Tilt your Head: Activating the Hidden Spatial-Invariance of Classifiers

Johann Schmidt [1] **Sebastian Stober** [1]

Abstract

Deep neural networks are applied in more and more areas of everyday life. However, they still lack essential abilities, such as robustly dealing with spatially transformed input signals. Approaches to mitigate this severe robustness issue are limited to two pathways: Either models are implicitly regularised by increased sample variability (data augmentation) or explicitly constrained by hard-coded inductive biases. The limiting factor of the former is the size of the data space, which renders sufficient sample coverage intractable. The latter is limited by the engineering effort required to develop such inductive biases for every possible scenario. Instead, we take inspiration from human behaviour, where percepts are modified by mental or physical actions during inference. We propose a novel technique to emulate such an inference process. This is achieved by traversing a sparsified inverse transformation tree during inference using parallel energy-based evaluations. Our proposed inference algorithm, called Inverse Transformation Search (ITS), is model-agnostic and equips the model with zero-shot pseudo-invariance to spatially transformed inputs. We evaluated our method on several benchmark datasets, including a synthesised ImageNet test set. ITS outperforms the utilised baselines on all zero-shot test scenarios.

1. Introduction

A primary objective in Deep learning is to develop models that exhibit robust generalisation during inference (commonly evaluated using test sets). To improve generalisation, models are trained with data sets of increasing variability and size. However, this often requires using larger models (Chefer et al., 2022). As novel datasets more closely mirror real-world scenarios, models undergo rigorous assessments in diverse, dynamic environments. This requires the models to learn intricate functions that generalise to all the variations presented in the data. It was shown that modern vision models break even under minor input perturbations (Engstrom et al., 2019), posing severe risks for real-world applications. Each new variation (dimension) scales the data space exponentially, known as the *combinatorial explosion*. Sampling in such spaces gets progressively harder (*curse of dimensionality*). For instance, consider an object from infinitely many perspectives. In a standard image classification task, all these variations have the same label assigned. Using 2D spatial transformations, these variations bring only redundant information. Hence, considering every conceivable variant is impractical, computationally infeasible, and even useless in terms of information gain. If a model can correctly predict an object from all perspectives, it is *(pseudo-)invariant* to these perspective changes.

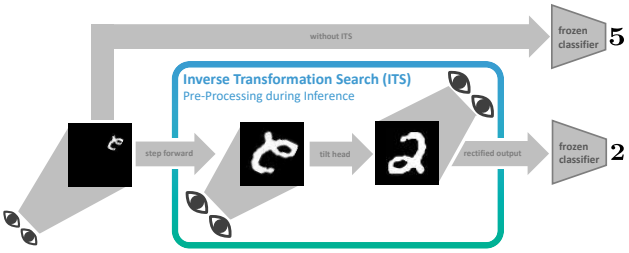

Figure 1. Inspired by the human recognition process for perturbed inputs, we propose a novel inference framework coined Inverse Transformation Search (ITS). Similar to humans transforming the given input mentally or physically by tilting their heads or stepping forward, for instance. Instead of the usual likelihood estimation of a given sample, the pre-trained model can evaluate different input transformations before a final decision is made. The output of this process is the most familiar transformation of the query.

Data-driven Pseudo-Invariance Over recent years, data augmentation (Hernández-García & König, 2018; Shorten & Khoshgoftaar, 2019) has become a routine instrument for increasing generalisation. The main drivers for this development are its simplicity and model agnosticism. Data augmentation enriches a dataset by applying transformations or modifications to existing data points. Augmented training data acts as an explicit Frobenius norm regularizer (LeJeune et al., 2019) promoting the extraction of consistent class features (Huang et al., 2021). Models either learn transformed

[1]Artificial Intelligence Lab, Otto-von-Guericke University, Magdeburg, Germany. Correspondence to: Johann Schmidt <johann.schmidt@ovgu.de>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

replicates of the same filter (Olah et al., 2020) or fuzzy averages of transformed per-class features, regularising model complexity (Chen et al., 2019; Dao et al., 2019). This behaviour can be amplified using invariance-promoting losses (Shakerinava et al., 2022).

Goodfellow et al. (2009) showed that the quality of pseudo-invariance in the learned representation increases with the depth of the network. Minor deformations, like small shifts and rotations, are compensated in lower layers, while higher layers progressively form invariances. Augmented training can encourage this behaviour, as the model aims to align the kernel space with the largest eigenvector of the sample covariance (Balestriero et al., 2022b). However, strong forms of data augmentation, like cropping, can also hurt generalization as it can introduce feature biases by varying the dominance of features (Balestriero et al., 2022a).

Model-driven (Pseudo-)Invariance Inductive biases for spatial symmetries are commonly developed for CNNs by exploiting their weight-shared filters (Schmidt & Stober, 2022). Instead of regularising the filter space using data, it can be constrained explicitly to symmetric weight matrices (so-called isotropic filters) (Hu et al., 2019). Invariance can also be achieved without such constraints by considering all transformed filter variations followed by a max-out (Gens & Domingos, 2014; Cohen & Welling, 2016; Bao & Song, 2020). This can be formulated more flexibly by a steerable basis for the space of convolutional kernels (Weiler et al., 2017). This allows for weight sharing across filter orientation without aliasing artifacts. Recently, various options for such basis functions were proposed, most of which are derived from circular symmetries. This includes Bessel functions (Qiu et al., 2018; Delchevalerie et al., 2021), Hermite polynomials (Ecker et al., 2019; Ustyuzhaninov et al., 2020) and irreducible representations to steer the harmonics (Weiler & Cesa, 2019). Zhou et al. (2020) proposed to meta-learn the filter structure (binary mask) based on the equivariance required to solve a task. Recently, it was shown that equivariant models can be *canonicalised* by fitting a prior over the group during training (Kaba et al., 2023; Mondal et al., 2023).

Unfortunately, these inductive biases come with a significant engineering effort, which renders a generalisation to other symmetries cumbersome. The same applies to methods, which learn the geometry of the latent space (Arvanitidis et al., 2017) to find equivariance paths. With the introduction of Spatial Transformer Network (STN) (Jaderberg et al., 2015), another promising avenue began. These networks remove the spatial perturbations, i.e., *canonicalise* the transformed input signals by learning a highly non-linear mapping from pixel to transformation space. A more effective alternative is the mapping to canonical coordinate spaces (Esteves et al., 2018; Tai et al., 2019). To learn transformation compositions, STNs can be stacked (Lin & Lucey, 2017; Detlefsen et al., 2018). These transformations can also act on each pixel individually using pixel-wise motion fields (Shu et al., 2018). However, all of these variants require augmented training data. *We propose a novel inference strategy that canonicalises inputs without augmented training.*

Inference-driven Pseudo-Invariance We take inspiration from human behaviour and propose a third way to integrate pseudo-invariances. The human brain learns to discard redundancies imposed by symmetries for a more efficient perception (Gómez et al., 2008; Leibo et al., 2011). To recognise a perceived object, our brain aligns it with our learned mental model of that object. This process is called *canonicalisation* (Graf, 2006) and it is either performed mentally or physically (or a mixture of both). We can leverage our physical abilities by squeezing our eyes, tilting our heads and slowly moving closer when triggered by an unclear perception. In this way, our trained mental model can properly recognise a variety of perturbed inputs by gradually aligning them to their known canonical form. This is a well-established theory supported by various experiments (Palmer et al., 1981; Harris et al., 2001; Gómez et al., 2008; Konkle & Oliva, 2011).

We can transfer this hypothesis by emulating the alignment search algorithmically as illustrated in Figure 1. *We propose an Inverse Transformation Search (ITS) for trained classifiers during inference, which renders them zero-shot pseudo-invariant by only requiring access to their logit scores.* We disentangle the process of class-specific feature learning (achieved during training) and canonical signal distillation (achieved during inference). The latter removes perturbations from the input and therefore eliminates variances from the query to ease the subsequent prediction.

Scope and Outline In this work, we particularly focus on 2D image classifiers and the special linear group, comprising combinations of rotation, scaling, and shearing operations. However, our proposed method is more generally applicable. In Section 2, we start with an introduction to the language of symmetries, i.e., group theory. Upon these fundamentals, we build a group-induced evaluation score in Section 3 used in our search algorithm presented in Section 4. We evaluate our methods in Section 5 and conclude with future directions in Section 6.

2. Preliminaries

Understanding how to undo transformations requires a basic comprehension of symmetries and their mathematical formulations as groups. A *symmetry* of an object is a transformation that preserves the object's structure. For instance, rotating an equilateral triangle around its centre by $0°$, $120°$ or $240°$ leaves the triangle unaffected, as illustrated in Figure 2 (left). These rotations form a *group* of finite size, i.e., a *finite group*.

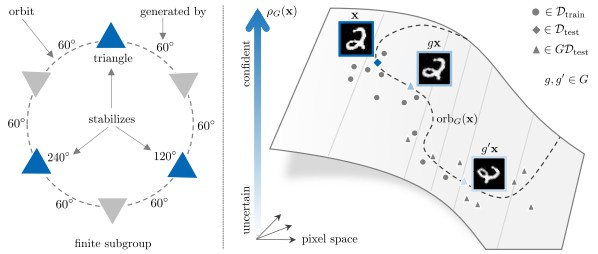

Figure 2. (left) Centre-Rotating an equilateral triangle and *(right)* an exemplary confidence surface.

Definition 2.1 (Group)**.** Groups are sets of objects equipped with a binary operation. For every group G the following axioms must hold

$$\begin{array}{ll} \text{closure:} & g,h \in G \rightarrow gh \in G, \\ \text{associativity:} & g(hk) = (gh)k, \\ \text{identity:} & eg = ge = g, \\ \text{invertibility:} & gg^{-1} = g^{-1}g = e, \end{array}$$

with $g,h,k,e \in G$ and e being the identity element.

This includes a composition of multiple group elements, like gh, which is the short form for $g \circ h$, where $\circ$ is some binary operator. Some groups possess a fifth property, that is, *commutativity*. We call them *Abelian groups*, e.g., centre-rotations. More details and proofs can be found in (Nash, 2016).

Let us extend the picture from discrete to continuous structures. For a circle, centre rotations of any angle send the object back to itself. Formally, these groups are called *Matrix Lie groups* (Hall, 2015), which form a smooth differentiable manifold. For the basic comprehension of this work, it is only important to understand that group elements can be represented by parameterised matrices. In this work, we focus on centre-rotation, isotropic-scaling, and shearing transformations (and their compositions). All of them are represented by 2×2 matrices $\mathbf{T}$. This particular group is called the Special Linear Group $\mathrm{SL}_2(\mathbb{R})$ (Hall, 2015).
When dealing with continuous structures, it is helpful to consider finite approximations instead. Given a group G we can have a *subgroup* $G' < G$, which still satisfies Def. 2.1 but with a smaller *cardinality* $|G'|$. For instance, a subgroup of all centre-rotations is the finite group with multiples of $10°$ rotations. Such groups can be generated by a single group element.

