Learning a Hierarchical Latent-Variable Model of Voxelized 3D Shapes

Shikun Liu  
Pennsylvania State University  
sol5542@psu.edu

Alexander G. Ororbia II  
Pennsylvania State University  
ago109@psu.edu

C. Lee Giles  
Pennsylvania State University  
giles@ist.psu.edu

Abstract

We propose the Variational Shape Learner (VSL), a hierarchical latent-variable model for 3D shape learning. VSL employs an unsupervised, variational approach to the inference and learning of the underlying structure of voxelized 3D shapes. Our model successfully learns 3D shapes via a hierarchical latent representation, made possible through the use of skip-connections. Realistic 3D objects can be generated by sampling its latent probabilistic manifold. We show that our inference and generative model can be trained end-to-end from 2D images to perform single image 3D model retrieval. Experiments show the improved performance of our model both quantitatively and qualitatively over a range of tasks.

1 Introduction

Over the past several years, impressive strides have been made in the study of 3D objects. Much of this progress can be attributed to recent advances in neural network research. With respect to learning deep representations of voxelized 3D shapes, notable attempts include the use of deep belief networks (Wu et al., 2015), auto-encoders (Zhu et al., 2016; Girdhar et al., 2016; Rezende et al., 2016), and 3D convolutional networks (Maturana and Scherer, 2015; Yan et al., 2016; Sedaghat et al., 2016; Choy et al., 2016). Furthermore, this promising line of work has also introduced several large 3D CAD model benchmarks, such as ModelNet (Wu et al., 2015) and ShapeNet (Chang et al., 2015).

However, despite the progress made so far, previous state-of-the-art methods have only focused on learning 3D shapes under a single unified representation. For example, the T-L Network from (Girdhar et al., 2016) used an auto-encoder-like structure to learn shapes in a single vector representation. On the other hand, though the 3D-GAN (Wu et al., 2016) was shown to disentangle generative and discriminative factors in its learned representation on different tasks, it still learned a single probabilistic latent representation of 3D shapes. Furthermore, other methods, such as (Kar et al., 2015), require multiple shape instances such as camera viewpoints, shape keypoints, and segmentations. Inspired by (Edwards and Storkey, 2016), we argue and show that an explicit hierarchical design of a generative model’s latent structure will improve performance across many 3D shape tasks.

Further motivated by recent advances in variational generative modeling using neural architectures (Kingma and Welling, 2013), we propose the Variational Shape Learner, a hierarchical latent-variable model capable of learning useful and rich latent representations of 3D shapes.

The main contributions of this paper are summarized as follows:

- We propose a novel latent-variable model that we call the Variational Shape Learner which is able to learn expressive features of 3D shapes.
- Our model is fully unsupervised, requiring no segmentation, keypoints, or pose information for both general 3D model learning and single image reconstruction.
• Our learned latent features outperform the current state-of-the-art methods in unsupervised model classification while significantly reducing the number of features required.

• Our extensive experimental analysis shows that our real-world image reconstruction surpasses state-of-the-art in 8 of 10 classes. Half of which we surpass by a large margin.

2 Related Work

Over the past several decades, 3D object recognition has been an often-studied problem in the computer vision literature. Originally, previous work (Patterson IV et al., 2008; Knopp et al., 2010; Rusu et al., 2009) made use of hand-crafted shape descriptors combined with other favorable methods derived from image classification tasks.

However, since the ImageNet contest of 2012 (Krizhevsky et al., 2012), deep convolutional networks (ConvNets) (Fukushima, 1988; LeCun et al., 1989) have swept the vision industry, becoming nearly ubiquitous in countless applications. The most relevant work to this paper is that of (Wu et al., 2015), which presented promising results and a useful benchmark for 3D model recognition: ModelNet. Following this work, researchers have used 3D ConvNets (Maturana and Scherer, 2015; Choy et al., 2016; Su et al., 2015; Yan et al., 2016), auto-encoders (Xie et al., 2015; Zhu et al., 2016; Girdhar et al., 2016; Rezende et al., 2016), and a variety of generative models (Wu et al., 2016; Rezende et al., 2016), where variants have been designed to progressively improve state-of-the-art results.

