The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Learning Generative Neural Networks for 3D Colorization

Zhenpei Yang
University of Texas at Austin
Austin, TX 78712, USA
zhenpei12@utexas.edu

Lihang Liu
University of Texas at Austin
Austin, TX 78712, USA
lihangliu@utexas.edu

Qixing Huang
University of Texas at Austin
Austin, TX 78712, USA
huangqx@cs.utexas.edu

Abstract

Automatic generation of 3D visual content is a fundamental problem that sits at the intersection of visual computing and artificial intelligence. So far, most existing works have focused on geometry synthesis. In contrast, advances in automatic synthesis of color information, which conveys rich semantic information of 3D geometry, remain rather limited. In this paper, we propose to learn a generative model that maps a latent color parameter space to a space of colorizations across a shape collection. The colorizations are diverse on each shape and consistent across the shape collection. We introduce an unsupervised approach for training this generative model and demonstrate its effectiveness across a wide range of categories. The key feature of our approach is that it only requires one colorization per shape in the training data, and utilizes a neural network to propagate the color information of other shapes to train the generative model for each particular shape. This characteristics makes our approach applicable to standard internet shape repositories.

Introduction

In this paper, we introduce an unsupervised method that trains a generative model to synthesize diverse and consistent color information across a shape collection (See Figure 1). Our work is motivated from recent advances in image colorization that train end-to-end neural networks for converting gray images into color images. However, our problem setting is different in several ways. First, we aim to generate diverse and consistent color content across a shape collection. In contrast, most existing image colorization techniques either can only synthesize a single color image for each input gray image or require that the training data contain multiple color images for each gray image for synthesizing diverse colorizations. Moreover, while each gray image possesses an effective regularization for the underlying color image, lacking such constraints on 3D geometry makes 3D colorizations considerably harder. Finally, unlike the breadth of resources available for 2D convolutional neural networks, 3D deep learning is a relatively under-explored field, and synthesizing appearance on surface geometry requires developing appropriate neural networks. The proposed neural network takes a shape and a latent color parameter as input and outputs a colorization of the input shape. By varying the latent parameters (e.g., the rows in Figure 1), we obtain diverse boundary-preserving colorizations for a single shape. Moreover, by varying the input shapes, we obtain consistent colorizations across a shape collection. We introduce a principled way to train this generative model from a training dataset that only offers a single colorization per shape (e.g., typical internet shape collections). The central idea is to design the generative model so that it takes a latent parameter and a shape as input to generate a 3D colorization, which can then be used to generate a colorization for a different shape. This network design enables us to train the generative model at each shape by implicitly propagating color information of neighboring shapes. We introduce a simple, yet effective formulation to train this generative model. The objective function consists of a regularization term and a data term. The regularization term constrains that latent color parameters of input shapes follow a prior distribution (e.g., the normal distribution). We formulate this term as minimizing the Earth-Mover distance (EMD) between the latent samples to an empirical distribution sampled from the prior distribution. We find that this strategy works remarkably well when the dimensionality of the parameter space is low (we use 5 for all the shape collections). The data term penalizes the difference between the synthesized colorizations and the real colorizations. The objective function can be easily optimized via alternating minimization, i.e., it alternates between optimizing the latent parameters and the network parameters. Each step leads to simple optimizations that can be solved very effectively. We have evaluated the proposed approach on five diverse categories collected from ShapeNet (Chang et al. 2015). We train a separate generative model for each category. The resulting network can synthesize diverse and consistent colorizations across each shape collection. We also compare the proposed approach with alternative approaches, demonstrating the effectiveness of the proposed network design and the training procedure.

Related Works

2D colorization. Generating colored digital content has been studied extensively in the 2D image domain. In terms of human interaction, early works on image colorization focused on interactive colorization (Levin, Lischinski, and

