Décor — Midterm Report

David Chuan-En Lin [*1] **Haocheng Han** [*1] **Yuanxin Wang** [*1] **Venkatesh Sivaraman** [*1]

Abstract

The current popularity of e-commerce motivates intelligent ways to help surface stylistically compatible products to customers. However, most existing machine-learning approaches have not directly addressed notions of style. In this project, we aim to tackle the challenge of learning style representations of furniture based on their text description and visual appearance. This midterm report introduces our three baseline models and analyzes their deficiencies. Our major finding is that ambiguity in defining style match, as well as discrepancies between the standard classification training procedure and the downstream ranking evaluation task, might hinder the development of a successful furniture style search model. We discuss several new research ideas that could enhance the current methods in our upcoming work.

1. Introduction

Deep learning algorithms have achieved state-of-the art performance for a variety of tasks. These algorithms work by learning semantic representations of image content, often utilizing auxiliary multimodal cues to improve granularity and accuracy. While such representations typically encode similarities related to the visual appearance of objects, many tasks such as stylistic compatibility require more abstract conceptualizations.

The capability of learning the stylistic compatibility of objects has applications in a variety of creative disciplines, including interior design, fashion, and architecture. While image similarity has been well-studied (Datta et al., 2008), stylistic compatibility, however, remains a challenging task due to its loose definition and inherent subjectivity. Prior works have largely explored stylistic compatibility by learning from a dataset of objects manually tagged with style categories (e.g. industrial, rustic, modern) (Kiapour et al., 2014; Takagi et al., 2017; Aggarwal et al., 2018). However, since these style categories are inherently vague and subjective, learning from objects labeled with style tags can be a noisy task.

In this work, we investigate the task of learning style compatibility in the context of IKEA furniture. We leverage the IKEA dataset (Tautkute et al., 2017a), which contains 2,193 product photos and text descriptions. Most interestingly, the dataset also includes 298 rooms, curated by IKEA designers, as well as information on which products appear within which rooms. This is valuable information. Our insight is that to create our style compatibility dataset, we can define two products as compatible if they co-occur within the same room. Our positive labels are therefore determined by professional designers in the context of rooms – realistic and free from subjective style vocabulary. Our code is available on `https://github.com/MichaelYxWang/DecorAssistant`.

2. Related Work

2.1. Computational Representations of Style

The computational representation of style has been a problem far from being solved for a long time, as there isn't a clear metric defining what's the definition of style. Blijlevens et al. (2009) researched how consumers perceive the product's appearance or style. They found that three main attributes mainly affected people: modernity, simplicity, and playfulness. Although they define dozens of styles based on the value of these three attributes, we expect a numerical representation without explicit semantic information.

Kiapour et al. (2014) realized that to generate a computational representation, a dataset of style and corresponding machine learning training method is required. They proposed a dataset called Hipster to classify style into five categories: hipster, bohemian, pinup, preppy, and goth. They collect the data from a crowd-sourcing game and finally get 1,893 images with annotation. They identify the style descriptor by accumulating features like RGB value, MR8 texture response, distance from image border, etc. They also tried to analyze the style indicator for individuals. However, since there are issues with obtaining high-quality ground truth, the problem remains.

[*]Equal contribution [1]Carnegie Mellon University, Pittsburgh, USA. Correspondence to: Louis-Philippe Morency <morency@cs.cmu.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

Another problem with the Hipster dataset is that it's relying on user annotations for a small number of very dissimilar classes. To solve this problem, Takagi et al. (2017) raised a new dataset for evaluation called FashionStyle14. It's focusing on more complicated classes with large variability and expert-curated annotations. The FashionStyle14 dataset has collected 13,126 images of 14 styles. They then tried several image models to do the classification task. The result shows that ResNet50 gets the best score, but still far from the performance of savvy users.

Han et al. (2017) realized that both images and semantic descriptions are critical to style. They raised a new dataset called Fashion200K, which contains over 200,000 images with their semantic descriptions. Trying to get a visual-semantic embedding, the model aims at minimizing the distance between features after CNN and bag-of-word embedding of semantic descriptions. Their result shows that a visual-semantic representation can cluster attributes into multiple groups to form spatially-aware concepts.

