INVARIANCE MAKES A DIFFERENCE: DISENTANGLING THE ROLE OF INVARIANCE AND EQUIVARIANCE IN REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Representations learned by deep neural networks are the foundation that enables their tremendous success and consequently a lot of work has been invested into understanding their properties. Most of this work, however, focuses on the relationships between representations and features in the input without explicitly characterizing their nature, *i.e.* whether they are *invariances* or *equivariances*. In this work, we concretely define and disentangle these relationships and show with carefully controlled experiments that, in fact, invariance is of central importance in achieving high generalization on downstream tasks, often more so than equivariance. To this end, we investigate the properties and performance of image classification models on synthetic datasets that we introduce and which allow us to precisely control factors of variation in the models' training and test data. With this method we explore a) the role of invariance in enabling high performance when transferring to target tasks and b) the factors that influence which invariances a model learns. We highlight the importance of representational invariance by showing that the representations learned by classification models transfer well to new classes but perform poorly when the required invariances change, and that learning the wrong invariances can be harmful. Additionally, we find that the invariances learned by models are primarily determined by the relationship of features in the training data with the training objective and that there are inductive biases that make certain invariances more difficult to learn than others.

1 INTRODUCTION

Deep Learning has achieved remarkable success, in large part due to the ability of deep neural networks to learn representations of inputs that enable high downstream performance. As a result, a large amount of work has been invested into studying the properties of representations (Bengio et al., 2013; Engstrom et al., 2019; Hermann & Lampinen, 2020). The majority of these efforts investigate the relationship between features of inputs and the representations learned by models (Kim et al., 2018; Zeiler & Fergus, 2013; Lundberg & Lee, 2017).

The two most interesting types of feature-representation relationships are *invariances* and *equivariances*. Intuitively, an invariance in a model's representations means that for certain changes to its inputs, its representations remain constant. An equivariance on the other hand is a consistent change in the representations in response to changes in the input. In this work, we are primarily interested in a) the effects that invariance has on the transfer performance of representations, b) how invariance is related to equivariance and c) the factors that influence which invariances a model's representations develop.

A number of prior works have investigated the role of representational invariance in domains such as robustness, generalization and transfer-learning (Azulay & Weiss, 2018; Zhou et al., 2022) and self-supervised learning (SSL). Some of this work, however, is merely interested in invariance as a means to a specific end, for instance the robustness literature is concerned with invariance in order to safeguard model performance against adversarial perturbations (Szegedy et al., 2013; Papernot et al., 2016) or natural image corruptions (Geirhos et al., 2018; Hendrycks & Dietterich, 2019; Taori et al., 2020), and SSL methods train models to be invariant to certain data transformations in order

to obtain pseudo labels (Doersch et al., 2015; Zhang et al., 2016) or to introduce contrast between similar and dissimilar inputs (Chen et al., 2020; Grill et al., 2020; Kim et al., 2020).

The field of generalization studies invariance in representations more generally, however, comprehensively and precisely understanding invariance is hard, especially when using real-world data. To understand invariance at a fine-grained level, it is necessary to know the different ways in which inputs to a model differ, in order to determine how those differences relate to changes in its representations. This, however, is not possible with commonly used real-world datasets such as CIFAR-10 (Krizhevsky et al., 2009), ImageNet (Russakovsky et al., 2015) or VTAB (Zhai et al., 2020), since the only information present in addition to the raw inputs are class labels which only allow for very coarse difference and similarity judgements.

Therefore, to study invariance carefully, we introduce a synthetic dataset, Transforms-2D, that allows us to precisely control the differences and similarities between inputs in a models training and test sets. Using this dataset, we explore the importance of invariance in achieving high downstream performance, as well as the factors that influence representation learning. Concretely, we make the following contributions:

- We evaluate how learning of certain invariances aids transfer to downstream tasks by introducing a synthetic dataset called Transforms-2D, which allows us to carefully control the differences and similarities in the inputs to models. By using this dataset, we are able to train models to exhibit specific invariances and equivariances in their representations and to evaluate the link between the invariance and performance of representations.
- We investigate the connection of invariance to downstream performance and find that it is more important than equivariance, by showing that models trained with the same transformations but different features as a target task perform better than models trained with the same features but different transformations. We further find that invariance to a surplus of required transformations is not a problem but that undesirable invariances can severely undermine downstream performance.
- We investigate the factors that play a role in determining which invariances a model learns and show that the relevance and availability of of features plays a crucial role, that there is a high degree of invariance transfer between classes and that there are inductive biases that make acquiring certain invariances easier than others.

2 Setting and Methodology

We begin by introducing terminology for the setting that underpins our investigation.

2.1 Terminology

Notation. We operate in the standard supervised learning setting and denote by $\mathcal{D} = \{(x_i, y_i)_{i=1}^N\}$ a dataset consisting of N examples $x \in \mathcal{X} = \mathbb{R}^d$ and associated labels $y \in \mathcal{Y} = \{1, \dots, K\}$. The task is to find a function $g : \mathcal{X} \mapsto \mathcal{Y}$ such that g minimizes the empirical risk on $\mathcal{D}$. We find such a g by minimizing the categorical cross-entropy loss over its predictions. For our purpose it is convenient to write g as $g = g_{cls}(g_{rep}(.))$ where $g_{rep} : \mathcal{X} \mapsto \mathcal{Z}$ maps inputs x to representations $z \in \mathcal{Z} = \mathbb{R}^l$ and $g_{cls} : \mathcal{Z} \mapsto \mathcal{Y}$ maps representations to predictions $\hat{y}$. In the following, we will refer to g_{rep} simply as g if the meaning is clear from the context.

