Can Functional Transfer Methods Capture Simple Inductive Biases?

Arne F. Nix[1,2,†] **Suhas Shrinivasan**[2,3] **Edgar Y. Walker**[1,4,5] **Fabian H. Sinz**[1,2,‡]
[1] Institute for Bioinformatics and Medical Informatics, University of Tübingen
[2] Department of Machine Learning, Institute of Computer Science, University of Göttingen
[3] Graduate Center for Neurosciences, Biophysics, and Molecular Biosciences, University of Göttingen
[4] Department of Physiology and Biophysics, University of Washington
[5] UW Computational Neuroscience Center, University of Washington
[†] `arne-fabian.nix@uni-tuebingen.de` [‡] `sinz@cs.uni-goettingen.de`

Abstract

Transferring knowledge embedded in trained neural networks is a core problem in areas like model compression and continual learning. Among knowledge transfer approaches, functional transfer methods such as knowledge distillation and representational distance learning are particularly promising, since they allow for transferring knowledge across different architectures and tasks. Considering various characteristics of networks that are desirable to transfer, equivariance is a notable property that enables a network to capture valuable relationships in the data. We assess existing functional transfer methods on their ability to transfer equivariance and empirically show that they fail to even transfer shift equivariance, one of the simplest equivariances. Further theoretical analysis demonstrates that representational similarity methods, in fact, cannot guarantee the transfer of the intended equivariance. Motivated by these findings, we develop a novel transfer method that learns an equivariance model from a given teacher network and encourages the student network to acquire the same equivariance, via regularization. Experiments show that our method successfully transfers equivariance even in cases where highly restrictive methods, such as directly matching student and teacher representations, fail.[1]

[1]Code: `https://github.com/sinzlab/orbit_transfer`

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151. Copyright 2022 by the author(s).

1 INTRODUCTION

With the rise of large and high-capacity models being trained on unprecedented amounts of data (Riquelme et al., 2021; Kolesnikov et al., 2020), the paradigm of transfer learning has grown in prominence (Zhuang et al., 2021). In transfer learning, the goal is to transfer useful functional properties learned by a *teacher*-model to a *student*-model. This is often realized by copying the teacher's parameters, or a subset thereof, to the student (weights-based transfer methods). However, copying the teacher's parameters is not feasible when the model architectures of student and teacher are different, for instance when the student has substantially fewer parameters (for the sake of efficiency) than the teacher (Hinton and Dean, 2015). Furthermore, not all of the teacher's parameters may be useful for the student. In such cases, *functional transfer* methods present an alternative to weights-based methods. Functional transfer methods rely on comparing and transferring functional properties, i.e., layer activation or outputs, for a given input. Transfer methods of this nature have applications in numerous areas like model compression (Bucilă et al., 2006; Hinton and Dean, 2015), continual learning (Pan et al., 2020; Titsias et al., 2019; Benjamin et al., 2019) or even neuroscience (Li et al., 2019).

Functional transfer methods exist in many variations. Some rely entirely on the network's final output, while others use activity within the network. Some methods require that the tasks shared by the teacher and the student be identical, while others only require that the network's hierarchical structure be somewhat relatable (McClure and Kriegeskorte, 2016). Regardless of differences, the goal remains the same—to transfer useful knowledge from teacher to student. This prompts us to question: *How effectively do existing functional transfer methods transfer useful knowledge from a teacher to a student?* Despite the significance of this question, thus far there hardly exists any definite answer regarding functional transfer methods. For instance Abnar et al. (2020) investigate whether during knowledge distillation (a functional transfer method) the teacher's inductive biases are also reflected in its logit outputs, but not specifically whether the inductive biases are transferred to the student. One reason for the lack of answers may be that the kind of knowledge that is useful to transfer is hard to characterize. It could be knowledge that the teacher learned from its training data, such as feature extraction capabilities resultant from large-scale pre-training (Beyer et al., 2021). It could also be knowledge that is inherent to the teacher's network architecture, such as the inductive bias of shift equivariance in convolutional neural networks, consequently used to improve a student network that does not have this knowledge built-in (Touvron et al., 2020).

As this could entail a very wide range of properties that are potentially useful to transfer, we choose to focus our study on one specific family of properties that are mathematically characterized and have a strong impact on the generalization of a model – equivariance. We believe transferring equivariance to be the minimal ability any useful functional transfer method ought to possess. We will thus investigate the following question: *Can functional transfer methods effectively transfer equivariance properties between student and teacher?* The answer to this question turns out to be that it is surprisingly difficult to transfer equivariance properties, and that existing functional transfer methods fail to do so. Abnar et al. (2020) investigate transfer of equivariances empirically and found that knowledge distillation improved shift and scale invariance of the student, but as we show, that is only guaranteed under strong assumptions (as we discuss in 3), and other methods do not focus on transferring equivariance itself. Creager et al. (2020) develop a framework for domain invariant learning, i.e., encouraging a model to be invariant to environment and background changes. Zhou et al. (2020) show that via meta-learning, one can recover convolutional architectures from the data itself, which they achieve by learning parameter sharing patterns, as opposed to functional transfer methods.

To approach our question, we first discuss output-level functional transfer methods (Section 3) such as knowledge distillation, as they represent one of the most popular transfer methods. Although they are theoretically capable of transferring equivariance for the entire network, we show empirically that they do not deliver on that promise in practice, if, for instance, the student is very flexible. Based on this, we hypothesize that additional within-network restriction is necessary for the transfer to be successful. This leads us to investigate representation-level transfer methods (Section 4), i.e. functional transfer between different layers inside the network. Here we show empirically that except for one-to-one matching of hidden representations, none of the methods are capable of even transferring shift equivariance. A theoretical exploration of this observation reveals that many representational similarity methods, a subclass of representational transfer methods, are not restrictive enough to guarantee a transfer of the same equivariance that was present in the teacher.

For each method, we first introduce the method followed by empirical and theoretical analysis of equivariance transfer, before moving on to the next method. Finally, we introduce a novel method of functional transfer to enable equivariance transfer (Section 5). Our method successfully captures equivariance properties of the teacher and transfers it to the student.

2 PRELIMINARIES AND SETUP

The no-free-lunch theorem (Shalev-Shwartz and Ben-David, 2013) shows that models need to be constrained in some way to have a chance of good generalization. A constraint like that is called an *inductive bias.* Our goal is to develop a method of functional transfer that guarantees transfer of knowledge—specifically transfer of useful inductive biases—between two neural networks. Here, we focus on equivariance properties, which, as alluded to earlier, constitute fundamentally important and useful knowledge to transfer.

2.1 Equivariance

Many tasks contain useful *symmetries* in their data that can be exploited by models to improve generalization. One famous example that successfully leverages symmetries are convolutional neural networks (CNNs, Fukushima, 1980). CNNs encode the shift symmetries of natural images in their architecture. They do this by being *equivariant* to shifts, which means that a shift in the input to a CNN(-layer) will result in a related shift in its output. Other examples of equivariance in models can be found in many areas from natural language processing (Gordon et al., 2020) to molecular biology (Thiede et al., 2020).

