol{\mu}, \boldsymbol{\sigma}) = -\frac{1}{2} \sum_{i=1}^{n} (1 + \log(\sigma_i^2) - \mu_i^2 - \sigma_i^2) \tag{3}$$



(a) Reconstruction tests with voxel input.

(b) Reconstruction tests with image input.

Fig. 3: Some reconstruction results from different models with only voxel or multi-view images as the test input.

Test Input	Training Model	Reconstruction Metrics		
		IoU	Precision	Accuracy
Image	Image VAE	58.52%	68.21%	93.21%
	SwitchVAE ($\lambda_{\text{contras}} = 0$)	56.82%	67.72%	93.00%
	SwitchVAE ($\lambda_{\text{contras}} = 1$)	**58.75%**	**68.50%**	**93.32%**
Voxel	Voxel VAE	78.86%	82.80%	97.01%
	SwitchVAE ($\lambda_{\text{contras}} = 0$)	77.27%	80.04%	96.67%
	SwitchVAE ($\lambda_{\text{contras}} = 1$)	**79.93%**	**84.68%**	**97.22%**

TABLE I: Reconstruction performance on the test set. Training and testing with the chair category from 3D-R2N2 dataset.

In order to further force a close distance between the latent representations learned from image and volumetric data with the SwitchVAE model, a contrastive loss between the encoders is proposed and used in the latent space during the training phase. The contrastive loss is defined as the Euclidean distance between the latent vectors from images and volumetric data.

$$L_{\text{contras}}(\boldsymbol{z}^{\text{img}}, \boldsymbol{z}^{\text{vox}}) = \|\boldsymbol{z}^{\text{img}} - \boldsymbol{z}^{\text{vox}}\|_2^2 \tag{4}$$

Although in most other contrastive learning methods some different contrastive losses has been used, e.g. InfoNCE loss in SimCLR [6], we find that with latent representations normalized, our method can already yield satisfying results with a simple L_2 Norm loss as the contrastive loss.

IV. Experiments

We use the 3D-R2N2 and ModelNet 10/40 datasets for our experiments. The 3D-R2N2 dataset [12] is a subset with 13 categories from the ShapeNet dataset [4]. It provides good quality rendered multi-view images alongside a class label and $32 \times 32 \times 32$ voxel representations. We divide the 3D-R2N2 dataset into a training set of 29,599 samples and a test set of 7406 samples. The ModelNet dataset [48] comes in two variations with either 10 or 40 classes of shapes. The ModelNet10 dataset contains 3991/908 training/test samples. ModelNet40 contains 9843/2468 training/test samples.

For the SwitchVAE models, we use both a voxel and a multi-view image encoder. The decoder always reconstructs the voxel representation. During training for voxel test input, the switch layer randomly selects either the voxel encoder with a probability of 80%, or the multi-view image encoder with a probability of 20%.

Concerning the other training parameters, we use a latent dimension of 128 for all experiments. The network parameters are trained by minimizing the loss function from Equation 1 using the SGD optimizer with a momentum of 0.9 and Nesterov accelerated gradients [40]. The learning rate is 2×10^{-4} with a decay of 0.96 per 10 epochs after the first 50 epochs. The batch size is 32 for all experiments. Training with multi-view image input uses 8 views for every sample as it has been reported in [12] and its subsequent works [49], [50] that the improvement from additional views is negligible after the first 6-10 views.

A. Detailed Network Configuration

Figure 2 shows based on our proposed generative-contrastive learning pipeline, how switched encoding is implemented for a VAE with multi-view images and voxel grids input. The encoder blocks of our SwitchVAE build on the idea of volumetric convolutional networks [48] for the voxel input, and 3D recurrent reconstruction neural networks [12] for the

Training Dataset	Test Input	Training Method	Classification Accuracy	
			ModelNet40	ModelNet10
Chair Category	Multi-view images	Image VAE	75.28%	81.36%
		SwitchVAE	**77.07%**	**84.26%**
	Voxel data	Voxel VAE	80.19%	86.38%
		SwitchVAE	**80.60%**	**87.05%**
ModelNet40	Multi-view images	Image VAE	**85.06%**	88.62%
		SwitchVAE	83.87%	**89.96%**
	Voxel data	Voxel VAE	83.12%	87.95%
		SwitchVAE	**84.01%**	**90.07%**

TABLE II: Classification accuracy on the ModelNet40/ModelNet10 classification tasks with models trained with Image VAE, Voxel VAE, and SwitchVAE on the chair category or on the full ModelNet40 dataset.

