Planckian Jitter: countering the color-crippling effects of color jitter on self-supervised training

Anonymous Author(s)
Affiliation
Address
email

Abstract

Several recent works on self-supervised learning are trained by mapping different augmentations of the same image to the same feature representation. The data augmentations used are of crucial importance to the quality of learned feature representations. In this paper, we analyze how the color jitter traditionally used in data augmentation negatively impacts the quality of the color features in learned feature representations. To address this problem, we propose a more realistic, physics-based color data augmentation – which we call *Planckian Jitter* – that creates realistic variations in chromaticity and produces a model robust to illumination changes that can be commonly observed in real life, while maintaining the ability to discriminate image content based on color information. Experiments confirm that such a representation is complementary to the representations learned with the currently-used color jitter augmentation and that a simple concatenation leads to significant performance gains on a wide range of downstream datasets. In addition, we present a color sensitivity analysis that documents the impact of different training methods on model neurons and shows that the performance of the learned features is robust with respect to illuminant variations.

1 Introduction

Self-supervised learning enables the learning of representations without the need for labeled data [8, 9]. Several recent works learn representations that are invariant with respect to a set of data augmentations and have obtained spectacular results [12, 6, 3], significantly narrowing the gap with supervised learned representations. These works vary in their architectures, learning objectives, and optimization strategies, however they are similar in applying a common set of data augmentations to generate different image views. These algorithms, while learning to map these different views to the same latent representation, learn rich semantic representations for visual data. The set of transformations (data augmentations) used induces invariances that characterizes the learned visual representation.

Before deep learning revolutionized the way visual representations are learned, features were hand-crafted to represent various properties, leading to research on shape [15], texture [16], and color features [10, 11]. Color features were typically designed to be invariant to a set of scene-accidental events such as shadows, shading, and illuminant and viewpoint changes. With the rise of deep learning, feature representations that simultaneously exploit color, shape, and texture are learned implicitly and the invariances are a byproduct of end-to-end training [14]. Current approaches to self-supervised learning learn a set of invariances implicitly related to the applied data augmentations.

In this work, we focus on the currently de facto choice for color augmentations. We argue that they seriously cripple the color quality of learned representations and we propose an alternative, physics-based color augmentation. Figure 1 (left) illustrates the currently used color augmentation on a sample image. It is clear that the applied color transformation significantly alters the colors of the

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

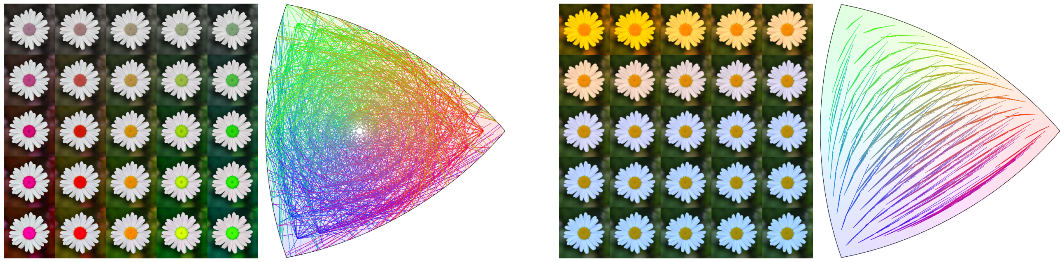


Figure 1: Default color jitter (left) and Planckian Jitter (right). Augmentations based on default color jitter lead to unrealistic images, while Planckian Jitter leads to a set of realistic ones. The ARC chromaticity diagrams for each type of jitter are computed by sampling initial RGB values and mapping them into the range of possible outputs given by each augmentation. These diagrams show that Planckian Jitter transforms colors along chromaticity lines occurring in nature when changing the illuminant, whereas default color jitter transfers colors throughout the whole chromaticity plane.

original image, both in terms of hue and saturation. This augmentation results in a representation that is invariant with respect to surface reflectance – an invariance beneficial for recognizing classes whose surface reflectance varies signficantly, for example many man-made objects such as cars and chairs. However, such invariance is expected to hurt performance on downstream tasks for which color is an important feature, like natural classes such as birds or food. One of the justifications for such strong color augmentations is that without large color changes, mapping images to the same latent representation can be purely done based on color and no complex shape features are learned. However, as a result the quality of the color representation learned with such algorithms is inferior and important information on surface reflectance might be absent.