Generative modeling has also been a widely studied topic. Generative Adversarial Networks (GANs), proposed in (Goodfellow et al., 2014) and Variational auto-encoders (VAEs), proposed in (Kingma and Welling, 2013), are some of the most popular and important frameworks in modern neural network literature. Successful adaptation of these frameworks range from a focus in natural language and speech processing (Chung et al., 2015; Serban et al., 2016) to realistic image synthesis (Gregor et al., 2015; Radford et al., 2015; Pu et al., 2016), yielding promising, positive results. Nevertheless, little work (Wu et al., 2016; Girdhar et al., 2016; Rezende et al., 2016) has focused on modeling 3D objects, where generative models are used to learn probabilistic embeddings of 3D structures.

3 The Variational Shape Learner

In this section, we introduce our proposed Variational Shape Learner (VSL), building upon the generative modeling framework of (Kingma and Welling, 2013), the Neural Statistician (Edwards and Storkey, 2016), and the volumetric convolutional network (Maturana and Scherer, 2015).

Figure 1: The network structure of the Variational Shape Learner. Solid lines represent synaptic connections either in fully-connected or convolutional layers, dashed lines represent concatenation, while the dotted-dashed lines represent possible applications. ○ means latent features, □ means concatenated features, and ◦ means equivalence relation.
3.1 Encoder: 3D-ConvNet + Skip-Connections

We use the volumetric 3D convolutional neural network of (Maturana and Scherer, 2015) to encode voxelized 3D shapes $x$ to both global and local features $z_{0:n}$. We use three fully convolutional layers with kernel size $\{6, 5, 4\}$, strides $\{2, 2, 1\}$ and channels $\{32, 64, 128\}$ respectively. The last layer of the network is flattened and followed by two fully-connected layers (256 neurons in the 1st, 100 neurons in the 2nd). For each layer, we use the rectified linear unit (ReLU) as the activation function.

Each local latent code $z_{i:0}$ is approximated by the global latent code, the input voxel $x$, and the previous latent code (except for $z_1$, which does not have a previous latent code) using two fully-connected layers with 100 neurons each. These skip-connections help to ease the process of learning hierarchical features and force each local latent code to learn one level of abstraction.

The approximate posterior for one single voxel is then given by,

$$ q(z_{0:n}|x; \phi) = q(z_0|x; \phi)q(z_1|z_0, x; \phi)\prod_{i=2}^{n} q(z_i|z_{i-1}, z_0, x; \phi) $$

where $\phi$ contains the variational parameters parametrized by neural networks. $n$ represents the number of local latent codes.

3.2 Image Regressor: 2D-ConvNet

We use a standard 2D convolutional network to encode input RGB images into a feature space with the same dimension as the sum of global and local latent codes. The network contains four fully-convolutional layers with kernel sizes $\{32, 15, 5, 3\}$, strides $\{2, 2, 2, 1\}$, and channels $\{16, 32, 64, 128\}$. The last convolutional layer is flattened and fed into two fully-connected layers with 200 and 100 neurons each. Unlike the encoder described in Section 3.1, we apply dropout (Srivastava et al., 2014) before the last fully-connected layer.

3.3 Decoder: 3D-DeConvNet

After we learn the global and local latent codes $z_{0:n}$, we then concatenate them into a single vector as shown in Figure 1 in blue dashed lines.

A 3D deconvolutional neural network with dimensions symmetrical to the encoder of Section 3.1 is used to decode the learned latent features into a voxel. An element-wise logistic sigmoid is applied to the output layer in order to convert the learned features to occupancy probabilities for each voxel cell.