To formalize equivariances, we extend the notation of Cohen (2021) to our setting of transfer learning. Thus we will be regarding equivariance in the context of symmetry groups. A simple example would be the set $\mathbb{Z}^2$ of integer shifts in a 2D pixel grid underlying an image. E.g. a group element $(m, n) \in \mathbb{Z}^2$ corresponds to a shift of all pixels by m positions along the vertical and n positions along the horizontal axis. With addition $(m, n) + (p, q) = (m + p, n + q)$ as the group operation, negative shifts $(m, n) + (-m, -n) = (0, 0)$ as the inverse for each group element and $(0, 0)$ as the identity element, we can show that $\mathbb{Z}^2$ is indeed a group.

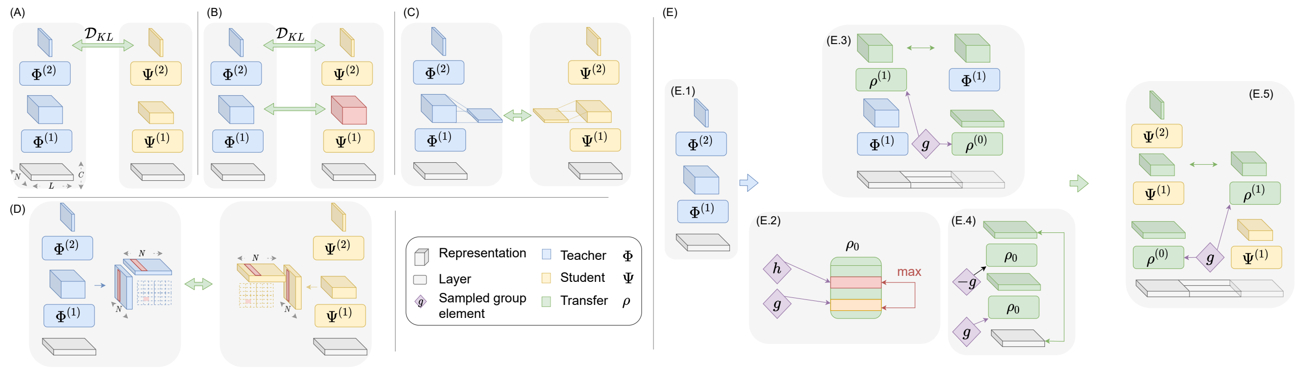

Figure 1: Overview of the transfer methods discussed in this paper: Knowledge distillation (A, Section 3), Direct matching (B, Section 4), Attention transfer (C, Section 4.1), RDL (D, Section 4) and Orbit transfer (E, Section 5).

We define a neural network as a composition of functions $\Phi = \Phi^{(L)} \circ ... \circ \Phi^{(1)}$ with layers $\Phi^{(l)} : \mathcal{X}^{(l-1)} \rightarrow \mathcal{X}^{(l)}$. Each individual layer $\Phi^{(l)}$ acts on an input or feature space $\mathcal{X}^{(l-1)}$ and projects to a feature space $\mathcal{X}^{(l)}$ that serves as input for the next layer. We further focus on symmetries generated by a group G. A group element $g \in G$ acts on an input element x via a *linear representation* ρ_g. Wherever it is clear from the context, we will leave the dependency on g implicit to simplify notation. If a function—or layer $\Phi^{(l)}$—is G-equivariant then its output will have a corresponding action $\rho^{(l)}$, that leads to the same results as transforming the input:

$$\rho_g^{(l)}\Phi^{(l)}(x) = \Phi^{(l)}(\rho_g^{(l-1)}x) \qquad (1)$$

for all $x \in \mathcal{X}_t^{(l-1)}$ and any group element $g \in G$.

Known equivariance properties are useful as they allow parameter sharing in the network (Cohen, 2021). However, it is generally hard to discover these symmetries from data alone, since it may require a lot of data. Hence it is beneficial if we could transfer them from a extensively trained teacher network to a student. To formalize the transfer, we refer to the teacher- and student-network as Φ and Ψ, respectively. The corresponding layers are then defined by $\Phi^{(l)} : \mathcal{X}_t^{(l-1)} \rightarrow \mathcal{X}_t^{(l)}$ and $\Psi^{(l)} : \mathcal{X}_s^{(l-1)} \rightarrow \mathcal{X}_l^{(s)}$. We assume that the student network differs in architecture from its teacher[2] and is not necessarily G-equivariant by default, but that it is expressive enough to learn equivariance with the right guidance. This means generally that the student has more capacity, or is less constrained. In other words the student has less inductive bias than the teacher model. This is important since the aim of the experiments is to see if functional transfer would be strong enough to constrain, or transfer inductive bias, to an over-parameterized model that inherently lacks it. In practice, we achieve this by combining the standard cross-entropy loss $\mathcal{L}_{\text{CE}}$ with a transfer term Ω. Therefore the final loss will be: $\mathcal{L} = (1-\gamma)\mathcal{L}_{\text{CE}} + \gamma\Omega$. The two components are combined by an interpolation weight $\gamma \in [0, 1]$ which is treated as a hyperparameter.

[2]We assume the number of layers is identical to simplify the derivations. Since layers are not necessarily expected to be linear, we can always get to the same number of layers by adding identity layers or by summarizing multiple layers in one.

2.2 Analysis Setup

An important aspect of both the empirical and analytical settings we address is the extent to which transformed group elements are present in the training data. Formally, the set of data points $\{\rho_g x : g \in G\}$ that can be reached from a single data point x is called *orbit*. As we will see later, the extent to which entire orbits are in the training data will affect the generalization performance of the student. For our theoretical analysis we generally consider the case of unlimited training data—in other words data that contains all orbit elements of the transformation we aim to transfer.

To empirically investigate the transfer abilities of the various methods we analyze in this work, we construct a simple experimental setup on the MNIST-1D (Greydanus, 2020) task. This task generates 4000 training and 1000 test inputs, which are 40-dimensional vectors that are procedurally generated based on templates inspired by the original MNIST digits. The synthetic generation process allows fine-grained control over various properties of the resulting samples. For our purposes importantly, we can control the range of orbits range present in the training data. This is done through a "shift limit" s that partitions the set of shifts for which we generate an MNIST-1D variant with random shifts into two sets: *seen shifts* $[0, s]$ and *unseen shifts* $(s, 39]$. Our training data consists of *seen shifts*. We can then test the resulting model on three different ranges of shift values applied to the test data: *seen shifts* $[0, s]$, *unseen shifts* $[s, 39]$, and *all shifts* $[0, 39]$. This setup allows us to verify whether a network has learned shift equivariance and generalizes well to unseen shifts. Since we aim to create a very simple evaluation task, we chose teacher and student networks such that their hidden layer outputs have the same size. Both have two hidden layers and a spatial max-pooling operation after the last layer to obtain the class predictions. The two networks only differ in the nature of the first two layers, where the teacher network uses convolutions, the student network instead uses fully connected layers (more flexible and less inductive bias). More details on the architecture and training setup can be found in the supplementary material. To fully explore the abilities of each transfer method, we first perform a hyper-parameter search for each on the setting with $s = 30$. Since we are mainly interested in generalization to unseen shifts, not images, we decided to select the best performing set of hyper-parameters on the image test set with seen shifts. The selected model is then trained and evaluated with $s = 0, 10, 20, 30, 40$ to explore the transfer abilities under different data conditions. As we introduce more shifts in the training data, the *seen shifts* test set will also contain those shifts. This makes it increasingly difficult to generalize for models without shift equivariance. As expected, the student network trained without any transfer knowledge degrades on seen shifts as training progresses (see Figure 2). Analogously, the performance on unseen shifts stays consistently bad until $s = 40$ where all shifts are seen.