In this paper we propose an alternative color augmentation (Figure 1, right). We draw on the existing color imaging literature on designing features invariant to illuminant changes commonly encountered in real-world scenes [10]. Our augmentation, which we called *Planckian Jitter*, applies physically realistic illuminant variation to images. We consider the illuminants described by Planck's Law for black-body radiation and that are known to be similar to illuminants encountered in real-life [21]. The aim of our color augmentation is to allow the representation to contain valuable information about the surface reflectance of objects – a feature that is expected to be important for a wide range of downstream tasks. Combining such a representation with the already high-quality shape representation learned with standard data augmentation leads to a more complete visual descriptor that describes both shape and color.

Our experiments show that self-supervised representations learned with Planckian Jitter are robust to illuminant changes. In addition, depending on the importance of color in the dataset, the proposed Planckian jitter outperforms the default color jitter. Moreover, for all evaluated datasets the combination of features of our new data augmentation with standard color jitter leads to significant performance gains of over 5% on several downstream classification tasks. Finally, we show that Planckian Jitter can be applied to several state-of-the-art self-supervised learning methods.

2 Background and related work

Self-supervised learning and contrastive learning. Recent improvements in self-supervision learn semantically rich feature representations without the need for labelled data. In SimCLR [4] similar samples are created by augmenting an input image, while dissimilar are chosen by random [4]. To make contrastive training more efficient, MoCo [13] and its improved version [5] use a memory bank for learned embeddings which makes sampling efficient. This memory is kept in sync with the rest of the network during training via a momentum encoder. Several methods do not rely on explicit contrastive pairs. BYOL uses an asymmetric network incorporating an additional MLP predictor between the outputs of the two branches [12]. One of the branches is kept "offline" and is updated by a momentum encoder. SimSiam goes even further with a simplified solution without a momentum encoder [6]. It obtains similar high-quality results and does not require a large minibatch size, in contrast to other methods.

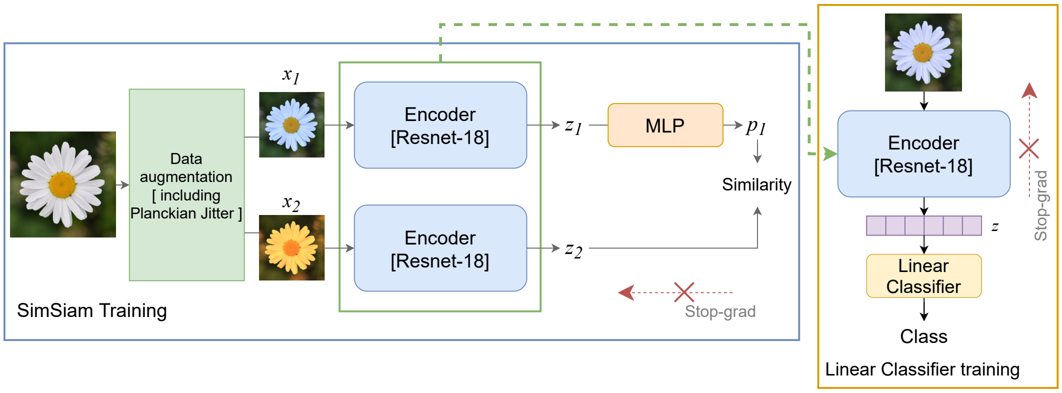


Figure 2: SimSiam training procedure exploiting Planckian-based data augmentation (left), and fine-tuning the linear classifier using the trained encoder (right).

We use the SimSiam method to verify our proposed color augmentation (we also apply it to SimCLR [4] and Barlow Twins [26] in the experiments). The main component of the network is CNN-based image encoder, learned end-to-end in an asymmetric Siamese architecture. One branch has an additional MLP predictor whose output aims to be as close as possible to other (see Figure 2). The second branch is not updated during backpropagation. A negative cosine loss function is used:

$$\mathcal{L} = \frac{1}{2}\left[\mathcal{D}(p_1, \text{stopgrad}(z_2)) + \mathcal{D}(p_2, \text{stopgrad}(z_1))\right] \tag{1}$$

$$\mathcal{D}(p_A, z_B) = -\frac{p_A}{\|p_A\|_2}\cdot\frac{z_B}{\|z_B\|_2}, \tag{2}$$

where z_1, z_2 are representations for two different augmented versions, x_1 and x_2, of the same image x. An additional predictor applied on z_1 and z_2 produces p_1 and p_2, respectively. The stopgrad$(\cdot)$ operation blocks the gradient during the backpropagation. In SimSiam no contrastive term is used and only similarity is enforced during learning.

