# 3D Volumetric Modeling with Introspective Neural Networks

**Wenlong Huang,*[1] Brian Lai,*[2] Weijian Xu,[3] Zhuowen Tu[3]**
[1]University of California, Berkeley
[2]University of California, Los Angeles
[3]University of California, San Diego

## Abstract

In this paper, we study the 3D volumetric modeling problem by adopting the Wasserstein introspective neural networks method (WINN) that was previously applied to 2D static images. We name our algorithm 3DWINN which enjoys the same properties as WINN in the 2D case: being simultaneously generative and discriminative. Compared to the existing 3D volumetric modeling approaches, 3DWINN demonstrates competitive results on several benchmarks in both the generation and the classification tasks. In addition to the standard inception score, the Fréchet Inception Distance (FID) metric is also adopted to measure the quality of 3D volumetric generations. In addition, we study adversarial attacks for volumetric data and demonstrate the robustness of 3DWINN against adversarial examples while achieving appealing results in both classification and generation within a single model. 3DWINN is a general framework and it can be applied to the emerging tasks for 3D object and scene modeling.[1]

## Introduction

The rich representation power of the deep convolutional neural networks (CNN) (LeCun et al. 1989), as a discriminative classifier, has led to a great leap forward for the image classification and regression tasks (Krizhevsky 2009; Szegedy et al. 2015; Simonyan and Zisserman 2015; He et al. 2016). The generative modeling aspect of the CNN is also under explosive development, due to the recent success of the generative adversarial networks (GAN) family models (Goodfellow et al. 2014a; Radford, Metz, and Chintala 2016; Arjovsky, Chintala, and Bottou 2017; Zhu et al. 2017; Karras et al. 2018) and the variational auto-encoder (VAE) model (Kingma and Welling 2014).

The field of 3D object modeling is also enjoying a steady improvement, but with less drastic developments when compared to the 2D image domain. The format of 3D input can be roughly divided into three categories: (1) voxel based (Wu et al. 2015; Qi et al. 2016; Maturana and Scherer 2015; Xie et al. 2018a), (2) multi-view based (Su et al. 2015; Qi et al. 2016), and (3) point-cloud based (Qi et al. 2017a; 2017b; Xie et al. 2018b).



[1]* indicates equal contribution

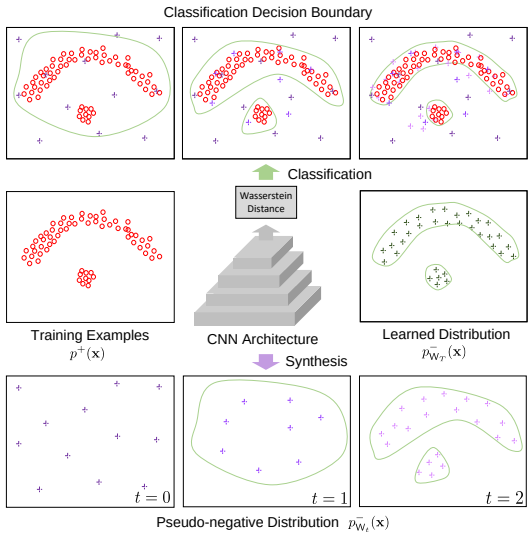


Figure 1: Diagram of the Wasserstein introspective neural networks (adapted from Figure 1 of (Lee et al. 2018)); the upper figures indicate the gradual refinement over the classification decision boundary between training examples (circles) and pseudo-negatives (pluses); the lower figures show that the pseudo-negative examples (pluses) are synthesized within current classification decision boundary.

Like in the 2D image domain, there are two typical tasks for 3D object modeling: supervised classification (Qi et al. 2017a; Xie et al. 2018b) and unsupervised generation (Wu et al. 2015; Xie et al. 2018a). In this paper, we attempt to build a 3D object model that is simultaneously generative and discriminative by extending the recently developed introspective neural networks (INN) (Lazarow, Jin, and Tu 2017; Jin, Lazarow, and Tu 2017; Lee et al. 2018) from modeling pixel-based 2D images to modeling voxel-based 3D volumetric data. Specifically, we adopt the Wasserstein introspective neural networks (WINN) method (Lee et al. 2018) and name our approach Wasserstein introspective neural networks for 3D modeling (3DWINN).

The main focus of our work lies in the extension of intro-

spective neural network framework (Tu 2007; Jin, Lazarow, and Tu 2017) to 3D volumetric object modeling. The contributions of this paper can be summarized as follows:

- 3DWINN achieves state-of-the-art results on both generative modeling and discriminative classification tasks for 3D volumetric data within the same model.
- In 3D generative modeling, 3DWINN incorporates the Fréchet Inception Distance (FID) scores as an evaluation metric. FID has demonstrated to be more consistent to human judgment than Inception Score and has been widely used in 2D case. We present comparisons against 3D-GAN and 3D-DescriptorNets using this metric and show that 3DWINN demonstrates state-of-the-art performance.
- In discriminative modeling, 3DWINN studies adversarial attacks for volumetric data, which has been previously under explored. We demonstrate that 3DWINN attains robustness to adversarial examples while achieving appealing results on both classification and generation as a single model.

