Efficiently Robustify Pre-Trained Models

Nishant Jain *  
IIT Roorkee 

Harkirat Behl  
Microsoft Research 

Yogesh Singh Rawat  
CRCV, UCF 

Vibhav Vineet  
Microsoft Research 

Abstract

A recent trend in deep learning algorithms has been towards training large scale models, having high parameter count and trained on big dataset. However, robustness of such large scale models towards real-world settings is still a less-explored topic. In this work, we first benchmark the performance of these models under different perturbations and datasets thereby representing real-world shifts, and highlight their degrading performance under these shifts. We then discuss on how complete model fine-tuning based existing robustification schemes might not be a scalable option given very large scale networks and can also lead them to forget some of the desired characteristics. Finally, we propose a simple and cost-effective method to solve this problem, inspired by knowledge transfer literature. It involves robustifying smaller models, at a lower computation cost, and then use them as teachers to tune a fraction of these large scale networks, reducing the overall computational overhead. We evaluate our proposed method under various vision perturbations including ImageNet-C,R,S,A datasets and also for transfer learning, zero-shot evaluation setups on different datasets. Benchmark results show that our method is able to induce robustness to these large scale models efficiently, requiring significantly lower time and also preserves the transfer learning, zero-shot properties of the original model which none of the existing methods are able to achieve.

1. Introduction

Large scale deep neural networks trained on large scale data have revolutionized the modern AI era. They are significantly effective in solving practical problems of high importance. These include object detection, zero-shot classification, image segmentation, image generation, and many other applications [18, 25, 27, 39, 5, 29, 35, 9].

Though the large models have shown impressive results on many vision problems [27, 29], their reliability under distribution shift e.g., under illumination changes, geographical variations, camera properties etc., is still underexplored. In this paper, we first investigate the behavior of large models under distribution shifts. We analyse popular models under synthetic perturbations to images [11], natural distribution shifts [10, 13], differently styled images [31] and dataset shift [2]. Our analysis of models of various sizes, architecture families (transformers or CNNs) and training modalities (uni or multi-modal) establishes their brittleness under distribution shifts.

This analysis begs the question: can we induce robustness to large vision models without sacrificing their original properties? It is critical to simultaneously maintain clean accuracy on the original datasets, improve robust accuracy on the shifted data and preserve the transfer learning capabilities of the large models. Further, computation efficiency during both training and inference is beneficial.

While several prior works can be used to make large-scale models robust, they do not possess the desired prop-
properties discussed above. One direction involves fine-tuning the model [33, 16]. This generally suffers from either poor performance under synthetic perturbations or requires significant training time. Another line of work could be to use advanced augmentation techniques (e.g., aug-mix, pix-mix) [14, 12, 10, 4]. They are effective under synthetic perturbations and natural shifts in the data. However, they require significantly larger compute time and lead to the large models forgetting their original and transfer learning properties. Figure 1 shows this analysis in a pareto-front plot for two of the recently proposed robustness methods.

To this end, we propose a knowledge transfer method to induce robustness to large models that possess all the desired properties discussed above. It makes large models robust efficiently (refer Fig. 1). We take a plug-and-play approach: insert an additional small robust module and update only a very small portion of the existing large models. To achieve robustness, we explore a new direction: a relatively much smaller but robust model inducing robust knowledge to a large model. Though this provides a novel look at the knowledge distillation approach, a straight-forward application leads to the large models forgetting their original properties. For this challenging task of ensuring that clean accuracy is preserved in the clean module, robustness induced into the robust module and correct module selected at test time, we propose a novel uncertainty-aware knowledge distillation technique. This allows us to fulfill all our required objectives. Since our method involves updating only a small chunk of the large network, it achieves low training latency (refer section 5). To the best of our knowledge, this is the first time such a setup has been used involving knowledge transfer from a smaller to a large model. Further, it should be noted that smaller models can be made robust by using prior works like advance augmentation methods [12, 10, 14].

We evaluate our method under various distribution shift on ImageNet data [28] in section 5. It includes ImageNet-C [11], ImageNet-R [10], ImageNet-A [13], ImageNet-sketch [31], ImageNet-V2. We also evaluate on ObjectNet [2] and its perturbed variations ObjectNet-C. We show results for both multi-modal (various CLIP models) and unimodal (various architectures including both ResNets and Vision Transformers). Alongside this, we also test our method on other datasets in the transfer learning setup, to analyze further if the desired properties of the model are restored. In all these cases, our method outperforms prior approaches on robust accuracy while still performing at par on clean accuracy. At the same time, possessing desired characteristics like transfer learning capabilities (refer section 5) and being efficient during training and inference.

