Image-Driven Furniture Style for Interactive 3D Scene Modeling

Tomer Weiss[1] (iD)   Ilkay Yildiz[2] (iD)   Nitin Agarwal[3] (iD)   Esra Ataer-Cansizoglu[4,5] (iD)   Jae-Woo Choi[5] (iD)

[1]New Jersey Institute of Technology   [2]Northeastern University   [3]University of California, Irvine   [4]Facebook Inc.   [5]Wayfair Inc.
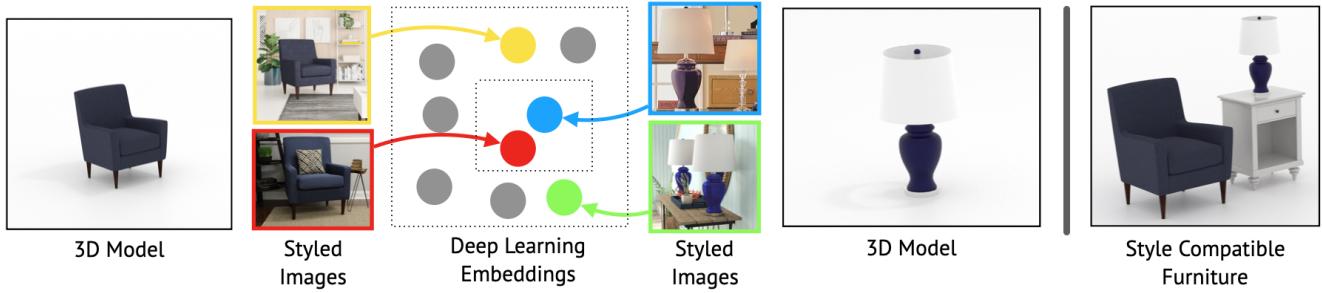
Figure 1: *We learn 3D furniture model style using a deep neural network that was trained on a data set of annotated curated styled images. Using the embeddings from the network, we estimate a 3D models's style compatibility with other models. We demonstrate our approach with an interactive system to create styled interior scenes.*

Abstract

Creating realistic styled spaces is a complex task, which involves design know-how for what furniture pieces go well together. Interior style follows abstract rules involving color, geometry and other visual elements. Following such rules, users manually select similar-style items from large repositories of 3D furniture models, a process which is both laborious and time-consuming. We propose a method for fast-tracking style-similarity tasks, by learning a furniture's style-compatibility from interior scene images. Such images contain more style information than images depicting single furniture. To understand style, we train a deep learning network on a classification task. Based on image embeddings extracted from our network, we measure stylistic compatibility of furniture. We demonstrate our method with several 3D model style-compatibility results, and with an interactive system for modeling style-consistent scenes.

1. Introduction

With the increasing demand for realistic virtual worlds, methods for effectively creating virtual scenes is greater than ever. Such demand is driven by multiple industries: high-fidelity gaming, visual effects and even retail, to name a few. For example, a realistic game, or scene in a movie, might need to convey a certain style, appropriate for the story, or time-period. Similarly, home goods and retail companies rely on virtual scenes for demonstrating their products. When creating such scene, a user selects objects from a catalog of 3D models that fits their theme, and scene style. Selecting 3D models that fit well together is necessary for creating appealing and consistent visual quality.

In this work, we propose a method for inferring stylistic compatibly of 3D furniture models, using style context images. We focus on a set of furniture classes that are prevalent in interior spaces. Contrary to previous approaches (c.f. Section 2) that mostly focus

on learning style from geometry, or rendered shots of 3D models, we decouple a 3D model's geometry by learning directly from images of styled scenes, since multiple elements influence style, such as color, texture, material, and the use of the space. Styled interior images, judged by professionals, contain such elements. Hence, we learn a 3D model's style based on images of furniture in a curated settings.

To learn style, we first need to define what "style" is. Interior designers define multiple style guidelines. Typically, each style is described with certain criteria about fabric, color scheme, material, use of space, light, texture, and flooring [KK14]. In this work, we focus on 4 major highly popular styles: *modern, traditional, cottage and coastal*, as shown in Figure 2. Associating a furniture, or scene with a certain style is a challenging and subjective task, since styles are loosely defined, and smoothly change with trends in the field. Hence, we employ a data-driven approach to estimate

style. We collect a data set of furniture and scene images that are annotated with styles by multiple interior style experts. Given such an annotated data set, we train a deep siamese network [BGL*94] based on the VGG16 architecture [SZ14], to classify an image's style. Since images in our data set depict furniture, by estimating an image's style, we estimate a furniture's style. We evaluate our network's performance on several scenarios, showing that it successfully learns the style spectrum of each image, and improves the classification accuracy compared to other baselines.

We utilze embeddings of the neural network to infer style-compatibility. Once our network is trained, we extract the embeddings for each image. Embeddings encode the low-level style features. Computing the euclidean distance between the embeddings of two images gives us a measure of their stylistic compatibility, e.g., the smaller such distance is, the more style-compatible images are. Since the images are also associated with 3D furniture models, we use image embeddings to predict the stylistic-compatibility between two furniture models.

We perform extensive evaluation experiments on our style compatibility framework, both qualitative and quantitative, on furniture images and 3D models. To demonstrate our approach, we implemented a tool for interactive, style-compatible scene building. In summary, our contributions are:

1. Learning a furniture's style-compatibility from styled images. Styled images contain more information than renderings of pure, isolated 3D models.
2. Our style estimation embodies texture, color and materials, which are not considered in previous geometry-focused work.
3. A novel comparison-based approach for learning style from images, which improves the classification accuracy over discrete style labels.
4. An interactive system for style-compatible modeling of virtual scenes.
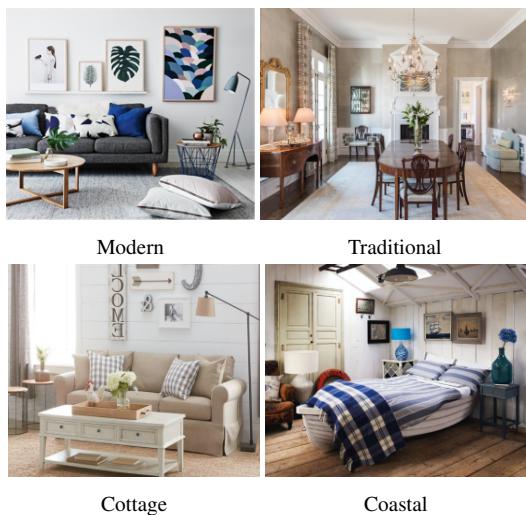
Figure 2: *Major interior styles. Each image in our data set is labeled with such styles, to be used by our deep learning style estimation network.*

2. Related Work

Real, and virtual scenes are often designed in a certain style. Understanding such style is difficult, as evident by numerous interior design professionals offering their services. Organizations, such as Ikea, Pinterest, and Wayfair are actively working toward understanding their customers style-needs [SLJ19; JLK*15; ALW*19]. Researchers have proposed numerous methods for understanding the style of 3D objects and learning their visual compatibility. Below we classify previous work based on their approach.

2.1. Geometry-based Style Estimation

Most earlier methods focused on the geometry of 3D models to infer style. Xu et al. [XLZ*10] defined style as anisotropic part scaling among models. They relied on consistent segmentation of parts and define the style distance between shapes based on the differences in scales and orientations of part bounding boxes. Yumer et al. [YCHK15] proposed to interactively deform models according to precomputed style handles, which are limited to predefined perceptual parameters. Both Liu et al. and Lun et al. [LHLF15; LKS15] learned 3D model style based on human perception. They employ crowd-sourcing to quantify the different components of style. In a later work, Lun et al. [LKWS16] transferred style from an example to target 3D model. Hu et al. [HLvK*17] present a method for discovering elements that characterize a given style, where the elements are co-located across different 3D model associated with such style.