Overall, 3DWINN exhibits a general classification capability and robustness over adversarial examples that do not exist within the previous 3D generators such as 3D-GAN (Wu et al. 2016) and 3D-DescriptorNet (Xie et al. 2018a) as they need an additional classifier, and demonstrates a generation capability that the existing 3D discriminative classifiers like 3D ShapeNets (Wu et al. 2015) do not possess. We evaluate 3DWINN on the standard benchmark dataset, ModelNet (Wu et al. 2015), for multiple tasks including 3D object classification, 3D object generation, and adversarial attacks on 3D classification.

## Related Work

In this section, we discuss existing work for 3D object modeling, which is itself a long standing problem (McInerney and Terzopoulos 1996). Here, we focus on a recent line of deep learning based approaches. As stated in the previous section, the existing literature typically builds on top of three types of input format, (1) volumetric data, (2) multi-view images, and (3) point clouds. In the past, various 3D object classifiers for discriminative classification have been proposed (Wu et al. 2015; Maturana and Scherer 2015; Su et al. 2015; Qi et al. 2016; 2017a; 2017b; Xie et al. 2018b). In this work, we give our special attention to 3D generative models (Wu et al. 2015; Xie et al. 2018a) which usually do not have the direct discriminative classification capability if no additional classifier is obtained.

Typical generative 3D models include volume-based methods such as 3D-GAN (Wu et al. 2016), 3D-VAE (Kingma and Welling 2014), 3D-DescriptorNet (Xie et al. 2018a), and point-cloud-based approaches like PointOutNet (Fan, Su, and Guibas 2017). However, none of these models (without an additional classifier) itself produces competitive results for the standard multi-class 3D object classification task. Here, we build a model that is simultaneously generative and discriminative and we are particularly inspired by the recent introspective neural networks (INN) (Jin, Lazarow, and Tu 2017; Lee et al. 2018) that are capable of performing the standard multi-class classification task while synthesizing new samples inside the classifier.

3D-GAN (Wu et al. 2016) successfully builds a 3D object model by adopting a VAE encoder (Kingma and Welling 2014) to generate 3D objects from 2D images using generative adversarial networks (GAN) (Goodfellow et al. 2014b). 3D-DescriptorNet (Xie et al. 2018a) also creates a very impressive generative model with the state-of-the-art results on various tasks including 3D object generation and classification. 3DWINN, in contrast, follows the direction of INN/WINN (Jin, Lazarow, and Tu 2017; Lee et al. 2018) by progressively updating the CNN classifier directly whereas 3D-DescriptorNet is itself a generator only. For example, to produce the 3D object classification result reported in (Xie et al. 2018a), an unsupervised learning process is first applied to 3D-DescriptorNet to perform feature extraction, followed by another step of training a separate logistic regression classifier. In contrast, 3DWINN is a single model that is simultaneously generative and discriminative, which is able to directly enhance the standard CNN classifier on the supervised classification task with additional robustness to adversarial attacks which 3D-DescriptorNet is not able to demonstrate.

The key difference between INN and WINN (Jin, Lazarow, and Tu 2017; Lazarow, Jin, and Tu 2017; Lee et al. 2018) and 3DWINN is that 3DWINN studies the classification and generation problems specifically for 3D volumetric data. Although 2D to 3D extension seems natural, several works including 3D ShapeNets (Wu et al. 2015), 3D-GAN (Wu et al. 2016), Volumetric CNN (Qi et al. 2016), and 3D-DescriptorNet (Xie et al. 2018a), have shown such extension is inherently non-trivial.

For classification, methods that use multi-view representation often significantly outperformed prior works like 3D ShapeNets that use volumetric data. In terms of generation, although 3D-GAN achieves compelling performance, it is not able to consistently generate samples faithful to training data as shown by FID scores (Heusel et al. 2017) (see Table 3). Two unique characteristics of volumetric data might cause such difficulty: (1) The high dimensionality constrains the model complexity so it cannot fully exploit the power of 3D representations; (2) The data sparsity introduces significant challenges: for instance, a pixel in 2D represents the observed information while a binary voxel in 3D only indicates whether the object lies within it.

## Method

In this section, we introduce our method, Wasserstein introspective neural networks for 3D volumetric modeling (3DWINN) that largely follows the basic principles of INN (Lazarow, Jin, and Tu 2017; Jin, Lazarow, and Tu 2017), and particularly draws inspiration from the WINN (Lee et al. 2018) algorithm. Specifically, we show how we leverage the previous advances in the introspective neural network method (Jin, Lazarow, and Tu 2017; Lazarow, Jin, and Tu 2017; Lee et al. 2018) and adopt it to build our unified framework for 3D object generative modeling and classification.

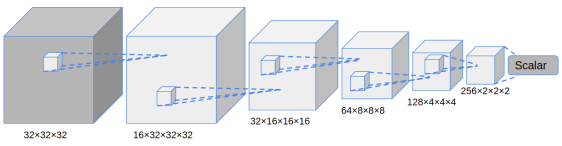


Figure 2: convolutional neural network used in 3DWINN. The convolution filters are of size $3 \times 3 \times 3$ with strides 1. Average pooling is used between each layer except the first layer.