Features and transformations. The term *feature* is commonly used in the Machine Learning literature to refer to properties of model inputs (Bengio et al., 2013). In the strictest sense, these properties are the numerical values of the dimensions of datapoints x. Here, we use a slightly wider interpretation of the term and also refer to semantic properties such as the presence or position of an object as features. Note that we use the term feature here to refer to properties of model inputs x, not of its representations z.

In addition to features, we are interested in *transformations* $t : \mathcal{X} \mapsto \mathcal{X}$ that change the values of one or more features of inputs x. Additionally, we also need to be able to refer to changes in the representation space. Therefore, we define an *effect* $u : \mathcal{Z} \mapsto \mathcal{Z}$ as a transformation acting on representations z.

Invariance and equivariance. In this work, we are primarily concerned with the relationships between transformations t acting on inputs x and their corresponding effects u on representations z. In particular, we want to know whether these relationships constitute *invariances* or *equivariances*. We say that the representations of model g are *invariant* to transformation t if $g(t(x)) = g(x) \, \forall x \in \mathcal{X}$. Further, we say that the representations of model g are *equivariant* to transformation t if for some non-constant effect u, $g(t(x)) = u(g(x)) \, \forall x \in \mathcal{X}$. In the remainder of the paper we will use these definitions to investigate the importance, sources and effects of in- and equivariance.

2.2 Controlling data transformations via synthetic data

To properly study invariance, it is important to know which transformations are present in a dataset and how specific datapoints are transformed, relative to others. For instance, to determine whether the representations of a model are invariant to a particular transformation, it is necessary to probe it with inputs that only differ by this transformation. This is very difficult to achieve with commonly used real-world datasets since there are many different transformations by which different samples differ, such as changes in object position, pose, size, texture, illumination, etc. about which no information is known.

Therefore, in order to properly study invariance and equivariance in representations, we introduce a synthetic dataset that allows us to precisely control the features present in the inputs as well as the transformations acting on them. We call our dataset the *Transforms-2D* dataset. Samples are constructed by pasting transformed versions of images of foreground objects (with alpha-mask) onto randomly chosen background images, such that each class consists of exactly one foreground image (its prototype) and class samples are obtained by applying transformations from 8 different categories to the prototypes (translation/movement, rotation, scale, horizontal and vertical flips, blurring, sharpening and color jitter) as well as sampling random backgrounds. Note that for each category there are many different transformations, for example there are many different ways in which an object can be moved. The foreground and background images are based on the SI-score dataset (Djolonga et al., 2021). We use all the background images from this dataset but only keep one image per foreground category. Additional information on how the dataset is constructed as well as additional examples can be found in Appendix A.

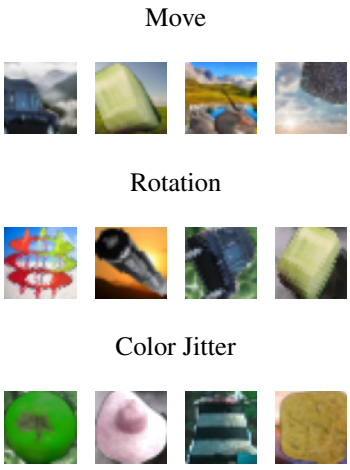


Figure 1: **[Transforms-2D examples.]** The synthetic dataset images with controlled transformations used to train and evaluate the invariances and equivariances of representations.

Examples images drawn from the dataset are shown in Figure 1. All images have a size of 32 x 32 pixels and if not stated differently, we use 50,000 train, 10,000 validation and 10,000 test samples to train models, mimicking the size of the CIFAR-10 dataset.

3 How important is invariance in representations?

With our synthetic dataset at hand, we now turn to investigating the importance of invariance in the representations of DNNs. We are interested in understanding how important invariances are for downstream applications, *i.e.* when transferring representations with certain invariances to different tasks, and what their relative importance is compared to equivariance.

3.1 Invariance is more important for transfer performance than equivariance

We want to understand, whether it is more important that a model's representations change based on features that should separate classes in a classification task (equivariance) or whether it is more important that they do not change based on changes to features that should not separate them (invariance).