Definition 2.2 (Generator)**.** Given $g \in G$, g can be used as a generator to form a finite (cyclic) subgroup $\{\langle g \rangle^k \,|\, k \in \mathbb{Z}\}$. $\langle g \rangle^k$ denotes g to be a generator acting k-times. To remove visual clutter, we use $\hat{g}$ to denote the generating group element.

A group element *acts* on an object $\mathbf{x}$ by transforming it. If the entire group G acts on an object, the resulting output set is called the orbit of the object under G.

Definition 2.3 (Orbit)**.** Let G be a group and $\mathbf{x}$ an object. Then, the orbit of $\mathbf{x}$ under G is

$$\mathrm{orb}_G(\mathbf{x}) = \{g\mathbf{x} \,|\, g \in G\}.$$

All g that leave $\mathbf{x}$ unchanged are symmetries of that object.

Definition 2.4 (Stabiliser)**.** The stabiliser of some $\mathbf{x}$ under G is a subgroup that leaves $\mathbf{x}$ unchanged,

$$\mathrm{stab}_G(\mathbf{x}) = \{g \in G \,|\, g\mathbf{x} = \mathbf{x}\}.$$

We use stabilisers to formalise spatial symmetries, e.g., transformations that preserve the pixel positions of an image. We can also formalise this from a function perspective. A function f (e.g., modelled by a neural network), which captures a symmetry defined by group G, is either *equivariant*, *invariant*, or *covariant* (Marcos et al., 2017).

Definition 2.5 (Variances)**.** Let $f(\mathbf{x})$ be a function of an input $\mathbf{x}$. Then, f is

$$\begin{array}{llll} \text{equivariant} & \text{iff} & f(g\mathbf{x}) = gf(\mathbf{x}), & \forall g \in G, \\ \text{invariant} & \text{iff} & f(g\mathbf{x}) = f(\mathbf{x}), & \forall g \in G, \\ \text{covariant} & \text{iff} & f(g\mathbf{x}) = g'f(\mathbf{x}), & \forall g \in G, g' \in G', \end{array}$$

with G' being isomorphic to G.

Note that covariance is the most general one, where both transformations differ in their domain and co-domain, respectively. Invariance and equivariance are special cases of covariance. If Def. 2.5 does not hold but is approximately equal, we call the variance a *pseudo-variance*. For more elaborated details about variances in deep learning, see (Bronstein et al., 2021).

3. Localizing Canonical Forms

In this section, we introduce a novel method for finding canonical forms. We use $f_\theta : \Omega \rightarrow \mathbb{R}^Y$ to denote a parameterised function mapping from pixel space to class-logits. We call this function a softmax-classifier if it parameterises a categorical distribution by its logits, such that

$$p_\theta(y \,|\, \mathbf{x}) = \mathrm{softmax}(f_\theta(\mathbf{x})) := \frac{\exp(f_\theta(y \,|\, \mathbf{x}))}{\sum_{y'} \exp(f_\theta(y' \,|\, \mathbf{x}))}. \quad (1)$$

During training, we obtain the posterior

$$p(y \,|\, \mathbf{x}, \mathcal{D}_{\text{train}}) = \int p_\theta(y \,|\, \mathbf{x}) p(\theta \,|\, \mathcal{D}_{\text{train}}) d\theta. \quad (2)$$

G-transformed Test Sets Commonly, models are tested on holdout sets with the same spatial variation. Instead, we aim to train a classifier on canonical examples (vanilla training data) and subsequently evaluate it on a G-transformed

test set. This transformation introduces a distribution shift between $\mathcal{D}_{\text{train}}$ and $G\mathcal{D}_{\text{test}}$. Therefore, requiring $p(y\,|\,\mathbf{x},\mathcal{D}_{\text{train}})$ to perform a zero-shot task on the out-of-distribution test set. We argue, that this more accurately reflects real-world scenarios. Under slight abuse of notations, we define $G\mathcal{D}_{\text{test}} := \{\langle \mathbf{x}',y\rangle \,|\, \mathbf{x}' = g\mathbf{x}, \langle \mathbf{x},y\rangle \sim \mathcal{D}_{\text{test}}, g\sim G\}$. Note that, $G\mathcal{D}_{\text{test}}$ and the vanilla $\mathcal{D}_{\text{test}}$ intersect as the stabilisers of all samples are contained. For all our experiments, we use $\mathcal{D}_{\text{train}}$ for training and $G\mathcal{D}_{\text{test}}$ during inference.

Max-Confidence Estimation We derive an energy-based evaluation from the concept of invariances. We assume that the function f_θ is sensitive to G, which is true for all neural networks when G is getting more complex. That is, for any non-trivial element g in G we have $f_\theta(\mathbf{x}) \neq f_\theta(g\mathbf{x})$. However, based on Def. 2.1, there exists a $g^{-1}\in G$ such that $f_\theta(\mathbf{x}) = f_\theta(g^{-1}g\mathbf{x})$. Inspired by Kaba et al. (2023) we can determine g^{-1} using a confidence measure $\rho\colon f_\theta(\cdot)\to\mathbb{R}$, such that

$$g^{-1} \leftarrow \operatorname*{argmax}_{h\in G} \rho(f_\theta(hg\mathbf{x})). \tag{3}$$

This would allow us to undo perturbations induced by real-world measurements. An illustration of a confidence surface is depicted in Figure 2 (right). As ρ is the limiting factor of such an approach, we study three different options in the following.

Reducing Prerequisites Any such confidence measure relies on a semantically structured latent space (Bengio et al., 2012). Although a multitude of methods exist to access prediction confidence. Many of these either pose a prior on the model to learn confidence during training (Moon et al., 2020) or require access to the training data to approximate the data manifold (Snell et al., 2017; Kotelevskii et al., 2022). To maintain a minimal level of obstacles, we seek a post-hoc method without any knowledge about the training set.

Energy-induced Confidence The probability masses of softmax-classifiers are often spoiled by overconfident posteriors even for out-of-distribution samples (Lee et al., 2018; Geifman et al., 2018). Therefore, the model's energy surface is often used as a straightforward approximation to the confidence surface (Liu et al., 2020). Duvenaud et al. (2020) showed that Energy-based models (Lecun et al., 2006) are implicitly contained in both discriminative and generative models. This renders Energy-based measures widely applicable. Following Duvenaud et al. (2020) we determine the total energy of a model f_θ by

$$E_\theta(\mathbf{x}) = -f_\theta(\mathbf{x}) = -\log\sum_y \exp f_\theta(y\,|\,\mathbf{x}), \tag{4}$$

with $p_\theta(\mathbf{x}) \propto \sum_{y'} \exp\left(f_\theta(y'\,|\,\mathbf{x})\right)$. We denote class-wise energies by $E_\theta(y\,|\,\mathbf{x}) = -f_\theta(y\,|\,\mathbf{x})$. Using this relation, we can use any trained classifier and interpret its logit scores as energies. Liu et al. (2020) showed that the negative energy is a straightforward confidence estimate.

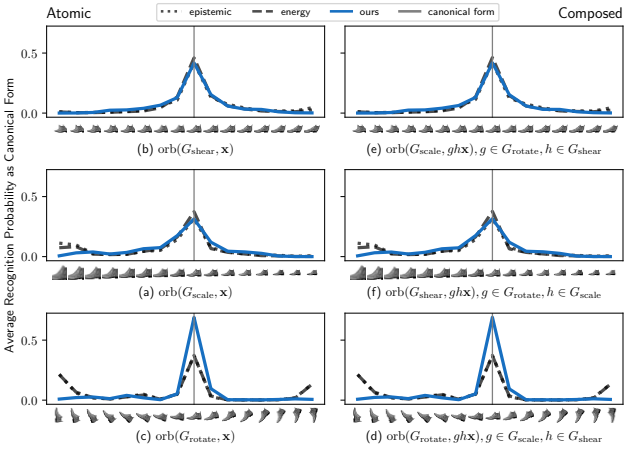

Figure 3. Probability of recognising a transformed sample to be the canonical form. Each evaluation point represents the mean over the FashionMNIST test set.

Bayesian-induced Confidence Commonly, logits in Equation (1) are parameterized by a single θ obtained by Maximum Likelihood Estimation on $\mathcal{D}_{\text{train}}$. Instead of such point estimates, Bayesian neural networks (Osband, 2016) approximate a parameter distribution, hence enabling richer statistical insights, such as confidence estimation. Unfortunately, exact posterior estimation is known to be intractable due to the integration over the parameter space. Unlike common approximation schemes (Osawa et al., 2019) including ensemble methods (Lakshminarayanan et al., 2017), we only have access to a trained classifier to ease the applicability of ITS.

To the best of our knowledge, the only method to fulfil this requirement is Monte-Carlo Dropout (Gal & Ghahramani, 2016). Each parameter is endowed with a random process regulating its influence in the forward pass during inference. Let $\mathbf{\Lambda}_l := \operatorname{diag}(\boldsymbol{\lambda}_l) \sim \text{Bernoulli}(\lambda \in [0,1])$ be a diagonal matrix of random variables for some layer l. For each inference pass, we sample $\mathbf{\Lambda}_l$ and apply it to the weights as multiplicative noise, $\mathbf{W}_l \leftarrow \mathbf{\Lambda}_l \mathbf{W}_l$. We can steer the chance of dropout by $1-\lambda$. This allows us to define an unbiased estimator for the expectation,

$$\mathbb{E}_{p_\theta(y|\mathbf{x})}[y] = \int y p_\theta(y\,|\,\mathbf{x}) dy \approx \frac{1}{M}\sum_{m=1}^{M} p_\theta(y\,|\,\mathbf{x}, \{\mathbf{\Lambda}_{m,l}\}_{\forall l}). \tag{5}$$

The model's confidence corresponds to the negative total uncertainty, which we can estimate using the entropy,

$$\mathbb{H}\left[\mathbb{E}_{p_\theta(y|\mathbf{x})}[y]\right] \approx -\sum_y \mathbb{E}_{p_\theta(y|\mathbf{x})}[y] \log_2 \mathbb{E}_{p_\theta(y|\mathbf{x})}[y]. \tag{6}$$

Confidence Anomalies The confidence measure should concentrate all its mass around the canonical form. However, we found that both baselines tend to increase their mass towards the domain borders. This is best observed in the rotation experiment in Figure 3 (last row). Here, the spatial alignment bias of FashionMNIST (Xiao et al., 2017) leads to increased confidence for 90° rotated images (high activation of horizontal edge detectors). Similar confidence bumps appear when scaling up or down a sample until pixelation or disappearance. Without explicit training or inductive biases, neural networks corrupt the geometry of group orbits during encoding (Arvanitidis et al., 2017). When traversing the orbit's trajectory, the energy will fluctuate (Li et al., 2018). As both confidence scores utilise the energy, the confidence will fluctuate likewise.

