Visual Chirality

Zhiqiu Lin[1]   Jin Sun[1,2]   Abe Davis[1,2]   Noah Snavely[1,2]

Cornell University[1]   Cornell Tech[2]

Abstract

How can we tell whether an image has been mirrored? While we understand the geometry of mirror reflections very well, less has been said about how it affects distributions of imagery at scale, despite widespread use for data augmentation in computer vision. In this paper, we investigate how the statistics of visual data are changed by reflection. We refer to these changes as "visual chirality," after the concept of geometric chirality—the notion of objects that are distinct from their mirror image. Our analysis of visual chirality reveals surprising results, including low-level chiral signals pervading imagery stemming from image processing in cameras, to the ability to discover visual chirality in images of people and faces. Our work has implications for data augmentation, self-supervised learning, and image forensics.

1. Introduction

> "...there's a room you can see through the glass—that's just the same as our drawing room, only the things go the other way."
>
> — Lewis Carroll,
> "Alice's Adventures in Wonderland & Through the Looking-Glass"

There is a rich history of lore involving reflections. From the stories of Perseus and Narcissus in ancient Greek mythology to the adventures of Lewis Carroll's Alice and J.K. Rowling's Harry Potter, fiction is full of mirrors that symbolize windows into worlds similar to, yet somehow different from, our own. This symbolism is rooted in mathematical fact: what we see in reflections is consistent with a world that differs in subtle but meaningful ways from the one around us—right hands become left, text reads backward, and the blades of a fan spin in the opposite direction. What we see is, as Alice puts it, "just the same... only the things go the other way".

Geometrically, these differences can be attributed to a world where distances from the reflecting surface are negated, creating an orientation-reversing isometry with objects as we normally see them. While the properties of such isometries are well-understood in principle, much less is known about how they affect the statistics of visual data at scale. In other words, while we understand a great deal about how reflection changes image data, we know much
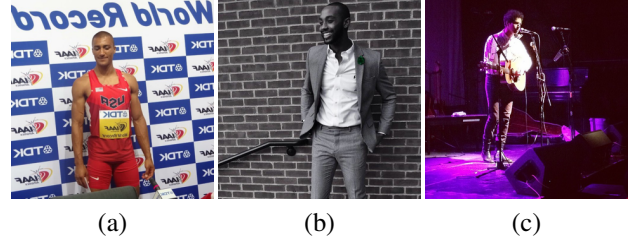

Figure 1. **Which images have been mirrored?** Our goal is to understand how distributions of natural images differ from their reflections. Each of the images here appears plausible, but some subset have actually been flipped horizontally. Figuring out which can be a challenging task even for humans. Can you tell which are flipped? Answers are in Figure 2.

less about how it changes what we learn from that data—this, despite widespread use of image reflection (e.g., mirror-flips) for data augmentation in computer vision.

This paper is guided by a very simple question: How do the visual statistics of our world change when it is reflected? One can understand some aspects of this question by considering the images in Figure 1. For individual objects, this question is closely related to the concept of *chirality* [12]. An object is said to be *chiral* if it cannot be rotated and translated into alignment with its own reflection, and *achiral* otherwise.[1] Put differently, we can think of chiral objects as being fundamentally changed by reflection—these are the things that "go the other way" when viewed through a mirror—and we can think of achiral objects as simply being moved by reflection. Chirality provides some insight into our guiding question, but remains an important step removed from telling us how reflections impact learning. For this, we need a different measure of chirality—one we call *visual chirality*—that describes the impact of reflection on distributions of imagery.

In this paper, we define the notion of visual chirality, and analyze visual chirality in real world imagery, both through new theoretical tools, and through empirical analysis. Our analysis has some unexpected conclusions, including 1) deep neural networks are surprisingly good at determining whether an image is mirrored, indicating a significant degree

---

[1]More generally, any figure is achiral if its symmetry group contains any orientation-reversing isometries.
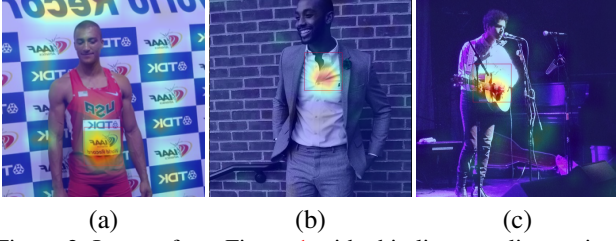
Figure 2. Images from Figure 1 with chirality-revealing regions highlighted. These regions are automatically found by our approach to chiral content discovery. (a, **flipped**) *Text chirality.* Text (in any language) is strongly chiral. (b, **not flipped**) *Object chirality.* The shirt collar, and in particular which side the buttons are on, exhibit more subtle visual chirality. (c, **flipped**) *Object interaction chirality.* While guitars are often (nearly) symmetric, the way we hold them is not (the left hand is usually on the fretboard).
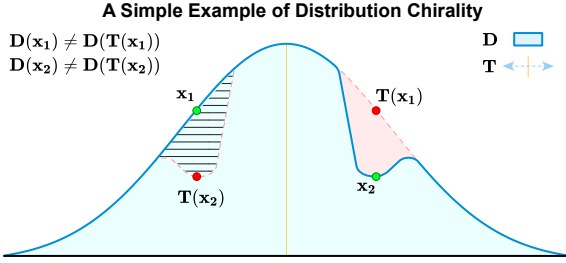

Figure 3. The curve above represents a distribution over images (shown as a 1D distribution for simplicity). Using a transformation $\mathbf{T}$ to augment a sample-based approximation of the distribution $\mathbf{D}$ assumes symmetry with respect to $\mathbf{T}$. We define visual chirality in terms of approximation error induced by this assumed symmetry when $\mathbf{T}$ is image reflection.

of visual chirality in real imagery, 2) we can automatically discover high-level cues for visual chirality in imagery, including text, watches, shirt collars, face attributes, etc, and 3) we theoretically and empirically demonstrate the existence of low-level chiral cues that are imprinted in images by common image processing operations, including Bayer demosaicing and JPEG compression. These conclusions have implications in topics ranging from data augmentation to self-supervised learning and image forensics. For instance, our analysis suggests that low-level cues can reveal whether an image has been flipped, a common operation in image manipulation.

1.1. Defining visual chirality

To define visual chirality, we first consider data augmentation for learning in computer vision to build intuition. Machine learning algorithms are based on the idea that we can approximate distributions by fitting functions to samples drawn from those distributions. Viewed in this light, data augmentation can be seen as a way to improve sampling efficiency for approximating a distribution $\mathbf{D}(\mathbf{x})$ (where $\mathbf{x}$ represents data from some domain, e.g., images) by assum-

ing that $\mathbf{D}$ is invariant to some transformation $\mathbf{T}$. More precisely, augmenting a training sample $\mathbf{x}$ with the function $\mathbf{T}$ assumes symmetry of the form:

$$\mathbf{D}(\mathbf{x}) = \mathbf{D}(\mathbf{T}(\mathbf{x})) \qquad (1)$$

which allows us to double our effective sampling efficiency for $\mathbf{D}$ at the cost of approximation error wherever the assumed symmetry does not hold. This idea is illustrated in Figure 3.

Recall that for achiral objects reflection is equivalent to a change in viewpoint. Therefore, if we consider the case where $\mathbf{D}$ is a uniform distribution over all possible views of an object, and $\mathbf{T}$ is image reflection, then Equation 5 reduces to the condition for achirality. We can then define *visual chirality* by generalizing this condition to arbitrary visual distributions. In other words, we define visual chirality as a measure of the approximation error associated with assuming visual distributions are symmetric under reflection. Defining visual chirality in this way highlights a close connection with data augmentation. Throughout the paper we will also see implications to a range of other topics in computer vision, including self-supervised learning and image forensics.

Note that our definition of visual chirality can also be generalized to include other transformations. In this paper we focus on reflection, but note where parts of our analysis could also apply more broadly.

Notes on visual chirality vs. geometric chirality. Here we make a few clarifying observations about visual chirality. First, while geometric chirality is a binary property of objects, visual chirality can be described in terms of how *much* Equation 5 is violated, letting us discuss it as a continuous or relative property of visual distributions, their samples, or, as we will see in Section 4, functions applied to visual distributions. Second, visual chirality and geometric chirality need not imply one another. For example, human hands have chiral geometry, but tend to be visually achiral because the right and left form a reflective pair and each occurs with similar frequency. Conversely, an achiral object with one plane of symmetry will be visually chiral when it is only viewed from one side of that plane. For the remainder of the paper we will refer to geometric chirality as such to avoid confusion.

2. Related work

Chirality has strong connections to symmetry, a long-studied topic in computer vision. Closely related to our work is recent work exploring the asymmetry of *time* (referred to as "Time's arrow") in videos, by understanding what makes videos look like they are being played forwards or backwards [20, 24]—a sort of temporal chirality. We explore the spatial version of this question, by seeking to understand what makes images look normal or mirrored. This spatial

chirality is related to other orientation problems in graphics in vision, such as detecting "which way is up" in an image or 3D model that might be oriented incorrectly [23, 4]. Compared to upright orientation, chirality is potentially much more subtle—many images may exhibit quite weak visual chirality cues, including a couple of the images in Figure 1. Upright orientation and other related tasks have also been used as proxy tasks for unsupervised learning of feature representations [5]. Such tasks include the arrow of time task mentioned above [24], solving jigsaw puzzles [19], and reasoning about relative positions of image patches [2].

Our problem represents an interesting variation on the classic task of detecting symmetries in images [15]. As such, our work is related to the detection and classification of asymmetric, chiral objects, as explored by Hel-Or *et al.*in their work on "how to tell left from right" [9], e.g., how to tell a left hand in an image from a right hand. However, this prior work generally analyzed *geometric* chirality, as opposed to the *visual chirality* we explore, as defined above— for instance, a right hand might be geometrically chiral but not visually chiral, while a right hand holding a pencil might visually chiral due to the prevalence of right-handed people.

Our work also relates to work on unsupervised discovery from large image collections, including work on identifying distinctive visual characteristics of cities or other image collections [3, 18] or of yearbook photos over time [6].

Finally, a specific form of chirality (sometimes referred to as *cheirality*) has been explored in geometric vision. Namely, there is an asymmetry between 3D points in front of a camera and points in back of a camera. This asymmetry can be exploited in various geometric fitting tasks [7].

3. Measuring visual chirality

In principle, one way to measure visual chirality would be to densely sample a distribution and analyze symmetry in the resulting approximation. However, this approach is inefficient and in most cases unnecessary; we need not represent an entire distribution just to capture its asymmetry. Instead, we measure visual chirality by training a network to distinguish between images and their reflections. Intuitively, success at this task should be bound by the visual chirality of the distribution we are approximating.

Given a set of images sampled from a distribution, we cast our investigation of visual chirality as a simple classification task. Let us denote a set of training images from some distribution as $C_{\text{positive}} = \{I_1, I_2, \cdots, I_n\}$ (we assume these images are photos of the real world and have not been flipped). We perform a horizontal flip on each image I_i to produce its reflected version I_i'. Let us denote the mirrored set as $C_{\text{negative}} = \{I_1', I_2', \cdots, I_n'\}$. We then assign a binary

label y_i to each image I_i in $C_{\text{positive}} \cup C_{\text{negative}}$:

$$y_i = \begin{cases} 0 & \text{if } I_i \in C_{\text{negative}}, \text{ i.e., flipped} \\ 1 & \text{if } I_i \in C_{\text{positive}}, \text{ i.e., non-flipped} \end{cases} \quad (2)$$

We train deep Convolutional Neural Nets (CNNs) with standard classification losses for this problem, because they are good at learning complex distribution of natural images [13]. Measuring a trained CNNs performance on a validation set provides insight on the visual chirality of data distribution we are investigating on.

