Décor: Multimodal Retrieval for Interior Design – Final Report

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

Abstract

The rising 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 sought to estimate the stylistic compatibility of furniture products based on their description and visual appearance, obtained from the little-studied IKEA dataset by Tautkute et al. (2017a). We developed three approaches that expand upon state-of-the-art baselines: intermediate fusion with convolutional instance statistics, multimodal variational autoencoders, and co-learning with room scene images. All three models performed better than the naive CLIP baseline, although fine-tuned CLIP remains strongest for both classification and ranking. These results suggest the potential of combining intermediate fusion or co-learning approaches with massive pre-training to further improve performance.

1. Introduction

Imagine you are on an e-commerce website, shopping for furniture to furnish your new home. You just bought a luxurious solid oak dining table. Now, you're thinking: Which set of chairs should I buy to match my beautiful table? Machine learning algorithms have achieved state-of-the-art performances for a variety of object similarity tasks, often exploiting multimodal cues to improve accuracy and granularity. However, most existing approaches have explored similarities with regards to the direct visual or textual appearances of objects. In contrast, many tasks, such as the stylistic compatibility of furniture, require much more abstract conceptualizations.

The capability of learning the stylistic compatibility of objects has applications to 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 in this domain 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 such 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 a self-supervised manner by leveraging IKEA furniture placed in the context of designer-curated showrooms. We use the IKEA dataset (Tautkute et al., 2017a), which contains 2,193 product photos and text descriptions. Most interestingly, the dataset also includes 298 rooms, designed by IKEA designers, as well as information on which products appear within which rooms. Our key 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. 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 used pre-trained as well as fine-tuned CLIP encoders as our strong baselines.

[*]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.

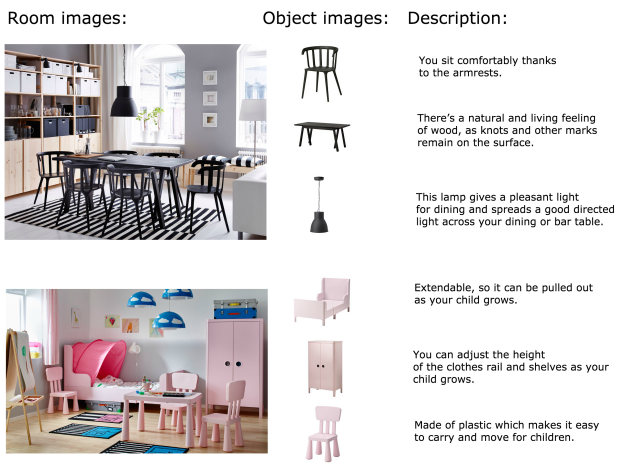

Figure 1. Example items in the IKEA dataset. Figure courtesy of https://github.com/IvonaTau/ikea.

2.2. 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 used features such as RGB value, MR8 texture response, distance from image border, etc. to measure style; however, because the representations were obtained by crowd-sourcing, the quality of the ground-truth may not be optimal. Another issue was its reliance on user annotations for a small number of very dissimilar classes, which motivated the creation of the FashionStyle14 dataset (Takagi et al., 2017). This dataset focuses on 14 more complicated classes with large variability and expert-curated annotations. Similarly, Han et al. (2017) created a 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 distance between features based on 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.

2.3. 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 which quantified style according to four predominant design trends (modern, traditional, cottage, and coastal). This led to improved interpretability, but may have constrained the expressiveness of the embeddings learned.

Meanwhile, early efforts in furniture recommendation from the ML 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. Their results benefited from having a dataset of around 14 million images; however, their model does not capture stylistic similarity across different types of furniture.

A small number of researchers have looked to multimodal information sources to avoid the need for *a priori* style knowledge and visual similarity. For example, Tautkute et al. (2017a) proposed 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. As an extension to their previous work, in Tautkute et al. (2019) the authors propose a new DeepStyle Siamese network with early fusion and a 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. In contrast to these efforts, our work specifically learns for the stylistic compatibility between furniture products with the context of designer-curated rooms as our ground truths.

3. Proposed Approaches

3.1. Model Description

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 calculate the predicted label $\hat{y} \in [0, 1]$ by learning a compatibility function C, which represents a distance measure between F_1 and F_2.

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

The ultimate purpose of this compatibility function is to rank furniture items from a set of products $\mathcal{F}$, where the retrieved items will have maximal values of $C(F_1, F_2)$.