3 OUTPUT-LEVEL TRANSFER

The most well-known methods for functional transfer fall in the category of output-level transfer methods. These methods aim to transfer the entire network function Φ by matching the function values directly at the final layer, i.e. $\Phi(x) = \Psi(x) \, \forall x$ in the case of infinite data. For classification, this is known as *knowledge distillation (KD)* (Hinton and Dean, 2015) which transfers Φ by minimizing the Kullback–Leibler divergence between the output distributions given by the softmax output of teacher and student network respectively (see Figure 1.A). Thus the transfer regularization would be $\Omega = p_t(x) \log \frac{p_t(x)}{p_s(x)}$ with softmax distributions $p_t(x)_i = \frac{\exp(\Phi(x)_i/\tau)}{\sum_j \exp(\Phi(x)_j/\tau)}$ and $p_s(x)_i = \frac{\exp(\Psi(x)_i/\tau)}{\sum_j \exp(\Psi(x)_j/\tau)}$. The corresponding transfer method for regression tasks would be *functional distance regularization* (Benjamin et al., 2019), which minimizes the euclidean distance between function values of student and teacher networks.

Can KD transfer equivariance? Abnar et al. (2020) investigated this question empirically and found that KD improved shift and scale invariance of fully-connected student model when they trained it with KD from the outputs of a scale and shift invariant teacher. At first glance, these results are expected: a student that is trained with KD will obtain the teacher's equivariance properties if the training is successful, i.e. if the function is matched perfectly. However, this is only guaranteed if the optimization leads to zero loss and the training happens in the limit of infinite data. It is unclear, however what happens if this is not the case. One scenario could be that the optimization yields zero loss, i.e. the teacher function is matched perfectly, but on the limited training data which might not contain all orbits of the symmetry group. Such a solution will likely not generalize well.

Our experimental results on MNIST1D verify this intuition. KD generally will not improve over the baseline student model when evaluated on unseen shifts (see Figure 2). We hypothesize that KD, by only regularizing on the final outputs, is simply not restrictive enough for this task.

4 REPRESENTATION-LEVEL TRANSFER

As we have just seen, it is hard to transfer equivariance on the level of final outputs and we hypothesized that this due to the lack of constraints on the representations within the network. One approach for this problem would be to directly transfer on the level of hidden representations, which we study in this section.

Can direct matching of internal representations transfer equivariance? *Direct matching* of the representations of the teacher and the student networks, e.g. by minimizing the mean-squared error between the teacher and student representations (see Figure 1.B), will likely be an effective way to transfer at this level.

To confirm this experimentally, we minimize the mean-squared distance on the internal layers' outputs and add a KD objective on the final layer. This approach works as suspected. Figure 2 shows a nearly perfect performance on unseen shifts as long as a few orbit elements are contained in the training data. However, making the problem slightly harder by replacing the student's final pooling operation with a linear projection while keeping the teacher the same reveals issues similar to KD as we no longer see any improvement over the student baseline in this scenario (see Figure 3).

Although we have just shown that direct matching is successful in transferring equivariance, a direct comparison of teacher and student representations requires the hidden representations to be of the same shapes. Enforcing this on all levels of the network would mean restricting the student network architecture to conform to that of the teacher, which is contrary to the goals of functional transfer methods. We thus explore alternative solutions to representation-level transfer that can be applied without requiring the same hidden layer shapes between student and teacher.

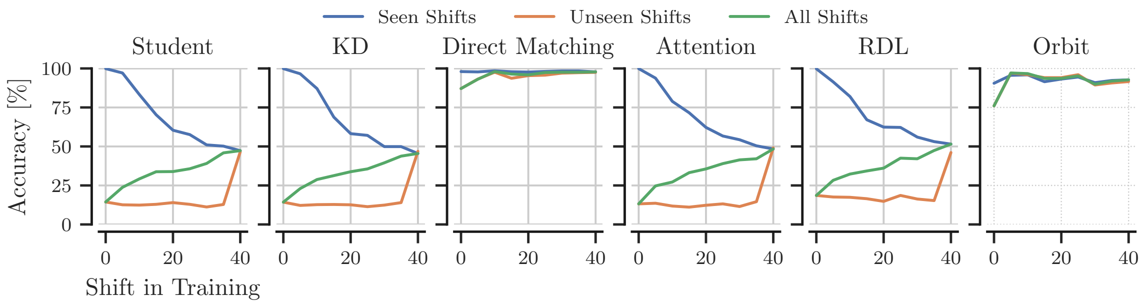

Figure 2: Test performance after transfer for a student with max pooling as final layer.

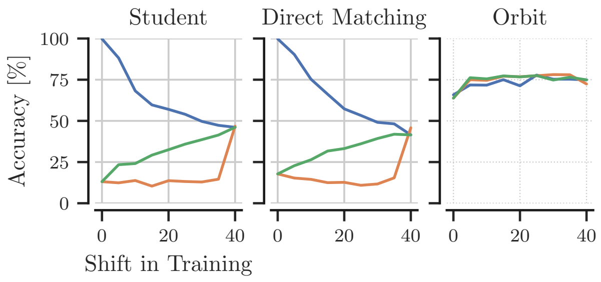

Figure 3: Test performance after transfer for a student with a linear final layer.

4.1 Attention Transfer

CNNs and related architectures generally maintain a three-dimensional structure throughout the outputs of most of their layers. This gives hidden activations the semantic interpretation of width, height and channel dimensions. *Attention transfer* (Zagoruyko and Komodakis, 2017) leverages this and the fact that layers from comparable processing steps usually have comparable spatial dimensions and only differ in the channel dimension. Slightly differing spatial dimensions can be aligned by up- or down-sampling, and the channel dimension is pooled by summation or maximum selection to extract an "attention map" that can be matched between teacher and student (see Figure 1.C).

$$\Omega^{(l)} = \| \frac{A_t^{(l)}(x)}{\|A_t^{(l)}(x)\|_2} - \frac{A_t^{(l)}(x)}{\|A_t^{(l)}(x)\|_2} \|_2^2$$

whith $A_t^{(l)}(x) = \sum_{c=1}^{C_s^{(l)}} |\Phi^{(l)}(x)_c|$ and $A_s^{(l)}$ defined analogously, channel size $C_s^{(l)}$ and $C_t^{(l)}$ for student and teacher layers respectively.

Attention transfer aligns the spatial dimensions between student and teacher. This should theoretically be beneficial when transferring equivariances to spatial effects like shift. However, our results show that that in practice this is not the case (Figure 2). Attention transfer fails to capture the shift equivariance in our experiment. The attention method has a potential drawback that could both be responsible for this outcome. Collapsing the channel dimension is certainly not a lossless operation and may likely hide information important for equivariance.