2. Related Work

Large scale models. In recent years, studies [38, 36, 8] have shown that training large models such as vision transformers [5] on large datasets can improve accuracies significantly. Several works [26, 3] have evaluated these models for robustness lately. Furthermore, these large models can be trained either in a unimodality setup [26] or multi modality setup [27, 39, 35]. Though they achieve good performance on several downstream tasks, any modification of these large models can lead to forgetting of the knowledge contained in them. In contrast, we propose a method that allows to adapt the model parameters without sacrificing their properties.

Finetuning and Augmentation based methods to achieve robustness. Several advanced augmentation techniques have been proposed to improve robustness of the deep learning based models. Examples includes cut-out, cutmix, mixup, augmix, pixmix and others [4, 37, 12, 40]. Further, a recently proposed method WISE interpolates the fine-tuned and original model parameters [33] for robustness [40]. Generally these techniques are a computation heavy process and also leads to modifying the main network parameters that could lead to these large models forgetting in their original properties. In our approach, we use some of these advanced augmentation technique to make our teacher network robust. We ensure that our robust approach does not sacrifice the large models’ properties and is also computationally efficient.

Knowledge distillation It involves transferring knowledge from a large network to a smaller network by minimizing the distance between the predicted logit distribution of the student and teacher networks [15]. It proved to be highly effective on standard downstream datasets [1, 7]. In all these KD applications, knowledge is transferred from a larger network to a smaller network. In contrast, we propose a method to induce robustness to a larger network by transferring knowledge from a smaller (teacher) network.

3. Robustness Analysis

In this section, we analyze the image classification performance of models of different shapes and sizes, with different training settings (unimodal or multimodal). We stress test the models under both synthetic and natural perturbations. Especially, contrasting the behaviour of multimodal models (vision-language) vs. unimodal (image only).

Models. In unimodal setting, we analyse Resnet-50, ResNet-101 and ResNet-150 [9], ViT-small, ViT-base and ViT-large models [5] trained on ImageNet [28]. In multimodal setting, we analyse CLIP [27] model with backbones including ResNets: CLIP-RN50 and CLIP-RN101, and transformers: CLiP-ViT B/16, CLiP-ViT B/32. We also analyze self-supervised unimodalities trained on large
datasets as against only ImageNet pretrained ones. For this, we use the masked autoencoders [8] and DINO V2 [24] models proposed recently, shown to be highly effective in representation learning. We analyze two architecture, ViT-B/16, ViT-B/32 for MAE and ViT-B/14 for DINO V2.

Datasets. We evaluate the models on various shifted version of ImageNet [28]: ImageNet-Corrupted (ImageNet-C) [11], ImageNet-Rendition (ImageNet-R) [10] and ImageNet-Sketch (ImageNet-S) [31] datasets. For natural shifts, we use ImageNet-Adversarial (ImageNet-A) comprising natural adversarial examples [13] (results in supplementary). Further, we also evaluate the models on ObjectNet [2] and its perturbed version ObjectNet-C, which we generate by applying all the 15 corruptions of ImageNet-C to ObjectNet data.

Experimental Setup. In the unimodal case, we evaluate ImageNet trained models from timm [32] on ImageNet-C, ImageNet-R, ImageNet-S (supplementary), ObjectNet and ObjectNet-C datasets. Further, we evaluate the multimodal models in the linear-probe [27] and zero-shot settings. For the linear probe setup, please refer to the CLIP paper. It is done on the ImageNet dataset and then evaluated on all of these.

Results. Fig. 3 (left) shows analysis of all the architectures on ImageNet-C data with varying severity levels, for ImageNet pretrained unimodalities (solid lines) and Zero-Shot CLIP-models (dashed lines). They all suffer similarly when severity is increased and even though start from different accuracies, converge to similar values at high severity. This implies under high perturbations they break-down equally. However, the robustness of CLIP based models is slightly higher. One possible reason is that they start from lower values of clean accuracy attributed to their zero-shot nature. Fig. 2 shows the analysis of various CLIP model architectures on the ImageNet-R, ObjectNet and ObjectNet-C datasets under both Linear Probe and zero-shot settings along with the unimodal (ImageNet pretrained) counterparts. For the linear probe setting, the models maintain accuracy on ImageNet-R and ObjectNet datasets whereas suffer significantly on ObjectNet-C. On the other hand, zero-shot models show better accuracy on the ImageNet-R (compared to ImageNet), slightly lower on the ObjectNet dataset and suffer on the ObjectNet-C dataset similar to linear probe setting. From the results, it can be observed that zero-shot CLIP is more robust on ObjectNet-C than linear probe CLIP based on the relative drop on accuracy. Also, the zero-shot CLIP model outperforms the linear probe one on the ImageNet-R dataset, even though the linear probe was on ImageNet. For unimodal case, all models observe significant drop in accuracy and perform poorly (much worse than CLIP Linear Probe and Zero-Shot) under all the shifts.