## Introspective Neural Networks

We first introduce the introspective neural network (INN) framework that was developed in (Jin, Lazarow, and Tu 2017; Lazarow, Jin, and Tu 2017). An introspective neural network (INN) is a convolutional neural network that is simultaneously discriminative and generative. Later, (Lee et al. 2018) shows a mathematical connection between the WGAN (Arjovsky, Chintala, and Bottou 2017) and INN (Lazarow, Jin, and Tu 2017) and adapts the Wasserstein distance into INN, resulting in a significant boost in modeling ability by INN. As shown by (Lee et al. 2018), the set of training examples $S = \{x_i \mid i = 1, \ldots, n\}$, where $x_i \in \mathbb{R}^{32\times32\times32}$ in our 3DWINN, constitutes the positive examples of the distribution we would like to model. Built upon the generative via discriminative (GDL) framework by (Tu 2007), INN is able to model the distribution $p(\mathbf{x}|y=+1)$ by sequentially defining new pseudo-negative samples $p_t(\mathbf{x}|y=-1; W_t)$ (shortened as $p^-_{W_t}(\mathbf{x})$) by computing the following:

$$\frac{1}{Z_t} \exp\{\mathbf{w}_t^{(1)} \cdot \phi(\mathbf{x}; \mathbf{w}_t^{(0)})\} \cdot p_0^-(\mathbf{x}),\ t = 1, \ldots, T \quad (1)$$

where $Z_t = \int \exp\{\mathbf{w}_t^{(1)} \cdot \phi(\mathbf{x}; \mathbf{w}_t^{(0)})\} \cdot p_0^-(\mathbf{x}) d\mathbf{x}$, $W_t$ is the model parameter including $\mathbf{w}_t^{(0)}$ and $\mathbf{w}_t^{(1)}$ at step $t$, and $p_0^-(\mathbf{x})$ is the initial distribution. In our work, we use a Gaussian distribution $N(0, 1)$. Following (Lee et al. 2018), we perform stochastic gradient Langevin dynamics (Welling and Teh 2011):

$$\Delta\mathbf{x} = \frac{\epsilon}{2} \nabla(\mathbf{w}_t^{(1)} \cdot \phi(\mathbf{x}; \mathbf{w}_t^{(0)})) + \eta$$

where $\eta \sim N(0, \epsilon)$ is a Gaussian distribution and $\epsilon$ is the step size that is annealed in the sampling process.

It is shown by (Jin, Lazarow, and Tu 2017; Lazarow, Jin, and Tu 2017) that we can obtain the following using the iterative reclassification-by-synthesis process guided by Eq. (1):

$$p^-_{W_t}(\mathbf{x}) \overset{t=\infty}{\rightarrow} p(\mathbf{x}|y=+1), \quad (2)$$

As noted in previous sections, 3DWINN enjoys the benefit of being simultaneously discriminative and generative; few modifications are needed if we desire a model that focuses on classification instead of generation. In supervised classification setting, since labels are provided during training, we synthesize pseudo-negative examples based on class categories. Following (Lee et al. 2018), we formulate the loss function as follows:

$$\begin{aligned} L(W_t) \quad &= -\sum_{\mathbf{x}_i \in S_+} \ln \frac{\exp\{\mathbf{w}_t^{(1)y_i} \cdot \phi(\mathbf{x}_i; \mathbf{w}_t^{(0)}\}}{\sum_{k=1}^{K} \exp\{\mathbf{w}_t^{(1)k} \cdot \phi(\mathbf{x}_i; \mathbf{w}_t^{(0)}\}} \\ &+ \alpha\big(\sum_{\mathbf{x}_i \in S_-^t} f_{W_t}(\mathbf{x}_i) - \sum_{\mathbf{x}_i \in S_+} f_{W_t}(\mathbf{x}_i) \\ &+ \lambda \mathbb{E}_{\hat{\mathbf{x}} \sim \mathbf{p}_{\hat{\mathbf{x}}}}[(||\nabla_{\hat{\mathbf{x}}} f_{W_t}(\hat{\mathbf{x}})||_2 - 1)^2]\big), \end{aligned} \quad (3)$$

where $W_t = \langle \mathbf{w}_t^{(0)}, \mathbf{w}_t^{(1)_1}, ..., \mathbf{w}_t^{(1)_K} \rangle$. $\mathbf{w}_t^{(1)_k}$ denotes the weights for the $k$-th class in the last linear layer and $\mathbf{w}_t^{(0)}$ denotes the weights for all the previous layers in the CNN.

## 3DWINN

3D geometric data, unlike 2D image data, does not have a canonical representation. Many geometric data structures, such as polygon meshes and point clouds, are often criticized for not being regular-structured, and thus cannot leverage the recent rapid advances in deep learning for 2D images. Although the multi-view rendered image representation achieved much success in 3D object recognition (Su et al. 2015; Qi et al. 2016), it cannot be used directly to represent 3D objects without conversion to geometric data structures because of its 2D nature. Therefore, in 3DWINN, a unified framework for both 3D object generative modeling and 3D object classification, we choose to represent the raw meshed shapes as 3D binary occupancy grids since their highly-organized structures offer rich contextual information that is crucial to 3D shape recognition and modeling. As done in 3D ShapeNets (Wu et al. 2015), VoxNet (Maturana and Scherer 2015), and 3D-GAN (Wu et al. 2016), we use the binary voxelization method: specifically, if the surface of an input raw mesh data lies within the voxel at a spatial location, the value at that location is 1. Otherwise, if the voxel at a spatial location is empty, the corresponding value is 0.