While geometry-based methods have made impressive progress in understanding 3D model style compatibility, style estimation is still a challenging problem in the interior furniture style domain, since style is subjective and is also influenced by multiple non-geometric criteria. Moreover, scaling geometric-based approaches to large data sets is difficult, as they often require manual processing of individual 3D models. For example, for most of the above methods, a user needs to provide hand-crafted geometric features specific for their application. Additionally, since such methods rely on part-aware geometric features, they require consistent part segmentation of 3D models within an object class. Acquiring such segmentation, especially for models from new object classes is often a manual and challenging step. Even though recent deep learning methods have recently shown considerable promise in this area [MZC*19], automatically segmenting 3D models is still an active area of research.

We propose to use deep neural networks to learn furniture style directly from annotated images reviewed by interior style experts. Images include more style-relevant features, such as material, textures, and color [KK14], which are not embodied in current geometric approaches. In this work, we demonstrate that such features are important for style compatibility. Additionally, we show that our workflow for learning style from images is fast and more flexible, since estimating the style of a new 3D models translates into a single pass of its image through the neural network (Figure 3) as opposed to the geometric approaches where the processing itself could be time consuming.
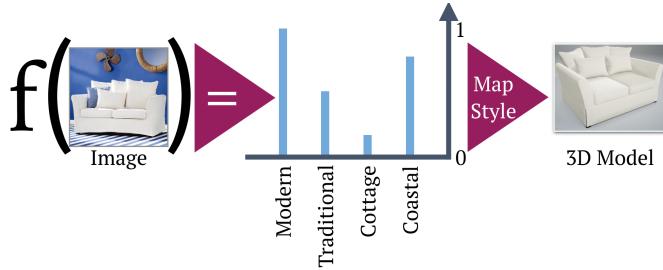
Figure 3: *Our deep learning model learns furniture style from images. Given a context image with a target furniture piece (left), we predict its style embeddings (middle). We then map such embedding values to its corresponding, visually similar 3D model (right).*

Figure 4: *Query furniture image and similar style furniture suggestions. Given the query furniture image on the left, our model suggests similar style furniture on the right.*

2.2. Image-based Style Estimation

With the rise of deep neural networks, there have been numerous work on methods which rely on synthetic and real images to learn style [Nik19]. Such approaches usually formulate style compatibility as a metric learning problem [BHS13] where they train a variation of a siamese network [BGL*94] to place stylistically compatible models close to each other in a embedding space. Using such a framework, Bell et al. [BB15] detect visual similarity between product images. Li et al. [LSQ*15] proposed to learn a joint embedding of images and 3D models for the task of image and shape based retrieval. Lim et al. [LGK16] employ deep metric learning to identify style using rendered views of 3D models. Specifically they use grayscale renderings of untextured objects and use triplet loss to cluster stylistically similar 3D models in a 512 dimensional embedding space. Similarly, Liu et al. [LTR19] also embeds renderings of 3D models in high dimensional embedding spaces. However, unlike [LGK16], they use textured renderings of isolated 3D models from the Unity Asset Store. They scrape approximately 4000 3D models from 121 model packages comprising of both indoor and outdoor 3D models and build their training data by assigning the same style label to all the models within the same package.

Our work specifically focuses on indoor furniture style compatibility where we propose to use *styled images* of furniture which are annotated by several design experts. Such styled images not only describe the texture, color, and material of the 3D object, but also portray how it is used in the space (Figure 4). All elements combined play a vital role in determining the style [KK14]. Further, we propose to use comparison labels (Section 3.2) as opposed to discrete style labels to asses style compatibility. Since design experts collectively often do not agree on a single style for a 3D object, we show that training with such comparison labels allows us to better learn the style spectrum of 3D models. Lastly, unlike previous works, we use high quality real world images and 3D models for our style compatibility framework.

3. Method

We use a deep neural network to estimate the style of images containing furniture. The input to our network are furniture images in a context scene. Each image is annotated with style labels by multiple interior design experts (Section 3.1). Since style experts

might disagree on an image's style, we add *comparison labels* to improve our network's accuracy (Section 3.2). Image pairs, along with their style and comparison labels are then consumed by our deep neural network (Section 3.3). For such image pair, our deep learning model is trained to predict whether the first image shows more style-specific characteristics than the second image, e.g., is the first image more modern than the second one (Section 3.4). Based on our experiments (Section 4), we found that our training procedure (Section 3.5) enables us to learn a more accurate style spectrum compared to a discrete classification framework. Using the trained style neural network, we extract image embeddings to calculate stylistic compatibility of furniture.

3.1. Style Labels

Each image in our data set is labeled with a style by $M = 10$ interior design experts. These styles include modern, traditional, cottage and coastal (Figure 2). Images are indexed by $i \in \mathcal{N} \equiv \{1, \ldots, N\}$ and experts by $e \in \mathcal{M} \equiv \{1, \ldots, M\}$. Thus, given an image i, each expert e provides a single style label $y_i^e \in \mathcal{L} \equiv \{1, \ldots, L\}$, where $\mathcal{L}$ consists of the four major style labels ($L = 4$). We denote the set of all such labels by $\mathcal{D}_s \equiv \{(i, e, y_i^e)\}$.

3.2. Comparison Labels

Style assessment is subjective, even among interior style experts as shown in Table 1. This ambiguity introduces noise and inaccuracies

Table 1: *Do experts agree on style? To answer this question, we asked multiple participants to label each image. The measure of agreement between participants is represented by the green shaded areas. Each cell's value is computed with:* $1 - \frac{||\{x | x \in I \cap J\}||}{||\{x | x \in I \cup J\}||}$, *where I and J are the set of images with style labels i for rows and j for columns. Note that there is a considerable percentage of images that interior style experts have labeled as both Traditional-Modern, and Cottage-Traditional. Based on such subjectivity, we introduce comparison labels for improved style estimation (Section 3.2).*

	Traditional	Cottage	Coastal	*Style*
	54%	73%	83%	Modern
		65%	87%	Traditional
			82%	Cottage

Table 2: *Style predictions from our method. Given images on the left, our model estimates their style. P denotes the Prediction of our network, and GT denotes Ground Truth labels given by* 10 *style experts. Note that in case of interior experts disagreeing on style, which manifests with mixed GT style labels, our model estimates that images belong to the same style subsets (Section 4).*

	P	GT	Style
	0	0	Modern
	1.0	10	Traditional
	0	0	Cottage
	0	0	Coastal
	0	0	Modern
	0.34	6	Traditional
	0.63	4	Cottage
	0.03	0	Coastal
	0	0	Modern
	0	2	Traditional
	0	0	Cottage
	1.0	8	Coastal

for learning style directly from such style labels (Section 4). Hence, we tackle the problem of noisy style labels by introducing *comparisons* into training. A comparison label indicates the *relative order* between a pair of data samples [YTD*19]. For each style and each pair of images, we generate a comparison label with respect to the relative order of the number of style labels each image receives. Formally, given a style $l \in \mathcal{L}$, we consider all pairs of images (i, j), where $i, j \in \mathcal{N}$, that receive at least one label for style l, i.e., $(i \ j)$ for which there exists e_1 such that $y_i^{e_1} = l$ *and* e_2 such that $y_j^{e_2} = l$. For each image pair (i, j), the comparison label is:

$$
y_{(i,j)}^l = \begin{cases} +1, & \text{if } \sum_{e \in \mathcal{M}} \left(\mathbb{1}(y_i^e = l) - \mathbb{1}(y_j^e = l) \right) > t \\ -1, & \text{if } \sum_{e \in \mathcal{M}} \left(\mathbb{1}(y_j^e = l) - \mathbb{1}(y_i^e = l) \right) > t \end{cases}
$$

In other words, $y_{(i,j)}^l = +1$ indicates that image i is ranked higher than image j with respect to style l, and $y_{(i,j)}^l = -1$, otherwise. We discard pairs that do not satisfy either one of these conditions. The label threshold t controls the trade-off between the noise and additional information introduced by the comparison labels. We repeat this process for all styles $l \in \mathcal{L}$ and obtain the comparison label set $\mathcal{D}_c \equiv \{(i, j, l, y_{(i,j)}^l)\}$, where i and $j \in \mathcal{N}$ and $l \in \mathcal{L}$.

