Supplementary: SkiLL: Skipping Color and Label Landscape: Self Supervised Design Representations for Products in E-commerce

Vinay K Verma, Dween Rabius Sanny, Shreyas Sunil Kulkarni, Prateek Sircar
Abhishek Singh and Deepak Gupta
International Machine Learning, Amazon
vinayugc@gmail.com, drsanny@amazon.com, sskshreyas@gmail.com, sircarp@amazon.com
p15abhisheks@iima.ac.in and deepakgupta.cbs@gmail.com

1. Augmentation Space

The use of data augmentation is crucial in self-supervised learning models, as it can greatly impact the performance of the model. In e-commerce, where there is a wide variety of product categories, it is not feasible to apply a single set of augmentations to all tasks. For example, augmentations designed for learning design representations in the fashion category may not be effective when applied to the furniture category. Therefore, it is necessary to use different augmentations to capture the relevant information corresponding to each task.The model also needs to be agnostic to certain visual attributes, such as being color invariant. This means that if two items, such as clothes or furniture, are significantly different in color but share similar style or pattern, their embedding distance must be low.

1.0.1 Intuition of Augmentation Space

To understand the intuition behind our proposed augmentation space, consider the example of a category of women's dresses. Our goal is to learn features that are invariant to color, pose, etc. and focus on style, pattern, and texture. To achieve this, we divide the local and global features and define the augmentation accordingly. To make the model agnostic to color, we apply global augmentations such as *RandomGray*, *ColorJitter*, and *GaussianBlur*. These augmentations also include random size crop and random scale. When we augment an image using *RandomGray*, the teacher model receives a color image while the student network receives the grayscale equivalent. In order to minimize the embedding distance, the model must learn features that are invariant to color. Similarly, *ColorJitter* and *GaussianBlur* help the model learn color-invariant embeddings. In addition to these global augmentations, we also define lower-resolution augmentations to capture pattern. A close embedding distance is only possible if the model focuses on patterns.

One of the properties of e-commerce that we can exploit is the availability of multiple images for a single product, which we can use as positive pairs. These images typically show the same product from different angles or perspectives. By making them or their augmentations positive pairs, the model learns to ignore some noisy variations of the same image (e.g. different angles at which the image is shot). For example, if there are 5 images of a person wearing a dress in different poses, making these images positive pairs allows the model to focus on the dress and makes it agnostic to the pose of the person or the angle at which the image is shot. Besides these global and lower-resolution augmentations, we also devise local augmentations that capture task-specific properties such as neck style, sleeve length, and knee-length for women's dresses. To determine image crops that contain information about these attributes, we use head detection to identify the head of the person wearing the dress[1]. We then use the image crops below the head to capture information about sleeve length and knee-length, and we crop a rectangle below the detected head to capture neck style. By using these crops as positive pairs among local and with the global augmentations, we can train a model that focuses on the required attributes. If we move to another product category, such as shirts, we can discard any augmentations that do not exist, like knee-length. Similarly, if our use case is limited to detecting designs and not capturing any other attributes, we can drop the corresponding augmentations. In image 1 we show examples where product pairs were identified as "similar" by our model and "dissimilar" by the closest baseline.

2. Datasets and Evaluation Details

The collection of the dataset are discussed in the main paper. Here we are discussing the details of the evaluation metric and classes used for the evaluation.

[1]We use https://github.com/deepinsight/insightface

Figure 1. Examples of samples identified as "similar" (embedding distance ≤ 0.15) by our model. Nearest baseline (DINO), trained on the same data, identified them as dissimilar (distance ≥ 0.8). Women-dress (left), Shirt (middle) and Furniture (right).