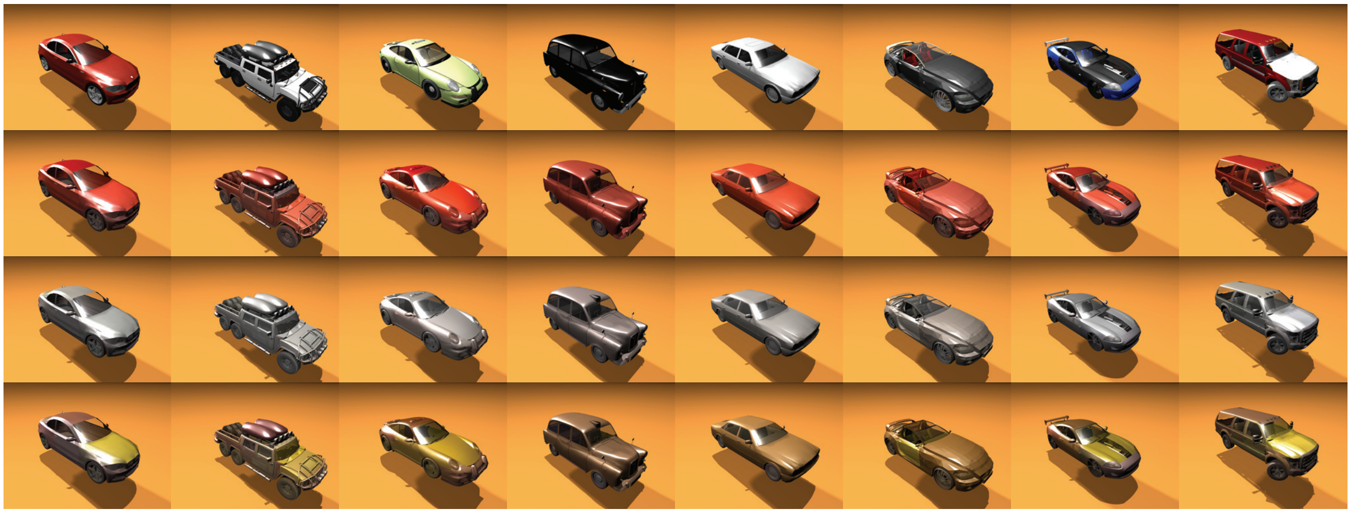

Figure 1: We introduce how to learn consistent and diverse colorizations across a shape collection. The first row shows the input color on each shape. The remaining rows show consistent (per row) and diverse (per column) colorizations obtained by varying a latent color parameter. Each row shares the same latent color parameter

Weiss 2004; Yatziv and Sapiro 2006; Pouli and Reinhard 2010), while more recent works have focused on automatic colorization (Cheng, Yang, and Sheng 2015). From a machine learning perspective, some methods are nonparametric (Welsh, Ashikhmin, and Mueller 2002). Other methods learn image colorization from large collections of data (Cheng, Yang, and Sheng 2015; Iizuka, Simo-Serra, and Ishikawa 2016). A fundamental challenge in colorization is that there may exist multiple plausible colorizations of one input image; this issue is partially addressed in recent works (Larsson, Maire, and Shakhnarovich 2016; Zhang, Isola, and Efros 2016; Deshpande et al. 2016) for the purpose of generating a single colorization. Specifically, (Larsson, Maire, and Shakhnarovich 2016; Zhang, Isola, and Efros 2016) trained convolutional neural networks to predict a color histogram for every pixel. (Deshpande et al. 2016) use a variational auto-encoder to model the conditional dependence between the input grayscale image and the color image. These methods, however, cannot generate consistent and boundary-preserving colorizations through sampling. In a recent work, (Zhang et al. 2017) tackle the diverse colorization generation problem by using a local-global-hint network. However, it requires large-scale training data, which potentially includes multiple colorizations of the same object. The problem studied in this paper differs from existing image colorization problems in that we want to train a network that predicts diverse colorizations for each individual shape, despite the fact that the training data only provides a single colorization for each one. Moreover, since the number of training instances we have for 3D models is significantly smaller than that in the image domain, we found that popular training methods (e.g., conditional GAN), which work well in the image domain, did not apply well to the 3D domain. **Generative modeling.** In a broader picture, our work is related to recent advances in generative modeling using neural networks. Popular generative modeling techniques include generative adversarial networks (GAN) (Goodfellow et al. 2014; Zhao, Mathieu, and LeCun 2016; Arjovsky, Chintala, and Bottou 2017), variational autoencoders (VAE) (Kingma and Welling 2013; Kingma, Salimans, and Welling 2016), and autoregression (Van Den Oord, Kalchbrenner, and Kavukcuoglu 2016). (Larsen et al. 2015) combine VAE with GAN to jointly learn a distance measure. The difference in our approach is that we drop the encoder by using an explicit latent coding assignment, and we use Earth-Mover distance as the distance measure. Besides colorization, people have applied generative models to other domains, such as image synthesis (Goodfellow et al. 2014; Arjovsky, Chintala, and Bottou 2017) and voxel-based 3D model synthesis (Wu et al. 2016). Our problem is significantly harder since we are learning a higher dimensional conditional distribution with small-scale training data.

3D colorization. The standard technique for associating a 3D model with color and texture is texture mapping (Blinn and Newell 1976; Lévy 2001). These approaches require predefined texture patterns, and thus are not suitable for the task of synthesizing new colorizations of 3D shapes. Other colorization techniques utilize user constraints. (Leifman and Tal 2012) colorize a 3D mesh model by propagating user input colors throughout the mesh. (Chajdas, Lefebvre, and Stamminger 2010) develop an algorithm to assist users in assigning textures to scenes. More recent work focuses on data-driven colorization. (Jain et al. 2012) provide color suggestions for man-made shapes by utilizing a database of 3D shapes. Their method builds upon part-based shape representations and infers part colors by leveraging shape similarity and color compatibility between adjacent parts. (Chen et al. 2015) optimize color suggestions for 3D scenes that satisfy user constraints and maximize an aesthetic score. None

