

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

Zhiqiu Lin¹

Jin Sun^{1,2}

Abe Davis^{1,2}

Noah Snavely^{1,2}

Cornell University¹

Cornell Tech²

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.

1. 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 2). Next, in Section 3, 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 3, 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 4). In Section 5 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 6, we extend 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.

2. 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.

2.1. 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* $\mathbf{D}_\mathbf{J} : \mathbb{R}^n \mapsto \mathbb{R}$ obtained by applying \mathbf{J} to the elements of \mathbf{D} :

$$\mathbf{D}_\mathbf{J}(\mathbf{y}) = \sum_{\mathbf{x} \in \mathbf{J}^{-1}(\mathbf{y})} \mathbf{D}(\mathbf{x}). \quad (1)$$

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. $\mathbf{D}_\mathbf{J}$ 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 $\mathbf{D}_\mathbf{J}$ is the distribution approximated by $\mathbf{J}(\mathbf{X})$, which we get by applying JPEG compression to every image in \mathbf{X} . The summation in Equation 1 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 1.

2.2. 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:

$$\mathbf{x} = \mathbf{T}\mathbf{x} \quad (2)$$

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

$$\mathbf{D}(\mathbf{x}) = \mathbf{D}(\mathbf{T}\mathbf{x}) \quad (3)$$

Which makes the distribution \mathbf{D} itself symmetric if and only if Equation 3 holds for all \mathbf{x} .

Importantly, Equation 3 can hold even when Equation 2 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.

3. 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 2 and 3 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 2 to both \mathbf{x} and $\mathbf{J}(\mathbf{x})$:

$$[\mathbf{x} = \mathbf{T}\mathbf{x}] \implies [\mathbf{J}(\mathbf{x}) = \mathbf{T}\mathbf{J}(\mathbf{x})] \quad (4)$$

The second type of symmetry preservation describes whether the symmetry of a distribution is preserved. It is defined by applying \mathbf{J} to Equation 3:

$$[\mathbf{D}(\mathbf{x}) = \mathbf{D}(\mathbf{T}\mathbf{x})] \implies [\mathbf{D}_\mathbf{J}(\mathbf{x}) = \mathbf{D}_\mathbf{J}(\mathbf{T}\mathbf{x})] \quad (5)$$

Note that neither of Equations 4 and 5 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.

3.1. Commutativity & Element Symmetry

Our first type of symmetry preservation describes whether elements that are symmetric with respect to \mathbf{T} remain so after applying \mathbf{J} . We now show that this holds if and only

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 <i>not</i> have to be invertible	Some combination of image processing operations, e.g., demosaicing and/or JPEG compression.
A transformed distribution $\mathbf{D}_\mathbf{J}$	$\mathbf{D}_\mathbf{J}(\mathbf{y}) = \sum_{\mathbf{x} \in \mathbf{J}^{-1}(\mathbf{y})} \mathbf{D}(\mathbf{x})$	The distribution of training data after every element has been transformed by \mathbf{J} .

Table 1. Terms and definitions used in derivations.

if \mathbf{J} commutes with \mathbf{T} when applied to symmetric elements, meaning:

$$[\mathbf{x} = \mathbf{T}\mathbf{x}] \implies [\mathbf{J}(\mathbf{T}\mathbf{x}) = \mathbf{T}\mathbf{J}(\mathbf{x})] \quad (6)$$

Proposition 1. *\mathbf{J} will preserve the symmetry of elements with respect to \mathbf{T} (Equation 4) if and only if \mathbf{T} and \mathbf{J} commute on symmetric elements (Equation 6).*

Proof. We start by showing that Equation 4 implies Equation 6. As equivalence implies equality under \mathbf{J} , we have:

$$[\mathbf{x} = \mathbf{T}\mathbf{x}] \implies [\mathbf{J}(\mathbf{x}) = \mathbf{J}(\mathbf{T}\mathbf{x})] \quad (7)$$

And combining the right sides of Equations 4 and 7 gives us Equation 6. To show the other direction we start by applying \mathbf{J} to both sides of Equation 2 to get the right side of Equation 7. From here we use Equation 6 to commute \mathbf{T} and \mathbf{J} and get Equation 4. This concludes the proof. \square

