Training Vision Transformers on Corrupted Data Using Knowledge Distillation

Hasan Lokhandwala (2774699)
Main Supervisor: Bob Borsboom
Second Reader: Koen Hindriks

Vrije Universiteit Amsterdam

Abstract. Robustness to image corruptions remains a core challenge for vision models trained on clean data. Despite substantial progress in architectural improvements and training schemes, achieving consistently high performance across diverse corruptions remains elusive. This work proposes a complementary approach: training directly on corrupted images using knowledge distillation from large, cleanly pretrained teacher models.

Our baseline student model is a plain, non-distilled ViT baseline which achieves 96.4% accuracy on corrupted images from CIFAR-10-C. We adopt the Data-efficient Image Transformer (DeiT) as the standard student model, leveraging its distillation framework and data efficiency. This model achieves 96.7% accuracy outperforming the baseline.

We further propose a novel DeiT variant with a corruption-aware token trained to predict type of corruption an image has. This model reaches 96.6% accuracy, surpassing the standard DeiT without increasing model size.

Keywords: Computer Vision · Transformer · Knowledge Distillation · Classification

1 Introduction

Image classification has been a central part of computer vision since the breakthrough success of CNNs in the 2012 ImageNet challenge [7]. Today, image classification is integral in areas such as medical imaging [11], self-driving cars [10], criminal tracking [9], and secure banking [8]. In these critical domains, models need to be highly reliable. But in reality, images are not always clean; they can be blurry, noisy, compressed, or distorted [31]. These issues are types of covariate shifts [12] that alter visual patterns and reduce model reliability [13, 3, 6]. For example, surveillance footage can obstruct identification of criminals or vehicles. Similarly, as shown in [3], medical imaging models trained on data from one hospital may fail on data from another due to hardware-induced noise – a trivial problem for human vision [14].

In such scenarios, models must be both accurate and well-calibrated to be trustworthy. Yet, despite extensive robustness research [6, 15, 16], most still

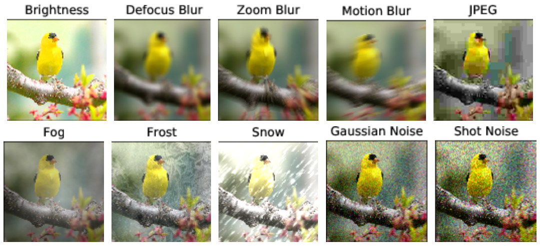

Fig. 1. Types of image corruptions in our dataset. Sourced from Hendrycks et al. (2019).

struggle under corruption. For example, MViTv2-L, one of the top-performing models in [3], achieving 0.977 ECE and 41.91% mCE on the ImageNet-1K-C benchmark is strong, but still not fully reliable. So, we try an unconventional approach: instead of training on clean data alone, we use a mix of clean and corrupted images. This is uncommon due to two main hurdles: 1) lack of large labeled corrupted datasets [6], and 2) wide variety in real-world corruptions [17]. This work addresses the first one by using knowledge distillation from a model trained on large-scale clean data.

To this end, we employ the "tiny" variants (5M parameters) of Data-efficient Image Transformers (DeiT-Tiny) [2] as our student models. These are vision transformers explicitly optimized for small datasets using knowledge distillation [20], aligning well with our goal. Vision Transformers (ViTs), introduced by Dosovitskiy et al. (2020) [1], gained significant attention after beating the state-of-the-art on benchmarks like ImageNet-1K with 88.55% accuracy. Their rise in popularity has also led researchers to delve into ViT robustness studies [4, 5]. In particular, [3] shows that ViTs outperform CNNs of similar scale in robustness. Consequently, we adopt DeiT-III Small [18], pretrained on ImageNet-21K [19] and fine-tuned on ImageNet-1K, as our teacher model for its large-scale training, subsequent robust representations, and hence, the capacity to effectively distill into students trained on corrupted data. Our experiments explore various augmentation strategies, distillation techniques, architectural modifications, model calibration, and performance across diverse corruption types.

Our paper is organized as follows: **Section 2** provides a brief overview of Vision Transformers and knowledge distillation to support reader comprehension. **Section 3** details the experimental setup, evaluation metrics, and introduces our corruption-aware image transformer. **Section 4** presents experiments and ablations on distillation strategies, data augmentations, regularization techniques, and hyperparameters. **Note:** Our work is stylistically and experimentally inspired by Touvron et al. (2020).