3.3. Network Architecture

Our network architecture is inspired by the siamese network [BGL*94]. A classical siamese network comprises of two base networks which share their weights, followed by a layer regressing the similarity between the network inputs. The parameters

between the twin networks are tied. Such relationship gurantees that for an input of two similar images, the networks would map the images close to each other in feature space, since each network computes the same function. Hence, siamese networks are useful for ranking similarity between images [KZS15].

We use VGG16 [SZ14] as our base network f and replace its last layer with two fully connected layers: a 16-dimensional layer with ReLU activation and an $L = 4$-dimensional layer with softmax activation. This allows us to associate each input image i to the base network f with a $d = 16$-dimensional feature vector, $\boldsymbol{x}_i \in \mathbb{R}^d$.

3.4. Loss Function

We extend the generic application of siamese networks to learn from comparison labels. Comparisons are regressed via the siamese architecture, consisting of two identical base networks $f(\cdot; \boldsymbol{W})$: $\mathbb{R}^d \rightarrow \mathbb{R}^L$, parameterized by the weight matrix $\boldsymbol{W}$. $f(\cdot; \boldsymbol{W})$ receives a image with feature vector $\boldsymbol{x}_i \in \mathbb{R}^d$ and regresses the style label predictions $f(\boldsymbol{x}_i; \boldsymbol{W})^l$, $l \in \mathcal{L}$. The siamese architecture receives a pair of images with feature vectors $\boldsymbol{x}_i \in \mathbb{R}^d$ and $\boldsymbol{x}_j \in \mathbb{R}^d$ and predicts the comparison labels $\hat{y}_{(i,j)}^l$, $l \in \mathcal{L}$, governed by the relative order between style predictions. For all $\{(i, j, l, y_{(i,j)}^l)\} \in \mathcal{D}_c$, the comparison label prediction is given by:

$$
\hat{y}_{(i,j)}^l = f(\boldsymbol{x}_i; \boldsymbol{W})^l - f(\boldsymbol{x}_j; \boldsymbol{W})^l \tag{1}
$$

where superscript $l \in \mathcal{L}$ indicates the l-th element in their softmax prediction. Given these predictions, we learn $f(\cdot; \boldsymbol{W})$ via the following optimization problem:

$$
\min_{\boldsymbol{W}} \sum_{\{(i,j,l,y_{(i,j)}^l)\} \in \mathcal{D}_c} \mathcal{L}(y_{(i,j)}^l, \hat{y}_{(i,j)}^l) \tag{2}
$$

For comparison loss $\mathcal{L}$, we extend and reparametrize the Bradley-Terry model [BT52], which considers relative order between input pairs and is used for modeling comparisons. The Bradley-Terry model assumes that for each image $i \in \mathcal{N}$ and style $l \in \mathcal{L}$, there exists a latent score $s_i^l \in [0, 1]$ governing all comparison events in $\mathcal{D}_c$ involving style l. Comparisons in $\mathcal{D}_c$ are independent, with marginal probabilities given by:

$$
P(y_{(i,j)}^l = +1) = \frac{s_i^l}{s_i^l + s_j^l}, \qquad \forall (i, j, l, y_{(i,j)}^l) \in \mathcal{D}_c. \tag{3}
$$

We reparametrize the Bradley-Terry model to regress scores s_i^l, $l \in \mathcal{L}$ as functions of image features $\boldsymbol{x}_i$. Hence, $s_i^l, l \in \mathcal{L}$ correspond to the style predictions from f, i.e., $s_i^l = e^{f(\boldsymbol{x}_i; \boldsymbol{W})^l}$, $i \in \mathcal{N}$, $l \in \mathcal{L}$. Then, given an image pair (i, j), $i, j \in \mathcal{N}$ and the corresponding comparison labels $y_{(i,j)}^l$, $l \in \mathcal{L}$, the comparison loss $\mathcal{L}$ is the negative log-likelihood under the Bradley-Terry model:

$$
\mathcal{L}(y_{(i,j)}^l, \hat{y}_{(i,j)}^l) = \log(1 + e^{-y_{(i,j)}^l \hat{y}_{(i,j)}^l}), \tag{4}
$$

where $\hat{y}_{(i,j)}^l$ is given by Eq. (1).

3.5. Training & Implementation Details

Image Dataset. We use images gathered from multiple sources: (i) Staging a scene in a physical room, and then capturing it with a

camera, (ii) Captured from a virtual scene, curated by a 3D artist, and (iii) Scraped from web and 3rd parties. Due to noisy image metadata, an image's origin might not always be known. Our data set contains about $N = 672K$ images, depicting 20 different furniture categories, such as sofas, coffee tables, and dining tables. Images depict furniture in a styled room, (Figure 4), and most show a wide view, providing ample style context (Table 2). We reserve 80% of N images for training, 10% for validation, and 10% for testing. Images in the training set are not paired with images in the validation or test sets for generating comparison labels. We also resize and apply white padding to all images, so that all images $\in \mathbb{R}^{224 \times 224 \times 3}$.

Validation Dataset. To tune the hyperparameters of our style estimation model, we create a validation data set of clean style labels. For each style $l \in \mathcal{L}$, we only consider images that receive at least $l_{\min}$ labels, i.e., image i is in style category l if $\sum_{e \in \mathcal{M}} \mathbb{1}(y_i^e = l) \geq l_{\min}$. We obtain two style validation data sets: (i) set $\mathcal{D}_s$ containing all images with style labels $l_{\min} = 1$ for all $l \in \mathcal{L}$, and (ii) set $\mathcal{D}_{s,\text{clean}}$ containing only images with high agreement style labels of at least $l_{\min} = 10$ for modern, $l_{\min} = 8$ for traditional, and $l_{\min} = 7$ for coastal and cottage. Such thresholds are selected empirically considering number of samples per style category and amount of labeling noise introduced. As a result, each image in set $\mathcal{D}_{s,\text{clean}}$ is associated with a single ground-truth that has the maximum number of style labels, whereas in set $\mathcal{D}_s$, the same image can appear with different style labels due to expert disagreement.

