MODELING SPATIAL TRANSFORMATIONS INSTEAD OF MEMORIZING TRANSFORMED APPEARANCES

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning spatial transformations from transformed images requires surprisingly large amounts of data and model capacity. We study this effect using controlled affine transformations of MNIST and examine how it changes with dataset size, model capacity, and generalization to unseen appearance and transformation combinations. We evaluate several existing classifier families, including Vision Transformers, ConvNeXts, Spatial Transformer Networks, and group-equivariant CNNs, spanning substantially different spatial inductive biases. We propose a Transform-Parameterized Implicit Neural Representation (TP-INR), which learns appearance while explicitly representing affine transformation structure and inferring its parameters through reconstruction-error minimization. Across architectures, TP-INR preprocessing substantially improves sample efficiency, with the largest gains in low-data regimes. We further find that a ViT can fail to generalize a known transformation distribution to known classes when their combination was not observed during training, while TP-INR partially mitigates this failure. These results suggest that explicitly encoding known transformation structure can make recognition more data- and parameter-efficient.

1 INTRODUCTION

Modern recognition systems achieve impressive robustness to spatial variation at substantial cost in data, parameters, and computation. Large models can learn to recognize objects across changes in position, rotation, scale, and shear by observing sufficiently many examples of those variations; foundation models push this strategy to an extreme through large-scale pretraining (Bommasani et al., 2021). This raises a basic question: when an imaged object is changed by a simple camera transformation, can a model learn to generalize that camera transformation independent of its appearance, or will it just memorize all joint appearance-transformation combinations?

Affine spatial transformations provide a clean setting in which to study this question. A translated, rotated, scaled, or sheared image can differ substantially in pixel space while preserving object identity. Yet a generic recognizer is not told that these observations are related by a transformation; it must infer from data which variations should be treated as equivalent. With sufficient coverage this can produce invariance, but it leaves open whether the model has learned a reusable notion of transformation or primarily memorized all transformed appearances encountered during training.

This motivates a different parameterization of the problem. Rather than representing transformation only implicitly inside a high-dimensional neural network, we can make transformation an explicit variable of the model that can fit a transformation instead of memorizing all of them. We propose a Transform-Parameterized Implicit Neural Representation (TP-INR), inspired by ideas from appearance models and analysis-by-synthesis (Cootes et al., 1998; Chen et al., 2020). TP-INR learns a continuous model of appearance which is fitted by an implicit neural representation (Sitzmann et al., 2020b; Dupont et al., 2022; Bauer et al., 2023), while a predefined family of spatial transformations acts directly on its coordinates. Translation, rotation, scale, and shear are therefore available by construction rather than memorized from transformed examples.

At inference time, TP-INR jointly optimizes a fit on the appearance representation and affine transformation parameters to reconstruct each observation. The transformation is inferred directly through reconstruction error rather than predicted by a separate localization network, and no class

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

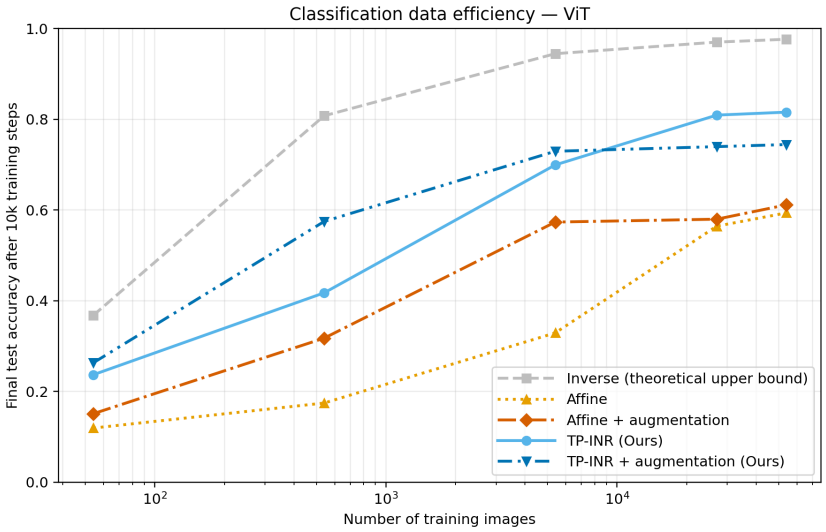


Figure 1: **Data efficiency under affine transformations.** Test accuracy of a ViT trained on increasing amounts of affine-transformed MNIST data. TP-INR preprocessing substantially improves recognition in the low-data regime, both with and without data augmentation.

label is required. The central distinction is therefore simple: *appearance is learned, transformation structure is specified, and their combination is inferred for each data sample*.