Group-induced Confidence We propose a more stable measure, which does not suffer from these confidence anomalies. First, we remove noise from the energy by estimating its expected value $\bar{E}_\theta(\cdot)$ using Monte-Carlo Dropout. This renders the measure stochastic (as the Bayesian-induced confidence), reducing the impact of outliers. To further mitigate instabilities, we evaluate local neighbourhoods instead of single point estimates. This can be achieved by convolving the energy estimate over the orbit with a kernel. Lastly, we compute the negative curvature of the resulting energy surface instead of using the energy estimate directly. This allows us to reduce the confidence mass at the domain borders by using "nearest" padding, which lowers the curvature at these points. We approximate the curvature by the second derivative, such that

$$\rho_G(\mathbf{x},g) = -\frac{\partial^2}{\partial g^2}\big(\bar{E}_\theta \star K\big)\big(g\mathbf{x}\big), \tag{7}$$

where K is a Gaussian smoothing kernel to reduce local over-confidence. We use a generator $\hat{g}$ to generate the finite group G. Therefore, we can evolve $\mathbf{x}$ along $\text{orb}_G(\mathbf{x})$ by incrementing or decrementing n in $\hat{g}^n\mathbf{x}$. In practice, we can approximate Equation (7) as follows.

Theorem 3.1 (Group Confidence)**.** *Let $\phi_{\bar{E}_\theta,K}(\mathbf{x},\, g) := \big(\bar{E}_\theta \star K\big)\big(g\mathbf{x}\big)$. The group-induced confidence defined in Equation (7) can be approximated by*

$$\rho_G(\mathbf{x},g) = -\lim_{|G|\to\infty} \frac{1}{\|\hat{g}-\hat{g}^2\|^2} \tag{8}$$
$$\Big(\phi_{\bar{E}_\theta,K}(\mathbf{x},\hat{g}\circ g) + \phi_{\bar{E}_\theta,K}(\mathbf{x},\hat{g}^{-1}\circ g) - 2\phi_{\bar{E}_\theta,K}(\mathbf{x},g)\Big).$$

Find the proofs in Appendix E.1 and Appendix E.2.

A comparison of all three measures is depicted in Figure 3. Our group-induced confidence either surpasses both baselines or achieves on par results.

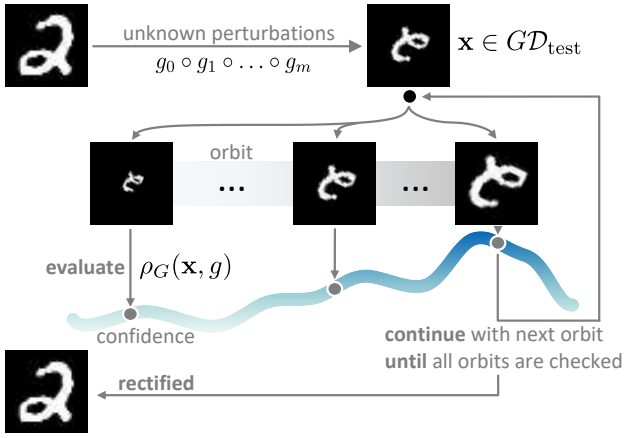

Figure 4. Our proposed Inverse Transformation Search (ITS) on a perturbed image. In each level of the search tree, the orbit of one subgroup is evaluated. All samples are evaluated using Equation (7) and a pre-trained classifier.

4. Inverse Transformation Search

An overview and an example of our proposed search procedure are illustrated in Figure 4. In the following, we spell out the most important aspects of it with more details in Appendix A and Appendix D (including pseudocode).

Transforming Images An image $\mathbf{x}$ is a collection of pixel vectors containing spatial and colour information. Spatial transformations are modifications of the location a pixel vector is pointing to. To resample the transformed image, colour values are interpolated between the modified pixel positions. This procedure can be efficiently executed in parallel on the GPU (Jaderberg et al., 2015).

Forming Search Levels We deal with problem scenarios, where samples are perturbed by a some unknown affine transformation g during test-time. We assume that g decomposes into various subgroup transformations, $g = g_0 \circ g_1 \circ \ldots \circ g_m$. We propose an iterative search procedure, which aims to eliminate one g_k at every step using its inverse g_k^{-1}. This heuristic (greedy) approach reduces the complexity of the problem significantly. However, this comes with the assumption that either the order of transformation is known, which is unlikely in real-world settings, or that we deal with Abelian groups. We show that the restrictions imposed by the latter can be attenuated, as only a sub-sequence needs to be Abelian.

Theorem 4.1 (Group Elimination)**.** *Let $s = g_0 \circ g_1 \circ \ldots \circ g_m$ be a finite sequence of group elements with $g_0 \in G_0, g_1 \in G_1, \ldots, g_m \in G_m$ and $m \in \mathbb{N}$. Further, let g_k^{-1} be the inverse of g_k in the sequence for some $k \in [0,m]$ and $s_{\setminus g_k}$ being s without g_k. We have, $g_k^{-1} s = s_{\setminus g_k}$ if all elements in the sub-sequence $s_{0:k-1}$ and g_k^{-1} are elements of an Abelian group.*

The proof can be found in Appendix E.3.

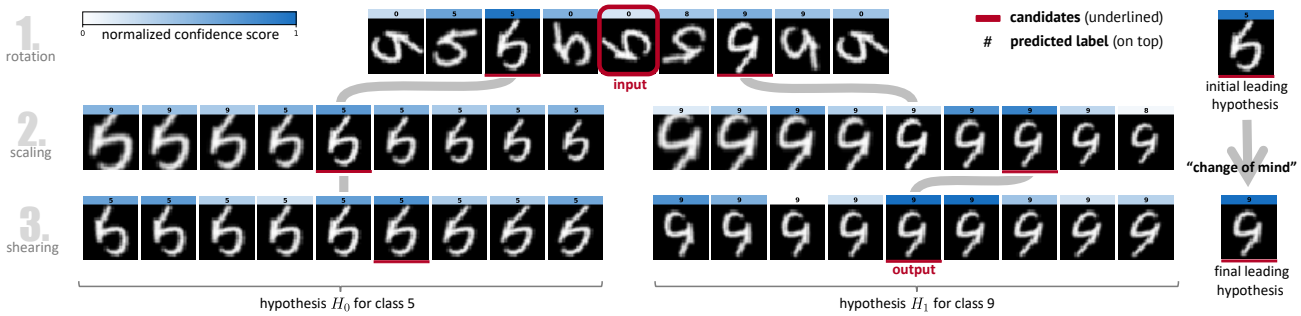

Figure 5. An example iteration of our algorithm with two hypotheses H_0 and H_1, and three generated subgroups, each with a cardinality of $|G| = 9$. The first layer rotates the input, the second layer scales it, and the last layer shears it.

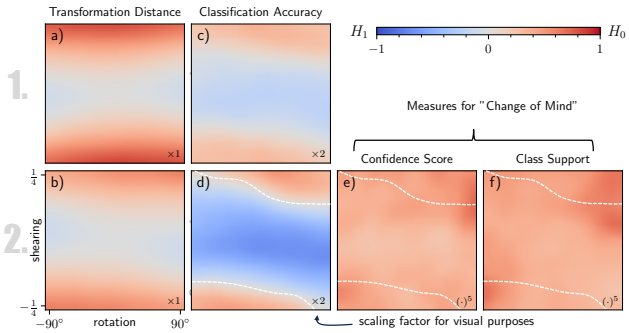

Figure 6. Evolution of H_0 and H_1 in a two-layer search tree (rotation in the top row and shearing in the bottom one) on Fashion-MNIST. Red areas indicate an average superiority of H_0, while blue areas indicate a superiority of H_1.

To cope with Lie groups, we generate subgroups of size $|G|$ using generators as already used in our group-induced confidence measure (see Theorem 3.1). In that light, each layer comprises $|G|$ elements (as a finite set of equidistant samples along the orbit). This allows us to express transformations by an exponential $\hat{g}^n$ with $n \in \mathbb{N}$. This renders the evaluation in each layer tractable, as only a finite set of points needs to be evaluated.[1]

Hypothesis Testing We define a hypothesis H_i as a trajectory down the search tree. Humans use selective hypothesis testing (Sanbonmatsu et al., 1998) when the perceived visual stimulus has high uncertainty. This includes evaluating multiple hypotheses until a final decision is made. We simulate this behaviour by traversing the search tree k times in parallel. This is done by updating a collection of candidate hypotheses $\{H_0, H_1, \ldots, H_{k-1}\}$. After the first expansion and evaluation step, we select the k best candidates ordered by confidence. Each further expansion and evaluation step is done for all H. The k best are selected from the cumulative candidate set, bounding each level of the tree to k children. Find an example of such a k-ary tree in Figure 5.

[1]All non-cyclic groups, like translations, would at some point shift the image signal outside the visible pixel grid. In practice, we consider only a reasonably bounded set of transformations.

Rather than assuming independent trajectories, we pose a linking condition between them to avoid redundant hypotheses. For instance, a rotated image of a 5 might share many features with a 9. If the confidence is high, a significant amount of runs would refine this hypothesis until the end, resulting in a one-sided perspective. We improve the hypothesis testing by allowing only one hypothesis per label to be further refined.

Change of Mind In theory, such a dynamic search process over multiple hypotheses would allow the model to "change its mind". That is, rejecting H_0 (the initially dominating hypothesis) and instead accepting some other H for the final incumbent (output of the inference process). Figure 5 already provides an example for such phenomena, which we investigate more thoroughly in Figure 6. We constructed a regular grid of rotated and sheared Fashion-MNIST test sets. Each point in Figure 6 is the average performance of a CNN with ITS (see Appendix C.3 for more details). Colour changes from layers 1 to 2 indicate that the initial hypothesis is on average superseded by the other. To maximise the performance outcome (we focus on classification accuracy), we need to substitute the leading hypothesis H_0 with its alternative H_1 in some regions. Without a ground-truth signal (during test time), we need a measure to determine when to swap hypotheses. In our experiments, we used class support, which is the number of occurrences of the predicted class over the entire tree. However, this indicator has only minor advantages over the confidence score (see Figure 6 e,f). We leave this promising avenue for future research.

Complexity and Runtime We have a fixed-width k-ary tree, where the number of elements in the first level is $n := |G|$ and in all consecutive levels kn. As all evaluations are performed in parallel, n does not affect the runtime. For our implementation, the Monte Carlo estimates are performed sequentially, scaling the complexity by M. This gives $\mathcal{O}(Mk^d)$, where d denotes the depth of the tree. We quantified the runtime cost in Appendix C.1.

Application Domains Given the runtime costs, we argue that our inference strategy is particularly interesting for applications where it is worth trading runtime against robustness. This includes medical applications, like X-ray classification (Ghamizi et al., 2022); analysis of satellite or drone imagery, like land suitability prediction for agriculture (Taghizadeh-Mehrjardi et al., 2020), monitoring crop health (Inoue, 2020), or identifying deforestation (Andrade et al., 2020); classifying celestial objects in telescopic images (Gómez et al., 2020) or analysing protein structures in biology from microscopic images (Moen et al., 2019).

5. Experiments

In each classifier, there is potential for pseudo-invariance, which we just have to activate. In that light, all investigations are zero-shot performance tests with out-of-distribution data (see Section 3). We split the vanilla datasets into disjunct training, validation, and test sets. We always employ the vanilla training set to fit the model and validate it on the vanilla validation set. All performance measures are subsequently evaluated on the perturbed test set $G\mathcal{D}_{\text{test}}$, where G is specified in each experiment.