Next we discuss details on training such a network and the techniques we use to discover the sources of visual chrality of the data distribution using a trained model as a proxy.

Network architecture. We adopt a ResNet network [8], a widely used deep architecture for image classification tasks. In particular, we use ResNet-50 and replace the last average pooling layer of the network with a global average pooling layer [16] in order to support variable input sizes.

Optimization. We train the network in a mini-batch setting using a binary cross-entropy loss. We optionally apply random cropping, and discuss the implications of such data augmentation below. We normalize pixel values by per-channel mean-subtraction and dividing by the standard deviation. We use a stochastic gradient descent optimizer [1] with momentum 0.9 and L_2 weight decay of 10^{-5}.

Hyperparameter selection. Finding a suitable learning rate is important for this task. We perform a grid search in the log domain and select the best learning rate for each experiment by cross-validation.

Shared-batch training. During training, we include both I_i and I_i' (i.e., positive and negative chirality versions of the same image) in the same mini-batch. We observe significant improvements in model performance using this approach, in alignment with prior self-supervised learning methods [5].

Discovering sources of visual chirality. If a trained model is able to predict whether an image is flipped or not with high accuracy, it must be using a reliable set of visual features from the input image for this task. We consider those cues as the source of visual chrility in the data distribution.

We use Class Activation Maps (CAM) [26] as a powerful tool to visualize those discriminative regions from a trained model. Locations with higher activation values in CAM make correspondingly larger contributions to predicting flipped images.

Throughout this paper, we visualize these activation maps as heatmaps using the Jet color scheme (red=higher activations, blue=lower activations). We only compute CAM heatmaps corresponding to an image's correct label. Figure 2 shows examples of such class activation maps.

In the following sections, we analyze visual chirality discovered in different settings using the tools described above.
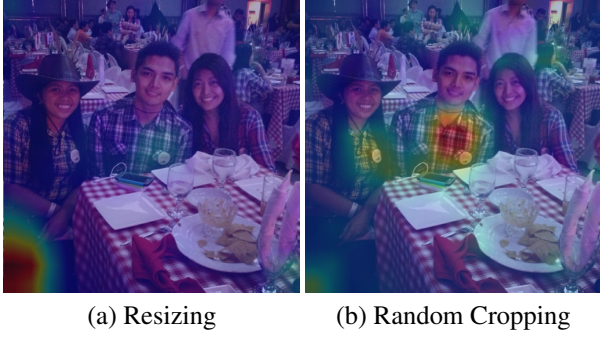
(a) Resizing   (b) Random Cropping

Figure 4. **Resizing vs. random cropping as dataset preprocessing**. This figure shows CAM heatmaps for an image from models trained with two preprocessing methods: (a) resizing and (b) random cropping. We observe that the resizing scheme learns cues in the edges or corners of images (note the focus on the lower left corner of (a)), where JPEG encoding can be asymmetric. On the other hand, the random cropping scheme captures the meaningful high-level cue—the chiral shirt collar.

4. The chirality of image processing

When we first attempted to train our model to distinguish between images and their reflections, we quickly observed that the network would find ways to accomplish this task using low-level cues that appeared only loosely correlated with the image's content. Furthermore, the strength of these cues seemed to vary a great deal with changes in how data was prepared. For example, Figure 4 shows two different CAM heatmaps for the same sample image. The left is derived from a network trained on resized data, and the right is derived from a network trained on random crops of the same data. Both maps identify a dark corner of the image as being discriminative, as well as part of the shirt on one of the the image's human subjects. However, these networks appear to disagree about the relative strength of the chiral cues in these regions. This result illustrates how the way we capture and process visual data—even down to the level of Bayer mosaics in cameras or JPEG compression—can have a significant impact on its chirality. In this section and the supplemental material we develop theoretical tools to help reason about that impact, and use this theory to predict what networks will learn in experiments.

4.1. Transformation commutativity

The key challenge of predicting how an imaging process will affect chirality is finding a way to reason about its behavior under minimal assumptions about the distribution to which it will be applied. For this, we consider what it means for an arbitrary imaging transformation $\mathbf{J}$ to preserve the symmetry of a distribution $\mathbf{D}$ (satisfying Equation 5) under a transformation $\mathbf{T}$. There are two ways we can define this. The first is simply to say that if some symmetry exists in the distribution $\mathbf{D}$ then the same symmetry should exist in $\mathbf{D_J}$, the transformation of that distribution by $\mathbf{J}$. The second is to say that if elements $\mathbf{x}_a$ and $\mathbf{x}_b$ are related by $\mathbf{x}_b = \mathbf{T}\mathbf{x}_a$, then this relationship should be preserved by $\mathbf{J}$, meaning $\mathbf{J}\mathbf{x}_b = \mathbf{T}\mathbf{J}\mathbf{x}_a$. In our supplemental material we show that both definitions hold when $\mathbf{J}$ commutes with $\mathbf{T}$, and that the second definition does not hold when $\mathbf{J}$ does not commute with $\mathbf{T}$. With these observations, commutativity becomes a powerful tool for predicting how a given process $\mathbf{J}$ can affect chirality.

4.2. Predicting chirality with commutativity

In our supplemental material we provide a detailed analysis of the commutativity of Bayer demosaicing, JPEG compression, demosaicing + JPEG compression, and all three of these again combined with random cropping. We then show that, in all six cases, our analysis of commutativity predicts the performance of a network trained from scratch to distinguish between random noise images and their reflection. These predictions also explain our observations in Figure 4. While the full details are presented in the supplemental material, some key highlights include:

- Demosaicing and JPEG compression are both individually chiral and chiral when combined.
- When random cropping is added to demosaicing or JPEG compression individually, they become achiral.
- When demosaicing, JPEG compression, and random cropping are all combined, the result is chiral.

This last conclusion is especially surprising—it implies that common image processing operations inside our cameras may leave a *chiral imprint*, i.e., that they imprint chiral cues that are imperceptible to people, but potentially detectable by neural networks, and that these features are robust to random cropping. Thus, these conclusions have implications on image forensics. For instance, our analysis gives us new theoretical and practical tools for determining if image content has been flipped, a common operation in image editing.

Finally, our analysis of how commutativity relates to the preservation of symmetries makes only very general assumptions about $\mathbf{J}$, $\mathbf{T}$, and $\mathbf{D}$, making it applicable to more arbitrary symmetries. For example, Doersch *et al.* [2] found that when they used the relative position of different regions in an image as a signal for self-supervised learning, the networks would "cheat" by utilizing chromatic aberration for prediction. Identifying the relative position of image patches requires asymmetry with respect to image translation. Applied to their case, our analysis is able to predict that chromatic aberration, which does not commute with translation, can provide this asymmetry.

5. High-level visual chirality

While analysis of chiralities that arise in image processing have useful implications in forensics, we are also interested in understanding what kinds of high-level visual content (objects, object regions, etc.) reveals visual chirality, and whether we can discover these cues automatically. As described in Section 4, if we try to train a network from scratch, it invariably starts to pick up on uninterpretable, low-level image signals. Instead, we hypothesize that if we start with a ResNet network that has been pre-trained on ImageNet object classification, then it will have a familiarity with objects that will allow it to avoid picking up on low-level cues. Note, that such ImageNet-trained networks should *not* have features sensitive to specifically to chirality—indeed, as noted above, many ImageNet classifiers are trained using random horizontal flips as a form of data augmentation.

Data. What distribution of images do we use for training? We could try to sample from the space of all natural images. However, because we speculate that many chirality cues have to do with people, and with manmade objects and scenes, we start with images that feature *people*. In particular, we utilize the StreetStyle dataset of Matzen *et al.* [17], which consists of millions of images of people gathered from Instagram. For our work, we select a random subset of 700K images from StreetStyle, and refer to this as the *Instagram* dataset; example images are shown in Figures 1 and 5. We randomly sample 5K images as a test set S_{test}, and split the remaining images into training and validation sets with a ratio of 9:1 (unless otherwise stated, we use this same train/val/test split strategy for all experiments in this paper).

Training. We trained the chirality classification approach described in Section 3 on *Instagram*, starting from an ImageNet-pretrained model. As it turns out, the transformations applied to images before feeding them to a network are crucial to consider. Initially, we downsampled all input images bilinearly to a resolution of 512×512. A network so trained achieves a 92% accuracy on the *Instagram* test set, a surprising result given that determining whether an image has been flipped can be difficult even for humans.

As discussed above, it turns out that our networks were still picking up on traces left by low-level processing, such as boundary artifacts produced by JPEG encoding, as evidenced by CAM heatmaps that often fired near the corners of images. In addition to pre-training on ImageNet, we found that networks can be made more resistant to the most obvious such artifacts by performing random cropping of input images. In particular, we randomly crop a 512×512 window from the input images during training and testing (rather than simply resizing the images). A network trained in such a way still achieves a test accuracy to 80%, still a surprisingly high result.

Non-text cues. Examining the most confident classifica-

Training set	Preprocessing	Test Accuracy	
		Instagram	F100M
Instagram	Resizing	0.92	0.57
Instagram	RandCrop	0.80	0.59
Instagram (no-text)	RandCrop	0.74	0.55

Table 1. **Chirality classification performance of models trained on *Instagram*.** Hyper-parameters were selected by cross validation. The first column indicates the training dataset, and the second column the processing that takes place on input images. The last columns report on a held-out test set, and on an unseen dataset (Flickr100M, or F100M for short). Note that the same preprocessing scheme (resize vs. random crop) is applied to both the training and test sets, and the model trained on *Instagram* without text is also tested on *Instagram* without text.

tions, we found that many involved text (e.g., on clothing or in the background), and that CAM heatmaps often predominantly focused on text regions. Indeed, text is such a strong signal for chirality that it seems to drown out other signals. This yields a useful insight: we may be able to leverage chirality to learn a text detector via self-supervision, for any language (so long as the writing is chiral, which is true for many if not all languages).

However, for the purpose of the current analysis, we wish to discover non-text chiral cues as well. To make it easier to identify such cues, we ran an automatic text detector [25] on *Instagram*, split it into text and no-text subsets, and then randomly sampled the no-text subset to form new training and text set. On the no-text subset, chirality classification accuracy drops from 80% to 74%—lower, but still well above chance.

Generalization. Perhaps our classifier learns features specific to Instagram images. To test this, Table 1 (last column) shows the evaluation accuracy of all models (without fine-tuning) on another dataset of Internet photos, a randomly selected subset of photos from Flickr100M [22]. Note that there is a significant domain gap between *Instagram* and Flickr100M, in that images in our *Instagram* dataset all contain people, whereas Flickr100M features more general content (landscapes, macro shots, etc.) in addition al people. While the performance on Flickr100M is naturally lower than on *Instagram*, our *Instagram*-trained models still perform above chance rates, with an accuracy of 55% (or 59% if text is considered), suggesting that our learned chiral features can generalize to new distributions of photos.