Training. We train our siamese architecture on the comparison label set $\mathcal{D}_c$. We employ transfer learning and initialize VGG16 with weights pre-trained on the Places365 data set [ZLK*17] for place classification. After initialization, we freeze the weights of the first 21 layers and fine tune the weights of the last 3 fully connected layers, comprising 16.8M parameters. We use RMSProp optimizer [TH12] with learning rate 0.0001, to optimize $\mathcal{L}$ in Eq. (2). We add L2 regularizers with a regularization parameter λ to all layers. We choose the regularization parameter λ, comparison label threshold t, and number of training comparisons N_c w.r.t. the classification performance on the validation set, via a grid search. For each combination of $\lambda \in [0.002, 0.0002, 0.00002]$, $t \in [1, 2, 3]$, and $N_c \in [500K, 1M, 2M, 3M, 4M, 5M, 10M, 15M]$, we train the style estimation model on N_c comparisons selected uniformly at random over the training set of $\mathcal{D}_c$. Finally, we evaluate the resulting models to determine optimal parameters for predicting style, in which we found $t = 3$ to be experimentally optimal.

Our machine learning model was implemented using Tensorflow. Training of our style-learning model took about 5 hours on a NVIDIA Tesla V100 GPU'.' Below we evaluate our network both qualitatively and quantitatively, w.r.t. several metrics.

4. Evaluating Image Style Compatibility

Qualitative Evaluation. In Table 2 we show example images comparing the predictions from our model with the ground truth labels provided by interior design experts. For images with distinct style labels, our model generates similar equivalent scores. In case of mixed style labels, our model recommends images with labels disagreeing on the same subset of styles as experts. For each mixed
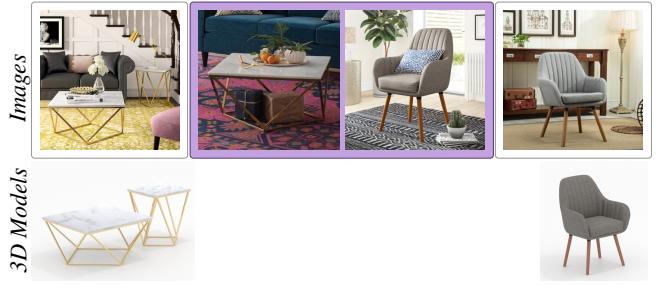

Figure 5: *We use images of curated interiors to estimate a 3D models style. In this example, we are interested to estimate the style compatibility between the coffee table set on the bottom left, to accent chair on the bottom right. We use images on the top to estimate such compatibility. Images framed in purple are the closest stylewise, according to the embeddings distance (Eq. 5).*

Figure 6: *A furniture piece, such as the accent chair, can appear in multiple styled scenes. We consider all associated images when searching for other style-complementary furniture.*

classification, we observe that the image's style is more of a spectrum of different visual features, and it can indeed belong to either of the estimated styles. Hence, in addition to estimating style, our framework has learned distinctive visual features reflecting style. Our results validate that training with comparison labels can successfully capture noise in the style labels and accurately model the style spectrum of each image.

Quantitative Evaluation. We evaluate our model with images in our test set, which comprise of 10% of our total image data set. Additionally, we also compare our model's performance against two baseline models which estimate style without comparison labels. Such baseline models were trained on: (i) set $\mathcal{D}_{s,\text{clean}}$ representing high agreement images where most experts agreed on the same style label. (ii) set $\mathcal{D}_s$ containing all images. High agreement images were given style labels of $l_{\min} = 10$ for modern, $l_{\min} = 8$ for traditional, and $l_{\min} = 7$ for coastal and cottage (Figure 2). Such thresholds were selected empirically for best classification accuracy. Below we summarize our quantitative evaluation outcomes.

Style Classification Accuracy. For each image i in the test set, we predict its style from its softmax prediction, i.e $\text{argmax}_{l \in \mathcal{L}} f(\mathbf{x}_i; \mathbf{W})^l$ and compute its accuracy against the ground truth. In Figure 7a we show the classification performance where we perform better than the baselines for styles such as Cottage and Coastal but worse for Modern and Traditional. However, our network has better classification accuracy on average over the baseline models with 79% accuracy and $4.7 - 8.1\%$ improvement.

Figure 7: *Learning from style comparison labels improve accuracy over non-comparison deep learning networks. We compare our model trained on comparison labels with two baseline models, one trained on all images ($\mathcal{D}_s$) and second trained on clean images ($\mathcal{D}_{s,clean}$). Classification accuracy increased by 24% on cottage, 45% on coastal, and 5% on average over styles. Such results validate that comparisons labels reveal more information than style labels, even with the challenges of class imbalance and label noise. Note that the poor performance of Cottage and Coastal is due to class imbalance. In our data set, the number of images with at least 1 label for each style is about 500k for modern, 400k for traditional, 300k for cottage and 150k for coastal.*


(a) *Classification Accuracy*

(b) *Recall rate at k=1*

(c) *Retrieval at k=5*
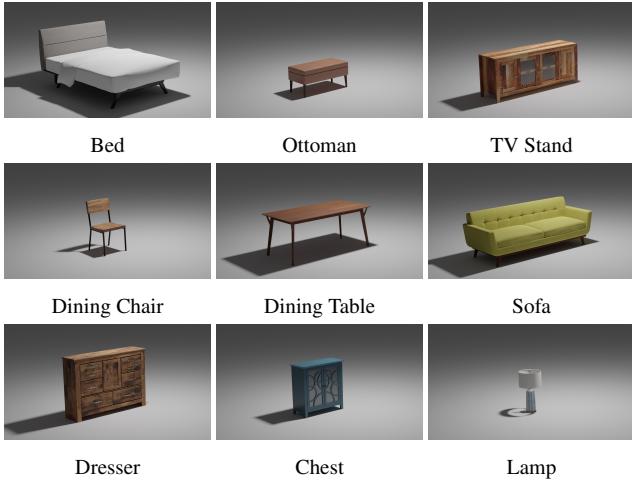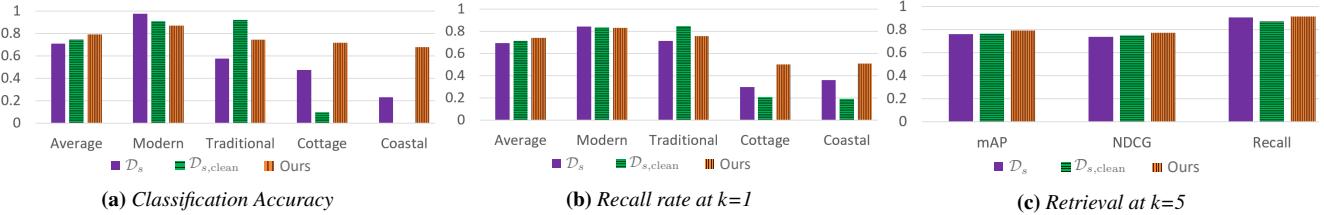
Figure 8: *Examples of 3D model classes in our data set.*

Style Retrieval Performance. To evaluate the retrieval performance, we use the 16-dimensional output of the second last layer of f as the embedding of each image in the test set. Then, for each image, we retrieve the nearest image(s) in the remaining test set w.r.t. the Euclidean distance of extracted embeddings. A retrieved image is considered relevant if its style matches with the style of the query image. We compute mean Average Precision (mAP), recall rate at k, and Normalized Discounted Cumulative Gain (NDCG) on the retrieved images [Liu11]. Figure 7b and 7c shows the retrieval performances of our network and the compared baselines. We observed approximately equally successful retrievals on all 4 style categories for recall rate at 5, with our model performing better in terms of mAP with $2.9 - 3.2\%$ improvement and NDCG with $2.4 - 3.5\%$ improvement. In recall rate at 1, our model improved per class retrieval performance by $15 - 21\%$.