If not further specified, we used zero-padding to define areas outside the pixel space Ω, bilinear interpolation, and a group cardinality of $n = 17$. For our experiments on (Fahsion)-MNIST and GTSRB we used two CNN-backbones simple as the LeNet architecture (Lecun et al., 1998) (see Appendix A.2). These models are trained with the AdamW optimizer (Loshchilov & Hutter, 2017) using default parameters. We minimised the negative log-likelihood using ground-truth image labels. We used a learning rate of $5e-3$, 3 epochs for MNIST, 5 epochs for Fashion-MNIST, 10 epochs for GTSRB and mini-batches of size 128. In all MNIST experiments, we removed the class 9 samples, as this introduces a class-superposition for $180°$ rotations. More details can be found in our publicly available source code.[2]

5.1. Zero-Shot rotated Image Classification

We trained a CNN, a GCNN (Cohen & Welling, 2016) and a RotDCF (Cheng et al., 2018) on the vanilla (canonical) MNIST. Both baselines use Filter Decomposition to gain $p4$-equivariance and $p8$-equivariance, respectively. We evaluated their zero-shot performances on the rotated test sets with results shown in Figure 7. Such a model would be robust to centre-rotations. Just by exploiting the confidence surface of the vanilla (shallow) baseline CNN our method boosts its robustness significantly. One can also see, that the rotation-equivariant GCNN outperforms our methods in the majority of test angles. However, such models require hard-coded inductive biases tailored to some specific groups[3]. Our method does not pose any such requirements in the hypothesis space, also allowing for smaller model complexity. We enable pseudo-equivariance to more transformations, which is shown in the subsequent experiments.

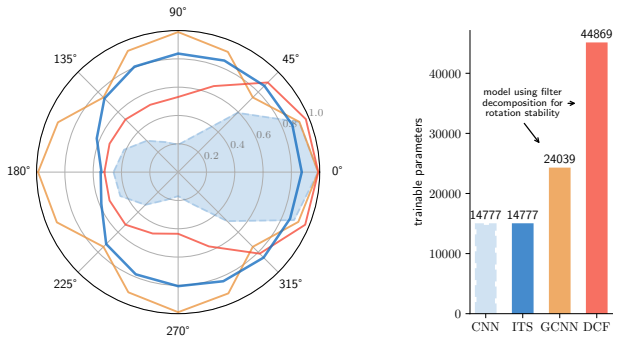

Figure 7. Test accuracy on rotated MNIST with fixed angles.

5.2. Zero-Shot Affine Canonicalisation

Similar to STNs, we can undo affine transformations (i.e., canonicalize inputs). We compare ITS to a STN (Jaderberg et al., 2015) with one and three transformation layers using inverse compositional layers (Lin & Lucey, 2017), a Diffeomorphic Transformer Network (DTN) (Detlefsen et al., 2018), and a Equivariant Transformer Network (ETN) (Tai et al., 2019). To study the impact of depth, we contrasted a 3- and a 5-layer ITS. We used a layer stack of rotation + scaling + shearing + (rotation + scaling). We refer to these as ITS^3 and ITS^5, respectively. Table 1 shows the quantitative results of our experiments. The associated qualitative results as reconstructions (after applying the predicted inverse transformation) can be found in Figure 8.

We found all baselines incapable of performing affine canonicalisations outside the training data distribution. This makes sense, as their localisation networks (regressing the transformation parameters) learn pixel-to-transformation mappings from training data. They are unable to generalise - particularly to rotations, which are mostly present during testing. The STN and the DTN estimate the composed transformation matrix in one forward pass. They need to extrapolate from nearly no rotations during training to $360°$ rotations during testing. As the rotation manifold is isomorphic to the unit circle $(\cos(\theta), \sin(\theta))$, this non-linearity needs to be learned implicitly. The ETN on the other hand uses canonical coordinates, which linearise the relationship between rotations.

While ITS does not always lead to correct results, it

[2] www.github.com/johSchm/ITS

[3] Developing inductive biases for all possible transformations is impossible due to the vastness of possible transformations.

Table 1. Classification accuracies (↑) and Frobenius distances (↓) between the ground truth and the predicted transformation matrix. Mean and standard deviations are estimated over 25 test set runs with random rotation, scaling, and shearing parameters.

DATASET	CNN	STN	STN^3	DTN	ETN	ITS^3	ITS^5
MNIST	50.15±9.29	49.78±9.13	13.85±4.05	48.61±8.99	56.59±9.87	88.89±13.86	**89.81±13.95**
F-MNIST	25.91±5.93	24.51±5.81	12.42±3.73	24.11±5.71	26.10±5.98	**38.47±7.77**	37.58±7.56
GTSRB	32.01±6.59	31.46±6.55	2.78±1.69	28.21±6.23	40.02±7.99	66.44±11.48	**67.09±11.53**
MNIST	23.03±2.48	25.43±2.72	23.61±2.54	31.02±3.31	26.10±2.83	**18.80±2.22**	18.98±2.32
F-MNIST	23.03±2.80	25.96±3.14	23.61±2.87	35.16±4.22	34.02±4.19	**20.25±2.59**	**20.30±2.56**
GTSRB	23.04±2.85	18.15±2.23	23.62±2.92	34.61±4.23	21.08±2.61	**17.87±2.26**	**17.93±2.27**

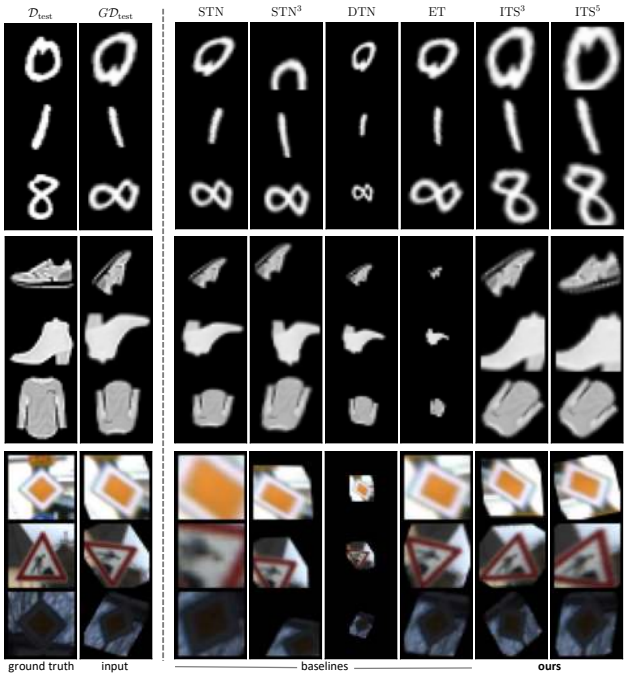

Figure 8. Three test set samples of MNIST (top), Fashion-MNIST (middle), GTSRB (Stallkamp et al., 2012) (bottom), which are transformed by a composed spatial transformation of rotation $[-\pi,\pi]$, scaling $[-\frac{1}{4},\frac{1}{4}]$ and shearing $[-\frac{1}{4},\frac{1}{4}]$. All models use the same CNN backbone, are trained on $\mathcal{D}_{\text{train}}$ and are tested on $G\mathcal{D}_{\text{test}}$. ITS is used with 3 and 5 levels and compared to a STN and a DTN with a single and 3 transformation layers, respectively.

works best among the contestants. In Table 1 we also see that for most cases the Frobenius distance is lowest, too. Interestingly, the distance to the ground truth matrix is lowest for ITS^5 while ITS^3 achieves higher accuracies. High Frobenius distance and high accuracy would indicate that the classifier got an incorrectly rectified sample but is still able to predict the correct class.

5.3. Scaling up to ImageNet

Si-Score (short SI) (Djolonga et al., 2021) is a synthetic vision dataset for robustness testing, comprising semantically masked ImageNet (Russakovsky et al., 2015) objects pasted on various backgrounds. The foreground objects are either arbitrarily (horizontally and vertically) translated (SI-Location), arbitrarily rotated (SI-Rotation), or arbitrarily scaled (SI-Size). We evaluate ITS using a Vision Transformer (ViT)-B16 (Dosovitskiy et al., 2021). The Vision Transformer is pre-trained and validated on ImageNet and tested on Si-Score. We trained the model with and without data augmentation using random rotations $[-\pi,\pi]$ and scales $[-\frac{1}{4},\frac{1}{4}]$.

We found significant improvements on SI-Rotation as denoted in Table 2. To some extent this was expected as our confidence measure is highly sensitive in detecting canonical rotations (see Figure 3). Surprisingly, however, our method even outperformed augmented fine-tuning and explicit training on the foreground object (Chefer et al., 2022). An example is depicted in Figure 9. This shows the brittleness of ImageNet-trained models to spatial transformed inputs (Engstrom et al., 2019) (especially to rotations), as the label is changing frequently along the orbits. An exception is H_1 (third row), where the correct label is preserved over almost the entire orbit of isotropic-scalings. This property is studied in SI-Size.

On SI-Size ITS maintains the performance of the baseline. We hypothesise that location and size are properties that have high variance in the vanilla ImageNet already. This puts the test performance on a high level without requiring any additional mechanisms. Equipping the backbone with the ability to translate the input query with two degrees of freedom causes it to focus on the wrong object. This behaviour is similar to attention but with a fixed centre bias of the backbone and a spatial translation of the signal instead. However, ITS does not get any feedback (gradient) if the object of focus is correct. This leads to slightly degenerated performance on SI-Location (see Appendix C.4).

Furthermore, we tested two variants to obtain acc@5: (i) We computed H_0's categorical distribution and picked the top-5 predictions. (ii) We used the top-1 predictions of each of the five hypotheses ($H_0 \dots H_4$). A higher score for the former would indicate that on average H_0 followed the correct class hypothesis. Whereas, when the latter is higher, one of the alternative hypotheses ($H_1 \dots H_4$) contained the true class

Table 2. Average acc@1 and acc@5 over the SI-Rotation test set using a ViT-B16 pre-trained on ImageNet. The vanilla performance is compared to a rotation and scale augmented fine-tuning, training on foreground objects with and without annotated segmentation masks (AS) and our ITS. We used two acc@5 estimates for ITS: over H_0's categorical distribution and over the top-1 predictions of all H.

SI-Rotation Test Set Samples	ViT	acc@1	acc@5
	vanilla	38.5	58.5
	Rotation + Scale Augm.	41.0 (+2.5)	63.2 (+4.7)
	(Chefer et al., 2022)	44.8 (+6.3)	65.2 (+6.7)
	(Chefer et al., 2022)+AS	46.2 (+7.7)	67.0 (+8.5)
	ITS *(ours)*	**49.0 (+10.5)**	68.7 (+10.2)
			70.9 (+12.4)

Figure 9. SI-Rotation sample, showing a tilted image of a cropped cab (ground-truth label) pasted on a mountain-view background. We applied ITS with rotation $[-\pi,\pi]$ (first row) and isotropic-scaling $[-\frac{1}{2},\frac{1}{2}]$ (second and third row) with two hypotheses (H_0 and H_1).

more frequently than H_0. Unfortunately, our experiments do not show a consistent pattern of which strategy yields higher accuracies. We leave further investigations to future work.