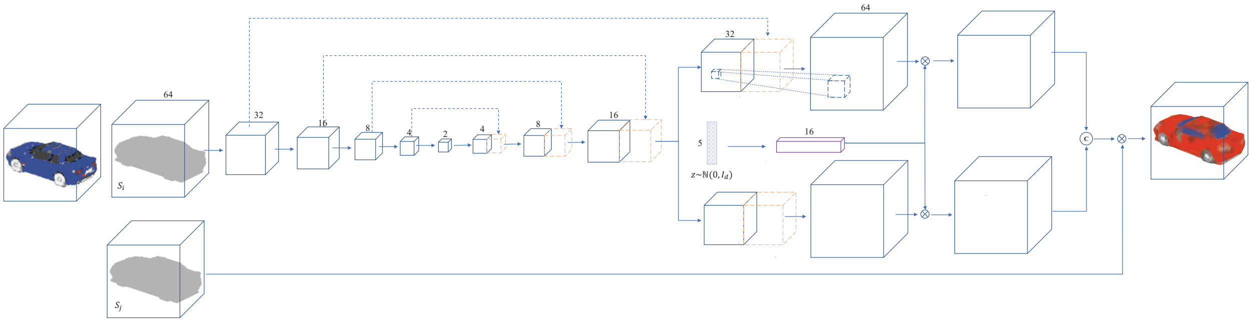

Figure 2: *Network Architecture.* This figure shows the network architecture used in our method. The network consists of three components. The first component takes the shape as input and outputs a compressed volume feature. The second component passes this compressed feature through several deconvolutional layers and incorporates the information from a latent z vector to generate a voxel-based colorization. The final component applies a mask to derive a colorization that corresponds to that mask.

of these works, however, are able to synthesize colorizations in a controlled and consistent manner.

Approach

In this section, we describe the technical details of the proposed approach for learning a consistent generative colorization model across a shape collection.

Approach Overview

Suppose we have a collection of colored 3D models $\{(C_1, S_1), \cdots, (C_n, S_n)\}$, where C_i and S_i denote the color and geometry information of the i-th shape, respectively. Our goal is to train a generative neural network that takes a shape S and a latent vector $\boldsymbol{z} \in \mathbb{R}^d$ as input and outputs its colorization C. In this paper, we employ a volumetric representation of 3D models and parameterize the generative neural network as $G_\theta(S^{\text{inp}}, \boldsymbol{z}, S^{\text{mask}})$. Here θ denotes the network parameters; S^{inp} denotes a volumetric representation of the input shape; $\boldsymbol{z}$ is a latent vector that controls the output; S^{mask} is a volumetric mask that determines the output shape and the synthesized colorization. As we will see later, introducing a separate mask shape S^{mask} gives us more flexibility to train the generative neural network. In particular, it allows us to propagate color information from similar shapes to train the generative model for each individual shape, addressing the issue of having a single color configuration per model in the training data. Note that in the inference stage, we always call $G_\theta(S^{\text{inp}}, \boldsymbol{z}, S^{\text{inp}})$ to obtain colorization results. To learn the generative model, we propose to utilize a classification loss function $\mathcal{D}(\cdot, \cdot)$ for comparing two colorizations. This is conceptually similar to utilizing a pre-trained perceptual distance metric (Dosovitskiy and Brox 2016) for image synthesis, except that our focus here is on color synthesis. Instead of comparing real data and synthetic data in the object space, we propose to compare them in the original space. This is done by measuring the Earth-Mover distance between the latent parameters of the input shapes to an empirical distribution sampled from the prior distribution. As we will see later, a key advantage of this formulation is that the objective function naturally decomposes into ancillary optimization problems that are easy to solve when optimizing the latent parameters. With this setup, we introduce a two-term objective function for learning the generative network:

$$\min_{\theta, \mathcal{Z}} \; E_{\text{data}}(\theta, \mathcal{Z}) + E_{\text{regu}}(\mathcal{Z}) \tag{1}$$

Here $\mathcal{Z} = \{\boldsymbol{z}_1, \cdots, \boldsymbol{z}_n\}$ collects the latent parameters of the input shape. The data term $E_{\text{data}}(\theta, \mathcal{Z})$ measures the similarity between the input colorizations and the synthesized ones. The regularization term $E_{\text{regu}}(\mathcal{Z})$ enforces that the empirical distribution specified by $\mathcal{Z}$ is aligned with the prior distribution of $\boldsymbol{z}$. In the remainder of this section, we describe these two objective terms, the network design, and how to optimize (1) in detail.

Network Architecture

Figure 2 illustrates the generative network architecture used in this paper. The network is adapted from the network used for 3D shape synthesis (Isola et al. 2016; Wu et al. 2016), with novel designs tailored for our problems. The input to our network is a binary voxel grid of dimension $64 \times 64 \times 64 \times 1$, representing the input shape S_i. S_i is passed through five 3D convolution layers with a down-sampling factor of 2, resulting in an intermediate layer L_m of size $2 \times 2 \times 2$. This intermediate encoding then passes through 3 deconvolution layers. Afterwards the network splits into two branches that predicts l channel and ab channel respectively. We use a $d = 5$ dimension vector $\boldsymbol{z}$ to encode the latent color parameter. $\boldsymbol{z}$ passes through a fully connected layer to enlarge its dimension and is then multiplied by each of the hyper-column features of the second-to-last layer. The product is then passed through the last convolutional layer. The two branches are concatenated together to form a $64 \times 64 \times 64 \times 3$ dimension color grid and then masked by shape S_j to get the final color voxel model. We incorporate skip layers between the corresponding convolution and deconvolution layers in order to compensate the spatial information loss during down-sampling. All the convolution layers have kernel size 3 × 3 × 3. We also use batch normalization (Ioffe and Szegedy 2015) to stabilize the training.

Data Term

The data term aligns the synthesized colorizations and the input ones with respect to the underlying color distance metric $\mathcal{D}(\cdot,\cdot)$. In the following we will first describe how to formulate the data term when $\mathcal{D}(\cdot,\cdot)$ is given. We then describe the formulation of $\mathcal{D}(\cdot,\cdot)$.

