Transferring Teacher's Invariance to Student Through Data Augmentation Optimization

Tamotsu Kurioka[1], Teppei Suzuki[2][0000−0003−3435−8556], Rei Kawakami[1][0000−0003−2342−3324], and Ikuro Sato[1,2][0000−0001−5234−3177]

[1] Tokyo Institute of Technology, Tokyo, Japan
[2] DENSO IT LABORATORY, INC., Tokyo, Japan

Abstract. Transfer learning is widely used as a means to leverage knowledge obtained from a source dataset to solve a downstream task on a target dataset. In the realm of image classification, models trained on large datasets often exhibit high degrees of invariance in the output spaces against intra-class variations such as geometric or chromatic changes that leave the class labels invariant. Leveraging these invariant characteristics from one model to enhance the performance of another is an attractive prospect. However, a previous study has pointed out that conventional transfer learning approaches often compromise the robustness of the pre-trained model when transferred. In this paper, we propose a novel transfer learning method called TransInv, aimed at transferring the invariant properties of a teacher model to a target model. The invariance of the teacher model is expressed by a set of augmented samples produced by our proposed data augmentation module, which is jointly optimized with the target model parameters. We demonstrate that our proposed method effectively transfers the invariant properties of the teacher model to the target model, resulting in superior model performance compared to baseline methods. *The code will be released upon acceptance.*

Keywords: Transfer Learning · Data Augmentation.

1 Introduction

Transfer learning has become a standard practice in various machine learning applications [8, 16, 13, 14, 1, 3]. Typically, this involves transferring knowledge acquired by a pre-trained model to a target model through fine-tuning process. For example, in image recognition, models pre-trained on ImageNet [9] exhibited impressive performance in downstream tasks, such as object detection [21, 4, 30, 27] and semantic segmentation [5, 35].

In the realm of image classification, networks trained on datasets with diverse intra-class variations often exhibit strong model performance. Such models tend to develop a high degree of invariance in the output space. The idea of transferring these invariant properties from a pre-trained model to a target model would robustify the target model against similar variations. However, Yamada *et al.* [36] demonstrated that conventional transfer learning approaches, involving network fine-tuning, do not consistently preserve the invariant properties of a pre-trained model. They found that when freezing the backbone and re-training only the head on the target dataset, the target model well maintains the original robustness, but the resulting performance on the downstream tasks is often limited. This study indicates that maintaining the robustness of the pre-trained model while adapting the entire network to the target dataset is challenging in current transfer learning at current methodologies.

Aside from transfer learning, data augmentation techniques [2, 10, 43, 37, 34, 42, 40] have proven effective in enhancing model robustness against various types of image deformations. In data augmentation, hyperparameters of data deformation were used to be hand-tuned, making optimal tuning difficult. However, recent advancements in automatic hyperparameter search methods have demonstrated improved model performance. But, when the target dataset has limited size, models may not acquire sufficient intra-class invariance only by data augmentation methods.

In this work, we propose a novel transfer learning method, TransInv, which leverages the invariance captured by a teacher model through data augmentation and transfers it to the target model. Our contributions are summarized below.

- We present TransInv, a new approach to transfer learning, aimed at transferring the invariance properties encapsulated by a teacher model to a target model. The hyperparameters of the data augmentation module are optimized with gradient descent, ensuring that augmented data express the range where the teacher model possesses good invariance.
- We demonstrate that TransInv effectively transfers the teacher's invariance to the target model, resulting in superior classification accuracies compared to both naive network fine-tuning and a strong baseline data augmentation method, AugMix [20].

2 Related Work

Machine-learning models are considered robust when their performance remains largely unaffected by addition of noise or deformation to the input data. Since Christian *et al.* [33] demonstrated that adversarial examples can significantly degrade model performance, robustness against adversarial noise has been studied in depth [28]. Additionally, robustness against common corruptions given to images (*e.g.*, ImageNet-C [4]) and domain shifts [20, 15, 13, 12] have been studied.

There are several aspects that influence robustness, as follows. *Model size*: larger models trained on large-scale datasets tend to exhibit greater robustness [39]. *Regularization*: Methods such as data augmentation typically enhance robustness. *Architecture*: Vision transformers [11] exhibit higher resilience against common corruptions compared to CNN-based models [35]. *Transfer learning*: While conventional transfer learning often leads to strong generalization on downstream tasks, it may not always preserve the robustness of the pre-trained model [36].

Data augmentation, a technique to effectively expand the training dataset by generating numerous artificial data from original data samples, generally enhances robustness. Image processing techniques such as random horizontal flips, random scale-and-crop, and random color shift are commonly employed in augmentation to enhance the model performance [2, 10, 43, 37, 34, 32]. By combining such random operations, a vast amount of training data can be generated through data augmentation. Research has demonstrated that employing more sophisticated data augmentation techniques can further improve robustness against image corruptions [4, 18, 20].