Data augmentation. Data augmentation plays an central role in the self-supervised learning process described above. The authors of [4] and [26] discuss the importance of the different data augmentations. A set of well-defined transformations was proposed for SimCLR [4]. This set is commonly accepted and used in several later works. The augmentations include: rotation, cutout, flip, color jitter, blur and Grayscale. These operations are randomly applied to an image to generate the different views x_1, x_2 used in the self-supervision loss in Eq. 2. Applied to the same image, contrastive-like self-supervised methods learn representations invariant to such distortions.

This multiple view creation is task-related [20], however color jittering operating on hue, saturation, brightness and contrast, is one of the most important ones in terms of overall usefulness of the learned representation for downstream tasks [4, 26]. Color jitter induces a certain level of color invariance (invariance to hue, saturation, brightnesss and contrast) which are consequently transferred to the downstream task. As a consequence, we expect these learned features to underperform on downstream tasks for which color is crucial. Xiao et al. [25] were the first point out that the imposed invariances might not be beneficial for downstream tasks. As a solution, they propose to learn different embedding spaces in parallel that capture each of the invariances. Differently than them, we focus on the color distortion and propose a physics-based color augmentation that allows learning invariance to physically realistic color variations.

The color imaging literature has a long tradition in research on color features invariant to scene-accidental events such as shading, shadows, and illuminant changes [11, 10]. Invariant features were found to be extremely beneficial for object recognition. The invariance to hue and saturation changes, induced by the color jitter operation, however, it detrimental to object recognition for those classes in which color characteristics are fundamentally discriminative. Therefore, in this work we revisit early theory on illuminant invariance [10] to design an improved color augmentation that induces invariances common in the real world and that, when used during self-supervised learning, does not damage the color quality of the learned features.

3 Methodology

The image transformations introduced by default color jitter creates variability in training data that indiscriminately explores all hues at various levels of saturation. The resulting invariance is useful for downstream tasks where chromatic variations are indeed irrelevant (e.g. car color in vehicle recognition), but is detrimental to downstream tasks where color information is critical (e.g. natural classes like birds and vegetables). The main motivation for applying strong color augmentations is that this it leads to very strong shape representations. Indiscriminately augmenting color information in the image requires that the representation solve the matching problem using shape [4][1].

As an alternative to color jitter, we propose a physics-based color augmentation that mimics color variations due to illuminant changes commonly encountered in the real world. The aim is to arrive at a representation that does not have the color crippling effects of color jitter and that can therefore better describe classes for which surface reflectance is a determining feature. The aim learn a representation that, when combined with default color jitter, provides a high-quality shape and color representation.

3.1 Planckian Jitter

We call our color data augmentation procedure *Planckian Jitter* because it exploits the physical description of a black-body radiator to re-illuminate training images within a realistic illuminant distribution [10, 21]. The resulting augmentations are more realistic than those of the default color jitter (see Fig. 1). The resulting learned, self-supervised feature representation is thus expected to be robust to illumination changes commonly observed in real-world images, while simultaneously maintaining the ability to discriminate the image content based on color information.

Given an input RGB training image I, our Planckian Jitter procedure applies a chromatic adaptation transform that simulates realistic variations in the illumination conditions. The data augmentation procedure is as follows:

1. we sample a new illuminant spectrum $\sigma_T(\lambda)$ from the distribution of a black-body radiator;
2. we transform the sampled spectrum $\sigma_T(\lambda)$ into its sRGB representation $\rho_T \in \mathbb{R}^3$;
3. we create a jittered image I' by reilluminating I with the sampled illuminant ρ_T; and
4. We introduce brightness and contrast variation, producing a Planckian-jittered image I''.