4.2 Representational Similarity Transfer

The problem of comparing representations of two networks with distinct architectures is of wider interest outside of transfer learning. Unsurprisingly, there exists a broad range of methods designed to compare neural network representations (Kornblith et al., 2019). For these methods, the general idea is to not match the two networks on the individual representations but to consider the representations for an entire batch, i.e. consider the representation as matrices $\mathbf{\Phi}^{(l)} = [\Phi^{(:l)}(x_1), \ldots, \Phi^{(:l)}(x_N)]^\top$ and $\mathbf{\Psi}^{(l)} = [\Psi^{(:l)}(x_1), \ldots, \Psi^{(:l)}(x_N)]^\top$ for a batch of inputs $x_1, \ldots, x_N \in \mathcal{X}^{(0)}$. Note that here we treat $\Phi^{(:l)}$ and $\Psi^{(:l)}$ as functions on the input features, which means that we consider the composition of all layers up to l as one function. Then a comparison can be done along the number of samples N, providing a similarity for general network representations even if $\mathcal{X}_t^{(l+1)} \neq \mathcal{X}_s^{(l+1)}$. Maximizing this similarity (or equivalently minimizing the distance between two representations) can thus be used as an objective for functional transfer.

Representational Distance Learning A method following this principle is *representational distance learning* (RDL, McClure and Kriegeskorte, 2016). Here the representational distance is measured through comparison of *representational dissimilarity matrices* (RDM). These matrices are essentially Gram matrices of the batch representations $\mathbf{\Phi}^{(l)}$ (with additional normalization). Thus each entry (n, m) in the RDM shows the dissimilarity of samples x_n and x_m w.r.t. representation $\Phi^{(:l)}$. Such RDMs are computed for student and teacher respectively and used to compute representation distance by minimizing the Frobenius norm between the two.

$$\Omega^{(l)} = \frac{1}{N^2} \| \mathbf{\Phi}^{(l)} \mathbf{\Phi}^{(l)\top} - \mathbf{\Psi}^{(l)} \mathbf{\Psi}^{(l)\top} \|_F^2$$

This approach could be used to enforce representations similar to a teacher on any layer of a neural network without restricting either network in any way. To ease notation, we will drop the layer index l for the remainder of this section.

However, this method does not match the individual responses directly, which raises questions as to how powerful and accurate the transfer will be. For RDL, an optimal solution has to fulfill $\mathbf{\Psi}\mathbf{\Psi}^\top = \mathbf{\Phi}\mathbf{\Phi}^\top$. However, $\mathbf{\Psi}\mathbf{\Psi}^\top = \mathbf{\Phi}\mathbf{\Phi}^\top$ if and only if $\mathbf{\Psi} = Q\mathbf{\Phi}$ with $Q \in SO(N)$ (Co, 2013, theorem 7.3.11). This means that RDL matches the teacher representation up to orthogonal transformations. In the following, we will show that this is not enough to guarantee the transfer of equivariance properties.

Alternative methods that follow the same principle as RDL to quantify representational similarity are summarized and compared in Kornblith et al. (2019). In principle all of them can be utilized to formulate a functional transfer objective similar to RDL. Nevertheless, most representational similarity methods have been shown to be invariant to orthogonal transformations (Kornblith et al., 2019). This means that a global optimum found by these methods will also have the property $\mathbf{\Psi} = Q\mathbf{\Phi}$ for $Q \in SO(N)$.

Can representational similarity methods transfer equivariance? Following the same methodology as before, we evaluate the transfer abilities of RDL. Similar to previous methods it fails in transferring shift equivariance (see Figure 2). In the following, we theoretically investigate why that is the case.

As we have shown above, functional transfer methods that rely on representational similarity are invariant to orthogonal transformations. That means that an optimal solution found by these methods can only restrict the student representations to match the teacher representations up to orthogonal transformation, i.e. $\mathbf{\Psi} = Q\mathbf{\Phi}$ for $Q \in SO(N)$. In the limit of infinite data, this also implies that the functions are equal up to the same orthogonal transformation, which we will denote (slight abuse of notation) as $\Psi = Q\Phi$. In this case the following theorem holds:

Lemma 1. *Given two representations Φ and Ψ with relationship $\Psi = Q\Phi$ for some orthogonal Q, the following holds: Φ is equivariant w.r.t. group representation $(\rho^{(0)}, \rho^{(1)})$ if and only if Ψ is equivariant w.r.t. group representation $(\rho^{(0)}, Q\rho^{(1)}Q^\top)$.*

Proof of Lemma 1. We first prove the forward direction, i.e. we show that if Φ is G-equivariant w.r.t. group representations $(\rho^{(0)}, \rho^{(1)})$, then Ψ is G-equivariant w.r.t. $(\rho^{(0)}, Q\rho^{(1)}Q^\top)$.

$$\begin{aligned}\Psi(\rho^{(0)}x) &= Q\Phi(\rho^{(0)}x) = Q\rho^{(1)}\Phi(x)\\ &= Q\rho^{(1)}(Q^\top Q)\Phi(x)\\ &= (Q\rho^{(1)}Q^\top)\Psi(x)\end{aligned}$$

Here we first use the fact that $\Psi = Q\Phi$, then we exploit the orthogonality of Q and the equivariance property of Φ w.r.t. $\rho^{(1)}$. Finally, we use the definition of Ψ a second time to get back from $Q\Phi$ to Ψ.

The backward direction the symmetric nature of Lemma 1 which lets us exploit the fact that $\Psi = Q\Phi$ entails $\Phi = \tilde{Q}\Psi$ where $Q \in SO(N)$ and $\tilde{Q} = Q^\top$. Then from the forward direction of Lemma 1 it follows that, Ψ is equivariant w.r.t. $(\rho^{(0)}, Q\rho^{(1)}Q^T)$ yields that Φ is equivariant w.r.t $(\rho^{(0)}, \tilde{Q}Q\rho^{(1)}Q^\top\tilde{Q}^\top) = (\rho^{(0)}, \rho^{(1)})$. □

The consequence of Lemma 1 is that representational similarity methods and other methods that can not ensure an exact matching of the teacher representations will not guarantee a transfer of equivariance properties w.r.t. the same group representation on the output. As we showed, the student representation will be equivariant w.r.t. the transformed group representation $Q\rho^{(1)}Q^\top$. Thus, RDL does transfer equivariance in the limit of infinite data, but might end up with a different "global" linear representation of the group at the final layer which might not aid generalization in the same way as the teacher representation itself. In particular the possible orthogonal transform Q will destroy the shift invariance of the last max-pooling layer since the supremum norm is not invariant under rotation. In this sense, the learned linear representation of the group does not fit to the expected input representation of the max-pooling layer which destroys its invariance property.

5 ORBIT MODEL TRANSFER

As we have seen in Section 3, output-level functional transfer from an equivariant teacher, such as performing KD, does not necessarily transfer equivariance to the student. However, we have also seen (Section 4) that trying to enforce similarity within the network, i.e. on the level of representations, is also insufficient. Such representational similarity methods are theoretically capable of enforcing an equivariance property on the student, but they cannot restrict the exact nature of that equivariance enough to guarantee a successful transfer.

These findings have revealed that matching the function of the entire network or even that of individual layers is too broad a task to reliably transfer specific equivariances. Hence we hypothesize that decoupling the equivariance property from the function is the issue.

We took this problem as inspiration and leveraged the fact that the problem definition for equivariance transfer is well-defined. For a transfer to be successful, the student has to fulfill the equation $\Psi(\rho_0 x) = \rho_1(\Psi(x))$ after training. Therefore, we propose a new approach where we directly learn the group representation ρ that the teacher is equivariant to and encourage the same equivariance in the student network.