Self Supervised Unimodalities. We further analyse another case where the unimodal models are trained in a self supervised fashion on large datasets as against the ImageNet pretrained models. For this, we use the masked autoencoders [8] and DINO V2 [24] models proposed recently, shown to be highly effective in representation learning. Fig. 3 (mid and right) provides their robustness analysis on the ImageNet-C,R datasets alongside the CLIP models. For ImageNet-C, similar to Fig. 3 (left), these models also converge similar to the multi-modal models at the highest severity levels and their relative drop from severity level 1 to 5 is higher. On ImageNet-R, again, multi-modal models perform significantly better than the uni-modal models. These observations are similar to the case seen in Fig. 2.

Empirical Conclusion. From all the plots, it can be observed that multi-modal networks are much better than the unimodal counterpart on ImageNet-R, ObjectNet and ObjectNet-C datasets and can be seen as more robust for these settings. However, they also see a significant drop in accuracies under the perturbations in ImageNet-C, especially at higher accuracy levels. Again all of the architectures used show similar drops in accuracy. Also, zero-shot multi-modal networks seem to be more robust than their linear probe counterpart in the presence of distribution shifts (comparing the drop in accuracy from ImageNet to other datasets). On the architectural side, transformer models appear to be more robust than the ResNets for both single and
4. Methodology

Problem Description. The goal of our work is to make large pre-trained computer vision models robust without sacrificing their original properties. This is critical because we want models to work well both in in-domain and out-of-domain settings. Computational efficiency is also important because pre-training already requires massive amounts of compute, so efficient techniques are more versatile. A popular technique to robustify a model is to use advanced augmentation techniques like aug-mix [12] or pix-mix [14]. This involves fine-tuning model parameters using the augmented dataset and is very effective at improving robustness. However, such fine-tuning is sub-optimal, as models could forget their original representation properties and at the same time require large computation resources.

Method Overview: To this end, we propose a novel Plug-and-Play method. First, alongside the original clean classification head, we plug a robust and a combined head into the model. Second, we induce robustness from a small robust teacher (here the small is relative to the large pretrained model) into the robust head. However, this leaves the challenging task of ensuring that clean accuracy is preserved in the clean head and robustness induced into the robust head. More importantly, we need to ensure that these heads can be correctly selected at test time. Third, we propose a novel uncertainty-aware knowledge distillation technique, which allows us to fulfil all our required objectives. The proposed method is discussed below and also shown in the Fig. 4.

4.1. Augmented Architecture

Let us denote the original model as $M$. The network can be seen as made of three components: $M_b$ the backbone network spanning from initial layer to the $K^{th}$ layer, $M_s$ the shared tunable section spanning $(K+1)^{th}$ layer to $(N-1)^{th}$ layer, and $M_h$ the prediction head section from $N^{th}$ layer till the end (refer figure 4). Thus the overall network can written as:

$$M = M_h \circ M_s \circ M_b,$$

where $\theta$, $\theta_b$, $\theta_s$ and $\theta_h$ denote the respective component parameters.

To address the issue of robustness transfer with preservation of desired characteristics like clean accuracy, transfer learning, etc., we plug-in two more prediction head sections on top of the shared tunable section. This results in a total of three classification heads as shown in figure 4.

4.2. Robustness Distillation from a Small Teacher

At the core of our approach lies use of a knowledge distillation (KD) framework to induce robustness to a large network. In standard KD, knowledge is usually transferred from a large network to a small network. Au contraire, we provide a novel view of KD. We show that robustness can be transferred from a small robust model to large models. For the small robust teacher (denoted as $M_t$), we take small image-classification models and robustify them using standard techniques (a combination of augmentation techniques AugMix [12] and DeepAugment [10]). A small teacher is essential for an efficient method.