We study this idea using affine-transformed MNIST (Deng, 2012). We examine how recognition under spatial variation depends on training-set size, model capacity, and generalization to unseen appearance–transformation combinations. We evaluate Vision Transformers (Dosovitskiy et al., 2020), ConvNeXts (Liu et al., 2022), Spatial Transformer Networks (Jaderberg et al., 2015), and group-equivariant convolutional networks (Cohen & Welling, 2016a;b), allowing us to test whether the cost of learning from transformed appearances persists across substantially different architectural biases. As illustrated in Figure 1, TP-INR preprocessing can substantially improve data efficiency.

More broadly, we ask whether known structure can reduce the burden otherwise placed on data and model capacity. TP-INR provides one instance of this principle: learn the space of appearances, encode the space of transformations, and use inference optimization to determine how they combine for each observation. Our main contributions are:

- We demonstrate that spatially transformed images incur a significant learning cost, as a function of dataset size, model capacity, and generalization across appearance–transformation combinations.
- We introduce Transform-Parameterized Implicit Neural Representations (TP-INR), which learn appearance while exposing a fixed family of spatial transformations at inference.
- We show across multiple recognition architectures that TP-INR preprocessing substantially improves data efficiency, including for models with existing spatial inductive biases.

2 Related Work

Appearance and geometry. Separating appearance from spatial variation has a long history in visual representation learning. Convolutional sparse coding learns reusable appearance primitives while handling translation structurally through convolution (Bristow et al., 2013). Active Appearance Models (AAMs) (Cootes et al., 1998) learn statistical models of shape and appearance and iteratively fit these models to new observations. More recent generative approaches explicitly factor appearance from deformation, geometry, or perspective (Shu et al., 2018; Xing et al., 2020; Skafte & Hauberg, 2019). These works establish appearance and geometry as useful separate factors of a visual model.

Learning and encoding spatial transformations. Spatial Transformer Networks (STNs) (Jaderberg et al., 2015; Lin & Lucey, 2017) introduce differentiable spatial manipulation into recognition networks by learning input-dependent transformation parameters, while Deformable Convolutional Networks (Dai et al., 2017; Zhu et al., 2019) similarly learn input-dependent sampling locations. In contrast, group-equivariant networks encode predefined transformation structure directly into the architecture (Cohen & Welling, 2016a;b), and related work extends geometric structure to continuous neural fields (Wessels et al., 2024). TP-INR lies between these approaches: the transformation family is specified rather than learned, but the transformation for a particular input is not represented throughout the recognition network or predicted by a feed-forward localization model. Instead, its parameters are optimized at test time according to reconstruction error.

Neural fields for image registration. Neural fields have also been optimized directly for geometric alignment. In deformable image registration, methods such as IDIR (Wolterink et al., 2022) and NIR (Sun et al., 2024) represent transformations as coordinate-based neural fields and optimize them to align observed images. The inference problem in TP-INR differs in an important respect: in image registration, both appearances to be aligned are observed and only the transformation between them is unknown. For a new TP-INR input, neither the underlying appearance nor its transformation is given. Starting from a meta-learned appearance initialization, TP-INR therefore adapts a neural field while simultaneously optimizing a predefined spatial transformation to explain the observation.

Implicit neural representations and meta-learned fitting. Implicit neural representations (INRs) is a type of neural field and parameterizes signals as continuous coordinate-based neural functions (Sitzmann et al., 2020b; Mildenhall et al., 2021). SIREN (Sitzmann et al., 2020b), in particular, uses periodic activations to represent images and other continuous signals with high spatial fidelity. Gradient-based meta-learning provides a way to learn network parameters that can rapidly adapt to new observations: MAML (Finn et al., 2017) introduced optimization of an initialization for fast adaptation, while Reptile (Nichol et al., 2018) provides a first-order alternative. This idea has been applied directly to coordinate-based representations. MetaSDF (Sitzmann et al., 2020a) learns an initialization of an implicit representation that can adapt to new shapes in a small number of gradient steps, while Learned Initializations (Tancik et al., 2021) extends this approach across images, shapes, and scenes. More recently, Gielisse & van Gemert (2025) optimize SIREN initialization and adaptation for downstream classification. TP-INR builds on this optimization-based view of INRs, but jointly adapts the appearance INR with explicit spatial transformation parameters.

Canonicalization and test-time transformation search. Canonicalization addresses spatial variation by mapping different transformed versions of the same input to a common representative. Kaba et al. (Kaba et al., 2023) develop learned canonicalization functions and additionally consider an optimization-based formulation in which a learned energy function determines which transformed version of an input should be preferred. More recently, FoCal (Singhal et al., 2025) searches over transformations at test time and selects the view that is most likely under a pretrained foundation-model prior.