Table 1: Volumetric Convolutional Neural Network used in 3DWINN. We apply Layer Normalization (Ba, Kiros, and Hinton 2016) and use Leaky ReLU (Xu et al. 2015) after each convolutional layer.

| Layer | Filter size/stride | Output size |
|---|---|---|
| Input | | 32×32×32×1 |
| Conv4-16 | 3×3×3/1 | 32×32×32×16 |
| Conv4-32 | 3×3×3/1 | 32×32×32×32 |
| Avg pool | 2×2×2/2 | 16×16×16×32 |
| Conv4-64 | 3×3×3/1 | 16×16×16×64 |
| Avg pool | 2×2×2/2 | 8×8×8×64 |
| Conv4-128 | 3×3×3/1 | 8×8×8×128 |
| Avg pool | 2×2×2/2 | 4×4×4×128 |
| Conv4-256 | 3×3×3/1 | 4×4×4×256 |
| Avg pool | 2×2×2/2 | 2×2×2×256 |
| FC-1 | | 1×1×1×1 |

We use a 5-layer convolutional neural network shown in Figure 2 for this paper. The network implementation details are shown in Table 1. Despite the network structure used in our work, however, it is noted that 3DWINN is a general framework for 3D volumetric modeling and is agnostic to the type of classifier used.

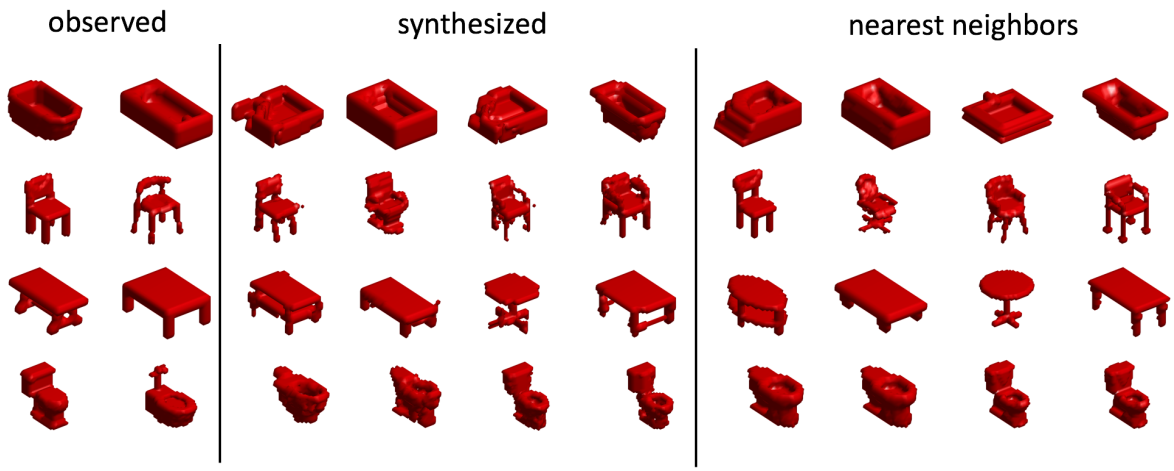


Figure 3: 3D objects synthesized by 3DWINN. "observed" denotes the 3D objects from the training set of ModelNet10. "synthesized" denotes the 3D objects synthesized by 3DWINN. "nearest neighbors" denotes the nearest neighbors retrieved from the training sets for their corresponding synthesized 3D objects, where the first column in "nearest neighbors" corresponds to the first column in "synthesized", and so on.

## Adversarial Examples

Adversarial attacks on neural networks are a well known weakness of neural networks and is an active area of research. The extension of adversarial attacks on image classifiers to 3D volumetric data is non-trivial as it has been previously under explored. The discretized nature of of voxelized objects, however, is the source of ambiguity. Since common volumetric deep learning based methods are computationally limited to $32 \times 32 \times 32$ voxel resolution, 3D voxelized representations are often more blocky than the real data and seemingly stray voxels where a small portion of the real, continuous 3D object overlaps into the region of a voxel and causes the voxel to be marked as filled are common around the edges and fine structures in the voxelized representation. Attacks that add small amounts of perturbation may also successfully attack voxel-based 3D classification models while still maintaining the primary structures of the 3D objects.

It is argued by (Jin, Lazarow, and Tu 2017; Lazarow, Jin, and Tu 2017; Lee et al. 2018) that the reclassification-by-synthesis process of introspective neural networks (INN) helps tighten the decision boundary as shown in Figure 1 and makes CNNs trained with the INN algorithm more robust to adversarial attacks since discriminative CNNs seem to be vulnerable to adversarial attacks because of their linear nature as posited by (Goodfellow et al. 2014b).

To verify this property of INN still holds in 3D, we use a vanilla volumetric CNN as the baseline model and a CNN with the same structure but trained with the INN algorithm to compare their classification accuracy against adversarial examples.

CNN classifiers trained with conventional discriminative learning can easily be fooled by adversarial examples, as they learn the most salient (thus discriminative) features. In contrast, CNN classifiers trained with our algorithm learn features that are necessary for fully reconstructing the input space, but with some redundancy for achieving good classification on unperturbed clean images. We hypothesize that this redundancy makes the classifier more robust to small perturbations in input space. Since our classifier has extra information beyond the most discriminative features, one needs to add more perturbation to fool the classifier trained with our method.