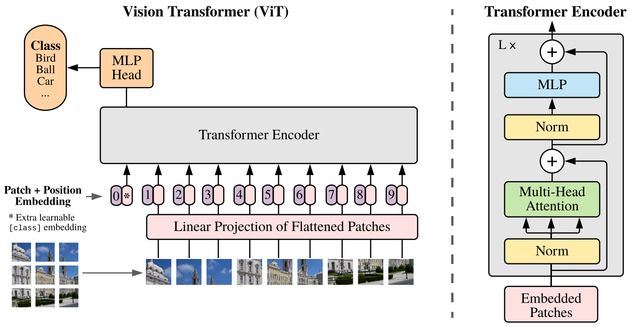

Fig. 2. Overview of the Vision Transformer architecture from Dosovitskiy et al (2020).

2 Background

2.1 Vision Transformer (ViT)

The Transformer, introduced by Vaswani et al. (2017) [21], has become the de facto standard for large-scale NLP tasks due to its fully attention-based mechanism and high parallelizability on GPUs. It offers significant computational efficiency and models long-range dependencies more effectively than RNN and LSTM-based architectures. Motivated by these advantages, Dosovitskiy et al. (2020) [1] extended the transformer architecture to vision tasks, proposing the Vision Transformer (ViT) which achieved state-of-the-art performance on benchmarks like ImageNet-1K, CIFAR-10, and CIFAR-100.

ViT treats an image as a sequence of patches, similar to word tokens in NLP. An image of size $H \times W \times C$ is split into $N = HW/P^2$ non-overlapping patches of size $P \times P$, each flattened to a vector in $\mathbb{R}^{C \cdot P^2}$ and then linearly projected into a fixed-dimensional embedding space. Here H,W,C and P represent the height, width, number of channels and patch size respectively. A special classification [CLS] token is prepended to these embeddings, and learnable positional encodings are added element-wise to preserve spatial order of patches. The resulting sequence of embeddings is then passed through n standard Transformer encoders, each composed of stacked layers of multi-head self attention and feed-forward networks.

Multi-head self attention (MHSA). This is the core operation in transformers, enabling the model to attend to different parts of an image regardless of their spatial distance. It allows each token in the sequence to attend to every other token and capture global context of the image. Given an input sequence of patch embeddings $X \in \mathbb{R}^{N \times D}$, where N is the number of tokens and D is the embedding dimension, the input is first projected into three matrices

$$Q = XW^Q, \quad K = XW^K, \quad V = XW^V$$

Where, $W^Q, W^K, W^V \in \mathbb{R}^{D \times d_k}$ are learned projection weights, and Q, K, and V are the **query**, **key**, and **value** matrices used to compute attention scores:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{1}$$

The input is split into h such heads, each of dimensionality $d_k = D/h$, and outputs of all heads are concatenated and passed through the linear projection $W^O \in \mathbb{R}^{D \times D}$:

$$\text{MHSA}(X) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{2}$$

Feed-forward network (FFN). The output of the MHSA layer is passed into a small feed-forward network consisting of two linear layers typically separated by a ReLU or GELU activation. Formally:

$$\text{FFN}(x) = \text{GELU}(xW_1 + b_1)W_2 + b_2 \tag{3}$$

Residual connections and layer normalization. Layer normalization normalizes inputs across the embedding dimension to stabilize training. Unlike the original Transformer that applies it after each layer (post-norm), ViTs use pre-normalization by applying it before self-attention and feed-forward layers.

Residual connections add the input of a layer directly to its output, helping preserve information and enable better gradient flow during training.

Class token and MLP head. The class [CLS] token is a learnable vector prepended to the sequence of patch embeddings before being input to the Transformer encoder. Throughout the attention layers, this token interacts with all other tokens and gradually aggregates global contextual information about the image. By the end of the encoder, it serves as a representation of the entire image and is hence fed into a final two-layered MLP/FFN head with a GELU activation in between to produce the final predictions.

2.2 Knowledge Distillation for Classification

In a standard classification task, the model learns by minimizing the loss between its predictions and the ground-truth labels. Knowledge Distillation (KD), introduced by Hinton et al. (2015) [20], proposes a paradigm where the model learns by minimizing the loss between its outputs and those of a "teacher" model. The teacher is typically a larger model pretrained on massive datasets covering generic domains. The model being trained, called the "student", is trained on a smaller domain specific dataset. It benefits by absorbing the teacher's learned representations without directly training on the large dataset, saving extensive computation costs. We use KD in combination with the standard true label loss in our experiments. This technique has shown better results on domain-specific tasks compared to training solely on ground truth labels [22, 2]. We experiment with two types of distillation techniques that have been well covered by Touvron et al. (2020):

Soft Distillation. The objective is to minimize the cross-entropy CE loss with the true labels and the Kullback–Leibler KL divergence loss with the softmax of the teacher's outputs. Let Z_s and Z_t be the output logits of the student and teacher models respectively, y the true labels, ψ the softmax function, τ the temperature of distillation, and λ the loss balancing factor. Then, the soft distillation loss is defined as:

$$\mathcal{L}_{\text{soft}} = (1-\lambda)\mathcal{L}_{CE}(\psi(Z_s), y) + \lambda\tau^2\mathcal{L}_{KL}(\psi(Z_s/\tau), \psi(Z_t/\tau)) \tag{4}$$

Higher values of τ produce softer probability distributions, helping the student model better capture the teacher's knowledge. Our experiments use $\tau = 3$ and $\lambda = 0.5$.