TP-INR is closely related in performing transformation search at inference time, but differs in the criterion that drives this search. Existing canonicalization approaches use a learned function or pretrained model to assign preference to candidate transformations. TP-INR introduces no learned canonicalization objective: the transformation parameters are optimized directly using a fixed pixel-level reconstruction loss. Starting from a meta-learned INR initialization, the appearance model and transformation parameters are jointly adapted until their composition reconstructs the observed image. Thus, the learned component provides a prior over plausible appearances, while the criterion for selecting a transformation is fixed by the reconstruction objective rather than learned from data.

3 METHOD

We introduce Transform-Parameterized Implicit Neural Representations (TP-INR), which model appearance with an implicit neural representation (INR) while representing geometric variation explicitly through an affine transformation. The central idea is to make appearance adaptation costly while making spatial transformation directly accessible: the INR learns how objects look, while affine variation can be explained through the coordinates at which the INR is queried.

3.1 TRANSFORM-PARAMETERIZED INR

Let $I : \Omega \to \mathbb{R}$, where $\mathbf{x} = (x, y)^\top \in \Omega \subset [-1, 1]^2$ denote an image, and let $f_{\boldsymbol{\theta}}(\mathbf{x})$ be a SIREN (Sitzmann et al., 2020b) with parameters $\boldsymbol{\theta}$. We represent an affine transformation by six parameters

$$\mathbf{A}_{\boldsymbol{\phi}} = \begin{bmatrix} a_{00} & a_{01} & t_x \\ a_{10} & a_{11} & t_y \end{bmatrix}, \qquad \boldsymbol{\phi} = (a_{00}, a_{01}, t_x, a_{10}, a_{11}, t_y), \tag{1}$$

initialized to the identity transformation.

Because an INR operates directly on coordinates, the affine can act on its input rather than on a rasterized image. Writing $\mathcal{T}_{\boldsymbol{\phi}}(\mathbf{x})$ for the affine-transformed coordinate, TP-INR reconstructs an observation as

$$\widehat{I}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}) = f_{\boldsymbol{\theta}}\left(\mathcal{T}_{\boldsymbol{\phi}}(\mathbf{x})\right). \tag{2}$$

Thus, during fitting, spatial transformations are applied by changing (x, y) before querying the INR; no intermediate image needs to be warped or interpolated. We use the convention that $\mathcal{T}_{\boldsymbol{\phi}}$ maps coordinates in the observed image to coordinates in the appearance INR.

3.2 ANCHORED COARSE-TO-FINE FITTING

Simply optimizing both $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ under pixel reconstruction is insufficient: a sufficiently flexible INR can absorb spatial variation into its weights instead of assigning it to the affine transformation. We therefore anchor each image-specific INR to the shared initialization $\boldsymbol{\theta}_0$ and penalize appearance changes:

$$\mathcal{L}_i^{(k)} = \frac{1}{|\Omega|} \sum_{\mathbf{x}\in\Omega} \left[f_{\boldsymbol{\theta}_i^{(k)}}\left(\mathcal{T}_{\boldsymbol{\phi}_i^{(k)}}(\mathbf{x})\right) - I_{i,\sigma_k}(\mathbf{x}) \right]^2 + \lambda \left\| \boldsymbol{\theta}_i^{(k)} - \mathrm{sg}(\boldsymbol{\theta}_0) \right\|_2^2, \tag{3}$$

where $\mathrm{sg}(\cdot)$ denotes stop-gradient and the norm is summed over all SIREN parameters. We use $\lambda = 1.0$. This proximal term encourages the appearance representation to change only when necessary, making an affine explanation preferable when the discrepancy can be explained geometrically.

Direct pixel alignment provides highly local gradients: when the initial misalignment between the input image and the appearance initialization is large, the corresponding image content may have little pixelwise overlap, providing weak gradients for the affine parameters. We therefore use a coarse-to-fine Gaussian target,

$$I_{i,\sigma} = \mathcal{N}\left(G_\sigma * I_i\right), \tag{4}$$

where G_σ is a Gaussian kernel and $\mathcal{N}$ denotes per-image min–max normalization for $\sigma > 0$. The final target uses the original image directly. Our inner loop uses

$$\sigma_k : \quad \underbrace{8, \ldots, 8}_{10} \to \underbrace{4, \ldots, 4}_{10} \to \underbrace{2, \ldots, 2}_{10} \to \underbrace{0, \ldots, 0}_{10}, \tag{5}$$

for a total of $K = 40$ updates. Large initial blur provides a broader alignment signal, after which progressively sharper targets refine the fit.

We further bias the inner optimization toward explaining geometric variation before adapting appearance. The affine learning rate starts high and gradually decays over the inner loop, while the SIREN learning rate follows the complementary schedule, starting near zero and gradually increasing. We use smooth tanh schedules for both. This lets the affine parameters make large alignment updates early, while allowing progressively more appearance adaptation once the image and the appearance template are better aligned.