A radiating black body at temperature T can be synthesized using Planck's Law [1]:

$$\sigma_T(\lambda) = \frac{2\pi hc^2}{\lambda^5(e^{\frac{hc}{kT\lambda}} - 1)} \text{ W/m}^3, \tag{3}$$

where $c = 2.99792458 \times 10^8$ m/s is the speed of light, $h = 6.626176 \times 10^{-34}$ Js is Planck's constant, and $k = 1.380662 \times 10^{-23}$ J/K is Boltzmann's constant. For our experiments we sampled T in the interval between $3000K$ and $15000K$ which is known to result in a set of illuminants that can be encountered in real life [21]. Then, we discretized wavelength λ in 10nm steps ($\Delta\lambda$) in the interval between 400nm and 700nm. The resulting spectra are visualized in Figure 4 (left) in the Supplementary Material.

The conversion from spectrum into sRGB is obtained through a series of intermediate steps [24]:

1. we first map the spectrum into the corresponding XYZ stimuli, using the 1931 CIE standard observer color matching functions $c^{\{X,Y,Z\}}(\lambda)$, in order to bring the illuminant into a standard color space that represents a person with average eyesight;
2. We normalize this tristimulus by its Y component, convert it into the CIE 1976 L*a*b color space, and fix its L component to 50 in a 0-to-100 scale, allowing us to constrain the intensity of the represented illuminant in a controlled manner as a separate task; and
3. we then convert the resulting values to sRGB, obtaining $\rho_T = \{R, G, B\}$; the resulting distribution of illuminants is visualized with the Angle-Retaining Chromaticity diagram [2] in Figure 4 (right) in the Supplementary Material.

[1] This is pointed out in the discussion of Figure 5 in [4]

All color space conversions assume a D65 reference white, which means that a neutral surface illuminated by average daylight conditions would appear achromatic. Once the new illuminant has been converted in sRGB, it is applied to the input image I by resorting to a Von-Kries-like transform [22] given by the following channel-wise scalar multiplication:

$$I'^{\{R,G,B\}} = I^{\{R,G,B\}} \cdot \{R, G, B\}/\{1, 1, 1\}, \tag{4}$$

where we assume the original scene illuminant to be white (1,1,1). Finally, brightness and contrast perturbations are introduced to simulate variations in the intensity of the scene illumination:

$$I'' = c_B \cdot c_C \cdot I' + (1 - c_C) \cdot \mu\left(c_B \cdot I'\right), \tag{5}$$

where $c_B = 0.8$ and $c_C = 0.8$ represent, respectively, brightness and contrast coefficients, and μ is a spatial average function.

3.2 Complimentarity of shape and color representations

The self-supervised learning paradigm involves a pretraining phase that relies on data augmentation to produce a set of features with certain invariance properties. These features are then used as the representation for a second phase, where we learn a given supervised downstream task. The default color jitter augmentation generates features that are strongly invariant to color information, resulting in high-quality representations of shape and texture, but that is an inferior descriptor of surface reflectances (i.e. the color of objects). Our augmentation based on Planckian Jitter (see Figure 1) is based on transformations mimicking the physical color variations in the real world due to illuminant changes. As a result, the learned representation yields a high-quality color description of scene objects. However, it likely leads to a drop in the quality of the shape representation (since color can be used to solve cases where previously shape was required). To exploit the complimentarity of the two representations, we propose to learn both – one with color jitter and one with Planckian Jitter – and to then concatenate the results in a single representation vector (of 1024 dimensions, i.e. twice the original size of 512). We call this *Latent space combination (LSC)*.

4 Experimental results

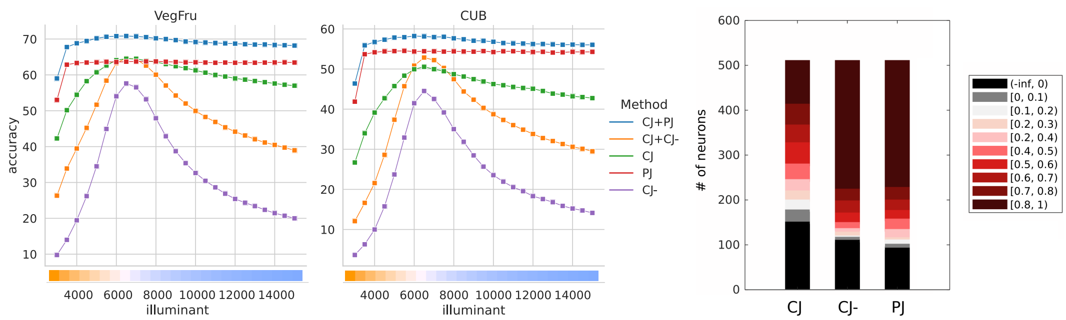
In this section, we analyze the color sensitivity of the learned backbone networks, verify the superiority of the proposed color data augmentation method compared to the default color jitter on color datasets, and evaluate the impact on downstream classification tasks. We report additional results on computational time of the proposed Planckian augmentation in the Supplementary Material.