4.3. Uncertainty Aware Knowledge Distillation

While we have introduced a robust head and plan to induce robustness from the small model. It is a challenging task to ensure that clean head preserves clean accuracy, robust head learns robustness and the heads are appropriately selected at test time. We next discuss a novel strategy to achieve these goals.

We update the parameters of shared-tunable $\theta_s$ and prediction sections $\theta_h$, keeping the backbone network frozen as shown in figure 4. We use the same augmented training data here as used for robustifying the small (teacher) model. It is denoted as $D^a$ and contains both clean $(x^c, y)$ and
The head section (with parameters $\theta_h$) corresponding to the clean head is updated only using the clean examples. The combined head ($\theta_m$) uses both clean and unclean examples and the unclean head ($\theta_u$) uses only the augmented examples. Thus, for clean examples in a randomly sampled batch of data, the set of updated parameters due to clean head section prediction is $\theta_c^u = \{\theta_s, \theta_h^c\}$ and combined head section is $\theta_c^u = \{\theta_s, \theta_h^m\}$. Similarly for augmented examples, updated parameter set due to unclean head section prediction is $\theta_c^u = \{\theta_s, \theta_h^u\}$ and combined head section is $\theta_c^u = \{\theta_s, \theta_h^u\}$.

Thus, the final loss function to update w.r.t. clean examples is (denoted as $\mathcal{L}_{\text{clean}}$):

$$\mathcal{L}_c(x, y, \theta^c) + \mathcal{L}_d(x, \theta^c, \theta_t) + \mathcal{L}_c(x, y, \theta_m^c) + \mathcal{L}_d(x, \theta_m^c, \theta_t),$$

and similarly for unclean examples (denoted as $\mathcal{L}_{\text{aug}}$):

$$\mathcal{L}_c(x, y, \theta_u^c) + \mathcal{L}_d(x, \theta_u^c, \theta_t) + \mathcal{L}_c(x, y, \theta_m^u) + \mathcal{L}_d(x, \theta_m^u, \theta_t).$$

Finally, the cost function $\mathcal{L}$ for a given batch of data can be written as:

$$\mathcal{L} = \beta \mathcal{L}_{\text{clean}} + (1 - \beta) \mathcal{L}_{\text{aug}},$$

where $\beta = 1$ for clean and $\beta = 0$ for unclean examples.

### 4.3.2 Uncertainty aware head selection

We need a reliable head selection, such that clean head is selected for clean examples (to preserve clean accuracy) and unclean head for shifted examples (robustness).

**Uncertainty.** Modelling uncertainty in predictions corresponding to each of the heads can be a way to select the final head as the most certain head. Clean head should be the most certain on clean examples and similarly unclean for augmented examples. For this, we use Dropout [6] as it gives allows Bayesian approximation. At training time, we update the tunable portion of $\mathcal{M}$, starting from encoder $L^{K+1}$ in Fig.4 using dropout regularization. This can be
done by just setting the dropout rate to some non-zero fraction in the existing implementation. This dropout is a part of all the three heads.

At the inference time, we activate the dropout, for each of the heads, to estimate a predictive distribution from the model as against a point estimate [6, 17]. We take \( K \) forward passes through each of these heads and then for each head we calculate mean and std of the outputs and use them as the mean and std of the predicted distribution from the model. This is referred as Monte Carlo Dropout. Finally, we use std directly as the measure of uncertainty (\( U_{mc} \)).

**KL Divergence.** Now, there can be a case where the clean model completely breaks down for a noisy input and predicts a random output with very low \( U_{mc} \) (highly certain). Given the test-distribution is unknown, this can be a case with significant probability. To handle this, we also calculate the distance between the predicted distributions of each of the clean and unclean head with the combined head using KL divergence only at Inference time. This results in the following objective for selecting the final prediction head \( h_f \):

\[
h_f = \arg \min_{k \in \{c,u\}} U_{mc} \cdot KL(\phi^m_l(x)||\phi^k_l(x)),
\]

where \( h_c, h_m, h_u \) correspond to clean, combined, unclean heads respectively and \( \phi^c_l, \phi^m_l, \phi^u_l \) are the corresponding prediction functions. Thus, the desired head out of clean/unclean is selected using eq. 6. Note, the third head is here just to select the correct desired head for the input from the clean and unclean head via the KL divergence term in the head selection metric in eq. 6. In supplementary, we have provided a detailed ablation on the utility of each of these components and also the comparison against a naive confidence based head selection baseline.