Each inner step performs

$$\boldsymbol{\theta}_i^{(k+1)} = \boldsymbol{\theta}_i^{(k)} - \eta_\theta^{(k)} \nabla_{\boldsymbol{\theta}} \mathcal{L}_i^{(k)}, \tag{6}$$

$$\boldsymbol{\phi}_i^{(k+1)} = \boldsymbol{\phi}_i^{(k)} - \eta_\phi^{(k)} \nabla_{\boldsymbol{\phi}} \mathcal{L}_i^{(k)}. \tag{7}$$

Together, the proximal anchor, blurred-target curriculum, and complementary learning rates bias the optimization toward explaining spatial discrepancies through $\boldsymbol{\phi}$ before modifying the appearance model.

3.3 META-LEARNING THE APPEARANCE INITIALIZATION

We learn the shared appearance initialization using Reptile (Nichol et al., 2018). Each training image is treated as a separate adaptation task. For image I_i, we initialize

$$\boldsymbol{\theta}_i^{(0)} = \boldsymbol{\theta}_0, \qquad \boldsymbol{\phi}_i^{(0)} = \boldsymbol{\phi}_{\mathrm{id}}, \tag{8}$$

and apply the K anchored coarse-to-fine updates above.

After adapting a meta-batch $\mathcal{B}$, the shared initialization is updated using the Reptile pseudo-gradient

$$\mathbf{g}_{\mathrm{Rep}} = \boldsymbol{\theta}_0 - \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \boldsymbol{\theta}_i^{(K)}. \tag{9}$$

We apply this pseudo-gradient using Adam with learning rate 10^{-4}. Only $\boldsymbol{\theta}_0$ is meta-learned; the fitted affine parameters are discarded after each task.

Importantly, Reptile operates on the *complete SIREN parameter vector*. TP-INR therefore does not learn a latent code that modulates a fixed decoder: it learns an initialization from which the entire appearance function can be adapted while remaining anchored to shared appearance structure.

3.4 INFERENCE AND TP-INR REPRESENTATION

For a new observation $I^\star$, we initialize a fresh copy of the SIREN at the learned $\boldsymbol{\theta}_0^\star$ and the affine at identity, and repeat the same anchored coarse-to-fine fitting procedure:

$$(\boldsymbol{\theta}_I^\star, \boldsymbol{\phi}_I^\star) \approx \arg\min_{\boldsymbol{\theta}, \boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; I^\star). \tag{10}$$

The shared initialization is never updated during this process, and no class label or transformation annotation is used. Unlike a localization network, TP-INR does not learn a direct mapping $I^\star \mapsto \boldsymbol{\phi}$; the affine is inferred by evaluating how well it explains the observation under the learned appearance model.

After fitting, we use the recovered affine to map the *original observation* into the inferred appearance reference frame:

$$I_{\mathrm{TP}}(\mathbf{x}) = I^\star\left(\mathcal{T}_{\boldsymbol{\phi}_I^\star}^{-1}(\mathbf{x})\right). \tag{11}$$

This final conversion uses bilinear sampling. Importantly, resampling occurs only after the affine has been inferred; during optimization, the affine acts directly on the continuous INR coordinates as in Eq. 2.

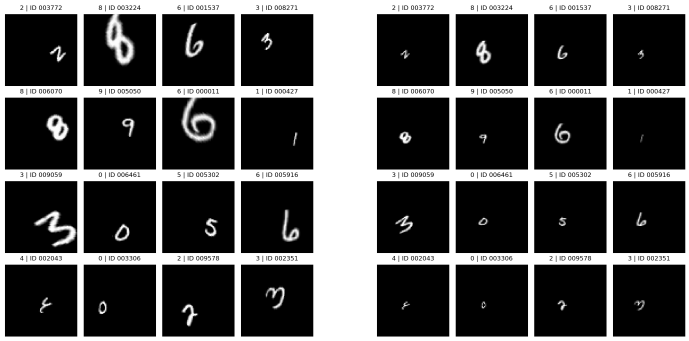
TP-INR therefore separates three roles: the appearance initialization is *learned*, affine structure is *specified*, and the appearance–transformation combination is *inferred* independently for each observation.

4 EXPERIMENTS

We evaluate TP-INR in three ways. First, we study how the amount of labeled training data affects recognition under spatial transformations, and whether TP-INR improves data efficiency across different recognition architectures. Second, we vary model capacity to test whether increasing the size of the recognizer can compensate for learning transformation invariance from data. Third, we study compositional generalization, asking whether a model can combine a known appearance with a known transformation when that particular appearance–transformation pairing was never observed during training.