3.2. Intermediate Fusion with Convolutional Instance-Level Statistics

While existing models for multimodal retrieval (e.g. (Radford et al., 2021a; Tautkute et al., 2017a)) typically embed the image and text separately, our first approach proposes to successively combine signals from each modality in an intermediate fusion architecture. Our idea is motivated by the observation underlying adaptive instance normalization (Huang & Belongie, 2017): the *summary statistics* of a convolutional layer's activations over all positions in a given image can convey information about the *style* of the image. Indeed, in qualitative comparisons of the embedding spaces generated by different layers of the VGG16 network, we found that earlier layers embedded product images with similar colors and textures while later layers were more closely clustered by product type. This led us to formulate an architecture that incorporated these image statistics instead of the final VGG16 embeddings, as shown in Fig. 2.

In this architecture, the embedding is generated by a series of multi-headed self-attention layers. The first set of inputs is the product description, transformed into a 300-dimensional distributed representation by a pre-trained word2vec model. Additionally, the product image is passed through the first set of convolutional layers of VGG16, and the means and variances of each kernel across the image are transformed into another 300-dimensional vector and prepended to the text inputs. At each successive attention layer, another set of convolutional layer statistics from VGG16 is prepended to the input. (For our experiments, we use only two attention layers to avoid overfitting on the small IKEA dataset.)

The architecture described above produces a joint 128-dimensional representation $z(F)$ for an image and description of a single product F. In order to perform the training task (binary classification for a pair of products), the representations for products F_1 and F_2 are concatenated and passed through a two-layer MLP g and a final sigmoid unit:

$$C_{\text{IF}} = \sigma\left(g(z(F_1) \oplus z(F_2))\right) \tag{3}$$

where $\oplus$ indicates vector concatenation. We experimented with performing a simple cosine similarity on the output representations, but the MLP appeared to work better at separating functionally similar products that were not likely to co-occur (although it overfits more easily).

To train this network, we apply a binary cross-entropy loss to the sigmoid output:

$$\mathcal{L}_{BCE} = y \log(\hat{y}) + (1 - y) \log(1 - \hat{y}) \tag{4}$$

The network is then trained using the Adam optimizer with a learning rate and weight decay parameter both equal to 10^{-5}. The training is terminated when the validation loss stops increasing, which generally occurs around 4-6 epochs.

3.3. Multimodal Variational Autoencoder

In our second approach, we experiment with whether we can determine stylistic compatibility without learning this task explicitly, but simply by using better representations. We focus on unsupervised representation learning with a multimodal variational autoencoder (MVAE) and use the pretext task of image reconstruction. Our MVAE architecture is shown in Figure 3, which consists of a probabilistic encoder, a sampled latent vector, and a probabilistic decoder. Our hypothesis is that the sampled latent vector will capture elements of style.

Our encoder takes in a furniture image and a furniture text description as inputs. For the image, we encode it through two convolutional layers followed by two separate linear layers. One linear layer produces a mean vector and the other produces a variance vector. For the text, we encode it through two linear layers followed by two separate linear layers, which also produce a mean vector and a variance vector. We apply ReLU activation after all our layers. We then concatenate the image and text means and concatenate the image and text variances. Given the mean and variance vectors of the furniture, we randomly sample a point within the distribution for the decoder. Our decoder consists of a linear layer followed by two convolutional layers to output a reconstruction our original image. We apply ReLU acti-