5.1. Revealing object-level chiral features

Inspecting the CAM heatmaps derived from our non-text-trained *Instagram* model reveals a network that focuses on a coherent set of local regions, such as smart phones and shirt pockets, across different photos. To further understand
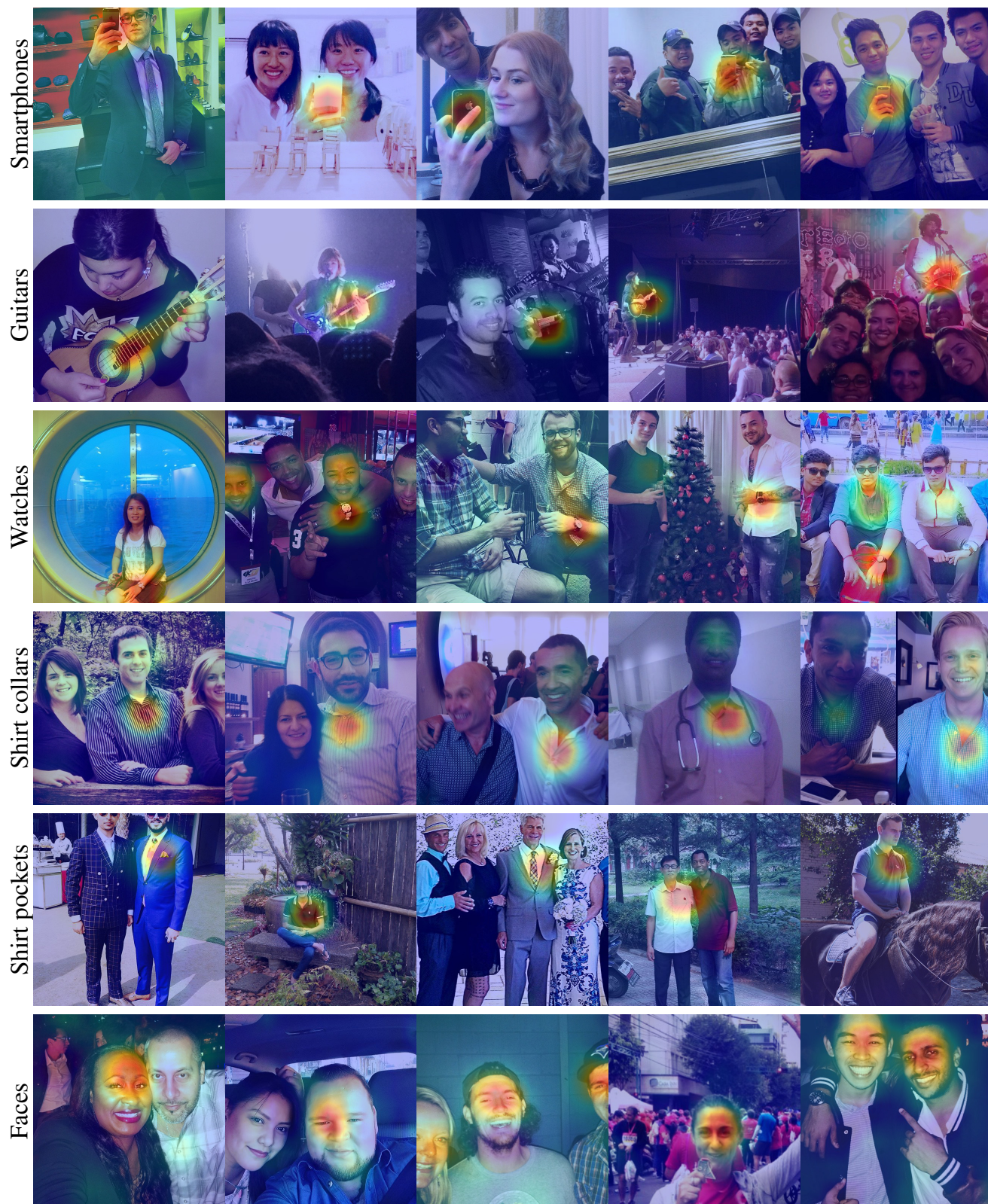
Figure 5. **Chiral clusters discovered in the Instagram dataset.** Each row shows selected images from a single discovered cluster. Each image is shown with its corresponding CAM heatmap superimposed, where red regions are highly correlated with its true chirality. We discover a range of object-level chiral clusters, such as cellphones, watches, and shirts.

what the network has learned, we develop a way to group the images, as well as their CAM heatmaps, to determine which cues are most common and salient. Inspired by work on mid-level discriminative patch mining [3, 21, 14, 18], we propose a method built upon CAM that we call *chiral feature clustering*, which automatically groups images based on the similarity of features extracted by the network, in regions deemed salient by CAM.

Chiral feature clustering. First, we extract the most discriminative local chiral feature from each image to use as input to our clustering stage. To do so, we consider the feature maps that are output from the last convolutional layer of our network. As is typical of CNNs, these features are maps with low spatial resolution, but with high channel dimensionality (e.g., 2048).

Given an input image, let us denote the output of this last convolutional layer as $\mathbf{f}$, which in our case is a feature map of dimensions $16 \times 16 \times 2048$ ($w \times h \times c$). Let $\mathbf{f}(x, y)$ denote the 2048-D vector at location (x, y) of $\mathbf{f}$. We apply CAM, using the correct chirality label for the image, to obtain a 16×16 weight activation map A. Recall that the higher the value of $A(x, y)$, the higher the contribution of the local region corresponding to (x, y) to the prediction of the correct chirality label.

We then locate the spatial maxima of A, $(x^*, y^*) = \arg\max_{(x,y)} A(x, y)$ in each image. These correspond to points deemed maximally salient for the chirality task by the network. We extract $\mathbf{f}(x^*, y^*)$ as a local feature vector describing this maximally chiral region. Running this procedure for each image yields a collection for feature vectors, on which we run k-means clustering.

Results of chiral feature clustering. We apply this clustering procedure to our no-text *Instagram* test set, using $k = 500$ clusters. We observe that this method is surprisingly effective and identifies a number of intriguing object-level chiral cues in our datasets. We refer to these clusters as *chiral clusters*. Examples of striking high-level chiral clusters are shown in Figure 5, and include phones (e.g., held in a specific way to take photos in a mirror), watches (typically worn on the left hand), shirt collars (shirts with buttoned collared typically button on a consistent side), shirt pockets, pants, and other objects.

Many of these discovered chiral clusters are highly interpretable. However, some clusters are difficult to understand. For instance, in the face cluster shown in the last row of Figure 5, the authors could not find obvious evidence of visual chirality, leading us to suspect that there may be subtle chirality cues in faces. We explore this possibility in Section 6. We also observe that some clusters focus on sharp edges in the image, leading us to suspect that some low-level image processing cues are being learned in spite of the ImageNet initialization and random cropping.

6. Visual chirality in faces

Inspired by our results on the Instagram dataset in Section 5, we now analyze chirality in face images. To do so, we use the FFHQ dataset [11] as the basis for learning. FFHQ is a recent dataset of 70K high-quality faces introduced in the context of training generative methods. We use 7% of the images as a test set and the remaining images for training and validation. We train various models on FFHQ, first downsampling images to a resolution of 520×520, then randomly cropping to 512×512. We train a standard model starting from ImageNet pre-trained features. This model achieves an accuracy of 81%, which is a promising indicator that our network can indeed learn to predict the chirality of faces with accuracy significantly better than chance.

However, perhaps there is some bias in FFHQ that leads to spurious chirality signals. For instance, since a face detector is used to create FFHQ, there is the possibility that the detector is biased, e.g., for left-facing faces vs. right-facing faces. To test this, we evaluate how well our FFHQ-trained model generalizes to other independent datasets. In particular, we evaluate this model (without fine-tuning) on another dataset, LFW, a standard face dataset [10]. We upsample the low-resolution images in LFW to 512×512 to match our input resolution. This yields an accuracy of 60%—not as high as FFHQ, perhaps due to different distributions of faces, but still significantly better than chance.

To qualitatively explore the chirality cues the model has identified, we show a sample of chiral clusters derived from the FFHQ test set in Figure 6. We can see that the CAM heatmaps in each cluster focus on specific facial regions. Based on these cluster, we have identified some intriguing preliminary hypotheses about facial chirality:

Hair part. The first cluster in Figure 6 indicates a region around the part of the hair on the left side of the forehead. We conjecture that this could be due to bias in hair part direction. We manually inspected a subset of the FFHQ test set, and found that a majority of people pictured parted their hair from left to right (the ratio is ∼2:1 for photos with visible hair part), indicating a bias for asymmetry in hair, possibly due to people preferentially using their dominant right hand to part their hair.

Predominant gaze direction, aka ocular dominance[2]. The second cluster cluster in Figure 6 highlights a region around the corner of the right eye. We conjectured that this may have to do with bias in gaze direction, possibly due to ocular dominance. We use gaze detection software[3] to determine and compare the locations of the pupil in the left and right eyes. We found that indeed more than two thirds of people in portrait photographs gaze more towards the left.

---

[2]https://en.wikipedia.org/wiki/Ocular_dominance
[3]https://github.com/shaoanlu/GazeML-keras

Figure 6. **Chiral clusters found in FFHQ.** It shows 3 chiral clusters of FFHQ dataset. The leftmost image of each row is the average face + CAM heatmap for all non-flipped images inside the each cluster. We also show some random non-flipped examples for each cluster.

Note that there are also other clusters left to be explained (for example the "beard" cluster, which may perhaps be due to males tending to use right hands to shave or groom their beard). Exploring such cues would make for interesting future work and perhaps reveal interesting asymmetries in our world.

7. Conclusion

We propose to discover visual chirality in image distributions using a self-supervised learning approach by predicting whether a photo is flipped or not, and by analyzing properties of transformations that yield chirality. We report various visual chirality cues identified using our tool on a variety of datasets such as Instagram photos and FFHQ face images. We also find that low-level chiral cues are likely pervasive in images, due to chiralities inherent in standard image processing pipelines. Our analysis has implications in data augmentation, self-supervised learning, and image forensics. Our results implies that visual chirality indeed exists in many vision datasets and such properties should be taken into account when developing real-world vision systems. However, our work suggests that it can also be used as a signal that can be leveraged in interesting new ways. For instance, since text is highly chiral, our work points to interesting future

direction in utilizing chirality in a self-supervised way to learn to detect text in images in the wild. We hope that our work will also inspire further investigation into subtle biases imprinted our visual world.