2.2. Multimodal Image Retrieval

Our work is situated among literature in multimodal image retrieval (Datta et al., 2008), which has been extensively studied over decades. Specifically, we build on recent works which have explored a combination of state-of-the-art machine learning methods and large-scale image datasets available on the web to train powerful models with good generalization capability. One example is CLIP (Radford et al., 2021b), a model that has learned a mapping between images and text from a dataset of 400 million image-text pairs through contrastive training. The model has demonstrated impressive zero-shot capabilities for many downstream tasks. In our work, we plan to build on pre-trained CLIP encoders to form our image and text representations. In addition, it may be interesting to use CLIP as a baseline model. Since CLIP was not trained specifically on IKEA furniture, our model should ideally outperform it.

2.3. Image Retrieval in Other Domains

Image retrieval with stylistic similarity can be well illustrated in the domain of fashion products. There are various prior works focusing on learning fashion characteristics and representations. Vaccaro et al. (2016) predicts high-level fashion style attributes such as tropical, exotic, effortless, radiant, and flowy from low-level design element languages such as color, material, and silhouette using polylingual topic modeling. Liao et al. (2018) design an end-to-end explainable image and text representation learning pipeline guided by a Exclusiveness-Independence Tree (EI-Tree) data structure that captures the multi-class and multi-label hierarchical relationships of fashion concepts. Guo et al. (2021) proposes a new fashion dataset and a interactive user simulator that seamlessly integrate explicit visual attributes and conversational user feedback for image retrievers.

Image retrieval applications in other domains can also be inspiring to our projects. As mentioned in Sharma et al. (2019), users get floor plan recommendations by either existing printed floor plan figures or manual sketch figures. All these image retrieval applications in other domains present us with inspirations in neural network architectures as well as evaluation metrics.

2.4. Interior Design Recommendation

Although the obvious application of furniture recommendation is for everyday home decorators, the initial forays into the problem of automatic style-compatible furniture recommendation were built for 3D modelers. For example, Liu et al. (2015) developed an interactive room-building tool for scene designers, augmented with style-compatible furniture suggestions based on 3D object models. They collected style compatibility data from Mechanical Turk workers, then trained an embedding to convert handcrafted features of the 3D models (such as the curvature of a chair arm) to a style feature space. Interestingly, they observed that only a small minority of compatibility pairs (although a larger fraction than random) exhibited strong agreement among crowd workers, indicating that most furniture item pairs may occupy a gray area of compatibility when labeled by non-experts. More recently, Weiss et al. (2020) developed a similar recommendation-augmented 3D modeling system, this time using Siamese networks trained on images labeled by interior-design experts. They quantified style according to four predominant design trends (modern, traditional, cottage, and coastal), which improved interpretability but may have constrained the expressiveness of the embeddings learned.

Meanwhile, early efforts in furniture recommendation from the machine-learning community focused on capturing notions of similarity through large amounts of image data. A foundational effort in this vein was put forward by Bell & Bala (2015), which trained furniture image embeddings using a Siamese CNN optimized with a contrastive loss. They scraped a dataset of around 14 million product and room photos from `houzz.com`, recruited MTurk workers to label the bounding boxes of featured products in each room, then trained the network to identify whether two images represented the same piece of furniture. Their results clearly benefit from the volume of available data; however, their model does not capture stylistic similarity across different types of furniture.

The approaches described above either require *a priori* style knowledge, or replace style with a more readily-learned concept such as visual similarity; to overcome these drawbacks, a small number of researchers have looked to mul-

timodal information sources. Most closely related to our work, Tautkute et al. (2017a) propose a multimodal search engine where users may input a scene and a text query to retrieve fitting furniture products. Their approach learns two separate embeddings of furniture, one based on visual features extracted using a CNN and one based on textual product descriptions. They propose a simple "blending" technique to re-rank textual and visual results based on visual features, which leads to an 11% increase in co-occurrence probability of the retrieved results. In contrast, our work specifically learns for the stylistic compatibility between furniture products.

2.5. Work Closely Related to Our Proposal

Among all related work, except for CLIP (Radford et al., 2021b) and the original paper for the IKEA dataset (Tautkute et al., 2017a), which we have already described in detail, there are two other papers that are closely related to our proposed work in terms of data, method, and evaluation metrics. As an extension to their previous work on the IKEA dataset, which is mainly based on late fusion of candidates from different modalities, in Tautkute et al. (2019) the authors propose a new DeepStyle Siamese network with early fusion scheme and contrastive loss to perform better representation learning and ranking. Aggarwal et al. (2018) also learns style compatibility of furniture with Siamese networks, using the Bonn Furniture Styles dataset containing furniture tagged with 17 style categories. We differentiate our work by making use of IKEA rooms designed by professional interior designers as our ground truths for the task of furniture style compatibility, which requires new machine learning methods.