4.1 IMPLEMENTATION DETAILS

We center each 28×28 MNIST image on a 64×64 canvas and apply a randomly sampled affine transformation consisting of translation, rotation, isotropic scaling, and shear. We sample rotations up to $\pm 45^\circ$, scales in $[0.5, 2.0]$, shear up to $\pm 15^\circ$, and translations up to ± 16 pixels. In addition to the transformed *affine* images, we construct an *inverse* condition by applying the exact inverse of the



(a) Affine inputs. (b) TP-INR outputs.

Figure 2: **Paired inputs and outputs of TP-INR preprocessing.** Sixteen randomly sampled affine images (left) and their corresponding TP-INR-processed images (right). Matching positions show the same example before and after preprocessing.

sampled transformation to the transformed image. This provides a reference for recognition when the applied transformation is known.

We use $54{,}000$ images from the MNIST training set and construct nested training subsets containing 54, 540, $5{,}400$, $27{,}000$, and $54{,}000$ examples. All models are evaluated on the full $10{,}000$-image MNIST test set. Recognition models are trained for a fixed number of optimization steps and evaluated once on the test set.

For TP-INR, a separate Reptile initialization is trained for each training-set size using only the corresponding affine training subset. The same subset-specific initialization is then used to fit the matching training and test images independently. Test-time fitting does not update the globally shared initialization and uses no class labels.

For recognition, we use AdamW with learning rate 3×10^{-4}, weight decay 0.05, a 250-step warmup followed by cosine decay, and batch size 50. Experiments 1 and 2 train for $10{,}000$ optimization steps. For augmented conditions, we apply an additional online random affine transformation.

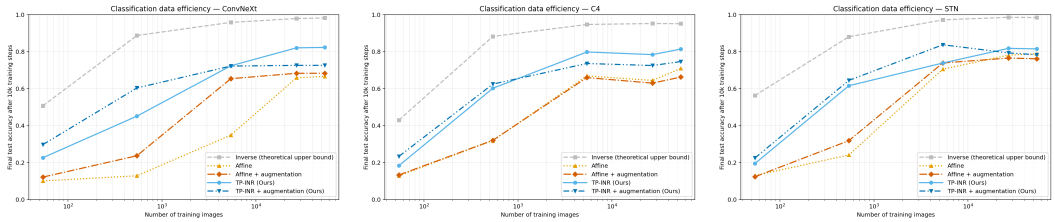
4.2 Experiment 1: Data Efficiency Across Architectures

Our first experiment studies how much labeled transformed data is required for recognition, and whether TP-INR preprocessing reduces this requirement. We compare five conditions across the nested training subsets: *Affine*, *Affine + Augmentation*, *TP-INR*, *TP-INR + Augmentation*, and *Inverse*. For every subset size, the corresponding TP-INR model is itself trained only on that same subset.

To test whether the effect depends on the inductive bias of the downstream recognizer, we repeat the experiment with four architectures: a Vision Transformer (ViT) (Dosovitskiy et al., 2020), a ConvNeXt (Liu et al., 2022), a Spatial Transformer Network (STN) (Jaderberg et al., 2015), and a C_4 rotation-equivariant convolutional network (Cohen & Welling, 2016a). We use compact models with approximately $100{,}000$ trainable parameters each: $109{,}578$ for the ViT, $97{,}322$ for ConvNeXt, $101{,}840$ for the STN, and $89{,}802$ for the C_4 model.

The ViT comparison is shown in Figure 1, with the remaining architectures in Figure 3. Across all four backends, TP-INR consistently improves data efficiency over learning directly from affine-transformed inputs. The gains are largest in the low- and medium-data regimes. For example, with roughly 500 training images, TP-INR improves accuracy from about 18% to 42% for the ViT, from 13% to 45% for ConvNeXt, from 24% to 62% for the STN, and from 32% to 60% for the C_4 model.

The effect remains visible even for architectures that already contain explicit spatial inductive biases. In particular, both the STN and C_4 model benefit substantially from TP-INR at small dataset sizes, suggesting that explicitly resolving affine variation before recognition provides information not fully recovered by these architectures from limited labeled data alone. As the amount of training data



(a) ConvNeXt (b) C_4 equivariant CNN (c) Spatial Transformer Network

Figure 3: **Data efficiency across recognition architectures.** Test accuracy as a function of the number of labeled training examples for ConvNeXt, a C_4 rotation-equivariant CNN, and an STN. Each architecture contains approximately one million trainable parameters and is trained for 10k optimization steps. The corresponding ViT experiment is shown in Figure 1.

increases, the gap generally narrows, especially for STN and C_4, which are increasingly able to learn robustness directly from the transformed examples.

Augmentation is most useful in the low-data regime. TP-INR with augmentation gives the strongest results for several of the smallest subsets, while its advantage diminishes once sufficient training data are available. Across conditions, the inverse representation remains substantially above the learned alternatives, indicating that there is still considerable headroom from more accurate transformation recovery.