Supervision	Method	Data Modality	Classification Accuracy	
			ModelNet40	ModelNet10
Supervised	3D ShapNets [48]	Voxels	77.30%	85.30%
	VoxNet [34]	Voxels	83.00%	92.00%
	MVCNN [43]	Images	90.10%	-
	FusionNets [23]	Images, Voxels	90.80%	93.11%
	3D2SeqViews [21]	Images	93.40%	94.71%
	VRN Ensemble [2]	Voxels	95.54%	97.14%
Self-supervised	LFD [5]	Images	75.50%	79.90%
	T-L Network [16]	Images, Voxels	74.40%	-
	VConv-DAE [41]	Voxels	75.50%	80.50%
	3D GAN [47]	Voxels	83.30%	**91.00%**
	CrossVoxel (modified from CrossPoint [1])	Images, Voxels	78.82%	86.34%
	SwitchVAE (trained on only chair category)	Images, Voxels	80.60%	87.05%
	SwitchVAE (trained on ModelNet40 dataset)	Images, Voxels	**84.01%**	90.07%

TABLE III: Classification accuracy on the ModelNet40/ModelNet10 dataset with different methods. Results from methods that only used images and/or voxels are listed. Note that our latent representations are only of 128 dimensions.

multi-view images input. More detailed network configurations are given as follows.

The image encoder of SwitchVAE learns the latent vector from multi-view images, and it is composed of a view feature embedding module and a view feature aggregator module. The view feature embedding module is a ResNet18 [22] whose weights are shared across all the views. The part with pre-trained weights maps a single view $137 \times 137 \times 3$ RGB image into $5 \times 5 \times 512$ feature maps. We then flatten these feature maps and add a fully connected layer, which outputs a 1024-dimensional feature for a single view image. For a 3D shape, 8 views of images are fed into the shared weights view feature embedding module while training, which outputs the 8×1024 view features.

For the view feature aggregator module, we first tried max pooling as MVCNN [43], but it did not yield satisfying results. Same for average pooling. To better aggregate the multi-view image features, we finally use the Gated Recurrent Unit (GRU) [10]. The view feature aggregator outputs a 1024-dimensional feature after aggregating features from all views. Then it is further fed into the last fully connected layers to generate the mean and the variance of the latent vector. By using the reparametrization trick introduced in [27], the image encoder finally outputs a 128-dimensional sampled latent vector.

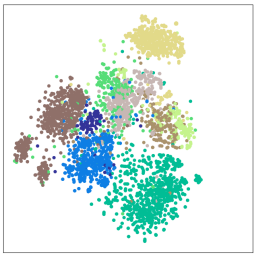
The voxel encoder is a 3D volumetric convolutional neural network. The encoder has 4 convolutional layers and two fully connected layers. All convolutional layers use kernels of size $3 \times 3 \times 3$, their strides are $\{1, 2, 1, 2\}$ and channel numbers are $\{8, 16, 32, 64\}$ respectively. All layers use the exponential linear unit (eLu) as the activation function except for the last fully connected layer. This layer maps a shape of $32 \times 32 \times 32$ voxels to a 343-dimensional feature. The 343-dimensional feature is further fed into the last fully connected layers to generate the mean and the variance of the latent vector to finally produce a 128-dimensional sampled latent vector.

The decoder of SwitchVAE mirrors the voxel encoder, except that the last layer uses a sigmoid activation function. The decoder maps a 128-dimensional latent vector, which was randomly sampled in the encoder, to a $32 \times 32 \times 32$ volumetric reconstruction. It represents the predicted voxel occupancy possibility of each voxel in the cube.

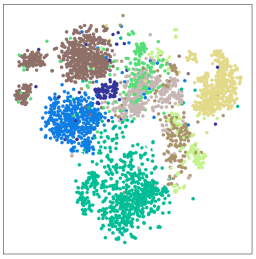
B. 3D Shape Reconstruction

We use Intersection-of-Union (IoU), precision, recall, and accuracy (referred to as average precision in [16]) as the quantitative metrics for the reconstruction of 3D shapes. The threshold at which a voxel is considered as filled is 50%. Similar to the last part, we show the results from only voxel input training, only image input training, and both input training with our SwitchVAE.