3.4 Loss: Variational Loss + Latent Loss

The Variational Shape Learner is a latent-variable model that falls under the family of variational generative models. The VSL’s learning objective contains a standard reconstruction loss $\mathcal{L}_{rec}$, as well as a regularization penalty $\mathcal{L}_{reg}$. Furthermore, its loss contains a term for the latent variables $\mathcal{L}_{lat}$, which is relevant for the 3D model retrieval task of Section 4.5. This term is a simple $\mathcal{L}_2$ penalty imposed on the difference between the learned features of the image regressor $z'$ and true latent features $z = [z_{0:n}]$.

We assume a fixed, spherical unit Gaussian prior, $p(z_0) = \mathcal{N}(0, I)$. The conditional distribution over each local latent code is defined as follows:

$$ p(z_i|z_{i-1}; \theta) = \mathcal{N}(\mu(z_{i-1}), \sigma^2(z_{i-1})) $$

where $p(z_0|z_{i-1}; \theta)$ and $p(z_i|z_{i-1}, z_0; \theta)$ are also spherical Gaussians and $\theta$ contains the generative parameters parametrized by neural networks.

Let $p(x|z_{0:n}; \theta)$ be a Bernoulli (in case of binary voxel data) whose parameters are computed from $z_{0:n}$ with neural networks. The probability for one single voxel can then be calculated by,

$$ p(x) = \int p(x|z_{0:n}; \theta)p(z_1|z_0; \theta)p(z_0)\prod_{i=2}^{n} p(z_i|z_{i-1}, z_0; \theta) dz_{0:n}. $$

Let the reconstructed voxel $\hat{x}$ be directly parametrized by occupancy probability. The loss $\mathcal{L}(x)$ for input voxel $x$ of the Variational Shape Learner can be computed by:

$$ \mathcal{L}(x) = \mathcal{L}_{rec} + \delta \mathcal{L}_{reg} + \gamma \mathcal{L}_{lat}, $$

where $\delta$ and $\gamma$ are weighting factors.

3.4 Loss: Variational Loss + Latent Loss

The Variational Shape Learner is a latent-variable model that falls under the family of variational generative models. The VSL’s learning objective contains a standard reconstruction loss $\mathcal{L}_{rec}$, as well as a regularization penalty $\mathcal{L}_{reg}$. Furthermore, its loss contains a term for the latent variables $\mathcal{L}_{lat}$, which is relevant for the 3D model retrieval task of Section 4.5. This term is a simple $\mathcal{L}_2$ penalty imposed on the difference between the learned features of the image regressor $z'$ and true latent features $z = [z_{0:n}]$.

We assume a fixed, spherical unit Gaussian prior, $p(z_0) = \mathcal{N}(0, I)$. The conditional distribution over each local latent code is defined as follows:

$$ p(z_i|z_{i-1}; \theta) = \mathcal{N}(\mu(z_{i-1}), \sigma^2(z_{i-1})) $$

where $p(z_0|z_{i-1}; \theta)$ and $p(z_i|z_{i-1}, z_0; \theta)$ are also spherical Gaussians and $\theta$ contains the generative parameters parametrized by neural networks.

Let $p(x|z_{0:n}; \theta)$ be a Bernoulli (in case of binary voxel data) whose parameters are computed from $z_{0:n}$ with neural networks. The probability for one single voxel can then be calculated by,

$$ p(x) = \int p(x|z_{0:n}; \theta)p(z_1|z_0; \theta)p(z_0)\prod_{i=2}^{n} p(z_i|z_{i-1}, z_0; \theta) dz_{0:n}. $$

Let the reconstructed voxel $\hat{x}$ be directly parametrized by occupancy probability. The loss $\mathcal{L}(x)$ for input voxel $x$ of the Variational Shape Learner can be computed by:

$$ \mathcal{L}(x) = \mathcal{L}_{rec} + \delta \mathcal{L}_{reg} + \gamma \mathcal{L}_{lat}, $$

where $\delta$ and $\gamma$ are weighting factors.
where each term in the equation above is defined as follows:

\[
L_{\text{rec}} = x \log(\hat{x}) + (1 - x) \log(1 - \hat{x})
\]

\[
L_{\text{reg}} = \text{KL}(q(z_0|x; \phi)\|p(z_0)) + \text{KL}(q(z_1|z_0, x; \phi)\|p(z_1|z_0; \theta))
\]

\[
+ \sum_{i=2}^{n} \text{KL}(q(z_i|z_{i-1}, z_0, x; \phi)\|p(z_i|z_{i-1}, z_0; \theta))
\]

\[
L_{\text{lat}} = -\|z' - z\|_2.
\]

Note that \(\delta\) and \(\gamma\), which weigh the terms of the overall cost function, are tunable hyper-parameters.