References

[1] L. Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade. Springer*, page 421–436, 2012. 3

[2] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. *ICCV*, 2015. 3, 4

[3] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros. What makes Paris look like Paris? *SIGGRAPH*, 31(4), 2012. 3, 7

[4] H. Fu, D. Cohen-Or, G. Dror, and A. Sheffer. Upright orientation of man-made objects. In *SIGGRAPH*, 2008. 3

[5] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018. 3

[6] S. Ginosar, K. Rakelly, S. Sachs, B. Yin, and A. A. Efros. A Century of Portraits: A visual historical record of american high school yearbooks. In *ICCV Workshops*, December 2015. 3

[7] R. Hartley. Cheirality invariants. In *Proc. DARPA Image Understanding Workshop*, 1993. 3

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 3

[9] Y. Hel-Or, S. Peleg, and H. Hel-Or. How to tell right from left. In *CVPR*, 1988. 3

[10] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007. 7

[11] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. *CoRR*, abs/1812.04948, 2018. 7

[12] W. T. Kelvin. The molecular tactics of a crystal. *J. Oxford Univ. Jr. Sci. Club*, 18:3–57, 1894. 1

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 3

[14] Y. Li, L. Liu, C. Shen, and A. van den Hengel. Mid-level deep pattern mining. In *CVPR*, 2015. 7

[15] Y. Liu, H. Hel-Or, C. S. Kaplan, and L. J. V. Gool. Computational symmetry in computer vision and computer graphics. *Foundations and Trends in Computer Graphics and Vision*, 5(1-2):1–195, 2010. 3

[16] Q. C. M. Lin and S. Yan. Network in network. In *International Conference on Learning Representations*, pages 2921–2929, 2014. 3

[17] K. Matzen, K. Bala, and N. Snavely. StreetStyle: Exploring world-wide clothing styles from millions of photos. *CoRR*, abs/1706.01869, 2017. 5

[18] K. Matzen and N. Snavely. BubbLeNet: Foveated imaging for visual discovery. In *ICCV*, 2015. 3, 7

[19] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016. 3

[20] L. C. Pickup, Z. Pan, D. Wei, Y.-C. Shih, C. Zhang, A. Zisserman, B. Schölkopf, and W. T. Freeman. Seeing the arrow of time. In *CVPR*, 2014. 2

[21] S. Singh, A. Gupta, and A. A. Efros. Unsupervised discovery of mid-level discriminative patches. In *ECCV*, 2012. 7

[22] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li. Yfcc100m: The new data in multimedia research. *CACM*, 59(2), Jan. 2016. 5

[23] A. Vailaya, H. Zhang, C. Yang, F.-I. Liu, and A. K. Jain. Automatic image orientation detection. *Trans. Image Processing*, 11(7):746–55, 2002. 3

[24] D. Wei, J. Y. S. Lim, A. Zisserman, and W. T. Freeman. Learning and using the arrow of time. In *CVPR*, 2018. 2, 3

[25] H. W. Y. W. S. Z. W. H. X. Zhou, C. Yao and J. Liang. East: An efficient and accurate scene text detector. In *CVPR*, 2017. 5

[26] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 3

Visual Chirality—Supplemental Material: Commutativity and the Chirality of Imaging Processes

Zhiqiu Lin[1]   Jin Sun[1,2]   Abe Davis[1,2]   Noah Snavely[1,2]

Cornell University[1]   Cornell Tech[2]

Abstract

In this document we examine the effect that an image operation can have on the symmetries of an image distribution. We show how an operation's commutativity with a transformation can be used to predict how it will affect symmetries of a distribution with respect to that transformation. We then use this observation to analyze how operations such as Bayer demosaicing, JPEG compression, and random cropping, affect visual chirality, and show that our analysis accurately predicts the performance of deep networks trained on processed data.

8. Introduction

A key goal of our work on visual chirality is to understand how reflection changes what we can learn from image data. We can think of this change as the difference between two distributions: one representing the data, and the other representing its reflection. In our main paper, we examine this difference by training a network to distinguish between samples drawn from each distribution. However, most image data undergoes extensive processing before it even leaves a camera, and it is easy to imagine how such processing could introduce asymmetry that makes it trivial to distinguish between original images and their reflections. For example, if a camera were to watermark every image with an asymmetric pattern, then nearly any distribution of images it produced would be chiral, even if the content of every image (watermark aside) were perfectly symmetric. This leads to an important question: when can we attribute visual chirality to the visual world, and when might it instead be a consequence of how we process images? To help answer this, we develop a theory relating the preservation of symmetry in an image distribution to the commutativity of image processing operations with symmetry transformations.

We begin by reviewing different types of symmetry and how they relate to data augmentation and machine learning (Section 9). Next, in Section 10, we define what it means for an operation to preserve symmetry, and derive various relationships between the commutativity of such operations with a transformation, and whether they preserve symmetries with respect to that transformation. Then, based on the theory we developed in Section 10, we introduce a simple technique that uses a small number of representative samples from a distribution to quickly estimate whether an imaging operation may introduce visual chirality into that distribution (Section 11). In Section 12 we apply this technique to common digital image processing operations, including Bayer demosaicing and JPEG compression, to analyze the effect that they have on visual chirality and learning. Finally, in Section 13, we extends our analysis to consider random cropping and show how it can sometimes be used to make chiral operations achiral.

We say that an operation is chiral if it can map an achiral (symmetric) distribution of images to a chiral (asymmetric) one, and say that it is achiral if it preserves symmetry. Some concrete results of our analysis include showing that Bayer demosaicing and JPEG compression are each achiral for certain image sizes and chiral for others, and that when combined they are chiral for all image sizes. When either is combined with random cropping individually it becomes achiral. Finally, when demosaicing and JPEG compression are both applied in combination with random cropping, the resulting operation remains chiral.

Our theoretical and empirical results altogether suggest that nearly imperceptible chiral traces may be left in photos by non-commutative imaging pipelines, which has implications on self-supervised learning, image forensics, data augmentation, etc.

9. Symmetry

We begin by reviewing what it means for a distribution to be symmetric, and for that symmetry to be preserved under different transformations. From this we will derive a relationship between the commutativity of an operation with a transformation, and the preservation of symmetries under that transformation.

9.1. Terms & Definitions:

We first define the terms of our analysis abstractly and with minimal assumptions to keep our conclusions as general as possible. We assume the following are given:

- A distribution $\mathbf{D} : \mathbb{R}^n \mapsto \mathbb{R}$ over some elements.

- A *symmetry transformation* $\mathbf{T} : \mathbb{R}^n \mapsto \mathbb{R}^n$, which we assume to be invertible and associative.

- A second (processing) transformation, $\mathbf{J} : \mathbb{R}^n \mapsto \mathbb{R}^n$, being applied to the domain of $\mathbf{D}$.

- A *transformed distribution* $\mathbf{D_J} : \mathbb{R}^n \mapsto \mathbb{R}$ obtained by applying $\mathbf{J}$ to the elements of $\mathbf{D}$:

$$\mathbf{D_J(y)} = \sum_{\mathbf{x} \in \mathbf{J}^{-1}(\mathbf{y})} \mathbf{D(x)}. \qquad (3)$$

These definitions intentionally omit assumptions that hold only in the specific case of analyzing visual chirality; for example, we do not assume that $\mathbf{T}$ is its own inverse, even though this holds for horizontal reflection. However, it is useful to remember how these abstract definitions apply to the concrete case of visual chirality, where $\mathbf{D}$ is a probability distribution over images, $\mathbf{T}$ is horizontal reflection, and $\mathbf{J}$ is some kind of image processing operation. $\mathbf{D_J}$ then describes the distribution associated with drawing images from $\mathbf{D}$ and subsequently applying $\mathbf{J}$ (i.e., the distribution of our training data if we apply $\mathbf{J}$ to every training image). So if $\mathbf{X}$ is a dataset of raw images that collectively approximate the distribution $\mathbf{D}$, and $\mathbf{J}$ is JPEG compression, then $\mathbf{D_J}$ is the distribution approximated by $\mathbf{J(X)}$, which we get by applying JPEG compression to every image in $\mathbf{X}$. The summation in Equation 3 accounts for the possibility that $\mathbf{J}$ is non-injective, in which case $|\mathbf{J}^{-1}(\mathbf{y})| \geq 1$ (i.e., $\mathbf{J}$ maps multiple distinct inputs to the same output). This is true, for example, of any lossy compression like JPEG. In such cases, the probability of a transformed element $\mathbf{y}$ is a sum over the probabilities associated with all inputs that map to $\mathbf{y}$. For convenience, a summary of each term and its meaning in the context of visual chirality is also given in Table 2.

9.2. Symmetry of Elements & Distributions

It is important to distinguish what it means for an individual element to be symmetric, and what it means for that element to be symmetric under some distribution $\mathbf{D}$. We say that an element $\mathbf{x}$ is symmetric with respect to a transformation $\mathbf{T}$ if:

$$\mathbf{x} = \mathbf{Tx} \qquad (4)$$

while symmetry with respect to $\mathbf{T}$ under some distribution $\mathbf{D}$ is defined by the condition:

$$\mathbf{D(x)} = \mathbf{D(Tx)} \qquad (5)$$

Which makes the distribution $\mathbf{D}$ itself symmetric if and only if Equation 5 holds for all $\mathbf{x}$.

Importantly, Equation 5 can hold even when Equation 4 does not, meaning that asymmetric elements may still be symmetric under the distribution $\mathbf{D}$. In the context of computer vision, this happens when an image and its reflection are different (i.e., the image itself is not symmetric) but share the same probability under $\mathbf{D}$. On the other hand, as equivalence implies equivalence under a distribution, the symmetry of an individual element does imply symmetry under $\mathbf{D}$, making the symmetry of an element a sufficient but not necessary condition for symmetry under a distribution. This makes any distribution over exclusively symmetric elements trivially symmetric; however, using $\mathbf{T}$ to augment data drawn from such a distribution would not be especially useful, as $\mathbf{T}$ would map every element to itself.

10. Symmetry Preservation

In order to reason about asymmetry in visual content we need to understand how symmetry is affected by image processing. In particular, we need to know whether a processing transformation preserves symmetries in the original data. Without this knowledge, we cannot be certain whether asymmetries that we observe in images are properties of visual content, or of how that visual content was processed. Equations 4 and 5 describe two distinct types of symmetry; the first relates two elements, while the second relates the images of these elements in $\mathbf{D}$. If we consider the effect that an operation $\mathbf{J}$ will have on each type of symmetry, we arrive at two different notions of what it means for symmetry to be preserved. The first describes whether the symmetry of individual elements is preserved. It is defined by applying Equation 4 to both $\mathbf{x}$ and $\mathbf{J(x)}$:

$$[\mathbf{x} = \mathbf{Tx}] \implies [\mathbf{J(x)} = \mathbf{TJ(x)}] \qquad (6)$$

The second type of symmetry preservation describes whether the symmetry of a distribution is preserved. It is defined by applying $\mathbf{J}$ to Equation 5:

$$[\mathbf{D(x)} = \mathbf{D(Tx)}] \implies [\mathbf{D_J(x)} = \mathbf{D_J(Tx)}] \qquad (7)$$

Note that neither of Equations 6 and 7 implies the other. For example, $\mathbf{J}$ will trivially preserve element symmetry when applied to a domain that does not contain symmetric elements, but can easily break distribution symmetry. Likewise, we can break element symmetry while preserving distribution symmetry by permuting a uniform distribution of elements such that any symmetric element maps to a non-symmetric element.

10.1. Commutativity & Element Symmetry

Our first type of symmetry preservation describes whether elements that are symmetric with respect to $\mathbf{T}$ remain so

Term	Definition	Meaning in Learning Applications
A distribution $\mathbf{D}$	$\mathbf{D} : \mathbb{R}^n \mapsto \mathbb{R}$	The underlying distribution our training data is drawn from for some task.
A symmetry transformation $\mathbf{T}$	$\mathbf{T} : \mathbb{R}^n \mapsto \mathbb{R}^n$, is associative and invertible	E.g., horizontal reflection, or any other associative and invertible transformation to be used for data augmentation.
A processing transformation $\mathbf{J}$	$\mathbf{J} : \mathbb{R}^n \mapsto \mathbb{R}^n$, does *not* have to be invertible	Some combination of image processing operations, e.g., demosaicing and/or JPEG compression.
A transformed distribution $\mathbf{D_J}$	$\mathbf{D_J}(\mathbf{y}) = \sum_{\mathbf{x} \in \mathbf{J}^{-1}(\mathbf{y})} \mathbf{D}(\mathbf{x})$	The distribution of training data after every element has been transformed by $\mathbf{J}$.