Adversarial examples of voxelized 3D objects behave differently from adversarial examples in the 2D case because of their binary nature. The perturbation that is introduced is much more apparent than in the 2D case. In the 2D case, each colored pixel has $256^3$ options, assuming 8-bit color channels, and adding a small perturbation is often times imperceptible by the human eye. In the 3D case, however, when a specific voxel is perturbed, the voxel changes from being empty to filled, or vice versa. Therefore, introducing even a small perturbation will noticeably alter the original example. Therefore, we adopt the FGSM method by (Goodfellow, Shlens, and Szegedy 2015) for generating adversarial examples with a smaller perturbation size than in the 2D case. Furthermore, we apply the stronger iterative FGSM attack (Kurakin, Goodfellow, and Bengio 2016) in order to validate the robustness of 3DWINN.

## Experiment

In this section, we present experiments conducted on two common tasks in 3D computer vision, 3D object generation and 3D object classification, in both of which 3DWINN demonstrates competitive results. Furthermore, we study the problem of 3D adversarial attacks, and we show that 3DWINN attains additional robustness against adversarial examples compared to the baseline method.

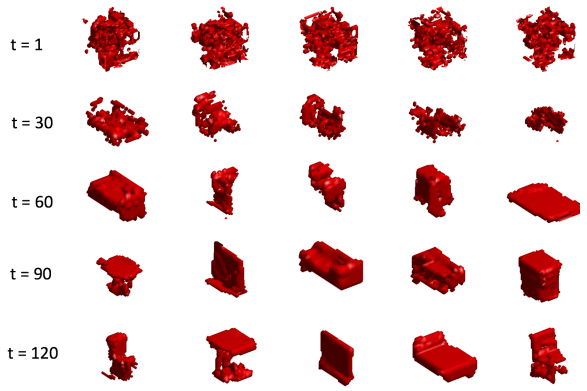


Figure 4: Pseudo-negative (fake) examples generated during classification training. As step $t$ increases, 3DWINN gradually obtains better generative modeling ability which strengthens its classification performance.

## 3D Object Generation

We evaluate our model in a widely used 3D CAD dataset ModelNet introduced by (Wu et al. 2015). In this experiment, we use a common testbed ModelNet10, which is a subset of ModelNet consisting of 10 categories of 3D CAD data with 3,991 training examples and 908 test examples.

**Training Details** We train the discriminator network with the Adam optimizer (Kingma and Ba 2015). The learning rate is 0.0001 with $\beta_1 = 0$ and $\beta_2$ = 0.9. As in (Lee et al. 2018), we keep the coefficient of the gradient penalty term $\lambda$ as 10. We perform mini-batch training with size 128, and half of the examples in each mini-batch are randomly chosen from the training set while the other half are taken from the set of pseudo-negative examples synthesized in previous iterations. We then normalize each mini-batch of 3D volumetric data by subtracting its mean value, as in (Xie et al. 2018a).

As done in the previous works of introspective neural networks (Lee et al. 2018; Lazarow, Jin, and Tu 2017; Jin, Lazarow, and Tu 2017), we synthesize pseudo-negative examples by performing gradient ascent on the input space. Similar to the classification step, we perform mini-batch training with size 128 and the Adam optimizer. We set the learning rate to be 0.005 with $\beta_1 = 0.8$ and $\beta_2 = 0$. We also perform cascade training as in (Lee et al. 2018): in each cascade, the input is initialized with the examples synthesized in the last cascade. If it is the first cascade, we initialize the input with a Gaussian noise $N(0, 1)$.

**Qualitative Evaluation** In order to obtain more detailed synthesized samples and to have a fair comparison with other recent methods (Wu et al. 2016) and (Xie et al. 2018a), we train one 3DWINN model for each object category. We show the synthesized 3D objects generated by 3DWINN and the observed 3D objects randomly sampled from the training set for comparison in Figure 3. In addition, to demonstrate that our model's capability of generalization after observing the training examples, we also show the nearest neighbor of each synthesized 3D object retrieved from the training set. Because direct calculation of L2 distance over raw 3D objects requires them to be perfectly aligned, which is an impractical constraint in practice and could result in the lack of abstraction of higher level features (such as the style of a chair's handle) when retrieving nearest neighbors, we calculate L2 distance on the extracted features of the last convolutional layer of the reference network as described in the later section. We show that our model does not simply memorize the observed dataset. Furthermore, it can generate realistic 3D objects with considerable diversity and detailed geometries at a level on par with the current state-of-the-art method.

**Quantitative Evaluation** We adopt the Inception score (Salimans et al. 2016), the Fréchet Inception Distance (FID) (Heusel et al. 2017), and the average softmax class probability assigned by the reference network as evaluation metrics for 3DWINN's performance in 3D object generation. The reference network used in all metrics is a volumetric convolutional neural network by (Qi et al. 2016) as in the work by (Xie et al. 2018a).

Inception score is the KL-divergence of the conditional distribution $p(y|x)$ and the marginal distribution $\int p(y|x = G(z))dz$. As suggested by (Salimans et al. 2016), the former indicates the meaningfulness of the test examples and the latter indicates the diversity of the test examples. For fair comparisons with other recent methods, we jointly train a single 3DWINN on all object categories. The conventional way for calculating an inception score is using a single trained model for synthesizing all examples. However, we empirically find that the synthesized examples by 3DWINN from a single iteration tends to have a skewed category distribution, e.g. the categories of synthesized 3D objects are not randomly distributed. We suspect that this may be a nature of introspective neural networks and is worsened by the fact that ModelNet10 is a very skewed dataset with the number of each object category ranging from 100 to 700. Therefore, while maintaining high perceptual quality, the 3D objects synthesized using a single model could still result in a relatively lower inception score because of the lack of diversity in category distribution. Accordingly, to quantitatively evaluate the perceptual quality and to be not affected by the skewness in category distribution, we also report the inception score on the synthesized 3D objects by the models retrieved from the last 10 iterations of the last cascade. As ModelNet10 is skewed in category distribution, we train a single 3DWINN on each object category and report the inception score on the combined synthesized data for comparison as well. Table 2 shows a comparison of different algorithms in terms of the inception score.