4.1 Training and evaluation setup

We perform unsupervised training on two datasets: CIFAR-100 [14] (32×32) and ImageNet (224×224). We slightly modify the ResNet18 architecture to accommodate 32×32 images: the kernel size of the first convolutional was reduced from 7×7 to 3×3 and the first max pooling layer was removed. SimSiam training was performed using Stochastic Gradient Descent with a starting learning rate of 0.03, a cosine annealing learning rate scheduler, and mini-batch size of 512 (as in original SimSiam work [6]). For the training on the 1000-class ImageNet training set, we follow the same procedure as [6] with ResNet50.

The linear classifier training at resolution 32×32 was performed on CIFAR-100 and FLOWERS-102 [17]. CIFAR-100 is used as a baseline for the classification task. The linear classifier training for CIFAR-100 is done with Stochastic Gradient Descent for 500 epochs with a starting learning rate 0.1, a cosine annealing learning rate scheduler, and mini-batch size of 512. The FLOWERS-102 dataset with 102 classes was selected to assess the quality of the features extracted in scenarios where color information plays an important role. Images from FLOWERS-102 are resized to 32×32 pixels to match the input dimensions of the pretrained model. Here we used the Adam optimizer with initial learning rate of 0.03.

For training linear classifiers at resolution 224×224 for downstream tasks we follow the evaluation protocol of [6]. We use five different datasets: IMAGENET, FLOWERS-102, VEGFRU [19], CUB-200 [23], and T1K+ [7]. These five datasets were resized to 224×224 pixels. More details about these datasets are provided in the Supplementary Material. In the case of CUB-200, each image was



(a) (b)

Figure 3: Color sensitivity analysis. (a) Robustness to illuminant change: we report the accuracies by differently-trained backbones as a function of illuminant. (b) The color sensitivity indexes computed for the different configurations used for training the backbone.

cropped using the bounding boxes given in the dataset annotations. For T1K+, we use the 266 class labeling to train and test the linear classifier.

To assess the impact of color data augmentations we define six different configurations:

- *Default Color Jitter (CJ):* the default configuration, as used in SimSiam and SimCLR, uses both Random Color Jitter and Random Grayscale operations.
- *Default Color Jitter w/o Grayscale (CJ-)*: same as *Default* without the Random Grayscale operation.
- *Planckian Jitter (PJ)*: uses the complete proposed Planckian Jitter operation operating on chromaticity, brightness, and contrast aspects of the images. No Random Grayscale is applied.
- *LSC Default Color Jitter + Planckian Jitter ([CJ,PJ]*: This latent space combination (simple concatenation of representations) combines the default color jitter with our Planckian jitter. It allows evaluation of the complimentary nature of the representations.
- *LSC Default Color Jitter + Default Color Jitter w/o Grayscale ([CJ,CJ-])*: We combine the default color jitter with a version without the Grayscale augmentation, since this representation is also expected to result in a better color representation.
- *LSC of two Default Color Jitter Models ([CJ,CJ])*: We also show results of simply concatenating two independently trained models (trained from different seeds) with default color jitter (an ensemble of two models).

In all experiments these augmentations are combined with the other default augmentations (crop, horizontal flip, and blur).

4.2 Color sensitivity analysis

To verify if our Planckian data augmentation actually leads to illuminant invariance, we performed a robustness analysis on the CUB-200 dataset with realistic illuminant variations and analyzed sensitivity to color information. We assume as reference point the D65 illuminant, which for the purpose of this test is considered the default illuminant in every image. Given the different backbones pretrained on IMAGENET, we then train a linear classifier on this dataset (assumed to be under white illumination). For testing we create different versions of CUB-200, each illuminated by illuminants of differing color temperature. This allows us to evaluate the robustness of the learned representations with respect to these illuminant changes. A similar experiment is performed on VEGFRU.