Comparison Labels and Global Compatibility. Comparison labels imply of a hidden global style score of each image, within each of our discrete styles. For example, given images with a style score of a, b and c, where $a > b$ and $b > c$, then $a > c$ should hold with high probability, since the Bradley-Terry model assumes stochas-

tic transitivity. We did not eliminate training comparisons that did not satisfy this condition. However, our network was able to rectify such conditions based on its performance. Overall, comparison labels exhibit less noise compared to discrete class labels (Figure 7). Elimination or correction of such noise is a subject of future work, and an active area of research [SR18].

5. Furniture Style Compatibility

5.1. 3D Model Similarity

We infer the style compatibility of 3D furniture models using images, similar to recent work that infers style of 3D furniture models from rendered images [LTR19; LGK16]. However, the *key difference* of our method is that we utilize the entire styled scene, to estimate a 3D furniture's style compatibility. The machinery we use to that end is a furniture's image embeddings. A 3D furniture model might appear in multiple images (Figure 6). Hence, we first validate that 3D models have a consistent visual appearance in all images they appear in (Figure 10). We then measure the stylistic compatibility between two furniture pieces by calculating the minimal embedding distance between all images the furniture items are associated with

$$d(m_i, m_j) = \min_{\forall i \in I, \forall j \in J} ||e(i), e(j)||, \quad (5)$$

where m_i and m_j are 3D furniture models, I is the set of images associated with the furniture piece m_i, and J is the set of images associated with furniture m_j, and e is a function that returns an image's embedding. Based on the above, we create a system that given a seed 3D furniture model(s), retrieves similar style 3D furniture's (Section 6).

5.2. Style-Compatible Scene Modeling

To qualitatively validate our 3D object style similarity, we developed a system for style-aware modeling of virtual scenes. If furniture has similar style, it will be preserved in a 3D setting, allowing to create a more harmonious ambience in the room in which furniture's fit-well together. Another motivation for such a system is that it is difficult for users to navigate a catalog of 3D furniture models and search for stylistically compatibility furniture fitting a

Query 3D Model

Ranked Results

Figure 9: *Our method provides style-aware 3D model recommendations. Here, based on the query coffee table on the left, we provide compatible sofas on the right.*

(a) *Similar*   **(b)** *Dissimilar*

Figure 10: *Validating visual similarity of furniture. Furniture might appear visually different when comparing rendered views of 3D models to curated scene images. We manually verify the furniture's similarity with a tool we created, which includes the following options: (a) Furniture in image, and its 3D model exhibit visual similarity. (b) Furniture's model has a visually different color, and hence disallows us from considering it part of our recommendation model. (c) A question mark button, in case the user is not sufficiently confident.*

user's scene [TV04]. To that end, we designed our system to provide style-aware suggestions based on selected query scene objects. A user may select one, or multiple 3D furniture's, for which our system generates a list of style-compatible suggestions. In response to a user's query, our system displays suggestions of top most compatible furniture, as shown in Figure 9. Suggestions are retrieved in real-time, since image embeddings and embedding distances can be calculated in advance. We included different strategies to retrieve style-compatible furniture suggestions for virtual scenes:

Single Seed Style Reference. In this 3D modeling scenario, a user selects a main 3D furniture piece to anchor a scene's style. All further style-compatible suggestions are based only on the seed 3D model:

$$d(m_i, m_{seed}), \qquad (6)$$

where d measures the stylistic compatibility for a pair of scene objects (Eq. 5), m_i are scene objects for which we measure the compatibility with the seed furniture m_{seed}, which is the scene's main 3D model. The motivation behind such a suggestion strategy is that 3D artists typically want to focus on a single, main furniture piece for the entire scene [LMLF15]. Other furniture objects are placed around the main furniture, to enhance it visually. For example, an artist is creating a scene to market a sofa or an accent chair. To exemplify such scenario, we created specific room types such bed-

room, studio apartment, and restaurant (Figure 11). Please see section 6.1 for more details.

Multiple Style References. In this scenario, we consider multiple scene objects for providing style compatible suggestions. [LHLF15] propose a compatibility energy function for an entire scene as the sum of the compatibility distances between all scene objects. Similarly, we define the style compatibility for a selected object m_{select} as the sum of the style distances with all existing scene objects:

$$\sum_{m_i \in S} d(m_{select}, m_i), \qquad (7)$$

where S is the set of scene 3D models, and d is a function that returns the style compatibility distance according to Eq. 5. Similarly, to measure the stylistic compatibility of an entire scene:

$$\sum_{\substack{m_i, m_j \in S \\ i \neq j}} d(m_i, m_j). \qquad (8)$$

To exemplify such a stylistic compatibility scenarios, we investigated style of furniture arrangements and introduce a tool for interactive style-aware scene building. Our tool allows users to create and modify existing furniture arrangements for various room types. Please see Section 6.3 for more details.

6. Evaluating 3D Furniture Style Compatibility

We implemented a system for interactive scene building of style-compatible furniture arrangements. The system's input is either single or multiple object queries and outputs suggestions of style-compatible furniture pieces, similar to prior work [LHLF15]. Suggestions are based on embeddings of furniture images. We precompute image style embeddings and pairwise 3D furniture compatibility distances to achieve interactive run-time for all furniture suggestion queries. Such quality assists in interactive modeling scenarios with multiple scene objects. Scene modeling, and other interactive components of our system are implemented in Python and JavaScript and tested on a 2.5 GHz Intel Core i7. We tested our system on the following scenarios: (i) Creating a scene to fit a single 3D model's style, and (ii) Modification of scenes for multiple 3D model's styles. Below we describe our experiment setup, including how we obtained our 3D model data set.

3D Furniture Models Dataset. We collected textured 3D models of furniture from a variety of sources: manually created, sourced from the web, and provided by the furniture's manufacturer. In our experiments we used about 1148 3D furniture models, which fit residential and public spaces such as restaurants. Models are divided

(a) *Junior Bedroom*   **(b)** *Dining Room*   **(c)** *Meeting Room*   **(d)** *Family Room*

(e) *Living Room*   **(f)** *Hip Studio*   **(g)** *Asian Restaurant*   **(h)** *Lounge*
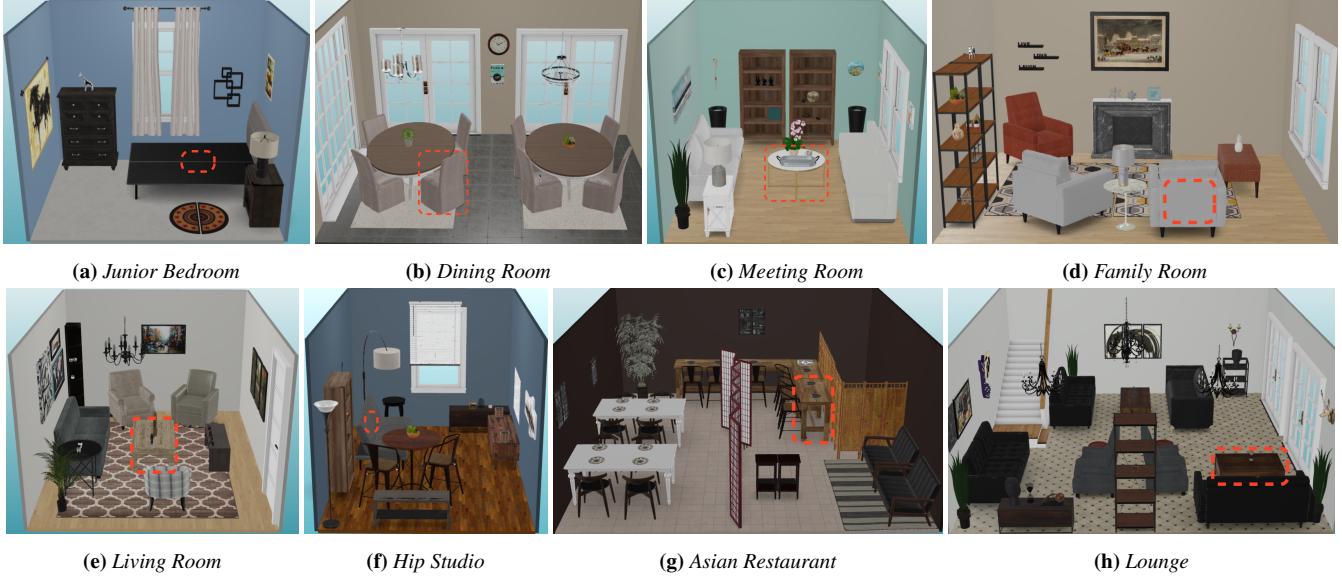
Figure 11: *Given a context seed furniture marked in red, we curate a scene by selecting from a set of stylistically-compatible furnitures recommended by our model. Our approach is flexible across different room types, from a bedroom to a lounge area.*