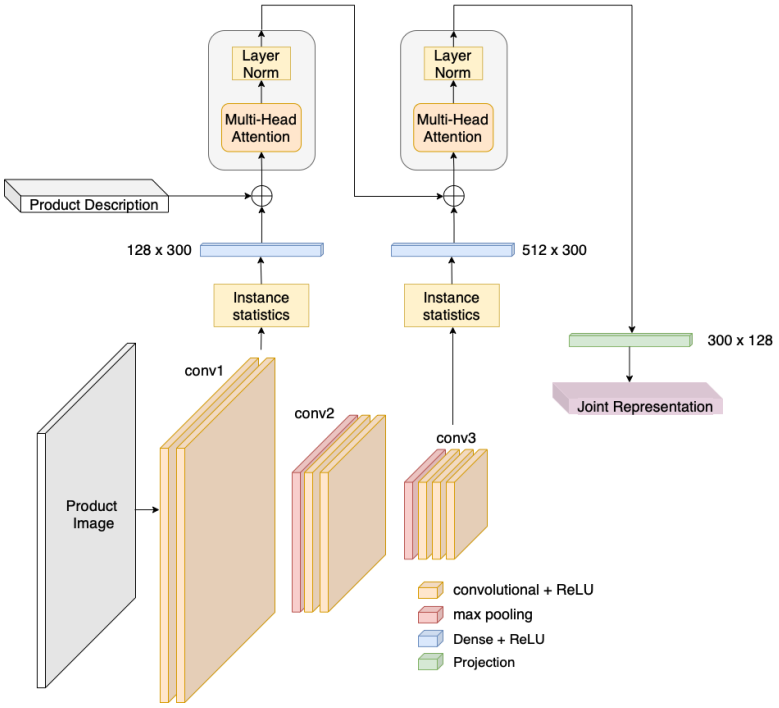

Figure 2. Architecture of the intermediate fusion module, which is applied to each product (image and text description) individually. The numbers next to Dense and Projection layers denote the input and output vector dimensions per instance.

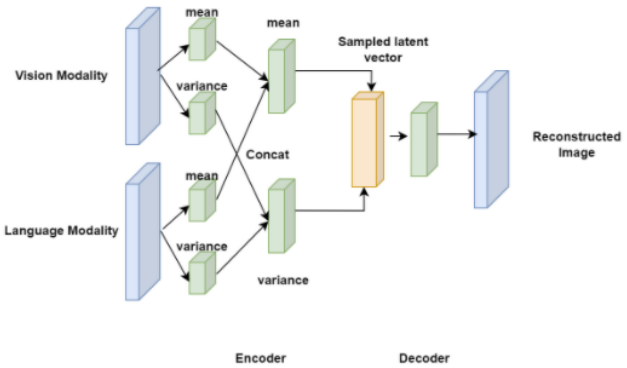

Figure 3. Architecture of MVAE

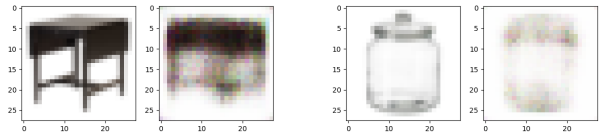

Figure 4. Examples of MVAE reconstructions. Note that the reconstructions capture the essence of the input distribution rather than being exact.

vation after the first two layers and Sigmoid activation after the final layer.

To train our MVAE, our loss function is a combination of a Binary Cross-Entropy (BCE) Loss (Equation 4) and a Kullback–Leibler (KL) Divergence Loss:

$$\mathcal{L}_{KLDivergence} = -1 - \log(\sigma) + \mu^2 + \sigma^2 \qquad (5)$$

BCE Loss computes the pixel-to-pixel difference between the reconstructed image and the original image and KL Divergence Loss computes the similarity of our two distributions, assumed to be normal. We adopt a learning rate of 10^{-4} using Adam optimization, a batch size of 128, and train for around 150 epochs. Examples reconstructions are shown in Figure 4.

Finally, we build a simple binary classifer for our orignal task of determining compatibility between two furnitures. We used our pre-trained MVAE as the encoder, extracting and concatenating the sampled latent vectors for each furniture, and passing the concatenated vector through three linear layers. We apply ReLU activation after the first two layers and Sigmoid activation after the final layer. To train our binary classifier, our loss function is a BCE Loss (Equation 4). We adopt a learning rate of 10^{-4} using Adam optimization, a batch size of 16, and train for around 10

epochs.

3.4. Co-Learning with Room Images

Although we already use room image as an implicit self-supervised signal for matching objects, we hypothesize that this signal is not strong enough for the model to capture the stylistic dependencies between objects given the low ranking scores for the baseline model. Room images can be viewed as "resource-rich" since there are much more object relationship information for us to mine while object images can be viewed as "resource-poor" for the lack of this information. Therefore, following this definition of resource-poor and resource-rich characteristics of data, co-learning becomes natural in solving this issue. Instead of treating room images as implicit signals, we explicitly put room images as an input to the model during the training process to reinforce the matching relationships. Our co-learning architecture is shown in Figure 5, which simply adds a branch (colored in red) for room images. We build this architecture using CLIP encoder for both text and image.