Results are given in Figure 3(a) (more results are provided in the Supplementary Material). *Planckian Jitter* obtains a remarkably stable performance from around 4000-14000K, while *Default Color Jitter* is more sensitive to the illumination color and the classification accuracy decreases when the scene illuminant moves away from white. We also see that the combination of default and Planckian Jitter obtains the best results for all illuminants and manages to maintain a high-level of invariance with respect to the illuminant color.

Table 1: Ablation on color augmentations. Self-supervised training is performed on CIFAR-100 and the learned features are evaluated at (32×32) on CIFAR-100 and FLOWERS-102. Augmentation techniques include variations in hue and saturation (H&S), brightness and contrast (B&C), Planckian-based chromaticity (P), and random Grayscale conversions (G). Accuracy refers to the results of the linear classifiers trained with features extracted from the different backbones.

	AUGMENTATION	H&S	B&C	G	P	ACCURACY
CIFAR-100	None					41.93%
	Default Color Jitter	✓	✓	✓		59.93%
		✓	✓			41.96%
		✓				32.46%
					✓	36.10%
			✓			31.78%
	Planckian Jitter		✓		✓	47.31%
FLOWERS-102	None					36.47%
	Default Color Jitter	✓	✓	✓		30.00%
		✓	✓			36.96%
		✓				39.11%
					✓	39.51%
			✓			41.96%
	Planckian Jitter		✓		✓	42.75%

Table 2: Results for self-supervised training on CIFAR-100 and evaluated at 32×32 on CIFAR-100 and FLOWERS-102. Accuracy refers to the results of the linear classifiers trained with features extracted from the different trained backbones.

	AUGMENTATION	ACCURACY
CIFAR-100	Default Color Jitter (CJ)	59.93%
	Default Color Jitter w/o Grayscale (CJ-)	41.96%
	Planckian Jitter (PJ)	47.31%
	LSC [CJ,CJ-]	62.27%
	LSC: [CJ,PJ]	63.54%
FLOWERS-102	Default Color Jitter (CJ)	30.00%
	Default Color Jitter w/o Random Grayscale (CJ-)	36.96%
	Planckian Jitter (PJ)	42.75%
	LSC: [CJ,CJ-]	47.65%
	LSC: [CJ,PJ]	51.66%

In order to understand the impact of the color information on each neuron in trained models, we conducted an analysis using the color selectivity index described in [18]. This index measures neuron activation when color is present or absent in input images. We computed the index for the last layer of different backbones, and high values indicate color-sensitive neurons. See the Supplementary Material for more details on color selectivity. The results are shown in Figure 3(b) and indicate the number of color-sensitive neurons for each of the considered models. It is clear that the default color jitter has far fewer neurons dedicated to color description. This result confirms the hypothesis that models trained in this way are color invariant, a property that negatively affects the model in scenarios where color information has an important role as seen in our experiments. We have also analyzed the results for the default color jitter without Grayscale augmentation (CJ-). These results show that removing the Grayscale augmentation improves color sensitivity significantly. We therefore also consider this augmentation in future experiments.

4.3 Ablation study

Six different models were trained and evaluated with a linear classification for image classification. For resolution 32×32 the model is evaluated on CIFAR-100 and FLOWERS-102. The results in terms of accuracy are reported in Table 2. We identify two different trends when interpreting these results. On CIFAR-100, removing color augmentations makes the model less powerful, due to the loss of color invariance in the features extracted by the encoder. This behaviour is consistent with what was reported in [4]. We see in Table 1 that if color augmentations (i.e. brightness/contrast and Random Grayscale) are removed completely (the *None* configuration), the accuracy drops by 18%. On FLOWERS-102 the behavior is the opposite however: removing color augmentations helps the model to better classify images, obtaining an improvement of 12.75% of accuracy with respect to the default color jitter. This behavior confirms that color invariance negatively impacts downstream tasks where color information plays an important role.

Taking a closer look at the various augmentation on FLOWERS-102, we see that introducing more realistic color augmentations positively impacts contrastive training and produces models that achieve even better results with respect to the configuration without any kind of image color manipulation. Removing all color augmentations (None) improves results already by over 6%. Then, by simply reducing the jittering operation to influence brightness and contrast, leaving hue and saturation unchanged, yields another boost in accuracy of 5.49% (to 41.96). When we start modifying chromaticity using a more realistic transformation (i.e *Planckian Jitter*), the final result is a boost of 6.28% in accuracy with respect to the *None* configuration. Also, on CIFAR-100 we see an improvement of 5.38% from Planckian Jitter with respect no color augmentation. Despite this improvement, in this scenario the contrastive training with the realistic augmentation does not yield better results with respect to the *Default* configuration because color only plays a minor role on this dataset.