3. Problem Statement

Given two furniture F_1 and F_2 as input, our objective is to determine whether they are compatible or incompatible. In our dataset, we define F_1 and F_2 as compatible if they have at least one co-occurrence within our set of IKEA designer rooms R.

$$\{F_1, F_2\} \subset R_k \text{ and } R_k \in R \tag{1}$$

where $k \in \{1, ..., n\}$ and n is the number of rooms. In other words, F_1 and F_2 appear together in a room R_k. We define F_1 and F_2 as incompatible if the condition is vice versa.

Given this definition, we compute a compatibility label $y \in \{0, 1\}$ for all possible furniture pairs in our dataset, where $y = 1$ is a compatible pair and $y = 0$ is an incompatible pair. We then try to predict y by learning a compatibility function C, which represents a distance measure between F_1 and F_2.

$$\hat{y} = C(F_1, F_2) \tag{2}$$

Since y is binary, we employ a cross-entropy loss $\mathcal{L}$ to learn C.

$$\mathcal{L}(B) = \sum_{F_1,F_2,y \in B} y \log(C(F_1, F_2)) + (1 - y) \log(1 - C(F_1, F_2)) \tag{3}$$

where B is a training batch.

4. Multimodal Baseline Models

As described in the problem statement section, we formulate the furniture search problem as a binary classification problem. During training time, two item images and their corresponding descriptions are passed as inputs and a binary label indicating whether they are matched is conceived as output. All three models use the same training and validation dataset to ensure fairness in comparison and consistency for reproduction. The training hyperparameters include learning rate, batch size, number of hidden layers for LSTM, dropout probability, and word embedding dimensions.

During inference time, we extend the binary classification problem to a search problem similar to information retrieval, where only the image and description of one piece of furniture is given and a scoring matrix is computed by evaluating the similarity between this input furniture and all furniture in the dataset. The output is a ranked list of the furniture ordered by similarity scores. Standard IR ranking metrics such as precision, recall, NDCG are utilized to quantitatively evaluate the ranking results. furniture is given and a scoring matrix is computed by evaluating the similarity between this input furniture and all furniture in the dataset. The output is a ranked list of the furniture ordered by similarity scores. Standard IR ranking metrics such as precision, recall, NDCG are utilized to quantitatively evaluate the ranking results.

4.1. CNN + LSTM

The first baseline model is a CNN-LSTM network following a two-tower model architecture. Our CNN component takes in an RGB image of a furniture as input and extracts a 512-dimensional feature embedding as output. We use four convolutions blocks, each with a 3x3 convolution layer and a 2x2 max-pooling layer, and two fully-connected layers. We apply batch normalization and ReLU activation after the convolutional layers and sigmoid activation after the fully-connected layers. The LSTM network takes the Word2Vec (Mikolov et al., 2013) embedding matrix as input, passes it to a two-layer LSTM layer, and generates 128-way feature vectors in a fully-connected layer with ReLU activation. Since we have two images and two texts as inputs, there are four feature vectors generated for both furniture, in image and text embedding spaces respectively. The similarity score of furniture F_1 and furniture F_2 is calculated by running a dense layer on the vector concatenated by the four embeddings. Let's denote CNN image embedding extractor

as *I*, LSTM text embedding extractor as *T*, the concatenation layer as *C*, the final dense layer to compute sigmoid probability as *D*, the output matching score of the two furniture *S*, the mathematical representation of the final layer can be illustrated in the following equation:

$$S(F_1, F_2) = D(C(I(F_1), I(F_2), T(F_1), T(F_2))) \quad (4)$$

The best test performance is achieved by the following training hyperparameters: we adopt the learning rate of 2e-4, the dropout probability of 0.3, the LSTM hidden dimension as 128, and the number of epochs to be 7.

4.2. Siamese Network

Our second baseline model is a Siamese network with twin branches (Koch et al., 2015). We experiment with a metric learning approach (Kulis, 2012) to learn a similarity function between two furniture F_1 and F_2 in a contrastive manner. Each branch resembles our first baseline model of CNN and LSTM with early fusion.

The CNN image embedder and LSTM text embedder part is very similar to that of the first baseline model with modifications in number of layers and type of layers.