4.3 Experiment 2: Model Capacity

Our second experiment studies how VIT (Dosovitskiy et al., 2020) capacity affects learning from affine-transformed data. There are two things we want to test here. First, sufficiently large models may have enough capacity to learn the transformation variability directly from the training data, reducing the need for explicit alignment. Conversely, smaller models may be forced to learn more compact, transformation-invariant representations because they have less capacity to model transformed appearances independently.

To test these possibilities, we hold the amount of training data fixed and vary the size of the ViT over a wide range, from extremely small models to substantially larger vision transformers. Embedding width, depth, and number of attention heads are varied jointly, allowing us to examine how the effect of affine variation changes with model capacity. For each model size, we evaluate the same five input conditions used in Experiment 1. All models are trained for the same $10{,}000$ optimization steps, and performance is compared as a function of the exact number of trainable parameters.

Figure 4 shows a strong dependence on model capacity, but provides little evidence that small models automatically learn affine invariance through a capacity bottleneck. At very small model sizes, performance on affine-transformed inputs remains poor, while TP-INR already provides substantial gains. For example, at roughly 10^4 parameters, TP-INR reaches about 62% accuracy compared with approximately 32% for the affine condition.

Increasing model capacity does make learning directly from affine-transformed data considerably easier. Affine accuracy rises steadily with model size, suggesting that sufficiently expressive models can learn part of the transformation variability from examples. However, this does not eliminate the advantage of explicit alignment: TP-INR remains consistently above the affine conditions across almost the entire capacity range, including once performance begins to saturate for the largest models. Thus, greater capacity reduces, but does not remove, the cost of learning from transformed appearances.

Interestingly, the smallest models do not exhibit improved invariance relative to larger models. Instead, their limited capacity appears to make the affine learning problem harder rather than forcing a transformation-invariant solution. Together, these results suggest that spatial variation can increasingly be absorbed by the recognizer as capacity grows, whereas explicitly representing the transformation substantially reduces the capacity required to achieve a given level of accuracy.

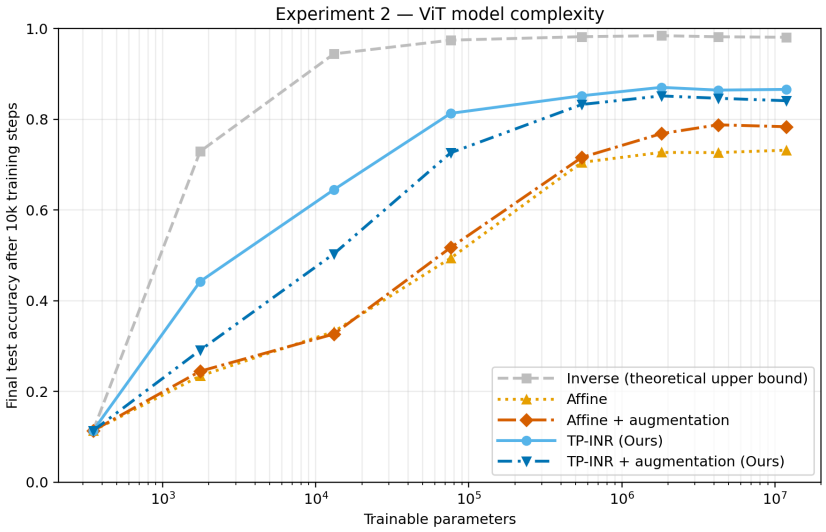


Figure 4: **Effect of recognition-model capacity.** Test accuracy for ViTs of increasing size trained on the same amount of data for 10k optimization steps. Larger models increasingly learn from affine-transformed inputs directly, but TP-INR retains an advantage across model capacities. Small models do not automatically acquire affine invariance through a capacity bottleneck.

4.4 Experiment 3: Compositional Generalization

Our third experiment asks whether a recognizer that has seen a class and has separately seen a transformation can combine the two at inference time. We construct a training set in which *even* digits are presented under affine transformations, while *odd* digits are shown only in the inverse representation. The model therefore observes the affine transformation distribution during training and also learns the appearance of every digit class, but it never sees affine-transformed odd digits. At test time, all digits are affine-transformed.

This setup tests whether the model learns a reusable notion of the transformation itself, or instead relies on class–transformation combinations observed during training. A model that has learned the transformation compositionally should be able to apply its experience with affine variation on even digits to the already-known appearances of odd digits.