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

3.2. 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:

$$\mathbf{x}_b = \mathbf{T}\mathbf{x}_a \quad (8)$$

We can think of Equation 8 as relaxing Equation 2 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:

$$[\mathbf{x}_b = \mathbf{T}\mathbf{x}_a] \implies [\mathbf{J}(\mathbf{x}_b) = \mathbf{T}\mathbf{J}(\mathbf{x}_a)] \quad (9)$$

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

Proposition 2. *\mathbf{J} preserves the mapping established by \mathbf{T} (Equation 9) if and only if \mathbf{J} commutes with \mathbf{T} .*

Proof. We start by showing that Equation 9 implies commutativity. As equivalence implies equality under \mathbf{J} , we have:

$$[\mathbf{x}_b = \mathbf{T}\mathbf{x}_a] \implies [\mathbf{J}(\mathbf{x}_b) = \mathbf{J}(\mathbf{T}\mathbf{x}_a)] \quad (10)$$

And combining the right sides of Equations 9 and 10 gives us $\mathbf{T}\mathbf{J}(\mathbf{x}_a) = \mathbf{J}(\mathbf{T}\mathbf{x}_a)$. To show that commutativity implies Equation 9, we start by applying \mathbf{J} to both sides of Equation 8 to get the right side of Equation 10

$$[\mathbf{x}_b = \mathbf{T}\mathbf{x}_a] \implies [\mathbf{J}(\mathbf{x}_b) = \mathbf{J}(\mathbf{T}\mathbf{x}_a)] \quad (11)$$

From here we commute \mathbf{T} and \mathbf{J} on the right side to get Equation 9. 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 9 does not hold.

3.3. 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 $\mathbf{D}_\mathbf{J}$ 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 $\mathbf{D}_\mathbf{J}(\mathbf{x})$ to $\mathbf{D}_\mathbf{J}(\mathbf{T}(\mathbf{x}))$.

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} :

$$\langle \mathbf{x}_i \rangle^\mathbf{T} = \{ \dots, \mathbf{T}^{-1}\mathbf{x}_i, \mathbf{x}_i, \mathbf{T}\mathbf{x}_i, \mathbf{T}^2\mathbf{x}_i, \mathbf{T}^3\mathbf{x}_i, \dots \} \quad (12)$$

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:

$$\mathbf{T}^a \mathbf{x}_i \cdot \mathbf{T}^b \mathbf{x}_i = \mathbf{T}^{a+b} \mathbf{x}_i \quad (13)$$

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 $\langle \mathbf{x}_i \rangle^{\mathbf{T}}$:

$$\mathbf{J} \langle \mathbf{x}_i \rangle^{\mathbf{T}} = \{ \dots \mathbf{J} \mathbf{T}^{-1} \mathbf{x}_i, \mathbf{J} \mathbf{x}_i, \mathbf{J} \mathbf{T} \mathbf{x}_i, \mathbf{J} \mathbf{T}^2 \mathbf{x}_i, \dots \} \quad (14)$$

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

$$\mathbf{J} \langle \mathbf{x}_i \rangle^{\mathbf{T}} = \{ \dots \mathbf{T}^{-1} \mathbf{J} \mathbf{x}_i, \mathbf{J} \mathbf{x}_i, \mathbf{T} \mathbf{J} \mathbf{x}_i, \mathbf{T}^2 \mathbf{J} \mathbf{x}_i, \dots \} \quad (15)$$

giving us

$$\mathbf{J} \langle \mathbf{x}_i \rangle^{\mathbf{T}} = \langle \mathbf{J} \mathbf{x}_i \rangle^{\mathbf{T}} \quad (16)$$