In principle, applying data augmentation methods could address the issue of maintaining the robustness of a pre-trained model during fine-tuning. However, it has not been obvious how to design the augmentation that matches the robustness of the pre-trained model, since it is generally unknown to what extent the pre-trained model is robust. Although methods for automatically searching data augmentation hyperparameters [6, 17, 25, 31, 7, 29, 26, 41] often achieve high performance, they are not specifically designed to preserve the robustness acquired by the pre-trained model when transferring knowledge to a target model.

Our work stands out in that we propose a novel type of transfer learning aimed at preserving the robustness acquired by the pre-trained model, through generation of augmented data that fall within a range where the pre-trained model produces nearly invariant outputs.

3 Transferring Teacher's Invariance to Student

TransInv is a framework that jointly optimizes two models, the data augmentation model and the target model, using a min-max objective. The data augmentation model learns to modify original image samples x in a way that (a) the generated data $\hat{x}$ become adversarial for the target model and (b) the pre-trained teacher model produces nearly identical outputs for both x and $\hat{x}$. Meanwhile, the target model learns to accurately classify the augmented data samples. Additionally, we demonstrate that TransInv can be extended to incorporate multiple teacher models.

3.1 Augmentation model

Our method uses K data augmentation primitives, denoted as $a^i_{\phi_i}(x, z)$, where $i = 1, \cdots, K$. Each primitive applies a specific type of data deformation, such as contrast shifting, blurring, rotation, etc., to an input image $x \in [0, 1]^{3\times H\times W}$, where H and W represent the height and width of the image, respectively. Additionally, each primitive takes a random vector $z \sim \mathcal{N}(\mathbf{0}, I)$, where $\mathcal{N}(\mathbf{0}, I)$ represents M-dimensional normal distribution with zero mean and identity covariance matrix. The i-th primitive is constructed using a combination of a neural network with learnable parameters ϕ_i and a predefined function to express a specific type of deformation, similar to the approach outlined in [31]. Concrete examples of these primitives, along with other experimental settings, will be provided in the next section.

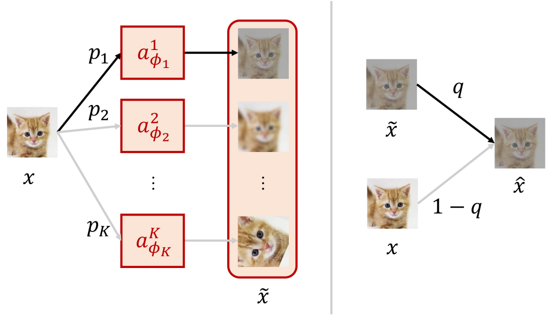


Fig. 1: The proposed process of data augmentation. Left: Each of the data augmentation primitives $\{a^i_{\phi_i}\}$ performs a particular type of image deformation. At each iteration, only i-th primitive is selected with probability p_i to produce the deformed input image, $\tilde{x}$. Right: The overall data augmentation model $A_{\phi,p,q}$ produces either the deformed image $\tilde{x}$ with probability qp_i or the original image x with probability $1-q$.

The proposed process of data augmentation is illustrated in Fig. 1. During training, for a given image x, the i-th primitive is randomly selected based on the selection probability p_i. The probability vector $p = [p_1, p_2, \cdots, p_K]^\top$ with $\sum_{i=1}^K p_i = 1$ is learned alongside the target classification model and the data augmentation model. This enables control over the frequency of different types of deformations. Let $\tilde{A}_{\phi,p}$ denote a function that selects the i-th primitive $a^i_{\phi_i}(x,z)$ with probability p_i, defined as follows:

$$\tilde{A}_{\phi,p}(x,z) = \begin{cases} a^1_{\phi_1}(x,z) & \text{with probability } p_1, \\ \vdots & \\ a^K_{\phi_K}(x,z) & \text{with probability } p_K. \end{cases} \tag{1}$$

To ensure that the target model explicitly learns the original image samples with probability $q \in [0,1]$, we define the following data augmentation function:

$$A_{\phi,p,q}(x,z) = \begin{cases} \tilde{A}_{\phi,p}(x,z) & \text{with probability } q, \\ x & \text{with probability } 1-q. \end{cases} \tag{2}$$

Equations (1) and (2) can be rewritten as Eqs. (3) and (4) by using a one-hot vector $\hat{g} = [\hat{g}_1, \hat{g}_2, \cdots, \hat{g}_K]^\top$ sampled from a categorical distribution Categorical(p), and an integer $\hat{b} \in \{0,1\}$ sampled from the Bernoulli distribution Bernoulli(q):

$$\tilde{A}_{\phi,p}(x,z) = \sum_{i=1}^{K} \hat{g}_i a^i_{\phi_i}(x,z), \quad \hat{g} \sim \text{Categorical}(p), \tag{3}$$