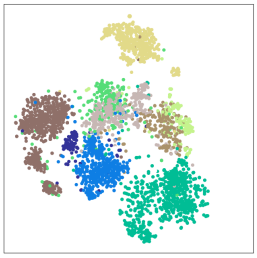
Table I shows that the reconstruction performance of SwitchVAE is similar or slightly better to that of image/voxel VAEs, and it focuses more on making every predicted occupied voxel correct (higher precision score). This characteristic may



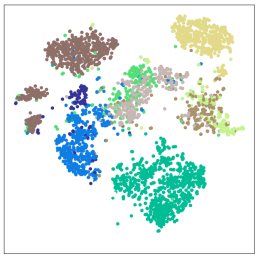
(a) Voxel VAE, trained on ModelNet10



(b) SwitchVAE ($\lambda_{\text{contras}} = 0$), trained on ModelNet10



(c) SwitchVAE ($\lambda_{\text{contras}} = 1$), trained on ModelNet10



(d) SwitchVAE ($\lambda_{\text{contras}} = 1$), trained on 3D-R2N2 dataset

Fig. 4: t-SNE plots of the latent representations for ModelNet10 shapes (10 categories) with (a) a vanilla voxel VAE model trained on ModelNet10 dataset. (b) a SwitchVAE model without contrastive loss trained on ModelNet10 dataset. (c) a SwitchVAE model with contrastive loss trained on ModelNet10 dataset. (d) a SwitchVAE model with contrastive loss trained on the full 3D-R2N2 dataset. Each color represents one category. All latent representations used for the plots use voxel data from the testing set as test input.

be more clearly observed in some reconstruction results. Table I also shows that the contrastive loss term is the key in our method. Figure 3 shows some qualitative reconstruction results from our SwitchVAE model that trained on the chair category. Comparisons with the results from the networks that only use one input format for training are also presented. From the figure we can observe that SwitchVAE takes more attention on not occupying the original negative voxels. This is quite obvious from the third row of Figure 3(b). Both Image VAE and SwitchVAE are not certain about the leg number of the office chair is 4 or 5. The Image VAE decides to merge them all together, while SwitchVAE decides to only guess and occupy some voxels with small sub-clusters in that area.

C. 3D Shape Classification

For the classification task, the networks are first trained to perform reconstruction of the ground-truth voxel representations. Then the encoder part of a trained network is used to produce latent representations of 128 dimensions as input for classification. An SVM with RBF kernel and hyper-parameter $\gamma = 1/128$ is trained on the latent representations to perform

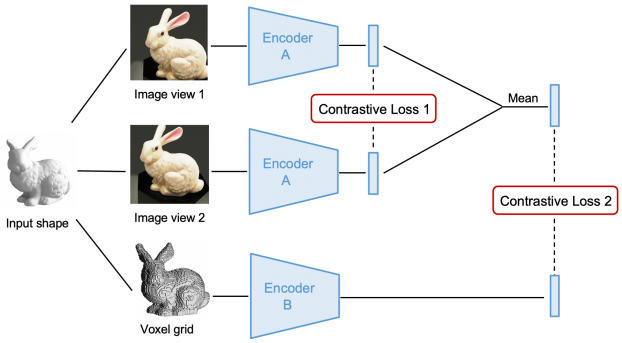


Fig. 5: The framework of CrossVoxel, which is modified from CrossPoint [1] for comparison expermiments.

classification. The same samples were used to train the networks for latent representations and the SVM. The evaluation of the SVM is performed with samples that were neither used to train the networks nor the SVM.

Table II shows the impact of switched training on the ModelNet 10/40 classification tasks. To make it more clear, let's take the voxel data tests as an example. During the training phase, a voxel VAE only trains on the voxel train set data, while a SwitchVAE trains on both the voxel train set data and the correspondent multi-view images train set data. During the testing phase, only the identical voxel test set data is given to the trained voxel VAE model and the trained SwitchVAE model. Latent representations of those 3D shapes (from the voxel test set) obtained from the SwitchVAE model always outperform that from the vanilla voxel VAE in the ModelNet classification tasks. Note that no multi-view image data of the test set is needed for SwitchVAE during the testing. From Table II, we can clearly observe that under the condition of the same training dataset and test input, the results from SwitchVAE are better than the results from the image VAE or the voxel VAE in most cases. Note that they are even trained with a same number of epochs. This means by using the data of other formats in the training phase, during the testing phase, the classification performance has been improved compared to the models that only use a single format for the training.