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} :

$$\begin{aligned} \mathbf{J}(\mathbf{T}^a \mathbf{x}_i \cdot \mathbf{T}^b \mathbf{x}_i) &= \mathbf{J}(\mathbf{T}^{a+b} \mathbf{x}_i) \\ &= \mathbf{T}^{a+b} \mathbf{J}(\mathbf{x}_i) \\ &= \mathbf{T}^a \mathbf{J}(\mathbf{x}_i) \cdot \mathbf{T}^b \mathbf{J}(\mathbf{x}_i) \\ &= \mathbf{J}(\mathbf{T}^a \mathbf{x}_i) \cdot \mathbf{J}(\mathbf{T}^b \mathbf{x}_i) \end{aligned} \quad (17)$$

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:

$$\mathbf{D} = \sum_{\mathbf{x}_i \in \mathcal{G}_{\mathbf{T}}} \mathbf{D}(\mathbf{x}_i) \cdot \mathbf{1}_{\langle \mathbf{x}_i \rangle^{\mathbf{T}}} \quad (18)$$

Now, using the first isomorphism theorem to account for the case where \mathbf{J} is non-injective, we can combine Equation 1 and 18 to write $\mathbf{D}_{\mathbf{J}}$ as

$$\mathbf{D}_{\mathbf{J}} = \sum_i (\mathbf{D}(\mathbf{x}_i) |\ker \mathbf{J} \mathbf{x}_i|) \cdot \mathbf{1}_{\langle \mathbf{J} \mathbf{x}_i \rangle^{\mathbf{T}}} \quad (19)$$

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

3.4. 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}_j 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 :

$$\mathbf{D}_{\mathbf{J}}(\mathbf{x}) = \sum_j \sum_{\mathbf{x}_i : \mathbf{J}_j \mathbf{x}_i = \mathbf{x}} \mathbf{D}(\mathbf{x}_i) \quad (20)$$

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

¹Recall 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 20, we can define a new set of transformations that sum to the same \mathbf{D}_J , ensuring that symmetry remains preserved.

$$\begin{aligned}\langle \mathbf{x}_i \rangle^T &= \mathbf{x}_i \quad \mathbf{T}\mathbf{x}_i \quad \mathbf{T}^2\mathbf{x}_i \quad \mathbf{T}^2\mathbf{x}_i \quad \dots \\ \mathbf{J}_1\langle \mathbf{x}_i \rangle^T &= \mathbf{J}_1\mathbf{x}_i \quad \mathbf{J}_1\mathbf{T}\mathbf{x}_i \quad \mathbf{J}_1\mathbf{T}^2\mathbf{x}_i \quad \mathbf{J}_1\mathbf{T}^2\mathbf{x}_i \quad \dots \\ \mathbf{J}_2\langle \mathbf{x}_i \rangle^T &= \mathbf{J}_2\mathbf{x}_i \quad \mathbf{J}_2\mathbf{T}\mathbf{x}_i \quad \mathbf{J}_2\mathbf{T}^2\mathbf{x}_i \quad \mathbf{J}_2\mathbf{T}^2\mathbf{x}_i \quad \dots \\ \mathbf{J}_3\langle \mathbf{x}_i \rangle^T &= \mathbf{J}_3\mathbf{x}_i \quad \mathbf{J}_3\mathbf{T}\mathbf{x}_i \quad \mathbf{J}_3\mathbf{T}^2\mathbf{x}_i \quad \mathbf{J}_3\mathbf{T}^2\mathbf{x}_i \quad \dots\end{aligned}$$

Figure 1. Permuted Commutativity. We can permute the elements on the right side of Equation 20 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 maintains symmetry when $\mathbf{T}\mathbf{J}_a = \mathbf{J}_b\mathbf{T}$ for a and b are indices of a found permutation (Figure 1). We define the commutativity under a particular kind of permutation obtained from random cropping, as *glide commutativity*.

4. 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 asymmetries introduced by an operation. We define $\mathbf{E}_J(\mathbf{x})$, the *commutative residual image* of operation J with respect to transformation T on the image \mathbf{x} , as follows:

$$\mathbf{E}_J(\mathbf{x}) = \mathbf{J}(\mathbf{T}(\mathbf{x})) - \mathbf{T}(\mathbf{J}(\mathbf{x})) \quad (21)$$

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, $\hat{\mathbf{e}}_J(\mathbf{x})$. We refer to $\hat{\mathbf{e}}_J(\mathbf{x})$ 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 3 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:

$$\mathbf{J}(\mathbf{T}(\mathbf{x})) - \mathbf{T}(\mathbf{J}(\mathbf{x})) \quad (22)$$

which is precisely how we define the commutative residual image above. Figure 2 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.