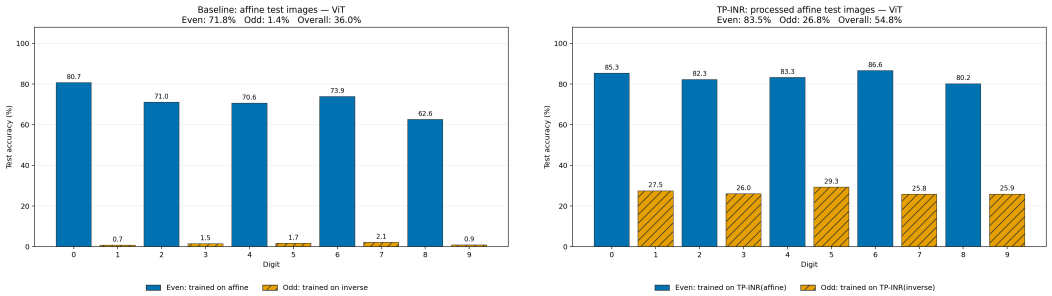
We compare a baseline ViT with an otherwise identical ViT trained on TP-INR-preprocessed images. TP-INR is meta-trained only on the same mixed training set, after which the shared initialization is frozen and each train/test image is fitted independently. Both classifiers have 109,578 trainable parameters and are trained for 20,000 steps.

As shown in Figure 5, the baseline reaches 71.76% accuracy on even digits but only 1.36% on odd digits. Thus, knowing the odd-digit appearances and having observed the same affine transformation family on other classes is not sufficient for the ViT to generalize to their unseen combination. This is consistent with the model learning transformation robustness in a strongly appearance-dependent manner rather than acquiring a reusable representation of the transformation itself.

TP-INR raises odd-digit accuracy to 26.84%. Explicitly separating appearance from transformation therefore improves compositional generalization, although the remaining gap shows that the problem is not fully resolved.

5 Discussion

Our results support the view that a substantial part of the difficulty of recognition under spatial variation comes from how that variation is represented. Across recognition architectures, TP-INR consistently improves data efficiency relative to training directly on affine-transformed images, with



(a) Baseline (b) TP-INR

Figure 5: **Generalization to unseen appearance–transformation combinations.** We test whether a model can combine a known appearance with a known transformation at inference when that pairing was never seen during training.

the largest gains in the low-data regime. This effect is not limited to generic architectures: STNs and C_4 group-equivariant networks also benefit from TP-INR preprocessing, suggesting that architectural spatial bias alone does not eliminate the cost of learning from transformed appearances.

Experiment 2 further shows that model capacity changes this cost, but does not remove it. Larger ViTs become increasingly capable of learning directly from affine-transformed data, yet TP-INR remains advantageous across most of the capacity range. Conversely, very small models do not appear to acquire invariance automatically through a capacity bottleneck; instead, affine variation makes the recognition problem substantially harder for them.

Experiment 3 provides a complementary view. A ViT that has observed both the relevant digit appearances and the affine transformation distribution nevertheless fails almost completely when these two factors are combined in a way not seen during training. This is consistent with transformation robustness being learned in an appearance-dependent manner rather than as a fully reusable transformation concept. TP-INR improves performance on these unseen appearance–transformation combinations, although the remaining gap shows that the solution is not yet perfect. The method should therefore be viewed as introducing a useful inductive bias rather than solving alignment exactly.

There are several limitations. First, our experiments are restricted to MNIST and a known affine transformation family. MNIST is particularly favorable for the proposed fitting procedure: images are simple and typically contain a single object, such that a global affine update can often reduce reconstruction error even when the initial alignment with the learned appearance template is poor. For more complex natural images, where multiple structures, backgrounds, or local deformations contribute to the reconstruction loss, a single global affine transformation may provide a much weaker optimization signal and may require a more structured or hierarchical fitting procedure. Extending TP-INR to richer or nonrigid transformation families would likewise require an appropriate parameterization. Second, inference requires iterative optimization for every image, making TP-INR more computationally expensive at test time than some feed-forward recognition models.

More broadly, our results suggest that spatial invariance should not be treated purely as something a high-dimensional neural network must discover from transformed examples. When the relevant transformation structure is known, we believe it should be made explicit in the model rather than left to emerge implicitly in learned weights and activations. At the same time, explicitly representing a transformation should not mean training another network to predict it directly from the image. TP-INR instead treats the transformation as an inference variable and determines it through optimization against the observation itself. We believe this separation is a promising direction: learn appearance from data, represent known transformations explicitly, and infer through test-time optimization how they combine for each observation.

REFERENCES

Matthias Bauer, Emilien Dupont, Andy Brock, Dan Rosenbaum, Jonathan Richard Schwarz, and Hyunjik Kim. Spatial functa: Scaling functa to imagenet classification and generation. *arXiv preprint arXiv:2302.03130*, 2023.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Hilton Bristow, Anders Eriksson, and Simon Lucey. Fast convolutional sparse coding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 391–398, 2013.

Xu Chen, Zijian Dong, Jie Song, Andreas Geiger, and Otmar Hilliges. Category level object pose estimation via neural analysis-by-synthesis. In *European Conference on Computer Vision*, pp. 139–156. Springer, 2020.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pp. 2990–2999. PMLR, 2016a.