Table 2. Terms and definitions used in derivations.

after applying $\mathbf{J}$. We now show that this holds if and only if $\mathbf{J}$ commutes with $\mathbf{T}$ when applied to symmetric elements, meaning:

$$[\mathbf{x} = \mathbf{Tx}] \implies [\mathbf{J}(\mathbf{Tx}) = \mathbf{TJ}(\mathbf{x})] \tag{8}$$

Proposition 1. $\mathbf{J}$ *will preserve the symmetry of elements with respect to* $\mathbf{T}$ *(Equation 6) if and only if* $\mathbf{T}$ *and* $\mathbf{J}$ *commute on symmetric elements (Equation 8).*

Proof. We start by showing that Equation 6 implies Equation 8. As equivalence implies equality under $\mathbf{J}$, we have:

$$[\mathbf{x} = \mathbf{Tx}] \implies [\mathbf{J}(\mathbf{x}) = \mathbf{J}(\mathbf{Tx})] \tag{9}$$

And combining the right sides of Equations 6 and 9 gives us Equation 8. To show the other direction we start by applying $\mathbf{J}$ to both sides of Equation 4 to get the right side of Equation 9. From here we use Equation 8 to commute $\mathbf{T}$ and $\mathbf{J}$ and get Equation 6. This concludes the proof. $\square$

Note this also means that, if $\mathbf{J}(\mathbf{Tx}) \neq \mathbf{TJ}(\mathbf{x})$, then $\mathbf{J}$ must break the symmetry of $\mathbf{x}$.

10.2. Commutativity & Element Mapping

Notice that Proposition 1 falls short of establishing general commutativity of $\mathbf{T}$ and $\mathbf{J}$; it only applies to elements that are symmetric with respect to $\mathbf{T}$. However, we can derive a stronger relationship related to general commutativity by considering whether $\mathbf{J}$ preserves the *mapping* of elements defined by $\mathbf{T}$. To see this, note that $\mathbf{T}$ defines a map from each element $\mathbf{x}_a$ to another element $\mathbf{x}_b$, where:

$$\mathbf{x}_b = \mathbf{Tx}_a \tag{10}$$

We can think of Equation 10 as relaxing Equation 4 to include asymmetric elements, for which $\mathbf{x}_a \neq \mathbf{x}_b$. We can then define the preservation of this mapping by $\mathbf{J}$ as:

$$[\mathbf{x}_b = \mathbf{Tx}_a] \implies [\mathbf{J}(\mathbf{x}_b) = \mathbf{TJ}(\mathbf{x}_a)] \tag{11}$$

From here, we can derive a stronger claim related to general commutativity.

Proposition 2. $\mathbf{J}$ *preserves the mapping established by* $\mathbf{T}$ *(Equation 11) if and only if* $\mathbf{J}$ *commutes with* $\mathbf{T}$.

Proof. We start by showing that Equation 11 implies commutativity. As equivalence implies equality under $\mathbf{J}$, we have:

$$[\mathbf{x}_b = \mathbf{Tx}_a] \implies [\mathbf{J}(\mathbf{x}_b) = \mathbf{J}(\mathbf{Tx}_a)] \tag{12}$$

And combining the right sides of Equations 11 and 12 gives us $\mathbf{TJ}(\mathbf{x}_a) = \mathbf{J}(\mathbf{Tx}_a)$. To show that commutativity implies Equation 11, we start by applying $\mathbf{J}$ to both sides of Equation 10 to get the right side of Equation 12

$$[\mathbf{x}_b = \mathbf{Tx}_a] \implies [\mathbf{J}(\mathbf{x}_b) = \mathbf{J}(\mathbf{Tx}_a)] \tag{13}$$

From here we commute $\mathbf{T}$ and $\mathbf{J}$ on the right side to get Equation 11. This concludes the proof. $\square$

From this we can also conclude that if $\mathbf{J}$ does not commute with $\mathbf{T}$ then there must be some pair of elements $\mathbf{x}_a, \mathbf{x}_b$ such that Equation 11 does not hold.

10.3. The Symmetry of Distributions

We have shown that commutativity implies the preservation of element symmetry. Now we show that it also implies the preservation of distribution symmetry. This is a bit more complicated than the element case. Our approach, based on group theory, is to show that when $\mathbf{J}$ commutes with $\mathbf{T}$, $\mathbf{J}$ maps between disjoint cyclic groups generated by $\mathbf{T}$.

Proposition 3. *If* $\mathbf{J}$ *commutes with* $\mathbf{T}$ *and a distribution* $\mathbf{D}$ *is symmetric with respect to* $\mathbf{T}$*, then the transformed distribution* $\mathbf{D_J}$ *will also be symmetric with respect to* $\mathbf{T}$.

Proof. We first show that $\mathbf{J}$ defines a mapping between disjoint cyclic subgroups. We then show that this map is a homomorphism, which we use to relate $\mathbf{D_J}(\mathbf{x})$ to $\mathbf{D_J}(\mathbf{T}(\mathbf{x}))$.

Since $\mathbf{T}$ is associative and invertible, we can use it to partition our domain into disjoint cyclic subgroups $\langle \mathbf{x}_i \rangle^{\mathbf{T}}$ generated by $\mathbf{T}$:

$$\langle \mathbf{x}_i \rangle^{\mathbf{T}} = \{..., \mathbf{T}^{-1}\mathbf{x}_i, \mathbf{x}_i, \mathbf{Tx}_i, \mathbf{T}^2\mathbf{x}_i, \mathbf{T}^3\mathbf{x}_i, ...\} \tag{14}$$

where the identity element $\mathbf{x}_i$ of each group can be chosen as any arbitrary element within the group. We refer to the set of such group *generators* as $\mathcal{G}_\mathbf{T}$. The group operation $\cdot$ can be thought of as a permutation of each specific cyclic group relative to its identity element:

$$\mathbf{T}^a\mathbf{x}_i \cdot \mathbf{T}^b\mathbf{x}_i = \mathbf{T}^{a+b}\mathbf{x}_i \tag{15}$$

As each such subgroup shares the same group operation and is closed under that operation, any two $\langle\mathbf{x}_i\rangle^\mathbf{T}$ must either be equivalent or disjoint. The order $|\langle\mathbf{x}_i\rangle^\mathbf{T}|$ of each subgroup depends on the symmetries of $\mathbf{x}_i$ with respect to $\mathbf{T}$. For example, if $\mathbf{T}$ is simple reflection about a particular axis then $|\langle\mathbf{x}_i\rangle^\mathbf{T}| = 1$ for images $\mathbf{x}_i$ that are symmetric about that axis, and $|\langle\mathbf{x}_i\rangle^\mathbf{T}| = 2$ for images that are asymmetric about that axis.

Now consider how $\mathbf{J}$ transforms each of the subgroups $\langle\mathbf{x}_i\rangle^\mathbf{T}$:

$$\mathbf{J}\langle\mathbf{x}_i\rangle^\mathbf{T} = \{...\mathbf{J}\mathbf{T}^{-1}\mathbf{x}_i, \mathbf{J}\mathbf{x}_i, \mathbf{J}\mathbf{T}\mathbf{x}_i, \mathbf{J}\mathbf{T}^2\mathbf{x}_i, ...\} \tag{16}$$

If $\mathbf{J}$ commutes with $\mathbf{T}$, we can rewrite the above as

$$\mathbf{J}\langle\mathbf{x}_i\rangle^\mathbf{T} = \{...\mathbf{T}^{-1}\mathbf{J}\mathbf{x}_i, \mathbf{J}\mathbf{x}_i, \mathbf{T}\mathbf{J}\mathbf{x}_i, \mathbf{T}^2\mathbf{J}\mathbf{x}_i, ...\} \tag{17}$$

giving us

$$\mathbf{J}\langle\mathbf{x}_i\rangle^\mathbf{T} = \langle\mathbf{J}\mathbf{x}_i\rangle^\mathbf{T} \tag{18}$$

This shows that $\mathbf{J}$ maps cyclic subgroups generated by $\mathbf{T}$ to cyclic subgroups that can be generated by $\mathbf{T}$.

Symmetry with respect to $\mathbf{T}$ can be restated as the condition that all elements within common cyclic subgroups generated by $\mathbf{T}$ share the same probability. In other words, for each cyclic subgroup $\langle\mathbf{x}_i\rangle^\mathbf{T}$, all elements of the subgroup have the same probability under $\mathbf{D}$, i.e., all elements have probability $\mathbf{D}(\mathbf{x}_i)$. It is therefore sufficient for us to show that the map $\mathbf{J} : \langle\mathbf{x}_i\rangle^\mathbf{T} \mapsto \langle\mathbf{J}\mathbf{x}_i\rangle^\mathbf{T}$ is a homomorphism, as the first isomorphism theorem ensures the same number of equal-probability elements from $\langle\mathbf{x}_i\rangle^\mathbf{T}$ will map to each element of $\langle\mathbf{J}\mathbf{x}_i\rangle^\mathbf{T}$.

Recall that a homomorphism $h : G \mapsto H$ is defined by the relation $h(u \cdot v) = h(u) \cdot h(v)$. It is simple to show that this holds for $\mathbf{J}$ and our cyclic subgroups when $\mathbf{J}$ commutes with $\mathbf{T}$:

$$\begin{aligned}
\mathbf{J}(\mathbf{T}^a\mathbf{x}_i \cdot \mathbf{T}^b\mathbf{x}_i) &= \mathbf{J}(\mathbf{T}^{a+b}\mathbf{x}_i) \\
&= \mathbf{T}^{a+b}\mathbf{J}(\mathbf{x}_i) \\
&= \mathbf{T}^a\mathbf{J}(\mathbf{x}_i) \cdot \mathbf{T}^b\mathbf{J}(\mathbf{x}_i) \\
&= \mathbf{J}(\mathbf{T}^a\mathbf{x}_i) \cdot \mathbf{J}(\mathbf{T}^b\mathbf{x}_i)
\end{aligned} \tag{19}$$

This is sufficient to prove our proposition. For completeness, we also reformulate $\mathbf{D}_\mathbf{J}$ in terms of the cyclic subgroups $\langle\mathbf{J}\mathbf{x}_i\rangle^\mathbf{T}$. We will use the notation $\mathbf{1}_{\langle\mathbf{x}_i\rangle^\mathbf{T}}$ to denote an indicator distribution that maps every element of $\langle\mathbf{x}_i\rangle^\mathbf{T}$ to 1, and every other element to 0. Note that any distribution we

can represent as the weighted sum of $\mathbf{1}_{\langle\mathbf{x}_i\rangle^\mathbf{T}}$ must preserve symmetry with respect to $\mathbf{T}$. We can express $\mathbf{D}$ as:

$$\mathbf{D} = \sum_{\mathbf{x}_i \in \mathcal{G}_\mathbf{T}} \mathbf{D}(\mathbf{x}_i) \cdot \mathbf{1}_{\langle\mathbf{x}_i\rangle^\mathbf{T}} \tag{20}$$

Now, using the first isomorphism theorem to account for the case where $\mathbf{J}$ is non-injective, we can combine Equation 3 and 20 to write $\mathbf{D}_\mathbf{J}$ as

$$\mathbf{D}_\mathbf{J} = \sum_i (\mathbf{D}(\mathbf{x}_i)|\ker \mathbf{J}\mathbf{x}_i|) \cdot \mathbf{1}_{\langle\mathbf{J}\mathbf{x}_i\rangle^\mathbf{T}} \tag{21}$$

where $\ker \mathbf{J}\mathbf{x}_i$ is the kernel of $\mathbf{J} : \langle\mathbf{x}_i\rangle^\mathbf{T} \mapsto \langle\mathbf{J}\mathbf{x}_i\rangle^\mathbf{T}$.[1] This concludes our proof. $\square$

10.4. Permuted Commutativity

From Proposition 1 we can conclude that if $\mathbf{J}$ does *not* commute with $\mathbf{T}$ when applied to some symmetric element $\mathbf{x}$ then $\mathbf{J}(\mathbf{x})$ will not be symmetric with respect to $\mathbf{T}$ (this is also simple to prove independently). However, our proof of Proposition 3 is not bi-directional; we only show that commutativity implies distribution symmetry will be preserved. What, then, can we conclude about operations that do not commute with $\mathbf{T}$?