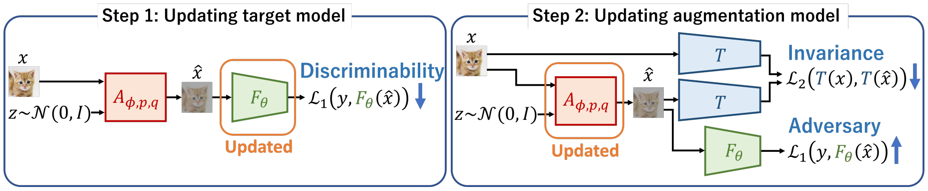

Step 1: Updating target model
x
$z \sim \mathcal{N}(0, I)$
$A_{\phi,p,q}$
$\hat{x}$
F_θ
Discriminability
$\mathcal{L}_1(y, F_\theta(\hat{x}))$
Updated
Step 2: Updating augmentation model
x
T
T
Invariance
$\mathcal{L}_2(T(x), T(\hat{x}))$
$z \sim \mathcal{N}(0, I)$
$A_{\phi,p,q}$
$\hat{x}$
Updated
F_θ
Adversary
$\mathcal{L}_1(y, F_\theta(\hat{x}))$

Fig. 2: Overview of the parameter optimization in TransInv. Step 1: It updates the target model parameters θ to reduce the classification loss. Step 2: It updates the augmentation model parameter ϕ, the primitive sampling probabilities p_i $(i = 1, \cdots, K)$, and the binary sampling probability q to bring the outputs from both the target and teacher models closer and to raise the classification loss for the target model.

$$A_{\phi,p,q}(x, z) = \hat{b}\tilde{A}_{\phi,p}(x, z) + (1 - \hat{b})x, \quad \hat{b} \sim \text{Bernoulli}(q). \tag{4}$$

The augmentation model described in Eq. (4) as it is non-differentiable with respect to p or q because it involves discrete variables $\hat{g}_1, \cdots, \hat{g}_K$, and $\hat{b}$. We address this issue by relaxing these discrete variables to continuous ones using the Gumbel-Softmax trick, allowing for the computation of derivatives. For further details on the Gumbel-Softmax trick, readers are referred to [23]. Henceforth, we assume that $A_{\phi,p,q}(x, z)$ is differentiable due to this trick.

3.2 Optimization of TranInv

Let $\mathcal{X} = \{(x_i, y_i)\}_{i=1,\cdots,N}$ denote the target training dataset, where $x_i \in \mathbb{R}^{3\times H\times W}$ represents the i-th image and y_i is its corresponding c-dimensional class label. The target model, parameterized by θ, is denoted as $F_\theta : \mathbb{R}^{3\times H\times W} \to \Delta^{c-1}$, while the teacher model is represented as $T : \mathbb{R}^{3\times H\times W} \to \Delta^{d-1}$. Here, Δ^n denotes n-unit simplex, defined as $\Delta^n = \{[b_1, \cdots, b_{n+1}]^\top \in \mathbb{R}_+^{n+1} | \sum_{i=1}^{n+1} b_i = 1\}$. In our setup, the number of classes in the source dataset and that in the target dataset may differ. The optimization problem for the augmentation model parameters ϕ, the primitive sampling probabilities p_i $(i = 1, \cdots, K)$, the binary sampling probability q, and the target model parameters θ can be formulated as follows:

$$\min_\theta \max_{\phi,p,q} \sum_{i=1}^{N} \mathop{\mathbb{E}}_{z\sim\mathcal{N}(\mathbf{0},I)} \left[\mathcal{L}_1(y_i, F_\theta(A_{\phi,p,q}(x_i, z))) - \mathcal{L}_2(T(x_i), T(A_{\phi,p,q}(x_i, z))\right]. \tag{5}$$

Here, $\mathcal{L}_1$ and $\mathcal{L}_2$ represent the cross entropy losses defined for the target and source datasets, respectively. Note that $\mathcal{L}_2$ is equal to $\mathcal{L}_1$ in a special case where the dimensions of y and $T(x)$ match. The first term represents the supervised loss for the target model F_θ with deformed input. The second term quantifies the discrepancy between the output of the teacher model with deformed and