2.1. Women-dress

The women-dress are evaluated over the two attribute *Sleeve* and *Pattern* with 9 and 18 classes respectively. The class details are giving below:

2.1.1 Sleeve

['3/4 sleeve', 'short sleeve', 'sleeveless', 'long sleeve', 'half sleeve', 'cap sleeve', 'puff sleeve', 'bell sleeve', 'flare sleeve']

2.1.2 Pattern

['solid', 'floral', 'striped', 'polka', 'checkered', 'geometric', 'graphic', 'animal', 'tribal', 'paisley', 'printed', 'animal', 'plain', 'chevron', 'screen print', 'embroidered', 'quilted', 'print']

2.2. Shirt

Shirt category also evaluated for the two attribute *Sleeve* and *Pattern* with 6 and 17 classes respectively. The class details are giving below:

2.2.1 Sleeve

['long sleeve', 'half sleeve', 'short sleeve', 'cap sleeve', '3/4 sleeve', 'sleeveless']

2.2.2 Pattern

['solid', 'striped', 'checkered', 'printed', 'geometric', 'floral', 'tribal', 'quilted', 'animal', 'graphic', 'paisley', 'plain', 'polka', 'tie and die', 'print', 'chevron', 'embroidered']

2.3. Furniture

The furniture class are evaluated for the sub-category. We have 19 subcategory that are discussed below.

2.3.1 Sub-category

['Sofas- Large', 'Chairs- Large', 'Tables- Large', 'Chairs-Large', 'Sofa Sets- Large', 'Desks- Large', 'Dining Sets-Large', 'Patio Furniture Sets- Large', 'Arm Chairs- Large', 'Bedside Cabinets - Large', 'Bean Bags- Large', 'Dining Tables- Large', 'Ottomans & Footstools- Large', 'Dining Chairs- Large', 'Closet Organization', 'Folding Chairs-Large', 'Beds- Large', 'Desks & Tables- Large', 'Recliners & Loungers- Large']

3. Evaluation Matrices

We have evaluated our model for the two scenario, the first focus over the discriminative feature learned by the model, however second evaluate the model's ability to learn the color agnostic feature. The discriminative power are evaluated using the standard matrices like, accuracy, Recall@P. However to evaluate the color invariant feature we don't have any standard metric. Therefore in this work we are proposing the evaluation metric for the same. The details of the color agnostic evaluation metric are discussed below.

3.1. Color Agnostic Metric

3.1.1 Color ID

For learning color representations, we use an open-source detectron model to detect "person" in the image, then we crop the image to maximize the presence of the "person" in the image. After that, we map the color from RGB to LUV space. Finally, we get frequency distribution of color in each image. Reason for moving from RGB to LUV space is that for 2 colors, correlation of Euclidean distance between the corresponding coordinates in LUV space and perceptual similarity that humans feel between them is higher than RGB space. We compare the histogram of the color coordinates with each color from a pre-defined list of colors. We assign the hex-code of the color in the list with which the apparel is closest. The hex-code is the color representation of the apparel.

3.1.2 Measuring the color invariant

we collected the embeddings of the test set from our proposed method and cluster them using agglomerative clustering. After we get the clusters, we calculate the histogram of colours based on their frequency. Then we calculate the entropy of the clusters using equation 10. We then take the average of the entropy across all clusters.

3.2. Recall@P

We calculate the micro average precision and recall. We are reporting the recall at a predefined value for precision. We calculate the class probabilities and using precision-recall curve we clculate the the threshold for predefined precision value. After we get the threshold we calculate the true-positives and false-positives and false-negatives. We repeat this for every class and aggregate the values. At the end, we calculate the recall by:

$$Recall = \frac{True\text{–}positives}{(True\text{–}positives + False\text{–}negatives)} \quad (1)$$

4. Experiment and Results

In order to evaluate the performance of our model, we conducted rigorous experiments on a highly diverse dataset. In this section, we will provide information on the implementation details, the baseline model, the evaluation metric, and the results of the experiments.