4 Experiments

To evaluate the 3D shape modeling ability of our proposed neural architecture, we conduct several extensive experiments.

In Section 4.3, experimental results in general 3D shape generalization and synthesis are presented as well as an analysis of nearest ground truth neighbors. Then, we evaluate our model on the task of unsupervised shape classification on both ModelNet10 and ModelNet40, introduced in (Wu et al., 2015), by directly using our learned latent features. We also compare our results to previous supervised and unsupervised state-of-the-art methods in Section 4.4. After that, we test our architecture on the task of real-world image reconstruction in Section 4.5, comparing our results to 3D-R2N2 (Choy et al., 2016) and NRSfM (Kar et al., 2015). Finally, we demonstrate the richness of the VSL’s learned semantic embeddings through vector arithmetic using the latent features trained on ModelNet40 in Section 4.6.

4.1 Datasets

ModelNet The ModelNet dataset consists of two variations, ModelNet10 and ModelNet40, introduced in (Wu et al., 2015) with 10 and 40 target classes respectively. ModelNet10 has 3D shapes which are pre-aligned with the same pose across all categories. Alternatively, ModelNet40 (which includes the shapes in ModelNet10) features a variety of poses. We voxelize both ModelNet10 and ModelNet40 with resolution \([30 \times 30 \times 30]\). We use ModelNet40 in most experiments, including those of Section 4.3 and 4.6, in order to test our model’s ability to handle 3D shapes of great variety and complexity. Both ModelNet10 and ModelNet40 are used in the shape classification experiments.

PASCAL 3D The PASCAL 3D dataset is composed of the images from PASCAL VOC 2012 dataset (Everingham et al., 2015), augmented with 3D annotations in PASCAL 3D+ (Xiang et al., 2014). We voxelize 3D CAD models in \([30 \times 30 \times 30]\) and use the same training and testing splits as in (Kar et al., 2015) which was also used in (Choy et al., 2016) to conduct real-world image reconstruction in Section 4.5. We use the bounding box information as provided in the dataset. The only pre-processing applied was image cropping and padding with 0-intensity pixels to create final samples of resolution \([100 \times 100]\) as required in our model.

4.2 Training Protocol

Since training was the same across all experiments with only minor details that were task-dependent, we describe the general procedure and variations in this section. The architecture of the proposed VSL consisted of 5 local latent codes, the dimension of each set to 10 variables. The global latent code was set to a dimension 20 on ModelNet40. For ModelNet10, the model, again, consisted with 5 local latent codes but the dimensionality of each was set to 5 and the global latent code set to a dimensionality of 10.

We set the hyper-parameter \(\delta = 10^{-3}\) across training on both ModelNet10 and ModelNet40. We optimize parameters by maximizing the loss function defined in Equation 4 using the ADAM adaptive learning rate scheme (Kingma and Ba, 2014), with step size set to \(5 \times 10^{-5}\). For the experiments of Sections 4.3, 4.4, and 4.6, parameter updates were calculated over mini-batches of 200 samples on ModelNet40 and 100 samples on ModelNet10, with training conducted over 2500 epochs.