Colorization propagation. A naive way to formulate the data term is to add the difference between the synthesized colorization and the given colorization of each shape in the training data:

$$\hat{E}_{\text{data}}(\theta,\mathcal{Z}) = \sum_{i=1}^{n} \mathcal{D}\big(C_i, G_\theta(S_i, \boldsymbol{z}_i, S_i)\big). \qquad (2)$$

In our implementation, we found that (2) did not work well. The main reason is that the training data only contains one colorization per model, which is insufficient for learning to synthesize multiple colorizations. To address this issue, we modify the basic formulation described in (2) such that colorizations of other shapes contribute to the color synthesis of each individual shape, with contributions being weighted by shape distances:

$$\hat{E}_{\text{data}}(\mathcal{D},\theta,\mathcal{Z}) = \sum_{1\le i,j\le n} w_{ij}\mathcal{D}\big(C_j, G_\theta(S_i, \boldsymbol{z}_j, S_j)\big), \quad (3)$$

where

$$w_{ij} = \exp\Big(-\frac{d^2(S_i,S_j)}{2\sigma^2}\Big), \qquad \sigma = \underset{1\le i,j\le n}{\text{median}}\ d(S_i,S_j),$$

and the shape distance metric is given by the Hausdorff distance between two shapes. Note that this simple formulation is made possible for the network design, where we replace the mask by different shapes for the purpose of propagating color information.

Classification loss. Popular choices for $\mathcal{D}(\cdot,\cdot)$ include regression loss (e.g., see (Larsson, Maire, and Shakhnarovich 2016)) and classification loss (e.g., see (Zhang, Isola, and Efros 2016)). We have tried both options and found that classification loss generally leads to better results. On the other hand, the cost we pay when utilizing classification loss is that the memory cost scales linearly with the number of classes. To address this issue, instead of clustering LAB values directly to obtain discrete color classes, we first run K-means clustering on the L channel to obtain 10 classes. We then run K-means clustering on the AB channel values to obtain 10 more classes, totaling to 100 representative clusters in the original LAB space. The loss function is thus the sum of two loss functions:

$$\mathcal{D}(\cdot,\cdot) = \mathcal{D}_l(\cdot,\cdot) + \mathcal{D}_{ab}(\cdot,\cdot). \qquad (4)$$

Each term $\mathcal{D}_t(\cdot,\cdot), t \in \{l, ab\}$ counts the misclassified voxels. Since the color distribution is highly imbalanced, we use the color-frequency to re-weight the misclassifications:

$$\mathcal{D}_t(\overline{C}, C) = \sum_{1\le i,j,k\le 64} \frac{1}{\gamma f(c_{ijk}) + (1-\gamma) f_{uni}} \delta(\overline{c}_{ijk} \neq c_{ijk})$$

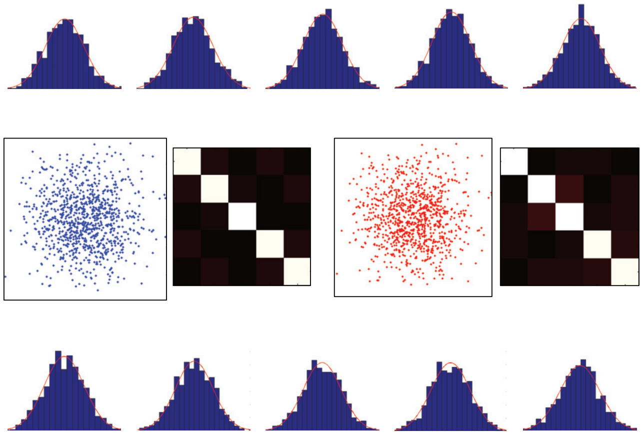

Figure 3: *Latent parameters.* The first row and the first two subfigures of the second row show the marginal distributions, plot of the first two dimensions, and the covariance matrix of the latent parameters $\mathcal{Z}$, respectively. The last two figures of the second row and the third row show those of $\overline{\mathcal{Z}}$.

where $t \in \{l, ab\}$, $\overline{c}_{ijk}$ and c_{ijk} denote the predicted class and the ground-truth of the ijk-th cell of the predicted volumetric coloring $\overline{C}$ and the ground-truth C. $f(c_{ijk})$ is the normalized frequency of class c_{ijk} (i.e., $f(c_{ijk}) = 0.1$ if the color distribution is uniform). f_{uni} is the uniform distribution (i.e., f_{uni} = 0.1). We set $\gamma = 0.5$ in our experiments.