However, there are two upfront challenges for this approach: first, from an engineering perspective, we have the discrepancy between training and testing where there is no room image input during test time and ranking time. This challenge can be easily resolved by either adding an white/blank image, or use the mean image of all training room images, or use a random noise image during inference time. The second challenge is related to how to cope with negative pairs during training. It is easy to use the substitute images mentioned above (e.g., white, mean, random noise) for negative pairs, but the model is also likely to learn that negative classification results has strong correlation with these substitute images, thus making every inference result to be negative as well. In other words, we do not want the model to rely too much on the room image during training and direct the learning to the wrong path. To resolve this issue, during the first few (e.g., 5 to 10) epochs, we train the model with actual room images and substitute images, but in the later epochs, we only input substitute images no matter whether the pair is positive or negative. The hypothesis behind this training scheme is that the model should be able to capture enough self-supervision signals in the previous epochs and the role of later epochs is to reduce unnecessary reliance and help the model the generalize at inference time.

4. Experimental Setup

4.1. Dataset and Sampling Procedures

As briefly introduced above and depicted in Fig. 1, the IKEA dataset (Tautkute et al., 2017b) consists of room and item images, item descriptions, room categories, and item categories. As described in the problem statement, our annotations are binary labels indicating whether two items are matched in style; however, this annotation is not explicitly presented in the original dataset. Therefore, in the data preparation stage, we treat *items within the same room* as stylistically compatible.

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. 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. Therefore, we split the dataset by item IDs instead of pairs, to ensure that the model has never seen any validation products in the training set.

Positive-Unlabeled Learning A challenge with this binary self-supervised formulation is that we have no hard negative examples, and many pairs of products may be compatible yet never seen in the same room. To overcome this, we experimented with applying a positive-unlabeled approach to the dataset generation. In particular, we sample three different training datasets, each one containing all positive examples and an equal number of randomly-sampled negative examples.

Weighted Negative Sampling While developing the current models, we found that despite using a downstream classifier, the models still predicted the positive class for products in the same category (e.g. two coffee tables). To encourage the model to learn that these products cannot co-occur, we first computed the Jaccard index J between the names of each product as a similarity metric. Then, we sampled negative product pairs such that given a product F_1, the likelihood of sampling another product F_2 was

$$p(F_2|F_1) = \frac{J(F_2, F_1)}{\sum_{F_i \in \mathcal{F}} J(F_i, F_1)}. \tag{6}$$

Samples that were found to be part of the positive pair set were discarded.

Upsampling Positive Examples Finally, we noticed that some models could memorize certain products and always predict the same value when those products are present, decreasing the model generalizability. For example, a product that appears with many other products will likely have more positive examples than sampled negative ones, so the model can always predict the positive class. Since we had already imposed a product distribution on the negative sampling, another approach was necessary to ensure that the positive examples matched it while preserving as many examples as possible.[1] We therefore implemented a greedy algorithm

[1]Rejection sampling would also be possible, but would have been computationally slow because the proportion of positive examples is quite small.

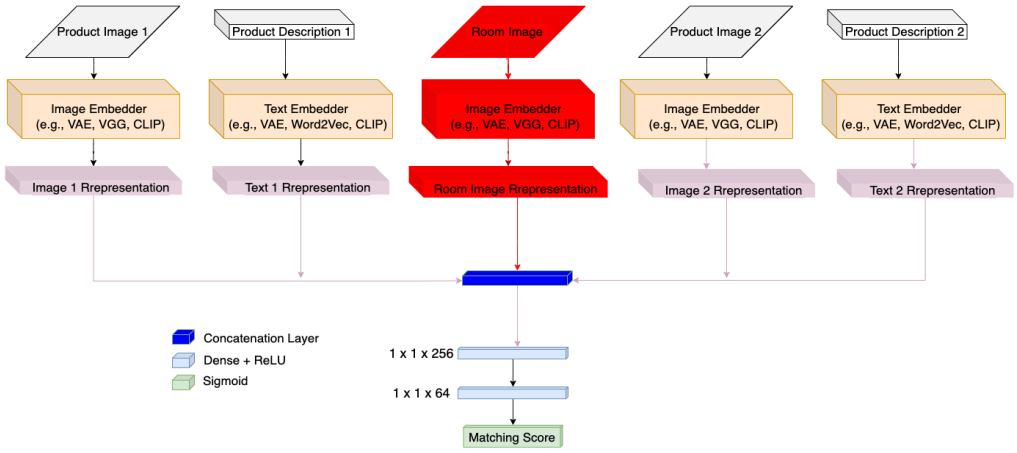

Figure 5. Co-Learning Architecture where the red path representing how the room images are incorporated

that, at each iteration, removes or adds copies of the positive examples that maximally balance the *counts* of each product in the dataset. We initialize this algorithm with two copies of every positive example, and ensure that no positive pair is deleted.