5.1 Method

Our approach separates the transfer process into two steps. First, we learn a model of the equivariance throughout the teacher network, and then we use this model to regularize the student network.

Learning the equivariance from the teacher The idea for capturing the teacher's equivariance is to learn a model ρ that fulfills $\Phi(\rho_g^{(0)}x) = \rho_g^{(1)}\Phi(x)$ for any given g and x, i.e. a model of the group representation that Φ is equivariant to. To find such a model, we freeze the teacher network Φ and minimize the following objective for a given g and x:

$$\mathcal{L}_{\text{equiv}} = \frac{1}{H}\|\Phi(\rho_g^{(0)}x) - \rho_g^{(1)}\Phi(x)\|_2^2 \tag{2}$$

for hidden size H. This minimizes the distance between the layer representation with the group operation applied on the input or the layer's output.

One can see that the "true" equivariance representation that Φ is equivariant to would minimize this objective. However, at the same time, a trivial solution where ρ simply learns an identity operation on $\mathcal{X}$ for every g would also achieve the same result. To prevent this, we additionally minimize the absolute cosine dissimilarity between all representations, i.e. the kernels, of distinct group elements:

$$\mathcal{L}_{\text{group}} = \frac{1}{|G|\cdot(|G|-1)}\sum_{g\in G}\sum_{h\in G\setminus\{g\}} |\cos(\rho_g^{(0)}, \rho_h^{(0)})| \tag{3}$$

Finally, we want to encourage a group structure in ρ and thus we add the following objective to encourage ρ to be invertible:

$$\mathcal{L}_{\text{inv}} = \|\rho_{-g}^{(0)}\rho_g^{(0)}x - x\|_2^2 \tag{4}$$

The parameters of the orbit model ρ are optimized to minimize a sum of all three objectives and the standard cross-entropy loss $\mathcal{L}_{\text{CE}}$ on the transformed inputs:

$$\min_{\rho} \gamma_{\text{CE}}\mathcal{L}_{\text{CE}} + \gamma_{\text{equiv}}\mathcal{L}_{\text{eqiv}} + \gamma_{\text{group}}\mathcal{L}_{\text{group}} + \gamma_{\text{inv}}\mathcal{L}_{\text{inv}} \tag{5}$$

with weights $\gamma_{\text{CE}}, \gamma_{\text{equiv}}, \gamma_{\text{group}}, \gamma_{\text{inv}} \in \mathbb{R}^+$.

Modeling group representations through convolutions So far, we have described an objective function to extract the equivariance from a given teacher network. This method is generally agnostic to the choice of group representation model ρ, nevertheless this decision is crucial for the effectiveness of the transfer.

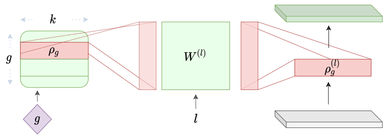

Figure 4: The group representation model first selects a filter of size k based on the group element g, then it applies the linear transformation $W^{(l)}$ to finally get the group representation that can be applied on the input x to get $\rho_g^{(l)}x$.

The group representation model needs to be powerful enough to capture the equivariance as well as the change of group representation throughout the depth of the network. At the same time, we need a model that is flexible enough to be applied on all layers of both teacher and student. To allow this flexibility, we decided to use convolution to model the group operation on every layer. Thus, for any given layer l and group element g, our model needs to provide a convolution filter that can be applied to the input (after padding it to preserve the size). A naive implementation would learn a separate filter for each group element and layer, which would not only require a lot of parameters, but also ignores the connectedness that group representations of the same g can have throughout an equivariant network. We decided to leverage this quality by factorizing the parameterization. Our model therefore learns one filter of size k per group element as well as L linear projections of size $k \times k$. To obtain the group operation $\rho_g^{(l)}$ the lth linear projections $W^{(l)}$ is applied to the filter corresponding to group element g. The group operation can then be applied to the input to get its group representation $\rho_g^{(l)}x$. The entire process is illustrated in Figure 4.

Modeling group representations as affine transformation on the coordinate space To demonstrate the flexibility of our transfer learning framework, we propose an alternative, more general, way of modeling group representations. For this, we parameterize a 3D transformation directly by learning an affine transformation matrix for each g and l. This $\rho_g^{(l)} \in \mathbb{R}^{3\times 4}$ determines the transformation of a 3D input feature map, akin to spatial transformer networks (Jaderberg et al., 2015). With a group representation modeled in this way, Orbit can, in principle, transfer equivariance to any affine transformation across both spatial dimensions and the channel dimension.

Training the student to have the same equivariance Once the group representation model is learned, applying it in the student training is straight-forward. We simply use the same objective that we used for training the group representation model on the teacher, but here we freeze the group representation model instead.

	CNN → MLP		ResNet18 → ViT	
Method	Centered	Translated	Centered	Translated
Teacher	99.0	93.4	99.6	90.3
Student	98.5	35.7	98.6	37.3
+ Augment	54.3	97.0	56.5	97.4
KD	**98.8**	41.1	98.8	41.2
Attention	98.4	31.9	—	—
RDL	98.6	31.9	**99.3**	59.6
Orbit	**98.8**	**95.2**	98.4	**84.0**

Table 1: MNIST (column 1 and 3) and MNIST-C (column 2 and 4) test results for four different transfer methods. Left two columns show the transfer results from a small CNN teacher to an MLP student. The right columns show analogous experiments between a ResNet18 teacher and a small ViT student. The best performing transfer is shown in bold for each column.

This means can ignore the objectives from Equation 3 and 4, which leaves us with:

$$\Omega^{(l)} = \|\Psi(\rho_g^{(0)}x) - \rho_g^{(1)}(\Psi(x))\|_2^2 \tag{6}$$

Additionally, we can also use our model to sample data $\rho_g^{(0)}$ which we can use as data augmentation when computing the standard cross-entropy loss. One potential caveat to our method is the fixed filter-size that limits the range of operations we can learn. For instance, a $5\times$ 5 filter can only learn shifts of length two in all directions. This can be solved by iteratively applying the same filter for a random number of repeats when computing the objectives in Equation 2 and 6.

5.2 Experiments

First, we evaluate the generalization abilities to unseen shifts for our student model after transfer. In contrast to most methods presented above, our approach manages to outperform the student baseline after transfer (see Figure 2). This is the case not only for unseen shifts, but also for seen shifts, as the transferred equivariance helps with the increasingly more difficult test set as we increase the shifts in training. Additionally we also see that orbit transfer helps with a student architecture that replaces the pooling with a linear layer before the network's output (see Figure 3). This is especially noteworthy since direct matching failed to perform well in this scenario.

After exploring the different transfer methods in a controlled environment, we finally verify our results on a slightly more realistic task, namely MNIST (Deng, 2012). We follow the example of Abnar et al. (2020) and use a CNN trained on standard MNIST as the teacher for a student network without a built-in inductive bias for shift equivariance. In the first, focused, setting, we use simple three layer networks for both student and teacher, whereas the teacher consists of convolutional layers with maximum pooling and the student is a purely fully-connected network. The second, more realistic setting transfers from a ResNet-18 (He et al., 2015) to a small six layer vision transformer (ViT; Dosovitskiy et al., 2020). More details on the architecture and the training procedure can be found in the supplementary material. After training, we evaluate the trained model on both the standard – *centered* – MNIST test set, as well as the *translated* version provided by MNIST-C (Mu and Gilmer, 2019). In Table 1 we report the results for hyper-parameters that were selected on the translated test set in the *CNN → MLP* setting.