Next, we concatenate the CNN and LSTM embeddings into a 640-dimensional embedding and pass it through two fully-connected layers, yielding the output embedding of the branch. We then compute the weighted L1-distance D between the two branch embeddings.

$$D(F_1, F_2) = |f(F_1) - f(F_2)| \quad (5)$$

Finally, we translate the distance function D into the compatibility function C, which predicts the probability that F_1 and F_2 are compatible, by passing it through a fully-connected layer with learned weights $\mathcal{W}$ and a sigmoid activation.

$$C(F_1, F_2) = \frac{1}{1 + \exp^{-\mathcal{W} \cdot D(F_1, F_2)}} \quad (6)$$

We use a cross-entropy loss (Equation 3) and the same best performing hyperparameters as Subsection 4.1.

4.3. CLIP

The last baseline model was generated by fine-tuning a pretrained state-of-the-art multimodal model called CLIP (Contrastive Language-Image Pre-Training) (Radford et al., 2021a). CLIP was pretrained on a large dataset of about 400 million image-text caption pairs, with the goal of generating a joint multimodal embedding space for both images and text. The model generates embeddings for each image (we used the variant incorporating a vision transformer, `ViT-B/32`) as well as each textual caption (using a basic transformer architecture). The embeddings are then evaluated based on the similarity of the correct pair of vectors compared to that of the incorrect pairs, using a multi-class N-pair loss. Namely, the loss for the i-th pair is given by

$$\mathcal{L}_i = -\log \frac{\exp \langle \mathbf{u}_i, \mathbf{v}_i \rangle / \tau}{\sum_{k=1}^{N} \exp \langle \mathbf{u}_i, \mathbf{v}_k \rangle / \tau} \quad (7)$$

where $\langle \cdot, \cdot \rangle$ represents the cosine similarity, and $\mathbf{u}_i$ and $\mathbf{v}_i$ are the embedded representations of the i-th image and text, respectively. While this loss function compares each image to all captions in the minibatch, a similar loss is also defined comparing each caption to all images.

CLIP exhibits good performance on a variety of multimodal tasks without fine-tuning. In fact, in our earlier exploration of the IKEA dataset, we observed that CLIP's image representations for this dataset captured functional and stylistic similarities much better than those of VGG16 and ResNet. Therefore, we sought to test CLIP's performance using a simple linear probe, as well as to assess whether fine-tuning CLIP specifically on style compatibility data from the IKEA dataset would improve its performance even further.

To fine-tune CLIP, we constructed a model to predict the compatibility of two furniture products given their images and text descriptions, similar to the CNN-LSTM architecture described above. Because the base CLIP model already has over 150M parameters, we opted for a simple augmentation: a linear layer with sigmoid activation, applied to the concatenated embeddings of each product's separate image and text descriptions (512 dimensions for each embedding, yielding 2,048 dimensions in total). To prevent overfitting, only the last layer of either the visual or text encoder was tuned along with training the output layer. We also trained a version that kept the encoders fixed, and only trained the linear output layer. Models were trained for 3 epochs using the Adam optimizer. A learning rate of 10^{-4} was used for the output layer, while a smaller learning rate of 10^{-6} was used for the CLIP layers to improve stability.[1]

5. Experimental Methodology

As briefly introduced in our proposal, the IKEA dataset (Tautkute et al., 2017b) consists of room and item images, item descriptions, room categories, and item categories. There are 2,300 items in 262 rooms. As described in the problem statement, our annotations are binary labels indicating whether two items are matched in style. This annotation is not explicitly presented in the original dataset. Therefore,

[1]These parameters are presented with the caveat that we were not able to conduct a thorough hyperparameter search for the purposes of this assignment. We generally observed that different hyperparameter settings resulted in either rapid overfitting or slow convergence to validation accuracy around the amount shown in Table **??**.

in the data preparation stage, we treat items within the same room as stylistically compatible. In the later sections of the report, we will elaborate on the pros and cons of this annotation strategy as well.

Since the inputs to the models are pairs represented as (F_1, F_2), the standard train-test split technique does not fit very well with our case. Namely, it is possible that (F_1, F_2) is placed in the training set while (F_3, F_1) is placed in the validation set, but F_1 has already been seen in the training process and generalization might be affected. Therefore, we split the dataset by item IDs instead of pairs, to ensure the aforementioned issue is resolved.

As indicated in the baseline section, in the training time, the results are evaluated from a binary classification's perspective, while in the inference time, the results are evaluated from an IR (Information Retrieval)'s perspective. The hyperparameter space is discussed in detail in the baseline section.

6. Results and Discussion

6.1. Metrics

To measure the effect of these baseline model, we need to find out suitable metrics. Basically, we are running a ranking problem, finding the most suitable images for one image query. Thus, we are defining the following five metrics: NDCG, Score_Eval, Precision, Recall and F-score.