It is argued by (Heusel et al. 2017) that the Inception Score is not always consistent with human judgement as it does not use the statistics of real world samples and compare it to the statistics of synthetic samples. Hence, (Heusel et al. 2017) proposed the Fréchet Inception Distance (FID) which better captures this similarity and is more consistent with human judgment than the Inception Score is. However, to the best of our knowledge, this metric has not been applied

Table 2: Inception scores on ModelNet10. "joint-single" denotes the single jointly trained classifier. "joint-multi" denotes multiple jointly trained classifiers. "separate-multi" denotes multiple classifiers separately trained on each category.

| Method | Score |
|---|---|
| 3D Shapenets (Wu et al. 2015) | $4.13 \pm 0.19$ |
| 3D-GAN (Wu et al. 2016) | $8.66 \pm 0.45$ |
| 3D VAE (Kingma and Welling 2014) | $11.02 \pm 0.42$ |
| 3D-DescriptorNet (Xie et al. 2018a) | $\mathbf{11.77 \pm 0.42}$ |
| 3DWINN-joint-single | $7.81 \pm 0.22$ |
| 3DWINN-joint-multi | $8.81 \pm 0.18$ |
| 3DWINN-separate-multi | $10.25 \pm 0.19$ |

to evaluate the quality of synthesized 3D objects. Therefore, we only compare the FID scores obtained by 3DWINN with the provided synthesized samples by 3D-DescriptorNet (Xie et al. 2018a) and the samples generated using the provided pre-trained models of 3D-GAN (Wu et al. 2016). We use the same reference network for evaluation as in the previous section. To calculate the FID score, we extract the activations from the last convolutional layer of the reference network to get a 4096 dimensional feature vector. The formulation of the FID is as follows:

$$d^2 = ||\mu_1 - \mu_2||^2 + Tr(C_1 + C_2 - 2 * \sqrt{(C_1 * C_2)})$$

The Fréchet distance between two multivariate Gaussians $X_1 \sim N(\mu_1, C_1)$ and $X_2 \sim N(\mu_2, C_2)$ where $X_1$ and $X_2$ are the activations of the last convolutional layer of synthetic and real samples, respectively. $C_n$ and $\mu_n$ are the covariance and mean of the activations of the convolutional layer, respectively.

As shown in Table 3, 3DWINN demonstrates significant improvement over 3D-GAN (Wu et al. 2016) and 3D-DescriptorNet (Xie et al. 2018a).

In addition to the above two Inception scores, we also evaluate 3DWINN's generative capability by calculating the average softmax class probability that the reference network assigns to the synthesized 3D objects for the corresponding class. By comparing the results by our method with previous works, we show in Table 4 that 3DWINN can synthesize meaningful and convincing 3D objects at a level on par with the state-of-the-art.

**Limitation** Despite appealing results on several benchmarks, we recognize that 3DWINN has certain limitations in 3D generative modeling. As mentioned in the previous section, a single 3DWINN classifier tends to synthesize samples with skewed category distribution; generating samples with diverse category distribution often requires using several classifiers saved at different stages during training. Although (Lee et al. 2018) proposed to use alternative initialization that uses a convolutional neural network to initialize noise instead of a Gaussian initialization to encourage diversity in synthesized samples, we find empirically that this method might not be as efficient as in 2D. Since the introspective neural network method (Jin, Lazarow, and Tu 2017; Lazarow, Jin, and Tu 2017; Lee et al. 2018) relies on performing gradient ascent via backpropagation on the input space, training on volumetric data of size $32 \times 32 \times 32$, corresponding to a high resolution of roughly $181 \times 181$ in 2D, is relatively a time-consuming process compared to other methods, such as 3D-GAN which carries out synthesis using forward passes. During inference, however, synthesizing realistic 3D objects using 3DWINN is still feasible as it takes about 7 seconds to synthesize a 3D object of size $32 \times 32 \times 32$.

## 3D Object Classification

We further examine 3DWINN's modeling ability for the 3D object classification task on ModelNet10. We use the training/test split included in the dataset for fair comparisons. Because 3DWINN is both a generator and a discriminator, we conduct classification experiments in a supervised manner, which is nearly the same as in the unsupervised generator training setting introduced in the previous section, except that we have negative/multi-class examples now. We follow the WINN algorithm (Lee et al. 2018) for the supervised classification task. To train to perform classification, an existing generative model based approach such as 3D-DescriptorNet (Xie et al. 2018a) would use the features extracted from its intermediate layers and train a separate discriminative classifier, which is not end-to-end and sub-optimal. 3DWINN instead can be trained end-to-end to perform classification, which is a clear advantage over 3D-DescriptorNet ($92.4\%$, shown in Table 6 as an unsupervised classification task) , and reports a competitive result ($93.6\%$, shown in Table 5 as a standard supervised classification task).