To answer this question, we train a model g on a datasets $\mathcal{D}_{train}$ based on the Transforms-2D dataset that contains foreground objects $o \in O_1$ which are transformed by a set of transformations $t \in T_1$. Additionally, we create evaluation datasets that differ in the properties that they share with $\mathcal{D}_{train}$: $\mathcal{D}_{all} = \mathcal{D}_{train}$ uses the same set of object O_1 and transformations T_1 as $\mathcal{D}_{train}$, $\mathcal{D}_{transforms}$ uses the same T_1 but a different set of objects O_2, $O_1 \cap O_2 = \emptyset$, $\mathcal{D}_{objects}$ uses the same set of objects O_1 but a different set of transformations T_2, $T_1 \cap T_2 = \emptyset$ and $\mathcal{D}_{none}$ uses both different objects O_2 and different transformations T_2. For example, O_1 might consist of prototype images of dogs, ships and barbells whereas O_2 might consist of cars, pizzas and bananas. T_1 might contain translations and blurring transformations whereas T_1 might consist of rotations and color changes. In our experiments we choose $|O_1| = |O_2| = 30$ and $|T_1| = |T_2| = 3$ (*i.e.* there are 3 different types of concatenated transformations in T_1 and T_2, not 3 total transformations) and sample prototype objects and transformations randomly.

To perform well at classification on $\mathcal{D}_{train}$, g's representations must become equivariant to the objects in O_1 and invariant to the transformations in T_1, *i.e.* training g on $\mathcal{D}_{train}$ allows us to control which invariances and equivariances its representations exhibit. To understand the relative importance of invariance compared to equivariance, we evaluate how well g transfers to $\mathcal{D}_{transforms}$ and $\mathcal{D}_{objects}$, with which it shares either the transformations or the objects in its training dataset, respectively, and which therefore either require the same invariances but different equivariances or the same equivariances but different invariances. $\mathcal{D}_{all}$ and $\mathcal{D}_{none}$ serve as reference. We evaluate g's transfer performance by freezing its weights up to the penultimate layer, keeping its representations the same, fine-tune the last linear layer on the target dataset and then compute its accuracy. Additional details on the training and evaluation setup can be found in Appendix B. Depending on whether g performs better on $\mathcal{D}_{transforms}$ or on $\mathcal{D}_{objects}$ after fine-tuning, invariance or equivariance is the more important property in determining transfer performance.

In Figure 2 we show the transfer performance of models on the different evaluation datasets. The results show that models trained on data with the same transformations as the target task perform better on it, compared to models trained with the same objects but with different transformations. In fact, the performance when transferring to datasets with the same transformations is close to that on the training dataset, whereas when the target dataset differs from the training dataset in its transformations and therefore its required invariances, transfer performance is similarly low both with and without training objects. This relationship clearly indicates that the absence or presence of shared transformations with the training data is the determining factor for transfer performance, whereas the degree of shared objects has little impact. This means that, indeed, learning the right invariances is more important for downstream performance than learning the right equivariances.

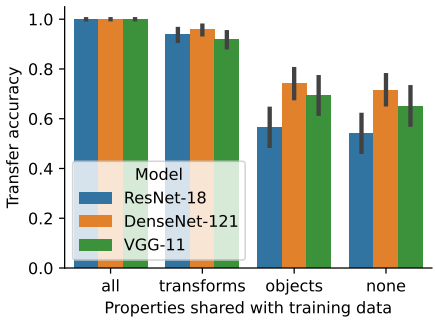


Figure 2: **[Invariance is more important for transfer than equivariance]** Transfer performance of different architectures trained on the Transforms-2D dataset on tasks with which their pre-training dataset shares both transformations and objects, only transformations, only objects or none of these properties. Models trained to be invariant to the same transformations as the target task perform better than models trained to recognize the same objects, but with different transformations, indicating that invariance to the right transformations is a more important property for representations than equivariance to the right features. All numbers reportedt in the paper are aggregated over 10 runs.

These results have interesting implications in the context of generalization and robustness to distribution shift (Koh et al., 2021; Taori et al., 2020). They show that models can generalize and adapt well as long as the set of transformations that they need to be invariant to remains the same, but they will perform poorly if different invariances are required.

In Appendix C.1 we show that learning more than the necessary invariances (as long as they should not be equivariances) is not a problem for transfer performance. Models trained on a superset of transformations of the target task perform well on it, whereas the performance of models that only saw a subset of the target transformations deteriorates significantly. We also find that the order in

which transformations are applied has only a small effect on model performance, which is unsurprising as most (but not all) transformations used in the Transforms-2D dataset commute.

3.2 THERE IS SUCH A THING AS TOO MUCH INVARIANCE

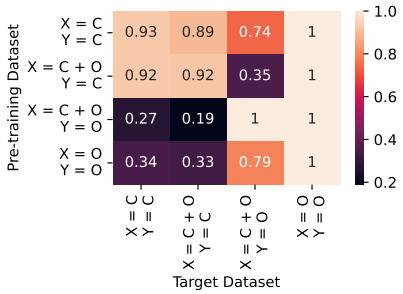


Figure 3: **[The impact of irrelevant features on downstream performance.]** Models trained with access to features that are irrelevant for their target task (objects for $X = C + O, Y = C$ and CIFAR backgrounds for $X = C + O, Y = O$) generalize worse to tasks where those features are important than their counterparts that did not have access to those features.

The previous results highlight the importance of representations having the right invariances. However, this importance cuts both ways. While section 3.1 shows that a *lack of desired invariance* in representations can be harmful for downstream performance, we now show that a *surplus of undesirable invariance* can be just as bad.

To understand the effect of undesirable invariance, we augment the training set of a classification task by adding information to the inputs that is irrelevant for classifying them correctly. Then, we evaluate how well a model trained on this augmented dataset can predict the irrelevant information by fine-tuning its last layer to predict it. We compare its performance to another model that has been trained on the unmodified dataset.