Table III lists the classification result in comparison to other network architectures. Note that there are not many papers on image-voxel multi-modal methods for 3D shapes. Hence, we have included some other methods that used images or voxels solely as input for additional comparison. Compared to most other unsupervised learning method, we achieve better classification performance. Compared to 3D-GAN, our method outperforms it on the ModelNet40 classification task and achieves competitive performance on the ModelNet10 classification task. However, our method uses a much smaller latent vector of only 128 dimensions. 3D-GANs use all feature maps in the last three convolution layers, which makes the presentation for each 3D shape a 2.5 million dimensional vector as input for the classification.

t-SNE Visualization: In order to visualize the learned latent representations, we use t-SNE [33] to map the latent representations to a 2D plane. Figure 4 gives the visualization

Method	Pre-train (unsupervised)	Fine-tune (supervised)	Classification Accuracy	
			ModelNet40	ModelNet10
Fully supervised	×	✓	89.74%	92.32%
SwithVAE	✓	×	84.01%	90.07%
SwithVAE (fine-tuned)	✓	✓	**91.25%**	**93.40%**

TABLE IV: Experimental results of further fine-tuning the pre-trained voxel encoder with a simple classification head in a supervised manner.

Switch Probability (image : voxel)	Classification Accuracy	
	ModelNet40	ModelNet10
0 : 10	83.12%	87.95%
1 : 9	83.47%	89.22%
2 : 8	**84.01%**	**90.07%**
3 : 7	83.92%	89.84%
4 : 6	83.21%	88.19%
5 : 5	82.63%	86.70%

TABLE V: Ablation study on switch probability. Using voxel data as the test data.

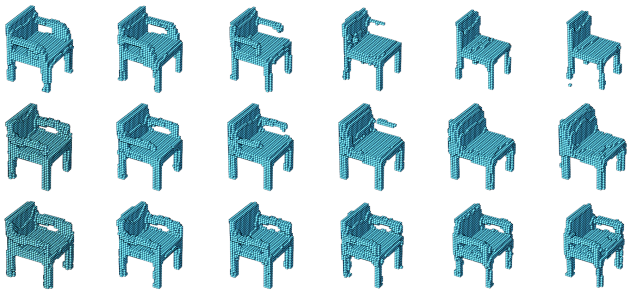
results. we use ModelNet10 for most t-SNE visualization experiments. Comparing Figure 4(a) and Figure 4(b), we can observe that the switching approach contributes to the inter-category classification while making the intra-category clustering a bit fuzzy (all categories are a bit far from each other, while each category itself is a bit less clustered). Comparing Figure 4(b) and Figure 4(c), we can observe that adding the contrastive loss term to SwitchVAE helps the intra-category clustering, making the performance of the whole classification task much better. Comparing Figure 4(c) and Figure 4(d), we can observe that with a larger training dataset, even better feature clustering results may be achieved. The increased gaps between different categories can be clearly observed.

Fine-tuned Classification: We additionally report the result of pre-train the encoder in a self-supervised manner with our SwitchVAE, then fine-tune it supervised with a simple classification head. The numerical results are reported in Table IV. For a better comparison, the result of the encoder directly trained supervised with the classification head is also reported. From the table, we can see that the voxel encoder can further improve its performance when pre-trained with our method before the training of supervised learning.

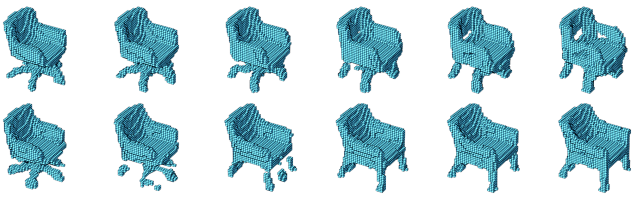
D. Comparison to CrossPoint

The closest work to ours is CrossPoint [1], which uses images and point clouds as input for better point cloud latent representation learning. Apart from the different input data representations, our method differs from CrossPoint in three perspectives: (i) we use an additional reconstruction loss with the help of a decoder; (ii) we use VAE, other than AE, which introduces variance during the training; (iii) most importantly, we use a switching approach to enable the dynamic training of the framework.

On the other hand, the encoder in CrossPoint is a plug-and-play module that can be easily replaced. We thus have replaced the point cloud encoder with a voxel encoder to conduct comparison experiments. The modified framework is illustrated



(a) Top Row: Trained on SwitchVAE, the "chair arm" feature and the "size" feature are entangled. Middle and Bottom Row: Trained on Switch-BetaTCVAE with $\beta = 5$, the "chair arm" feature and the "size" feature are more disentangled, changing one feature does not impact the other one too much.



(b) Top Row: Trained on SwitchVAE, changing the "chair leg type" feature also leads to the morphing of the top part. Bottom Row: Trained on Switch-BetaTCVAE with $\beta = 5$, the top part stays more fixed while changing the "chair leg type" feature.