NDCG, which is short for Normalized Discounted Cumulative Gain, is a measure of ranking quality. In information retrieval, it is often used to measure effectiveness of web search engine algorithms or related applications. Using a graded relevance scale of documents in a search-engine result set, DCG measures the usefulness, or gain, of a document based on its position in the result list. The gain is accumulated from the top of the result list to the bottom, with the gain of each result discounted at lower ranks.

Score_Eval is a simpler way to measure the ranking result. It is calculated by the sum of the result of top k result that shows up in the ground truth. The score represents the confidence of model to the correct top K result.

Precision, Recall and F-score is a classic metric in machine learning. Precision is the number of true positive results divided by the number of all positive results, including those not identified correctly, and the recall is the number of true positive results divided by the number of all samples that should have been identified as positive. While the F1 score is the harmonic mean of the precision and recall. The highest possible value of an F-score is 1.0, indicating perfect precision and recall, and the lowest possible value is 0, if either the precision or the recall is zero.

6.2. Quantitative Results

We ran these five metrics on our three models and the quantitative results are available in Table 1. Note that all five ranking metrics are computed using $K = 10$, and products are retrieved from the 718 products in the validation set only. This ensures that models did not memorize product relationships from the training phase.

The results show that the fine-tuned CLIP models achieve the highest accuracies on the binary style compatibility task, as one would expect considering their large number of parameters and pretraining datasets. On the other hand, all five models perform extremely poorly on the ranking evaluation task. While the CLIP model with the fine-tuned image embedding layer performs the best, none of them perform as well as we would expect. For example, even a simple nearest-neighbors retrieval using pretrained CLIP embeddings performs considerably better than any of the models (NDCG: 0.2389, F-score 0.0886). We investigate and discuss reasons for this discrepancy and possible improvements in the following sections.

6.3. Qualitative Examples and Error Analysis

In Fig. 1, we include several mis-predicted examples to more intuitively illustrate our error analysis. As discussed above, the performance of all three baseline models is much lower than our expectation. We have performed a significant amount of model tuning, such as varying the embedding methods, number of CNN and LSTM layers, and optimization parameters, but the improvement is still negligible. Given that all three baselines contain a "two-tower" architecture where the image and the text are encoded separately before fusion, it is rather easy to change the encoders. However, currently we do not plan to add more complex attention blocks / transformers and pretrained large-scale CNN image feature extractors to the model. The reason is that the performance of CLIP, a rather complex pretrained SoTA architecture, already illustrates that the problem might not come from the modeling side. Based on these findings, we start troubleshooting the dataset side.

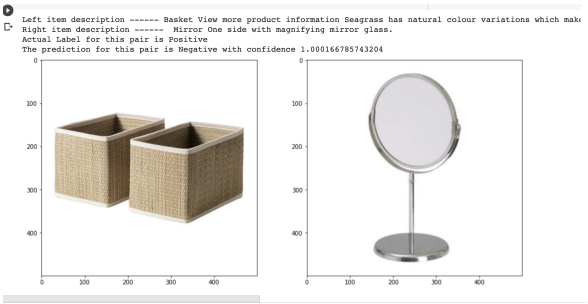
Fig. 1 shows several bad predictions of the three baselines to showcase potential flaws of the dataset. From Figure 1a to Figure 1f, all these pairs are labeled positive simply because they are placed in the same room. It is obvious that these pairs are not matched from not only style's or but also appearance's perspectives. It is unreasonable to expect the models to learn if we as humans cannot agree on these matches.

After talking about the image pair's perspective, the quality of the text pairs are also concerning. As shown in Figure 1a to Figure 1c, some labeled positive text description pairs such as *"Box, set of 3..."* and *"Chest with two drawers*

	CNN-LSTM	Siamese	CLIP (text)	CLIP (image)	CLIP (linear probe only)
Train Acc	0.60	0.61	**0.629**	0.628	0.615
Test Acc	0.52	0.50	**0.584**	0.582	0.580
NDCG	0.0010	0.0020	0.0045	**0.0062**	0.0031
Score	0.0010	0.0020	0.0034	**0.0047**	0.0043
Precision	0.0060	0.0060	0.0060	**0.0080**	0.0080
Recall	0.0022	0.0024	0.0025	**0.0035**	0.0031
F-Score	0.0031	0.0031	0.0035	**0.0048**	0.0044

Table 1. Quantitative measurements of the three baseline models. The three CLIP variants denote versions in which the text embedding is fine-tuned, the image embedding is fine-tuned, and the embeddings are fixed (linear probe only), respectively.