4.1. 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} .

5. Analysis of Demosaicing & JPEG Compression

With the theoretical tools derived in previous sections, we evaluate two standard imaging processes: Bayer demo-

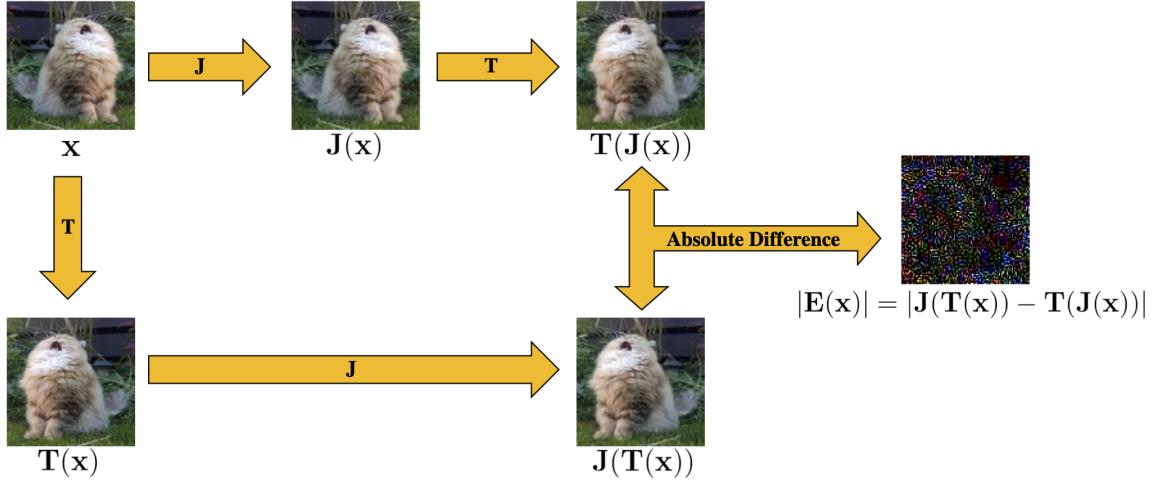


Figure 2. Example commutative residual image: This figure illustrates the application of the commutative residual method to a natural image. Here T is the horizontal reflection operation, and \mathbf{J} is the composition of Bayer demosaicing and JPEG compression. The image used above has a width of 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 4, the residual image is not zero (which would be all black), i.e., the commutative residual is non-zero.

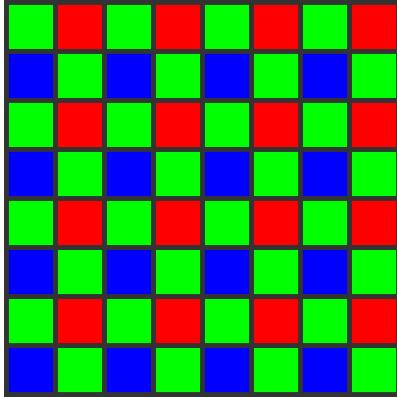


Figure 3. Example 8×8 Bayer pattern mosaic: A typical Bayer filter mosaic consists of tiled 2×2 blocks of pixels with two green filters and one red and one blue filter. Note that an even-sized Bayer filter, like the one pictured, is asymmetric (mirror flipped version is not equal to itself), while an odd-sized version of this filter pattern would be symmetric.

saicing 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 cam-

eras (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 3. 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 *demosacing*. 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 3) and we consider the demosaicing method of Malvar [2].