### 4.4. Zero-Shot Multi-Modal scenario

The method described above can be directly applicable to uni-modalities and vision encoders of multi-modalities by attaching a classification head, similar to the Linear Probe setup [27]. We further adapt our scheme to zero-shot setup for these multi-modal networks which comprise both vision and text encoders. For this, we apply our scheme in the presence of text encoder and use the dot products between the vision encoder embeddings (\( \phi_v(x) \)) and the prompt embedding obtained from the text encoder (\( \phi_{text}(Y) \), where \( Y \) denotes the set of prompts corresponding all classes present in the data) , \( \phi_t(x) = \phi_v(x) \cdot \phi_{text}(Y) \), as the model prediction for both classification and distillation losses.

## 5. Experiments

We evaluate the presented approach in making large models robust on the benchmark datasets and perturbations described in the section 3 for image classification task. We show performance on clean accuracy, robust accuracy and transfer learning properties on downstream datasets.

**Experimental Setup.** We demonstrate results of our approach under two settings. First corresponds to using visual encoders of uni or multi-modal models (as described in linear probing approach in the Sec. 5.1) and attaching a classification head to the visual encoders. We term this as the Visual Evaluation or VE setup. Next, we also provide results (Sec. 5.1) in multi-modal models settings where both text and vision are used in the zero-shot (or ZS) setting under dataset shift. Along with clean and robust accuracy, we also compare different methods on whether they can preserve the transfer learning capabilities of the large models. The robust teacher for both settings is a single modal network trained by finetuning complete model using augmentation based techniques using a combination of augmix and deep augmentation techniques (as described in Sec. 4).

**Datasets.** We evaluate the presented approach on the ImageNet validation set and its perturbed variations, ImageNet-C [11], ImageNet-R [10], ImageNet-S [31] and ImageNet-A [13] for robust analysis in the VE setup. Further, we use the ObjectNet data [2] and its perturbed variation ObjectNet-C (Corrupted) version for the zero-shot evaluation tasks or ZS setup. For transfer learning experiment, we show results five datasets (Tiny-ImageNet [21], Flowers [23], PLACES025 [22], iNaturalist2021 [30] and SUN397 [34]) in the VE setup. For the ZS setup, we instead show results on dataset shift on six datasets (Cars [19], Flowers [23], CIFAR100 [20], SUN397 [34], ObjectNet [2]), where the zero-shot model is directly evaluated on these datasets. More information about these datasets have been provided in the supplementary material.

**Baselines and metrics.** We now describe baselines used for the VE and ZS setups. For the VE experiments involving only the visual encoders, we compare against five prior approaches. First approach involves adapting the the same number of parameters as in the presented linear probe approach (Sec. 3). Second baseline involves naive finetuning full network using the current dataset. Third baseline is the visual prompt tuning approach [16] that involves adapting input prompt parameters while fixing the feature network parameters. We also compare against the recently proposed WISE [33] framework for finetuning large networks. Finally, we define a new baseline, Augmentation based Partial Fine-Tuning or APT to show the effectiveness of multi-headed scheme. It involves directly updating the small part of the large network (same number of tunable layers as ours), using the augmentation based technique.

We further define two more baselines, as variations of our proposed scheme, to further highlight its importance.
The first baseline, Only K.D., involves doing knowledge distillation directly from the Small Teacher Network (STN) to Large Learner Network (LLN) student without using the proposed multi-headed architecture and copy of initial LLN as teacher for clean examples. The second baseline, combined head, involves using copy of initial LLN as teacher for clean examples and STN for augmented, but without the multi-headed architecture, i.e., requiring only the combined head.

For the ZS setting, the set of baselines involves the existing zero-shot multi-modal network, along with the APT baseline (ZS-APT) and complete fine-tuning baseline (ZS-Full tuning). Both these baselines are tuned similar to our method, in presence of the text encoder, as described in section 4 (Zero Shot Multi-Modal scenario).

We use the accuracy metric on clean and perturbed dataset to evaluate each of the methods. Furthermore, we also calculate the robustness metrics for evaluating under perturbations, proposed for the ImageNet-C dataset in the supplementary.