(a) Bad example 1 by CNN-LSTM

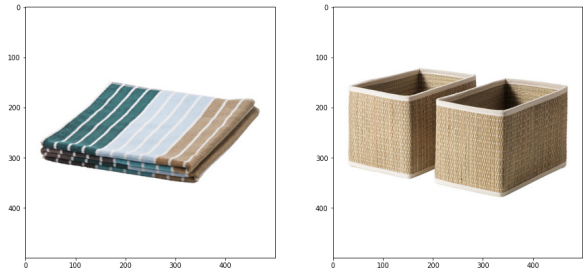


(b) Bad example 2 by CNN-LSTM

(c) Bad example 3 by CNN-LSTM

(d) Bad example 1 by Siamese Network

(e) Bad example 2 by Siamese Network



(f) Bad example 3 by Siamese Network

Figure 1. Examples of errors by each baseline model.

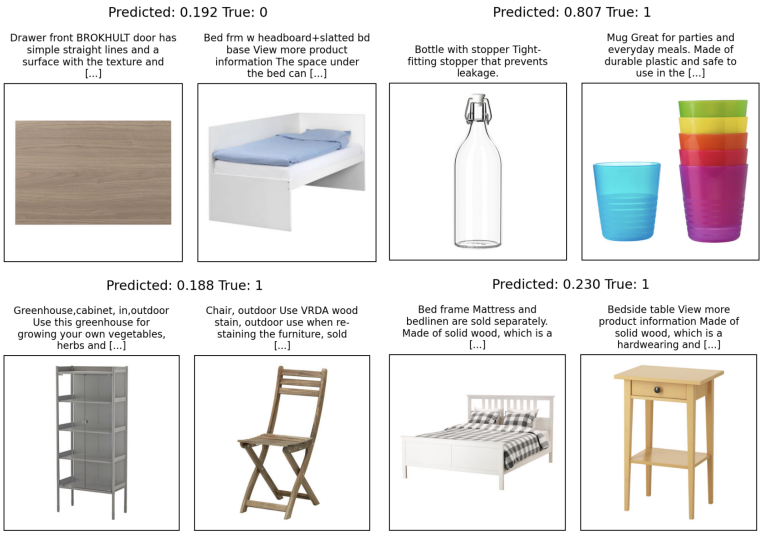


Figure 2. Correct (top) and incorrect (bottom) predictions by fine-tuned CLIP.

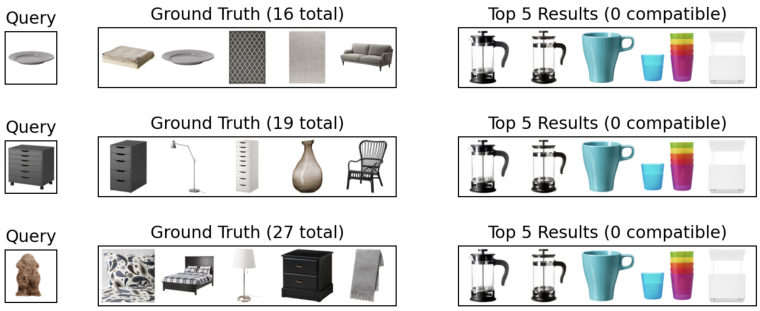


Figure 3. Top-ranked compatible furniture by CLIP for three random examples from the test set.

integrated damper catches...." also present no similarity intuitively and might not contribute to the matching as well. In addition, as we discussed in the unimodal analysis report, the descriptions in the IKEA dataset seem to be served as a complementary element to the image and do not possess much informative information if treated independently.

Another possible reason for the models' poor performance, particularly in the ranking evaluation task, could be the discrepancy between the binary compatibility task and the ranking process. For example, in Fig. 3 we see that the model retrieves the same top 5 kitchen-related products for almost all queries, despite having reasonable performance on binary compatibility (see Fig. 2). We hypothesize that the two-tower architecture may make it easy for the models to memorize certain products or product categories, particularly those that appear in many rooms with a variety of different objects. These products would then boost the activation value of the final combining layer of the network, *regardless* of what product it is being compared against.

7. New Research Ideas

7.1. Late Fusion

All our baseline models are currently employing a early fusion mechanism where the representations of the image pairs and text pairs are first computed individually and then fused together. We plan to experiment with late fusion where the sigmoid matching probabilities of the images and the texts are first generated independently and then summed using specific learn-able weights. The intuition behind this idea is that late fusion allows us to understand unimodal performance better with more explainability. Given the current results, it would be time consuming to directly analyze the joint representation space of both modalities.