The 8×8 Bayer filter mosaic in Figure 3 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 images into the $Y' C_b C_r$ 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

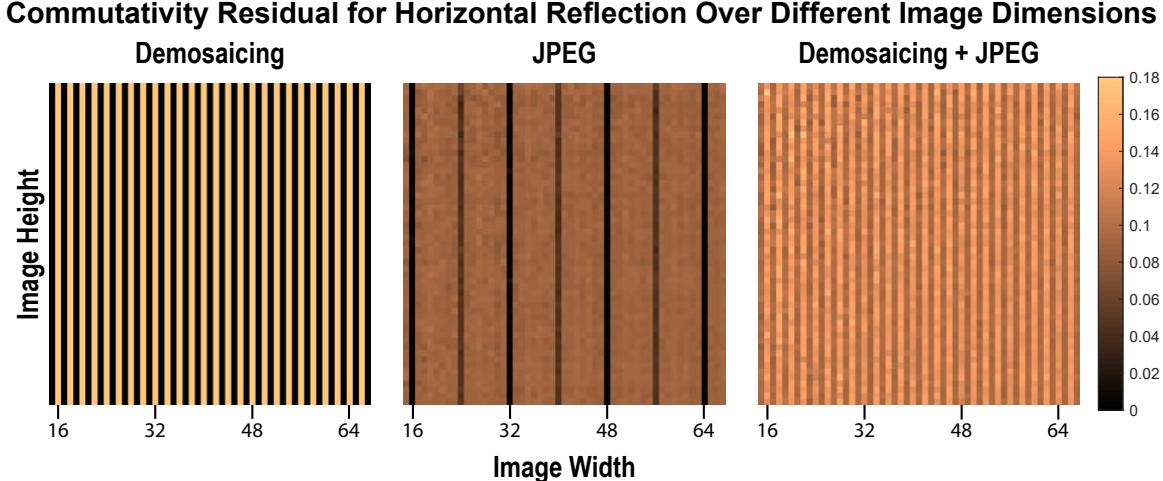


Figure 4. Commutativity residuals for demosaicing (left), JPEG compression (middle) and their composition (right): Each image shows how commutativity residual, measured in absolute average percent error per pixel, varies with different image sizes. For integers n we see commutativity in demosaicing at 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.

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× 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.

5.1. 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 4 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 images 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 3) 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 analysis² of commutative residuals shown in Figure 4 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 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 distribu-**

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

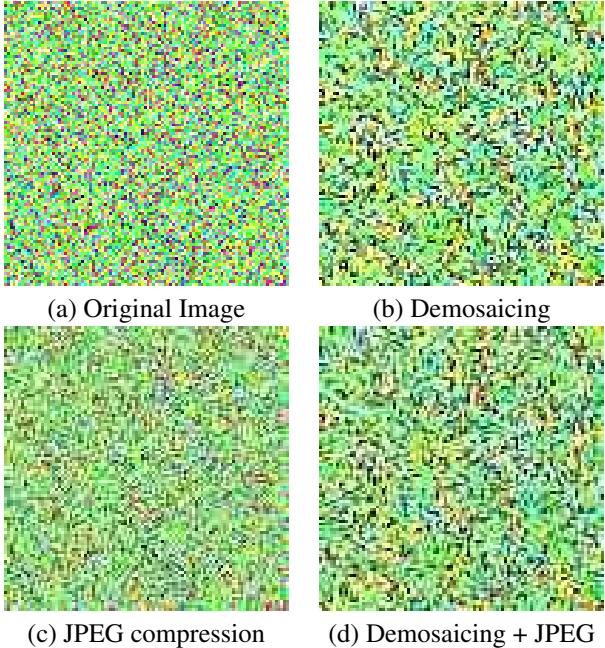


Figure 5. A sample image from our Gaussian noise image distribution after different imaging operations. This image is of size (100,100) and is generated using the Gaussian noise method described in Section 5.1.

tion 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 5.

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 2.

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

Table 2. 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.

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 6 shows that networks trained to classify chirality on resized (but not cropped) Instagram images often seem to focus on image 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.

6. 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 aug-



Figure 6. 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.