Hard Distillation. The teacher's outputs are treated as true labels (hard targets) by taking the *argmax* of its predicted logits. The student is then trained using standard cross-entropy loss with both the true labels and the teacher's predicted hard targets. This simplifies the distillation process by avoiding soft probability distributions. The hard distillation loss is defined as:

$$\mathcal{L}_{\text{hard}} = \frac{1}{2}\mathcal{L}_{CE}(\psi(Z_s), y) + \frac{1}{2}\mathcal{L}_{CE}(\psi(Z_s), \arg\max(Z_t)) \tag{5}$$

2.3 Data-Efficient Image Transformer (DeiT)

While Vision Transformers (ViTs) have demonstrated strong performance across various benchmarks, their success has depended on training with massive datasets (e.g., JFT-300M).To address this limitation, Touvron et al. (2020) [2] introduced Data-efficient Image Transformers (DeiT) which use knowledge distillation to train ViTs on smaller datasets like ImageNet-1K without heavy pretraining. Their base models (86M parameters) achieved up to 84.5% top-1 accuracy on ImageNet-1K compared to ViT-B which got 77.9% top-1 accuracy and slightly surpassing state-of-the-art. A key difference in this paper is that while [2] used CNNs (RegNet) as teachers, our setup uses a DeiT-III transformer as the teacher.

DeiT introduces two key architectural modifications to the standard ViT. First, it adds a second learnable token called the distillation token, in addition to the [CLS] token. While the [CLS] token is supervised using the ground truth

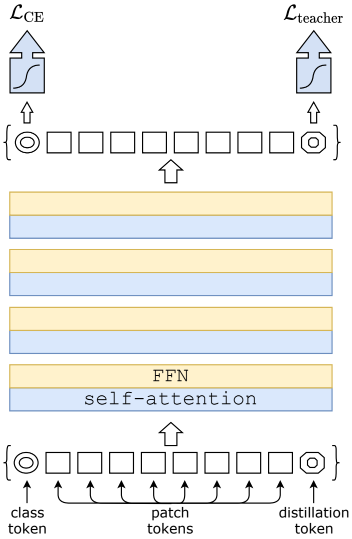

Fig. 3. Overview of the Data-Efficient Image Transformer architecture from Touvron et al. (2020). The class token is supervised by the true label while the distillation token is supervised by the teacher's outputs.

label, the distillation token is used for knowledge distillation using the predictions of the teacher model. Second, instead of using a two-layer MLP head, both tokens are passed through respective single linear classification heads. During inference, the outputs from the class token and the distillation token are combined by taking the average of their logits. Touvron et al. (2020) demonstrate that separating the roles of label prediction and knowledge distillation across two tokens rather than overloading the [CLS] token improves performance: their DeiT-Ti achieves 73.9% accuracy on ImageNet-1K when using the [CLS] token for both classification and distillation. Using separate [CLS] and distillation tokens improves this to 74.5%.

3 Methods

3.1 Data Augmentation and Regularization

Since the DeiT is designed to work with datasets of limited size, it heavily relies on strong data augmentation techniques. All augmentation implementations used in our experiments are sourced from the PyTorch library [23], and we adopt a subset of the strategies used by Touvron et al. (2020).

RandAugment [24] applies a random combination of image transformations (e.g., rotation, translation, color jitter) with a fixed strength (magnitude 9) and number of operations (2). Mixup [26] generates new training samples by linearly interpolating pairs of images and their corresponding labels. This encourages the model to behave linearly between training examples, improving generalization and robustness. CutMix [27] replaces a random patch of an image with a patch from another image, and the corresponding labels are mixed proportionally. This helps the model learn to associate objects with their spatial context.

The regularization techniques we employ to reduce overfitting are also adopted from Touvron et al. (2020). Random Erasing [25] randomly removes rectangular regions of the input image during training. This forces the model to rely on less discriminative features and improves generalization. Dropout [29] randomly sets a fraction of neuron activations to zero during training, preventing the network from relying too heavily on any single feature. Stochastic Depth [28] randomly skips some encoder blocks with a fixed probability, acting as a form of layer-wise dropout in deep models.

3.2 Models.

As mentioned earlier, we adopt the DeiT-III Small model [18] as our teacher network due to its strong robustness and large-scale pretraining on ImageNet-21k, followed by fine-tuning on ImageNet-1k. It has an embedding dimension $D = 384$, number of attention heads $h = 6$, and 12 encoder blocks, totaling approximately 22 million parameters.