Fig. 6: Disentangling latent features with Beta-TCVAE.

in Figure 5. Note that in the original framework of CrossPoint, it uses multi-point clouds and single-image as the input, while in our case, we use multi-images and single-volumetric data as the input. Hence for a fair comparison, the intra-modal contrastive loss is computed with image data in the modified framework. The numerical result is reported in Table III under the method "CrossVoxel". We can observe that it achieves remarkable results, but not on par with ours. This is probably due to our framework (i) introduces another reconstruction loss with a decoder; (ii) allows latent representation variance using VAE; and (iii) uses more views of image during the training.

E. Ablation Study on Switch Probability

One important parameter in our experiments is the switch probability, which decides the actual updating step ratio between two encoders. For a certain target data representation, e.g. the volumetric data, in order to improve the final performance of the voxel encoder for downstream tasks, the parameters of the voxel encoder should be updated more often. This means we should set the "switched probability" to the voxel encoder larger, compared to that of the image encoder. However, on the other hand, if the model focuses too much on the training of the voxel encoder and ignores or only trains slightly on the image encoder, it can hardly make use of the information from the image input. A trade-off between these two perspectives must be carefully made. Hence, an ablation study regarding the switch probability has been conducted and the numerical results are given in Table V. We train a same SwitchVAE model but with different switch probabilities on the ModelNet40 dataset, and test it with the voxel data using the same classification

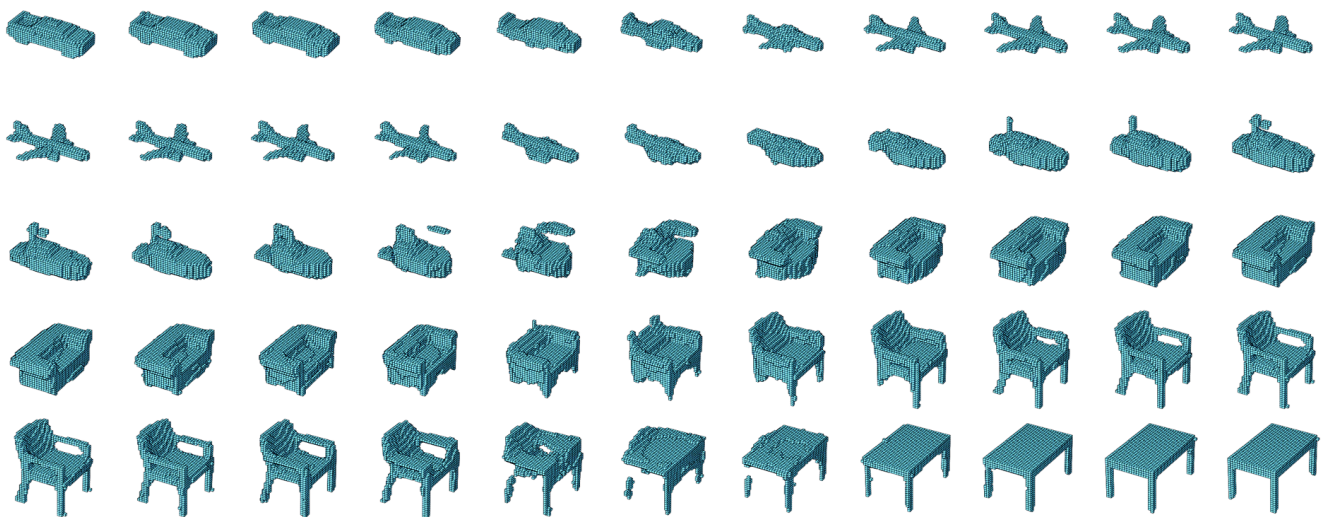
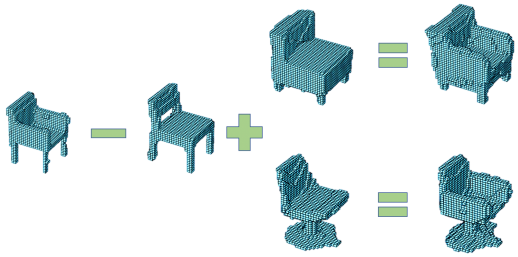
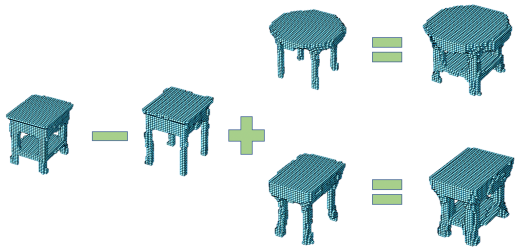


Fig. 7: Shape interpolation between different categories.