Algorithm 1 TransInv

```
Require: Training dataset $\mathcal{X}$; target model $F_\theta$; teacher model $T$; augmentation model $A_{\phi,p,q}$; learning rates $\eta_{\text{tar}}, \eta_{\text{aug}}$; batch size $B$; the number of outer iterations $n_{\text{outer}}$; the number of inner iterations $n_{\text{inner}}$
  for $1, .., n_{\text{outer}}$ do
    for $1, .., n_{\text{inner}}$ do
      Randomly sample a mini-batch $\mathcal{B}$ of size $B$ from $\mathcal{X}$.
      $z \sim \mathcal{N}(\mathbf{0}, I)$
      Compute loss for the target model, i.e., $\mathcal{L}_{\text{tar}} = \sum_{b \in \mathcal{B}} \mathcal{L}_1(y_b, F_\theta(A_{\phi,p,q}(x_b, z)))$.
      Update $\theta$ by the gradient descent, i.e., $\theta \leftarrow \theta - \eta_{\text{tar}} \partial \mathcal{L}_{\text{tar}} / \partial \theta$.
    end for
    Randomly sample a mini-batch $\mathcal{B}$ of size $B$ from $\mathcal{X}$.
    $z \sim \mathcal{N}(\mathbf{0}, I)$
    Compute loss for the augmentation model, i.e.,
    $\mathcal{L}_{\text{aug}} = \sum_{b \in \mathcal{B}} [\mathcal{L}_1(y_b, F_\theta(A_{\phi,p,q}(x_b, z))) - \mathcal{L}_2(T(x_b), T(A_{\phi,p,q}(x_b, z)))]$.
    Update $\phi, p, q$ by gradient ascent, i.e., $(\phi, p, q) \leftarrow (\phi, p, q) + \eta_{\text{aug}} \partial \mathcal{L}_{\text{aug}} / \partial (\phi, p, q)$.
  end for
```

undeformed inputs. The augmentation parameters ϕ and the sampling probabilities p, q are updated to simultaneously increase the classification loss for the target model F_θ and reduce the discrepancy in the teacher's outputs. This formulation aims to make the deformation nearly invariant to the teacher's output, while generating the data adversarial to the target model F_θ. Conversely, the target model parameters θ are trained to counteract adversarial deformations in the minimization of the supervised loss. Through this iterative process, F_θ gradually acquires the discriminative capabilities within the range of deformations where the teacher model T exhibits some degree of invariance.

The optimization problem described in Eq. (5) can be approximately solved by alternately updating (ϕ, p, q) and θ. As illustrated in Fig. 2, Step 1 involves updating the target model parameters θ while keeping the augmentation parameters ϕ and the sampling probabilities p, q fixed. Step 2 updates (ϕ, p, q) with fixed θ. This process is iterated for a predefined number of epochs. The algorithm is summarized in Algorithm 1.

Extension to multiple teacher models: We propose an extension to the scenario where multiple teacher models, possibly trained on different datasets, are utilized. Let T^k $(k = 1, \cdots, L)$ denote the teacher models, associated with independent augmentation models A_{ϕ^k, p^k, q^k} $(k = 1, \cdots, L)$. The optimization of parameters associated with the augmentation models (ϕ^k, p^k, q^k) and the target model parameters θ is formulated as follows:

$$\begin{aligned}\min_\theta \sum_{k=1}^{L} \max_{\phi^k, p^k, q^k} \sum_{i=1}^{N} \mathop{\mathbb{E}}_{z \sim \mathcal{N}(\mathbf{0}, I)} [&\mathcal{L}_1(y_i, F_\theta(A_{\phi^k, p^k, q^k}(x_i, z))) \\ &-\mathcal{L}_2(T^k(x_i), T^k(A_{\phi^k, p^k, q^k}(x_i, z)))]. \qquad (6)\end{aligned}$$

This extension allows for the transfer of invariant properties from multiple teacher models. Additionally, as a training option, one may introduce additional parameter vector $r = [r_1, \cdots, r_L]^\top$, specifying the probability of selecting augmentation model at each iteration. The probability r can be trained in the same manner as the selection probability p for the augmentation primitives, as described earlier.

4 Experiments

We evaluate TransInv under three experimental settings. In Section 4.1, we demonstrate how our method can transfer teacher's invariance to a target model. In Section 4.2, we compare model performance of TransInv with baseline methods under an ordinary transfer learning setting, where one teacher model is used. In Section 4.3, we examine TransInv under the scenario where multiple teacher models are simultaneously used to train a target model. Below, we describe the general experimental settings.

Model architecture: We used the WideResNet-40-2 [38] architecture for both teacher (or pre-trained) and target models in Sections 4.1 and 4.2. We used the WideResNet-28-10 architecture for both teacher and target models in Section 4.3.

Datasets: Teacher models were trained on the corrupted image dataset CIFAR-10-C [19] and its uncorrupted version CIFAR-10 [24]. For each type of corruptions we trained one teacher model using data augmentation with the corresponding corruption, ensuring that the generated teacher is robust to that specific corruption. The resulting teacher models are named according to the type of corruption they are robust to, including Gaussian Noise, Shot Noise, Impulse Noise, Brightness, Contrast, Elastic Transform, Defocus Blur, Glass Blur, Motion Blur, Zoom Blur, Snow, Frost, Fog, Pixelate, and JPEG Compression. The target models were trained on the uncorrupted image dataset CIFAR-100 [24].