into 20 object classes, including: accent chairs, sofas, sectionals, end tables, coffee tables, beds, dressers, bookcases, lamps, dining tables, dining chairs and similar. We manually normalized the models to a consistent scale and orientation, keeping 3D models that are visually distinct from each other. Figure 8 shows few examples of the furniture classes we used.

Validating Visual Appearances between 3D Furniture and Images. Our data set contains images scrapped form the web, which might contain noise and irregularities either in images or in 3D models. Irregularities might result from various causes: errors in the 3D model creation pipeline, light conditions, material setup, among other reasons. The style compatibility system we propose is dependent on consistent visual appearance of 3D models and associated images. Typically, several images are associated with each 3D model, showing the model from different camera angles. To assess such visual similarity, we manually compare a furniture's 3D model and its images. Such manual comparison is necessary since it is difficult to computationally judge similarity between 3D models and images. For annotating such comparisons, we use 3 options: similar, not similar, and unknown, in case of insufficient confidence in the decision. Experimentally, we found that only 60% (1148 out of 1911) of the 3D models we sampled have a valid visual appearances, suggesting the importance of this step. Selected 3D models have a matching visual appearance to their respective images (Figure 10).

6.1. Experiment: Single Style Reference

Manually searching for 3D models that fit a certain style or reference object is laborious. We wanted to evaluate if we can assist professional users in creating virtual scenes by constraining available 3D models to a filtered, style-compatible set. To that end, we asked a professional designer to complete several room building tasks.

For each task, we placed one 3D model in a generic virtual room, which provides a style reference. The goal is to complete-the-look of a specific scene type, by adding objects from a style-compatible set, which is generated based on the reference object's embeddings (Eq. 6). Such set contains up to 150 3D models. We choose this number experimentally for diversity in object types. After placing style-compatible objects, users could add minor room object types not supported by out method, such as accessories, clutter, wall art, and other decor objects. Users were also allowed to modify architectural features, such as the floor plan, colors, doors and windows. Figures 11 show completed room examples.

A natural question to ask is whether our tool assists in creating style-compatible scenes. We investigated this question by asking non-professionals, since our system is ultimately targeted at such users. We first educate users by showing multiple images for each style. Then, we instruct users to assess whether created scenes are style compatible with scores ranging from 1 to 5, where 1 denotes low agreement, and 5 denotes high agreement. Table 3 summarizes their responses. The results were collected from 14 participants with no prior professional experience in interior design or scene modeling. The majority of the responses from participants were positive regarding the style-compatibility of the resulting scenes.

6.2. Experiment: Multiple Style References

Next, we evaluated our method in case of multiple inputs (Section 5.2). We first studied our style-compatibility in case of object triplets of several common furniture combinations. To generate the triplets, we selected furniture classes, such as sofas, accent chairs, and tables. Then, for all objects that belong to such classes, we measured the distance compatibility for permutations created by selecting objects from each class (Eq. 8). For each of the combi-

More Compatible   Less Compatible

Figure 12: *Comparing style-compatibility of furniture, where scenes on the left are stylistically more compatible that the one on the right. We measure compatibility between multiple 3D models using embeddings extracted from our network (Eq. 8). To find whether one collection of 3D models is more style-wise compatible than the other, we calculate the sum of the style-compatibility distances between all object pairs. In the above examples, each row contains the same object classes, e.g., sofa, and accent chair and a coffee table for the first row.*

nations in Figure 12, the more style-wise compatible a furniture is, the less the total embedding distance is:

a. Sofa, coffee tables and accent chair: The scene on the left has a more modern style, with color and textures complementing each other. The scene on the right is mixed between the somewhat traditional coffee table, and modern accent chair.

b. Dining table and a pair of dining chairs: Note, the configuration on the left is more traditional in style. The configuration on the right is a combination of a traditional-looking table with a more modern chair.

c. Two different accent chairs and an end table: The two accent chairs and an end table on the left are leaning to the traditional style, and are more similar in shape. On the right, the style is not conclusive, and the color of each object does not complement each other.

d. Accent chair, nightstand and table lamp: For the arrangement on the left, our method picked a lamp that has similar color to the accent chair, while the nightstand does not detract from the scenes color harmony. On the right, the furniture might be considered compatible, however the accent chair and nightstand are not quite stylistically compatible.

Table 3: *User study on scenes created based on a single furniture style reference. For each scene, a user initially selects a seed 3D model. Our system recommends compatible furniture's which are then added to the scene. Further, auxillary 3D models such as doors, windows, and clutter were later added to enhance the scene (Figure 11). Column "style" denotes the leading scene style. Column "#" denotes the number of objects selected with our recommendation model. The last column refers to results from our user study, where we asked users to rank the resulting scene is stylistically compatible, where 1 denotes low agreement, and 5 denotes high agreement. Error bars indicate one standard deviation.*

Scene	#	Style	Compatible
Dining Room	2	Traditional	
Junior Bedroom	4	Modern	
Meeting Room	6	Coastal	
Family Room	6	Modern	
Asian Resturant	6	Modern	
Hip Studio	8	Modern	
Lounge	8	Traditional	
Living Room	9	Cottage	
			1 2 3 4 5

6.3. Application: Style-Aware Scene Modeling

To further experiment with style compatibility for multiple reference 3D models, we implemented a web-based interactive system for style-aware augmentation of scenes. Our system provides user with style-compatible furniture suggestions. We based our application on the proposal of [LKS15], which demonstrates style-based suggestions for scene modeling, based on a context furniture. Similar to their suggestion, we present style-compatible suggestions with a list, in which furniture is ranked according style compatibility fit, from most compatible to least compatible (Eq. 7). To generate such list, users select existing scene objects. Such selection constraints suggestions to the type of the selected object, e.g., if the user selects a coffee table, the suggestion list will only display coffee tables. After user selections, the suggestions list is interactively generated, allowing users to swap scene objects with 3D models in the suggestion list (Figure 13).

Similar-style lamp suggestions for a bedroom.

Our system allows several user interactions. To anchor the visual experience, we set the background to be static for all interaction modes. We use a warehouse-like space to be room-function and style generic, so that architectural details are static. Initially, users either select an existing scenario, or create a new one in a free flow interactive manner. After selecting a scenario, users can either add new furniture, or swap existing furniture. Users control the placement of furniture, in terms of position, rotation, and can also place on top of other furniture, e.g., a lamp on top of a night stand. For style compatible suggestions, users select a target object in the scene. Users may then click on "Get Suggestion" button for style-compatible suggestions, and swap selected furniture with the most-compatible furniture. We allow users to revert their actions, and save their work.