	Other Hard-Line (OHL) Subcategories					
	Linear			Non-liner		
	Acc	R@90	R@95	Acc	R@90	R@95
BYOL	56.45	27.46	20.39	62.14	33.44	27.37
SimCLR	72.30	49.88	38.53	75.19	54.61	43.08
DINO	83.10	70.52	58.25	85.04	75.34	64.23
SkiLL	**83.77**	**71.79**	**61.35**	**86.02**	**77.45**	**65.32**

Table 1. Hard-line Subcategories classification result, there are 19 subcategories on the hard-line dataset.

4.1. Implementation Details

5. Implementation Details

We used patch sizes of eight in our models, which are more costly to use compared to patch sizes of 16, but provided better results. To aggregate information for the entire sequence, we added a learnable token to our model [9] without using label information. Our transformer module consisted of self-attention, residual connections, and fully connected layers. We trained our model with a batch size of 32 distributed over four GPUs. We employed a similar learning rate schedule as in [4], which utilized linear scaling for the first 10 epochs and then used cosine scheduling for the decay of the learning rate. The temperature parameter also played a crucial role in our model, and we set it to a lower value in the range of $\tau \in [0.04, 0.07]$. As the epochs progressed, we increased the l_2 penalty from an initial value of 0.04 to $10\times$ its original value. We used Pytorch libraries such as torchvision and opencv to train our model, which was done on an Nvidia V100 GPU.

5.1. Baseline

In our experiments, we compared our model (SkiLL) to several strong baselines in the self-supervised learning field. One of these baselines, the Transformer-based DINO model [4], is similar to our approach in that it uses a teacher-student framework with a transformer architecture. However, our approach differs in the use of local and global task-specific augmentation. Another baseline, BYOL [14], also uses a teacher-student framework with a convolutional network, but it measures the cosine similarity between positive pairs. Finally, SimCLR [5] employs a contrastive loss and negative samples for self-supervision. We evaluated all of these approaches, against SkiLL, using linear and non-linear classifier for the learned features and the proposed color invariant metric.

5.2. Evaluation Metric

We collected the data from a popular a e-commerce service where corresponding to each product we have a set of attribute value for e.g., the women-dress has attribute sleeve length, neck style, pattern, etc. We collected the data from a popular a e-commerce service where corresponding to each product we have a set of attribute value for e.g., the women-dress has attribute sleeve length, neck style, pattern, etc. To evaluate the models, we decided on available attributes with good fill rate collected the data for these attributes for each product type (task), to train a classification model. For example, in the women-dress the *pattern* attribute has 18 labels *['solid', 'floral', 'striped', 'polka',...]*. The self-supervised learned embeddings are classified into one of the 18 classes. We measure the accuracy, Recall@90 (R@90) and Recall@95 (R@95) of the product over the selected attribute individually and report the result. The accuracy for the print-type attribute over the product $\mathcal{T}_i$ is given as:

$$A_{print-type}(\mathcal{T}_i) = \frac{1}{N_i}\sum_{j=1}^{N_i} Acc(\hat{\boldsymbol{y}}, \boldsymbol{y}) \quad (2)$$

Here $\hat{\boldsymbol{y}}$ and $\boldsymbol{y}$ are the predicted and ground truth value $Acc(\hat{\boldsymbol{y}}, \boldsymbol{y}) = 1$ if both belong to same class else zero.

Similarly, we define the evaluation metric to measure the color invariance, to our best knowledge, this is the first metric to measure the color invariant property for the model. We first use the embeddings to perform an agglomerative clustering, such that "visually similar" product as per the model get grouped into one cluster. Then we convert the image into LUV space, in this space we define each image

by a unique color ID. See the supplement for the process to assign color ID to each product. Once we have color ID, for each cluster we can find the histogram of based on the frequency of the color id in a particular cluster. Let h_i is the histogram of a particular cluster j, then its entropy of the task is defined as:

$$C_i = \frac{1}{L} \sum_{j=1}^{L} -h_j \log h_j \tag{3}$$

More the color diversity in a cluster shows the higher average entropy. In this way, we can compare the two model's color invariant learning ability, however having the same number of cluster L. A model which is not agnostic to color, would tend to have a peaky histogram and hence lower entropy.