(a) Shape arithmetic example with chair objects.



(b) Shape arithmetic example with table objects.

Fig. 8: Shape arithmetic for chairs and tables. (a) Adding chair arms to chair objects in the latent space. (b) Adding a middle layer to table objects in the latent space.

benchmark in subsection IV-C. From it, we can see that a decent choice is setting the "switched probability" for the voxel encoder as 80%, while 20% for the image encoder.

F. Exploring Latent Representations

This subsection showcases some qualitative results to give an indication that SwitchVAE training results in a superior latent representation that allows for better disentanglement between categories, as well as between the salient features of the 3D shapes in each category.

Latent space interpolation: Similar to most 3D reconstruction papers, we also do the inter-class interpolation with our trained models as shown in Figure 7. It can be observed that our proposed method has the ability to perform a smooth transition between two shapes, even if they are from different categories.

Shape arithmetic: Another way to explore the learned latent representations is to perform arithmetic operations in the latent space whilst observing their effect on the reconstructed geometry. We show some shape arithmetic results in Figure 8 with a model trained on the full 3D-R2N2 dataset. The model seems to capture the underline information and is capable of generating meaningful combined shapes that do not occur as 3D shapes in the original dataset.

Feature disentanglement with VAE variations: One good thing with VAE models is that the latent space learned from it is more "meaningful" compared to that from GAN models. By tuning the value in one specific latent dimension, one can observe certain features on the output side changing smoothly. However, most features get entangled in multiple latent dimensions with the vanilla VAE. It has been reported that β-VAE [24] and β-TCVAE [7] can produce better disentangled features in the latent space. We merge it with our proposed method into SwitchBTCVAE. We train our model with $\beta = 5$ on the chair category with a same number of epochs as the other experiments. Although a small decrease in the reconstruction performance metrics is observed, by investigating the learned latent representations, we find that some features have been better disentangled as shown in Figure 6.

V. Conclusion and Outlook

In this paper, we propose a generative-contrastive learning pipeline for learning better latent representations for 3D volumetric shapes, with the help of additional modality input. The switching approach makes the joint training for both encoders possible with competitive reconstruction results. Classification experiments on ModelNet have also been carried out to validate the effectiveness of the proposed method. Improved classification results indicate that better latent representations have been learned with our proposed SwitchVAE architecture.

For future directions, other 3D data modalities, e.g., point clouds and meshes may also be used. A new contrastive loss may be designed and an optimal switching policy may be studied. More research may be conducted to make the latent presentations more feature-disentangled or more interpretable. In our experiments, although an additional contrastive loss has been applied, we still observe large distances between latent representations generated from the different input formats. A thorough study of how to force a closer distance between representations from different formats without leading to a collapse due to increased contrastive loss could provide deeper insights into the fundamental principles of contrastive learning.

REFERENCES

[1] Mohamed Afham, Isuru Dissanayake, Dinithi Dissanayake, Amaya Dharmasiri, Kanchana Thilakarathna, and Ranga Rodrigo. Crosspoint: Self-supervised cross-modal contrastive learning for 3d point cloud understanding. *CVPR*, pages 9892–9902, 2022.

[2] A. Brock, T. Lim, J. M. Ritchie, and N. Weston. Generative and discriminative voxel modeling with convolutional neural networks. *NeurIPS*, 2016.

[3] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *NeurIPS*, 2020.

[4] 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. *ArXiv*, abs/1512.03012, 2015.

[5] D. Chen, X. Tian, E. Shen, and M. Ouhyoung. On visual similarity based 3d model retrieval. *Computer Graphics Forum*, 22, 2003.

[6] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton. A simple framework for contrastive learning of visual representations. *ICML*, 2020.

[7] T. Q. Chen, X. Li, R. B. Grosse, and D. Duvenaud. Isolating sources of disentanglement in variational autoencoders. *NeurIPS*, 2018.

[8] X. Chen, H. Fan, R. B. Girshick, and K. He. Improved baselines with momentum contrastive learning. *CVPR*, 2020.

[9] X. Chen and K. He. Exploring simple siamese representation learning. *CVPR*, 2021.

[10] K. Cho, B. V. Merrienboer, C. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *ArXiv*, abs/1406.1078, 2014.

[11] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. *CVPR*, pages 539–546, 2005.