Prior Term

The prior term ensures that the latent variables in $\mathcal{Z}$ follow the normal distribution $\mathcal{N}(0, I_d)$. Since we employ a relatively low dimensional parameter space ($d = 5$ in this paper), we introduce a simple but effective formulation, i.e., we first take n samples $\{\overline{\boldsymbol{z}}_1, \cdots, \overline{\boldsymbol{z}}_n\}$ from the normal distribution $\mathcal{N}(0, I_d)$. We then formulate the prior term as

$$E_{\text{regu}}(\mathcal{Z}) = \mu \min_{p\in\mathcal{P}_n} \sum_{i=1}^{n} \|\boldsymbol{z}_i - \overline{\boldsymbol{z}}_{p(i)}\|^2, \qquad (5)$$

where $\mathcal{P}_n$ is the space of permutations of order n. In other words, the prior term minimizes the Earth-Mover distance between $\mathcal{Z}$ and $\overline{\mathcal{Z}} = \{\overline{\boldsymbol{z}}_1, \cdots, \overline{\boldsymbol{z}}_n\}$ under the L^2 norm. μ balances the prior term and the data term. In this paper, we choose $\mu = 1$ for all of our experiments. (5) is motivated from the VAE-GAN (Larsen et al. 2015), which aligns distributions in the parameter space, and WGAN (Arjovsky, Chintala, and Bottou 2017), which demonstrates that the EMD metric is superior to KL-divergence for comparing distributions. Figure 3 compares the marginal distributions of $\overline{\mathcal{Z}}$ and the optimized $\mathcal{Z}$ on the car category. We can see that they are comparable with regards to the approximation power of the underlying normal distribution.

Optimization

Substituting (5) and (3) into (1), we arrive at the following objective function for learning the generative network for

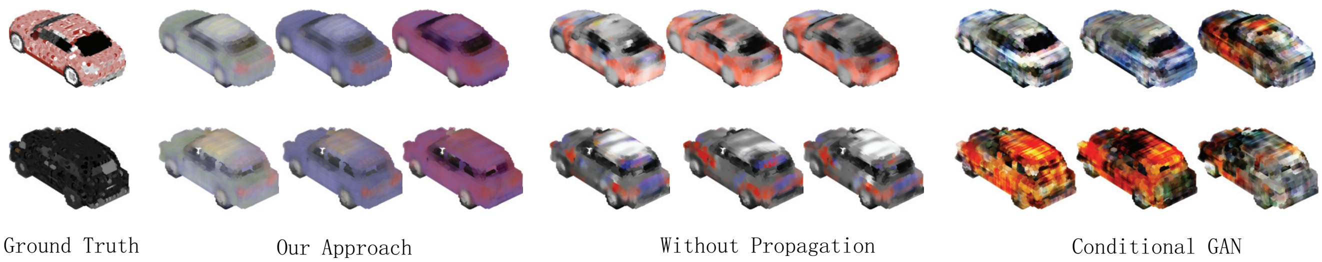

Figure 4: *Visual comparison in volumetric representation.* In this figure, we show the visual comparison between the synthesis voxels of our method and other methods. For each model, we show the original model, as well as the colorization of each method.

colorization:

$$\min_{\theta,\mathcal{Z}} \sum_{1\le i,j\le n} w_{ij}\mathcal{D}(C_j, G_\theta(S_i,\boldsymbol{z}_j,S_j)) + \mu \min_{p\in\mathcal{P}_n}\sum_{i=1}^{n}\|\boldsymbol{z}_i - \overline{\boldsymbol{z}}_{p(i)}\|^2 \tag{6}$$

Since the network parameters θ, the latent parameters $\mathcal{Z}$, and the permutation p are naturally decoupled in (6), we apply alternating minimization to solve (6), i.e., each step of the alternating minimization optimizes one group of variables while fixing the other group of variables. Since (6) is highly non-convex, we need to initialize the variables properly. In our implementation, we first initialize the latent parameters $\mathcal{Z}$ and the permutation p. We then alternate between optimizing (θ, $\mathcal{Z}$) and p. The rest of this section provides the details.

Latent parameter initialization. A good initialization should result in input shapes with different appearances being distant from each other and shapes with similar appearances being close to each other within the latent parameter space, respectively. In our implementation, we initialize the latent parameters by solving a quadratic assignment problem:

$$\min_{p\in\mathcal{P}_n} \sum_{1\le i<j\le n} (d_{ij} - \|\boldsymbol{z}_{p(i)} - \boldsymbol{z}_{p(j)}\|)^2 \tag{7}$$

where $d_{ij} = \|C_i - C_j\|$ is the color difference between the volumetric representations of S_i and S_j. Before solving (7), we scale d_{ij} so that $\{d_{ij}, 1 \le i < j \le n\}$ and $\{\|\boldsymbol{z}_i - \boldsymbol{z}_j\|, 1 \le i < j \le n\}$ have the same mean. Solving (7) exactly is known to be NP-hard. We therefore adapt the spectral relaxation of MAP inference described in (Leordeanu and Hebert 2006), which delivers satisfactory approximate solutions to (7) in our experiments.

Network parameter optimization. It is easy to see that optimizing the network parameter amounts to solving the following optimization problem

$$\min_{\mathcal{Z}} \sum_{1\le i\le j\le n} w_{ij}\mathcal{D}\big(C_j, G_\theta(S_i,\boldsymbol{z}_j,S_j)\big) \tag{8}$$

(8) becomes the standard network regression problem and we apply stochastic gradient descent for optimization.

Latent parameter optimization. When optimizing the latent parameters $\mathcal{Z}$, (6) reduces to

$$\min_{\mathcal{Z}} \sum_{(i,i')\in\mathcal{G}} w_{ij}\mathcal{D}\big(C_j, G_\theta(S_i,\boldsymbol{z}_j,S_j)\big) + \mu\sum_{i=1}^{n}\|\boldsymbol{z}_i - \overline{\boldsymbol{z}}_{p(i)}\|^2$$

We employ stochastic gradient descent with the Adam optimizer for optimization (Kingma and Ba 2014).