Taco S Cohen and Max Welling. Steerable cnns. *arXiv preprint arXiv:1612.08498*, 2016b.

Timothy F Cootes, Gareth J Edwards, and Christopher J Taylor. Active appearance models. In *European conference on computer vision*, pp. 484–498. Springer, 1998.

Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 764–773, 2017.

Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *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, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Emilien Dupont, Hyunjik Kim, SM Eslami, Danilo Rezende, and Dan Rosenbaum. From data to functa: Your data point is a function and you can treat it like one. *arXiv preprint arXiv:2201.12204*, 2022.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Alexander Gielisse and Jan van Gemert. End-to-end implicit neural representations for classification. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 18728–18737, 2025.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28, 2015.

Sékou-Oumar Kaba, Arnab Kumar Mondal, Yan Zhang, Yoshua Bengio, and Siamak Ravanbakhsh. Equivariance with learned canonicalization functions. In *International Conference on Machine Learning*, pp. 15546–15566. PMLR, 2023.

Chen-Hsuan Lin and Simon Lucey. Inverse compositional spatial transformer networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2568–2576, 2017.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 11966–11976. IEEE, 2022.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106, 2021.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

Zhixin Shu, Mihir Sahasrabudhe, Riza Alp Guler, Dimitris Samaras, Nikos Paragios, and Iasonas Kokkinos. Deforming autoencoders: Unsupervised disentangling of shape and appearance. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 650–665, 2018.

Utkarsh Singhal, Ryan Feng, Stella X Yu, and Atul Prakash. Test-time canonicalization by foundation models for robust perception. *arXiv preprint arXiv:2507.10375*, 2025.

Vincent Sitzmann, Eric Chan, Richard Tucker, Noah Snavely, and Gordon Wetzstein. Metasdf: Meta-learning signed distance functions. *Advances in Neural Information Processing Systems*, 33:10136–10147, 2020a.

Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33: 7462–7473, 2020b.

Nicki Skafte and Søren Hauberg. Explicit disentanglement of appearance and perspective in generative models. *Advances in Neural Information Processing Systems*, 32, 2019.

Shanlin Sun, Kun Han, Chenyu You, Hao Tang, Deying Kong, Junayed Naushad, Xiangyi Yan, Haoyu Ma, Pooya Khosravi, James S Duncan, et al. Medical image registration via neural fields. *Medical Image Analysis*, 97:103249, 2024.

Matthew Tancik, Ben Mildenhall, Terrance Wang, Divi Schmidt, Pratul P Srinivasan, Jonathan T Barron, and Ren Ng. Learned initializations for optimizing coordinate-based neural representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2846–2855, 2021.

David R Wessels, David M Knigge, Samuele Papa, Riccardo Valperga, Sharvaree Vadgama, Efstratios Gavves, and Erik J Bekkers. Grounding continuous representations in geometry: Equivariant neural fields. *arXiv preprint arXiv:2406.05753*, 2024.

Jelmer M Wolterink, Jesse C Zwienenberg, and Christoph Brune. Implicit neural representations for deformable image registration. In *International Conference on medical imaging with deep learning*, pp. 1349–1359. PMLR, 2022.

Xianglei Xing, Ruiqi Gao, Tian Han, Song-Chun Zhu, and Ying Nian Wu. Deformable generator networks: Unsupervised disentanglement of appearance and geometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3):1162–1179, 2020.

Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9308–9316, 2019.

ETHICS STATEMENT

This work uses the publicly available MNIST benchmark and does not involve human subjects, personal data, or sensitive attributes. Our broader motivation is to investigate methods that can reduce the amount of training data and model capacity required for visual recognition, potentially lowering the computational cost and barriers to developing effective models. As with improvements in computational efficiency more generally, such methods may also make machine learning systems more accessible for a wider range of applications, including potentially harmful ones. However, the methods studied here are general-purpose and we do not identify application-specific ethical risks beyond those associated with machine learning research more broadly.

REPRODUCIBILITY STATEMENT

We aim to make all experiments fully reproducible. The paper and supplementary material describe the dataset generation procedure, model architectures, transformation parameterizations, training and test-time optimization procedures, hyperparameters, and evaluation protocols. Experiments use explicitly specified random seeds and controlled MNIST subsets to enable direct replication of the reported comparisons. We additionally provide the implementation used for dataset generation, TP-INR training and inference, downstream model training, and evaluation as supplementary material. All code will be made available.

USE OF AI ASSISTANCE

Large language models (LLMs) were used to assist with writing and editing, literature retrieval and related-work discovery, research ideation and execution, and drafting portions of the manuscript. LLMs were also used during software development and debugging. All generated text, code, and research suggestions were reviewed and, where necessary, modified by the authors.