6. Conclusion

Inverse Transformation Search (ITS) equips any vanilla image classifier with pseudo-invariance to various spatial transformations during inference. This increases its robustness to spatially perturbed inputs without requiring extra training data. We activate this potential, which lies hidden in any classifier, without requiring any inductive biases or augmented training. This is achieved by an incremental search for the inverse perturbation. At the canonical form, the features of the transformed input match the features known from the training data. At these points, the model is most confident to classify this sample properly. We successfully showed the superiority of our methods for zero-shot image classification tasks.

6.1. Limitations

Mitigating the following three limitations should be the focus of future works: *(1) False Hypotheses:* Heavily perturbed images can have multiple proper hypotheses. See the example in Figure 6, where both labels (5 or 9) seem reasonable. As ITS uses the representation space of the trained classifier, its performance depends on the quality of these representations. The model might detect features in transformed signals, which lead to false hypotheses. Even worse, these hypotheses might be reinforced down the search tree. See Figure 12 for an example of such behaviour. *(2) Abelian Assumption:* In Theorem 4.1, we showed that only a subsequence of group actions need to be Abelian. Although, this alleviates the impact of the assumption, the issue is not solved. *(3) Runtime Cost:* Due to the complexity of the search process (see Section 4) the walltime increases significantly during inference (see Appendix C.1). In many setups, like video understanding (Bexten et al., 2021), this renders real-time applications intractable.

6.2. Future Work

Given the energy field of the target model, any post-hoc optimisation method can be applied to maximise the confidence of a test sample over a transformation group. This search can be done over the entire group space, not only over subgroups (as used here) tackling the issue with Abelian assumption. This opens up a wide range of algorithms, including for example simulated annealing, which is used in a wide range of domains (Landau & Binder, 2005; Schmidt et al., 2024). As such algorithms are applicable to Quantum-Computers, they hold the potential to eliminate the runtime issue.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

Acknowledgement

Special thanks to all our lab members for the extremely valuable feedback and support during the process. Additionally, we thank the anonymous reviewers for their insightful comments that helped improve this work.

References

Andrade, R. B., Costa, G. A. O. P., Mota, G. L. A., Ortega, M. X., Feitosa, R. Q., Soto, P. J., and Heipke, C. Evaluation of semantic segmentation methods for deforestation detection in the amazon. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2020. 7

Arvanitidis, G., Hansen, L., and Hauberg, S. Latent space oddity: on the curvature of deep generative models. *International Conference on Learning Representations (ICLR)*, 2017. 2, 5

Balestriero, R., Bottou, L., and LeCun, Y. The effects of regularization and data augmentation are class dependent. *Conference on Neural Information Processing Systems (NeurIPS)*, 2022a. 2

Balestriero, R., Misra, I., and LeCun, Y. A data-augmentation is worth a thousand samples: Analytical moments and sampling-free training. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022b. 2

Bao, E. and Song, L. Equivariant neural networks and equivarification. *ArXiv*, 2020. 2

Bengio, Y., Courville, A. C., and Vincent, P. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012. 4

Bexten, S., Schmidt, J., Walter, C., and Elkmann, N. Human action recognition as part of a natural machine operation framework. In *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2021. 9

Borodin, A. N. *Stochastic Processes*. Birkhäuser Cham, 2017. 5

Bronstein, M. M., Bruna, J., Cohen, T., and Veličković, P. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *ArXiv*, 2021. 3

Chefer, H., Schwartz, I., and Wolf, L. Optimizing relevance maps of vision transformers improves robustness. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022. 1, 8, 9, 3

Chen, S., Dobriban, E., and Lee, J. Invariance reduces variance: Understanding data augmentation in deep learning and beyond. *ArXiv*, 2019. 2

Cheng, X., Qiu, Q., Calderbank, R., and Sapiro, G. Rotdcf: Decomposition of convolutional filters for rotation-equivariant deep networks. *ArXiv*, 2018. 7

Cohen, T. and Welling, M. Group equivariant convolutional networks. In *International Conference on Machine Learning (ICML)*, 2016. 2, 7

Dao, T., Gu, A., Ratner, A., Smith, V., De Sa, C., and Re, C. A kernel theory of modern data augmentation. In *International Conference on Machine Learning (ICML)*, 2019. 2

Delchevalerie, V., Bibal, A., Frénay, B., and Mayer, A. Achieving rotational invariance with bessel-convolutional neural networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021. 2

Detlefsen, N. S., Freifeld, O., and Hauberg, S. Deep diffeomorphic transformer networks. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 7

Djolonga, J., Yung, J., Tschannen, M., Romijnders, R., Beyer, L., Kolesnikov, A., Puigcerver, J., Minderer, M., D'Amour, A., Moldovan, D., Gelly, S., Houlsby, N., Zhai, X., and Lucic, M. On robustness and transferability of convolutional neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 8

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations (ICLR)*, 2021. 8

Duvenaud, D., Wang, J., Jacobsen, J., Swersky, K., Norouzi, M., and Grathwohl, W. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations (ICLR)*, 2020. 4

Ecker, A. S., Sinz, F. H., Froudarakis, E., Fahey, P. G., Cadena, S. A., Walker, E. Y., Cobos, E., Reimer, J., Tolias, A. S., and Bethge, M. A rotation-equivariant convolutional neural network model of primary visual cortex. *International Conference on Learning Representations (ICLR)*, 2019. 2

Engstrom, L., Tran, B., Tsipras, D., Schmidt, L., and Madry, A. Exploring the landscape of spatial robustness. In *International Conference on Machine Learning (ICML)*, 2019. 1, 8

Esteves, C., Allen-Blanchette, C., Zhou, X., and Daniilidis, K. Polar transformer networks. In *International Conference on Learning Representations (ICLR)*, 2018. 2

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning (ICML)*, 2016. 4

Geifman, Y., Uziel, G., and El-Yaniv, R. Bias-reduced uncertainty estimation for deep neural classifiers. In *International Conference on Learning Representations (ICLR)*, 2018. 4

Gens, R. and Domingos, P. Deep symmetry networks. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2014. 2

Ghamizi, S., Cordy, M., Papadakis, M., and Traon, Y. L. On evaluating adversarial robustness of chest x-ray classification: Pitfalls and best practices. In *ArXiv*, 2022. 7

Goodfellow, I. J., Le, Q. V., Saxe, A. M., Lee, H., and Ng, A. Y. Measuring invariances in deep networks. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2009. 2

Graf, M. Coordinate transformations in object recognition. *Psychological bulletin*, 2006. 2

Gómez, C., Neira, M., Hernández Hoyos, M., Arbeláez, P., and Forero-Romero, J. E. Classifying image sequences of astronomical transients with deep neural networks. *Monthly Notices of the Royal Astronomical Society*, 2020. 7

Gómez, P., Shutter, J., and Rouder, J. Memory for objects in canonical and noncanonical viewpoints. *Psychonomic bulletin and review*, 2008. 2

Hall, B. *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*. Graduate Texts in Mathematics. Springer International Publishing, 2015. 3

Harris, I. M., Harris, J. A., and Caine, D. Object Orientation Agnosia: A Failure to Find the Axis? *Journal of Cognitive Neuroscience*, 2001. 2

Hendrycks, D. and Gimpel, K. Gaussian error linear units (gelus). *ArXiv*, 2016. 1

Hernández-García, A. and König, P. Data augmentation instead of explicit regularization. *ArXiv*, 2018. 1

Hu, S. X., Zagoruyko, S., and Komodakis, N. Exploring weight symmetry in deep neural networks. *Computer Vision and Image Understanding*, 2019. 2

Huang, T., Sankar, C., Amini, P., Kottur, S., Geramifard, A., Razaviyayn, M., and Beirami, A. Dair: Data augmented invariant regularization. *ArXiv*, 2021. 1

Inoue, Y. Satellite- and drone-based remote sensing of crops and soils for smart farming – a review. *Soil Science and Plant Nutrition*, 2020. 7

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, 2015. 1

Jaderberg, M., Simonyan, K., Zisserman, A., and Kavukcuoglu, K. Spatial transformer networks. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2015. 2, 5, 7

Kaba, S.-O., Mondal, A. K., Zhang, Y., Bengio, Y., and Ravanbakhsh, S. Equivariance with learned canonicalization functions. In *International Conference on Machine Learning (ICML)*, 2023. 2, 4, 1

Konkle, T. and Oliva, A. Canonical visual size for real-world objects. *Journal of experimental psychology, human perception and performance*, 2011. 2

Kotelevskii, N., Artemenkov, A., Fedyanin, K., Noskov, F., Fishkov, A., Shelmanov, A., Vazhentsev, A., Petiushko, A., and Panov, M. Nonparametric uncertainty quantification for single deterministic neural network. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022. 4

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017. 4

Landau, D. and Binder, K. *A Guide to Monte Carlo Simulations in Statistical Physics*. Cambridge University Press, 2005. 9

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998. 7

Lecun, Y., Chopra, S., Hadsell, R., Ranzato, M., and Huang, F. A tutorial on energy-based learning. *Predicting structured data*, 2006. 4

Lee, K., Lee, K., Lee, H., and Shin, J. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2018. 4

Leibo, J., Mutch, J., and Poggio, T. Learning to discount transformations as the computational goal of visual cortex. *Nature Precedings*, 2011. 2

LeJeune, D., Balestriero, R., Javadi, H., and Baraniuk, R. G. Implicit rugosity regularization via data augmentation. *ArXiv*, 2019. 1

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. Visualizing the loss landscape of neural nets. In *Neural Information Processing Systems (NeurIPS)*, 2018. 5

Lin, C.-H. and Lucey, S. Inverse compositional spatial transformer networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 7

Liu, W., Wang, X., Owens, J., and Li, Y. Energy-based out-of-distribution detection. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020. 4

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. A convnet for the 2020s. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3, 4

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2017. 7

Marcos, D., Volpi, M., Komodakis, N., and Tuia, D. Rotation equivariant vector field networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 3

Moen, E., Bannon, D., Kudo, T., Graf, W., Covert, M., and Valen, D. Deep learning for cellular image analysis. *Nature Methods*, 2019. 7

Mondal, A. K., Panigrahi, S. S., Kaba, S.-O., Rajeswar, S., and Ravanbakhsh, S. Equivariant adaptation of large pretrained models. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023. 2, 1

Moon, J., Kim, J., Shin, Y., and Hwang, S. Confidence-aware learning for deep neural networks. In *International Conference on Machine Learning (ICML)*, 2020. 4

Nash, D. *A Friendly Introduction to Group Theory*. CreateSpace Independent Publishing Platform, 2016. 3

Olah, C., Cammarata, N., Voss, C., Schubert, L., and Goh, G. Naturally occurring equivariance in neural networks. *Distill*, 2020. 2