Figure 13: *We implemented an interactive system for suggesting style-compatible furniture. The top image shows the initial scene configuration, where the user has selected two chairs to be swapped, annotated with circles. By clicking, "Get Suggestion", suggestions are generated on the right. The bottom image shows the result of swapping chairs using our style-compatible suggestions.*

To evaluate the effectiveness of our system, we experimented with several scenarios, including bedroom, dining and living room furniture arrangements (Figure 13). Interacting with our system is simple and requires little to no training. Through clicking and dragging actions, users have intuitive control, without needing to be aware of the style labels. Usability testing would be needed to further substantiate the intuitive nature of our user interface. Our system generated style-compatible suggestions in real-time, which we recorded (see supplemental video).

6.4. Comparison with Previous Work

We compare the results from our image-based style framework with [LHLF15] which uses geometric features of 3D models to compute style compatibility across different furniture classes. We show this comparison in Figure 14 where we retrieve the top 4 stylistically compatible chairs, along with their compatibility distance, to a query dining table using both methods. We make the following observations:

- Our system is able to successfully retrieve stylistically compatible chairs for the query table model. For both the tables, the first 4 chairs from the left are stylistically more compatible in terms of color and texture. Whereas the last 4 chairs do not complement the tables, reflected in the high compatibility distance from our method.
- Style suggestions based purely on geometric approaches dif-

fer from image-based frameworks such as ours. This is natural as while our framework focuses on color and texture of furniture pieces in an image, geometric-based approaches use their 3D models to compute other properties such as curvature, surface area, and bounding boxes. This can also be seen in the last 4 chairs of the second column where the ordering of the chairs by our compatibility distance is different than the ranking by [LHLF15].
- Models which are close in the geometric embedding space, may not be as close in the image embedding space. For example, in second row, the 5th and 6th chair from the left have a minor difference in their geometric compatibility distance, but texturally differ, a quality which is reflected in a large image compatibility distance. Such style distance has foundation since a model can exhibit different styles by using different textures while retaining the same 3D geometry.

7. System Scope and Limitations

Our system provides style-based suggestions for input furniture, working with both images and 3D models. Since style-compatibility of 3D furniture models is inferred from images, suggested 3D models might not fit together for non-style reasons, which we summarize below:

Functionality and Synthesis. Considering we infer style-compatibility from images, our method does not take furniture functionality into account, e.g., bedroom versus living room furniture. Similarly, we do not incorporate furniture arrangement and synthesis [YYT*11; WLD*18; ZWK19] when modeling 3D scenes. While important, such methods are orthogonal to our style-focused scope.

Scene Parameters. 3D scenes vary in terms of conditions, ranging from lights [JL19], camera setup [LMLF15], to contents. Such conditions have a perceptual affect on style. For example, in case of more multiple scene objects, their color and combined appearance might not be harmonious. However, experimentally we found that selecting from a set of style-compatible furniture is sufficient for creating an harmonious look and ambience.

Architectural Elements. We do not consider scene architectural space elements, such as wall and floor color. Nevertheless, our current approach demonstrates that existing scene objects provide an anchor for selecting new furniture pieces for a room.

Interaction and Visual Quality. We implemented our system to work within a web-based framework. Currently, such web-based frameworks limits lighting configurations, texture, and mesh complexity. Such frameworks do not easily incorporate quality real-time rendering. Additionally, since our goal is to support multiple users, visual quality is set to allow real-time performance for most web users, including mobile phones. However, since all our 3D models are real-time PBR models in GLTF format, our framework enables cross-platform usage with web and other interactive use cases.

Compatibility of Furniture from Different Styles. Furniture items from different style categories might be suggested to users. For example, a modern furniture piece could be suggested for a traditional

Figure 14: *Style-aware shape retrieval. Given a query 3D model such as a dining table, we compare the suggestions from our method (green) which use image features such as color, texture and material with [LHLF15] (blue) which use geometric features computed directly on the 3D shape. For both methods we show their top 4 most stylistically compatible chairs along with their compatibility distances to the query table. We also show for all chairs the compatibility distance if the other method was used, in their top inset. Note, both methods report the compatibility distances at different scales but share similar meaning of lower values being more compatible.*

scene, since our network learns the abstract notion of style from pixels in order to satisfy expert labels. Since style is subjective, the resulting scene might be satisfactory. Additionally, our system allows users to edit and select preferred furniture with a user interface.
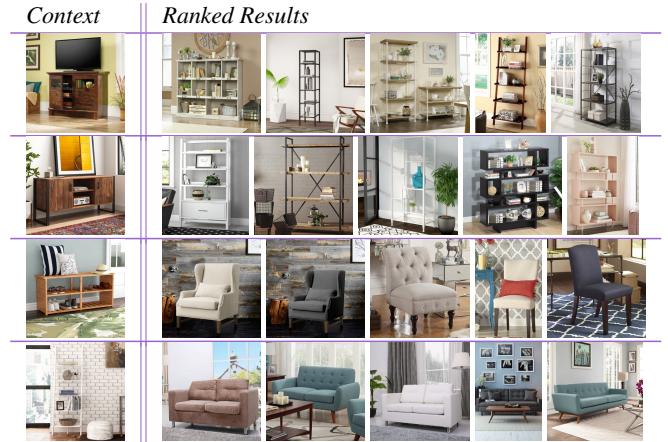
Affects of Curated Imagery on Style Estimation. As mentioned in Section 5.1, style compatibility between two furniture pieces is calculated via their minimum image embedding distance. Images depict furniture in the context of a styled scene, with multiple surrounding features, such as floor, wall, decor and other furniture items. Our system considers the entire image, including the surrouding features, when predicting style. Currently, we do not experiment with the affects of individual surrounding features on percieved style, since such features may be multiple and high-dimensional. Nevertheless, our system successfully learns style from images, using both the surroundings and the furniture, as we show qualitatively in Figures 11, 12 and 14, and quantitatively in Figure 7.

8. Conclusion & Future Work

In this work, we introduced a deep learning based system for estimating style-compatibility of furniture images and 3D models. We then integrated our method with a multiple component system to exemplify 3D model style-compatibility. The system includes a deep learning network for style estimation, visual validation of furniture images and respective 3D models, and a web-based application for creating style-compatible scenes.

In contrast to previous work, we learn furniture style using curated interior images. Images are labeled with 4 styles by multiple experts. We introduced comparison labels, which allows us to distinguish between a wider spectrum of styles, even for ambiguous inputs. Furthermore, comparing style of different classes is not possible. For example, being modern cannot be greater or less than

Table 4: *Similar style furniture images. Given a context image on the left, our model recommends similar style furniture. Here, we demonstrate recommending a specific furniture type for context furniture. From top to bottom: given a TV stand, entertainment center, bench, and bookcase on the left, we recommend bookcases (x2), accent chairs, and sofas on the right.*

Context	Ranked Results

being traditional. To evaluate our method, we measured accuracy and performance of various style estimation tasks. We conducted multiple scene modeling experiments for demonstrating 3D model style-compatibility in a variety of scene and style settings.