### 5.1. Evaluating Multi Modality Trained Methods

#### Visual Evaluation

Table 1 shows the comparison of our method with the baselines under various distribution shifts for ImageNet dataset along with the transfer learning capabilities and training time per image. It uses the CLIP ViT-L/14@333px model for all the methods. Our method uses ViT-B/16 as the teacher model. It can be observed that even though WISE-E2E performs best for two shift scenarios, it suffers from high training time and poor performance under transfer learning tasks (accuracy drop of 1.8 and 1.4) which is a bottleneck with E2E fine-tuning methods. On the other hand, the methods like VPT which require low training time perform poorly under distribution shift (average accuracy drop greater than 5% when compared with Zero-Shot model). On the other hand, performs best on four distribution shift tasks boosting up accuracy upto 2% (ImageNet-C) and is able to achieve the same accuracy as the zero-shot model on the transfer learning experiments and also in a significantly lesser time compared to WISE-E2E (approximately 5 times faster).

We further evaluate our method against the APT baseline, we defined, for various model architectures as Teachers and Learners. Figure 5 (top left and bottom left) shows this analysis, with visual encoder of four CLIP models (RN-50, RN101, ViT-B/16 and ViT-B/32) as students and single modal networks as teacher evaluated on ImageNet-R and ImageNet-C datasets. The rows with teacher as None correspond to the APT baseline. For ImageNet-C, accuracy is improved by more than 3% for most cases, and is at least 2.5% for all cases. This knowledge transfer doesn’t rely much on the architectural family of student or teacher as only marginal difference is there in the improvements offered by ViT or ResNet architectures as teachers on CLIP ViT/ResNet students (less than 0.3% difference observed when updating CLIP RN-50 with ResNet-101 or ViT-Small).

For ImageNet-R, our method provides gains 2.0% for most cases with maximum as 2.9%, compared to APT baseline. For accuracy of the teacher models, please refer supplementary.

#### Zero Shot

We now apply our scheme on multi-modal CLIP networks using both text and visual encoder for a complete zero-shot setup, as discussed in section 4. Table 2 shows the results for our method and baselines under this setup, for various distribution shifts to the ImageNet data and also for zero-shot evaluation on new datasets (dataset shift). Here again, ViT-B/16 is used as the teacher for our method. It can be observed that our method shows best performance under all the distribution shifts with minimum difference of 1% (IN-R) and maximum of 4.5% (IN-C). Also, it is the best performing zero-shot method for four out of five dataset shifts with the maximum improvement being on ObjectNet-C 3.9%). This shows that it improves zero-shot properties of the model as compared to the complete fine-tuning which instead degrades the performance under dataset shift as observed in table 2. Table 5 (top mid
### Table 2. Zero Shot results
Comparison with existing robustification methods and complete fine-tuning on various distribution and dataset shifts under the zero shot setup using CLIP model. Last column shows the average accuracy including all the dataset and distribution shifts. All these results are a result of zero-shot evaluation of the robustly tuned classifier model using only the ImageNet-1K dataset. No further tuning on any of the other datasets.

<table>
<thead>
<tr>
<th>Method</th>
<th>IN</th>
<th>IN-V2</th>
<th>IN-R</th>
<th>IN-S</th>
<th>ON</th>
<th>IN-A</th>
<th>IN-C</th>
<th>Cars</th>
<th>Flowers</th>
<th>CIFAR100</th>
<th>SUN397</th>
<th>ON-C</th>
<th>Avg. shifts</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLIP ViT-L/14@336px Zero-Shot</td>
<td>76.2</td>
<td>70.1</td>
<td>88.9</td>
<td>60.2</td>
<td>70.0</td>
<td>77.2</td>
<td>60.6</td>
<td>78.8</td>
<td>78.3</td>
<td>77.5</td>
<td>68.4</td>
<td>52.2</td>
<td>71.1</td>
</tr>
<tr>
<td>ZS-Full Tuning</td>
<td>86.5</td>
<td>77.1</td>
<td>88.2</td>
<td>58.9</td>
<td>65.5</td>
<td>78.2</td>
<td>53.3</td>
<td>76.3</td>
<td>76.8</td>
<td>77.1</td>
<td>67.2</td>
<td>51.5</td>
<td>70.0</td>
</tr>
<tr>
<td>ZS-APT</td>
<td>84.8</td>
<td>76.3</td>
<td>89.2</td>
<td>60.7</td>
<td>68.8</td>
<td>77.9</td>
<td>62.3</td>
<td>77.2</td>
<td>77.9</td>
<td>77.3</td>
<td>67.9</td>
<td>54.3</td>
<td>71.8</td>
</tr>
<tr>
<td>Ours (ViT-B/16 teacher)</td>
<td>86.3</td>
<td>76.8</td>
<td>90.2</td>
<td>62.9</td>
<td>71.6</td>
<td>74.6</td>
<td>78.9</td>
<td>78.9</td>
<td>78.1</td>
<td>78.1</td>
<td>69.3</td>
<td>58.2</td>
<td>73.6</td>
</tr>
</tbody>
</table>