**Training Details** In the classification step, we train the discriminator network with Adam (Kingma and Ba 2015). The learning rate is 0.00002 with $\beta_1 = 0$ and $\beta_2 = 0.9$. In the synthesis step, we perform gradient ascent on the input space with Adam. The learning rate is 0.002 with $\beta_1 = 0$ and $\beta_2 = 0.9$. Both the classification and synthesis steps use mini-batch size of 32. The rest of the training details is the same as in the unsupervised synthesis case.

**Supervised 3D Object Classification** Following the reclassification-by-synthesis scheme introduced by (Jin, Lazarow, and Tu 2017), we jointly train a single model on ModelNet10 with corresponding class labels.

As shown in Table 5, 3DWINN achieves results on par with the state-of-the-art when compared to other volumetric based supervised methods. Its performance is also comparable to many methods using other 3D representations, such as rendered multi-view images, which are often pre-trained on large-scale image dataset such as ImageNet (Deng et al. 2009). However, it is worth noting that the test set of ModelNet10 likely contains harder examples than those in the training set: both our baseline model and 3DWINN obtain significantly better results on the validation set, which we manually split from the given training set prior to training, and 3DWINN obtains a 50% error reduction on the validation set over the baseline model. We show the pseudo-negative examples synthesized during classification training in Figure 4. The faithfulness of the synthesized examples to real 3D objects gradually increases as time proceeds. The

Table 3: FID scores on ModelNet10 (lower is better).

| Method | Dresser | Toilet | Night stand | Chair | Table | Sofa | Monitor | Bed | Bathtub | Desk |
|---|---|---|---|---|---|---|---|---|---|---|
| 3D-GAN (Wu et al. 2016) | - | - | - | 469 | - | 517 | - | - | - | 651 |
| 3D-DescriptorNet (Xie et al. 2018a) | 414 | 662 | 517 | 490 | 538 | 494 | 511 | 574 | - | - |
| 3DWINN(ours) | **305** | **474** | **456** | **225** | **220** | **151** | **181** | **222** | 305 | **322** |

Table 4: Softmax Class Probability (higher is better)

| Method | Dresser | Toilet | Night stand | Chair | Table | Sofa | Monitor | Bed | Bathtub | Desk |
|---|---|---|---|---|---|---|---|---|---|---|
| 3D-GAN (Wu et al. 2016) | 0.6314 | 0.8569 | 0.6853 | 0.9700 | 0.8377 | 0.9276 | 0.2493 | 0.7775 | 0.7017 | 0.7936 |
| 3D-DescriptorNet (Xie et al. 2018a) | 0.7678 | **0.9701** | **0.7195** | 0.9920 | 0.8910 | 0.9480 | 0.9473 | 0.9202 | 0.8348 | 0.8203 |
| 3D-VAE (Kingma and Welling 2014) | 0.7010 | 0.6943 | 0.6592 | 0.9892 | 0.8751 | 0.3017 | 0.8559 | 0.3963 | 0.7190 | 0.8145 |
| 3D ShapeNets (Wu et al. 2015) | 0.2166 | 0.8832 | 0.4969 | 0.8482 | 0.7902 | 0.4888 | 0.2767 | 0.3239 | 0.1644 | 0.1068 |
| 3DWINN(ours) | **0.8114** | 0.9570 | 0.5723 | **0.9938** | **0.9055** | **0.9538** | **0.9820** | **0.9301** | **0.9477** | **0.9184** |

Table 5: Test accuracy of supervised classification on ModelNet10.

| Method | ModelNet10 |
|---|---|
| 3D ShapeNets (Wu et al. 2015) | 83.5% |
| DeepPano (Shi et al. 2015) | 85.5% |
| VoxNet (Maturana and Scherer 2015) | 92.0% |
| ORION (Sedaghat et al. 2016) | **93.8**% |
| Baseline | 93.1% |
| 3DWINN (ours) | 93.6% |

newly synthesized pseudo-negative examples will be used in later iterations to improve classification results.

**Unsupervised 3D Object Classification** We use a linear one-versus-all SVM to quantitatively evaluate the learned features by our unsupervised model trained on all ten object categories of ModelNet10. We use a L2 penalty with regularization penalty parameter $C = 0.5$. We train a linear SVM on top of the features extracted from the final convolution layer for classification and we find that the accuracy is on par with state-of-the-art results as shown in Table 6.

Table 6: Test accuracy of unsupervised classification on ModelNet10.

| Method | ModelNet10 |
|---|---|
| SPH (Kazhdan, Funkhouser, and Rusinkiewicz 2003) | 79.8% |
| LFD (Chen et al. 2003) | 79.9% |
| VConv-DAE (Sharma, Grau, and Fritz 2016) | 80.5% |
| 3D-GAN (Wu et al. 2016) | 91.0% |
| 3D-DescriptorNet (Xie et al. 2018a) | **92.4**% |
| 3DWINN (ours) | 91.9% |

## Robustness to Adversarial Examples

Following the method section, we demonstrate the robustness of 3DWINN against adversarial attacks. Note again that existing 3D object generators (Wu et al. 2015; Xie et al. 2018a) are not directly classifiers so no results on the adversarial attacks have been demonstrated in (Wu et al. 2015; Xie et al. 2018a). Here, we employ attacks generated by FGSM (Goodfellow, Shlens, and Szegedy 2015) and iterative FGSM (Kurakin, Goodfellow, and Bengio 2016) on supervised classification tasks. In FGSM method, we set the perturbation size to 0.005 instead of 0.125 in (Lee et al. 2018) to account for the binary nature of voxelized 3D adversarial examples. In iterative FGSM method, we set the perturbation size to 0.005 and iterate for 5 steps and 10 steps separately. The results in Table 7 show that 3DWINN reaches the significantly lower adversarial error and the higher correction rate compared to the baseline on ModelNet10 under both the FGSM and iterative FGSM attacks.