[12] C. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. *ECCV*, 2016.

[13] L. Ericsson, H. Gouk, and T. M. Hospedales. How well do self-supervised models transfer? *CVPR*, 2021.

[14] S. Ghadai, X. Lee, A. Balu, S. Sarkar, and A. Krishnamurthy. Multi-level 3d cnn for learning multi-scale spatial features. *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1152–1156, 2019.

[15] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. *ICLR*, 2018.

[16] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta. Learning a predictable and generative vector representation for objects. *ECCV*, 2016.

[17] I. J. Goodfellow, Y. Bengio, and A. C. Courville. Deep learning. *Nature*, 521:436–444, 2015.

[18] P. Goyal, D. Mahajan, A. Gupta, and I. Misra. Scaling and benchmarking self-supervised visual representation learning. *ICCV*, 2019.

[19] J. Grill, F. Strub, F. Altch'e, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko. Bootstrap your own latent: A new approach to self-supervised learning. *NeurIPS*, 2020.

[20] J. Gwak, C. B. Choy, M. Chandraker, A. Garg, and S. Savarese. Weakly supervised 3d reconstruction with adversarial constraint. *2017 International Conference on 3D Vision (3DV)*, pages 263–272, 2017.

[21] Z. Han, H. Lu, Z. Liu, C. Vong, Y. Liu, M. Zwicker, J. Han, and C. Chen. 3d2seqviews: Aggregating sequential views for 3d global feature learning by cnn with hierarchical attention aggregation. *IEEE Transactions on Image Processing*, 28:3986–3999, 2019.

[22] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CVPR*, pages 770–778, 2016.

[23] V. Hegde and R. Zadeh. Fusionnet: 3d object classification using multiple data representations. *ArXiv*, abs/1607.05695, 2016.

[24] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

[25] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 1997.

[26] Ji Hou, Benjamin Graham, Matthias Nießner, and Saining Xie. Exploring data-efficient 3d scene understanding with contrastive scene contexts. *CVPR*, pages 15582–15592, 2020.

[27] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2014.

[28] D. P. Kingma and M. Welling. An introduction to variational autoencoders. *Foundations and Trends in Machine Learning*, 12:307–392, 2019.

[29] R. Klokov, J. Verbeek, and E. Boyer. Probabilistic reconstruction networks for 3d shape inference from a single image. *BMVC*, 2019.

[30] Wei Li, Can Gao, Guocheng Niu, Xinyan Xiao, Hao Liu, Jiachen Liu, Hua Wu, and Haifeng Wang. Unimo: Towards unified-modal understanding and generation via cross-modal contrastive learning. *ArXiv*, abs/2012.15409, 2020.

[31] X. Liu, F. Zhang, Z. Hou, Z. Wang, L. Mian, J. Zhang, and J. Tang. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[32] Yueh-Cheng Liu, Yu-Kai Huang, HungYueh Chiang, Hung-Ting Su, Zhe-Yu Liu, Chin-Tang Chen, Ching-Yu Tseng, and Winston H. Hsu. Learning from 2d: Pixel-to-point knowledge transfer for 3d pretraining. *ArXiv*, abs/2104.04687, 2021.

[33] L. V. D. Maaten and G. E. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[34] D. Maturana and S. Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. *IROS*, pages 922–928, 2015.

[35] S. Muralikrishnan, V. G. Kim, M. Fisher, and S. Chaudhuri. Shape unicode: A unified shape representation. *CVPR*, pages 3785–3794, 2019.

[36] Yatian Pang, Wenxiao Wang, Francis E. H. Tay, W. Liu, Yonghong Tian, and Liuliang Yuan. Masked autoencoders for point cloud self-supervised learning. In *European Conference on Computer Vision*, 2022.

[37] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. Guibas. Volumetric and multi-view cnns for object classification on 3d data. *CVPR*, pages 5648–5656, 2016.

[38] Guocheng Qian, Xingdi Zhang, Abdullah Hamdi, and Bernard Ghanem. Pix4point: Image pretrained transformers for 3d point cloud understanding. *ArXiv*, abs/2208.12259, 2022.

[39] D. J. Rezende, S. Eslami, S. Mohamed, P. Battaglia, M. Jaderberg, and N. Heess. Unsupervised learning of 3d structure from images. *NeurIPS*, 2016.

[40] S. Ruder. An overview of gradient descent optimization algorithms. *ArXiv*, abs/1609.04747, 2016.