Permutation optimization. When optimizing the permutation ϕ with other variables being fixed, (6) reduces to

$$\min_{p\in\mathcal{P}_n}\sum_{i=1}^{n}\|\boldsymbol{z}_i - \overline{\boldsymbol{z}}_{p(i)}\|^2 \quad\Leftrightarrow\quad \max_{p\in\mathcal{P}_n}\sum_{i=1}^{n}\boldsymbol{z}_i^T\overline{\boldsymbol{z}}_{p(i)} \tag{9}$$

In other words, optimizing the permutation is equivalent to solving an assignment problem. In our implementation, we use the Hungarian algorithm to solve this, which is sufficiently tractable for all the datasets used in this paper.

Convergence of alternating minimization. In our implementation, we found that the alternating optimization usually converges in 500-700 iterations (i.e., the training error becomes stable).

Experimental Evaluation

In this section, we provide an experimental evaluation of the proposed colorization method.

Experimental Setup

	n	$\|\sigma_{\text{geo}}\|$	$\|\sigma_{\text{color}}\|$	t_{train}
Car	400	2.3e5	3.2e5	15h
Airplane	400	1.3e5	2.3e5	15h
Chair	400	3.4e5	8.9e5	15h
Table	400	3.1e5	7.2e5	15h
Motorbike	213	2.0e5	2.3e5	14h

Table 1: Statistics of the datasets used in this paper. From left to right: n: number of models; $\|\sigma_{geo}\|$: norm of the geometric variance; $\|\sigma_{color}\|$: norm of the color variance; t_{train}: training time;

Dataset. We evaluated our approach on 5 representative categories of ShapeNetCore (Chang et al. 2015) (See Table 1). The models on ShapeNetCore come from 3D Warehouse [1], and thus are associated with part information and color information. For each category, we manually select models that have rich color information(400 models for each category). We reserve 25% of these models as testing data. Table 1 provides the data statistics. In particular, we provide

[1] https://3dwarehouse.sketchup.com/

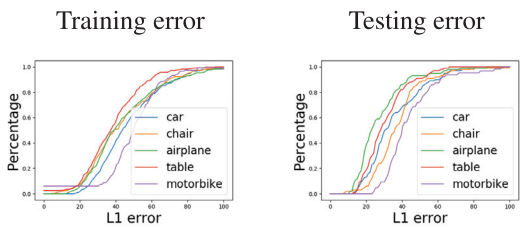

Figure 5: This figure shows the reconstruction errors of our method on the training and testing datasets of each category.

two measures, $\|\sigma_{geo}\|$ and $\|\sigma_{color}\|$, to show the variance of each category. Here, σ_{geo} is a covariance matrix of the vectorized voxel representation of the shape, and σ_{color} is that of the corresponding color channel. Their spectral norms provide an assessment of the variance of the training data.

Baseline Comparison. We consider two alternative approaches for baseline comparison:

- **Conditional WGAN.** The first baseline is conditional GAN (Mirza and Osindero 2014), which has been adapted in recent state-of-the-art image colorization techniques (See e.g., (Zhang, Isola, and Efros 2016)). In our experiments, we enhance the original formulation of Conditional GAN with the Wasserstein distance measure (Arjovsky, Chintala, and Bottou 2017), leading to a strong baseline. This baseline is used to demonstrate the ability of our method for synthesizing diverse colorizations for each shape.
- **Without propagation.** The second baseline is the modified training objective function (2), where we do not propagate the color information across different shapes. This network is used to show that propagating color information across adjacent shapes is crucial for training a generative model that can synthesize diverse colorizations.

Evaluation protocols. We conduct both qualitative and quantitative evaluations. Qualitatively, we output 6 colorization results for each method and compare the results visually in Figure 4. Quantitatively, we compare the reconstruction errors of each method on both training and testing datasets. For the reconstruction error of the training set, we simply compute the distance between the ground-truth color grid and generated color grid using corresponding latent parameters. For the reconstruction error of the testing set, We find the optimal latent parameter for each input model and compute the discrepancy. Specifically, for each model S_j in the test set and each trained neural network $G_\theta(S_j, \boldsymbol{z}, S_j)$, we measure the reconstruction error with respect to the optimal latent parameter $\boldsymbol{z}^\star$:

$$\boldsymbol{z}^\star = \min_{\boldsymbol{z}} \mathcal{D}(C_j, G_\theta(S_j, \boldsymbol{z}, S_j)). \tag{10}$$

The test set reconstruction error provides an assessment of the generalization behavior of a generative network.

Analysis of Results

Figure 1 and the supplemental material illustrate representative results of the proposed technique. Each result is interpreted as a table, where the rows index through different shapes and the columns index through different latent parameters. They are obtained by aggregating the color information within each object part and rendered using *povray*. Voxel-level results can be seen in Figure 4. Overall, the colorizations are fairly comparable to those in the training data. The synthesized colorizations are also diverse — they are generally different from the original colorization associated with each shape. It is worth to note that, as can be seen from Figure 4, our method can produce consistent boundaries when varying the latent coding z whereas other methods fail. This feature is critical in the sense that we want the latent coding to interpret semantic colors. Moreover, the colorizations are fairly consistent among shapes in each shape collection. We can obtain similar colorization by applying the same latent coding z to all models. This justifies the design purpose of the proposed approach, i.e., generating consistent colorizations across a shape collection.