The first thing to note is that non-commutativity does *not* imply that distribution symmetry will be broken. There are various ways for symmetry to be preserved even when $\mathbf{J}$ and $\mathbf{T}$ do not commute, but here we consider a case where groups of operations that do not commute with $\mathbf{T}$ individually combine to preserve distribution symmetry. For instance, imagine that given a training set of images drawn from a distribution $\mathbf{D}$, and that we generate a new training set by applying multiple random crops to each original image—we can think of each different crop offset j as a different transformation $\mathbf{J}_i$ applied to the original distribution, and the accumulation of all random crops to reflect a new, accumulated distribution. As such, this accumulation of transformed images is particularly relevant to computer vision, as it will help us explain an effect that random cropping can have on bias introduced by data augmentation.

We proved Proposition 3 by showing that $\mathbf{J}$ formed a homomorphism between cyclic subgroups $\langle\mathbf{x}_i\rangle^\mathbf{T}$ and $\langle\mathbf{J}\mathbf{x}_i\rangle^\mathbf{T}$. We now consider the case where $\mathbf{D}_\mathbf{J}$ is the sum of multiple such homomorphisms $\mathbf{J}_j$, as would result from accumulating the results of multiple transformations $\mathbf{J}_j$:

$$\mathbf{D}_\mathbf{J}(\mathbf{x}) = \sum_j \sum_{\mathbf{x}_i : \mathbf{J}_j\mathbf{x}_i = \mathbf{x}} \mathbf{D}(\mathbf{x}_i) \tag{22}$$

In this case, the sum of symmetric distributions is a symmetric distribution, which tells us that $\mathbf{D}_\mathbf{J}$ will still be symmetric.

---

[1]Recall that the *kernel* of a homomorphism $\mathbf{J}$, $\ker \mathbf{J}$, is the subset of elements that $\mathbf{J}$ maps to the identity element.

Now note that by permuting the elements on the right side of Equation 22, we can define a new set of transformations that sum to the same $\mathbf{D_J}$, ensuring that symmetry remains preserved.

$$
\begin{array}{ccccccc}
\langle \mathbf{x}_i \rangle_{\mathbf{T}} & = & \mathbf{x}_i & \mathbf{T}\mathbf{x}_i & \mathbf{T}^2\mathbf{x}_i & \mathbf{T}^2\mathbf{x}_i & \ldots \\[10pt]
\mathbf{J}_1\langle \mathbf{x}_i \rangle_{\mathbf{T}} & = & \mathbf{J}_1\mathbf{x}_i & \mathbf{J}_1\mathbf{T}\mathbf{x}_i & \mathbf{J}_1\mathbf{T}^2\mathbf{x}_i & \mathbf{J}_1\mathbf{T}^2\mathbf{x}_i & \ldots \\[10pt]
\mathbf{J}_2\langle \mathbf{x}_i \rangle_{\mathbf{T}} & = & \mathbf{J}_2\mathbf{x}_i & \mathbf{J}_2\mathbf{T}\mathbf{x}_i & \mathbf{J}_2\mathbf{T}^2\mathbf{x}_i & \mathbf{J}_2\mathbf{T}^2\mathbf{x}_i & \ldots \\[10pt]
\mathbf{J}_3\langle \mathbf{x}_i \rangle_{\mathbf{T}} & = & \mathbf{J}_3\mathbf{x}_i & \mathbf{J}_3\mathbf{T}\mathbf{x}_i & \mathbf{J}_3\mathbf{T}^2\mathbf{x}_i & \mathbf{J}_3\mathbf{T}^2\mathbf{x}_i & \ldots
\end{array}
$$

Figure 7. **Permuted Commutativity.** We can permute the elements on the right side of Equation 22 to define a set of processing transformations $\mathbf{J}_j$ that may not commute with $\mathbf{T}$, but in aggregate still sum to a symmetric distribution. These $\mathbf{J}_j$ are characterized by a permuted commutativity relationship. In this example, that relationship is given by $\mathbf{T}\mathbf{J}_a = \mathbf{J}_b\mathbf{T}$, where $a = k \pmod 3$ and $b = k + 1 \pmod 3$.

This means that if we can find a permutation of $\mathbf{J}_j$ such that the aggregated transformed distribution still maintain symmetric when $\mathbf{T}\mathbf{J}_a = \mathbf{J}_b\mathbf{T}$ for a and b are indices of a found permutation (Figure 7). This notion of permuted commutativity will lead to a definition of *glide commutativity* in section 13 concerning groups related by translation in 2D image plane.

11. Commutative Residuals

Propositions 2 and 3 establish a connection between the commutativity of two operations (a processing operation and a symmetry transformation), and the preservation of two kinds of symmetries. Notably, commutativity guarantees that the symmetry of a distribution under a transformation is preserved. How do we apply this finding in practice on a specific distribution of images and a specific processing operation?

It can be difficult to model complex processing operations like JPEG compression and Bayer demosaicing analytically, and it may be the case that such operations commute with a transformation when applied to certain inputs, and not when applied to others. Furthermore, the impact of non-commutativity is not binary: if we think of the asymmetries introduced by an operation as some signal indicating, for example, whether an image has been flipped, then it is useful to consider the magnitude of that signal relative to variations in the distribution that contains it. These concerns lead us to derive a numerical measure of commutativity that we can evaluate on representative samples of a distribution to gauge the strength of assymmetries introduced by an operation. We define $\mathbf{E_J}(\mathbf{x})$, the *commutative residual image* of operation $\mathbf{J}$

with respect to transformation $\mathbf{T}$ on the image $\mathbf{x}$, as follows:

$$
\mathbf{E_J}(\mathbf{x}) = \mathbf{J}(\mathbf{T}(\mathbf{x})) - \mathbf{T}(\mathbf{J}(\mathbf{x})) \tag{23}
$$

We can get a rough measure of the commutativity between an imaging processing step and a transformation on some representative samples $\mathbf{x}$ by looking at the value of $|\mathbf{E_J}(\mathbf{x})|$, which we summarize by its average across all pixels, $\hat{\mathbf{e}}_\mathbf{J}(\mathbf{x})$. We refer to $\hat{\mathbf{e}}_\mathbf{J}(\mathbf{x})$ as a *commutative residual*. A commutative residual of 0 on a particular image $\mathbf{x}$ means that $\mathbf{T}$ and $\mathbf{J}$ commute for that image, and a non-zero commutative residuals means that they do not commute for that image. As the derivations in Section 10 show, if the commutative residual is 0 for all elements of a distribution (i.e., the processing operation commutes with $\mathbf{T}$), the symmetry will be preserved. If not, symmetries may be broken.

Commutative Residuals for Mirror Reflections. An alternative intuition of commutative residuals can be arrived at in the case where $\mathbf{T}$ is its own inverse, as is true of mirror reflections. Consider the effect of $\mathbf{J}$ on a distribution represented by a dataset with two elements, $\mathbf{D} = \{\mathbf{x}, \mathbf{T}(\mathbf{x})\}$. This simple distribution is trivially symmetric, since $\mathbf{D}$ is closed under $\mathbf{T}$.

But what happens when we apply $\mathbf{J}$? $\mathbf{D}$ becomes $\mathbf{D_J} = \{\mathbf{J}(\mathbf{x}), \mathbf{J}(\mathbf{T}(\mathbf{x}))\}$, and we can measure the asymmetry of this new distribution by taking the difference between one element and the reflection of the other:

$$
\mathbf{J}(\mathbf{T}(\mathbf{x})) - \mathbf{T}(\mathbf{J}(\mathbf{x})) \tag{24}
$$

which is precisely how we define the commutative residual image above. Figure 8 shows an example computation of a commutative residual image when $\mathbf{T}$ is image flipping and $\mathbf{J}$ is the composition of Bayer demosaicing and JPEG compression.

11.1. Evaluating the Chirality of Operations

We propose two methods to evaluate the chirality introduced to an originally achiral distribution $\mathbf{D}$ by an operation $\mathbf{J}$. The first approach, based on the theory we have derived about commutativity, is to evaluate the commutative residual with respect to $\mathbf{J}$ on a small representative set of sample images. The second method, as described in the main paper in the context of analyzing real image datasets, is to train a neural network to empirically distinguish between flipped and unflipped images sampled from a much larger, symmetric dataset after transforming every image in that dataset by $\mathbf{J}$. Since we are interested in demonstrating the possibility of introducing chirality through low-level imaging operators, we study image distributions that are originally symmetric to ensure that any learned chirality cues can be attributed solely to the effect of $\mathbf{J}$.
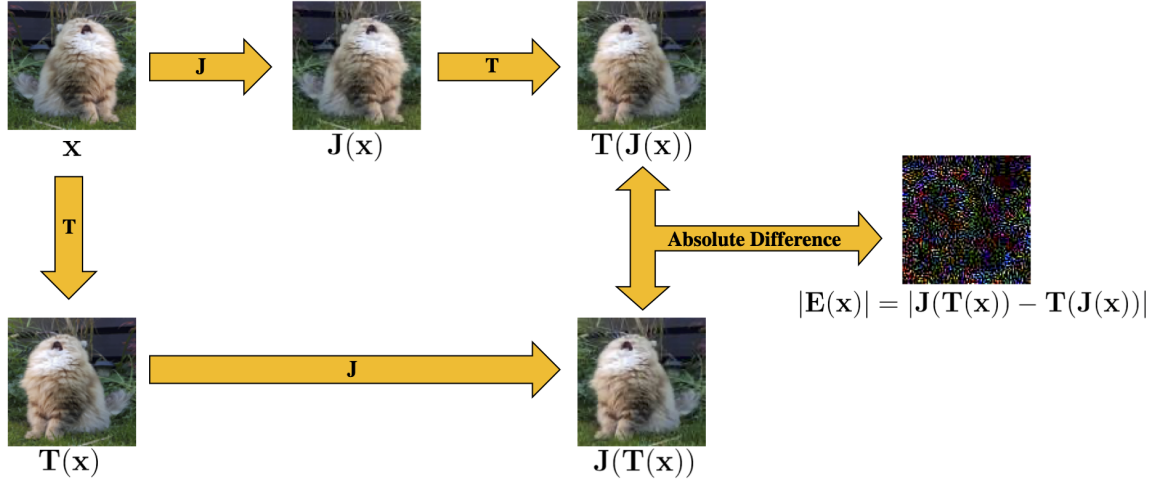
Figure 8. **Example commutative residual image:** This figure illustrates the application of the commutative residual method to a natural image. Here $\mathbf{T}$ is the horizontal reflection operation, and $\mathbf{J}$ is the composition of Bayer demosaicing and JPEG compression. The image used above has a width of 100px. For better visualization of the imperceptible differences shown in the residual image, we scale the resulting residual by a factor of 10. Consistent with the results in Figure 10, the residual image is not zero (which would be all black), i.e., the commutative residual is non-zero.
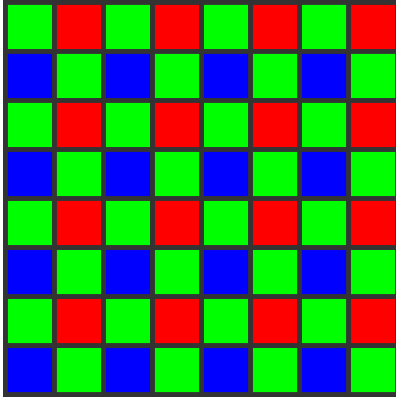

Figure 9. **Example 8×8 Bayer pattern mosaic**: A typical Bayer filter mosaic consists of tiled 2×2 blocks of pixels with two green filters and one red and one blue filter. Note that a even-sized Bayer filter, like the one pictured, is asymmetric (mirror flipped version is not equal to itself), while an odd-sized version of this filter pattern would be symmetric.