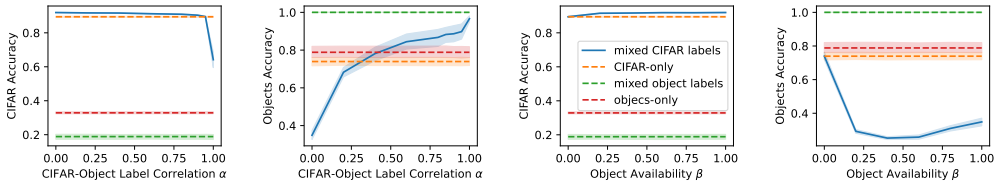
Concretely, we augment the CIFAR-10 dataset (Krizhevsky et al., 2009) by pasting small versions of the foreground objects from the Transforms-2D dataset described in section 2.2 onto the images in a completely random manner, *i.e.* uncorrelated with the CIFAR-10 labels. In total, we use four datasets which differ in their combinations of features X and labels Y: the standard CIFAR-10 dataset ($X = C, Y = C$) the CIFAR-10 dataset with random objects pasted on it with the task of predicting CIFAR-10 classes ($X = C + O, Y = C$), the same augmented CIFAR-10 dataset with the task of predicting the category of the pasted objects ($X = C + O, Y = O$) and a dataset with only objects pasted on a black background, with the task of predicting the object category ($X = O, Y = O$). We use 10 object prototypes that are scaled down and pasted in the upper right corner. Examples of images and additional information about the dataset can be found in Appendix A.1. We train models on each of the datasets and then fine-tune and evaluate each model's last layer on each of the other datasets.

Figure 3 shows the results of this process. Interestingly, both models trained on the augmented CIFAR images ($X = C + O$) perform quite poorly on the same augmented dataset when the prediction task Y changes, in fact considerably worse than their counterparts that were trained to predict the same Y, but not on the augmented CIFAR images. In other words, seeing a certain feature (pasted objects or CIFAR backgrounds) during the training while not being relevant makes a model perform worse at predicting it than another model trained for the same task but without access to the feature.

The takeaway from these results is that invariances are a very important property of representations, to the extend that when representations exhibit an invariance to a feature that is needed in a downstream task, it severely hampers their utility for that task. Therefore, care must be taken when training models, particularly when the goal is to use them as feature extractors for downstream tasks, to ensure they do not pick up undesirable invariances.

4 HOW DO MODELS LEARN INVARIANCES?

So far, we have shown that invariances are important properties of representations. In this section, we study the process by which models acquire invariances, which factors control what invariances are learned and whether models have inductive biases towards acquiring certain types of invariances.



(a) Correlation-dependent transfer performance to CIFAR-10 labels

(b) Correlation-dependent transfer performance to object labels

(c) Availability-dependent transfer performance to CIFAR-10 labels

(d) Availability-dependent transfer performance to object labels

Figure 4: **[The effect of feature relevance on learned invariances]** Transfer accuracy on the $X = C + O$ dataset for models trained with different correlation strengths of pasted object categories with CIFAR labels. As the correlation resp. relevance of the pasted objects for the target task increases, models become increasingly better at predicting them and their representations become more sensitive to their presence.

4.1 Which features do models become equi- and invariant to?

To understand which factors determine invariances in representations, we revisit the observations made in Section 3.2. There are two crucial takeaways from the experiment there. First, the two models trained on CIFAR-10 images with objects pasted on them ($X = C + O$) perform very differently after fine-tuning at predicting CIFAR-10 classes ($Y = C$) and object categories ($Y = O$) respectively, depending on whether they were pre-trained to predict the CIFAR-10 classes or object categories, even though they saw exactly the same input data. This result shows that the *relevance* of a feature for a target task is a key driver in determining which input features a model becomes equi- and invariant to.

Second, the models trained on only CIFAR-10 images to predict CIFAR-10 classes ($X = C, Y = C$) and on only object images to predict object categories ($X = O, Y = O$) show better transfer performance on the respective other task than their counterparts that were trained on CIFAR-10 images with pasted objects. These two models did not have access to the object- and CIFAR-features respectively and could therefore not develop invariances towards them, as did the models that saw them during training. This means that another important property in determining the invariances learned by a model is the *availability* of features during training. Next, we investigate the effects of relevance and availability more carefully.

Relevance. To understand how the invariance to features changes with their relevance for the target task, we train models on the dataset described in Section 3.2. However, we now introduce correlation of different strengths between the objects and the CIFAR labels. In particular, we train models g_α on datasets $\mathcal{D}_\alpha$, where $\alpha \in \{0, 0.2, 0.4, 0.6, 0.8, 0.85, 0.9, 0.95, 1.0\}$ is the correlation strength between CIFAR labels and object categories. Input images in $\mathcal{D}_\alpha$ are constructed by pasting one specific object per CIFAR-10 class (out of a set of 10 total objects) on the CIFAR-10 training images α-fraction of the time and pasting a random object otherwise. Then, we fine-tune the g_αs' last layer and evaluate them on the augmented CIFAR-10 images, both for predicting the CIFAR-10 class ($X = C + O, Y = C$) as well as the object category ($X = C + O, Y = O$).