Our system has several notable directions for future work. First, we are working toward adding more styles to our model. The main challenge would be in accumulating sufficient annotated images, while maintaining style estimation performance attained on less styles. As previously reported (Table 1), styles are subjective even among experts, so it will be beneficial to understand what new styles to introduce that would be sufficiently distinct from cur-

rent styles. It will also be interesting to utilize an unsupervised approach to learn style. Second, reducing the number of manual user steps in the system would be beneficial to future users. One such opportunity is when validating if a furniture image's is visually consistent with its 3D model. Here we can draw inspiration from deep networks that aim to learn a common embedding between 3D models and images [LSQ*15]. Third, our model infers style compatibility based on categories such as modern, traditional, cottage and coastal. Style-compatibility is not sufficient to guarantee that scenes will be harmonious, for non stylistic reasons. For example, furniture combinations might not match in respect to function or color. It will be worthwhile to extend our framework for such considerations. Lastly, our network measures style compatibility of individual objects by using images of curated scenes as a context. Inferring interior style-compatibility spectrum of individual furniture objects directly would be beneficial. We believe this would be an exciting direction for future work.

Acknowledgements

References

[ALW*19] ATAER-CANSIZOGLU, ESRA, LIU, HANTIAN, WEISS, TOMER, et al. "Room Style Estimation for Style-Aware Recommendation". *2019 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*. IEEE. 2019, 267–2673 2.

[BB15] BELL, SEAN and BALA, KAVITA. "Learning visual similarity for product design with convolutional neural networks". *ACM Transactions on Graphics (TOG)* (2015) 3.

[BGL*94] BROMLEY, JANE, GUYON, ISABELLE, LECUN, YANN, et al. "Signature verification using a "siamese" time delay neural network". *Advances in Neural Information Processing Systems (NeurIPS)*. 1994 2–4.

[BHS13] BELLET, AURÉLIEN, HABRARD, AMAURY, and SEBBAN, MARC. "A survey on metric learning for feature vectors and structured data". *arXiv preprint arXiv:1306.6709* (2013) 3.

[BT52] BRADLEY, RALPH ALLAN and TERRY, MILTON E. "Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons". *Biometrika* (1952) 4.

[HLvK*17] HU, RUIZHEN, LI, WENCHAO, van KAICK, OLIVER, et al. "Co-Locating Style-Defining Elements on 3D Shapes". *ACM Transactions on Graphics (TOG)* (2017) 2.

[JL19] JIN, SAM and LEE, SUNG-HEE. "Lighting Layout Optimization for 3D Indoor Scenes". *Computer Graphics Forum*. 2019 10.

[JLK*15] JING, YUSHI, LIU, DAVID, KISLYUK, DMITRY, et al. "Visual search at pinterest". *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. 2015 2.

[KK14] KILMER, ROSEMARY and KILMER, W OTIE. *Designing interiors*. John Wiley & Sons, 2014 1–3.

[KZS15] KOCH, GREGORY, ZEMEL, RICHARD, and SALAKHUTDINOV, RUSLAN. "Siamese neural networks for one-shot image recognition". *ICML deep learning workshop*. Vol. 2. Lille. 2015 4.

[LGK16] LIM, ISAAK, GEHRE, ANNE, and KOBBELT, LEIF. "Identifying style of 3D shapes using deep metric learning". *Computer Graphics Forum*. 2016 3, 6.

[LHLF15] LIU, TIANQIANG, HERTZMANN, AARON, LI, WILMOT, and FUNKHOUSER, THOMAS. "Style compatibility for 3D furniture models". *ACM Transactions on Graphics (TOG)* (2015) 2, 7, 10, 11.

[Liu11] LIU, TIE-YAN. *Learning to rank for information retrieval*. Springer Science & Business Media, 2011 6.

[LKS15] LUN, ZHAOLIANG, KALOGERAKIS, EVANGELOS, and SHEFFER, ALLA. "Elements of style: learning perceptual shape style similarity". *ACM Transactions on graphics (TOG)* (2015) 2, 9.

[LKWS16] LUN, ZHAOLIANG, KALOGERAKIS, EVANGELOS, WANG, RUI, and SHEFFER, ALLA. "Functionality preserving shape style transfer". *ACM Transactions on Graphics (TOG)* (2016) 2.

[LMLF15] LIU, TIANQIANG, MCCANN, JIM, LI, WILMOT, and FUNKHOUSER, THOMAS. "Composition-Aware Scene Optimization for Product Images". *Computer Graphics Forum*. 2015 7, 10.

[LSQ*15] LI, YANGYAN, SU, HAO, QI, CHARLES RUIZHONGTAI, et al. "Joint embeddings of shapes and images via CNN image purification". *ACM Transactions on Graphics (TOG)* (2015) 3, 12.

[LTR19] LIU, YIFAN, TANG, RUOLAN, and RITCHIE, DANIEL. "Learning Style Compatibility Between Objects in a Real-World 3D Asset Database". *Computer Graphics Forum*. 2019 3, 6.

[MZC*19] MO, KAICHUN, ZHU, SHILIN, CHANG, ANGEL X, et al. "Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding". *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019 2.

[Nik19] NIKOLENKO, SERGEY I. "Synthetic Data for Deep Learning". *arXiv preprint arXiv:1909.11512* (2019) 3.

[SLJ19] SACHIDANANDAN, SANDHYA, LUONG, RICHARD, and JOERGENSEN, EMIL S. "Designer-driven add-to-cart recommendations". *ACM Conference on Recommender Systems*. 2019 2.

[SR18] SAHA, AADIRUPA and RAJKUMAR, ARUN. "Ranking with Features: Algorithm and A Graph Theoretic Analysis". *arXiv preprint arXiv:1808.03857* (2018) 6.

[SZ14] SIMONYAN, KAREN and ZISSERMAN, ANDREW. "Very deep convolutional networks for large-scale image recognition". *arXiv:1409.1556* (2014) 2, 4.

[TH12] TIELEMAN, TIJMEN and HINTON, GEOFFREY. "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude". *COURSERA: Neural networks for machine learning* (2012) 5.

[TV04] TANGELDER, JOHAN WH and VELTKAMP, REMCO C. "A survey of content based 3D shape retrieval methods". *Proceedings Shape Modeling Applications*. 2004 7.

[WLD*18] WEISS, TOMER, LITTENEKER, ALAN, DUNCAN, NOAH, et al. "Fast and Scalable Position-Based Layout Synthesis". *IEEE Transactions on Visualization and Computer Graphics* (2018) 10.

[XLZ*10] XU, KAI, LI, HONGHUA, ZHANG, HAO, et al. "Style-content separation by anisotropic part scales". *ACM Transactions on Graphics (TOG)* (2010) 2.

[YCHK15] YUMER, MEHMET ERSIN, CHAUDHURI, SIDDHARTHA, HODGINS, JESSICA K, and KARA, LEVENT BURAK. "Semantic shape editing using deformation handles". *ACM Transactions on Graphics (TOG)* (2015) 2.

[YTD*19] YILDIZ, İLKAY, TIAN, PENG, DY, JENNIFER, et al. "Classification and comparison via neural networks". *Neural Networks* (2019) 4.

[YYT*11] YU, LAP FAI, YEUNG, SAI KIT, TANG, CHI KEUNG, et al. "Make it home: Automatic optimization of furniture arrangement". *ACM Transactions on Graphics (TOG)* (2011) 10.

[ZLK*17] ZHOU, BOLEI, LAPEDRIZA, AGATA, KHOSLA, ADITYA, et al. "Places: A 10 million image database for scene recognition". *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* (2017) 5.

[ZWK19] ZHOU, YANG, WHILE, ZACHARY, and KALOGERAKIS, EVANGELOS. "SceneGraphNet: Neural Message Passing for 3D Indoor Scene Augmentation". *International Conference on Computer Vision (ICCV)*. 2019, 7384–7392 10.