Given the results obtained using the data augmentations reported in Table 1, and given the considerations made in Section 3.2, we evaluate the complementarity of the learned representation by combining latent spaces from different backbones. Results for two different latent space combinations are given in Table 4. On both datasets the *Latent space combination* of Default and Planckian Jitter configurations achieves the best results. On the original CIFAR-100 task, this combination achieves a total accuracy of 63.54%, a 3.61% improvement over the *Default* configuration and 16.23% more compared to *Planckian Jitter* alone. Comparing to the LSC using the Default ColorJitter w/o Grayscale, the version with Planckian Jitter achieves a small improvement of 1.27% in classification accuracy.

On the downstream FLOWERS-102 task, the *Latent space combination* reaches an accuracy value of 51.66%: an improvement of 21.66% and 8.91% in accuracy respectively compared to the two original configurations. Compared to the LSC using Default ColorJitter w/o Grayscale, the combination with Planckian Jitter achieves a higher result, with a bigger gap in terms of accuracy with respect to the CIFAR-100 scenario. Here the use of Planckian Jitter brings in an improvement of 4.01%, confirming the impact of using realistic augmentation on classification tasks for which color is important.

4.4 Evaluation on downstream tasks

Given the results obtained from the ablation study, we performed the analysis of the proposed configurations on other downstream tasks using the backbone trained on higher resolution images (224×224 pixels). We report in Table 3 the results for: *Default Color Jitter*, *Planckian Jitter*, and several latent space combinations.

Looking at the results, we see that the *Planckian Jitter* augmentation outperforms default color jitter on two datasets (CUB-200 and T1K). Comparing the results on FLOWERS-102 with those reported above at (32×32) pixels, we see that default color jitter actually obtains good results. We hypothesize that for high-resolution images the shape information is very discriminative, and the additional color information yields little gain.

Table 3 also contains results for latent space combination. The results confirm that the two learned representation are complimentary and that their combination leads to significant performance gains of up to 9% on T1K when compared to default color jitter. As a sanity check, we have also included the latent space combination of two networks separately trained with color jitter. This provides a small ensemble performance gain on some datasets but yields significantly inferior results compared to our proposed LSC.

Table 3: Evaluation on downstream tasks. Self-supervised training was performed on IMAGENET at (224×224) and testing performed on the downstream datasets resized to (224×224).

AUGMENTATION	CUB-200	VEGFRU	T1K+	FLOWERS-102
Default Color Jitter (CJ)	54.52%	67.63%	71.44%	93.16%
Planckian Jitter (PJ)	56.28%	65.84%	77.42%	90.29%
LSC [CJ,PJ]	60.70%	74.73%	80.49%	93.99%
LSC [CJ,CJ]	56.16%	70.59%	73.47%	93.13%
LSC [CJ,CJ-]	53.14%	70.54%	78.32%	93.47%

Table 4: Effect of Plackian Jitter on different contrastive learning models. Self-supervised training was performed on CIFAR-100 and the learned features are evaluated at (32×32) on CIFAR-100 and FLOWERS-102. We report the best configurations obtained on SimSiam model and retrained SimCLR and Barlow Twins with those selected configurations.

FRAMEWORK	AUGMENTATION	CIFAR-100	FLOWERS-102
SimSiam	Default Color Jitter	59.93%	30.00%
	Planckian Jitter	47.31%	42.75%
	LSC [CJ,PJ]	63.54%	51.66%
SimCLR	Default Color Jitter	56.99%	35.29%
	Planckian Jitter	47.75%	45.00%
	LSC [CJ,PJ]	61.07%	55.78%
Barlow Twins	Default Color Jitter	56.60%	40.78%
	Planckian Jitter	52.71%	54.50%
	LSC [CJ,PJ]	62.85%	62.55%

4.5 Generality of Planckian Jitter

To show that our approach is generally applicable to self-supervised methods which exploit color augmentations, we also performed experiments using SimCLR and Barlow Twins. This comparison is given in Table 4. Independently of the model used, the *Default Color Jitter* configuration of data augmentation gives the worst results on the FLOWERS-102 dataset. The *Latent space combination* configuration consistently achieves better results on both datasets.