Osawa, K., Swaroop, S., Jain, A., Eschenhagen, R., Turner, R. E., Yokota, R., and Khan, M. E. Practical deep learning with bayesian principles. *International Conference on Neural Information Processing Systems (NeurIPS)*, 2019. 4

Osband, I. Risk versus uncertainty in deep learning: Bayes, bootstrap and the dangers of dropout. *Conference on Neural Information Processing Systems (NeurIPS)*, 2016. 4

Palmer, S. E., Rosch, E., and Chase, P. Canonical perspective and the perception of objects. *Vision Research*, 1981. 2

Qiu, Q., Cheng, X., Calderbank, R., and Sapiro, G. DCFNet: Deep neural network with decomposed convolutional filters. *International Conference on Machine Learning (ICML)*, 2018. 2

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 2015. 8

Sanbonmatsu, D. M., Posavac, S. S., Kardes, F. R., and Mantel, S. P. Selective hypothesis testing. *Psychonomic Bulletin & Review*, 1998. 6

Schmidt, J. and Stober, S. Learning continuous rotation canonicalization with radial beam sampling. *ArXiv*, 2022. 2

Schmidt, J., Köhler, B., and Borstell, H. Reviving simulated annealing: Lifting its degeneracies for real-time job scheduling. In *Hawaii International Conference on System Sciences (HICSS)*, 2024. 9

Shakerinava, M., Mondal, A. K., and Ravanbakhsh, S. Structuring representations using group invariants. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022. 2

Shorten, C. and Khoshgoftaar, T. M. A survey on image data augmentation for deep learning. *Journal of Big Data*, 2019. 1

Shu, C., Chen, X., Yu, C., and Han, H. A refined spatial transformer network. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2018. 2

Snell, J., Swersky, K., and Zemel, R. Prototypical networks for few-shot learning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017. 4

Stallkamp, J., Schlipsing, M., Salmen, J., and Igel, C. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 2012. 8

Taghizadeh-Mehrjardi, R., Nabiollahi, K., Rasoli, L., Kerry, R., and Scholten, T. Land suitability assessment and agricultural production sustainability using machine learning models. *Agronomy*, 2020. 7

Tai, K. S., Bailis, P., and Valiant, G. Equivariant Transformer Networks. In *International Conference on Machine Learning (ICML)*, 2019. 2, 7

Tan, M. and Le, Q. EfficientNet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning (ICML)*, 2019. 3, 4

Tuggener, L., Stadelmann, T., and Schmidhuber, J. Efficient rotation invariance in deep neural networks through artificial mental rotation. *ArXiv*, 2023. 1

Ustyuzhaninov, I., Cadena, S. A., Froudarakis, E., Fahey, P. G., Walker, E. Y., Cobos, E., Reimer, J., Sinz, F. H., Tolias, A. S., Bethge, M., and Ecker, A. S. Rotation-invariant clustering of neuronal responses in primary visual cortex. In *International Conference on Learning Representations (ICLR)*, 2020. 2

Wang, D., Shelhamer, E., Liu, S., Olshausen, B., and Darrell, T. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations (ICLR)*, 2021. 1

Wang, Q., Fink, O., Van Gool, L., and Dai, D. Continual test-time domain adaptation. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1

Weiler, M. and Cesa, G. General e(2)-equivariant steerable cnns. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019. 2

Weiler, M., Hamprecht, F. A., and Storath, M. Learning steerable filters for rotation equivariant cnns. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

Wu, Y., Chi, Z., Wang, Y., Plataniotis, K. N., and Feng, S. Test-time domain adaptation by learning domain-aware batch normalization. In *Conference on Artificial Intelligence (AAAI)*, 2024. 1

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *ArXiv*, 2017. 5

Zhou, A., Knowles, T., and Finn, C. Meta-learning symmetries by reparameterization. *International Conference on Learning Representations (ICLR)*, 2020. 2

Algorithm 1 Pseudocode for ITS with one hypothesis H.

Input: $\mathbf{x} \sim \mathcal{GD}_{\text{test}}$
$\rho_{\text{primal}} = 0$ and $h = \{e\}$
for G **in** $\{G\}$ **do**
  $\rho, g = \text{argmax}_{g \in G}[\rho_G(\mathbf{x}, g \circ h)]$
  if $\rho > \rho_{\text{primal}}$ **then**
    $\rho_{\text{primal}} \leftarrow \rho$
    $H \leftarrow p_\theta(y \mid g \circ h\mathbf{x})$
    $h \leftarrow g \circ h$
  end if
end for
Output: h

A. Implementation Details

If not otherwise stated in the experiment descriptions, we used a group cardinality of $n = 17$. The choice of odd numbers is due to our algorithmic implementation. We divide the parameter space into n equally distant chunks. An odd n ensures that the identity is included.

A.1. Search Algorithm

Supplementary to the descriptions and illustrations in Figure 4 and Figure 5, we provide the pseudocode of our proposed algorithm in Algorithm 1. The complexity of the search increases if more than one hypothesis is maintained. We refer the reader to our publicly available source code for more details.[4]

A.2. Model Architectures

Small-Sized CNN This is used for experiments on MNIST and Fashion-MNIST. This model comprises three 2D-convolutional layers with 5×5 kernels and 8, 16, and 16 channels. We add GeLU non-linearities (Hendrycks & Gimpel, 2016) after each convolution layer. 2D Max-Pooling with 2×2 kernels is added after the second and the third convolutional layer. The final stack of feature maps is flattened into a vector and further compressed to the number of classes by two linear layers. After the first linear layer, we also add a GeLU non-linearity and a dropout-layer with 50% of dropout. From here we read out the logit scores.

Large-Sized CNN This is used for experiments on GTSRB. This model comprises three 2D-convolutional layers with kernel sizes of 5, 3 and 1 and 100, 150, and 250 channels. We add GeLU non-linearities (Hendrycks & Gimpel, 2016), 2D Max-Pooling with 2×2 kernels, Batch Normalisations (Ioffe & Szegedy, 2015) and Dropout after each convolution layer. The final stack of feature maps is flattened into a vector and further compressed to the number of classes by two linear layers. After the first linear layer, we also add a GeLU non-linearity and a dropout-layer with 50% of dropout. From here we read out the logit scores.

[4] www.github.com/johSchm/ITS

B. Further Related Works

B.1. Test-time Domain Adaptation

In our work, we model a canonicalisation function by a search process. Canonicalisation (Kaba et al., 2023; Mondal et al., 2023) is build on first principles and supported by the mathematical guarantees of group theory. This is highly distinctive to learnable test-time adaptions (TTAs) (Wang et al., 2021; 2022; Wu et al., 2024). We do not need to modify the parameters of the neural network, we can even treat the model as a black box only having access to the logit-scores. Source-free TTAs often suffer from error accumulations and catastrophic forgetting (Wang et al., 2022). For our method, the latter is avoided a priori, as at no point the neural network and its weights are modified. Similar to (Wang et al., 2022), we use weight averages (approx. Bayesian) to mitigate error accumulations.

C. Further Experiments

Remark regarding Image-Backgrounds Spatial transformations can introduce background artefacts for datasets without mono-colour backgrounds, like ImageNet. There exist only solutions for a few quotient groups (like rotations (Tuggener et al., 2023)) not for the entire affine group (as considered in this work). Fortunately, this is only an issue during training as the model might exploit these artefacts, which does not apply for any conducted experiment. Either all models are trained on the vanilla dataset (as in Section 5.1 and Section 5.2) or are trained with artefacts (baselines in Section 5.3) but tested on an artefact-free dataset (here SiScore).

C.1. Runtime

All experiments are performed on an Nvidia A40 GPU (48GB) node with 1 TB RAM, 2x 24core AMD EPYC 74F3 CPU @ 3.20GHz, and a local SSD (NVMe). The software specifications of our implementations can be found in our open-sourced code. Figure 10 shows how the runtime behaves under different hyperparameter settings compared to the pure backbone. The most significant factor is the depth of the search tree. In Section 5.2 we found that the performance does not notably increase with more hypotheses. The number of Monte-Carlo (MC) dropout steps, stabilises the predictions by reducing temporal confidence anomalies. Although, we used 50 MC samples for our experiments, the impact on the performance is minor. Another factor is the number of hypotheses tested. These hypotheses are processed in parallel but due to the unique class condition (see Section 4), the runtime scales with the number of hypotheses. Empirically,

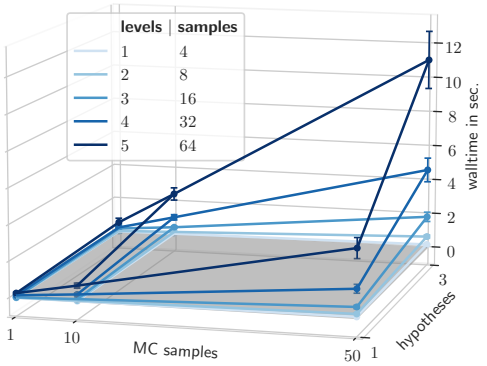

Figure 10. Walltime of ITS (shades of blue) with a CNN backbone (gray) on a transformed MNIST test set (standard deviation shown as error bars).

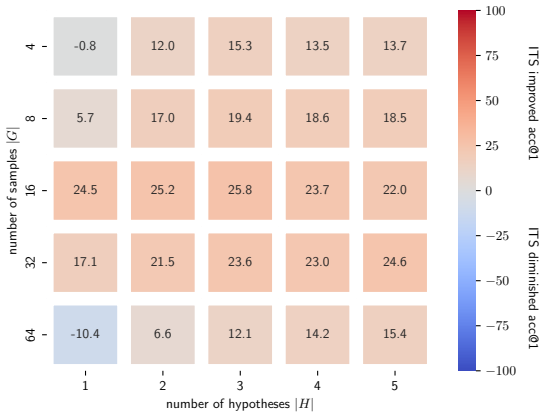

Figure 11. Average acc@1 improvement on the randomly rotated and scaled MNIST test set compared to the baseline (without ITS) performance.

we found that three to five produces the best results.

C.2. Hyperparameter Tuning for Accuracy

We continue the previous investigation by studying the influence of these hyperparameters on the accuracy. Increasing the number of MC samples stabilises predictions but also reduces exploration. When increasing the MC samples from 1 to 10 on affine-transformed Fashion-MNIST the acc@1 increases 9.46 ± 1.14 points (mean and standard deviation over 10 runs). Increasing the MC samples from 10 to 50 leads to an acc@1 increase of only 1.38 ± 0.47.

In Figure 11, we illustrated the performance improvements with respect to the number of samples per layer and the number of hypotheses. Interestingly, the performance does not increase (or stagnate) when increasing either value. It seems that the middle ground around $|H| = 3$ and $|G| = 16$ results in maximum improvements compared to the vanilla model (without ITS) prediction. Due to the unique-label constraint of each hypothesis (see Section 4) many hypotheses with initial low confidence are pursued. This might lead to catastrophic reinforcement of many false hypotheses, obtaining higher confidences down the line.