7.2. Data Augmentation

Another important reason that our model cannot perform well is the limited dataset. The current dataset composed of 2193 product photos and text descriptions in 298 rooms,

which is far more less than the dataset that the main-stream pre-training model needs. One technique we can perform is called Data Augmentation(Shorten & Khoshgoftaar, 2019). It includes a lot of useful ways to increase meaningful datapoints and thus increase the model effect. We can use two ways of data augmentation in our context. The first is to perform some transformation on our current images, such as rotation, reflection, fliping, scaling, or adding some random noise. On the language side, we can use translation. For example, translating the description to Germany and translating it back will generates a similar description which has the same semantics but consists of different words. In this way, we can make the model more robust. Another way is to collect more similar images on the Internet and expand the dataset manually.It's important to notice that in this way, we need to especially take care that the distribution of new dataset should be consistent with the original one, otherwise we will easily fall into the trap of data drift.

7.3. PU Learning

We currently define furniture pairs that do not have co-occurrence as incompatible. However, this may not necessarily be true, but rather, their compatible may be simply *unknown*. Positive unlabeled learning (PU learning) (Bekker & Davis, 2020) is an approach that is suitable for datasets with relatively little positive labels and large quantities of unknown labels. We may perform PU bagging (Mordelet & Vert, 2014) by creating an ensemble of weak classifiers in parallel. Each classifier takes in all positives (1) and randomly samples subsets of unknowns (0) to create a balanced training set. After training our ensemble of weak classifiers, we can then apply the classifiers to Out-of-Bag (OOB) samples (i.e. unknown samples that were not included in their training sets) and repeat this multiple times to assign labels to unknown samples by averaging its OOB scores.

7.4. Soft Labels

As discussed in the error analysis section, we currently formulate our task as a binary classification problem, where compatibility labels of 1 or 0. However, some furniture pairs may co-occur more frequently than others (i.e. appear together in multiple rooms). Should we give more weight to these pairs that are "more compatible"? Instead of predicting a hard label of 0 or 1, we may try predicting a soft label, such as 0.3 or 0.7. We may compute these soft labels with point-wise mutual information (PMI) (Church & Hanks, 1990), which can represent the probability of two furniture co-occurring $P(F_1, F_2)$ compared to the probability of each element occurring separately $P(F_1)$ and $P(F_2)$.

$$PMI(F_1; F_2) = \log \frac{P(F_1, F_2)}{P(F_1)P(F_2)} \tag{8}$$

PMI is also designed such that it penalizes highly popular furniture, which may co-occur with a lot of other furniture but may not necessarily be compatible with all of them. Likewise, less popular furniture with high co-occurrence may indicate a solid signal of compatibility. Nonetheless, if we change our hard labels into soft labels such as those generated with PMI, we would need to use a loss function for regression, such as mean squared error (MSE), instead.

7.5. Co-Learning with Room Images

An important distinguishing characteristic of the IKEA dataset is that it not only contains images of isolated products, but also images of those products in expert-designed room scenes. Currently, our modeling approach has only used the room images as binary evidence of stylistic compatibility between products. However, it is likely that these images contain more valuable information about *how* products are stylistically related. For example, two products may not be visually similar to each other, but conditioned on other objects in the room, they become compatible. This presents a viable opportunity for co-learning (Zadeh et al., 2020), either through passing the scene images directly to the network as input or by extracting more granular information about product co-occurrence from the scene composition.

References

Aggarwal, D., Valiyev, E., Sener, F., and Yao, A. Learning style compatibility for furniture. In *German Conference on Pattern Recognition*, pp. 552–566. Springer, 2018.

Bekker, J. and Davis, J. Learning from positive and unlabeled data: A survey. *Machine Learning*, 109(4):719–760, 2020.

Bell, S. and Bala, K. Learning visual similarity for product design with convolutional neural networks. *ACM Transactions on Graphics*, 34(4), 2015.

Blijlevens, J., Creusen, M. E. H., and Schoormans, J. P. L. How Consumers Perceive Product Appearance: The Identification of Three Product Appearance Attributes. *International Journal of Design*, 3(3):27–35, 2009.

Church, K. and Hanks, P. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29, 1990.

Datta, R., Joshi, D., Li, J., and Wang, J. Z. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (Csur)*, 40(2):1–60, 2008.

Guo, X., Wu, H., Gao, Y., Rennie, S. J., and Feris, R. S. Fashion iq: A new dataset towards retrieving images by natural language feedback. In *CVPR*, 2021.