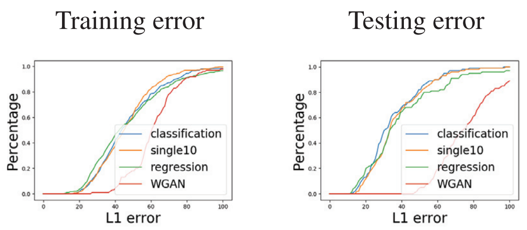

Figure 6: *Loss function comparison.* We compare the proposed classification loss with alternative loss functions. The proposed loss function leads to the closest reproduction result compared to utilizing other loss functions.

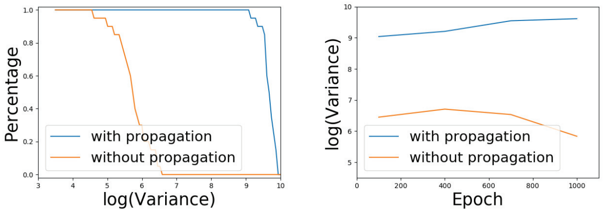

Figure 7: *Color variance* We illustrate the importance of color propagation by comparing the color variance of generated models. Using proposed propagation leads to much more diverse color generation.

To compare the colorizations across different categories, we also plotted the reconstruction errors of the participating categories (See Figure 5) on both the training and testing sets. The colorizations exhibit similar reconstruction errors across different groups, which suggest that the proposed approach is fairly robust across different categories.

Further Analysis of the Proposed Approach

Baseline comparison. We have compared the proposed approach with two baseline approaches, namely, conditional WGAN and the variant of our method that does not propagate the color information across adjacent shapes. As shown in Figure 4, our approach is far better than conditional

WGAN and the 'without propagation' variant. Specifically, our approach is the only approach among the three that can generate diverse and consistent colorizations. The colorizations also nicely capture the semantic features of the underlying objects, including wheels, windows, and front and back lights. In contrast, neither conditional WGAN nor 'without-propagation' can generate diverse and realistic results. Moreover, the visual quality of their results are significantly worse than ours, containing salient high-frequency noise. This is a sign of overfitting, which is incurred by limited training data from 3D shapes both in terms of database size and the fact that there is only one colorization per shape. We demonstrate the effectiveness of color propagation from similar shapes in Figure 7. We can clearly see that color propagation enables the generative model to produce diverse colorization.

We also investigated the effectiveness of information passed through the encoder network. We tried to remove the encoder part, and masked the output color grid of decoder with S_i to get colorization C_i. This approach turned out to be very hard to train, and we were not able to get reasonable results using this approach. Our explanation is that different types of objects within the same category compete against each other. For example, car has multiple types(e.g. Sedan, SUV, Truck), and the output should be dependent on such information.

We conducted two user studies that respectively examine the coloriziation quality and diversity. For each test, we asked each of the 20 participants 5 questions. To test the colorization quality, each question contained a set of three images that include results from our approach, our approach without propagation, and conditonal WGAN. Participants were required to rank the three images according to their visual quality. For the diversity test, each question contained 9 colorization results (3 for each method) of 3 models. Participants were required to rank different methods according to the diversity of generated colorizations. The survey shows that 79% of the time our generated images were ranked as the most visually-appealing colorization, whereas 'without propagation' and conditional WGAN were 15% and 6%, respectively. The diversity test shows that our method was the best among 73% of the instances, while 'without propagatio' and'conditional WGAN' were 12% and 15%, respectively.

Loss function. We next compare the classification loss employed in this paper with two alternative loss functions:

- Single-10: We use a single classification loss to train the neural network. The original continuous colors were discretized into 10 classes using K-means clustering.
- Regression loss: We minimize $\|C - G_\theta(S, \boldsymbol{z}, S)\|_{\mathcal{F}}^2$.

Figure 6 compares the proposed classification loss used in this paper, the two loss functions described above, and conditonal WGAN. We can see that employing separate loss functions on L and AB channels leads to more fine-grained results than Single-10. This is not surprising since the proposed approach utilizes a greater number of latent color classes. Moreover, both classification loss functions are superior to the regression loss in terms of testing error. This behavior is consistent with the results on image colorization, where the classification loss performs better than the regression loss (c.f. (Zhang, Isola, and Efros 2016)). In addition, all three methods are significantly better than conditional WGAN, which indicates the advantage of our training framework when the size of the training data is small.

fferent layers. we can see that the learned neurons capture the semantics of the underlying model, e.g., part structures, and correlations among the parts (the back and front of the cars tend to have simiar colors, and the wheels have the similar colors, too).

Conclusions and Future Work

In this paper, we have introduced a novel approach for colorization synthesis on 3D models. Instead of generating a single colorization configuration per 3D model, our approach learns a generative model that maps a prior distribution of a latent colorization parameter space to the space of colorization configurations on each model. This enables us to generate a diverse set of colorizations that are consistent across a shape collection. We showed a principled approach for learning such a generative model from datasets in which each shape only possesses a single colorization configuration. The usefulness of this generative model is demonstrated both experimentally on benchmark datasets as well as in the application of colorization interpolation.