The results confirm the findings from the MNIST-1D experiments we report above. We again see that transferring shift equivariance to a fully connected network is a hard task for conventional functional transfer methods. Attention transfer (-3.8%) and RDL (-3.8%) both underperform compared to the student's baseline performance in the *CNN → MLP* setting. Even KD, which was reported by Abnar et al. (2020) to improve performance on the very same evaluation set, only achieves marginal improvements (+5.4%) in our setting, that focuses on shift equivariance exclusively. Our approach of Orbit transfer shows a strong improvement over the baseline performance of the student network (+59.5%). We observe similar results in the *ResNet18 → ViT* setting. KD only slightly improves shift performance (+3.9%) and Orbit shows major gains on the translated test set (+46.7%). Interestingly, RDL performs significantly better in this setting, both on the translated (+22.3%) as well as the centered test set (+0.7%).

5.3 Analysis

A control that trains the student with data augmentation identical to the translated test set reveals that our method almost reaches the same performance (-1.8%), even though it has never seen the translated set before test time. Following the same procedure of Mu and Gilmer (2019), we trained the student network with shift augmentations that do not include the center position. Therefore this student network shows overfitting behavior on the test shifts, which leads to a large drop in centered performance (-44.2%). The same is not true for our approach, since technically there are no "seen" or "unseen" shifts in our approach.

One big advantage of our equivariance transfer method is its interpretability. In Figure 5 we inspect the kernels that are learned for each group element in our group representation model. For a perfect model of shift equivariance, we would expect each filter to have a single non-zero position and all filters to be distinct. The learned filters resemble this expectation to a some

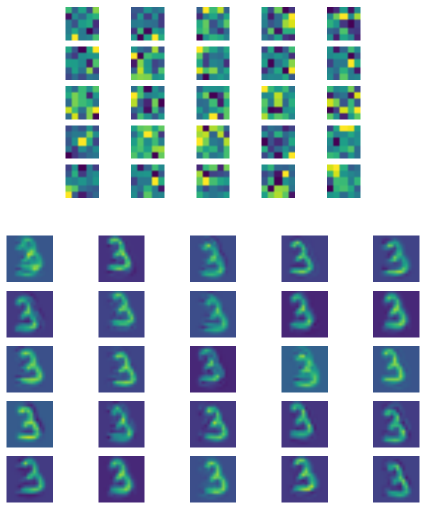
Figure 5: Rows 1-5: Kernels for all 25 linear representation of the group elements learned in the equivariance model. Rows 6-10: Kernels (in identical order) applied to an example input.

Method	Upright	Rotated
Teacher (G-CNN)	99.1	87.7
Student (MLP)	98.3	39.5
KD	98.7	46.3
RDL	**99.0**	45.6
Attention	98.3	41.1
Orbit	97.5	**76.5**

Table 2: Results on the MNIST test set with images in upright (column 1) or randomly rotated (column 2) orientation for four different transfer method for a G-CNN teacher and an MLP student.

extend. In cases where the filters are somewhat ambiguous, or where we do not know how the filters are expected to look like, we can even look at an example input after transforming it with out learned transformation network. In our case it becomes clear that the model did learn shifts, as we clearly see shifted versions of the input when applying the filters in Figure 5.

5.4 Rotation Experiments

In order to verify that our Orbit method can generalize to inductive biases other than shift equivariance, we apply it to the task of transferring equivariance to random rotations by multiples of ninety degree. Here the teacher model is a group-convolutional neural network (G-CNN; Cohen and Welling, 2016) and the goal is to transfer its inductive bias to a simple MLP. For Orbit, we model $\rho_g^{(l)}$ as an affine transformation (see above). We observe an effect similar to the shift equivariance setting (cf. Table 2), where established transfer methods show minimal effectiveness (up to +6.8% for KD) and Orbit performs remarkably well (+37.0%). However, we have to note that the corresponding optimization problem suffers from local minima, making it sensitive to initialization and – so far – preventing us from jointly transferring shift and rotation equivariance with the same model. More details on the experimental setup can be found in the supplementary material.

6 CONCLUSION

We investigated the transfer abilities of functional transfer methods and empirically showed in a simple controlled example that they are incapable of transferring even simple equivariances such as shift. We then showed for methods based on representational similarity that they cannot guarantee that the student network has the same linear representation of the equivariance as the teacher after training. Based on our insights, we developed Orbit, a novel transfer method that learns the equivariance properties of a given network and transfers them to a student network. Finally, we demonstrated that our method surpasses other methods by a large margin when transferring shift equivariance from a CNN to a fully-connected network. For future work, we are expanding these experiments to larger models and datasets, especially to more challenging symmetries such as rotations. Most importantly, our work shows promise and hopes to inspire approaching the transfer learning problem from the view of transferring useful and interpretable inductive biases.

Acknowledgements

We thank all reviewers for their constructive and thoughtful feedback. Furthermore, we thank Mohammad Bashiri for helpful comments and discussions. The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Arne Nix.

This work was supported by the Cyber Valley Research Fund (CyVy-RF-2019-01). FHS is supported by the Carl-Zeiss-Stiftung and acknowledges the support of the DFG Cluster of Excellence "Machine Learning – New Perspectives for Science", EXC 2064/1, project number 390727645. This work was supported by an AWS Machine Learning research award to FHS. This work was supported by the German Research Foundation (DFG): SFB 1233, Robust Vision: Inference Principles and Neural Mechanisms, TP 06, project number: 276693517.

References

Samira Abnar, Mostafa Dehghani, and Willem Zuidema. Transferring Inductive Biases through Knowledge Distillation. 2020. URL `https://github.com/samiraabnar/Reflect.http://arxiv.org/abs/2006.00555`.

Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

Ari S Benjamin, David Rolnick, and Konrad P Kording. Measuring and regularizing networks in function space. In *7th International Conference on Learning Representations, ICLR 2019*, 2019.

Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander Kolesnikov. Knowledge distillation: A good teacher is patient and consistent. jun 2021. URL `https://arxiv.org/abs/2106.05237v1http://arxiv.org/abs/2106.05237`.

Cristian Bucilă, Rich Caruana, and Alexandra Niculescu-Mizil. Model compression. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, volume 2006, pages 535–541, 2006. ISBN 1595933395. doi: 10.1145/1150402.1150464.

Tomas B. Co. Matrix Analysis. In *Methods of Applied Mathematics for Engineers and Scientists*, pages 99–146. Cambridge University Press, oct 2013. ISBN 9781139020411. doi: 10.1017/cbo9781139021821.005.

Taco S. Cohen. *Equivariant convolutional networks*. PhD thesis, University of Amsterdam, 2021. URL `https://dare.uva.nl`.

Taco S. Cohen and Max Welling. Group Equivariant Convolutional Networks. *33rd International Conference on Machine Learning, ICML 2016*, 6: 4375–4386, feb 2016. URL `https://arxiv.org/abs/1602.07576v3`.

Elliot Creager, Jörn-Henrik Jacobsen, and Richard Zemel. Environment Inference for Invariant Learning. 2020. URL `http://arxiv.org/abs/2010.07249`.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. 2020. URL `https://github.com/http://arxiv.org/abs/2010.11929`.

Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202, 1980. ISSN 03401200. doi: 10.1007/BF00344251.

Jonathan Gordon, David Lopez-Paz, Marco Baroni, and Diane Bouchacourt. Permutation Equivariant Models for Compositional Generalization in Language. *Iclr*, (2019):1–12, sep 2020. URL `https://github.com/facebookresearch/Permutation-Equivariant-Seq2Seqhttps://github.com/facebookresearch/Permutation-Equivariant-Seq2Seq%0Ahttps://openreview.net/forum?id=SylVNerFvr#`.

Sam Greydanus. Scaling down Deep Learning. nov 2020. URL `https://arxiv.org/abs/2011.14439v3http://arxiv.org/abs/2011.14439`.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem:770–778, dec 2015. ISSN 10636919. doi: 10.1109/CVPR.2016.90. URL `https://arxiv.org/abs/1512.03385v1`.

Geoffrey Hinton and Jeff Dean. Distilling the Knowledge in a Neural Network. Technical report, 2015.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, volume 2015-Janua, pages 2017–2025. Neural information processing systems foundation, jun 2015. URL `https://arxiv.org/abs/1506.02025v3`.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, dec 2014. URL `https://arxiv.org/abs/1412.6980v9`.

Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big Transfer (BiT): General Visual Representation Learning. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 12350 LNCS, pages 491–507. Springer Science and Business Media Deutschland GmbH, dec 2020. ISBN 9783030585570. doi: 10.1007/978-3-030-58558-7_29. URL `https://arxiv.org/abs/1912.11370v3`.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of Neural Network Representations Revisited. *36th International Conference on Machine Learning, ICML 2019*, 2019-June:6156–6175, may 2019. URL `https://arxiv.org/abs/1905.00414v4`.

Zhe Li, Wieland Brendel, Edgar Y Walker, Erick Cobos, Taliah Muhammad, Jacob Reimer, Matthias Bethge, Fabian H Sinz, Xaq Pitkow, and Andreas S Tolias. Learning from brains how to regularize machines, 2019. ISSN 23318422.

Patrick McClure and Nikolaus Kriegeskorte. Representational distance learning for deep neural networks. *Frontiers in Computational Neuroscience*, 10(DEC): 131, dec 2016. ISSN 16625188. doi: 10.3389/fncom.2016.00131. URL `http://journal.frontiersin.org/article/10.3389/fncom.2016.00131/full`.

Norman Mu and Justin Gilmer. MNIST-C: A Robustness Benchmark for Computer Vision. jun 2019. doi: 10.5281/zenodo.3237938. URL `https://arxiv.org/abs/1906.02337v1http://arxiv.org/abs/1906.02337`.

Pingbo Pan, Siddharth Swaroop, Alexander Immer, Runa Eschenhagen, Richard E Turner, and Mohammad Emtiyaz Khan. Continual deep learning by functional regularisation of memorable past. In *Advances in Neural Information Processing Systems*, volume 2020-Decem, 2020.

Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling Vision with Sparse Mixture of Experts. jun 2021. URL `https://arxiv.org/abs/2106.05974v1http://arxiv.org/abs/2106.05974`.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*, volume 9781107057. 2013. ISBN 9781107298019. doi: 10.1017/CBO9781107298019. URL `https://books.google.de/books?hl=en&lr=&id=Hf6QAwAAQBAJ&oi=fnd&pg=PR15&ots=2IqhMhjJL2&sig=jvTRcWythPVe0bK8Ettw0i7g6no`.

Erik Henning Thiede, Truong Son Hy, and Risi Kondor. The general theory of permutation equivarant neural networks and higher order graph variational encoders. 2020. URL `http://arxiv.org/abs/2004.03990`.

Michalis K Titsias, Jonathan Schwarz, Alexander G. de G. Matthews, Razvan Pascanu, and Yee Whye Teh. Functional Regularisation for Continual Learning with Gaussian Processes. 2019. URL `http://arxiv.org/abs/1901.11356`.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers distillation through attention. 2020. URL `http://arxiv.org/abs/2012.12877`.

Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 2017. URL `https://github.com/szagoruyko/attention-transfer`.

Allan Zhou, Tom Knowles, and Chelsea Finn. Meta-Learning Symmetries by Reparameterization. 2020. ISSN 2331-8422. URL `http://arxiv.org/abs/2007.02933`.

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A Comprehensive Survey on Transfer Learning, nov 2021. ISSN 15582256. URL `https://arxiv.org/abs/1911.02685v3`.

Supplementary Material: Can Functional Transfer Methods Capture Simple Inductive Biases?

A EXPERIMENTAL SETUP

A.1 MNIST-1D Experiments

Our goal was to design a simple teacher model that would generalize well to unseen shifts of the MNIST1D task in our evaluation setup. We ended up with a two-layer fully convolutional architecture with ReLU activations (Agarap, 2018), a stride of one and kernels of size five for all convolutions, fifteen channels in the first and ten in the second layer. A spatial max-pooling operation is performed at the final layer to obtain predictions for ten classes.

The student network is designed to replicate the teacher network as closely as possible, while using fully-connected layers instead of convolutions. Thus, to match the output-size of each layer, we use two hidden layers of size 600 and 400 with a ReLU activation in-between. To replicate the final max-pooling layer, we experiment with two variants of the student architecture. One architecture that replaces the max-pooling with a fully-connected layer, and then an alternative architecture that reshapes the output of the previous fully-connected layer into spatial and channel dimensions and on this representation performs a spatial max-pooling operation. The latter architecture simplifies the transfer problem, as only the equivariance of the lower layers has to be transferred instead of invariance of the entire network. More details on the architectures can be found in Table 3.

The training is always performed using Adam optimizer (Kingma and Ba, 2014) for 40,000 iterations with a batch size of 1000 examples. After every 100 steps the validation performance is evaluated and the learning rate is reduced by a factor of 0.8 if the accuracy has not improved for 20 evaluations. The training is interrupted if the learning rate has been reduced five times or if the maximum of 40,000 iterations is reached.

Teacher	Student	
1×15 Conv	40×600 FC	
ReLU	ReLU	
15×10 Conv	600×400 FC	
max-pooling	max-pooling	ReLU
		400×10 FC

Table 3: Comparison of different architectures we use for the MNIST1D experiments. (Channel-size for Conv and hidden-size for FC.) Note that both student architectures are identical for the first two layers.

Orbit Model

The model for the orbit transfer is selected in a way that allows it to capture the shift equivariance that we hope to transfer. Thus we chose a model that learns 40 filters of size 40 to model $\rho_g^{(0)}$ for $g = [0, 39]$. To additionally model $\rho_g^{(l)}$, we learn affine transformations (40×40 projections) for each layer as well as a down-projection for the output layer (40×10 projection).

Transfer Hyperparameters

As discussed in Section 2, we use a hyperparameter γ to interpolate between the standard cross-entropy loss $\mathcal{L}_{CE}$ and the transfer term Ω. In an extensive hyperparameter search, we selected the γ for each transfer method that performed best on the validation set. This left us with the following values: Attention 0.9, KD 0.6, Direct matching 0.4, RDL 0.9, Orbit 1.0. The same hyperparameter search gave us weights for the different components of the objective for learning the Orbit model ρ. The values selected for the final evaluation were $\gamma_{\text{equiv}} = 0.1, \gamma_{\text{group}} = 10.0, \gamma_{\text{inv}} = 10.0$, and γ_{CE} was fixed to 0.0.