The resulting dataset contained 55,432 pairs in each training set (1,379 unique products) and 12,489 validation pairs (664 products). For the co-learning approach, we associated each positive pair with the room scene in which the two products co-occurred, and each negative pair with a random room scene containing one of the two products. Because some room scenes are missing from the IKEA dataset, we ensured that the proportion of positive and negative examples with missing room scenes was balanced.

4.2. Multimodal Baseline Models

We compared our proposed approaches to three baseline models trained in the midterm report. It is important to note that these three models were trained using an older version of the dataset, which consisted of the same train/val split of products but lacked the more sophisticated sampling procedures we applied to the new models. Therefore, comparisons against the baseline must also account for the differences in dataset generation.

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. We use the following training hyperparameters: learning rate $= 2 \times 10^{-4}$, dropout probability $= 0.3$, LSTM hidden dimension $= 128$, and number of epochs $= 7$.

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 the CNN-LSTM architecture described above. 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. 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. We use a cross-entropy loss (Equation 4) and the same best performing hyperparameters as for the CNN-LSTM.

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 straightforward transformer architecture). Because CLIP exhibits good performance on a variety of multimodal tasks without fine-tuning, we included one baseline that simply consisted of a single fully-connected layer atop CLIP's embeddings ("naive CLIP"). Additionally, we fine-tuned the last layer of each CLIP encoder to generate a stronger baseline. 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[2].

4.3. Evaluation Methodology

We sought to evaluate each model on a diverse set of tasks, capturing their accuracy on the binary classification task (accuracy and area under the receiver operating characteristic curve – AUC) as well as on ranking tasks. We define two types of ranking tasks:

1. *Single query* – Given a query furniture item q, compute the K results $r \in \mathcal{F} \setminus \{q\}$ that maximize $C(q, r)$.
2. *Double query* – Given two query items q_1 and q_2, compute the K results $r \in \mathcal{F} \setminus \{q_1, q_2\}$ that maximize the average compatibility with the two queries: $\frac{1}{2}(C(q_1, r) + C(q_2, r))$. Providing two queries serves to give the model more "clues" about what could be stylistically compatible, since many products are fairly generic in isolation.

For each ranking task, we define the following three metrics. All ranking results are provided with $K = 5$ retrieved examples, and are only computed within the validation set to prevent memorization.

NDCG This metric, 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.

F1 score Precision, Recall and F1 score are classic metrics in machine learning. Precision is the number of true positive results divided by the total number of results, including irrelevant ones; the recall is the number of true positive results divided by the total number of positive examples in the dataset. The F1 score is defined as the harmonic mean of the precision and recall. For brevity, we present the F1 score values only.

Category-limited Recall This metric is a slightly easier version of the recall metric described above, which helps to discriminate better among our models (which have very low recall in the general case). For each query item and the corresponding item we want to retrieve, we evaluate the ranking within products whose name is sufficiently similar to the target item. For example, if the query is a lamp and the target is a rug, this metric will be 1 if the target item is within the top K most compatible products whose name contains the word "rug." The metric is averaged over queries and target items which have at least 10 candidates, so that the Recall@5 task is not too easy.

5. Results and Discussion

The main quantitative results of our three methods are shown in Table 1. Additionally, we gain qualitative insight into the behavior of these models by visualizing how they embed each product in the validation set, as shown in Fig. 6. Embeddings are projected to 2D using AlignedUMAP and rendered using Emblaze.

5.1. Intermediate Fusion

The two intermediate fusion (IF) configurations perform the best out of the proposed approaches and naive CLIP on the binary classification task, as measured by validation accuracy and AUC. This makes sense with respect to the multimodal VAE (MVAE), because IF is specifically trained for binary classification. With respect to co-learning, we hypothesize that the pretrained CLIP modules quickly overfit to the training set and were unable to learn a more generalizable classification boundary. In addition, the ensemble model of IF (average of three IF models trained with PU-bagging) generally performs better than the single model on all metrics, indicating support for the value of PU learning on this dataset.