Our student models are based on the DeiT-Tiny architecture from Touvron et al. [2], with $D = 192$, heads $h = 3$, and 12 encoders, resulting in a total of approximately 5 million parameters. The baseline model is a plain ViT with the same parameters, except that it does not have a distillation mechanism. Further, the corruption-aware DeiT student is explained in the next section.

3.3 CdeiT: Our Novel Architecture.

We propose CdeiT, a corruption-aware variant of the DeiT architecture in [2]. This model introduces a third special token in addition to the class and distillation tokens - the corruption token. This token is trained to predict the type of corruption present in the image. Like the other tokens, it has the same embedding dimension (192), but its output logits at the end of the model are in a separate prediction space, as the number of corruption types differs from the number of classes. This design aims to help the model develop specialized internal representations for each corruption type without increasing overall model size.

For clarity, we refer to the baseline as ViT, DeiT-Tiny model as DeiT, our corruption-aware variant as CdeiT and the teacher model as DeiT3 throughout the remainder of this paper.

In the standard DeiT, class and distill token logits are combined by simple averaging. However, in CdeiT, the corruption token produces logits in a different dimension and thus cannot be element-wise averaged with the other tokens. To address this, we propose three strategies for integrating corruption token information at inference:

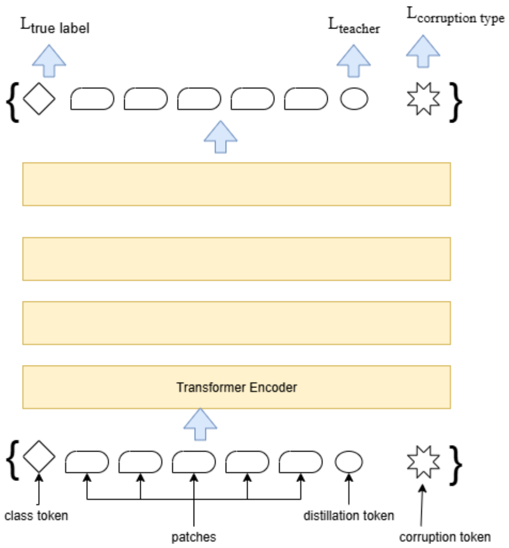

Fig. 4. The CdeiT architecture Overview. The additional corruption token is supervised by the type of corruption. Rest of the model is the same as the DeiT proposed by Touvron et al. (2020).

1. No Integration. The corruption token is excluded from the final prediction. It functions solely as an internal modifier, influencing the model's representations of images during self-attention. At inference, predictions are made exactly as in the standard DeiT, i.e, by averaging the logits of the class and distillation tokens without any contribution from the corruption token.

2. Single-Layer FFN Head. The output of the corruption token is passed through a single linear layer to project it into the class token space. It is then averaged with the logits from the class and distillation tokens to form the final prediction. This method incorporates corruption-specific information with minimal computational cost.

3. Two-Layer FFN Head. This strategy extends the single-layer approach by using a two-layer feedforward network with a GELU activation in between to project the corruption token into class space. Again, the output logits are averaged with the class and distillation logits.

Training for strategies 2 & 3. With Strategy 1, CdeiT follows the standard DeiT training setup, where the loss is the average of the individual losses of the three tokens. However, in Strategies 2 and 3, because the corruption token is projected into the class space through the FFN, we introduce a new hard-label loss to train this projection head. This is done by taking the cross entropy loss of the sum of the [CLS] logits and the logits from the FFN.

Let Z_c be the corruption token's logits, Z_d the distillation token's logits, Z_s the [CLS] token's logits and, $\hat{c}$ the type of corruption. The rest of the notation follows Section 2.2:

$$\begin{aligned}\mathcal{L}_{\text{hard}} = \frac{1}{3}\mathcal{L}_{\text{CE}}(\psi(Z_s + \text{FFN}(Z_c)), y) + \frac{1}{3}\mathcal{L}_{\text{CE}}(\psi(Z_d), \arg\max(Z_t)) \\ + \frac{1}{3}\mathcal{L}_{\text{CE}}(\psi(Z_c), \hat{c}) \qquad (6)\end{aligned}$$

Similarly we also introduce a new soft-label loss for the two strategies:

$$\begin{aligned}\mathcal{L}_{\text{soft}} = (1-\lambda)\frac{\mathcal{L}_{\text{CE}}(\psi(Z_s + \text{FFN}(Z_c)), y) + \mathcal{L}_{\text{CE}}(\psi(Z_c), \hat{c})}{2} \qquad (7) \\ + \lambda\tau^2\mathcal{L}_{\text{KL}}(\psi(Z_d/\tau), \psi(Z_t/\tau))\end{aligned}$$

3.4 Training Setup

Training, Validation and Test Data. We use CIFAR-10 and CIFAR-10-C 6 to construct our dataset with 11 image types. It includes clean and 10 types of corruptions: brightness, defocus blur, zoom blur, motion blur, fog, frost, snow, shot noise, Gaussian noise, and JPEG compression. These are mixed uniformly in the dataset to ensure diversity during training.