12. Analysis of Demosaicing & JPEG Compression

With the theoretical tools derived in previous sections, we evaluate two standard imaging processes: Bayer demosaicing and JPEG compression. We analyze when these two operations (and their composition) will preserve existing symmetries in a distribution of images, and when they may break them. In real camera systems, Bayer demosaicing and JPEG compression are typically two operations in a much larger image signal processing pipeline. We analyze these two operations specifically because (a) they are ubiquitous and implemented in most cameras, and (b) they have interesting symmetry properties, as we will show below. We begin with a brief summary of these two operations.

Bayer filters and demosaicing. Many modern digital cameras (including cellphone cameras) capture color by means of a square grid of colored filters that lies atop of the grid of photosensors in the camera. An 8×8 example of such a color filter grid, known as a Bayer filter mosaic, is shown in Figure 9. In such cameras, each pixel's sensor measures intensity for a single color channel (red, green, or blue), and so to produce a full color image at full resolution, we must interpolate each color channel such that each pixel ultimately has an R, G, and B value. This interpolation process is known as *demosaicing*. For our analysis we assume, as is typical, that a Bayer filter mosaic pattern consists of a tiled 2×2 element (GRBG in the case of Figure 9) and we consider the demosaicing method of Malvar [2].

The 8×8 Bayer filter mosaic in Figure 9 has interesting symmetry properties. The 8×8 pattern as a whole is asymmetric—flipping it horizontally will result in a red pixel in the upper-left corner, rather than a green pixel. The same is true for any even-sized Bayer filter mosaic. However, from the perspective of the center of any pixel, the pattern is locally symmetric. Moreover, if we imagine a 9×9 version of this mosaic (or indeed any odd-sized pattern), that mosaic would be symmetric.

JPEG compression. JPEG is one of the most common (lossy) image compression schemes. There are two main ways that JPEG compresses image data. First, it converts
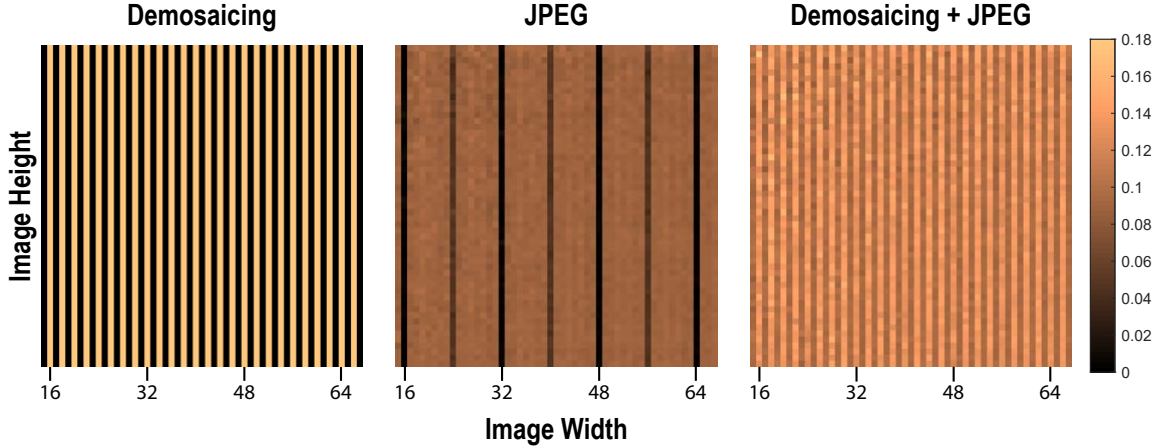
Figure 10. Commutativity residuals for demosaicing (left), JPEG compression (middle) and their composition (right): Each image shows how commutativity residual, measured in absolute average percent error per pixel, varies with different image sizes. For integers n we see commutativity in demosaicing at image widths of $2n - 1$ (i.e., odd widths), and in JPEG compression at widths of $16n$. We do not see commutativity when both are applied.

images into the $Y'C_bC_r$ colorspace and downsamples the chroma channels (C_b and C_r), typically by a factor of two. Then it splits each channel into a grid of 8×8 pixel blocks and computes the discrete cosine transform (DCT) of each block. In the luminance (Y') channel, each block covers an 8×8 pixel region of the original image, while for the chroma channels, each block corresponds to a 16×16 pixel region in the original image, due to the $2\times$ downsampling. Finally, the DCT of each block is strategically quantized to further compress the data at low perceptual cost.

For the purposes of our analysis, one noteworthy aspect of JPEG compression is that for images with dimensions that are not a multiple of 16, there will be boundary blocks that do not have a full 8×8 complement of pixels. These are handled specially by the JPEG algorithm, which can lead to breaking of symmetry for such images because the special boundary blocks are always at the right (and bottom) edges of the image, never at the left (and top) edges.

12.1. Commutative Residuals and Image Size

As an initial experiment, we generate a completely random image with random dimensions (i.e., choosing the width and height uniformly at random from some uniform distribution, and then selecting each value for each color channel at random from the range $[0, 255]$). Then we compute commutative residuals under the operations of (1) Bayer demosaicing (i.e., first synthetically generating a Bayer mosaic, then demosaicing it), (2) JPEG compression, and (3) the composition of these operations.

If we actually perform this experiment for randomly sized images, then under demosaicing, commutative residuals are nonzero about half of the time, and under JPEG compression,

they are nonzero over 90% of the time. But if we sample over different image sizes more systematically, a pattern begins to emerge.

Figure 10 visualizes commutative residuals for random noise images as a function of image width and height for the three operations described above. We can see that demosaicing appears to commute with image flipping (and therefore preserve symmetries) for images with odd widths, while JPEG compression appears to preserve symmetries for image with widths that are divisible by 16. Finally and most notably, commutativity never seems to hold for the composition of demosaicing and JPEG compression for any width. We can explain this result by considering the geometry of Bayer patterns and JPEG block grids. Bayer patterns (Figure 9) have horizontal symmetry when reflected about any line centered on a pixel column, while the JPEG block grid, which consists of 8×8 blocks that correspond to 8×8 or 16×16 blocks of the original image, is horizontally symmetric only around grid lines, which rest between columns at 16-pixel intervals. A corollary is that the combination of demosaicing followed by JPEG compression can never be commutative with respect to flipping because these two imaging processes never have zero commutative residual for the same image width (since multiples of 16 are never odd).

The black-box analysis [2] of commutative residuals shown in Figure 10 reveals the grid structures underlying these processing algorithms, and illustrates how each grid structure impacts preservation of symmetries. When the commutative residual for any transformation for a given image width is

---

[2]All analysis and experiments in this and the next section are available in Python at https://github.com/linzhiqiu/digital_chirality.

zero, we know that this transformation preserves the symmetry of the original distribution with such width. ***Hence, a key result is that demosaicing followed by JPEG compression always yields asymmetric distributions for images of arbitrary widths and heights even when the input distribution of these images is symmetric.*** Since the combination of these two operations is very standard in imaging pipelines, we can expect results on synthetic data to apply to real images as well.

When the commutative residual is non-zero, we hypothesize that in practice symmetries will be broken, i.e., a non-commutative imaging process will make an originally achiral distribution chiral. To test this hypothesis, we trained deep neural networks on three synthetic achiral distributions of Gaussian noise images, corresponding to three different square images sizes: one with odd width (99×99), one with even width **not** divisible by 8 (100×100), and one that is a multiple of 16 (112×112). To generate a sample image from each distribution, for each pixel, we sample its color value from a per-channel Gaussian distribution. The mean of each color channel (in the range $[0, 1]$) was set to $(0.6, 0.5, 0.9)$ (for red, green, and blue, respectively), and the standard deviation to $(0.3, 0.25, 0.4)$. We use per-channel means and standard deviations (rather than the same Gaussian distribution for all channels) to reduce the source of symmetries present other than symmetry with respect to $\mathbf{T}$. An example image from this distribution, before and after each processing step, is shown in Figure 11.

If we apply each processing operation to all images from these three distribution over three image sizes, our hypothesis predicts that the operations will either preserve or break achirality according to Table 3.

We train a binary chirality prediction (flip/no-flip) network using the same ResNet model as in the main paper (with randomly initialized weights) for each of these nine datasets (3 image sizes times 3 processing operations), with learning rates obtained from log-scale grid searches. As predicted by our hypothesis, trained network models can never achieve more than 50% test classification accuracy on processed distributions that our analysis suggests to be achiral (i.e., commutative residual is zero). And, intriguingly, our trained network models achieve near perfect classification accuracy on processed distributions resulting from non-commutative imaging processes. This experiment hence gives empirical evidence that non-commutativity of a processing operation strongly suggests a loss of achirality.

Note that this analysis assumes that we use the whole images after Bayer demosaicing and/or JPEG compression, i.e., no cropping. These results nicely mirror the situation of training networks on *real* images with no random cropping, as described in the main paper. Figure 12 shows that networks trained to classify chirality on resized (but not cropped) Instagram images often seem to focus on image
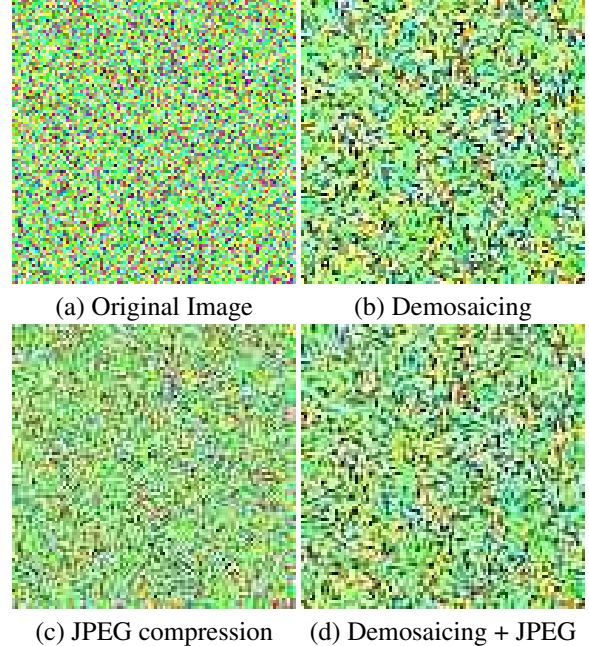
(a) Original Image   (b) Demosaicing

(c) JPEG compression   (d) Demosaicing + JPEG

Figure 11. **A sample image from our Gaussian noise image distribution after different imaging operations**. This image is of size (100,100) and is generated using the Gaussian noise method described in Section 12.1.