A.2 MNIST-2D Shift Equivariance Experiments

A.2.1 CNN → MLP

In contrast to Abnar et al. (2020), we want to isolate the effect of shift equivariance and thus end up with a simpler architecture for both teacher and student. The teacher network has three convolutional layers with 128, 64 and 64 channels, a stride of one and a filter size of 3×3 each. Maximum-pooling is performed after layers two and three. A final linear layer is used to obtain the network's output. The student equals the teacher network in depth, but differs in layer size, with fully-connected layers of size 512,128 and 32. This discrepancy is expected in architectures as different as fully connected networks and CNNs, and it prevents the use of direct matching within the network for function transfer, as the layer outputs do not match in size. Both student and teacher use ReLU activation (Agarap, 2018), as well as dropout with $p = 0.1$ after each layer. More architectural details can be found in Table 4.

The training is performed using Adam optimizer (Kingma and Ba, 2014) with a learning-rate of 0.0003, a schedule of linear learning-rate warmup for 20 epochs, and a decay by factor 0.8 if there is no improvement in validation accuracy for 20 epochs. The training stops if the learning-rate is reduced five times or if it reaches 400 epochs. This training schedule is intentionally designed to be rather conservative, specifically to benefit functional distance methods as it was reported that knowledge distillation needs a lot of patience in training (Beyer et al., 2021).

Teacher	Student
1×128 Conv	784×512 FC
ReLU	ReLU
Dropout ($p = 0.1$)	Dropout ($p = 0.1$)
128×64 Conv	512×128 FC
ReLU	ReLU
max-pooling (2×2)	
Dropout ($p = 0.1$)	Dropout ($p = 0.1$)
64×64 Conv	128×32 FC
ReLU	ReLU
max-pooling (3×3)	
Dropout ($p = 0.1$)	Dropout ($p = 0.1$)
avg-pooling (global)	
64×10 FC	32×10 FC

Table 4: Comparison of student and teacher architecture for the MNIST experiments. (Channel-size for Conv and hidden-size for FC.)

A.2.2 ResNet18 → ViT

For the more realistic setting where we transfer a ResNet18 teacher to a ViT student, we use the same training and evaluation setup as above. The only change comes through the architectures we use. This is on the one hand a standard ResNet18, as it was described by He et al. (2015), and on the other hand a smaller variant of the vision transformer architecture (Dosovitskiy et al., 2020). The major changes compared to the original design are that we use six layers with eight attention heads and an embedding-size of 64. The patch-size applied on the input image is 7×7 pixels and we use a hidden-size of 128 for all position-wise feed-forward operations.

A.2.3 Orbit Model

The orbit model is trained with the same schedule as teacher and student. Here the validation loss is used to determine learning-rate decay and early stopping. We learn 25 kernels of size 5×5 to model $\rho_g^{(0)}$ for $g \in [0, 24]$. Additional affine projections are learned for each layer of the student, i.e. to transform $\rho_g^{(0)}$ into $\rho_g^{(l)}$. These projections are fully-connected, and thus we learn matrices of size 25×25 for each layer except for the output layer. There the filters are projected to size 10 instead. In order to stabilize the training, we cut off the gradient to the transformed input $\rho_g^{(0)}x$, which means that $\rho_g^{(0)}$ will only receive its feedback through $\rho_g^{(l)}$ with $l > 0$, which are affine projections of $\rho_g^{(0)}$.

A.2.4 Transfer Parameters

In prior experiments, we determined a reasonable value for γ for each transfer method. This left us with the following values: Attention 0.9, KD 1.0, RDL 0.8, Orbit 1.0. We also found that γ_{group} has a significant effect on the performance. Thus we searched for its value as part of our hyperparameter search and found γ_{group} to work best. The other components of the objective are all set to one.

A.3 MNIST-2D Rotation Equivariance Experiments

A.3.1 G-CNN $\rightarrow$ MLP

The student model in the rotation transfer experiments is the same MLP network that we used in the MNIST-2D experiments above. The training setup also remained unchanged, except that for the rotation transfer experiments we set a larger batch-size of 256 and train only for 200 epochs, as we noticed no change in performance after that point during prior experiments.

For the teacher model, we mainly followed the architecture of the G-CNN by Cohen and Welling (2016), but reduced the depth. Details can be seen in Table 5.

Teacher	Student
1×8 $p4$-Conv (kernel-size 5)	784×512 FC
max-pooling (2×2)	
ReLU	ReLU
8×32 $p4$-Conv (kernel-size 3)	512×128 FC
max-pooling (2×2)	
ReLU	ReLU
32×64 $p4$-Conv (kernel-size 3)	128×32 FC
max-pooling (2×2)	
ReLU	ReLU
64×10 $p4$-Conv (kernel-size 3)	32×10 FC
max-pooling over rotations	
Global spatial avg-pooling	

Table 5: Comparison of student and teacher architecture for the MNIST experiments. (Channel-size for Conv and hidden-size for FC.)

A.3.2 Orbit Model

As mentioned in Section 5, we decided to use a more general architecture for the orbit representation. Thus, we learn a separate affine transformation matrix $\rho_g^{(l)}$ for each group element $g = 1, \ldots, G$ and layer $l = 1, \ldots, L$. Each $\rho_g^{(l)}$ is initialized to a random affine transformation within a fair range (i.e. such that an image would still be recognizable). For this, we are considering shifts in range $[-0.1, 0.1]$ in x and y direction, rotations in range $[0, 360]$ degree along all axes, scaling by a factor in range $[0.8, 1.2]$ and shears by a factor in range $[0.0, 15.0]$. We trained models for $G = 4$, 8 and 25. All of our models outperformed existing transfer methods, but $G = 25$ performed the best. For this model, we report the average performance across three seeds in Table 2.

B ABLATION STUDY ON LOSS COMPONENTS

We performed an ablation study to determine the interplay and importance of individual loss components. The results show that the best performance is reached when all loss components are used together (see Table 6). However, there are many combinations that show an acceptable performance on centered images while achieving above-baseline results on translated inputs. The most notable negative effect is observed when both the equivariance, as well as the invariance components are left out.

γ_{group}	γ_{equiv}	γ_{inv}	γ_{CE}	Centered	Translated
10.0	1.0	1.0	1.0	**99.02**	**96.48**
10.0	1.0	1.0	-	99.02	96.48
10.0	-	1.0	1.0	89.98	88.65
10.0	-	1.0	-	92.82	89.68
10.0	1.0	-	1.0	98.36	89.98
10.0	1.0	-	-	98.45	93.8
10.0	-	-	1.0	37.25	52.82
10.0	-	-	-	43.38	58.69
-	1.0	1.0	1.0	98.78	85.01
-	1.0	1.0	-	98.48	81.18
-	-	1.0	1.0	98.37	89.54
-	-	1.0	-	98.59	90.98
-	1.0	-	1.0	98.24	77.36
-	1.0	-	-	98.05	83.51

Table 6: Ablation study deactivating the different components of the objective: γ_{group}(Eq. 3), γ_{equiv} (Eq. 2), γ_{inv} (Eq. 4) and γ_{CE} (standard loss on transformed inputs)