4.1 Illustration of invariance transfer

This toy experiment serves to illustrate how the teacher's invariance can be transferred. We used following six teacher models: Gaussian Noise, Contrast, Elastic Transform, Glass Blur, Snow, and JPEG Compression. Subsequently, six target models were trained on CIFAR-100 using our method, with each target model employing one of the teacher models. For the augmentation model, we employed 15 primitives, each corresponding to one of the corruptions provided by CIFAR-C [19]. It is worth noting that in this toy experiment, we explicitly provided information about the types of deformations the teacher models are robust to. In this toy experiment, only the augmentation probabilities p and q were optimized for the augmentation model, while the strength of the deformations was manually specified.

The primitive-selection probabilities p after applying TransInv are depicted in Fig. 3. A high probability indicates that the target exhibits relatively strong

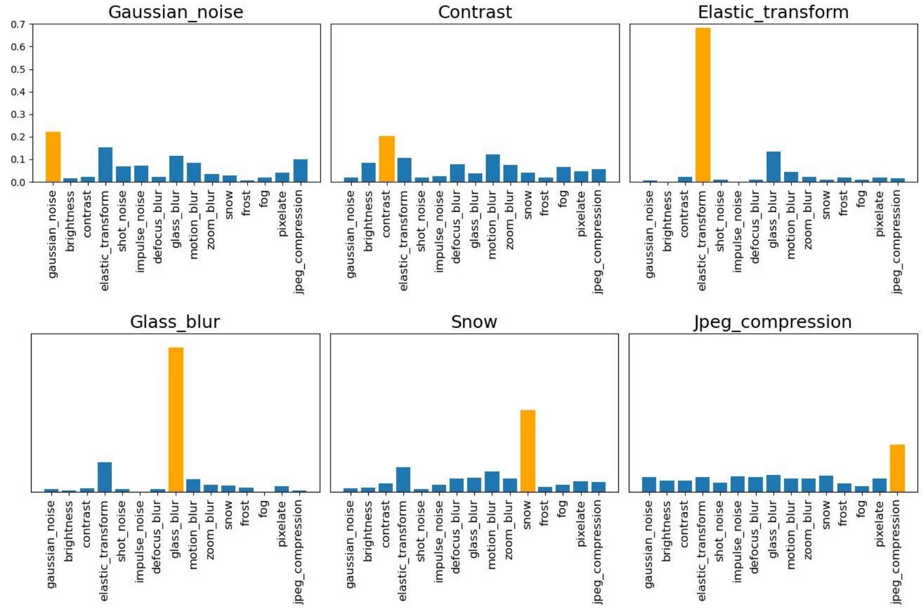


Fig. 3: The primitive selection probabilities p optimized by TransInv in the Section 4.1 experiment. The graph titles indicate the types of teacher models. The x-axis lists 15 data augmentation primitives and the y-axis shows the primitive selection probabilities optimized by TransInv. The results indicate that in the training TransInv most frequently generates the same type of deformations, to which the teacher model is most robust.

invariance to that particular deformation. In all cases, the highest probabilities appear when the corresponding teachers are used. For instance, when the Elastic transform teacher model is employed, the target model encounters elastic-transformed samples most frequently. It is important to note that TransInv learns these frequencies from data. This observation highlights that TransInv effectively learns the robustness of the teacher models through adaptive data augmentation.

4.2 Transfer learning with one teacher

In this experiment, we evaluate how TransInv enhances the robustness of the target models against input perturbations, leveraging the strong invariance demonstrated by the teacher models. We utilized all 15 teacher models, including Gaussian Noise (GN), Shot Noise (SN), Impulse Noise (IN), Brightness (Br), Contrast (Co), Elastic Transform (ET), Defocus Blur (DB), Glass Blur (GB), Motion Blur (MB), Zoom Blur (ZB), Snow (Sn), Frost (Fr), Fog (Fo), Pixelate (Pi), and JPEG Compression (JC), as mentioned earlier.

Augmentation model: We defined $K = 4$ augmentation primitives. The i-th primitive $a^i_{\phi_i}$ is constructed using a combination of a neural network with

	GN	SN	IN	Br	Co	ET	DB	GB	MB	ZB	Sn	Fr	Fo	Pi	JC	Mean
FT	61.5	61.6	62.2	62.6	61.6	65.1	62.2	62.7	62.5	63.2	63.1	62.7	62.5	61.9	60.5	62.4
AM	**65.8**	65.7	65.3	**67.0**	**66.0**	67.8	65.8	66.1	66.6	66.0	66.2	66.1	65.8	65.6	63.9	66.0
FA	65.6	65.4	66.1	66.0	65.3	66.7	65.8	65.3	66.9	66.4	66.7	65.9	65.9	66.7	**64.0**	65.9
Ours	64.5	**66.3**	**66.9**	66.5	**66.0**	**69.6**	**67.3**	**67.6**	**68.2**	**67.5**	**68.1**	**67.6**	**66.7**	**67.4**	63.0	**66.9**