For the experiment in Section 4.5, we train VSL in two variations, jointly on all classes and separately on each class. We use 5 local latent codes with each set to a dimensionality of 5 and a global latent
code of dimensionality 20 for the jointly trained model. For the separately trained model, we use 3 local latent codes with each set to dimensionality of 2 and a global latent code of dimensionality 5. Mini-batches of 40 were used to train the joint model and 5 samples were used for the separately trained model. For both model variations, dropout (Srivastava et al., 2014) with $p_{\text{drop}} = 0.2$ was used to control for over-fitting, and early stopping was employed (resulting in only 150 epochs).

For Section 4.5, which involved image reconstruction and thus required the loss term $L_{\text{lat}}$, instead of searching for an optimal value for the hyper-parameter $\gamma$ through cross-validation, we employed a “warming-up” schedule, similar to that of (Sønderby et al., 2016). “Warming-up” involves gradually increasing $\gamma$ (log-scale as shown in Figure 2), which controls the relative weighting of $L_{\text{lat}}$ in Equation 4. The schedule is defined as follows,

$$
\gamma = \begin{cases} 
10^{(t/10)-8} & t \leq 50 \\
(t - 40) \cdot 10^{-3} & 50 < t < 100 \\
5 \cdot 10^{-3} & t \geq 100.
\end{cases}
$$

Figure 2 depicts, empirically, the benefits of employing a warming-up schedule over using a fixed, externally set coefficient for the $L_{\text{lat}}$ term in our image reconstruction experiment. We remark that using a warming-up schedule plays an important role in acquiring good performance on the image reconstruction task.

### 4.3 Shape Generation and Learning

To examine our model’s ability to generate high-resolution 3D shapes with realistic details, we design a task that involves shape interpolation and shape generation. We add Gaussian noise to the learned latent embeddings on test data of ModelNet40 and then use our model to generate “unseen” samples that are similar to the input voxel. In effect, we generate objects from our VSL model directly from vectors, without a reference image/object.
4.4 Shape Classification

One way to test the expressiveness of our model would be to conduct shape classification directly using the learned embeddings. We evaluate our learned features in a fully unsupervised manner on the ModelNet dataset [Wu et al. (2015)] by concatenating both the global latent variable and the local latent layers $[z_0:m]$. We train a Support Vector Machine with an RBF kernel for classification.

Table 1 shows the performance of previous state-of-the-art unsupervised and unsupervised methods in shape classification on the ModelNet dataset. Notably, the most recent unsupervised state-of-the-art results were produced by the 3D-GAN (Wu et al., 2016), which used features from 3 layers of convolutional networks with total dimensions $[62 \times 32^3 + 128 \times 16^3 + 56 \times 8^3]$. This is a far larger feature space than that required by our own model, which is simply $[5 \times 5 + 10]$ (for 10-way classification) and $[5 \times 10 + 20]$ (for 40-way classification).

![Figure 4: Randomly generated results from the proposed Variational Shape Learner trained on ModelNet40. The nearest neighbors are the ground-truth shapes, fetched from the test data, and placed for reference in the last column of the table.](image)

![Figure 5: Shape generation from previous state-of-the-art approaches. Up: generated shapes in resolution $[30 \times 30 \times 30]$ from (Wu et al., 2015); Down: generated shapes in resolution $[64 \times 64 \times 64]$ from (Wu et al., 2016).](image)
Table 1: ModelNet classification results for both unsupervised and supervised methods.

| Supervision | Method                        | Classification Rate |
|-------------|-------------------------------|---------------------|
|             |                               | ModelNet10 | ModelNet40 |
| Supervised  | 3D ShapeNets (Wu et al., 2015) | 83.5%      | 77.3%      |
|             | DeepPano (Shi et al., 2015)   | 85.5%      | 77.6%      |
|             | VoxNet (Maturana and Scherer, 2015) | 92.0%      | 83.0%      |
|             | MVCNN (Su et al., 2015)       | -          | 90.1%      |
|             | ORION (Sedaghat et al., 2016) | 93.8%      | -          |
| Unsupervised| SPH (Kazhdan et al., 2003)    | 79.8%      | 68.2%      |
|             | LFD (Chen et al., 2003)       | 79.9%      | 75.5%      |
|             | T-L Network (Girdhar et al., 2016) | 74.4% | -          |
|             | VConv-DAE (Sharma et al., 2016) | 80.5%      | 75.5%      |
|             | 3D-GAN (Wu et al., 2016)      | 91.0%      | 83.3%      |
|             | VSL (ours)                    | 91.0%      | 84.5%      |

Our model also outperforms other supervised methods, including 3D ShapeNet (Wu et al., 2015) and DeepPano (Shi et al., 2015), by a large margin.