Imaging Operation	Image size		
	99	100	112
Demosaicing	A	C	C
JPEG	C	C	A
Demosaicing+JPEG	C	C	C

Table 3. **Predicted chirality of three (initially achiral) Gaussian noise image distributions (corresponding to three different square image sizes) under each of three processing schemes.** 'C' means chiral, and 'A' means achiral. Explanation: 99px images should remain achiral under demosaicing, since the images have odd size. 112px images should remain achiral under JPEG compression since they have size divisible by 16. Everything else becomes chiral as hypothesized. We verify this table empirically by training network models on the nine distributions resulting from these transformations.

evidence near boundaries (first row), which we hypothesis is due exactly to the kinds of chiral boundary artifacts discussed in this section in the context of JPEG compression. On the other hand, training with random cropping data augmentation yields networks that appear to focus on much more high-level features (second row). In the next section, we discuss the interaction of processing with random cropping (or image translation) and how the addition of random cropping can either make a chiral imaging process achiral, or can sometimes still introduce chirality.
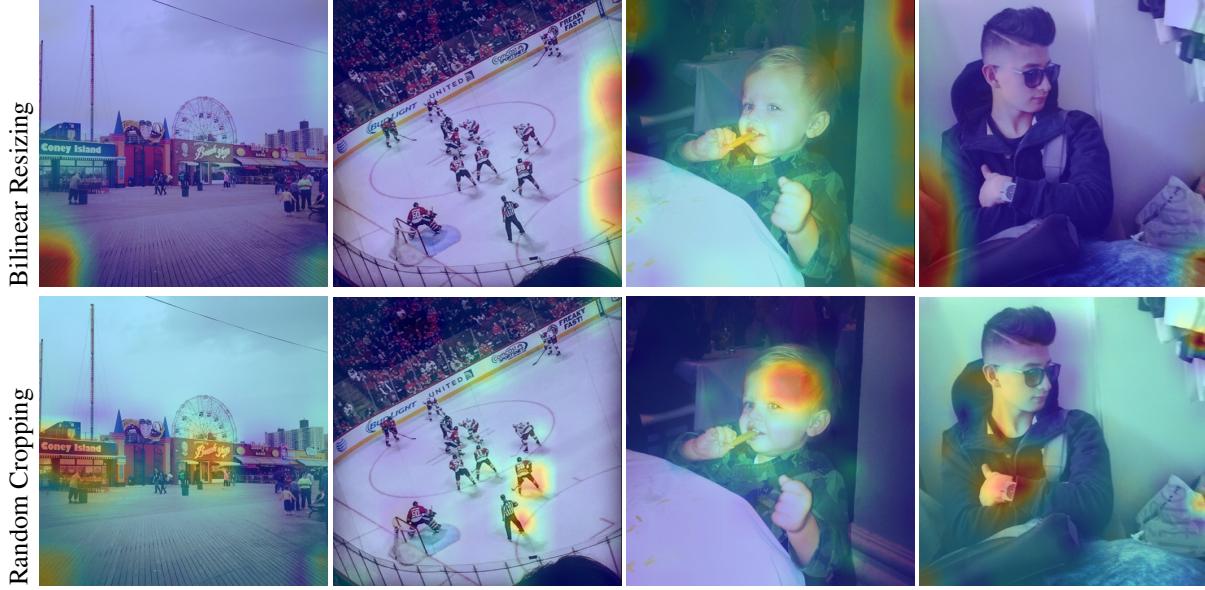
Figure 12. Class Activation Maps (CAM) resulting from two preprocessing procedures used in training ImageNet-pretrained models on the chirality task: (top row) simple bilinear resizing and (bottom row) random cropping. Recall from the main paper that the CAM tends to fire on discriminative regions for classification. Note the heavy focus on edge and corner regions on bilinear resized images, likely due to edge artifacts caused by JPEG compression or demosaicing (or both). These artifacts disappear when random cropping is applied.

Figure 13. **Glide Symmetry**: Human footprints often exhibit glide symmetry. The infinitely repeating footstep pattern shown here is equivalent to the reflection of a shifted version of itself.

13. Random Cropping and Glide Symmetry

Because our analysis makes few assumptions about $\mathbf{T}$, $\mathbf{J}$ and $\mathbf{D}$, we can apply it to other symmetries and data augmentation strategies used in computer vision. For example, translational invariance is a common and useful prior in images that is often applied to data through the use of random crops as a type of data augmentation. Here we consider how our theory can be used to understand the effect of random cropping on training.

13.1. Random Cropping as a Symmetry Transform

Doersch et al.[2] found that when they trained a network to predict the relative position of different regions in an image, it would sometimes "cheat" by utilizing chromatic aberration for prediction. We can use our observation about commutativity to explain this behavior by considering a family of transformations in the 2D image plane. The self-supervision task used in Doersch et al.requires the network to distinguish between different translations, which is only possible when the following symmetry does not hold:

$$\mathbf{D}(\mathbf{x}) = \mathbf{D}(\mathbf{T_v}(\mathbf{x})), \qquad (25)$$

where $\mathbf{T_v}$ is translation by some vector $\mathbf{v} \in \mathbb{R}^2$. Our commutativity analysis tells us that this symmetry can be broken by any $\mathbf{J}$ that does not commute (or glide commute) with translation. This agrees with the findings of Doersch et al.that the network was able to "cheat" using artifacts caused by chromatic aberration, which is not translation-invariant, as its effect is spatially varying.

13.2. Random Cropping as an Image Operation

If we revisit our analysis of commutative residuals under an assumption of translation invariance, we can draw new conclusions about the chirality of demosaicing and JPEG compression. In particular, by incorporating translation invariance in the form of random cropping, we can change the chirality of these operations by creating the kind of permuted commutativity described in Section 10.4. In the case where permuted commutativity happens among groups related by translation, we call it *glide-commutativity*.

To test for glide-commutativity, we must look for the permutation pattern described in Section 10.4. To do this, we first define a way of phase-shifting $\mathbf{T}(\mathbf{J}(\mathbf{x}))$ and $\mathbf{J}(\mathbf{T}(\mathbf{x}))$. For this, we define $\mathbf{JT}_\phi(\mathbf{x})$ and $\mathbf{TJ}_\phi(\mathbf{x})$ as the process of:

1. Padding $\mathbf{x}$ with a large, constant number of pixels on all sides.

2. Translating the padded image by ϕ.

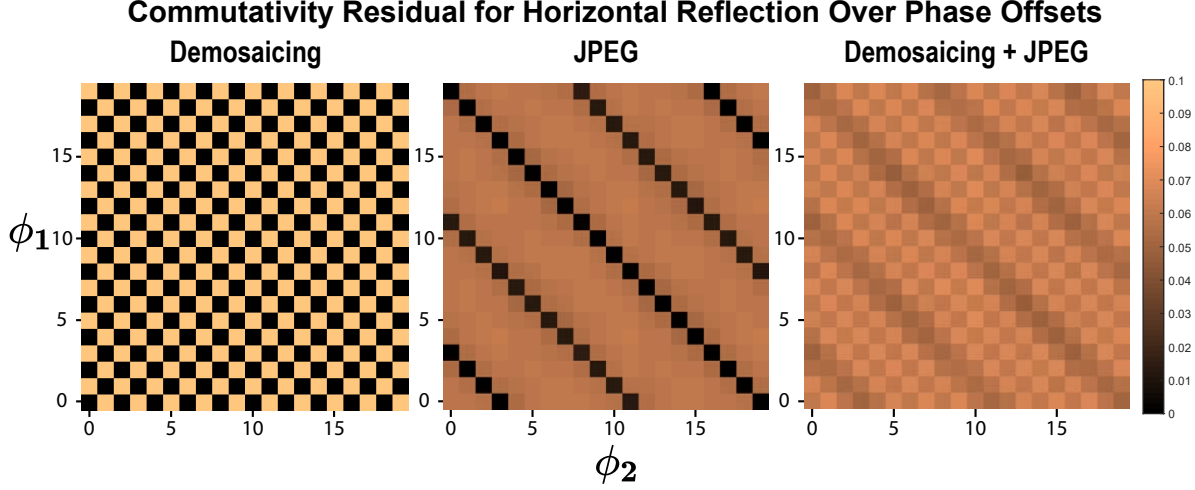**Commutativity Residual for Horizontal Reflection Over Phase Offsets**

Figure 14. **Glide Commutativity Residuals for demosaicing (left), JPEG compression (middle) and their composition (right)**: Each image shows the glide commutativity residual, measured in absolute average percent error per pixel, measured over different phase shifts. For certain ϕ_1 and ϕ_2 we see commutativity in demosaicing and in jpeg compression alone. We do not see commutativity when both are applied.

3. Applying **T** then **J** for $\mathbf{JT}_\phi(\mathbf{x})$, or **J** then **T** for $\mathbf{TJ}_\phi(\mathbf{x})$.

4. Translating by $\mathbf{T}(-\phi)$.

5. Cropping out the previously padded pixels.

This has the effect of performing **J** and **T** as if the image had occurred at a translation of ϕ from its original position. For grid-based algorithms like demosaicing and JPEG compression, this effectively phase-shifts the grid structure used in the algorithm.

To test for glide-commutativity we simply look for some repeating pattern of zeros in residuals of the form:

$$\mathbf{E_J}(\mathbf{x}, \phi_1, \phi_2) = \mathbf{JT}_{\phi_1}(\mathbf{x}) - \mathbf{TJ}_{\phi_2}(\mathbf{x}) \qquad (26)$$

This pattern of zeros describes the permutation pattern described in Section 10.4. As the results in Figure 10 show, we verified that the vertical components of ϕ_1 and ϕ_2 do not matter. We therefore set them only to vary in the x dimension of the image. Figure 14 shows the residuals calculated for a range of phase shifts. We see that both demosaicing and JPEG compression appear to be glide-commutative due to the regular repeating pattern of zeros. However, the combination of demosaicing and JPEG compression does not appear to be glide-commutative, and we can see this is because zeros always occur at different phase shifts for each of the two operations.

13.3. Empirical chirality in the presence of random crops

The analysis from the previous section has simple implications (in terms of random cropping on images): (1) The distribution of random crops (while avoiding cropping from the boundary of 16 pixels) from an originally achiral distribution of images that has undergone either demosaicing or JPEG compression (but not both) should remain achiral. (2) On the other hand, surprisingly, random crops (avoiding a 16-pixel margin around the boundary in the cropped image) on that achiral distribution of images after both demosaicing and JPEG compression may likely become chiral.

To verify this analysis empirically, we again train ResNet models on the same achiral Gaussian distributions as introduced in Section 12.1. Specifically, we take random crops of size (512, 512) from the center (544, 544) of the (576,576) Gaussian noise images to avoid possible boundary effects from a 16-pixel margin. We train separate networks on each of the three output image distributions obtained from applying each of the three imaging operations (demosaicing, JPEG compression, and composition of demosaicing followed by JPEG compression) on the initial Gaussian noise image distribution. Note that, as before, we perform a log-scale grid search over learning rates.

The network training results show that neither demosaicing nor JPEG compression alone is sufficient to produce a chiral distribution under random cropping: models trained with such images fail to achieve more than 50% accuracy. This suggests that chirality is preserved when those operations are applied in isolation. But, as our theory predicts, when both operations are applied the image distribution becomes chiral: the trained network achieves 100% training and test accuracy. This supports our theoretical analysis of the glide-commutativity. Together, our analysis and empirical study suggest that chiral traces are left in photographs via the Bayer demosaicing and JPEG compression imaging processes.

14. Conclusion

In this document we have developed theory relating the preservation of symmetry by various operations to their commutativity with corresponding symmetry transformations. We proposed the commutative residual as a tool for analyzing symmetry preservation, and predicting how different operations will affect the results of deep learning. We also extend our theory to random cropping and show how to evaluate glide commutativity to detect permuted commutativity. Our theoretical analysis and empirical experiment suggest that when demosaicing and JPEG compression are applied together, achiral distributions can becomes chiral, which has implications on several areas, including self-supervised learning, image forensics, data augmentation.

References

[1] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. *ICCV*, 2015. 3, 4

[2] H. S. Malvar, L. wei He, and R. Cutler. High-quality linear interpolation for demosaicing of bayer-patterned color images. *IEEE IntL Conf. on Acoustics,Speech,and Signal Procesing*, March 2004.