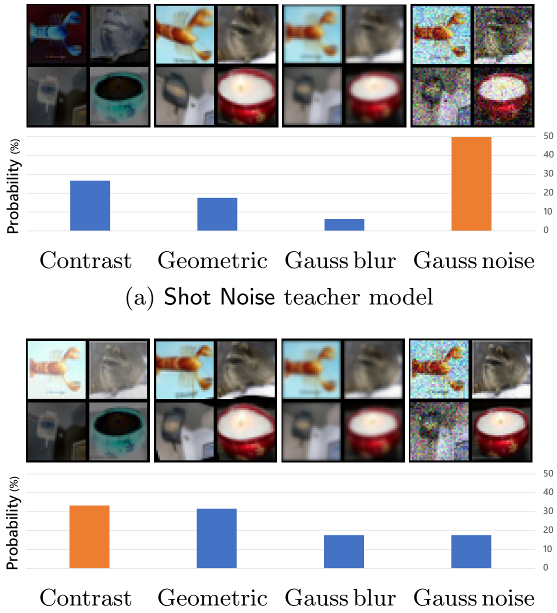
Table 1: Accuracies (%) on the clean CIFAR-100 test set in the single teacher experiment (Section 4.2). The teacher/pre-trained models are trained on CIFAR-10-C and CIFAR-10. GN indicates Gaussian noise, etc. (see the text). FT, AM, and FA indicates Fine-tuning, Fine-tuning with AugMix, and Fine-tuning with FixedAug, respectively. Our TransInv achieves the best in 12/15 cases.

parameters ϕ_i and a predefined function to express a specific type of deformation, as follows:

- Contrast and brightness transformation
  $a^1_{\phi_1}(x, z) = \alpha_{\phi_1}(z)x + \beta_{\phi_1}(z)$
- Geometric transformation
  $a^2_{\phi_2}(x, z)$ (following implementation in Spatial Transformer Networks [22])
- Gaussian blur
  $a^3_{\phi_3}(x, z) = \gamma_{\phi_3}(z)x + (1 - \gamma_{\phi_3}(z))\mathrm{Conv}(x, \kappa_{\phi_3}(z))$
- Gaussian noise
  $a^4_{\phi_4}(x, z) = x + \mathcal{N}(\mathbf{0}, \nu_{\phi_4}(z))$

Function α_{ϕ_1} is an MLP that produces a scalar in $(0.2, 1.8)$, and β_{ϕ_1} is an MLP that produces a scalar in $(-0.5, 0.5)$ to change contrast and brightness respectively. Function γ_{ϕ_3} is an MLP that produces blur intensity in $(0, 1)$, and κ_{ϕ_3} is an MLP that produces a Gaussian filter with variance in $(0, 1.8)$. 'Conv' denotes convolution operation. Function ν_{ϕ_4} is an MLP that produces the variance of Gaussian noise in $(0, 0.12)$. The scalar ranges described above are loosely defined only to prevent the data becoming too adversarial to provoke instability. The idea is that each primitive learns an appropriate range from data. Coarsely speaking, these four primitives can embody image corruptions, such as Gaussian noise, shot noise, impulse noise, brightness change, contrast change, and elastic transform. For the detailed definitions of the primitive, readers are referred to our implementation.

Baselines: We evaluated TransInv alongside three baselines: Fine-tuning, Fine-tuning with AugMix, and Fine-tuning with FixedAug. In Fine-tuning, the target network was initialized using one of the teacher models and fine-tuned on the clean target dataset without any data augmentation. In Fine-tuning with AugMix, the target network was initialized using one of the teacher models and fine-tuned with data augmentation using AugMix technique [20]. In Fine-tuning with FixedAug, the target network was initialized using one of the teacher models and fine-tuned with the same data augmentation used in TransInv, but with fixed primitive selection probabilities and augmentation strengths. Specifically, in Fine-tuning with FixedAug, one of the four deformations was randomly selected,



(a) Shot Noise teacher model

(b) Fog teacher model

Fig. 4: Visualization of augmented data of CIFAR-100 obtained by TransInv when transferred from (a) Shot Noise teacher model and (b) Fog teacher model trained on CIFAR-100 with shot noise and fog data augmentation, respectively. Images generated by the augmentation primitives $a^1_{\phi_1}$ (Contrast and brightness transformation), $a^2_{\phi_2}$ (Geometric transformation), $a^3_{\phi_3}$ (Gaussian blur), and $a^4_{\phi_4}$ (Gaussian noise) are shown from left. Corresponding primitive-selection probabilities p_1, p_2, p_3, and p_4 optimized by TransInv are shown below. See the main text for interpretation.

and the strength was uniformly sampled from six levels at each iteration. We followed the implementation adopted in [19], where each deformation has six levels of strength.