Table 7: Adversarial examples comparison between the baseline model and 3DWINN on ModelNet10. We adopt the same methodology used by (Lee et al. 2018) for evaluation. First, we generate $N$ adversarial examples from model A and count the number of adversarial examples misclassified by A ($= N_A$). **Adversarial error** of A is defined as test error rate against adversarial examples ($= N_A/N$). Secondly, among A's wrong predictions, we count the number of adversarial examples misclassified by B ($= N_{A\cap B}$). Then **correction rate** by B is $1 - N_{A\cap B}/N_A$. $\uparrow$ denotes higher is better; $\downarrow$ denotes lower is better.

| Adversarial Settings | FGSM | Iterative FGSM (5 steps) | Iterative FGSM (10 steps) |
|---|---|---|---|
| Adversarial error of **Baseline** $\uparrow$ | 27.64% | 30.50% | 53.19% |
| Adversarial error of **3DWINN** $\downarrow$ | **19.05**% | **19.27**% | **43.83**% |
| Correction rate by **Baseline** $\downarrow$ | 46.12% | 50.60% | 51.88% |
| Correction rate by **3DWINN** $\uparrow$ | **73.98**% | **70.06**% | **75.46**% |

## Conclusion

In this paper, we have presented a new 3D object modeling approach by extending the WINN method (Lee et al. 2018) from 2D images to 3D volumetric data. The proposed 3DWINN algorithm is applied to multiple 3D object modeling tasks with competitive results including 3D object generation, unsupervised object classification, supervised object classification, and adversarial attacks. Compared with the existing methods for 3D volumetric modeling, 3DWINN demonstrates its clear advantages on the standard supervised classification tasks while attaining comparable/better results

on the unsupervised problems. 3DWINN is general framework and is shown to be an effective approach modeling challenging volumetric 3D objects. The source code of this project will be made publicly available.

## Acknowledgement

This work is supported by NSF IIS-1618477 and NSF IIS-1717431. The authors thank Kwonjoon Lee for valuable discussions.

## References

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein generative adversarial networks. In *ICML*.

Ba, L. J.; Kiros, R.; and Hinton, G. E. 2016. Layer normalization. *CoRR* abs/1607.06450.

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, 223–232. Wiley Online Library.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*, 248–255.

Fan, H.; Su, H.; and Guibas, L. J. 2017. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, volume 2, 6.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014a. Generative adversarial nets. In *NIPS*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014b. Generative adversarial nets. In *NIPS*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and harnessing adversarial examples. In *ICLR*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*, 6626–6637.

Jin, L.; Lazarow, J.; and Tu, Z. 2017. Introspective classification with convolutional nets. In *NIPS*.

Karras, T.; Aila, T.; Laine, S.; and Lehtinen, J. 2018. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*.

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

Kingma, D., and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Kingma, D. P., and Welling, M. 2014. Auto-encoding variational bayes. In *ICLR*.

Krizhevsky, A. 2009. Learning Multiple Layers of Features from Tiny Images. *CS Dept., U Toronto, Tech. Rep.*

Kurakin, A.; Goodfellow, I.; and Bengio, S. 2016. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*.

Lazarow, J.; Jin, L.; and Tu, Z. 2017. Introspective neural networks for generative modeling. In *ICCV*.

LeCun, Y.; Boser, B.; Denker, J. S.; Henderson, D.; Howard, R.; Hubbard, W.; and Jackel, L. 1989. Backpropagation applied to handwritten zip code recognition. In *Neural Computation*.

Lee, K.; Xu, W.; Fan, F.; and Tu, Z. 2018. Wasserstein introspective neural networks. In *CVPR*.

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

McInerney, T., and Terzopoulos, D. 1996. Deformable models in medical image analysis: a survey. *Medical image analysis* 1(2):91–108.

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

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. *CVPR* 1(2):4.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, 5099–5108.

Radford, A.; Metz, L.; and Chintala, S. 2016. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*.

Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; and Chen, X. 2016. Improved techniques for training gans. In *NIPS*.

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

Sharma, A.; Grau, O.; and Fritz, M. 2016. Vconv-dae: Deep volumetric shape learning without object labels. In *ECCV*, 236–250.

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.

Simonyan, K., and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. In *ICLR*.

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

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *CVPR*.

Tu, Z. 2007. Learning generative models via discriminative approaches. In *CVPR*.

Welling, M., and Teh, Y. W. 2011. Bayesian learning via stochastic gradient langevin dynamics. In *ICML*.

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 *CVPR*, 1912–1920.

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 *NIPS*, 82–90.

Xie, J.; Zheng, Z.; Gao, R.; Wang, W.; Zhu, S.-C.; and Wu, Y. N. 2018a. Learning descriptor networks for 3d shape synthesis and analysis. In *CVPR*, 8629–8638.

Xie, S.; Liu, S.; Chen, Z.; and Tu, Z. 2018b. Attentional shapecontextnet for point cloud recognition. In *CVPR*, 4606–4615.

Xu, B.; Wang, N.; Chen, T.; and Li, M. 2015. Empirical evaluation of rectified activations in convolutional network. *CoRR* abs/1505.00853.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*.