In order to visualize the learned feature embeddings, we employ t-SNE (Maaten and Hinton, 2008) to map high dimensional features to a 2D plane. The visualization is shown in Figure 6.

---

4.5 Single Image 3D Model Retrieval

Another application of our proposed VSL model would be to perform real-world, 3D model retrieval. This is a challenging problem given that the model needs to cope with various real-world images under a variety of lighting conditions and resolutions. Furthermore, there are many instances of model occlusion as well as different color gradings.

We use the PASCAL 3D (Xiang et al., 2014) dataset and utilize the same exact training and testing splits from (Kar et al., 2015). These samples correspond to images from the PASCAL VOC 2012 (Everingham et al., 2015) of which we crop and resize to a resolution of $[100 \times 100]$, padded with 0-intensity pixels to ensure square-shaped images. We compare our results with recent methods including the NRSfM (Kar et al., 2015) and 3D-R2N2 (Choy et al., 2016) models, which also used the exact same experimental configurations.

We train our model in two different ways: 1) jointly on all categories, and 2) separately on each category. Visually, we observe better reconstruction results in Figure 7. Unlike the NRSfM (Kar et al., 2015), our model does not require any segmentation, pose information, or keypoints. In addition, our model is trained from scratch while the 3D-R2N2 is pre-trained using the ShapeNet dataset (Chang et al., 2015). However, our jointly trained architecture does not surpass 3D-R2N2, which is also a jointly trained model. This performance difference is largely due to the fact that the 3D-R2N2 is specifically designed for the image reconstruction task and furthermore employs a deep residual network (He et al., 2016) to help the model learn richer semantic features.
Figure 7: Reconstruction samples of PASCAL VOC dataset from the separately trained VSL. Note that, our model uses resolution $[30 \times 30 \times 30]$ while both 3D-R2N2 and NRSfM made use of resolution $[32 \times 32 \times 32]$, thus visualizations will differ slightly.

Quantitatively, we compare our model to the NRSfM (Kar et al., 2015) and two versions of the 3D-R2N2 (Choy et al., 2016), one with a simple LSTM structure and another that uses a deep residual network. Results are shown in Table 2 using the Intersection-of-Union (IoU) metric. Our results show that our jointly trained model is comparable to the 3D-R2N2 LSTM variant and our separately trained model surpasses the 3D-R2N2 ResNet structure in 8 out of 10 categories, half of them surpassed by a wide margin. Note that one could replace our convolutional network components with residual network components instead, of which we leave to future work.

Table 2: Per-category voxel prediction on PASCAL VOC dataset using Intersection-of-Union (IoU).

|        | aero | bike | boat | bus | car | chair | mbike | sofa | train | tv | mean |
|--------|------|------|------|-----|-----|-------|-------|------|-------|----|------|
| NRSfM  | 0.298| 0.144| 0.188| 0.501| 0.472| 0.234 | 0.361 | 0.149| 0.249 | 0.492| 0.318 |
| 3D-R2N2 [LSTM-1] | 0.472 | 0.330 | 0.466 | 0.677 | 0.579 | 0.203 | 0.474 | 0.251 | 0.518 | 0.438 | 0.456 |
| 3D-R2N2 [Res3D-GRU-3] | 0.544 | 0.499 | 0.560 | 0.816 | 0.699 | 0.280 | 0.649 | 0.332 | 0.672 | 0.574 | 0.571 |
| VSL (jointly trained) | 0.514 | 0.269 | 0.327 | 0.558 | 0.633 | 0.199 | 0.301 | 0.173 | 0.402 | 0.337 | 0.432 |
| VSL (separately trained) | 0.631 | 0.657 | 0.554 | 0.856 | 0.786 | 0.311 | 0.666 | 0.601 | 0.804 | 0.454 | 0.619 |