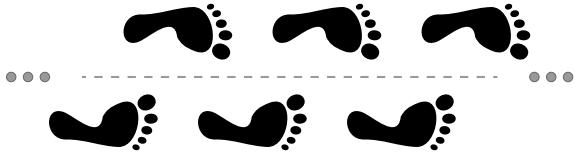


Figure 7. **Glide Symmetry:** Human footprints often exhibit glide symmetry. The infinitely repeating footstep pattern shown here is equivalent to the reflection of a shifted version of itself.

mentation 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.

6.1. Random Cropping as a Symmetry Transform

Doersch *et al.* [1] 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:

$$\mathbf{D}(\mathbf{x}) = \mathbf{D}(\mathbf{T}_v(\mathbf{x})), \quad (23)$$

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.

6.2. 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 3.4. 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 3.4. 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 $\mathbf{J}\mathbf{T}_\phi(\mathbf{x})$ and $\mathbf{T}\mathbf{J}_\phi(\mathbf{x})$ 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 .
3. Applying \mathbf{T} then \mathbf{J} for $\mathbf{J}\mathbf{T}_\phi(\mathbf{x})$, or \mathbf{J} then \mathbf{T} for $\mathbf{T}\mathbf{J}_\phi(\mathbf{x})$.
4. Translating by $\mathbf{T}(-\phi)$.

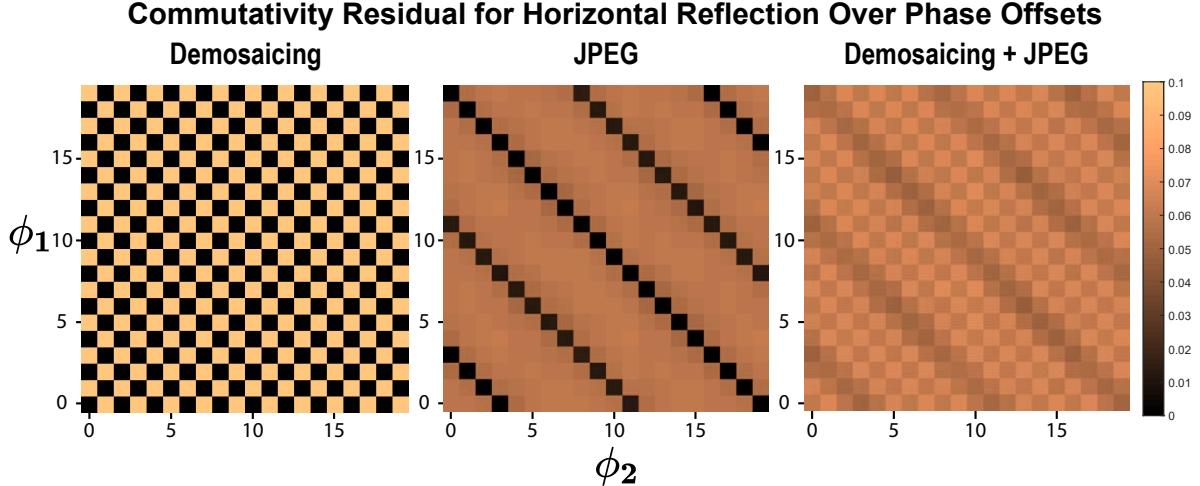


Figure 8. Glide Commutativity Residuals for demosaicing (left), JPEG compression (middle) and their composition (right): Each image shows the glide commutativity residual, measured in absolute average percent error per pixel, measured over different phase shifts. For certain ϕ_1 and ϕ_2 we see commutativity in demosaicing and in jpeg compression alone. We do not see commutativity when both are applied.

5. Cropping out the previously padded pixels.

This has the effect of performing \mathbf{J} and \mathbf{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:

$$\mathbf{E}_\mathbf{J}(\mathbf{x}, \phi_1, \phi_2) = \mathbf{J}\mathbf{T}_{\phi_1}(\mathbf{x}) - \mathbf{T}\mathbf{J}_{\phi_2}(\mathbf{x}) \quad (24)$$

This pattern of zeros describes the permutation pattern described in Section 3.4. As the results in Figure 4 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 8 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.

6.3. 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 5.1. 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.

7. 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. [9](#)
- [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 Processing*, March 2004. [6](#)