C.3. Change of Mind

To compute the results in Figure 6, we used the following measures: We estimated the transformation distance between the predicted matrix, $\mathbf{T}_{H_0}$ or $\mathbf{T}_{H_1}$, and the ground-truth matrix, $\mathbf{T}_{\text{GT}}$. Let $d(\mathbf{T}_{H_i}, \mathbf{T}_{\text{GT}}) := \|\mathbf{T}_{H_i} - \mathbf{T}_{\text{GT}}\|_F$ for $i \in \{0,1\}$. To ease comparability, we used a cross-normalisation to relate the outcomes. Let $\mathcal{T}_{G\mathcal{D}_{\text{test}}}[H_i]$ be the set of predicted transformation matrices of H_i for all samples in $G\mathcal{D}_{\text{test}}$. Then, the minimum and maximum distance can be obtained by

$$d_{\text{min}} := \min_{\mathbf{T} \in \mathcal{T}_{G\mathcal{D}_{\text{test}}}[H_0] \cup \mathcal{T}_{G\mathcal{D}_{\text{test}}}[H_1]} d(\mathbf{T}, \mathbf{T}_{\text{GT}}), \tag{9}$$

$$d_{\text{max}} := \max_{\mathbf{T} \in \mathcal{T}_{G\mathcal{D}_{\text{test}}}[H_0] \cup \mathcal{T}_{G\mathcal{D}_{\text{test}}}[H_1]} d(\mathbf{T}, \mathbf{T}_{\text{GT}}). \tag{10}$$

We can min-max cross-normalise each distance by

$$\bar{d}(\mathbf{T}_{H_i}, \mathbf{T}_{\text{GT}}) = \frac{d(\mathbf{T}_{H_i}, \mathbf{T}_{\text{GT}}) - d_{\text{min}}}{d_{\text{max}} - d_{\text{min}}} \in [0,1]. \tag{11}$$

We are interested in the difference between both predictions,

$$\bar{d}(\mathbf{T}_{H_0}, \mathbf{T}_{\text{GT}}) - \bar{d}(\mathbf{T}_{H_1}, \mathbf{T}_{\text{GT}}). \tag{12}$$

This is the measure used to produce each point in Figure 6 *(a,c)*. The construction of the classification accuracy plots in Figure 6 *(b,d)* is done similarly. We counted the correct top-1 class-predictions of H_0 and H_1 over $G\mathcal{D}_{\text{test}}$. Again, cross-normalised the results and computed the difference.
The confidence score differences and the class support differences are only plotted for the last layer. This is done as the re-ranking of the hypotheses is only required at the end of the search procedure for the final read-out of the incumbent. Both measures are computed similarly to the previous two. The confidence scores and the class support counts are compared by relational operators and the result is stored as binary matrices. We sum their entries, normalise and compute the difference as before.
The idea of the class support is to identify the hypothesis where the associated class label appears most often during the search. Therefore, most transformed variations fall into one class cluster within the representation space. We argue that this might be evidence for the correctness of the prediction. Also, an equivariant model would maximise this measure and assign all samples to one class.

C.4. Continuation of Section 5.3

Table 3 shows the continuation of our experimental results presented in Table 2. On SI-Size and SI-Location ITS is not able to push the performance of the baseline ViT.

Test Set Samples	ViT	acc@1	acc@5
	vanilla	54.9	77.5
	Rotation + Scale Augm.	45.3 (-9.6)	67.9 (-9.6)
	(Chefer et al., 2022)	60.2 (+5.3)	80.6 (+3.1)
	(Chefer et al., 2022)+AS	61.0 (+6.1)	81.4 (+3.9)
	ITS	55.1 (+0.2)	77.7 (+0.2)
			72.3 (-5.2)
	vanilla	33.3	52.2
	(Chefer et al., 2022)	38.6 (+5.3)	57.0 (+4.8)
	(Chefer et al., 2022)+AS	38.6 (+5.3)	57.8 (+5.6)
	ITS	29.7 (-3.6)	45.0 (-7.2)
			43.6 (-8.6)

Table 3. Continuation of Table 2 with SI-Size and SI-Location.

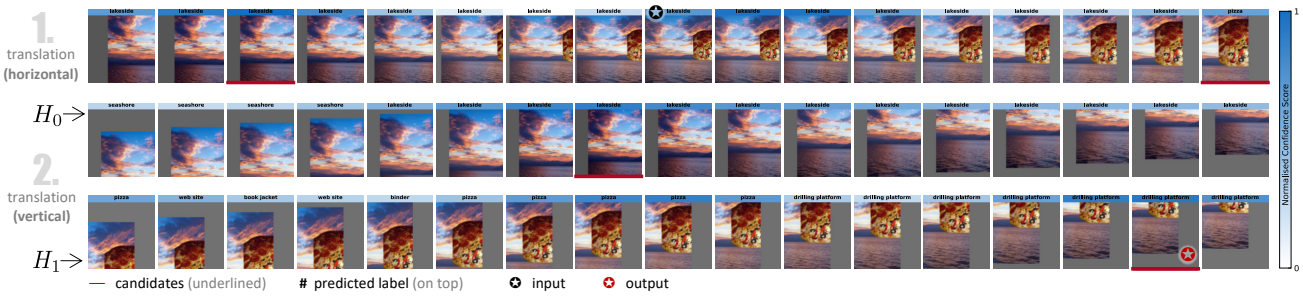

Figure 12. SI-Location sample showing a pizza (ground-truth label) piece on a lakeside background. In the first layer, H_0 classified the image as a lakeside and H_1 as a pizza. In the second layer, H_0 got refined but H_1 changed its hypothesis to now being a drilling platform.

We found no improvement on both test sets. One driving factor is the ability of ITS to focus on different parts of the image *if* translation is used in at least one of the layers. This "issue" is reinforced when another layer can scale the signal. This creates a magnifying glass effect on objects, that the backbone networks predicts are the target objects. As ImageNet classes are diverse, features might appear in the background leading the network to focus here. In Figure 12 we provide an example of a focus failure.

Furthermore, we tested other popular backbone architectures, covering a EfficientNet (Tan & Le, 2019) and a ConvNeXt (Liu et al., 2022). Both models were pretrained on ImageNet-1K. The results are shown in Table 4.

D. Theoretical Specifications

D.1. Transformation Decomposition

Group representations of $\mathrm{Aff}_2(\mathbb{R})$ are defined as 3×3 matrices with real entries (Hall, 2015). These matrices can be decomposed into a centre-rotation, isotropic scaling, reflection, shearing and translation matrix of the same size. We use these basic (or atomic) transformation sub-groups to define the orbits of each search level.

Centre-rotations and isotropic scaling can be represented by a single parameter. Therefore, these sub-groups can be generated by a single generator. For the remaining sub-groups, two generators are required. Reflection is either done about the horizontal or vertical centre axis. Shearing and translation are further decomposed as follows: Either of these transformations is performed along and against the direction of some vector positions at the centre.

D.2. Convolution over Finite Orbits

With this decomposition, convolution over the orbit of any such sub-group can be defined. This is essential for our group-induced confidence measure in Equation (7).

Lemma D.1 (Convolution over Finite Orbits)**.** *Convolving some function f with the orbit, $\mathrm{orb}_G(\mathbf{x})$, can be expressed by*

$$(\mathrm{orb}_G(\mathbf{x})[n] \star f)(g) = \sum_m \mathrm{orb}_G(\mathbf{x})[m] f((\hat{g}^n)^{-1} \hat{g}^m).$$

The proof can be found in Appendix E.1.

	model	acc@1	acc@5
SI-Rotation	EfficientNet	25.1	43.0
	EfficientNet + ITS	34.4 (+9.3)	47.9 (+4.9)
	ConvNeXt	42.4	65.3
	ConvNeXt + ITS	54.7 (+12.3)	70.0 (+4.7)
SI-Size	EfficientNet	35.7	57.3
	EfficientNet + ITS	36.9 (+1.2)	50.5 (-6.8)
	ConvNeXt	53.2	75.8
	ConvNeXt + ITS	56.4 (+3.2)	68.3 (-7.5)
SI-Location	EfficientNet	15.7	30.3
	EfficientNet + ITS	11.4 (-4.3)	17.9 (-12.4)
	ConvNeXt	32.1	58.6
	ConvNeXt + ITS	31.6 (-0.5)	51.5 (-7.1)

Table 4. Continuation of the SiScore experiment with an EfficientNet-B0 (Tan & Le, 2019) and a ConvNeXt-tiny (Liu et al., 2022) backbone.

D.3. Markov Interpretation

Each branching candidate g^{-1} in the search process is a result of Equation (3). All of the aforementioned measures (see Equations (4), (6) and (7)) are stochastic when simulating Bayesian inference (as we do using Monte-Carlo dropout). The evaluations are therefore based on a random variable, which allows us to formulate the process of traversing the search tree as a stochastic process.

More specifically, the search describes a Markov process. The Markov property of a stochastic process guarantees that the probability of a future state depends only on the current state independently of its past behaviour. This implies that the branching candidate in each level only depends on the previous candidate. This is true, as every roll-out depends only on the prior candidate in the first place. These one-step dynamics simplify the system. In Appendix E.4 we provide a sketch of the proof.

E. Proofs

E.1. Proof: Lemma D.1

Proof. Given some kernel K over the orbit of some finite group, G. Let $\mathrm{orb}_G(\mathbf{x})$ be the orbit of $\mathbf{x}$ under the action of G. Convolving orb_G with K is per definition given by

$$(\mathrm{orb}_G(\mathbf{x})[g]\star K)(g) := \int_G \mathrm{orb}_G(\mathbf{x})[h]K(g^{-1}h)dh. \quad (13)$$

As G is a finite group, we can simplify the integration by the Riemann summation over G,

$$(\mathrm{orb}_G(\mathbf{x})[g]\star K)(g) = \sum_{h\in G} \mathrm{orb}_G(\mathbf{x})[h]K(g^{-1}h). \quad (14)$$

This can be rewritten using generators, such that

$$(\mathrm{orb}_G(\mathbf{x})[n]\star K)(g) = \sum_m \mathrm{orb}_G(\mathbf{x})[m]K((\hat{g}^n)^{-1}\hat{g}^m). \quad (15)$$

It is to be shown that $\mathrm{orb}_G(\mathbf{x})$ can be evaluated by $n \in \mathbb{N}$ instead of g. This needs to be justified for the single and dual generator case.