4.6 Shape Arithmetic

Another way to explore the learned embeddings is to perform various vector operations in the latent space, much like that done in (Wu et al., 2016; Girdhar et al., 2016). We show some interesting results of our shape arithmetic experiment in Figure 8. Different from previous results, all of our objects are sampled from the model embeddings trained using the whole dataset with 40 classes. Furthermore, unlike the blurrier generations of (Girdhar et al., 2016), the VSL seems to generate very interesting combinations of the input embeddings (without the need for matching to actual shapes in the original dataset). The resultant objects appear to clearly embody the intuitive meaning of the vector operators.

Figure 8: Visualization of various samples of our shape arithmetic experiment.

5 Conclusion

In this paper, we proposed the Variational Shape Learner, a hierarchical latent-variable model for 3D shape understanding, learnable through variational inference. In particular, we have demonstrated 3D shape generation results on a popular benchmark, the ModelNet dataset. We also use the learned embeddings to obtain state-of-the-art unsupervised shape classification results as well as generate unseen shapes using shape arithmetic. Future work will entail a more thorough investigation of the embeddings learned by our hierarchical latent-variable model as well as integration of better prior distributions into the framework.
References
Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al. (2015). Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012.

Chen, D.-Y., Tian, X.-P., Shen, Y.-T., and Ouhyoung, M. (2003). On visual similarity based 3d model retrieval. In Computer graphics forum, volume 22, pages 223–232. Wiley Online Library.

Choy, C. B., Xu, D., Gwak, J., Chen, K., and Savarese, S. (2016). 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. arXiv preprint arXiv:1604.00449.

Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A. C., and Bengio, Y. (2015). A recurrent latent variable model for sequential data. In Advances in neural information processing systems, pages 2980–2988.

Edwards, H. and Storkey, A. (2016). Towards a neural statistician. arXiv preprint arXiv:1606.02185.

Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. International Journal of Computer Vision, 111(1):98–136.

Fukushima, K. (1988). Neocognitron: A hierarchical neural network capable of visual pattern recognition. Neural networks, 1(2):119–130.

Girdhar, R., Fouhey, D. F., Rodriguez, M., and Gupta, A. (2016). Learning a predictable and generative vector representation for objects. arXiv preprint arXiv:1603.08637.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In Advances in Neural Information Processing Systems, pages 2672–2680.

Gregor, K., Danihelka, I., Graves, A., Rezende, D. J., and Wierstra, D. (2015). Draw: A recurrent neural network for image generation. arXiv preprint arXiv:1502.04623.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 770–778.

Kar, A., Tulsiani, S., Carreira, J., and Malik, J. (2015). Category-specific object reconstruction from a single image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1966–1974.

Kazhdan, M., Funkhouser, T., and Rusinkiewicz, S. (2003). Rotation invariant spherical harmonic representation of 3d shape descriptors. In Symposium on geometry processing, volume 6, pages 156–164.

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

Knopp, J., Prasad, M., Willems, G., Timofte, R., and Van Gool, L. (2010). Hough transform and 3d surf for robust three dimensional classification. In European Conference on Computer Vision, pages 589–602. Springer.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. Neural computation, 1(4):541–551.

Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. Journal of Machine Learning Research, 9(Nov):2579–2605.
Maturana, D. and Scherer, S. (2015). Voxnet: A 3d convolutional neural network for real-time object recognition. In Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, pages 922–928. IEEE.

Patterson IV, A., Mordohai, P., and Daniilidis, K. (2008). Object detection from large-scale 3d datasets using bottom-up and top-down descriptors. In European Conference on Computer Vision, pages 553–566. Springer.