The training set includes 495k images (4,500 per class per corruption), and validation/test sets each contain 22k images (200 per class per corruption). Hence, the dataset is balanced for each corruption type as well as for each class. During training, a resized 224×224 copy of each image is created for the teacher model before augmentations are applied. All images are normalized before being passed to the models.

Hyperpararameters. Table 1 summarizes the hyperparameters that trained our best-performing DeiT. While our choices are largely inspired by Touvron et al. [2], we apply a few adjustments tailored to our setting such as reducing the Mixup probability and warmup epochs, and omitting some of the augmentations they used.

3.5 Evaluation Metrics

In addition to standard classification metrics such as top-1 accuracy and loss, we evaluate model reliability and robustness using the following metrics:

Methods	Touvron et al. (2020) [2]	Ours
Epochs	300	15
Batch size	1024	1024
Optimizer	AdamW	AdamW
learning rate	$5e-4\times\frac{\text{batchsize}}{512}$	$5e-4\times\frac{\text{batchsize}}{512}$
Learning rate decay	cosine	cosine
Weight decay	0.05	0.05
Warmup epochs	5	3
Label smoothing ε	0.1	0.1
Dropout	✗	✗
Stoch. Depth	0.1	0.1
Mixup prob.	0.8	0.3
Erasing prob.	0.25	0.25

Table 1. Comparison of hyper-parameters for our method and the one used by Touvron et al. (2020)

Expected Calibration Error (ECE). ECE [30] measures how well a model's predicted probabilities reflect the true likelihood of being correct. In simple terms, it compares the model's confidence in its predictions to its actual accuracy. For example, if a model assigns 90% confidence to a group of predictions, it should be correct on about 90% of them. If it's only right 70% of the time, it's overconfident and miscalibrated. Lower ECE values indicate more calibrated (i.e., reliable) predictions. Unlike accuracy, which only reflects correctness, ECE helps assess the trustworthiness of the model's confidence—especially useful under distribution shifts such as image corruptions.

Formally, test predictions are divided into M bins based on their predicted confidence values. For each bin B_m, we compute the accuracy $\mathrm{acc}(B_m)$ and the average confidence $\mathrm{conf}(B_m)$ where

$$\mathrm{conf}(B_m) = \frac{1}{|B_m|}\sum_{i\in B_m}\hat{p}_i \tag{8}$$

with $\hat{p}_i$ = probability of the model for sample i in bin B_m. Then, the Expected Calibration Error is calculated as:

$$\mathrm{ECE} = \sum_{m=1}^{M}\frac{|B_m|}{n}\left|\mathrm{acc}(B_m) - \mathrm{conf}(B_m)\right| \tag{9}$$

where M is the number of bins (15 in our experiments), B_m is the set of predictions falling into bin m, n is the total number of test samples, $\mathrm{acc}(B_m)$ is the accuracy within bin m, and $\mathrm{conf}(B_m)$ is the average predicted confidence in that bin.

Table 2. Ablation studies with accuracy (in%) on learning rates and weight decays for DeiT. The first row corresponds to the best performing configuration. Augmentations used for these experiments are based on the first row of Table 4.

LR	Weight Decay	Acc.
5e-4	0.05	96.7
5e-4	0.01	96.4
3e-4	0.05	96.2
3e-4	0.01	96.3

Table 3. Comparison of the two distillation types for each model with accuracy and mCE(in%). Soft distillation works better for DeiT while CdeiT with Hard distillation beats all configurations.

Model	Distill. Type	Acc.	mCE
DeiT	Hard	96.7	1.93
DeiT	Soft	97.0	0.46
CdeiT	Hard	97.5	2.10
CdeiT	Soft	96.7	2.94

Mean Corruption Error (mCE). mCE quantifies a model's robustness by comparing its classification error on corrupted images to that of a baseline model, averaged over multiple corruption types [3]. Formally, mCE is given as:

$$\mathrm{mCE} = \frac{1}{|C|} \sum_{c \in C}^{C} \frac{\sum_{s=1}^{5} \mathrm{E}_{s,c}^{f}}{\sum_{s=1}^{5} \mathrm{E}_{s,c}^{baseline}} \tag{10}$$

where C = set of corruptions in the test data and $\mathrm{E}_{s,c}^{f}$ = error rate of a model f for corruption type c with severity s.