Single Generator For cyclic groups, where one generator suffices, every $g \in G$ can be substituted by the generator raised to some power. The orbit map from Def. 2.3 can be rewritten as

$$\begin{aligned}\mathrm{orb}_G(\mathbf{x}) &= \{g\mathbf{x} \mid g \in G\} \\ &= \{\hat{g}^n\mathbf{x} \mid n \in [0,1,\ldots,|G|-1]\}. \end{aligned} \quad (16)$$

where $\hat{g}^n$ denotes the generator raised to power n. This isomorphism between g and n also simplifies the evaluation

$$\mathrm{orb}_G(\mathbf{x})[g] = \mathrm{orb}_G(\mathbf{x})[n]. \quad (17)$$

Dual Generator The second case assumes, that G is non-cyclic and requires two generators, $\hat{g}$ and $\hat{h}$. Let $\hat{g}$ be the generator for the positive direction along some axis and $\hat{h}$ the generator for the negative direction. Hence, elements in G are either generated by $\hat{g}$ or $\hat{h}$. This allows us to define the following generating function

$$\phi(n) = \begin{cases} \hat{g}^n, & \text{if } n \geq 0 \\ \hat{h}^n, & \text{otherwise.} \end{cases} \quad (18)$$

This generalizes the orbit definition to

$$\begin{aligned}\mathrm{orb}_G(\mathbf{x}) &= \{g\mathbf{x} \mid g \in G\} \\ &= \{\phi(n)\mathbf{x} \mid n \in [0,1,\ldots,|G|-1]\}. \end{aligned} \quad (19)$$

□

To further extend this idea, we can also let any function $f(\mathbf{x}) : \mathrm{orb}_G(\mathbf{x}) \to \mathbb{R}$ act on the orbit. We can define f to be an Energy function, $\bar{E}_\theta(\mathbf{x})$. This forms the foundation for our group-induced confidence measure, $\rho_G(\mathbf{x}, g)$, in Equation (7).

E.2. Proof: Theorem 3.1

Proof. Let $E_\theta(\mathbf{x})$ be the energy of a θ-parameterised energy model given a point $\mathbf{x}$ as input as defined in Equation (4). Let $\bar{E}_\theta(\mathbf{x})$ be the Monte-Carlo estimate of the energy using Equation (5). As shown in Lemma D.1 we can convolve this energy function on the orbit with a kernel, K. To remove visual clutter, we substitute

$$\phi(\mathbf{x},g) := \left(\bar{E}_\theta \star K\right)(g\mathbf{x}). \quad (20)$$

We are interested in the second derivative of $\phi(\mathbf{x},g)$ w.r.t. some $g \in G$. We use the second-order Taylor polynomial and assume sufficiently small remainders. Hence, allowing us to approximate the derivative of ϕ by infinitesimal small steps,

$$\phi(\mathbf{x},g+h) \approx \phi(\mathbf{x},g) + \phi'(\mathbf{x},g)h + \frac{1}{2}\phi''(\mathbf{x},g)h^2 \tag{21}$$

$$\phi(\mathbf{x},g-h) \approx \phi(\mathbf{x},g) - \phi'(\mathbf{x},g)h + \frac{1}{2}\phi''(\mathbf{x},g)h^2. \tag{22}$$

Adding both together and rearranging for ϕ'' gives

$$\phi(\mathbf{x},g+h) + \phi(\mathbf{x},g-h) \approx 2\phi(\mathbf{x},g) + \phi''(\mathbf{x},g)h^2$$
$$\phi''(\mathbf{x},g) \approx \frac{1}{h^2}\Big(\phi(\mathbf{x},g+h) + \phi(\mathbf{x},g-h) - 2\phi(\mathbf{x},g)\Big). \tag{23}$$

This formulation uses incrementally small steps $g+h$ on the orbit of G and $\mathbf{x}$, which we have to define next.
Given a Matrix Lie Sub-Group, H and a finite sub-group, $G \leq H$. We can approximate H by increasing the cardinality of G, such that in the limit $G = H$. Let $\langle g \rangle$ be the generator of G, which forms an atomic element of G. Then, every $g \in G$ can be expressed by $g = \hat{g}^k$ for some k. Using composition, a step along the orbit starting at g can be expressed by $(\hat{g} \circ g)\mathbf{x}$. Similarly, a step in the opposite direction can be performed by $(\hat{g}^{-1} \circ g)\mathbf{x}$. We substitute $g\mathbf{x}+h$ by $(\hat{g} \circ g)\mathbf{x}$ in Equation (23). Let $\mathbf{T}_g$ be the matrix representation of g. The step length along the orbit is defined by $\|\mathbf{T}_{\hat{g}} - \mathbf{T}_{\hat{g}^2}\|$. These two observations lead to the following reformulation of Equation (23),

$$\phi''(\mathbf{x},g) \approx \frac{1}{\|\mathbf{T}_{\hat{g}} - \mathbf{T}_{\hat{g}^2}\|^2} \tag{24}$$
$$\Big(\phi(\mathbf{x},\hat{g} \circ g) + \phi(\mathbf{x},\hat{g}^{-1} \circ g) - 2\phi(\mathbf{x},g)\Big).$$

Remember that G is a finite sub-group of some Matrix Lie Group, H. In the limit, however, when the cardinality of G approaches the one of H, the approximation error will shrink to zero. Therefore, we consider

$$\phi''(\mathbf{x},g) = \lim_{|G| \to \infty} \frac{1}{\|\mathbf{T}_{\hat{g}} - \mathbf{T}_{\hat{g}^2}\|^2} \tag{25}$$
$$\Big(\phi(\mathbf{x},\hat{g} \circ g) + \phi(\mathbf{x},\hat{g}^{-1} \circ g) - 2\phi(\mathbf{x},g)\Big).$$

This concludes the proof. □

E.3. Proof: Theorem 4.1

Proof. Given a finite sequences $s = g_0 \circ g_1 \circ \ldots \circ g_m$ of group elements and a sub-sequence $s_{0:k-1} = g_0 \circ g_1 \circ \ldots \circ g_{k-1}$ with $k \leq m \in \mathbb{N}$. It is to be shown that additional action by g_k^{-1} on s eliminates g_k from s (this sequence is coined $s_{\setminus g_k}$). We prove the statement $g_k^{-1}s = s_{\setminus g_k}$ by successive refinement and ignore the trivial case where $k=m$, $k=0$ or $m=0$. We have

$$g_k^{-1}s = g_k^{-1} \circ (g_0 \circ g_1 \circ \ldots \circ g_m). \tag{26}$$

Due to the associativity axiom of any group, the parentheses can be ignored. Furthermore, s can be substituted by $s_{0:k-1} \circ s_{k:m}$, such that

$$g_k^{-1}s = g_k^{-1} \circ \underbrace{g_0 \circ \ldots \circ g_{k-1}}_{s_{0:k-1}} \circ \underbrace{g_k \circ \ldots \circ g_m}_{s_{k:m}}. \tag{27}$$

If elements $s_{0:k-1}$ are non-trivial, then commutativity is required to move g_k^{-1} in front of g_k to cancel them out. Therefore, we need to assume that $s_{0:k-1}$ and g_k^{-1} are elements of an Abelian group. Then,

$$g_k^{-1}s = s_{0:k-1} \circ g_k^{-1} \circ g_k \circ \ldots \circ g_m, \tag{28}$$
$$g_k^{-1}s = s_{0:k-1} \circ \ldots \circ g_m, \tag{29}$$
$$g_k^{-1}s = s_{\setminus g_k}, \tag{30}$$

which concludes the proof. □

E.4. Proof: Markov Process

We begin by formally defining a stochastic and a Markov process.

Definition E.1 (Stochastic Process (Borodin, 2017))**.** A stochastic process is a family of random variables $\{X_t : t \in \mathcal{T}\}$, where the index t belongs to some arbitrary parameter set, $\mathcal{T}$. Each of the random variables can take the same set of possible values, which are called the *state space*.

Definition E.2 (Markov Process (Borodin, 2017))**.** Given a stochastic process, $\{X_t : t \in \mathcal{T}\}$. We say $\{X_t : t \in \mathcal{T}\}$ has the Markov property if for all t and all states $\{x_i : i \in [0,n]\}$, we have

$$\mathbb{P}(X_{n+1} = x_{n+1} \,|\, \{X_i = x_i : i \in [0,n]\}) = \mathbb{P}(X_{n+1} = x_{n+1} \,|\, X_n = x_n). \tag{31}$$

When traversing the search tree, a composition of transformation matrices is formed, $\mathbf{T}_t \circ \mathbf{T}_{t+1} \circ \ldots \mathbf{T}_T$. These matrices form the state space. At each state, the group-induced confidence measure, $\rho_G(\mathbf{x},g)$, is used to determine the next most probable action. To reduce visual clutter, we define $g_t := g \in G_t$, i.e., some group element from the group at level t. Given a finite set of actions, we use the Boltzmann distribution (softmax) to formulate state-action probabilities,

$$\mathbb{P}(\mathbf{T}_t \,|\, \hat{\mathbf{x}}) = \frac{\exp \rho_{G_t}(\hat{\mathbf{x}},g_t)}{\sum_{g_t} \exp \rho_{G_t}(\hat{\mathbf{x}},g_t)} = \frac{\exp -\frac{\partial^2}{\partial g_t^2}\big(\bar{E}_\theta(g_t\hat{\mathbf{x}}) \star K\big)}{\sum_{g_t} \exp -\frac{\partial^2}{\partial g_t^2}\big(\bar{E}_\theta(g_t\hat{\mathbf{x}}) \star K\big)}, \tag{32}$$

where $\hat{\mathbf{x}}$ represents $\mathbf{x}$ after all previous group elements on the trajectory acted on it, i.e., $\hat{\mathbf{x}} := (g_{t-1} \circ g_{t-2} \circ \ldots \circ g_0)\mathbf{x}$. The mapping from $\mathbf{x}$ to $\hat{\mathbf{x}}$ is surjective. Trivially, the identity can be injected at all positions without changing $\hat{\mathbf{x}}$. The same is

true for (non-trivial) equivariance classes. As long as $\hat{\mathbf{x}}$ stays the same, the conditioning $\mathbb{P}(\mathbf{T}_{g_t} \mid \hat{\mathbf{x}})$ will not be affected by the choice of transformation sequence. Due to this finding, we have

$$\mathbb{P}(\mathbf{T}_t \mid \hat{\mathbf{x}}) = \mathbb{P}(\mathbf{T}_t \mid \mathbf{T}_{t-1}). \tag{33}$$

This conditioning fulfils the definition of a Markov property. Next, we complete this proof by showing that we have a stochastic process. $\bar{E}_\theta(\cdot)$ is the expected energy obtained using approximate Bayesian inference,

$$\mathbb{E}_{\mathbf{\Lambda}}[E_\theta(\mathbf{x})] := \bar{E}_\theta(\mathbf{x}) = \int z \mathbb{P}_\theta(z \mid \mathbf{x}, \mathbf{\Lambda}) dz, \tag{34}$$

where z is a latent scalar for the total energy. As defined in the main corpus of this work, we use Monte-Carlo Dropout to simulate multiple stochastic forward passes, where $\Lambda := \mathrm{diag}(\boldsymbol{\lambda}) \sim \text{Bernoulli}(\lambda \in [0,1])$. So, the measure itself is a random variable sampled from a distribution defined by the neural network. We use kernel density estimation with Gaussian kernels to estimate the energy probabilities, $\mathbb{P}_\theta(z \mid \mathbf{x}, \mathbf{\Lambda})$. It is this stochastic measure which renders the process a stochastic process. The transition matrix is formed by $\mathbb{P}(\mathbf{T}_t \mid \mathbf{T}_{t-1})$ for all t. We are only interested in the forward pass, i.e., traversing the tree from root to leaf. As, however, all group actions are reversible per definition (see Def. 2.1), the backward pass is well-defined, too.