5.3. Results

We have evaluated the learned self-supervised embedding for the two scenario, that are given as follows:

5.3.1 Class Discrimination

The results for class discrimination on the Shirt, Women-Dress, and Hard-Line datasets were evaluated using the metric discussed in Section 5.2. The pattern and sleeve attributes of the Shirt and Women-Dress datasets were evaluated. The Women-Dress dataset had 18 classes for the pattern attribute and 9 classes for the sleeve attribute, while the Shirt dataset had 17 and 6 classes, respectively, for the same attributes. The Hard-Line dataset, which included products such as sofas and furniture, had 19 subcategories that were used as targets for classification. More details on the dataset are provided in the supplmentary sheet. The results for these categories are shown in Table 3 and 2. The convolutional network-based models BYOL and SimCLR showed reasonable results for the easy categories of hard-line and shirt sleeves. However, for the pattern and women-dress sleeve categories, the models showed significantly degraded performance. These models struggled to capture the fine-grain details and discriminate between small variations in the complex categories. Table 2 demonstrates that the sleeve category for the shirt showed better results because it is simpler, while the women-dress sleeve category with more complex classes resulted in significantly degraded model performance. The DINO model shows promising results for most product types and outperformed the BYOL and SimCLR models by a significant margin. However, the proposed model, SkiLL, consistently outperforms the state-of-the-art DINO model and showed promising results even when all other models failed, such as the women-dress sleeve category. Figures 2, 3 and 4 are some examples from women's dress sleeves classification task, where other baseline fails to properly predict the sleeve type. Some more examples can be seen in the supplementary sheet. While other models showed good accuracy, they struggled with high recall scores, particularly at the Recall@90 (R@90) and Recall@95 (R@95) levels. The proposed SkiLL model, however, shows high recall scores. Finally, the model was evaluated using both linear and non-linear classifiers, and the linear classifier also showed promising results, indicating that the learned embedding can be used for downstream tasks, such as clustering.

Figure 2. DINO: long sleeve, SkiLL:

Figure 3. DINO: short sleeve, SkiLL:

Figure 4. DINO: three quarter sleeve, SkiLL: short sleeve

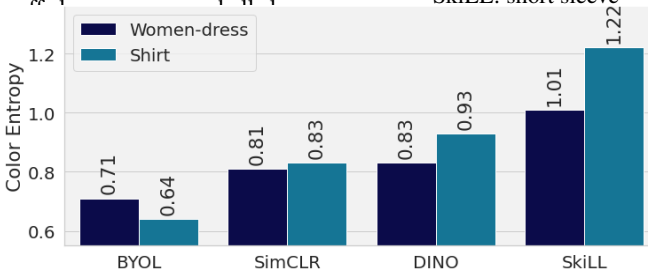

Figure 5. The result for the color invariant measurement, similar in style but agnostic to color shows high color entropy.

5.3.2 Color Invariant

We evaluated the effectiveness of our learned embeddings in terms of color invariance by using the metrics outlined in Section-5.2. The results, which depict the average color entropy across all clusters, are presented in Figure-5. It can be seen that our proposed model, SkiLL, demonstrates the highest level of color agnosticism compared to the other methods. Further, the embedding for shirts exhibits greater color agnosticism than for women's dresses. This may be attributed to the higher level of color variation present in the training data for shirts, leading to a more robust model that can effectively ignore color information.

6. Ablations

Local Augmentation and Preprocessing: The local augmentation plays a key role in the specific product type. We have conducted the ablation over the proposed task specific augmentation for the challenging dataset women-dress and pattern attribute type. We observe that if from $SkiLL$ we remove the local augmentation ($SkiLL \setminus \mathcal{L}$) the model per-

formance degraded. Also, the image pre-processing like object detection head cropping overcome the model bias towards the background or undesired feature. The $SkiLL$ without pre-processing ($SkiLL \setminus \mathcal{H}$) significantly reduce the model performance. The results for the women dress are shown in the Table-2 for the linear and non-linear evaluation.