Table 3. Unimodal results. Comparison with existing robustification methods and complete fine-tuning on various distribution shift and transfer learning benchmarks for single-modal ViT-L/14 model, pretrained on JFT dataset. Second last column shows average accuracy over the distribution shifts and the last one shows training latency per image.

<table>
<thead>
<tr>
<th>Method</th>
<th>IN</th>
<th>IN-V2</th>
<th>IN-R</th>
<th>IN-S</th>
<th>ObjectNet</th>
<th>IN-A</th>
<th>IN-C</th>
<th>Tiny-IN</th>
<th>Flowers</th>
<th>Avg. shifts</th>
<th>Latency (ms/img)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ViT-L/14 Standard Training</td>
<td>82.8</td>
<td>75.3</td>
<td>49.4</td>
<td>40.4</td>
<td>45.2</td>
<td>51.9</td>
<td>51.5</td>
<td>84.5</td>
<td>98.1</td>
<td>52.3</td>
<td>2200</td>
</tr>
<tr>
<td>VPT [16]</td>
<td>82.1</td>
<td>74.4</td>
<td>47.2</td>
<td>38.1</td>
<td>45.3</td>
<td>51.2</td>
<td>50.3</td>
<td>84.1</td>
<td>97.9</td>
<td>51.1</td>
<td>420</td>
</tr>
<tr>
<td>WISE-FT (LC, optimal α) [33]</td>
<td>82.6</td>
<td>76.2</td>
<td>54.2</td>
<td>48.3</td>
<td>49.2</td>
<td>55.3</td>
<td>54.2</td>
<td>84.0</td>
<td>98.1</td>
<td>56.2</td>
<td>500</td>
</tr>
<tr>
<td>WISE-FT (E2E, optimal α) [33]</td>
<td>83.6</td>
<td>78.4</td>
<td>58.3</td>
<td>52.1</td>
<td>52.3</td>
<td>57.7</td>
<td>55.1</td>
<td>83.8</td>
<td>97.5</td>
<td>58.9</td>
<td>3400</td>
</tr>
<tr>
<td>K.D. from ViT-Small teacher</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Single-Teacher</td>
<td>81.9</td>
<td>75.9</td>
<td>52.7</td>
<td>46.3</td>
<td>47.4</td>
<td>53.2</td>
<td>53.3</td>
<td>83.8</td>
<td>97.2</td>
<td>54.8</td>
<td>470</td>
</tr>
<tr>
<td>Only K.D.</td>
<td>82.2</td>
<td>76.7</td>
<td>53.4</td>
<td>46.9</td>
<td>48.1</td>
<td>54.2</td>
<td>53.2</td>
<td>84.3</td>
<td>97.9</td>
<td>55.4</td>
<td>550</td>
</tr>
<tr>
<td>Ours</td>
<td>82.7</td>
<td>78.2</td>
<td>59.1</td>
<td>52.7</td>
<td>51.6</td>
<td>58.5</td>
<td>58.3</td>
<td>84.5</td>
<td>98.1</td>
<td>59.7</td>
<td>700</td>
</tr>
</tbody>
</table>

#### 5.2. Evaluating Unimodally Trained Methods
We now analyze our method for a unimodal Learner network scenario, comparing it with all the baselines on different distribution shifts/transfer learning tasks, as done for the multi-modal VE setup. Table 3 shows this comparative results. Here again, our method emerges as the winner for four out of the six distribution shifts with minimum gain of 0.8% and maximum gain of 3.2%. Furthermore, it is also able to preserve transfer learning capabilities, as shown in table 3 for Tiny-IN, Flowers, PLACES205, iNaturalist2021, SUN397 datasets, of the original model whereas other baselines (VPT and WISE) suffer.