[41] A. Sharma, O. Grau, and M. Fritz. Vconv-dae: Deep volumetric shape learning without object labels. *ArXiv*, abs/1604.03755, 2016.

[42] O. Sorkine-Hornung. Laplacian mesh processing. *Eurographics*, 2005.

[43] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. *ICCV*, pages 945–953, 2015.

[44] J. Sun, M. Ovsjanikov, and L. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. *Computer Graphics Forum*, 28, 2009.

[45] S. Tulsiani, A. A. Efros, and J. Malik. Multi-view consistency as supervisory signal for learning shape and pose prediction. *CVPR*, pages 2897–2905, 2018.

[46] Yi Wang, Zhiwen Fan, Tianlong Chen, Hehe Fan, and Zhangyang Wang. Can we solve 3d vision tasks starting from a 2d vision transformer? *ArXiv*, abs/2209.07026, 2022.

[47] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *ArXiv*, abs/1610.07584, 2016.

[48] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. *CVPR*, pages 1912–1920, 2015.

[49] H. Xie, H. Yao, X. Sun, S. Zhou, S. Zhang, and X. Tong. Pix2vox: Context-aware 3d reconstruction from single and multi-view images. *ICCV*, pages 2690–2698, 2019.

[50] H. Xie, H. Yao, S. Zhang, S. Zhou, and W. Sun. Pix2vox++: Multi-scale context-aware 3d object reconstruction from single and multiple images. *International Journal of Computer Vision*, pages 1 – 17, 2020.

[51] J. Xie, Yi Fang, F. Zhu, and E. K. Wong. Deepshape: Deep learned shape descriptor for 3d shape matching and retrieval. *CVPR*, pages 1275–1283, 2015.

[52] Saining Xie, Jiatao Gu, Demi Guo, C. Qi, Leonidas J. Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. *ArXiv*, abs/2007.10985, 2020.

[53] X. Yan, J. Yang, E. Yumer, Y. Guo, and H. Lee. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. *NeurIPS*, 2016.
[54] Rong Ye, Mingxuan Wang, and Lei Li. Cross-modal contrastive learning for speech translation. *ArXiv*, abs/2205.02444, 2022.
[55] Han Zhang, Jing Yu Koh, Jason Baldridge, Honglak Lee, and Yinfei Yang. Cross-modal contrastive learning for text-to-image generation. *CVPR*, pages 833–842, 2021.
[56] Z. Zhu, X. Wang, S. Bai, C. Yao, and X. Bai. Deep learning representation using autoencoder for 3d shape retrieval. *IEEE International Conference on Security, Pattern Analysis, and Cybernetics*, pages 279–284, 2014.
[57] Mohammadreza Zolfaghari, Yi Zhu, Peter Gehler, and Thomas Brox. Crossclr: Cross-modal contrastive learning for multi-modal video representations. *ICCV*, pages 1430–1439, 2021.

Chengzhi Wu Chengzhi Wu received both B.E. and M.S. degrees from Nanjing University, Nanjing, China. He is currently pursuing a Ph.D. degree in the Department of Computer Science at Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany. His research interests include machine learning and optimization algorithms, robust control of dynamic systems, deep learning, computer vision for remanufacturing, and 3d data analysis.

Julius Pfrommer Dr.-Ing. Julius Pfrommer leads the Department for Cognitive Industrial Systems at Fraunhofer IOSB. He holds engineering degrees from Karlsruhe Institute of Technology (KIT) and the Institut National Polytechnique in Grenoble. His PhD in computer science (summa cum laude) was awarded at KIT. His research interests include distributed systems, planning under uncertainty, and the use of machine learning for modeling, optimization and control in cyber-physical systems.

Mingyuan Zhou Mingyuan Zhou received the B.E. degree from the Beijing Jiaotong University, Beijing, China, and the M.S. degree from the Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany. He is currently pursuing the Ph.D. degree with the Internet of Things thrust in The Hong Kong University of Science and Technology (Guangzhou), China. His research interests include deep learning, mechanical engineering, and structural engineering.

Jürgen Beyerer Prof. Dr.-Ing. Jürgen Beyerer has been a full professor for informatics at the Institute for Anthropomatics and Robotics at the Karlsruhe Institute of Technology KIT since March 2004 and director of the Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB in Ettlingen, Karlsruhe, Ilmenau, Görlitz, Lemgo, Oberkochen and Rostock. Research interests include automated visual inspection, signal and image processing, variable image acquisition and processing, active vision, metrology, information theory, fusion of data and information from heterogeneous sources, system theory, autonomous systems and automation.