Availability. To investigate the effect of availability, we train models on datasets $\mathcal{D}_\beta, \beta \in \{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ that are equivalent to the $X = C + O, Y = C$ dataset, *i.e.* objects are pasted randomly onto CIFAR backgrounds, but with the difference that objects are only pasted β-fraction of the time for dataset $\mathcal{D}_\beta$. We again fine-tune and evaluate the last layer of the models trained on the $\mathcal{D}_\beta$s on the same datasets as for the relevance analysis.

Figure 4a and Figure 4b show the transfer accuracies of models trained to predict CIFAR-10 classes and object categories, respectively. The blue curve correspond to the models pretrained on $X = C + O$ datasets where the objects were pasted with different correlations with the CIFAR labels. Dashed lines show the performance of reference models. As the correlation of the pasted object categories with the CIFAR-10 classes increases, CIFAR-10 transfer accuracy mostly remains constant, whereas object category accuracy increases steadily. Only when pasted objects and CIFAR labels become perfectly correlated, CIFAR-10 accuracy drops and object detection accuracy reaches

100%. This trend shows that as the pasted objects become more relevant for the target task, the models' representations gradually become less invariant to them and allow for increasingly better prediction of the object category. CIFAR-10 accuracy on the other hand remains constant, up to the point where the pasted object can reliably replace the CIFAR background.

The CIFAR performance does not depend on the availability of the pasted objects, *i.e.* whether or not an irrelevant object is present in the training data has no impact on it. Object category prediction performance on the other hand quickly drops as soon as the model has access to the irrelevant object features, showing that it immediately develops an invariance towards the objects. In summary, relevance and availability both play a role in determining which invariances a model learns, but relevance seems to be the more important property.

4.2 Does invariance transfer between classes?

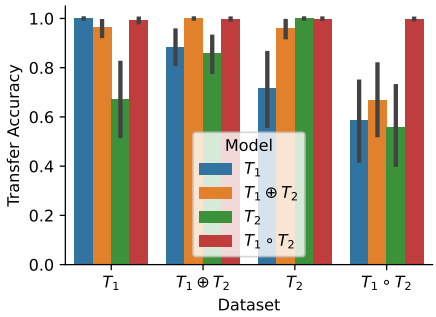


Figure 5: **[Invariance transfer performance]** Models are trained on datasets with only some of the transformations, different transformations for different classes and a concatenation of all transformations and they can generalize invariance from one set of classes to another.

Another question concerning the mechanisms by which models learn invariances is whether invariances learned for a set of classes or objects transfer to other classes. For instance if a model is trained on a dataset with two classes, each consisting of one object prototype that is transformed by a different set of transformations from the other class, will the model transfer the invariances corresponding to each class to the other one? For example, if one class consists of a car prototype that is rotated and the other class consists of a bird whose color is changed, will the model be rotation-invariant on the bird class and color change-invariant on the car class as well?

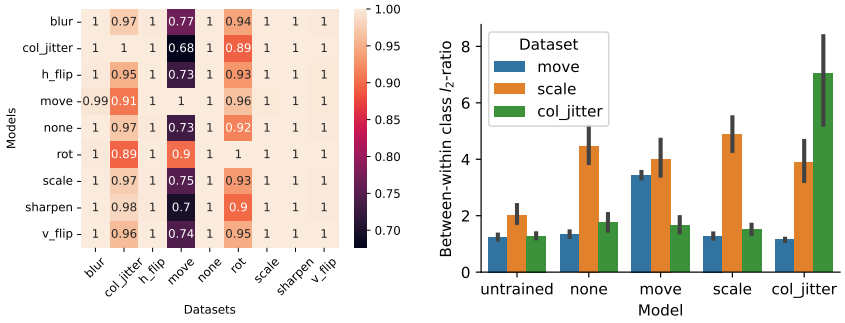
To test whether this is the case, we sample sets of objects O_1, $O_2, O_1 \cap O_2 = \emptyset$ and transformations $T_1, T_2, T_1 \cap T_2 = \emptyset$ and use them to construct three datasets $\mathcal{D}_{T_1}$ with objects $O_1 \cup O_2$ that are all transformed by transformations from T_1, $\mathcal{D}_{T_2}$ with objects $O_1 \cup O_2$ that are all transformed by transformations from T_2, $\mathcal{D}_{T_1 \oplus T_2}$ consisting of objects from O_1 transformed by transformations from T_1 and objects from O_2 transformed by transformations from T_2. For reference, we also include a dataset $\mathcal{D}_{T_1 \circ T_2}$ where both sets of objects $O_1 \cup O_2$ are transformed by the concatenations of transformations from T_1 and T_2. On each of the datasets we train a model and then fine-tune and evaluate its last linear layer on each of the other datasets. We choose $|O_1| = |O_2| = 15$ and $|T_1| = |T_2| = 3$.