IF performs worse than fine-tuned CLIP and co-learning on the ranking tasks. However, we can see from the qualitative examples in Fig. 7 that the retrieved products are often fairly realistic. In the two depicted queries, although the model does not retrieve the correct item, it does retrieve items that contain similar colors and patterns, as well as similar implied purposes from the text descriptions. These observations support the hypothesis that CNN instance statistics can improve attention to low-level visual features. Unfortunately, however, the VGG16 and from-scratch transformer are no match for CLIP's extremely descriptive prior.

[2]Please see our midterm report for more details on how CLIP was fine-tuned.

	CNN + LSTM	Siamese	Naive CLIP	Fine-tuned CLIP	IF (single)	IF (ensemble)	MVAE	Co-Learning
Val Acc	0.52	0.50	0.549	0.661	0.580	0.583	0.484	0.551
AUC	–	–	0.559	0.742	0.604	0.611	0.456	0.591
NDCG@5	0.0010	0.0020	0.0028	0.1078	0.0070	0.0092	0.0085	0.0198
Pair NDCG@5	–	–	0.0018	0.0676	0.0041	0.0044	0.0038	0.0126
F1@5	0.0031	0.0031	0.0024	0.0191	0.0030	0.0046	0.0028	0.0074
Pair F1@5	–	–	0.0015	0.0105	0.0026	0.0032	0.0023	0.0067
Cat Recall@5	–	–	0.364	0.505	0.357	0.391	0.382	0.407
Cat Pair Recall@5	–	–	0.372	0.506	0.371	0.388	0.394	0.406

Table 1. Quantitative measurements of all models. Validation accuracy and AUC are based on the binary classification task, while all other metrics use ranking. "Pair" indicates that instead of providing a single input as a ranking query, two furniture items were provided. "Cat" = in-category, signifying that the retrieval was for stylistically-compatible items within a set of objects matching a desired category. "IF" = intermediate fusion, "MVAE" = multimodal variational autoencoder. (Missing values were for baseline models not tested as part of the final report evaluation.)

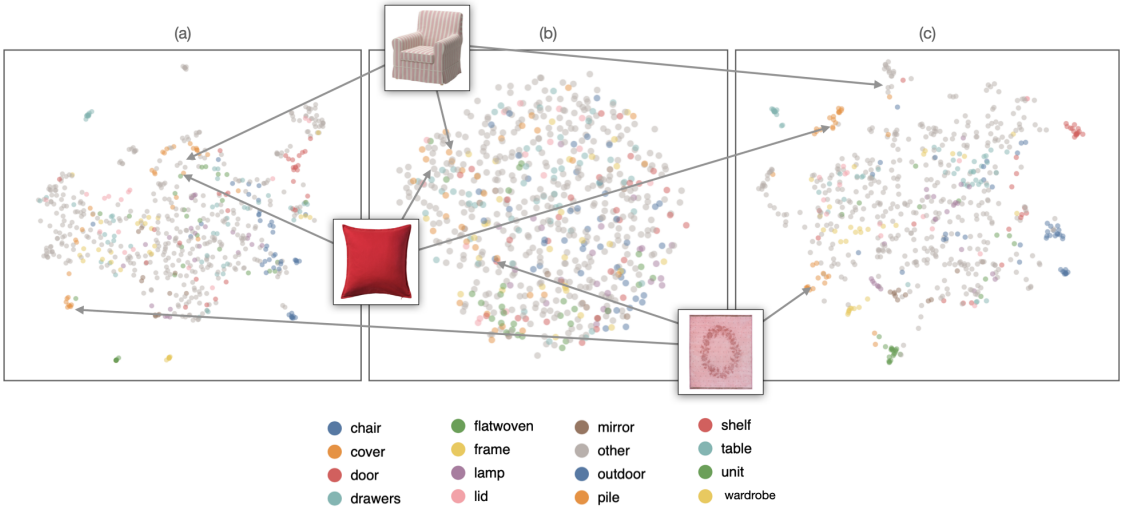

Figure 6. Comparison of embeddings generated by our three proposed approaches on the products in the validation set: (a) Intermediate Fusion, (b) Multimodal VAE, and (c) Co-Learning. The three examples have close embedding vectors in the Intermediate Fusion architecture, and are shown for comparison in the other models as well. We can see that they are closest together in the MVAE space, while in the other two spaces they are preferentially associated with objects of similar types. It should be noted that distance in these projections does not necessarily correspond to distance in the high-dimensional embedding space, particularly when items are not well-clustered.