4 Experiments and Ablation Studies

Before detailing our experiments, it is important to describe the baseline model, which achieves 96.4% ± 0.1 accuracy, 0.0002 ECE, and 0.36 loss, indicating strong calibration. We vary learning rates and weight decays in Table 2, observing only minor accuracy changes over the configurations. Table 3 shows mCE for DeiT and CdeiT under different distillation methods. While CdeiT yields higher accuracy, DeiT, especially with soft distillation achieves significantly lower mCE ($\approx 4.5\times$ lesser). This gap stems from CdeiT's added corruption-based pretraining, which reduces its reliance on soft targets, and from hard distillation being less effective at transferring robustness.

4.1 Fine-Tuning Teacher Head

To leverage DeiT3 as a teacher, its output must align with the class space of our dataset. The original DeiT3 is fine-tuned on ImageNet-1k and thus outputs 1000-dimensional logits, which is incompatible with CIFAR-10's 10 classes. To address this, we freeze the entire model and train a new classification head consisting of a single linear layer producing 10-dimensional outputs.

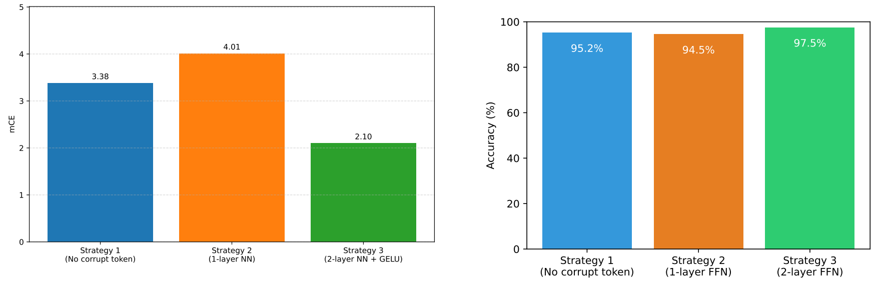

Fig. 5. Comparison of strategies to integrate corruption token information in CdeiT at inference. Left: mCE across different strategies and Right: accuracy across the strategies.

We observed that training without any data augmentations yielded the best performance, achieving 87.4% accuracy. Even light augmentations such as random erasing reduced accuracy to 86.8%. Training was done using a learning rate of 5e-5 and weight decay of 0.05. Reducing the learning rate to 3e-5 also decreased accuracy to 86.7%. Across all configurations, the Expected Calibration Error (ECE) remained low at 0.0001, indicating the teacher's predictions are highly well-calibrated.

4.2 Data Augmentation and Regularisation

We experiment with the techniques discussed in Section 3.1. Due to the small resolution of CIFAR-10 images (32×32), consistent high accuracy and very low ECE ($\approx$ 1e-3) is observed across most configurations. As shown in Table 4, Random Erasing notably improves performance, while CutMix significantly degrades accuracy, even at low application probabilities. RandAugment achieves comparable performance to our best configuration, but incurs a $\sim$10% increase in training time.

The SGD optimizer with momentum = 0.9 performs substantially worse than AdamW, due to the absence of adaptive learning rate used in AdamW which helps stabilize training on noisy data where gradient magnitudes can vary significantly. Overall, these findings suggest that applying excessive augmentation on already corrupted data is counterproductive and conflicts with the corruption distribution, confusing the model.

4.3 CdeiT Heads

The best strategy we found for incorporating corruption token information at inference was Strategy 2, described in Section 3.3. This approach yielded the lowest mCE (See Figure 5) and the highest accuracy of **97.5%** ± 0.3 across all tested configurations. All experiments in this work involving CdeiT utilize this inference strategy.

Ablation on	Optimizer	RandAug	Mixup	CutMix	Erasing	Dropout	Stoch. Depth	Accuracy (%)
DeiT (Best-Performing)	AdamW	✗	✓	✗	✓	✗	✓	96.7 ± 0.2
	SGD	✗	✓	✗	✓	✗	✓	23.0
Data Aug.	AdamW	✓	✓	✗	✓	✗	✓	96.7
	AdamW	✗	✗	✗	✓	✗	✓	96.3
	AdamW	✗	✓	✓	✓	✗	✓	95.5
	AdamW	✗	✗	✓	✓	✗	✓	95.7
Regularization	AdamW	✗	✓	✗	✗	✗	✓	95.2
	AdamW	✗	✓	✗	✓	✓	✓	96.5
	AdamW	✗	✓	✗	✓	✗	✗	96.2

Table 4. Ablation study on augmentation and regularization techniques for training DeiT on corrupted data. Inspired by Touvron et al. (2020). The top row corresponding to "Best-Performing" is the configuration we use in later experiments. Symbols: ✓= used, ✗= not used. Red color indicate a deviation from the best-performing configuration.