Pu, Y., Gan, Z., Henao, R., Yuan, X., Li, C., Stevens, A., and Carin, L. (2016). Variational autoencoder for deep learning of images, labels and captions. In Advances in Neural Information Processing Systems, pages 2352–2360.

Radford, A., Metz, L., and Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.

Rezende, D. J., Eslami, S. A., Mohamed, S., Battaglia, P., Jaderberg, M., and Heess, N. (2016). Unsupervised learning of 3d structure from images. In Advances in Neural Information Processing Systems, pages 4996–5004.

Rusu, R. B., Blodow, N., and Beetz, M. (2009). Fast point feature histograms (fpfh) for 3d registration. In Robotics and Automation, 2009. ICRA’09. IEEE International Conference on, pages 3212–3217. IEEE.

Sedaghat, N., Zolfaghari, M., and Brox, T. (2016). Orientation-boosted voxel nets for 3d object recognition. arXiv preprint arXiv:1604.03351.

Serban, I. V., Ororbia, I., Alexander, G., Pineau, J., and Courville, A. (2016). Multi-modal variational encoder-decoders. arXiv preprint arXiv:1612.00377.

Sharma, A., Grau, O., and Fritz, M. (2016). Vconv-dae: Deep volumetric shape learning without object labels. In Computer Vision–ECCV 2016 Workshops, pages 236–250. Springer.

Shi, B., Bai, S., Zhou, Z., and Bai, X. (2015). Deeppano: Deep panoramic representation for 3-d shape recognition. IEEE Signal Processing Letters, 22(12):2339–2343.

Sønderby, C. K., Raiko, T., Maaløe, L., Sønderby, S. K., and Winther, O. (2016). Ladder variational autoencoders. In Advances in Neural Information Processing Systems, pages 3738–3746.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1):1929–1958.

Su, H., Maji, S., Kalogerakis, E., and Learned-Miller, E. (2015). Multi-view convolutional neural networks for 3d shape recognition. In Proceedings of the IEEE International Conference on Computer Vision, pages 945–953.

Wu, J., Zhang, C., Xue, T., Freeman, B., and Tenenbaum, J. (2016). Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In Advances in Neural Information Processing Systems, pages 82–90.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., and Xiao, J. (2015). 3d shapenets: A deep representation for volumetric shapes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1912–1920.

Xiang, Y., Mottaghi, R., and Savarese, S. (2014). Beyond pascal: A benchmark for 3d object detection in the wild. In IEEE Winter Conference on Applications of Computer Vision (WACV).

Xie, J., Fang, Y., Zhu, F., and Wong, E. (2015). Deepshape: Deep learned shape descriptor for 3d shape matching and retrieval. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1275–1283.

Yan, X., Yang, J., Yumer, E., Guo, Y., and Lee, H. (2016). Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In Advances in Neural Information Processing Systems, pages 1696–1704.

Zhu, Z., Wang, X., Bai, S., Yao, C., and Bai, X. (2016). Deep learning representation using autoencoder for 3d shape retrieval. Neurocomputing, 204:41–50.
• **Source Code.** The source code for this paper has been uploaded to the public repository: [https://github.com/LorenMt/vsl](https://github.com/LorenMt/vsl). This code can be used to easily re-implement most of the experiments presented in the paper.

• **More results on shape interpolation.**

![Interpolation Results](image)

Figure 9: Interpolation results of the Variational Shape Learner on ModelNet40.
• More results on shape generation.

| Shape Generation | Nearest Neighbor |
|------------------|------------------|
| *bathtub*        | *bathtub*        |
| *bed*            | *bed*            |
| *bookshelf*      | *bookshelf*      |
| *car*            | *car*            |
| *cup*            | *cup*            |
| *guitar*         | *guitar*         |
| *lamp*           | *lamp*           |
| *piano*          | *piano*          |
| *stool*          | *stool*          |
| *tv-stand*       | *tv-stand*       |

Figure 10: Shape generation results of the Variational Shape Learner on ModelNet40.
More results on image reconstruction.

Figure 11: Image reconstruction results of the Variational Shape Learner on PASCAL3D.