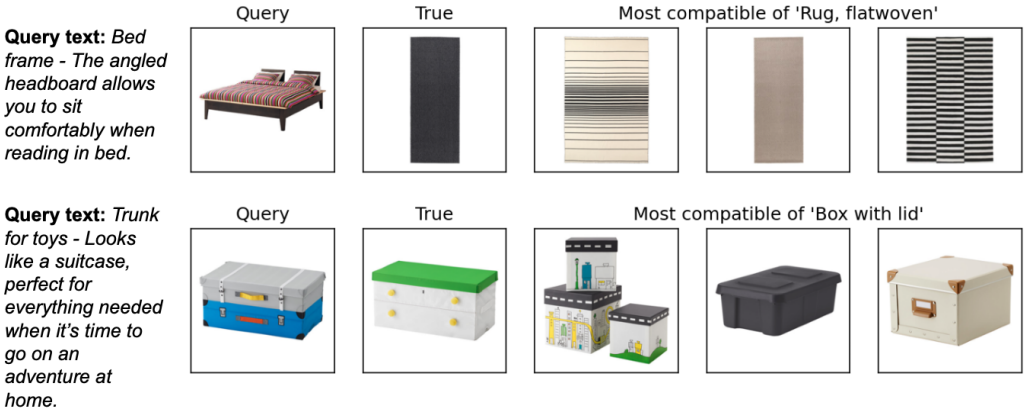

Figure 7. Example queries and results by the intermediate fusion model from the category-limited single-query ranking retrieval task. Top: for a query depicting a striped red duvet, the model retrieves rugs that are also striped or reddish. Bottom: for a query whose description indicates it is made for children, the first result is similarly childish and multicolored, although the other examples are largely irrelevant.

5.2. Multimodal VAE

Our MVAE model achieves a better ranking performance than Naive CLIP on NDCG, F1 and recall metrics. As shown in Fig. 6(b), the product embeddings according to the MVAE form a Gaussian distribution, as subjected to its loss function. In addition, note that different categories of furniture are generally not clustered closely with each other compared with the embedding spaces for IF and Co-Learning (Figs. 6(a) and (c)). This is actually a positive outcome, since our baseline models such as CLIP were often biased towards functional aspects of furniture, as opposed to stylistic compatibility.

5.3. Co-Learning

For the co-learning approach, one empirical observation is that by using our alternating training scheme, where the actual room images are used in the first few epochs and a substitute image is used in subsequent ones, the model turns out to avoid the issue of constant negative prediction for the test set. This method is fairly computationally expensive because of the additional overhead introduced by the extra room image. It is also likely to overfit after a few epochs and we also found this behavior for CLIP baseline in the midterm report. As shown in Table 1, the co-learning method produces minor improvement compared to the baseline. However, it is hasty to argue that this method is not of great potential since the current results are limited by the following factors: (1) It is time-consuming to find appropriate hyperparameters to avoid the overfitting problem given the constrained time. (2) The decision of the representation of the substitute room image as either blank, random noise, or mean training image might not be very informed; we also have not found enough prior work on how to better cope with this missing input issue. (3) More training schemes should be investigated such as balancing the proportion of substitute images in the batch level.

6. Conclusion and Future Directions

In this paper, we have deeply explored the Multimodal IKEA dataset and defined new classification and ranking tasks that have not yet been attempted on this data. We trained and evaluated three strong baseline models, then created proof-of-concepts of three augmentations to these models: intermediate fusion with convolutional instance statistics, multimodal VAEs, and co-learning with room scenes. Our results indicate that the fine-tuned CLIP baseline still receives the best scores for both training and ranking metrics among all models; however, our methods show promise compared to the CLIP prior without fine-tuning. Given the limitations on compute resources, it was infeasible to exhaustively search the optimal hyperparameters for each model in order to maximize their performances. It may therefore be possible that the three proposed models will result in comparative or stronger performance compared to the baselines.

For future work, we aim to continue improving performance from both modeling and data perspectives. From the modeling perspective, there are still a wide range of combinations to attempt including intermediate fusion with the strong CLIP model or co-learning MVAE representations with room images. From the data perspective, we may take advantage of crowdsourcing as an supplementary signal of positive and negative furniture pairs.

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.

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.

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.

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.

Huang, X. and Belongie, S. Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization. *Proceedings of the IEEE International Conference on Computer Vision*, 2017-October:1510–1519, 2017. ISSN 15505499. doi: 10.1109/ICCV.2017.167.

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.

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.

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.

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.

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.