4.4 Model Comparisons

For both DeiT and CdeiT models, the error rates across all corruption types remain consistently low, typically below 0.02, indicating strong robustness to various corruptions. Interestingly, clean images exhibit a comparatively higher error rate of around 0.25 for both models. This unexpected outcome suggests that the models may have implicitly deprioritized clean patterns during training, potentially due to overfitting to the corrupted distributions.

At the beginning of training, the cosine similarity between the CLS and distillation tokens is low for both DeiT (0.10) and CdeiT (0.13). By the end of training, the similarity reaches 0.54 for DeiT and 0.84 for CdeiT. Interestingly, the distillation token in CdeiT becomes more aligned with the CLS token compared to DeiT. This may be due to the corruption-aware nature of CdeiT, which benefits from richer supervision, allowing both tokens to converge on more shared semantic information. Meanwhile, the cosine similarity between the CLS and corrupt tokens in CdeiT remains low (0.21), which is expected, as the corrupt token is optimized for a distinct auxiliary task and encodes different feature signals.

We analyze the models' losses to understand their training behavior and how it relates to final performance. Figure 6 shows the hard-label loss curves for DeiT, CdeiT, and the ViT baseline. Losses for the distilled models converges to lower values while the ViT's loss remains higher. This confirms that corruptions negatively impact model training, while distillation mitigates this effect by

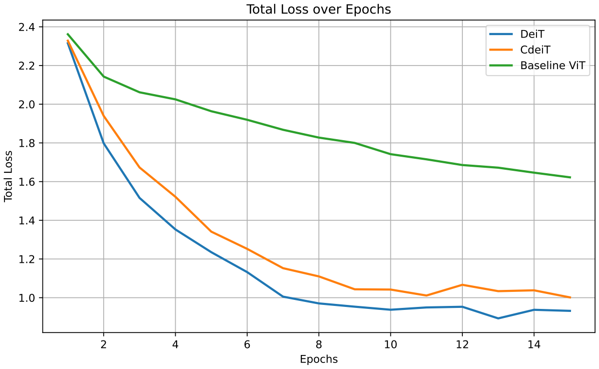

Fig. 6. Total hard-label losses of DeiT and CdeiT during training.

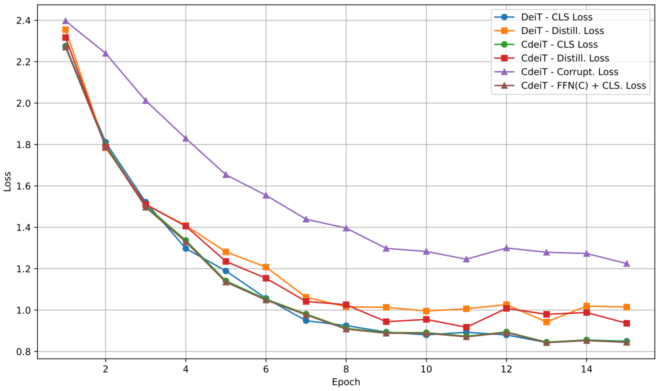

Fig. 7. Cross-Entropy losses of individual tokens during training.

transferring knowledge and robustness from a larger teacher model. Additionally, despite the higher loss values of the ViT, it still performs at par with the distilled models at inference. This is likely due to the small image sizes in CIFAR-10. As the image size increases, with the amount of data kept constant, the distilled models would outperform the baseline by a significant margin. We also analyzed the individual token losses in both models and found that all tokens converge similarly, except the newly introduced corrupt token, which maintains relatively higher loss values. This indicates that the corrupt token remains more sensitive to corruption-specific signals, allowing it to specialize in modeling degraded features without disrupting the learning of other tokens.

5 Conclusion

To conclude, our study demonstrates that knowledge distillation, particularly soft-label distillation is a highly effective approach for training models from scratch on corrupted datasets when guided by a robust teacher. We also find that incorporating corruption awareness improves model performance without significantly increasing model size, and that the method of integrating this corruption information plays a critical role in its effectiveness. Additionally, we observe that heavy data augmentation on already corrupted inputs can degrade performance, likely due to compounding distribution shifts. Finally, in setups involving corruption-aware training, hard-label distillation proves more effective on corrupted data, whereas soft-label distillation excels when used independently, as it tends to smooth out corruption-specific signals captured by the corrupt token.