Color Agnostic: As we discussed earlier to learn the color agnostic embedding, we incorporate the *RandomGray*, *ColorJitter* and *GaussianBlur* as a global transformation with the probability $p_1 = 0.5$, $p_2 = 0.8$ and $p_3 = 0.5$ respectively. We observe that for small value of $p_1 = p_2 = p_3 = 0.1$ the color agnostic property of the model significantly degraded and color entropy reduce from 1.01 to 0.90. Therefore, these global augmentations are necessary to achieve the color agnostic embedding.

	Linear			Non-liner		
	Acc	R@90	R@95	Acc	R@90	R@95
$SkiLL \setminus \mathcal{H}$	83.67	**68.73**	50.85	90.92	87.81	78.59
$SkiLL \setminus \mathcal{L}$	83.74	68.15	52.27	91.91	89.25	79.02
SkiLL	**83.99**	68.04	**52.30**	**92.32**	**91.34**	**84.46**

Table 2. Ablation on the women-dress for the pattern classes, without local augmentation ($SkiLL \setminus \mathcal{L}$) model performance degraded.

7. Examples where SkiLL classification is better than Baseline

Figure 6. DINO: three-fourth, SkiLL: bell sleeve

Figure 7. DINO: sleeveless, SkiLL: long sleeve

Figure 8. DINO: short sleeve, SkiLL: sleeveless

8. Qualitative Clusters with SkiLL embeddings

In figures 9 and 10 we show examples of clusters formed via agglomerative clustering of the SkiLL embeddings of the images. Note that the embeddings focus primarily on style of the apparels and is agnostic to the color of the product or pose of the human. These us analyse design trends and recommend a different color for the same design to customers.

Figure 9. Cluster example 1

Figure 10. Cluster example 2

References

[1] Paul Baltescu, Haoyu Chen, Nikil Pancha, Andrew Zhai, Jure Leskovec, and Charles Rosenberg. Itemsage: Learning product embeddings for shopping recommendations at pinterest. *arXiv preprint arXiv:2205.11728*, 2022.

[2] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541, 2006.

[3] Xuefei Cao, Bor-Chun Chen, and Ser-Nam Lim. Unsupervised deep metric learning via auxiliary rotation loss. 2019.

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 3

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020. 3

[6] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[7] Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *CoRR*, abs/2003.04297, 2020.

[8] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*, 2020.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020. 3

[10] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. *Advances in neural information processing systems*, 27, 2014.

[11] Ujjal Kr Dutta, Sandeep Repakula, Maulik Parmar, and Abhinav Ravi. A tale of color variants: Representation and self-supervised learning in fashion e-commerce. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 12482–12488, 2022.

[12] Zhiyuan Fang, Jianfeng Wang, Lijuan Wang, Lei Zhang, Yezhou Yang, and Zicheng Liu. Seed: Self-supervised distillation for visual representation. *arXiv preprint arXiv:2101.04731*, 2021.

[13] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018.

[14] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent a new approach to self-supervised learning. In *Proceedings of the 34th NeurIPS*, 2020. 3

[15] Geonmo Gu and Byungsoo Ko. Symmetrical synthesis for deep metric learning. 01 2020.

[16] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

[17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[18] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

[19] Elad Levi, Tete Xiao, Xiaolong Wang, and Trevor Darrell. Reducing class collapse in metric learning with easy positive sampling. 06 2020.

[20] Yang Li, Shichao Kan, and Zhihai He. Unsupervised deep metric learning with transformed attention consistency and contrastive clustering loss. 08 2020.

[21] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016.

[22] Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. pages 6397–6406, 06 2020.