Figure 5 shows how models transfer to the different datasets. Both models trained on $\mathcal{D}_{T_1}$ and $\mathcal{D}_{T_2}$ perform well on their respective datasets, but perform poorly on all other datasets. This is to be expected from the results in Section 3.1. The model trained on $\mathcal{D}_{T_1 \oplus T_2}$, however, performs well on both $\mathcal{D}_{T_1}$ and $\mathcal{D}_{T_2}$, which shows that it is able to generalize its learned invariances from each set of objects O_1 and O_2 to the other. Generalization is not perfect, however, and the model achieves slightly lower accuracy than the "native" models on $\mathcal{D}_{T_1}$ and $\mathcal{D}_{T_2}$. On the other hand, all models other than the one trained on $\mathcal{D}_{T_1 \circ T_2}$ achieve poor accuracy on that dataset, especially also the one trained on $\mathcal{D}_{T_1 \oplus T_2}$, which shows that while it can generalize invariance to either transformations from T_1 or from T_2 this does not lead to invariance to the concatenations of transformations from both of these sets.

4.3 What is the relationship between different invariances?

Finally, we are also interested in understanding whether models have inductive biases towards acquiring invariances to certain transformations and what the relationship between invariances are, *i.e.* whether learning an invariance to one type of transformation automatically leads to invariance for another one.

We pre-train models on datasets with 30 object prototypes that are each modified by transformations of a single category. Then, to understand how invariance to different types of transformations is related, we fine-tune and evaluate each model on the datasets corresponding to every other type of transformations.



(a) Transfer performance of invariances between different classes

(b) Representation distance ratios for different models

Figure 6: **[Relationships between different transformations]** a) Models pretrained on single transformation datasets perform well on most other single transformation datasets, with the exception of movement, rotation and color jitter transformations, indicating that these transformations are harder to transfer to. b) The ratios of l_2-distances between penultimate layer representations of the same classes and between representations of objects from different classes correspond to the transfer difficulty of different transformations. Scale transformations are easy to transfer to and also consistently have a high l_2-ratio, whereas l_2-ratios for movement and color jitter transformations, which are harder to generalize to, are only high for models trained for these transformations.

Results are shown in Figure 6a. We find that in general, all models transfer quite well to most of the transformations. There are three prominent outliers, however, to which other types of invariances do not seem to transfer well — color jitter, movement/translation and rotation — indicating that transferring to certain transformations is inherently harder than to others. Interestingly, we observe that within these "difficult" transformations, movement and rotation are qualitatively similar and transfer better to each other, but do not transfer as well to color jitter, and vice versa. The difficulty of acquiring movement invariance is also discussed in recent work (Zhang, 2019; Azulay & Weiss, 2019) which shows that, somewhat counterintuitively, the representations of many CNN architectures are often not shift-invariant. Note that, in contrast to the previous results, the transfer performances here are high in general because there is only one transformation per dataset acting on the object prototypes instead of a concatenation of multiple transformations.

To understand how learning invariance to these transformations helps the models achieve higher accuracies and why we see differences in transformation difficulty, we investigate the models' penultimate layer representations. To do this, for each transformation dataset and model, we compute pairwise l_2-distances between penultimate layer representations of samples from the same class (*i.e.* between inputs with the same prototype object that differ via the respective transformation and have different backgrounds) and between representations of samples from different classes. Then, we compute the between-within class ratios that show how much larger the l_2-distances between samples of different classes are compared to samples from the same class [1]. Intuitively, a high l_2-ratio means that the distances between the representations of different classes are higher than the distances between representations of the same class, which makes it easier to linearly separate them.

Figure 6b shows the results of this analysis for the movement, scale and color jitter transformations for the respective models as well as an untrained model and a model trained to classify objects without any transformations applied to them ("none"). The scale transformation is one that all models consistently transfer well to and correspondingly, its l_2-ratio is quite high for every model. The movement and color jitter transformations on the other hand are harder to transfer to and conse-

[1] We also tried using representation similarity metrics, in particular CKA (Kornblith et al., 2019), however, it is only meant to be applied for different representations of the same input, not representations of different inputs and therefore produces very low values for this analysis.

quently, their l_2 ratios are only high for the models specifically trained for them. In general, the value of the l_2-ratios corresponds well to the transfer performance of a model to the respective task.

5 Related Work

A number of prior studies have investigated the relationship between invariance and generalization performance (Azulay & Weiss, 2018; Zhou et al., 2022; Ortiz-Jimenez et al., 2020; Lyle et al., 2020). They differ from our work in that they only use partially synthetic data in the form of transformations being applied to examples from real-world datasets, which limits the breath and precision of control that is possible over the transformations in the data and therefore makes it difficult to cleanly evaluate invariance to them. For instance, both Azulay & Weiss (2018) and Zhou et al. (2022) find that invariance does not generalize well outside the training distribution, however, it might be that this lack of generalization is due to the addition of other transformations, which as we show has a significant impact, and which is hard to assess with non-synthetic data where not all degrees of variation can be controlled for.

There have been a few proposals for fully synthetic datasets that allow for a more careful study of the properties of representations (Hermann & Lampinen, 2020; Matthey et al., 2017), but they are aimed at understanding other properties such as equivariance and disentanglement.