Figure 5 (last column) shows this comparison for various Leaner-Teacher pairs consisting both ViT and ResNet architectures. Again, the rows with Teacher None correspond to the APT baseline. For majority cases on ImageNet-C, our method improves accuracy by greater than 3 percent, when compared to the baseline. Similarly, on the ImageNet-R dataset, it shows greater than 5% for most cases with maximum going to 3.2% for RN-50 teacher and RN-101 student. Furthermore, increasing the size of teacher model (from RN-34 to 50) results in improved performance. Finally, both our method and the baseline we compare to, make significant improvement in performance on the perturbed datasets (especially ImageNet-C), compared to the

**Inference time Latency.** Since our method attaches extra heads, although quite small, on top of the existing model, we also analysed the inference time overhead it adds on top of the naive model. We observed that this overhead was quite small. For instance, CLIP ViT-L/14 model used in table 1, GFLOPs for baselines is 81.17 and ours is 83.7, (3% overhead).
Figure 5. Analysis of our methods for various Teacher (RN-34, RN-50)-Student pairs and the APT baseline (None) applied to students. For Zero-Shot, None corresponds to the given zero-shot model without any tuning. x-axis: Students, y-axis: Accuracy, vertical bars: Teacher.

Table 4. Ablation of various robustification schemes used to robustify the teacher used in our approach. Here, a ResNet-101 is the student and ResNet-50 is the teacher.

<table>
<thead>
<tr>
<th>Method</th>
<th>ImageNet</th>
<th>ImageNet-C</th>
<th>ImageNet-R</th>
</tr>
</thead>
<tbody>
<tr>
<td>Augmix [12]</td>
<td>78.3</td>
<td>55.2</td>
<td>49.1</td>
</tr>
<tr>
<td>Augmix+DA [10]</td>
<td>78.9</td>
<td>56.1</td>
<td>49.8</td>
</tr>
<tr>
<td>PixMix [14]</td>
<td>79.1</td>
<td>56.8</td>
<td>49.9</td>
</tr>
<tr>
<td>AugMix+PixMix</td>
<td>79.0</td>
<td>57.1</td>
<td>50.9</td>
</tr>
</tbody>
</table>

Table 5. Ablation showing variation in model performance as a function of its proportion of tuned parameters using our method. Here, ResNet-101 is the student and ResNet-50 is the teacher.

<table>
<thead>
<tr>
<th>Fraction-tuned</th>
<th>ImageNet</th>
<th>ImageNet-C</th>
<th>ImageNet-R</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>77.5</td>
<td>39.2</td>
<td>40.1</td>
</tr>
<tr>
<td>0.05</td>
<td>77.4</td>
<td>52.9</td>
<td>46.5</td>
</tr>
<tr>
<td>0.10</td>
<td>77.2</td>
<td>55.1</td>
<td>47.7</td>
</tr>
<tr>
<td>0.20</td>
<td>76.4</td>
<td>57.2</td>
<td>48.8</td>
</tr>
<tr>
<td>0.25</td>
<td>76.2</td>
<td>57.4</td>
<td>49.2</td>
</tr>
<tr>
<td>0.30</td>
<td>75.9</td>
<td>57.5</td>
<td>50.1</td>
</tr>
</tbody>
</table>

5.3. Ablations

Robustification schemes. We now compare the effect of using a different robustification schemes for the teacher model, used in our method. We limit ourselves to augmentation based robustification schemes such as AugMix, PixMix, DeepAugment etc. Table 4 shows the results for this comparison for the single modal setup when a ResNet-101 student model is using ResNet-50 teacher. Using PixMix or combining with AugMix does improves the accuracy by around 1-1.5 over our current technique (Augmix+DeepAugment). Our scheme shows that that gains in teacher model can be efficiently transferred to the student.

Amount of parameters tuned. We further analyze the effect of tuning different proportions of the LLN using our scheme. It can be observed that tuning more parameters increases the accuracy on ImageNet-C and R datasets at the cost of clean accuracy on ImageNet dataset.

Please refer to the supp. material for more description about the ablations for understanding model design choices (KL Div., Uncertainty, Multiple-heads) and other details.

6. Conclusion

We first benchmark and discuss existing large pre-trained models under various shifts to the data. Following this, we proposed an effective method to distill robustness to these networks via small robust models, at the same time preserving the characteristics of the large models. Results on various distribution shift settings showed that our method is effective and efficient in making large pretrained models robust to several distribution shifts, and also retaining their transfer learning properties.

Limitations. Though we have provided extensive empirical evidence to demonstrate the benefit of our approach, a theoretical underpinning is missing. We leave theoretical analysis as an interesting future work.

References


[37] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019. 2