5 Limitations

Firstly, A drawback of Planckian jitter is that it reduces the quality of the shape representation, because the extreme color transformation of the standard color jitter force the network to solve the contrastive learning problem mainly using shape information. As shown in this article, this problem can be addressed by exploiting their complimentary nature. Secondly, our current latent space combination requires the training of two separate backbones, which certainly will also learn partially overlapping features. A training scenario, with both augmentations simultaneously in a single network while reserving part of the latent space for each augmentation, could be pursued to address this limitation.

6 Conclusion

Existing research on self-supervised learning mainly focuses on tasks where color is not a decisive feature, and subsequently exploits data augmentation procedures that negatively affect color-sensitive tasks. We propose an alternative color data augmentation technique, called Planckian Jitter, that is based on the physical properties of light. Our experiments demonstrate its positive effects on a wide variety of tasks where the intrinsic color of the objects (related to their reflectance) is crucial for discrimination, while the illumination source is not. We also proposed a solution that exploits both color and shape information by concatenating features learned with different modalities of self-supervision, leading to significant overall improvements in learned representations. Planckian Jitter can be easily incorporated into any self-supervised learning pipeline based on data augmentations, as shown by our results demonstrating improved performance for three self-supervised learning models.

References

[1] David G Andrews. *An introduction to atmospheric physics*. Cambridge University Press, 2010.

[2] Marco Buzzelli, Simone Bianco, and Raimondo Schettini. Arc: Angle-retaining chromaticity diagram for color constancy error analysis. *JOSA A*, 37(11):1721–1730, 2020.

[3] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

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

[6] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

[7] Claudio Cusano, Paolo Napoletano, and Raimondo Schettini. T1k+: A database for benchmarking color texture classification and retrieval methods. *Sensors*, 21(3):1010, 2021.

[8] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015.

[9] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. *Advances in neural information processing systems*, 27:766–774, 2014.

[10] Graham D Finlayson and Gerald Schaefer. Solving for colour constancy using a constrained dichromatic reflection model. *International Journal of Computer Vision*, 42(3):127–144, 2001.

[11] J-M Geusebroek, Rein Van den Boomgaard, Arnold W. M. Smeulders, and Hugo Geerts. Color invariance. *IEEE Transactions on Pattern analysis and machine intelligence*, 23(12):1338–1350, 2001.

[12] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21271–21284. Curran Associates, Inc., 2020.

[13] 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.

[14] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[15] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[16] Bangalore S Manjunath and Wei-Ying Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on pattern analysis and machine intelligence*, 18(8):837–842, 1996.

[17] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008.

[18] Ivet Rafegas and Maria Vanrell. Color encoding in biologically-inspired convolutional neural networks. *Vision research*, 151:7–17, 2018.

[19] Yushan Feng Saihui Hou and Zilei Wang. Vegfru: A domain-specific dataset for fine-grained visual categorization. In *IEEE International Conference on Computer Vision*, 2017.

[20] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6827–6839. Curran Associates, Inc., 2020.

[21] Shoji Tominaga, Satoru Ebisui, and Brian A Wandell. Color temperature estimation of scene illumination. In *Color and Imaging Conference*, volume 1999, pages 42–47. Society for Imaging Science and Technology, 1999.

[22] J von Kries. Theoretische studien über die umstimmung des sehorgans. *Festschrift der Albrecht-Ludwigs-Universität*, pages 145–158, 1902.

[23] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[24] Gunter Wyszecki and Walter Stanley Stiles. *Color science*, volume 8. Wiley New York, 1982.

[25] Tete Xiao, Xiaolong Wang, Alexei A Efros, and Trevor Darrell. What should not be contrastive in contrastive learning. *arXiv preprint arXiv:2008.05659*, 2020.

[26] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 12310–12320. PMLR, 2021.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section 4.1
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See section 5
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]

3. If you ran experiments...
    (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] It will be included in the supplementary material.
    (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
    (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
    (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] It will be included in the supplementary material.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
    (a) If your work uses existing assets, did you cite the creators? [Yes]
    (b) Did you mention the license of the assets? [N/A]
    (c) Did you include any new assets either in the supplemental material or as a URL? [No]
    (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
    (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
    (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
    (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
    (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]