Han, X., Wu, Z., Huang, P. X., Zhang, X., Zhu, M., Li, Y., Zhao, Y., and Davis, L. S. Automatic Spatially-Aware Fashion Concept Discovery. *Proceedings of the IEEE International Conference on Computer Vision*, 2017-October:1472–1480, 2017. ISSN 15505499. doi: 10.1109/ICCV.2017.163.

Kiapour, M. H., Yamaguchi, K., Berg, A. C., and Berg, T. L. Hipster Wars: Discovering elements of fashion styles. *European Conference on Computer Vision*, 2014.

Koch, G., Zemel, R., Salakhutdinov, R., et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015.

Kulis, B. Metric learning: A survey. *Foundations and Trends in Machine Learning*, 5, 01 2012. doi: 10.1561/2200000019.

Liao, L., He, X., Zhao, B., Ngo, C.-W., and Chua, T.-S. Interpretable multimodal retrieval for fashion products. In *Proceedings of the 26th ACM International Conference on Multimedia*, MM '18, pp. 1571–1579, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356657. doi: 10.1145/3240508.3240646. URL https://doi.org/10.1145/3240508.3240646.

Liu, T., Hertzmann, A., Li, W., and Funkhouser, T. Style compatibility for 3D furniture models. *ACM Transactions on Graphics*, 34(4):1–9, 2015. ISSN 15577368. doi: 10.1145/2766898.

Mikolov, T., Chen, K., Corrado, G. S., and Dean, J. Efficient estimation of word representations in vector space. In *ICLR*, 2013.

Mordelet, F. and Vert, J.-P. A bagging svm to learn from positive and unlabeled examples. *Pattern Recognition Letters*, 37:201–209, 2014.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning Transferable Visual Models From Natural Language Supervision. *arXiv*, 2021a. URL http://arxiv.org/abs/2103.00020.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. In *ICML*, 2021b.

Sharma, D., Gupta, N., Chattopadhyay, C., and Mehta, S. A novel feature transform framework using deep neural network for multimodal floor plan retrieval. *International Journal on Document Analysis and Recognition (IJDAR)*, 22:417 – 429, 2019.

Shorten, C. and Khoshgoftaar, T. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6, 07 2019. doi: 10.1186/s40537-019-0197-0.

Takagi, M., Simo-Serra, E., Iizuka, S., and Ishikawa, H. What Makes a Style: Experimental Analysis of Fashion Prediction. *Proceedings - 2017 IEEE International Conference on Computer Vision Workshops, ICCVW 2017*, 2018-January:2247–2253, 2017. doi: 10.1109/ICCVW.2017.263.

Tautkute, I., Mozejko, A., Stokowiec, W., Trzcinski, T., Brocki, L., and Marasek, K. What looks good with my sofa: Multimodal search engine for interior design. *Proceedings of the 2017 Federated Conference on Computer Science and Information Systems, FedCSIS 2017*, pp. 1275–1282, 2017a. doi: 10.15439/2017F56.

Tautkute, I., Możejko, A., Stokowiec, W., Trzciński, T., Łukasz Brocki, and Marasek, K. What looks good with my sofa: Multimodal search engine for interior design. In Ganzha, M., Maciaszek, L., and Paprzycki, M. (eds.), *Proceedings of the 2017 Federated Conference on Computer Science and Information Systems*, volume 11 of *Annals of Computer Science and Information Systems*, pp. 1275–1282. IEEE, 2017b. doi: 10.15439/2017F56. URL http://dx.doi.org/10.15439/2017F56.

Tautkute, I., Trzciński, T., Skorupa, A. P., Łukasz Brocki, and Marasek, K. Deepstyle: Multimodal search engine for fashion and interior design. *IEEE Access*, 7:84613–84628, 2019.

Vaccaro, K., Shivakumar, S., Ding, Z., Karahalios, K., and Kumar, R. The elements of fashion style. In *Proceedings of the 29th annual symposium on user interface software and technology*, pp. 777–785, 2016.

Weiss, T., Yildiz, I., Agarwal, N., Ataer-Cansizoglu, E., and Choi, J. W. Image-Driven Furniture Style for Interactive 3D Scene Modeling. *Computer Graphics Forum*, 39(7):57–68, 2020. ISSN 14678659. doi: 10.1111/cgf.14126.

Zadeh, A., Liang, P. P., and Morency, L. P. Foundations of Multimodal Co-learning: Multimodal Co-learning. *Information Fusion*, 64(June):188–193, 2020. ISSN 15662535. doi: 10.1016/j.inffus.2020.06.001.