Results: Table 1 shows the accuracies on the clean test set (i.e., no corruption applied) of CIFAR-100 when transferring from CIFAR-10 to CIFAR-100. Compared to the baseline methods, the proposed TransInv demonstrates superior accuracies on the clean test set across most of the corruption categories. This suggests that the target model tends to perform well when trained with data augmentation to which the teacher is robust. Although the baseline models Fine-tuning with AugMix and Fine-tuning with FixedAug incorporate relatively strong data augmentation during training, the pre-trained model may not be necessarily robust to such augmentation. The mismatch in the robustness under different data deformations between the teacher and the target models likely

Table 2: Accuracies (%) on the clean CIFAR-100 test set in the multiple teacher experiment (Section 4.3). TransInv uses three teacher models simultaneously.

Scratch	RandAug[7]	TeachAug[31]	TransInv
81.2	83.3	83.2	**83.5**

explains the reason why these baseline methods underperform TransInv in most cases.

The visualization of augmented data samples and optimized primitive selection probabilities is given in Fig.4. We show the cases of the Shot Noise teacher model, which is robust to shot noise, and the Fog teacher model, which is robust to foggy noise. In the Shot Noise case, it is evident from the optimized primitive probabilities that the target model is trained with similar noise most frequently, indicating a shared robustness between the teacher and target models. The Fog case is a little harder to make similar observation, since a foggy noise is not generated by a single image deformation defined here. However, qualitatively speaking, some images generated by the contrast and brightness transformation, which has the highest primitive-selection probability, somewhat resembles foggy images.

4.3 Transfer learning with multiple teachers

We extended the proposed method to a scenario where multiple teacher models are employed. For L teacher models $T^1, \cdots, T^L$, we prepare corresponding augmentation models $A_{\phi^1,p^1,q^1}, \cdots, A_{\phi^L,p^L,q^L}$. During the training of the target model, we optimized the selection probabilities $r = [r_1, \cdots, r_L]^\top$ for the augmentation models. This means that model A_{ϕ^k,p^k,q^k} is selected with probability r_k at a given iteration, similar to how the primitive selection probabilities p_i $(i = 1, \cdots, K)$ are optimized.

In the experiment with TransInv, we utilized three teacher models (i.e., $L = 3$) including a CIFAR-10 pre-trained model, an ImageNet pre-trained model, and the exponential moving average model of the target model itself. The target model was trained from scratch on CIFAR-100. All models utilize WideResNet-28-10 architecture.

Classification accuracies on the clean CIFAR-100 test set are presented in Table 2. In the table, 'Scratch' denotes the model trained from scratch on CIFAR-100 without data augmentation, 'RandAug' ('TeachAug') refers to the model trained from scratch on CIFAR-100 with RandAug[7] (TeachAug [31]) data augmentation techniques. 'TransInv' indicates the proposed method, which achieves higher accuracy compared to the baseline methods. This result suggests that TransInv effectively transfers the properties of multiple teacher models to a target model, resulting in improved performance on the CIFAR-100 dataset. In the conventional transfer learning, known as network finetuning, it is not possible to utilize more than one pre-trained model, since the pre-trained model parameters are directly updated in this process. In contrast, our method trains a new model, while teacher model(s) guide the augmentation model to generate augmented data. In this new scheme, arbitrary many teacher models can be accommodated to produce a single target model. This scheme provides a flexibility of choosing different teacher models in transferring invariant properties to a target model. The performance boost shown in Table 2 is likely explained by the use of multiple teacher models with distinct properties; namely, one trained on CIFAR-10, another one trained on ImageNet, and the other being the exponential moving average of the target model.

5 Conclusion

We proposed a novel transfer learning method, TransInv, which leverages data augmentation to transfer the robustness of teacher model to target model. The range of invariance exhibited by the teacher model is learned through a set of augmentation primitives, whose parameters and selection probabilities are optimized via gradient descent. These primitives are then utilized to generate deformed images in the target domain, enabling the target model to learn and acquire similar invariance. Our experimental results validate that target models trained using TransInv demonstrate similar invariance properties exhibited by the teacher models. Overall, TransInv consistently outperform naive transfer learning and data augmentation methods in terms of model performance. Furthermore, we demonstrated that TransInv is capable of utilizing multiple teacher models to transfer knowledge from diverse datasets, enhancing its versatility and effectiveness in real-world scenarios.

As a future work, we will investigate the effectiveness on various target datasets, although we demonstrated successful transfer from multiple teacher models trained on different datasets in this work. Incorporating various image domains, such as driving environment or human monitoring, is intriguing for widening visual applications, as TransInv has a potential to generate robust and light-weight models.

Acknowledgement

This work is an outcome of a research project, Development of Quality Foundation for Machine-Learning Applications, supported by DENSO IT LAB Recognition and Learning Algorithm Collaborative Research Chair (Tokyo Tech.).