References

1. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.
2. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., & Jégou, H. (2021, July). Training data-efficient image transformers & distillation through attention. In International conference on machine learning (pp. 10347-10357). PMLR.
3. Wang, S., Veldhuis, R., Brune, C., & Strisciuglio, N. (2023). A Survey on the Robustness of Computer Vision Models against Common Corruptions. arXiv preprint arXiv:2305.06024.
4. Guo, Y., Stutz, D., & Schiele, B. (2023). Improving robustness of vision transformers by reducing sensitivity to patch corruptions. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4108-4118).
5. Mao, X., Qi, G., Chen, Y., Li, X., Duan, R., Ye, S., ... & Xue, H. (2022). Towards robust vision transformer. In Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition (pp. 12042-12051).
6. Hendrycks, D., & Dietterich, T. (2019). Benchmarking neural network robustness to common corruptions and perturbations. arXiv preprint arXiv:1903.12261.
7. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84-90.
8. Vishnuvardhan, G., & Ravi, V. (2021). Face recognition using transfer learning on facenet: application to banking operations. In Modern Approaches in Machine Learning and Cognitive Science: A Walkthrough: Latest Trends in AI, Volume 2 (pp. 301-309). Cham: Springer International Publishing.
9. Ratnaparkhi, S. T., Singh, P., Tandasi, A., & Sindhwani, N. (2021, September). Comparative analysis of classifiers for criminal identification system using face recognition. In 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO) (pp. 1-6). IEEE.
10. Chen, C., Seff, A., Kornhauser, A., & Xiao, J. (2015). Deepdriving: Learning affordance for direct perception in autonomous driving. In Proceedings of the IEEE international conference on computer vision (pp. 2722-2730).
11. Zhang, J., Wu, X., & Sheng, V. S. (2016). Learning from crowdsourced labeled data: a survey. Artificial Intelligence Review, 46, 543-576.
12. Maia Polo, F., & Vicente, R. (2023). Effective sample size, dimensionality, and generalization in covariate shift adaptation. Neural Computing and Applications, 35(25), 18187-18199.

13. Zhou, K., Liu, Z., Qiao, Y., Xiang, T., & Loy, C. C. (2022). Domain generalization: A survey. IEEE transactions on pattern analysis and machine intelligence, 45(4), 4396-4415.
14. Recht, B., Roelofs, R., Schmidt, L., & Shankar, V. (2019, May). Do imagenet classifiers generalize to imagenet?. In International conference on machine learning (pp. 5389-5400). PMLR.
15. Volpi, R., & Murino, V. (2019). Addressing model vulnerability to distributional shifts over image transformation sets. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 7980-7989).
16. Modas, A., Rade, R., Ortiz-Jiménez, G., Moosavi-Dezfooli, S. M., & Frossard, P. (2022, October). Prime: A few primitives can boost robustness to common corruptions. In European Conference on Computer Vision (pp. 623-640). Cham: Springer Nature Switzerland.
17. Michaelis, C., Mitzkus, B., Geirhos, R., Rusak, E., Bringmann, O., Ecker, A. S., ... & Brendel, W. (2019). Benchmarking robustness in object detection: Autonomous driving when winter is coming. arXiv preprint arXiv:1907.07484.
18. Touvron, H., Cord, M., & Jégou, H. (2022, October). Deit iii: Revenge of the vit. In European conference on computer vision (pp. 516-533). Cham: Springer Nature Switzerland.
19. Ridnik, T., Ben-Baruch, E., Noy, A., & Zelnik-Manor, L. (2021). Imagenet-21k pretraining for the masses. arXiv preprint arXiv:2104.10972.
20. Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531.
21. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.
22. Howell, K., Wang, J., Hazare, A., Bradley, J., Brew, C., Chen, X., ... & Widdows, D. (2022, May). Domain-specific knowledge distillation yields smaller and better models for conversational commerce. In Proceedings of the Fifth Workshop on e-Commerce and NLP (ECNLP 5) (pp. 151-160).
23. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., . . . Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Advances in Neural Information Processing Systems 32 (pp. 8024–8035). Curran Associates, Inc. Retrieved from http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
24. Cubuk, E. D., Zoph, B., Shlens, J., & Le, Q. V. (2020). Randaugment: Practical automated data augmentation with a reduced search space. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops (pp. 702-703).
25. Zhong, Z., Zheng, L., Kang, G., Li, S., & Yang, Y. (2020, April). Random erasing data augmentation. In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, No. 07, pp. 13001-13008).
26. Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412.
27. Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412.
28. Huang, G., Sun, Y., Liu, Z., Sedra, D., & Weinberger, K. Q. (2016). Deep networks with stochastic depth. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14 (pp. 646-661). Springer International Publishing.

29. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1), 1929-1958.
30. Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017, July). On calibration of modern neural networks. In International conference on machine learning (pp. 1321-1330). PMLR.
31. Liu, J., Wang, Z., Ma, L., Fang, C., Bai, T., Zhang, X., ... & Chen, Z. (2024). Benchmarking Object Detection Robustness against Real-World Corruptions. International Journal of Computer Vision, 132(10), 4398-4416.