There also has been a body of work proposing measures for the amount of invariance in the representations of DNNs (Goodfellow et al., 2009; Fawzi & Frossard, 2015; Gopinath et al., 2019; Nanda et al., 2022). This work, however, is aimed at determining whether models trained on real-world data exhibit certain invariances, whereas we create models models that have specific invariances by design by using synthetic data.

Invariance has been widely studied in the robustness literature (Szegedy et al., 2013; Papernot et al., 2016; Recht et al., 2019; Geirhos et al., 2018; Hendrycks & Dietterich, 2019; Taori et al., 2020). There, the focus is on the degradation effects that specific types of transformations such as adversarial changes to or corruptions of real-world data have on model performance, whereas we are interested in invariance as a more fundamental property of representations. It is worth pointing out though, that the findings made in the robustness literature align well with our observations in the sense that changes in a model's input data to which its representations are not invariant can severely impact its performance.

Finally, data transformations and — to a lesser extend — invariances have been studied as tools to improve model performance via data augmentation (Perez & Wang, 2017; Shorten & Khoshgoftaar, 2019) or to enable training models in a self-supervised manner (Chen et al., 2020; Grill et al., 2020). However, this work is not so much interested in using transformations to specifically achieve certain representational invariances or in understanding their impact on representations, but rather leverages them as a tool for creating representations that achieve high downstream performance.

6 Conclusion

We study the importance of invariance in representation learning by using synthetic data that allows us to precisely control differences and similarities between input points. By leveraging this method, we are able to show that invariance is critically important for downstream performance, often more so than equivariance, because both missing required invariances as well as possessing harmful invariances significantly decreases the usefulness of representations. We further investigate factors that play a role in determining which feature transformations a model becomes invariant to and find that the relevance and availability of features plays an important role. Additionally, we show that models largely transfer invariance between classes and that they possess inductive biases that influence the acquisition of certain invariances. In summary, our work reveals the often overlooked importance of invariance in representation learning and highlights to need to take invariance into account when studying transfer and generalization of representations.

REFERENCES

Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *arXiv preprint arXiv:1805.12177*, 2018.

Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *Journal of Machine Learning Research*, 20(184):1–25, 2019. URL http://jmlr.org/papers/v20/19-519.html.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

Josip Djolonga, Jessica Yung, Michael Tschannen, Rob Romijnders, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Matthias Minderer, Alexander D'Amour, Dan Moldovan, Sylvain Gelly, Neil Houlsby, Xiaohua Zhai, and Mario Lucic. On robustness and transferability of convolutional neural networks. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16453–16463, 2021. doi: 10.1109/CVPR46437.2021.01619.

Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Brandon Tran, and Aleksander Madry. Adversarial robustness as a prior for learned representations, 2019.

Alhussein Fawzi and Pascal Frossard. Manitest: Are classifiers really invariant? *CoRR*, abs/1507.06535, 2015. URL http://arxiv.org/abs/1507.06535.

Robert Geirhos, Carlos R. M. Temme, Jonas Rauber, Heiko H. Schütt, Matthias Bethge, and Felix A. Wichmann. Generalisation in humans and deep neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/0937fb5864ed06ffb59ae5f9b5ed67a9-Paper.pdf.

Ian J. Goodfellow, Quoc V. Le, Andrew M. Saxe, Honglak Lee, and Andrew Y. Ng. Measuring invariances in deep networks. In *Proceedings of the 22nd International Conference on Neural Information Processing Systems*, NIPS'09, pp. 646–654, Red Hook, NY, USA, 2009. Curran Associates Inc. ISBN 9781615679119.

Divya Gopinath, Hayes Converse, Corina S. Pasareanu, and Ankur Taly. Property inference for deep neural networks, 2019. URL https://arxiv.org/abs/1904.13215.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJz6tiCqYm.

Katherine Hermann and Andrew Lampinen. What shapes feature representations? exploring datasets, architectures, and training. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9995–10006. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/71e9c6620d381d60196ebe694840aaaa-Paper.pdf.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pp. 2668–2677. PMLR, 2018.

Minseon Kim, Jihoon Tack, and Sung Ju Hwang. Adversarial self-supervised contrastive learning. *arXiv preprint arXiv:2006.07589*, 2020.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pp. 5637–5664. PMLR, 2021.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/kornblith19a.html.

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

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems*, pp. 4768–4777, 2017.

Clare Lyle, Mark van der Wilk, Marta Kwiatkowska, Yarin Gal, and Benjamin Bloem-Reddy. On the benefits of invariance in neural networks, 2020. URL https://arxiv.org/abs/2005.00178.

Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

Vedant Nanda, Till Speicher, Camilla Kolling, John P. Dickerson, Krishna P. Gummadi, and Adrian Weller. Measuring representational robustness of neural networks through shared invariances. In *ICML*, 2022.

Guillermo Ortiz-Jimenez, Apostolos Modas, Seyed-Mohsen Moosavi, and Pascal Frossard. Hold me tight! influence of discriminative features on deep network boundaries. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 2935–2946. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1ea97de85eb634d580161c603422437f-Paper.pdf.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pp. 372–387. IEEE, 2016.

Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning, 2017. URL https://arxiv.org/abs/1712.04621.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pp. 5389–5400. PMLR, 2019.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, 2019. doi: 10.1186/s40537-019-0197-0. URL https://doi.org/10.1186/s40537-019-0197-0.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. In *Advances in Neural Information Processing Systems*, 2020. URL https://proceedings.neurips.cc/paper/2020/file/d8330f857a17c53d217014ee776bfd50-Paper.pdf.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks, 2013.

Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, Lucas Beyer, Olivier Bachem, Michael Tschannen, Marcin Michalski, Olivier Bousquet, Sylvain Gelly, and Neil Houlsby. The visual task adaptation benchmark, 2020. URL https://openreview.net/forum?id=BJena3VtwS.

Richard Zhang. Making convolutional networks shift-invariant again. In *International conference on machine learning*, pp. 7324–7334. PMLR, 2019.

Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (eds.), *Computer Vision – ECCV 2016*, pp. 649–666, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46487-9.

Allan Zhou, Fahim Tajwar, Alexander Robey, Tom Knowles, George J. Pappas, Hamed Hassani, and Chelsea Finn. Do deep networks transfer invariances across classes? In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Fn7i_r5rR0q.

A SYNTHETIC DATASETS

We create the Transforms-2D dataset based on the foreground and background images provided by the SI-score dataset (Djolonga et al., 2021). There are 61 foreground categories containing images of objects with an alpha-mask from which we randomly sample one of the images to use it as class-prototype. Additionally, there are 882 background images which we randomly sample for each input from the dataset. We resize all images to a size of 32x32 pixels and use 50,000 training, 10,000 validation and 10,000 test images, to resemble the properties of the CIFAR-10 dataset.

The foreground objects are transformed by transformations from 8 different categories:

- Blurring
- Color jitter
- Horizontal flips
- Vertical flips
- Translations/movement
- Rotations
- Scaling
- Sharpening

Transformations are implemented via the standard PyTorch data augmentations.

A.1 ADDITIONAL INFORMATION ABOUT THE IRRELEVANT FEATURES DATASET

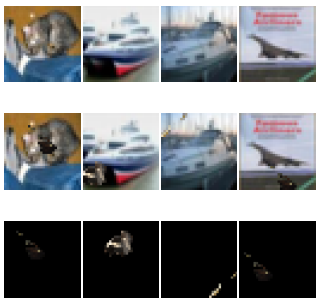


Figure 7: **[Examples of test images in the irrelevant feature analysis]**. From top to bottom: CIFAR-10 only ($X = C$), CIFAR-10 with pasted objects ($X = C + O$), objects only ($X = O$)

Figure 7 shows example images for the augmented CIFAR-10 images that we use in the analysis in section 3.2 to investigate the effects of irrelevant transformations on learned invariances.

B ADDITIONAL DETAILS ON THE EVALUATION SETUP

In the paper, we evaluate models based on the ResNet-18, VGG-11 and DenseNet-121 architectures. We use PyTorch and architectures adapted for the CIFAR-10 dataset that can be found here: https://github.com/kuangliu/pytorch-cifar

We train models on the synthetic datasets for 50 training epochs and on the augmented CIFAR-10 datasets for 200 training epochs. Fine-tuning of the last linear layer is done for 25 epochs. Model parameters are optimized using Adam.

C ADDITIONAL RESULTS ON THE IMPORTANCE OF INVARIANCE FOR DOWNSTREAM PERFORMANCE

C.1 LEARNING MORE THAN THE NECESSARY INVARIANCES DOES NOT HURT PERFORMANCE

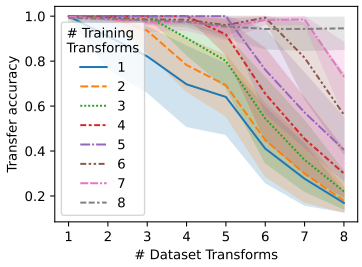


Figure 8: **Impact of varying the number of training transformations on transfer performance** Models are evaluated on datasets with subsets or supersets of their training transformations. Models trained on a superset of transformations of the target task perform well on it, whereas the performance of models that only saw a subset of the target transformations deteriorates significantly.

We have shown that models need to acquire at least the invariances corresponding to irrelevant transformations in the target domain in order to perform well on it. A natural question is whether acquiring more than the necessary invariances is detrimental for performance.

To investigate this question, we create four datasets $\mathcal{D}_1, \ldots, \mathcal{D}_n$ with the same set of randomly sampled objects O and sets of randomly sampled, concatenated transformations $T_1, \ldots, T_n$ with $|T_i| = i$ such that $T_i \subset T_{i+1}$. We train models $g_1, \ldots, g_n$ on $\mathcal{D}_1, \ldots, \mathcal{D}_n$ respectively, then fine-tune each models' last linear layer and evaluate it on each of the datasets, with $n = 8$.

Figure 8 shows the result of this analysis. Models trained on data with the same set or a superset of transformations as the target dataset consistently achieve 100% accuracy The model trained on all 8 transformations for instance achieves close to 100% accuracy on all datasets. However, models trained with only a subset of the transformations show considerably lower performance that decreases the smaller the subset of training transformations is compared to the target task. This shows that being invariant to more than the necessary transformations does not negatively impact performance, but if any required invariances are missing in the representations, a models' performance quickly takes a hit.