References

1. Abuduweili, A., Li, X., Shi, H., Xu, C.Z., Dou, D.: Adaptive consistency regularization for semi-supervised transfer learning. In: CVPR (2021)

2. Bengio, Y., Bastien, F., Bergeron, A., Boulanger-Lewandowski, N., Breuel, T.M., Chherawala, Y., Cissé, M., Côté, M., Erhan, D., Eustache, J., Glorot, X., Muller, X., Lebeuf, S.P., Pascanu, R., Rifai, S., Savard, F., Sicard, G.: Deep learners benefit more from out-of-distribution examples. In: AISTATS (2011)
3. Bu, X., Peng, J., Yan, J., Tan, T., Zhang, Z.: Gaia: A transfer learning system of object detection that fits your needs. In: CVPR (2021)
4. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV (2020)
5. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE TPAMI (2018)
6. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation strategies from data. In: CVPR (2019)
7. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.: Randaugment: Practical automated data augmentation with a reduced search space. In: NeurIPS (2020)
8. Dai, A.M., Le, Q.V.: Semi-supervised sequence learning. In: NeurIPS (2015)
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009)
10. DeVries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552 (2017)
11. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021)
12. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: ICML (2015)
13. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. J. Mach. Learn. Res. (2016)
14. Gao, X., He, Y., Dong, S., Cheng, J., Wei, X., Gong, Y.: Dkt: Diverse knowledge transfer transformer for class incremental learning. In: CVPR (2023)
15. Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W.: Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In: ICLR (2019)
16. Guo, Y., Shi, H., Kumar, A., Grauman, K., Rosing, T., Feris, R.: Spottune: Transfer learning through adaptive fine-tuning. In: CVPR (2019)
17. Hataya, R., Zdenek, J., Yoshizoe, K., Nakayama, H.: Faster autoaugment: Learning augmentation strategies using backpropagation. In: ECCV (2020)
18. Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., Song, D., Steinhardt, J., Gilmer, J.: The many faces of robustness: A critical analysis of out-of-distribution generalization. In: ICCV (2021)
19. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. In: ICLR (2019)
20. Hendrycks*, D., Mu*, N., Cubuk, E.D., Zoph, B., Gilmer, J., Lakshminarayanan, B.: Augmix: A simple method to improve robustness and uncertainty under data shift. In: ICLR (2020)
21. Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., Murphy, K.: Speed/accuracy trade-offs for modern convolutional object detectors. In: CVPR (2017)

22. Jaderberg, M., Simonyan, K., Zisserman, A., kavukcuoglu, k.: Spatial transformer networks. In: NeurIPS (2015)
23. Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. In: ICLR (2017)
24. Krizhevsky, A.: Learning multiple layers of features from tiny images. Technical Report (2009)
25. Li, R., Li, X., Heng, P.A., Fu, C.W.: Pointaugment: An auto-augmentation framework for point cloud classification. In: CVPR (2020)
26. Li, Y., Hu, G., Wang, Y., Hospedales, T., Robertson, N.M., Yang, Y.: Differentiable automatic data augmentation. In: ECCV (2020)
27. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV (2021)
28. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: ICLR (2018)
29. Muller, S.G., Hutter, F.: Trivialaugment: Tuning-free yet state-of-the-art data augmentation. In: ICCV (2021)
30. Su, W., Zhu, X., Tao, C., Lu, L., Li, B., Huang, G., Qiao, Y., Wang, X., Zhou, J., Dai, J.: Towards all-in-one pre-training via maximizing multi-modal mutual information. In: CVPR (2023)
31. Suzuki, T.: Teachaugment: Data augmentation optimization using teacher knowledge. In: CVPR (2022)
32. Suzuki, T., Sato, I.: Adversarial transformations for semi-supervised learning. In: AAAI (2020)
33. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
34. Tokozume, Y., Ushiku, Y., Harada, T.: Between-class learning for image classification. In: CVPR (2018)
35. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: Segformer: Simple and efficient design for semantic segmentation with transformers. In: Neural Information Processing Systems (NeurIPS) (2021)
36. Yamada, Y., Otani, M.: Does robustness on imagenet transfer to downstream tasks? In: CVPR (2022)
37. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: ICCV (2019)
38. Zagoruyko, S., Komodakis, N.: Wide residual networks. arXiv preprint arXiv:1605.07146 (2016)
39. Zhai, X., Kolesnikov, A., Houlsby, N., Beyer, L.: Scaling vision transformers. In: CVPR (2022)
40. Zhang, X., Tseng, N., Syed, A., Bhasin, R., Jaipuria, N.: Simbar: Single image-based scene relighting for effective data augmentation for automated driving vision tasks. In: CVPR (2022)
41. Zhang, X., Wang, Q., Zhang, J., Zhong, Z.: Adversarial autoaugment. In: ICLR (2020)
42. Zhao, Z., Yang, L., Long, S., Pi, J., Zhou, L., Wang, J.: Augmentation matters: A simple-yet-effective approach to semi-supervised semantic segmentation. In: CVPR (2023)
43. Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation. AAAI (2020)