The presented approach can be generalized in many ways. Two natural extensions include material synthesis for 3D scenes as well as texture synthesis on 3D geometry. Moreover, although the learning framework proposed in this paper can be applied in both cases, the voxel representation employed in this paper would be inadequate. The technical challenge is thus to design suitable geometric representations for these two tasks. Finally, it would be interesting to learn generative models by transferring colorization and texture information from images. This could potentially address both the quality and scalability issues of existing 3D datasets. One potential formulation is to learn the generative model so that the distribution of projected images match that of the input images. However, the major challenge in this case still lies in designing suitable data representations for 3D geometry.

References

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein GAN. *CoRR* abs/1701.07875.

Blinn, J. F., and Newell, M. E. 1976. Texture and reflection in computer generated images. *Communications of the ACM* 19(10):542–547.

Chajdas, M. G.; Lefebvre, S.; and Stamminger, M. 2010. Assisted texture assignment. In *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D '10, 173–179. New York, NY, USA: ACM.

Chang, A. X.; Funkhouser, T. A.; Guibas, L. J.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; Xiao, J.; Yi, L.; and Yu, F. 2015. Shapenet: An information-rich 3d model repository. *CoRR* abs/1512.03012:1–10.

Chen, K.; Xu, K.; Yu, Y.; Wang, T.-Y.; and Hu, S.-M. 2015. Magic decorator: automatic material suggestion for indoor digital scenes. *ACM Transactions on Graphics (TOG)* 34(6):232.

Cheng, Z.; Yang, Q.; and Sheng, B. 2015. Deep colorization. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, 415–423. Washington, DC, USA: IEEE Computer Society.

Deshpande, A.; Lu, J.; Yeh, M.-C.; and Forsyth, D. 2016. Learning diverse image colorization. *arXiv preprint arXiv:1612.01958*.

Dosovitskiy, A., and Brox, T. 2016. Generating images with perceptual similarity metrics based on deep networks. In *Advances in Neural Information Processing Systems (NIPS)*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In Ghahramani, Z.; Welling, M.; Cortes, C.; Lawrence, N. D.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems 27*. Red Hook, NY: Curran Associates, Inc. 2672–2680.

Iizuka, S.; Simo-Serra, E.; and Ishikawa, H. 2016. Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Transactions on Graphics (TOG)* 35(4):110.

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. 37:448–456.

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2016. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*.

Jain, A.; Thormählen, T.; Ritschel, T.; and Seidel, H.-P. 2012. Material memex: Automatic material suggestions for 3d objects. *ACM Transactions on Graphics (TOG)* 31(6):143.

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. *CoRR* abs/1312.6114.

Kingma, D. P.; Salimans, T.; and Welling, M. 2016. Improving variational inference with inverse autoregressive flow. *CoRR* abs/1606.04934.

Larsen, A. B. L.; Sønderby, S. K.; Larochelle, H.; and Winther, O. 2015. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*.

Larsson, G.; Maire, M.; and Shakhnarovich, G. 2016. Learning representations for automatic colorization. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, volume 9908 of *Lecture Notes in Computer Science*, 577–593. Berlin, Germany: Springer.

Leifman, G., and Tal, A. 2012. Mesh colorization. *Comput. Graph. Forum* 31(2pt2):421–430.

Leordeanu, M., and Hebert, M. 2006. Efficient map approximation for dense energy functions. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, 545–552. New York, NY, USA: ACM.

Levin, A.; Lischinski, D.; and Weiss, Y. 2004. Colorization using optimization. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, 689–694. New York, NY, USA: ACM.

Lévy, B. 2001. Constrailevin2004colorizationned texture mapping for polygonal meshes. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, 417–424. New York, NY, USA: ACM.

Mirza, M., and Osindero, S. 2014. Conditional generative adversarial nets. *CoRR* abs/1411.1784.

Pouli, T., and Reinhard, E. 2010. Progressive histogram reshaping for creative color transfer and tone reproduction. In *Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering*, NPAR '10, 81–90. New York, NY, USA: ACM.

Van Den Oord, A.; Kalchbrenner, N.; and Kavukcuoglu, K. 2016. Pixel recurrent neural networks. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, 1747–1756. JMLR.org.

Welsh, T.; Ashikhmin, M.; and Mueller, K. 2002. Transferring color to greyscale images. *ACM Trans. Graph.* 21(3):277–280.

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 Lee, D. D.; Sugiyama, M.; Luxburg, U. V.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 29*. Red Hook, NY: Curran Associates, Inc. 82–90.

Yatziv, L., and Sapiro, G. 2006. Fast image and video colorization using chrominance blending. *IEEE Transactions on Image Processing* 15(5):1120–1129.

Zhang, R.; Zhu, J.-Y.; Isola, P.; Geng, X.; Lin, A. S.; Yu, T.; and Efros, A. A. 2017. Real-time user-guided image colorization with learned deep priors. *arXiv preprint arXiv:1705.02999*.

Zhang, R.; Isola, P.; and Efros, A. A. 2016. Colorful image colorization. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III*, volume 9907 of *Lecture Notes in Computer Science*, 649–666. Berlin, Germany: Springer.

Zhao, J. J.; Mathieu, M.; and LeCun, Y. 2016. Energy-based generative adversarial network. *CoRR* abs/1609.03126.